Amazon Neptune

使用者指南



Amazon Neptune: 使用者指南

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任從何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 Neptune？？？ 1
最新更新 3
入門 25
什麼是圖形資料庫？ 25
為什麼使用圖表？ 26
圖形資料庫應用 26
图表查詢語言 28
查詢範例 28
線上 Neptune 課程 30
深入挖掘 30
使用圖形筆記本 30
使用海王星工作 31
啟用CloudWatch記錄 33
本地託管 34
遷移至 JupyterLab 3 35
Workbench 魔術 36
可變注入 37
常用查詢引數 38
% 種子 38
% 載入 39
% 載入識別碼 39
% 載入狀態 39
% 取消載入 39
% 狀態 39
%gremlin_ 狀態 39
% 打開狀態，或 %oc_ 狀態 39
%sparql_ 狀態 40
%stream_viewer 40
%圖形/筆記本設定 40
%圖形筆記本主機 40
%圖形筆記本版本/版本 40
% 圖形筆記本瀏覽選項 40
% 統計數字 41
% 摘要 41
%%圖形筆記本設定/設定 41
%%sparql 42
%% 麻黃 42
%% 開封密碼，或 %%oc 43
%%圖形筆記本_vis_選項 43
% 海王星 _ 毫升 43
% 海王星 _ 毫升 46
圖表視覺化 47
圖表介面 47
怪靈可視化 49
可視化 50
視覺化教程 50
海王星設置 52
數據庫實例類型 52
實例資源配置 52
t3 和 t4g 53
r4 執行個體 53
r5 執行個體 53
r5d 執行個體 53
r6g 執行個體 54

r6i 執行個體 54
x2g 執行個體 54
serverless 執行個體 54
建立資料庫叢集 54
先決條件 56
建立叢集 59
設定 VPC 60
新增子網路 61
建立子網路群組 61
建立安全群組 62
VPC 中的 DNS 62
連接到您的圖表 63
設定curl或 awscurl 63
連接方式 63
從虛擬私人雲端內部 63
從不同的 VPC 65
從私人網路 65
海王星安全 65
IAM 政策 66
VPC security groups (VPC 安全群組) 66
IAM 身分驗證 66
存取圖表 67
設定 curl 63
查詢語言 67
使用小精靈 67
使用開放密碼 71
使用 RDF/SPLQL 71
載入 資料 71
監控海王星 72
故障診斷和最佳實務 72
全域資料庫 73
概要 73
優點 74
限制 74
設定 74
組態需求 75
建立全域資料庫 75
使用現有的資料庫叢集做為主要的 76
新增次要區域 77
連接 77
管理 Neptune 全域資料庫 78
移除叢集 78
刪除全球資料庫 78
修改全球資料庫 78
的容錯移轉 79
分離並提升 79
管理規劃的容錯移轉 80
監視 Nal Database ase ase 81
海王星概述 82
標準合規 83
小鬼符合標準 83
符合標準 92
符合開放密碼標準 95
圖形資料模型 104
字典 104
索引策略 104
怪靈數據模型 106

查閱快取 107
查詢快取的使用案例 107
使用快取 107
交易語意 109
隔離層級 109
Neptune 隔離層級 109
交易範例 113
例外狀況和重試 115
叢集和執行個體 117
主要資料庫執行個體 117
僅供讀取複本執 117
調整實例大 118
監控執行個 118
儲存、可靠性和可用性 119
分配 119
儲存帳單 119
儲存最佳做法 119
可靠性和高可用性 120
端點連線 121
叢集端點 121
讀取器端點 121
執行個體端點 122
自訂端點 122
端點考量 122
使用自訂端點 123
自訂 queryId 126
使用 HTTP 標頭。 126
使用 SPARQL 查詢提示 126
使用 queryId 檢查狀態 126
實驗室模式 127
使用實驗室模式 127
OSGP 指數 128
交易語意 128
海王星 DFE 发动机 129
控制 DFE 的使用 129
DFE 執行的查詢 130
DFE 統計資料 131
大小限制 131
統計狀態 131
停用自動運算 133
重新啟用自動計算 133
手動產生統計 134
監控統計 134
IAM 身分驗證 136
刪除統計 136
常見錯誤 136
圖形摘要 API 138
擷取圖表摘要 138
mode 參數 139
屬性圖摘要 139
RDF 圖表摘要 140
PG 摘要範例 141
RDF 摘要範例 143
IAM 和圖表摘要 146
常見的圖形摘要錯誤 146
JDBC 連線能力 148
入門 148

使用畫面 149
疑難排解 151
海王星引擎更新 152
安全性 153
資料保護 153
亞馬遜 VPC 保護 154
傳輸中加密 154
靜態加密 155
IAM 概觀 157
不同角色 158
使用身分 158
啟用 IAM 160
連線與簽署 160
EC2 先決條件 161
使用命令行 161
Gremlin 主控台 163
Gremlin Java 165
SPARQL Java (RDF4J 和 Jena) 167
使用 Node.js 的 SPARQL 168
Python 範例 169
使用 IAM 政策 176
身分型政策 177
服務控制政策 (SCP) 177
海王星控制台訪 177
連接政策 177
IAM 政策的類型 177
使用條件鍵 178
IAM 功能支援 178
IAM 政策限制 179
受管政策 179
條件索引鍵 187
行政政策聲明 187
資料存取政策聲明 203
海王星服務連結角色 215
角色許可 215
建立服務連結角色 216
編輯服務連結角色 217
刪除服務連結角色 217
暫時登入資料 218
使用 AWS CLI 取得登入資料 219
設定 Lambda 221
設置亞馬遜 EC2 222
記錄和監控 223
合規驗證 223
復原能力 224
遷移至 Neptune 225
從遷移 226
一般資訊 226
遷移的準備 228
佈建設施 231
資料移轉 233
應用遷移應 237
Neptune 相容性 239
密碼重寫 242
移轉資源 247
移轉來源 TinkerPop 248
從移移移移移移移移 249

用AWS DMS於移轉 ... 250
從大火影遷移 ... 251
Neptune 相容 ... 251
此基礎設施 ... 251
匯出資料 ... 252
建立 Amazon S3 儲存貯體 ... 253
匯入資料 ... 254
載入資料 ... 255
海王星散裝載機 ... 255
IAM 角色和亞馬遜 S3 存取 ... 256
資料格式 ... 261
載入範例 ... 270
最佳化大量負載 ... 273
載入器參考 ... 274
使用 DMS 載入資料 ... 292
GraphMappingConfig ... 292
複製到海王星 ... 295
查詢 ... 299
查詢佇列 ... 299
找出佇列中有多少查詢 ... 299
查詢逾時時間 ... 299
Gremlin ... 300
安裝小精靈控制台 ... 301
HTTPS REST ... 304
Java ... 305
Python ... 313
.NET ... 314
Node.js ... 316
Go ... 317
查詢提示 ... 319
查詢狀態 ... 324
取消查詢 ... 326
基於小鬼腳本的會話 ... 326
小鬼交易 ... 328
使用 Gremlin API ... 329
快取查詢結果 ... 330
從 3.6.x 向前開始的高效提升者 ... 334
3.6.x 之前的高效提升者 ... 338
小精靈 explain ... 347
小精靈與 DFE ... 381
开放密码 ... 382
小精靈與開放密碼 ... 382
使用開放密碼 ... 383
狀態端點 ... 383
端點 ... 386
使用博爾特通訊協定 ... 389
參數化範例 ... 403
資料模型 ... 404
开放密码 explain ... 404
交易 ... 415
限制 ... 421
例外狀況 ... 421
SPARQL ... 422
RDF4J 主控台 ... 423
RDF4J Workbench ... 425
Java ... 426
HTTP API ... 428

查詢提示 437
描述和默認圖形 447
查詢狀態 449
取消查詢 450
图形存储协议 452
SPARQL explain 453
擴充功能 SERVICE 473
視覺化工具 476
圖形資源管理器 476
法蓋特上的圖形資源管理器 476
示範 478
湯姆索亞軟件 480
劍橋情報局 482
石墨 484
隱喻 485
G.V () 488
匯出資料 489
海王星-匯出 490
海王星-出口服務 491
安裝該服務 491
啟用對 Neptune 的存取 493
啟用對海王星-出口的訪問 493
執行匯出工作 493
監控任務任務 494
取消任務 495
海王星-出口實用程序 496
先決條件 496
運行海王星-出口 497
命令範例 497
匯出的檔案 499
匯出結果 500
command 501
輸出 3 路徑 501
工作大小 501
params 502
附加參數 502
params 503
篩選範例 510
疑難排解 513
管理海王星 515
海王寶藍/綠解決方 516
海王幕藍/綠的必備 517
限制 518
使用主控台 518
使用 CLI 519
不執行 AWS CloudFormation 520
監控海王星藍綠解決方案 520
切換至更新的叢集 522
清除 523
最佳實務 523
疑難排解 523
IAM 使用者許可 525
服務角角角角角角 525
建立新的 IAM 使用者使用者 525
參數群組 527
編輯參數群組 528
建立參數群組 528

參數 530
海王星啟用審計日誌 530
海王星啟用 _ 慢速查詢日誌 530
海王星慢查詢日誌閾值 531
海王星實驗室模式 531
海王星查詢超時 531
海王星溪 531
海王星流 _ 過期 _ 日 532
海王星查找緩存 532
海王星 _ 自動擴展 _ 配置 532
海王星 _ 美国角色 532
海王星端點 533
海王星查詢引擎 533
海王星查詢超時 533
海王星結果緩存 533
海王星強制執行 533
使用主控台啟動 534
停用和啟動叢集 537
停止和啟動概觀 537
停止叢集 537
啟動資料庫叢集 538
快速重設 API 539
使用 IAM 身份驗證 541
該 % 數據庫重置魔術 541
常見錯誤 542
建立一個複本 543
修改資料庫叢集 544
修改執行個體 544
新增 Neptune 複本 546
效能和擴展 548
儲存體擴展 548
執行個體擴展； 548
讀取擴展 548
自動調整規模 549
自動調整規模和無伺服器 550
啟用自動縮放 550
移除自動縮放 552
維持資料庫叢集 553
維護時段 553
海王星更新 555
手動升級 555
自動升級 555
通過更新 CloudFormation 556
一點零至 556
預設值為 1.2.0.2 點 558
1. 1 至 2 點 (自訂) 559
1. 1 至 2 月 2 日 (混合) 561
複製資料庫叢集 564
限制 565
寫入時複製通訊協定 565
刪除來源資料庫 567
管理執行個體 568
T3 高載執行個體 569
修改執行個體 571
將 Neptune 資料庫執行個體 573
將資料庫執行個體重新開機 574
刪除資料庫執行個體 575

無伺服器 577
Serverless 577
Constraints 577
容量擴充 578
設定最小值 579
設定最大值 579
估計容量 579
其他組態 580
混合配置 580
設定促銷層級 580
讀寫器對齊 581
避免非常大的逾時值 581
最佳化您的組態 581
使用 Serverless 581
建立無伺服器叢集 582
轉換為 Serverless 582
修改容量範圍 583
將執行個體變更為已佈建 584
監控 584
Neptune 585
使用串流 586
啟用串流 586
停用串流 586
呼叫串流 API 587
串流回應 588
串流例外狀況 589
串流記錄格式 590
PG_JSON 590
RDF-NQUADS 592
串流範例 592
AT_SEQUENCE_NUMBER 範例 593
AFTER_SEQUENCE_NUMBER 範例 594
TRIM_HORIZON 範例 594
最新的例子 594
壓縮範例 595
Neptune 到海王星的複製設置 596
選擇一個AWS CloudFormation範本 596
新增堆疊細節 598
執行範本 600
更新串流輪詢器 600
災難復原串流 600
複寫設定 601
其他考量 603
海王星全文搜索 604
全文檢索搜尋設定 604
設定您地區的全文檢索 605
新增堆疊細節 606
執行範本 608
更新串流輪詢器 608
啟用全文搜尋的兩種方法 609
暫停流輪詢器 609
自訂OpenSearch欄位 610
使用精細的存取控制進行查詢 610
光世語法的使用 610
Neptune 全文搜尋資料模型 611
範例文件 612
小精靈範例文件 613

全文檢索搜尋 614
非字符串索引 616
堆疊 617
排除欄位 618
資料類型對映 619
數據類型驗證 621
範例查詢 624
Full-text-search 查詢執行 626
SPARSQL 全文檢索搜尋查詢範例 627
比對查詢 627
字首 627
fuzzy 628
term 628
query_string 628
simple_query_string 628
按字符串字段排序 629
按非字符串字段排序 629
依 ID 排序 629
按標籤排序 630
按文檔類型排序 630
Lucene 語法 630
範例 Gremlin 全文檢索搜尋 630
基本match 631
match 631
fuzzy 631
query_string模糊 632
query_string正規則 632
混合查詢 632
全文檢索搜尋 632
query_string, '+' 和 '-' 633
query_string、AND 和 OR 634
term 634
prefix 634
Lucene 語法 634
現代, TinkerPop 圖表 635
按字符串字段排序 636
按非字符串字段排序 636
按 ID 字段排序 636
按標籤欄位排序 636
按document_type字段排序 636
疑難排解和指標 637
讀取疑難排 637
寫入疑難排 637
AWS Lambda 函式 638
小鬼連 WebSocket 接 638
小鬼 StambStStStStStStSt 638
寫請求建議 639
讀取要求建議 639
冷啟動延遲 639
建立 Lambda 函數 640
Lambda 函數範例 642
Java 範例 642
JavaScript 例子 645
Python 範例 648
Neptune Leptune 652
Neptune ML DEDEDE 652
Neptune ML DE程 653

設定使用 AWS CloudFormation 654
手動設定 657
使用 AWS CLI 662
使用 Neptune ML 665
開始工作流 665
處理不斷變化的 667
更新模型人工因素 667
自訂模型工作流 668
實體選擇 669
用於資料處理 669
用於模型訓練和模型轉換 669
對於推論端點 669
資料匯出 670
海王星-出口實例 670
參數設定 671
附加參數 671
targets 674
特色 678
範例 684
資料處理 694
管理資料處理 694
更新的處理 694
功能編碼 695
編輯訓練資料檔案 700
模型訓練 707
模型與訓練 709
自訂超參數 711
訓練最佳實務 719
模型轉換 721
增推論 721
任何工作的模型轉換 721
模型成品 722
不同任務的成品 722
建立新成品 722
自定義模型 724
自訂模型 725
自訂模型開發 727
Inference endpoint 730
管理論端點 730
推論 731
小鬼推論查詢 732
推論查詢 748
Neptune API 752
資料處理命令 753
模型訓練指令 757
模型轉換命令 761
端點指令 765
例外狀況 768
限制 769
SageMaker 限制 769
監控海王星 770
執行個體狀態 770
範例輸出 772
使用 CloudWatch 772
使用主控台 773
使用 AWS CLI 773
使用 CloudWatch API 773

監控實例性能 774
海王星度量 774
海王星尺寸 780
使用海王星稽核記錄 781
啟用稽核日誌 781
檢視稽核日誌 781
稽核日誌詳細資訊 781
海王星CloudWatch日誌 782
將記錄檔發佈到CloudWatch記錄檔 (主控台) 783
將稽核記錄發佈至CloudWatch記錄檔 (CLI) 783
將慢速查詢記錄檔發佈至記CloudWatch錄檔 (CLI) 783
監控日誌事件 784
筆記本CloudWatch日誌 784
慢速查詢記錄檔 785
在 主控台中檢視 日誌 785
慢速查詢記錄檔 785
info模式屬性 786
debug模式屬性 787
輸出範例 788
使用記錄海王星 API 呼叫 AWS CloudTrail 789
海王星信息 CloudTrail 790
瞭解海王星記錄檔項目 790
事件通知 791
類別和訊息 792
訂閱事件 798
管理訂閱 799
標記海王星資源 799
標記概觀 800
在主控台中標記 801
使用 CLI 標記 802
使用 API 標記 802
使用 ARN 803
備份與還原 806
備份與還原概觀 807
容錯能力 807
備份 807
Backup 指標 808
還原資料 808
備份時段 809
建立快照 810
使用主控台 810
從快照還原 811
重要的還原 811
還原 812
複製快照 813
限制 813
快照副本保留 813
加密 814
跨區域快照複製 814
在主控台上複製快照 814
使用 AWS CLI 複製快照 815
共享快照 817
加密快照 817
共享 819
刪除快照 821
使用主控台 821
使用 AWS CLI 821

使用 Neptune API 821
最佳實務 822
基本操作準則 823
安全性 824
避免不同的實例大小 824
避免大量負載重新啟動 824
如果你有很多謂詞 824
避免長時時間執行的 824
使用指標 825
調校查詢 825
負載平衡 825
使用暫時執行個體 826
調整實體大小 826
工作中斷錯誤 826
Gremlin (一般) 827
GLV 執行差異 827
優化更新查詢 828
多執行緒寫入 828
清除記錄 828
datetime( ) 829
原生日期與時間 829
小鬼 (Java 客戶端) 830
使用最新版的用戶端 830
重複使用客戶端對象 830
用於讀取和寫入的個別用戶端 830
多個複本端點 831
完成後關閉用戶端 831
容錯移轉後新連線 831
設定 maxInProcessPerConnection = maxSimultaneousUsagePerConnection 831
以位元組碼傳送查詢 831
完全使用查詢結果 832
大量加入頂點和邊 832
停用 DNS 快取 833
每個查詢逾時 833
解決早期版本中的錯誤 834
故障排除 834
開放密碼和螺栓 835
偏好直接邊緣 835
沒有並發交易查詢 835
容錯移轉後重新 836
重新使用驅動程序對象 836
Lambda 接處理 836
關閉動因物件 836
使用明確交易模式 836
重試邏輯 838
SPARQL 840
查詢所有具名圖表 840
指定要載入的具名圖表 841
過濾與值對比 841
海王星極限 843
區域 843
叢集磁碟區大小 843
中國地區 843
實例大小 844
每個 帳戶 844
需要 VPC 844
需要 SSL 844

可用區域與子網路群組 844
有效負載 845
Gremlin 845
無空字元 845
SPARQL UPDATE LOAD 845
身分驗證與存取 845
WebSockets 限制 846
性質和標示 847
大量載入 847
將 Neptune 848
Neptune 錯誤 849
引擎錯誤代碼 849
錯誤格式 849
查詢錯誤 849
IAM 錯誤 852
API 錯誤 853
載入器錯誤 854
引擎版本 856
引擎版本壽命 857
海王星引擎更新 857
升級前先測試 858
拍攝手動快照 858
升級程序 858
升級至 1.2.0.0 或更高版本 859
引擎版本號碼 859
使用主控台手動升級 860
使用 CLI 進行次要升級 860
使用 CLI 進行主要升級 860
發行版本:1.2.1.0 861
修補程式版本 862
新功能 862
改善項目 862
修正的缺陷 863
支援的查詢語言版本 863
升級途徑 864
Upgrading (正在升級) 864
發行版本: 865
發行版本: 868
已發行版本: 870
修補程式版本 871
新功能 871
改善項目 871
支援的查詢語言版本 871
升級途徑 871
Upgrading (正在升級) 871
發行版本: 872
發行版本: 874
發行版本: 877
已發行版本: 879
修補程式版本 880
新功能 880
改善項目 880
修正的缺陷 880
支援的查詢語言版本 880
升級途徑 880
Upgrading (正在升級) 880
維護版本: 882

已發行版本: ... 884
修補程式版本 ... 884
新功能 ... 884
改善項目 ... 885
修正的缺陷 ... 885
支援的查詢語言版本 ... 886
升級途徑 ... 887
Upgrading (正在升級) ... 887
版本發行: ... 888
發行版本: ... 891
已發行版本: ... 894
修補程式版本 ... 894
新功能 ... 895
改善項目 ... 895
修正的缺陷 ... 896
支援的查詢語言版本 ... 896
升級途徑 ... 896
Upgrading (正在升級) ... 897
發行版本: ... 898
發行版本: ... 901
版本發行日期: ... 904
發行版本: ... 907
發行版本: ... 910
維護版本:1.1.1.2 版本 ... 912
已發行版本: ... 915
修補程式版本 ... 916
新功能 ... 916
改善項目 ... 916
修正的缺陷 ... 917
支援的查詢語言版本 ... 917
升級途徑 ... 917
Upgrading (正在升級) ... 918
維護版本: ... 919
維護版本: ... 921
已發行版本: ... 924
修補程式版本 ... 924
新功能 ... 924
改善項目 ... 924
修正的缺陷 ... 925
支援的查詢語言版本 ... 925
升級途徑 ... 925
Upgrading (正在升級) ... 925
維護版本:4 月 5 日 ... 926
發行版本: ... 928
發行版本: ... 929
已發行版本: ... 931
修補程式版本 ... 931
新功能 ... 931
改善項目 ... 931
修正的缺陷 ... 932
支援的查詢語言版本 ... 932
升級途徑 ... 932
Upgrading (正在升級) ... 932
維護版本:5 月 5 日 ... 933
發行版本: ... 935
發行版本: ... 937
已發行版本: ... 938

發行版本: 938
發行版本: 939
發行版本: 939
發行版本: 940
發行版本: 943
版本版本:1.0.4.1 943
修補程式版本 943
新功能 943
改善項目 943
修正的缺陷 943
支援的查詢語言版本 944
升級途徑 944
Upgrading (正在升級) 944
發行版本: 945
發行版本: 946
發行版本： 950
修補程式版本 950
新功能 950
改善項目 950
修正的缺陷 951
支援的查詢語言版本 951
升級途徑 951
Upgrading (正在升級) 951
發行版本: 952
發行版本: 954
修補程式版本 954
新功能 954
改善項目 954
修正的缺陷 955
支援的查詢語言版本 955
升級途徑 955
Upgrading (正在升級) 955
發行版本: 956
版本:1.0.0.0.0.0.2 958
發行版本： 960
修補程式版本 960
改善項目 960
修正的缺陷 960
支援的查詢語言版本 961
升級途徑 961
Upgrading (正在升級) 961
發行版本: 962
版本:1.0.2.2.R5 964
發行版本:1.0.2.2.R4 965
發行版本:1.0.2.2.R3 967
版本:1.0.2.2.R2 967
發行版本: 969
修補程式版本 969
新功能 969
改善項目 969
修正的缺陷 970
支援的查詢語言版本 970
升級途徑 970
Upgrading (正在升級) 970
版本:1.0.2.1.R6 971
發行版本: 973
發行版本:1.0.2.R 973

版本: ... 975
版本: ... 976
發行版本: ... 978
重要提示：此引擎版本現已棄用 ... 978
修補程式版本 ... 978
新功能 ... 978
支援的查詢語言版本 ... 978
升級途徑 ... 978
Upgrading (正在升級) ... 979
發行版本: ... 980
版本: ... 981
已發行版本: ... 983
重要：此引擎版本現已棄用 ... 983
改善項目 ... 983
修正的缺陷 ... 983
支援的查詢語言版本 ... 984
發行版本: ... 984
重要：此引擎版本現已棄用 ... 984
修正的缺陷 ... 984
支援的查詢語言版本 ... 984
發行版本: ... 984
重要：此引擎版本現已棄用 ... 984
二 ○○ 五零二年版本 ... 984
版本 1.0.0.0.463.0 ... 985
版本 1.0.0.200457.0 (2019 年 9 月 19 日) ... 985
二 ○○ 三七年版本 ... 986
發行版本 1.0.0.200366.0 (2019 年 7 月 26 日) ... 987
發行版本 1.0.0.200348.0 ... 988
舊版 ... 988
Neptune 管理 API ... 996
叢集 ... 1000
DBCluster ... 1001
資料庫 ClusterMember ... 1004
資料庫 ClusterRole ... 1004
ClusterPendingModifiedValues ... 1004
CreateDBCluster ... 1005
DeleteDBCluster ... 1012
ModifyDBCluster ... 1017
StartDBCluster ... 1023
StopDBCluster ... 1027
AddRoleTo数据库集群 ... 1031
RemoveRoleFrom数据库集群 ... 1031
FailoverDBCluster ... 1032
PromoteReadReplica数据库集群 ... 1036
DescribeDBClusters ... 1040
全球資料庫 ... 1041
GlobalCluster ... 1041
CreateGlobalCluster ... 1042
DeleteGlobalCluster ... 1044
ModifyGlobalCluster ... 1045
DescribeGlobalClusters ... 1047
FailoverGlobalCluster ... 1048
RemoveFromGlobalCluster ... 1050
執行個體 ... 1051
DBInstance ... 1051
資料庫InstanceStatusInfo ... 1054
可訂購數據庫InstanceOption ... 1055

驗證InstanceModificationsMessage 1056
CreateDBInstance 1056
DeleteDBInstance 1065
ModifyDBInstance 1069
RebootDBInstance 1078
DescribeDBInstances 1082
DescribeOrderable資料庫InstanceOptions 1083
DescribeValid資料庫InstanceModifications 1084
參數 1084
參數 1085
資料庫ParameterGroup 1086
資料庫ClusterParameterGroup 1086
資料庫ParameterGroupStatus 1087
複製資料庫ParameterGroup 1087
複製資料庫ClusterParameterGroup 1088
創建數據庫ParameterGroup 1090
創建數據庫ClusterParameterGroup 1091
刪除資料庫ParameterGroup 1092
刪除資料庫ClusterParameterGroup 1093
修改資料庫ParameterGroup 1093
修改資料庫ClusterParameterGroup 1094
重置資料庫ParameterGroup 1095
重置資料庫ClusterParameterGroup 1096
DescribeDBParameters 1097
描述 BParameterGroups 1098
描述 BClusterParameters 1099
描述 BClusterParameterGroups 1100
DescribeEngineDefaultParameters 1101
DescribeEngineDefaultClusterParameters 1102
子網 1102
子網 1103
資料庫SubnetGroup 1103
創建數據庫SubnetGroup 1104
刪除資料庫SubnetGroup 1105
修改資料庫SubnetGroup 1106
描述 BSubnetGroups 1107
快照 1108
資料庫 ClusterSnapshot 1108
資料庫 ClusterSnapshotAttribute 1110
資料庫 ClusterSnapshotAttributesResult 1110
創建數據庫 ClusterSnapshot 1110
刪除資料庫 ClusterSnapshot 1113
複製資料庫 ClusterSnapshot 1115
修改資料庫 ClusterSnapshotAttribute 1117
恢復 ClusterFromSnapshot 1119
恢復 ClusterToPointInTime 1125
描述 B ClusterSnapshots 1131
描述 B ClusterSnapshotAttributes 1133
事件 1134
事件 1134
EventCategoriesMap 1134
EventSubscription 1135
CreateEventSubscription 1136
DeleteEventSubscription 1138
ModifyEventSubscription 1139
DescribeEventSubscriptions 1141
AddSourceIdentifierToSubscription 1142

RemoveSourceIdentifierFromSubscription 1143
DescribeEvents 1144
DescribeEventCategories 1146
其他 1146
CloudwatchLogsExportConfiguration 1147
資料庫EngineVersion 1147
EngineDefaults 1148
PendingMaintenanceAction 1148
PendingModifiedValues 1149
PendingCloudwatchLogsExports 1149
ResourcePendingMaintenanceActions 1150
UpgradeTarget 1150
Tag 1150
AddTagsToResource 1151
ListTagsForResource 1151
RemoveTagsFromResource 1152
ApplyPendingMaintenanceAction 1152
DescribePendingMaintenanceActions 1153
描述 BEngineVersions 1154
資料類型 1155
AvailabilityZone 1156
資料庫 SecurityGroupMembership 1156
DomainMembership 1156
DoubleRange 1156
端點 1157
篩選條件 1157
範圍 1157
無伺服器 V2 ScalingConfiguration 1158
無伺服器 V2 ScalingConfigurationInfo 1158
時區 1158
VpcSecurityGroupMembership 1158
API 故障 1159
AuthorizationAlreadyExistsFault 1160
AuthorizationNotFoundFault 1161
AuthorizationQuotaExceededFault 1161
CertificateNotFoundFault 1161
資料庫ClusterAlreadyExistsFault 1161
資料庫ClusterNotFoundFault 1162
資料庫ClusterParameterGroupNotFoundFault 1162
資料庫ClusterQuotaExceededFault 1162
資料庫ClusterRoleAlreadyExistsFault 1162
資料庫ClusterRoleNotFoundFault 1162
資料庫ClusterRoleQuotaExceededFault 1163
資料庫ClusterSnapshotAlreadyExistsFault 1163
資料庫ClusterSnapshotNotFoundFault 1163
資料庫InstanceAlreadyExistsFault 1163
資料庫InstanceNotFoundFault 1164
資料庫LogFileNotFoundFault 1164
資料庫ParameterGroupAlreadyExistsFault 1164
資料庫ParameterGroupNotFoundFault 1164
資料庫ParameterGroupQuotaExceededFault 1164
資料庫SecurityGroupAlreadyExistsFault 1165
資料庫SecurityGroupNotFoundFault 1165
資料庫SecurityGroupNotSupportedFault 1165
資料庫SecurityGroupQuotaExceededFault 1165
資料庫SnapshotAlreadyExistsFault 1166
資料庫SnapshotNotFoundFault 1166

資料庫SubnetGroupAlreadyExistsFault 1166
資料庫SubnetGroupDoesNotCoverEnough可用區 1166
資料庫SubnetGroupNotAllowedFault 1166
資料庫SubnetGroupNotFoundFault 1167
資料庫SubnetGroupQuotaExceededFault 1167
資料庫SubnetQuotaExceededFault 1167
資料庫UpgradeDependencyFailureFault 1167
DomainNotFoundFault 1168
EventSubscriptionQuotaExceededFault 1168
GlobalClusterAlreadyExistsFault 1168
GlobalClusterNotFoundFault 1168
GlobalClusterQuotaExceededFault 1168
InstanceQuotaExceededFault 1169
資料庫不足ClusterCapacityFault 1169
資料庫不足InstanceCapacityFault 1169
InsufficientStorageClusterCapacityFault 1169
无效ClusterEndpointStateFault 1170
无效ClusterSnapshotStateFault 1170
无效ClusterStateFault 1170
无效InstanceStateFault 1170
无效ParameterGroupStateFault 1170
无效SecurityGroupStateFault 1171
无效SnapshotStateFault 1171
无效SubnetGroupFault 1171
无效SubnetGroupStateFault 1171
无效SubnetStateFault 1172
InvalidEventSubscriptionStateFault 1172
InvalidGlobalClusterStateFault 1172
InvalidOptionGroupStateFault 1172
InvalidRestoreFault 1172
InvalidSubnet 1173
无效NetworkStateFault 1173
公里KeyNotAccessibleFault 1173
OptionGroupNotFoundFault 1173
PointInTimeRestoreNotEnabledFault 1174
ProvisionedIopsNotAvailableInAZ 錯誤 1174
ResourceNotFoundFault 1174
SNSInvalidTopicFault 1174
SNSNoAuthorizationFault 1174
SNSTopicArnNotFoundFault 1175
SharedSnapshotQuotaExceededFault 1175
SnapshotQuotaExceededFault 1175
SourceNotFoundFault 1175
StorageQuotaExceededFault 1176
StorageTypeNotSupportedFault 1176
SubnetAlreadyInUse 1176
SubscriptionAlreadyExistFault 1176
SubscriptionCategoryNotFoundFault 1176
SubscriptionNotFoundFault 1177
..... mclxxviii

什麼是 Amazon Neptune？？？？

Amazon Neptune 是快速、可靠、全受管的圖形資料庫服務，可讓您輕鬆建置和執行搭配高度連線資料集使用的應用程式。專為高效能圖 Neptune 資料庫引擎所打造的高效能圖形資料庫引擎。此引擎最適合存放的數十億筆關係最佳化，且查詢圖形時只會有數毫秒的延遲。Neptune 支持流行的屬性圖查詢語言阿帕奇TinkerPop格林和 Neo4j 的開放密碼，以及 W3C 的 RDF 查詢語言，SPARQL。這可讓您建立查詢，以有效率地瀏覽高度連線的資料集。Neptune 為圖形使用案例提供支援，例如推薦引擎、詐騙偵測、知識圖表、藥物發現和網路安全性。

Neptune 具有高可用性，具有僅供讀取複本、point-in-time復原、持續備份至 Amazon S3，以及跨可用區域的複寫。Neptune 提供資料安全功能，且支援靜態及傳輸中加密的加密。Neptune 完全受管，因此您不再需要擔心硬體佈建、軟體修補、設定、組態或備份等資料庫管理工作。

為了解有關使用 Amazon Neptune 的資訊，我們建議您從下列各節開始著手：？？？？？？？

- Amazon Neptune 入門 (p. 25)
- 亞馬遜海王星功能概述 (p. 82)

如果您不熟悉圖形，或尚未準備好投資於完整的 Neptune 生產環境，請造訪我們的入門 (p. 25)主題，瞭解如何使用 Neptune Jupyter 筆記本進行學習和開發，而不會產生成本。

此外，在開始設計資料庫之前，我們建議您參考使用 Graph Database 的GitHub存放庫AWS參考架構，您可以在其中告知您有關圖形資料模型和查詢語言的選擇，並瀏覽參考部署架構的範例。

關鍵服務元件

- 主要資料庫執行個體 – 支援讀寫操作，並對叢集磁碟區執行所有資料修改。每個 Neptune 資料庫叢集都有一個主要資料庫執行個體，負責寫入 (亦即載入或修改) 圖形資料庫內容。
- Neptune 複本-連接到與主要資料庫執行個體相同的儲存磁碟區，且僅支援讀取操作。除了主要資料庫執行個體之外，每個 Neptune 資料庫叢集最多可以擁有 15 個 Neptune 複本。這可透過將 Neptune 複本置放在不同可用區及來自讀取用戶端的分散負載，來改善可用性。
- 叢集磁碟區-Neptune 資料會存放在叢集磁碟區，該磁碟區是為了可靠性和高可用性所設計的。叢集磁碟區包含跨單AWS一區域的多個可用區域的資料複本。因為您的資料會自動跨可用區域複寫，因此不僅耐用性高，且資料遺失的可能性也非常低。

支援開放式圖形 API

Amazon Neptune 支持這兩個屬性圖（小鬼和開放密碼）和 RDF 圖（SPARQL）的開放圖形 API。它提供這兩種圖形模型及其查詢語言的高效能。您可以選擇屬性圖（PG）模型，並訪問 OpenCypher 查詢語言和/或格林 (p. 300)查詢語言 (p. 382)相同的圖形。如果您使用 W3C 標準資源描述架構 (RDF) 模型，您可以使用標準 SPARQL 查詢 (p. 422)語言存取您的圖形。

高度安全

Neptune 為您的資料庫提供多層級的安全性。安全功能包括使用 Amazon VPC 進行網路隔離，以及使用您透過 AWS Key Management Service(AWS KMS) 建立和控制的金鑰靜態加密。在加密的 Neptune 執行個體上，基礎儲存體中的資料會加密，以及同一叢集中的自動備份、快照和複本。

全受管

使用 Amazon Neptune，您不必擔心硬體佈建、軟體修補、設定、組態或備份等資料庫管理任務。

您可以使用 Neptune 建立複雜的互動式圖形應用程式，以毫秒為單位查詢數十億個關係。高度連線資料的 SQL 查詢通常很複雜，而且不易調整效能。使用 Neptune，您可以使用流行的圖形查詢語言

Gremlin，OpenCypher 和 SPARQL 來執行功能強大的查詢，這些查詢易於編寫並在連接的數據上執行良好。此功能可大幅降低程式碼的複雜性，讓您可以快速建立處理關係的應用程式。

Neptune 的設計可提供超過 99.99% 的可用性。藉由將資料庫引擎與專為資料庫工作負載建置的 SSD 虛擬化儲存層緊密整合，藉此提升資料庫效能和可用性。Neptune 儲存具有容錯能力和自我修復能力。磁碟故障會在背景修復，而不會遺失資料庫可用性。Neptune 會自動偵測資料庫損毀並重新啟動，而不需要損毀復原或重建資料庫快取。如果整個執行個體發生故障，Neptune 會自動容錯移轉至最多 15 個僅供讀取複本之一。

亞馬遜海王星的變更和更新

下表說明亞馬遜海王星的重要變更。

變更	描述	日期
发动机版本 (p. 865)	截至二零二三年六月十三日，引擎版本 1.2.1.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.1.0. R3。	2023 年 6 月 13 日
第四版引擎 (p. 872)	截至二零二三年五月零八日，引擎版本 1.2.0.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.2. R4。	2023 年 5 月 8 日
海王星在中東（阿聯酋）地區啟動 (p. 3)	亞馬遜海王星現已在中東 (阿拉伯聯合大公國) (`me-central-1`) 區域推出。	2023 年 5 月 2 日
发动机版本 (p. 868)	截至二零二三年五月二日，引擎版本 1.2.1.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.1.0. R2。	2023 年 5 月 2 日
有關使用 Media2Cloud 採用 AI 驅動的視頻分析在海王星上構建知識圖表的博客文章 (p. 3)	請參閱使用 Mike Havey 的 Media2Cloud 提供的 AI 技術支援的視訊分析，在亞馬遜海王星上建立知識圖表。	2023 年 5 月 2 日
關於如何使用 Neptune DevOcean 建置弱點修復平台的部落格文章 (p. 3)	請參閱如何使用 Amazon Neptune 的雲端原生應用程式DevOcean建置弱點修復管理平台 (Gil Makmel) 和查爾斯·伊維。	2023 年 4 月 25 日
有關 Getir 如何使用 Neptune 構建欺詐檢測系統的博客文章 (p. 3)	了解 Getr 如何使用亞馬遜海王星和亞馬遜 DynamoDB 建立全面的欺詐檢測系統由伯凱伯克曼，馬赫穆特圖蘭，穆特魯波拉坎，烏穆特克拉克，雅赫茲揚科魯和埃斯拉卡亞巴利。	2023 年 4 月 6 日
有關 Wiz 如何使用 Neptune 重新構想雲端安全性的部落格文章 (p. 3)	看世界是一個圖表：奇才如何使用亞馬遜海王星的圖形重新構想雲安全由阿米·盧特瓦克和布拉德·貝比。	2023 年 3 月 31 日
发动机版本 (p. 874)	截至二零二三年三月二十七日，引擎版本 1.2.0.2.R3 正在普遍部署。請注意，新版本需要數天才能	2023 年 3 月 27 日

	在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.2. R3。	
有關 CSC 產生如何使用 Neptune 加強產品探索的部落格文章 (p. 3)	請參閱 CSC 產生如何運用 Amazon Neptune 的知識圖表，由鄭博柏、魯德拉和郭明莎提供的產品探索。	2023 年 3 月 21 日
引擎版本 (p. 861)	截至二零二三年三月零八日，引擎版本 1.2.1.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.1.0。	2023 年 3 月 8 日
關於探索海王星 TinkerPop 3.6.x 新功能的博客文章 (p. 3)	請參閱由斯蒂芬·馬萊特在亞馬遜海王星探索 Apache TinkerPop 3.6.x 的新功能。	2023 年 3 月 8 日
關於使用語義推理從 RDF 圖中推斷新事實的博客文章 (p. 3)	請參閱查爾斯·伊維和戴安娜馬克斯集成 RDFx 與亞馬遜海王星，使用語義推理從 RDF 圖中推斷出新的事實。	2023年2月20日
有關使用 Neptune 分析醫療保健 FHIR 資料的部落 (p. 3)	請參閱由阿琳娜·施米克爾透過亞馬遜海王星分析醫療保健 FHIR 資料。	2023 年 2 月 13 日
有關使用 Neptune ML 構建實時欺詐檢測解決方案的博客文章 (p. 3)	請參閱使用亞馬遜海王星 ML 製作的即時詐欺偵測解決方案，由華樹和索吉 Aeshina 提供。	2023 年 2 月 8 日
第 7 版引擎 (p. 898)	截至二零二三年一月二十三日，引擎版本 1.1.1.0.R7 現正普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R7。	2023 年 1 月 23 日
圖形資源管理器發布 (p. 476)	圖形資源管理器是一個開源的前端 Web 應用程序工具，用於可視化圖形數據。請參閱 https://github.com/aws/graph-explorer.	2023 年 1 月 3 日
維護發行版本 1.1.0.0.R3 (p. 919)	截至 2022 年 12 月 23 日，正在普遍部署引擎版本 1.1.0.0.0.R3 維護版本。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.0.0. R3。	2022 年 12 月 23 日
海王星工作台現在在亞馬遜 Linux 2 和 JupyterLab 3 上運行。 (p. 30)	海王星圖形筆記本現在在具有 JupyterLab 3 的亞馬遜 Linux 2 環境中運行。如需有關如何移至此新環境的詳細資訊，請參閱將 Neptune 筆記本從 Jupyter 遷移到 JupyterLab 3。	2022 年 12 月 21 日

關於電源建議和使用 IMDb 知識圖進行搜索的博客文章（第 3 部分）(p. 3)	請參閱電源建議和搜索使用 IMDb 知識圖 — 第 3 部分由 Divya Bhargavi，索吉·阿德西納，高拉夫·雷勒，卡蘭·辛德瓦尼，維迪亞薩加爾·拉維帕蒂和馬修·羅德斯。	2022年12月20日
關於電源建議和使用 IMDb 知識圖進行搜索的博客文章（第 2 部分）(p. 3)	請參閱電源建議和搜索使用 IMDb 知識圖-第 2 部分由馬修·羅德斯，索吉·阿德西納，迪維亞·巴爾加維，高拉夫·雷勒，卡蘭·辛德瓦尼和維迪亞薩加爾·拉維帕蒂。	2022年12月20日
關於電源建議和使用 IMDb 知識圖進行搜索的博客文章（第 1 部分）(p. 3)	請參閱電源建議和搜索使用 IMDb 知識圖 — 第 1 部分由高拉夫·雷勒，索吉·阿德西納，迪維亞·巴爾加維，卡蘭·辛德瓦尼，維迪亞薩加爾·拉維帕蒂和馬修·羅德斯。	2022年12月20日
Neptune 無伺服器現已在新AWS區域推出 (p. 577)	截至 2022-12-16，Neptune 無伺服器已在以下新AWS區域推出：加拿大（中部）、歐洲（斯德哥爾摩）、歐洲（法蘭克福）、亞太區域（新加坡）和亞太區域（雪梨）。請參閱所有提供海王星無伺服器服務的區域的 Amazon Neptune 無伺服器限制。	2022 年 12 月 16 日
发动机版本 (p. 877)	截至 2022 年 12 月 15 日，引擎版本 1.2.0.2.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.2. R2。	2022 年 12 月 15 日
发动机版本 (p. 888)	截至 2022 年 12 月 15 日，引擎版本 1.2.0.0.R3 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.0. R3。	2022 年 12 月 15 日
发动机版本 (p. 882)	截至 2022 年 12 月 13 日，引擎版本 1.2.0.1.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.1. R2。	2022 年 12 月 13 日
關於設計教育性大數據分析架構的部落格文章 AWS (p. 3)	請參閱AWS由 Lavanya Sood 設計教育大數據分析架構。	2022 年 12 月 13 日
關於圖形資料庫如何加強學習的部落格文章 (p. 3)	請參閱圖形資料庫如何透過 Lavanya Sood 提升學習效果。	2022 年 12 月 8 日
關於使AWS用 Glue 將 RDF 資料載入海王星的部落格文章 (p. 3)	請參閱使用麥克·哈維和法布里奇奧·納波利塔諾的AWS膠水將 RDF 數據加載到亞馬遜海王星。	2022 年 11 月 23 日

引擎版本 (p. 870)	截至 2022 年 11 月 20 日，引擎版本 1.2.0.2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2. 0.2。	2022年11月20日
引擎版本 (p. 879)	截至 2022 年 10 月 26 日，正在普遍部署引擎版本 1.2.0.1。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2. 0.1。	2022 年 10 月 26 日
關於使用 Neptune 偵測詐騙的部落格 (p. 3)	請參閱威爾遜唐，阿米爾·埃爾納加爾，馬蒂亞斯·龐斯，穆罕默德·阿扎姆，薩拉伯德·德斯潘德和路易斯·羅德里格斯蘇亞雷斯的亞馬遜海王星在交付英雄的欺詐檢測。	2022 年 10 月 26 日
海王星無伺服器的部落格文章 (p. 3)	請參閱亞馬遜海王星無伺服器簡介 — 全受管圖形資料庫，可為您的工作負載調整容量 Danilo Poccia。	2022 年 10 月 26 日
關於使用 Lambda 和 SPARQL 更新負載將事件驅動的 RDF 匯入海王星的部落格文章 (p. 3)	請參閱恩智浦如何使用約翰·沃克，昂諾·貝伊斯，查爾斯·伊維和爪哇德國王使用 AWS Lambda 和 SPARQL 更新負載執行事件驅動的 RDF 導入到亞馬遜海王星。	2022 年 10 月 20 日
发动机版本 (p. 891)	截至 2022 年 10 月 14 日，引擎版本 1.2.0.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2.0.0. R2。	2022 年 10 月 14 日
有關在 Neptune 中編碼多語言文本屬性的博客文章 (p. 3)	請參閱在 Amazon Neptune 中編碼多語言文字屬性，以訓練張佳妮的預測模型。	2022 年 10 月 14 日
有關 Neptune 資料存取自動測試的部落格文章 (p. 3)	請參閱格雷格·比格利用 Apache TinkerPop Grimlin 對亞馬遜海王星資料存取進行自動化測試。	2022年9月28日
发动机版本 (p. 901)	截至 2022 年 9 月 23 日，引擎版本 1.1.1.0.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R6。	2022 年 9 月 23 日
有關信息® 如何使用海王星的博客文章 (p. 3)	請參閱 Informatica® 雲端資料控管和目錄如何使用 Amazon 海王星取得提朱提圖約翰、迪帕克 Ram 和法魯克·阿什拉夫的知識圖表。	2022 年 9 月 20 日

關於利用海王星和湯姆索亞觀點發現金融欺詐的博客文章 (p. 3)	請參閱湯姆索亞軟件高級產品經理珍妮特 M. Six 的觀點，利用亞馬遜海王星和湯姆·索亞（Tom Sawyer）的觀點發現金融欺	2022 年 8 月 30 日
有關使用 Neptune 和 DGL 構建基於 GNN 的實時欺詐檢測解決方SageMaker案的博客文章 (p. 3)	請參閱使用 Amazon SageMaker、Amazon Neptune 和張健、王浩珠和朱孟心的深度圖庫建立以 GNN 為基礎的即時詐騙偵測解決方案。	2022 年 8 月 11 日
關於使用資源標籤停止和啟動 Neptune 環境資源的部落格文章 (p. 3)	請參閱 Kevin Phillips 的使用資源標籤自動化 Amazon 海王星環境資源的停止和啟動。	2022 年 8 月 1 日
關於為 Apache 做出貢獻的藝術家的博客文章 TinkerPop (p. 3)	看到超越代碼：誰為阿帕奇做出貢獻的藝術家斯蒂芬TinkerPop·馬萊特和凱特里納·湯普森。	2022 年 8 月 1 日
有關 Neptune 資料平面動作的精細存取控制的部落格文章 (p. 3)	請參閱亞馬遜海王星資料平面動作的精細存取控制 (Abhishek Mishra) 和 Ankit Gupta 的資料平面動作。	2022 年 7 月 29 日
有關海王星全球資料庫的部落 (p. 3)	請參閱 Navtanay Sinha 介紹亞馬遜海王星全球數據庫。	2022 年 7 月 27 日
引擎版本 (p. 884)	截至二零二一年七月二十一日，引擎版本 1.2.0.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.2. 0.0。	2022 年 7 月 21 日
发动机版本 (p. 904)	截至二零二一年七月二十一日，引擎版本 1.1.1.0.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R5。	2022 年 7 月 21 日
发动机版本 (p. 907)	截至 2022 年 6 月 23 日，引擎版本 1.1.1.0.R4 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R4。	2022 年 6 月 23 日
透過 Python 整合簡化圖形分析和機器學習工作流程 (p. 3)	您現在可以使用可簡化資料科學和機器學習工作流程的開放原始碼 Python 整合，在 Amazon Neptune 中存放的圖形資料上執行圖形分析和機器學習任務。請參閱 N AWSeptune 的資料牧馬人文件。	2022 年 6 月 7 日

发动机版本 (p. 910)	截至 2022 年 6 月 07 日，引擎版本 1.1.1.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R3。	2022 年 6 月 7 日
有關檢測虛假新聞的博客文章 (p. 3)	請參閱使用亞馬遜海王星 ML 使用圖形機器學習來檢測社交媒體虛假新聞，由 Hasan Shojaei 和 Sarita joshi 撰寫。	2022 年 5 月 19 日
有關使用 SQL 伺服器整合服務 (SSIS) 搭配海王星的部落格文章 (p. 3)	請參閱使用 SQL 伺服器整合服務 (SSIS) 和亞馬遜海王星由梅斯甘娜·葛姆利和郭梅麗莎發現資料的新見解。	2022 年 5 月 18 日
維護發行版本 1.1.1.0.R2 (p. 912)	截至 2022 年 5 月 16 日，引擎版本 1.1.1.0.R2 維護版本正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0. R2。	2022 年 5 月 16 日
維護發行版本 1.1.0.0.R2 (p. 921)	截至 2022 年 5 月 16 日，引擎版本 1.1.0.0.0.R2 維護版本正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.0.0. R2。	2022 年 5 月 16 日
維護發行版本 1.0.5.1.R4 (p. 926)	截至 2022 年 5 月 16 日，引擎版本 1.0.5.1.R4 維護版本正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.1. R4。	2022 年 5 月 16 日
維護發行版本 1.0.5.0.R5 (p. 933)	截至 2022 年 5 月 16 日，引擎版本 1.0.5.0.0.R5 維護版本正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.0. R5。	2022 年 5 月 16 日
關於海王星中 OpenCypher 一般可用性的博客文章 (p. 3)	請參閱宣布對亞馬遜海王星 OpenCypher 支持的一般可用性，由納瓦奈·辛納和戴夫·貝奇貝格。	2022 年 4 月 22 日
引擎版本 (p. 894)	截至 2022 年 4 月 19 日，引擎版本 1.1.1.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1.1.0。	2022 年 4 月 19 日

關於資料歷程的部落格文章 (p. 3)	請參閱使用 G AWS lue、亞馬遜海王星和雲形線建立資料湖的資料歷程，由 Khoa Nguyen、克里希瓦桑巴拉蘇布拉曼尼亞和 Rahul Shaurya 撰寫。	2022 年 4 月 1 日
引擎版本 1.1.0.0 的升級已重新啟用 (p. 3)	截至 2022 年 2 月 21 日，引擎版本 1.1. 0.0 的升級已暫時停用。它們現在已重新啟用。	2022 年 3 月 22 日
工程公用程式可靠性的部落格 (p. 3)	請參閱 Abhineet Parchure 撰寫的使用雲端、資料通知的電力系統模型來設計公用事業可靠性。	2022 年 3 月 22 日
海王星在非洲（開普敦）發射 (p. 3)	亞馬遜海王星現已在非洲（開普敦）（`af-south-1`）上市。不過，Neptune 工作台筆記型電腦支援在該區域的 Neptune 主控台上暫時停用。	2022 年 2 月 24 日
關於使用 OWL 模型導向圖形的部落格文章 (p. 3)	請參閱在亞馬遜海王星使用 OWL 模型驅動的圖表，由邁克·哈維。	2022 年 2 月 23 日
關於使用 Rhizomer 探索語義知識圖的博客文章 (p. 3)	請參閱羅伯托·加西亞（羅伯托·加西亞）使用亞馬遜海王星搭配根狀蟲探索語義知識圖，而不使用 SPARQL。	2022 年 2 月 22 日
關於繪製公用事業網格圖形的博客文章 (p. 3)	見圖形上的實用電網 AWS, 由鮑比·威爾遜和約瑟夫·啤酒.	2022 年 2 月 18 日
新的海王星 ML 文字功能編碼選項 (p. 3)	海王星現在支持FastText和句子 BERT 文本編碼進行培訓。請參閱海王星 ML 中的FastText功能和句子 BERT 功能在海王星 ML 中。	2022 年 2 月 15 日
關於OpenSearch與 Neptune 一起使用的地理空間查詢 (p. 3)	請參閱結合亞馬遜海王星和亞馬遜 OpenSearch服務以獲取地理空間查詢，羅斯·加貝和阿布拉什維諾德。	2022 年 2 月 1 日
關於使用亞馬遜 EKS 和海王星發現金融犯罪的博客文章 (p. 3)	請參閱金融犯罪發現使用亞馬遜 EKS 和圖形數據庫，由塞弗林·加桑爾·弗萊斯納和扎希·本·沙巴特。	2022 年 2 月 1 日
海王星叢集磁碟區現在可以成長為 128 兆位元組 (TiB) (p. 3)	除了中國以外的所有支援區域 GovCloud，海王星叢集磁碟區的大小限制現在已從 64 TiB 增加到 128 TiB。這適用於從 1.0.2.2 版開始的所有引擎版本。請參閱亞馬遜海王星儲存頁面。	2022 年 2 月 1 日
海王星全文檢索搜尋現在整合了所有版本的 OpenSearch. (p. 3)	請參閱使用亞馬遜OpenSearch服務在亞馬遜海王星進行全文搜索	2022 年 1 月 28 日

有關使用 Docker 容器部署圖形筆記本的部落格文章 (p. 3)	請參閱使用 Docker 容器部署圖形筆記本 AWS，由 Ganesh Swhney 和張強製作。	2022年1月22日
发动机版本 (p. 928)	截至 2022 年 1 月 13 日，引擎版本 1.0.5.1.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.1. R3。	2022 年 1 月 13 日
關於使用 Neptune ML 的圖形式建議系統的部落格文章 (p. 3)	請參閱採用 Neptune ML 的基於圖形的推薦系統：關於社交網絡鏈接預測挑戰的圖例，由崔延偉和 Wir Badr 撰寫。	2022 年 1 月 12 日
有關自動調度海王星的博客文 (p. 3)	請參閱 Navtanay Sinha 和蘇丹舒古普塔自動擴展您的 Amazon Neptune 資料庫以滿足工作負載需求。	2021 年 11 月 29 日
關於互動式圖表資料分析和視覺化的部落格 (p. 3)	請參閱使用亞馬遜海王星、亞馬遜雅典娜聯邦查詢和亞馬遜建立互動式圖表資料分析和視覺化 QuickSight，由桑迪普·韋爾迪和阿布舍克密斯拉製作。	2021 年 11 月 24 日
關於使用圖形式深度學習偵測身分詐騙的部落格文章 (p. 3)	了解 Careem 如何使用基於圖形的深度學習和亞馬遜海王星來檢測身份欺詐，凱文·奧布萊恩，卡姆蘭·哈比卜和威爾·巴德爾。	2021 年 11 月 23 日
引擎版本 (p. 915)	截至二零二一九年一月十一日，引擎版本 1.1.0.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.1. 0.0。	2021 年 11 月 19 日
有關集中化資料保護和合規性的部落格文章 (p. 3)	請參閱使用 Brian O'Keefe 撰寫的 AWS備份在 Amazon Neptune 中集中資料保護和合規。	2021 年 11 月 8 日
關於打擊欺詐和不當付款的博客文章 (p. 3)	請參閱弗拉迪·羅伊斯曼和斯賓塞·史密斯（Spencer Smith）的聯邦支出規模實時打擊欺詐和不當付款。	2021 年 11 月 2 日
有關防止虛假帳戶註冊的博客文章 (p. 3)	請參閱 Anjan Bisis 撰寫的使用亞馬遜詐騙偵測器使用 AI 即時防止虛假帳戶註冊。	2021 年 10 月 29 日
发动机版本 (p. 929)	截至二零二一年十月二十六日，引擎版本 1.0.5.1.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.1. R2。	2021 年 10 月 26 日

有關使用深圖庫 HawkEye 360 預測船舶風險的博客文章 (p. 3)	見 HawkEye360 預測船舶風險使用深圖庫和亞馬遜海王星蒂姆·帕夫利克, 伊恩·阿維萊茲, 丹·福特, 和高拉夫·雷勒.	2021 年 10 月 15 日
引擎版本 (p. 924)	截至二零二一年十月一日，引擎版本 1.0.5.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0. 5.1。	2021 年 10 月 1 日
關於為什麼開發人員喜歡博客 TinkerPop (p. 3)	請參閱為什麼開發人員喜歡 ApacheTinkerPop，圖形計算的開源框架，由布拉德·貝比，凱爾文勞倫斯和斯蒂芬·馬萊特。	2021 年 9 月 27 日
发动机版本 (p. 935)	截至二零二一九年九月十五日，引擎版本 1.0.5.0.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.0. R3。	2021 年 9 月 15 日
關於使用 Amundsen 和 Neptune 建立資料探索解決方案的部落格文章 (p. 3)	請參閱使用彼得·漢森斯和唐辛普森撰寫的阿蒙森和亞馬遜海王星構建數據發現解決方案。	2021 年 9 月 8 日
Neptune 更新了流輸詢器以支持非字符串全文搜索查詢 (p. 3)	此版本中包含許多全文檢索搜尋的改良功能，包括對非字串的屬性值建立索引的支援。請參閱亞馬遜海王星中的非字串OpenSearch索引。	2021 年 8 月 23 日
发动机版本 (p. 937)	截至二零二一年八月十六日，引擎版本 1.0.5.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.5.0. R2。	2021 年 8 月 16 日
发动机版本 (p. 938)	截至二零二一年八月十六日，引擎版本 1.0.4.2.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.2. R5。	2021 年 8 月 16 日
有關 Neptune 中圖形存儲協議支持的博客文章 (p. 3)	請參閱克里斯·史密斯為亞馬遜海王星介紹圖形存儲協議支持。	2021 年 8 月 2 日
有關海王星 ML 新功能的博客文章 (p. 3)	請參閱利用亞馬遜海王星 ML 提供的全新功能，探索圖表中的更多見解。	2021 年 7 月 30 日
有關使用 Neptune ML 更快獲得預測的博客文章 (p. 3)	請參閱使用亞馬遜海王星 ML 更快地獲取有關不斷發展的圖形數據的預測，由 Soji Adeshina 撰寫。	2021 年 7 月 30 日

有關使用 Neptune ML 更輕鬆快速地繪製機器學習的部落格文 (p. 3)	查看使用亞馬遜海王星 ML 更輕鬆，更快速的圖形機器學習。	2021 年 7 月 30 日
關於海王星開放密碼支持的博客文章 (p. 3)	請參閱為亞馬遜海王星宣布 OpenCypher：建立更好的圖形應用程序與 OpenCypher 和小鬼在一起，由布拉德·貝比。	2021 年 7 月 29 日
引擎版本 (p. 931)	截至二零二一年七月二十七日，引擎版本 1.0.5.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0. 5.0。	2021 年 7 月 27 日
第四版引擎 (p. 939)	截至二零二一年七月二十三日，引擎版本 1.0.4.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.2. R4。	2021年7月23 日
海王星在中國（北京）發射 (p. 3)	亞馬遜海王星現已在中國（北京）（cn-north-1）上市。	2021 年 7 月 21 日
发动机版本 (p. 939)	截至二零二一年六月二十八日，引擎版本 1.0.4.2.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.2. R3。	2021 年 6 月 28 日
有關 Dream11 如何使用海王星擴展其社交網絡的博客文章 (p. 3)	請參閱瞭解全球最大的夢幻運動平台 Dream11 如何透過亞馬遜海王星和亞馬遜擴展其社交網路。ElastiCache	2021 年 6 月 25 日
關於使用海王星和PoolParty語義套件將數據轉化為知識的博客文章 (p. 3)	請參閱使用PoolParty語義套件和亞馬遜海王星將數據轉換為知識，由約安娜 Lytra 和阿爾賓·艾哈梅蒂。	2021 年 6 月 16 日
關於使用 Neptune 探索UniProt知識庫的部落格文章 (p. 3)	請參閱透過AWS開放資料和亞馬遜海王星探索UniProt蛋白質知識庫，由 Eric Greene、Rafa Xu 和袁施撰寫。	2021 年 6 月 10 日
有關使用 Neptune 進行數據驅動風險分析的博 (p. 3)	請參閱亞馬遜海王星和亞馬遜 OpenSearch服務進行數據驅動的風險分析，由阿德里安·德容格和羅希特·薩蒂亞納拉亞納。	2021 年 6 月 10 日
发动机版本 (p. 940)	截至二零二一年六月一日起，引擎版本 1.0.4.2.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.2. R2。	2021 年 6 月 1 日

關於使用 Neptune 將AWS基礎結構視覺化的部落格 (p. 3)	請參閱使用亞馬遜海王星和AWS配置將您的AWS基礎設施視覺化，由羅漢·雷扎達和 Amey Dhavle 撰寫。	2021 年 5 月 25 日
關於將資料透鏡與 Neptune 搭配使用的組態部落格 (p. 3)	請參閱羅素沃特森使用資料鏡頭在 Amazon 海王星中設定AWS服務以建立知識圖表。	2021 年 5 月 5 日
關於使用資料透鏡在海王星建立知識圖表的部落格文章 (p. 3)	請參閱羅素沃特森使用資料鏡頭在亞馬遜海王星建立知識圖表。	2021 年 5 月 5 日
引擎版本 1.0.1.0、1.0.1.1 和 1.0.1.2 版本現在已淘汰 (p. 3)	從現在開始，將不會使用任何這些引擎版本或與之相關的任何修補程式來建立新的資料庫執行個體。	2021 年 4 月 26 日
關於日本經濟產業省使用海王星的案例研究的英文翻譯 (p. 3)	請參閱日本經濟產業省提供 GBizInfo 企業資訊搜尋資料庫。AWS	2021 年 3 月 31 日
關於使用海王星與亞馬遜理解和 Lex 的博客文章 (p. 3)	請參閱使用亞馬遜海王星，亞馬遜理解和亞馬遜萊克斯增壓您的知識圖，由戴夫·貝奇伯格。	2021 年 3 月 31 日
有關將 Lambda 函數與海王星搭配使用的 (p. 3)	請參閱 Ian Robinson 撰寫的搭配亞馬遜海王星使用 AWS Lambda	2021年3月26日
发动机版本 (p. 945)	截至二零二一年三月二十二日，引擎版本 1.0.4.1.R1.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.1. R1.1。	2021 年 3 月 22 日
发动机版本 (p. 948)	截至二零二一年三月十一日，引擎版本 1.0.4.1.R2.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.1. R2.1。	2021 年 3 月 11 日
有關使用 Neptune 的開源圖形筆記本進行圖形可視化的博客文章 (p. 3)	請參閱 Joy Wang，Ora Lassila 和斯蒂芬·馬萊特的圖形可視化的開源圖形筆記本入門。	2021 年 3 月 10 日
有關將 Neptune 與 Amundsen 資料探索和中繼資料引擎整合的教學 (p. 3)	看看如何使用阿蒙森與亞馬遜海王星，由安德魯·西姆布朗.	2021 年 3 月 2 日
发动机版本 (p. 946)	截至二零二一年二月二十一日，引擎版本 1.0.4.1.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.1. R2。	2021 年 2 月 24 日

发动机版本 (p. 952)	截至二零二一年二月二十一日，引擎版本 1.0.4.0.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.0. R2。	2021 年 2 月 24 日
发动机版本 (p. 956)	截至二零二一九年二月二十一日，引擎版本 1.0.3.0.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.3.0. R3。	2021 年 2 月 19 日
发动机版本 (p. 962)	截至二零二一九年二月二十九日，引擎版本 1.0.2.2.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.2. R6。	2021 年 2 月 19 日
關於使用亞馬遜事件建立知識圖的部落格文章 (p. 3)	請參閱使用亞馬遜 Comprehend 事件在亞馬遜海王星建立知識圖, 由布萊恩·奧基夫, 格雷厄姆霍伍德, 和納瓦奈·辛哈.	2021年1月19日
有關啟用低代碼圖形數據應用程序的博客文 (p. 3)	請參閱利奧·梅耶羅維奇、戴夫·貝赫貝格和泰勒·里根的使用亞馬遜海王星和石墨技術啟用低代碼圖形資料應用程式。	2021年1月18日
已新增筆記本文件，以開始使用圖形資料。 (p. 25)	已新增與 Neptune 工作台整合的區段，可協助您開始建立圖形資料和開發圖形應用程式，而不必啟動 Neptune 叢集，直到您準備好為止。	2021 年 1 月 15 日
關於在幾秒鐘內重置海王星圖形數據的博客文章 (p. 3)	請參閱在幾秒鐘內重置您的圖形資料在亞馬遜海王星, 由 Niraj Jetly 和納瓦奈辛哈.	2020 年 12 月 17 日
關於諾華股份公司如何使用 BERT SageMaker 和海王星的部落格文章 (p. 3)	請參閱諾華股份公司使用亞馬遜 SageMaker和亞馬遜海王星使用 BERT 構建和豐富知識圖，由奧特馬尼·哈姆扎烏伊，法特瑪阿爾干納齊和維克多·馬列塞維克。	2020 年 12 月 14 日
關於使用主題網路建立知識圖表的部落格文章 (p. 3)	請參閱利用亞馬遜海王星的主題網路建立知識圖表，由 AI 專案負責人愛德華·布朗、建築師愛德華多·皮亞羅、建築師、首席資料科學家馬西亞·奧利維拉和深入見解執行長傑克·漢普森撰寫的。	2020 年 12 月 14 日

引擎版本 (p. 943)	截至 2020 年 12 月 08 日，引擎版本 1.0.4.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.4.1。	2020 年 12 月 8 日
關於開始使用海王星 ML 的部落格文章 (p. 3)	請參閱如何開始使用海王星 ML，喬治·卡里皮斯，戴夫·貝奇貝格和卡爾蒂克·巴拉希。	2020 年 12 月 8 日
海王星現在擁有快速重置 API (p. 3)	使用快速重置 API，您可以快速輕鬆地刪除數據庫集群中的所有數據。請參閱快速重設 API。	2020 年 12 月 4 日
關於在鐘擺建立生物知識圖的博客文章 (p. 3)	請參閱康納·斯肯納頓使用亞馬遜海王星在鐘擺上構建生物知識圖。	二零二零年十一月二
關於 TinkerPop 3.4.8 海王星新功能的博客文章 (p. 3)	見探索阿帕奇 TinkerPop 3.4.8 在亞馬遜海王星的新功能，由斯蒂芬·馬萊特。	2020 年 11 月 18 日
有關在海王星中使用亞馬遜 Kendra 搜索服務的博客文章 (p. 3)	請參閱將您的企業知識圖結合到亞馬遜肯德拉，雅茲丹·希爾瓦尼，莫哈特·梅赫塔和切克拉瓦蒂。	2020 年 11 月 17 日
事件通知現在可用 (p. 3)	Neptune 現在支援事件通知，您可以使用這些通知更輕鬆地監視資料庫叢集。請參閱由使用 Neptune 事件通知。	2020 年 10 月 29 日
海關端點現已推出 (p. 3)	Neptune 現在支援自訂端點，以提高連線至資料庫執行個體的控請參閱連接到亞馬遜海王星端點。	2020 年 10 月 29 日
有關使用AWS資料庫移轉服務 (DMS) 填入 Neptune 圖形的部落格文章 (p. 3)	請參閱使用資料庫遷移服務 (DMS) 從關聯式AWS資料庫在 Amazon Neptune 中填入圖形 — 第 4 部分：將它們放在一起 (Chris Smith) 撰寫。	2020 年 10 月 22 日
有關使用AWS資料庫移轉服務 (DMS) 填入 Neptune 圖形的部落格文章 (p. 3)	請參閱使用資料庫遷移服務 (DMS) 從關聯式資料AWS庫在 Amazon Neptune 中填入圖形 — 第 3 部分：設計 RDF 模型 (Chris Smith)。	2020 年 10 月 22 日
有關使用AWS資料庫移轉服務 (DMS) 填入 Neptune 圖形的部落格文章 (p. 3)	請參閱使用資料庫遷移服務 (DMS) 從關聯式資AWS料庫在 Amazon Neptune 中填入圖形 — 第 2 部分：設計屬性圖模型 (Chris Smith)。	2020 年 10 月 22 日
有關使用AWS資料庫移轉服務 (DMS) 填入 Neptune 圖形的部落格文章 (p. 3)	請參閱使用資料庫移轉服務 (DMS) 從關聯式資料AWS庫在 Amazon Neptune 中填入圖形 — 第 1 部分：設定階段 (Chris Smith) 撰寫。	2020 年 10 月 22 日

引擎版本 (p. 950)	截至 2020 年 10 月 12 日，引擎版本 1.0.4.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0. 4.0。	2020年10月12日
发动机版本 (p. 958)	截至 2020 年 10 月 12 日，引擎版本 1.0.3.0.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.3.0. R2。	2020年10月12日
发动机版本 (p. 964)	截至 2020 年 10 月 12 日，引擎版本 1.0.2.2.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.2. R5。	2020年10月12日
關於針對 SPARQL 聯合查詢設定您的 VPC 的部落格文章 (p. 3)	請參閱為查爾斯·伊維的亞馬遜海王星設定 SPARQL 1.1 聯合查詢的亞馬遜 VPC。	2020年10月12日
關於撰寫 SPARQL 串聯刪除的部落格文章 (p. 3)	請參閱在 SPARQL 中編寫級聯刪除，由奧拉拉·拉西拉。	2020 年 10 月 5 日
關於使用 Neptune 繪製AWS資源的博客文章 (p. 3)	請參閱圖表您的AWS資源與亞馬遜海王星, 由戴夫·貝奇貝格.	2020 年 9 月 28 日
關於使用 Neptune 為藥物警戒和不良事件報告建立 Meddra 術語映射的部落格文章 (p. 3)	請參閱建立基於亞馬遜海王星的 MedRA 術語映射，用於藥物警戒和不良事件報告，由 Vaijayanti Joshi，德文阿特諾爾，博士學位和薩德漢舒馬霍特拉。	2020 年 9 月 24 日
有關使用 Neptune 從數據倉庫構建知識圖表的博客文章，以補充商業智能 (p. 3)	請參閱透過使用亞馬遜海王星從資料倉儲建立知識圖表來補充商業智慧 (Shahria Hossain) 和 Mikael Graindorge 撰寫的。	2020 年 9 月 23 日
關於使用海王星 Gemlin 用戶端進行負載平衡的部落格文章 (p. 3)	請參閱 Ian Robinson 撰寫的使用亞馬遜海王星 Grimlin 用戶端的負載平衡圖查詢。	2020 年 9 月 16 日
關於 Cox 汽車公司使用身份圖進行數字個性化的博客文章 (p. 3)	請參閱考克斯汽車使用由亞馬遜海王星提供支持的身份圖來擴展數字個性化，卡洛斯·倫登和尼拉傑特利。	2020 年 9 月 16 日
有關 Yelp 數據協作過濾的博客文章 (p. 3)	請參閱在 Yelp 資料上使用協作篩選，在亞馬遜海王星中建立推薦系統，由查德·廷德爾撰寫。	二 ○ 二 ○ 年九月八日
有關亞馬遜海王星中查詢結果可視化的博客文章 (p. 3)	請參閱凱爾文勞倫斯使用 Amazon 海王星工作台將查詢結果視覺化。	2020 年 9 月 2 日

海王星發布圖可視化 (p. 3)	Amazon Neptune 現在在 Neptune 工作台的 Jupyter 筆記本中提供廣泛的圖形視覺化功能，以及許多讓筆記本更容易使用的新功能。請參閱圖形視覺化。	2020 年 8 月 12 日
海王星在南美洲（聖保羅）發射 (p. 3)	亞馬遜海王星現已在南美洲（聖保羅）（`sa-east-1`）上市。	2020 年 8 月 6 日
海王星在亞太區（香港）成立 (p. 3)	亞馬遜海王星現已在亞太區域（香港）（`ap-east-1`）上市。	2020 年 8 月 6 日
引擎版本 (p. 954)	截至 2020 年 8 月 3 日，引擎版本 1.0.3.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.3.0。	2020 年 8 月 3 日
第四版引擎 (p. 965)	截至 2020 年 7 月 23 日，引擎版本 1.0.2.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.2. R4。	2020 年 7 月 23 日
有關 ZeroBase 使用 Amazon Neptune 自動化聯絡人追蹤的部落格文章 (p. 3)	請參閱 ZeroBase 使用亞馬遜海王星創建私有，安全和自動化的聯繫人跟踪，由大衛·哈里斯和亞倫·薩桑托。	2020年7月13日
海王星在美國西部 (加利福尼亞北部) 啟動 (p. 3)	亞馬遜海王星現已在美國西部 (加利佛尼亞北部) (`us-west-1`) 推出。	2020 年 7 月 9 日
亞馬遜海王星支援標籤式存取控制 (p. 3)	您現在可以在 IAM 政策中使用 AWS標籤來控制對 Neptune 資料庫的存取。請參閱 Amazon 海王星中的基於標籤的存取控制。	2020 年 7 月 7 日
一個 Java 流輸詢器現在可用 (p. 3)	亞馬遜海王星現在支持用於海王星流以及 Python 流輸詢器的 Java 版本。請參閱新增您正在建立的 Neptune 串流用戶堆疊的詳細資料。	2020 年 7 月 6 日
關於 AWS COVID-19 知識圖表的部落格文章 (p. 3)	請參閱建立和查詢 AWS COVID-19 知識圖，由尼納德庫爾卡尼，科爾比智者，喬治·普萊斯和米格爾·羅梅羅。	2020 年 7 月 1 日
引擎版本 (p. 984)	截至 2020 年 6 月 26 日，引擎版本 1.0.1.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.1.1。	2020 年 6 月 26 日

關於從火焰圖遷移到亞馬遜海王星的博客文章 (p. 3)	請參閱 Dave Bechberger 撰寫的 Moving to the cloud: Migrating Blazegraph to Amazon Neptune。	2020 年 6 月 25 日
關於將數據捕獲從 Neo4j 更改為亞馬遜海王星的博客文章 (p. 3)	請參閱 Sanjeet Sahay 撰寫的 Change data capture from Neo4j to Amazon Neptune using Amazon Managed Streaming for Apache Kafka。	2020 年 6 月 22 日
關於 Waves 如何使用亞馬遜海王星的博客文章 (p. 3)	請參閱 Pavel Vasilyev 撰寫的 How Waves runs user queries and recommendations at scale with Amazon Neptune。	2020 年 6 月 16 日
引擎版本 1.0.1.2 (p. 983)	截至 2020 年 6 月 10 日，引擎版本 1.0.1.2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.1.2。	2020 年 6 月 10 日
關於建立客戶知識庫的部落格文章 (p. 3)	請參閱 Ram Bhandarkar 撰寫的 Building a customer 360 knowledge repository with Amazon Neptune and Amazon Redshift。	2020 年 6 月 9 日
關於古諾西如何使用亞馬遜海王星的博客文章 (p. 3)	請參閱 Yosuke Uchiyama 撰寫的 How Gunosy built a comment feature in News Pass using Amazon Neptune。	二○二○年六月八日
關於 AWS COVID-19 知識圖表的部落格文章 (p. 3)	請參閱由尼納德庫爾卡尼，科爾比智者，喬治普萊斯和米格爾·羅梅羅構建和查詢 AWS COVID-19 知識圖。	2020 年 6 月 2 日
博客文章關於使用亞馬遜海王星探索 COVID-19 研究 (p. 3)	請參閱 George Price、Colby Wise、Miguel Romero 和 Ninad Kulkarni 製作的 Exploring scientific research on COVID-19 with Amazon Neptune、Amazon Comprehend Medical 和 Tom Sawyer Graph Database Browser。	2020 年 6 月 2 日
您現在可以使用將數據加載到海王星 AWS DMS (p. 3)	請參閱使用資料AWS庫遷移服務將資料從不同的資料存放區載入 Amazon Neptune。	2020 年 6 月 1 日
引擎版本 1.0.2.0 已被棄用 (p. 978)	亞馬遜海王星引擎版本 1.0.2.0 現已棄用。在此引擎版本上執行的叢集會在 2020 年 6 月 1 日之後的第一個維護時段自動升級至 1.0.2.1 版。	2020 年 5 月 19 日

關於使用 Neptune 構建客戶身份圖的博客文章 (p. 3)	請參閱 Rajesh Wunnava 和 Taylor Riggan 所著的 Building a customer identity graph with Amazon Neptune。	2020 年 5 月 12 日
发动机版本 (p. 980)	截至 2020 年 5 月 5 日，引擎版本 1.0.2.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.0. R3。	2020 年 5 月 5 日
第 6 版引擎 (p. 971)	截至 2020 年 4 月 22 日，引擎版本 1.0.2.1.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.1. R6。	2020 年 4 月 22 日
關於將數據從 Neo4j 遷移到海王星的博客文章 (p. 3)	請參閱 Sanjeet Sahay 的將 Neo4j 圖形資料庫遷移到有全自動公用程式的 Amazon Neptune。	二零二零年四月13日
關於降低使用 Neptune 建置圖形應用程式成本的部落格文章 (p. 3)	請參閱 Karthik Bharathy and Brad Bebee 的使用 Amazon Neptune T3 執行個體，將建置圖形應用程式的成本降低 76%。	二零二零年四月九日
海王星提供了一個 T3 突發實例類 (p. 3)	您現在可以建立 Amazon Neptune T3 高載執行個體，以實現符合成本效益的開發和測試目的。請參閱 Neptune T3 高載執行個體類別。	2020 年 4 月 8 日
发动机版本 (p. 967)	截至 2020 年 4 月 2 日起，引擎版本 1.0.2.2.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.2. R2。	2020 年 4 月 2 日
關於繪製 EDGAR 投資依賴性的博客文章 (p. 3)	請參閱 Lawrence Verdi 的用 Amazon Neptune 繪製投資相依性。	2020 年 3 月 17 日
海王星在歐洲（巴黎）發射 (p. 3)	亞馬遜海王星現已在歐洲（巴黎）（歐盟-西部 -3）上市。	2020 年 3 月 11 日
引擎版本 (p. 960)	截至 2020 年 3 月 09 日，1.0.2.2 版引擎正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。如需有關此引擎版本的詳細資訊，請參閱海王星引擎版本 1.0.2.2。	2020 年 3 月 9 日

停止並重新啟動資料庫複製程式 (p. 3)	您現在可以使用 Neptune 主控台停止資料庫叢集 7 天，稍後在再次需要時重新啟動它。資料庫叢集停用時，只需支付叢集儲存、手動快照和自動備份儲存的費用，但不需支付任何資料庫執行個體時數的費用。請參閱停止和啟動 Amazon Neptune 資料庫叢集。	2020 年 2 月 19 日
關於 Nike 社交圖表的視頻 (p. 3)	聽聽托德·埃斯卡洛納（Todd Escalona）與耐克高級工程經理馬克·旺恩海姆（Marc Wangenheim）的AWS談話，了解該公司如何通過在亞馬遜海王星上構建的社交圖為許多應用程序提供支持 請參閱 Nike：使用 Amazon Neptune 的大型社交圖。	2020 年 2 月 11 日
海王星叢集現在可以設定為需要 SSL 連線 (p. 3)	在仍支援 HTTP 連線的區域中，現在所有新參數群組都預設為開啟 SSL。現有的參數群組沒有變更，但您可以將 `neptune_enforce_ssl` 參數變更為 1，強制用戶端使用 SSL。請參閱傳輸中加密：使用 SSL/HTTPS 連接到 Neptune，以取得如何在仍支援 HTTP 連線的區域中為叢集啟用 HTTP 連線的相關資訊。如需叢集和執行個體參數的描述，請參閱您可以用來設定 Amazon Neptune 的參數。	2020 年 2 月 10 日
您現在可以在 Neptune 的 CloudFormation範本中指定引擎版本和刪除保護 (p. 3)	Amazon Neptune 已更新 CloudFormation範本，以包含可讓您為新資料庫叢集指定特定引擎版本的`AWS::Neptune::DBCluster.DeletionProtection`參數，以及可讓您為其啟用刪除保護的參數。`AWS::Neptune::DBCluster.EngineVersion`	二 ○ 二 ○ 年二月九日
刪除保護 (p. 3)	Amazon Neptune 已為資料庫叢集和執行個體提供刪除保護。一旦在資料庫叢集或執行個體上啟用刪除保護，就無法刪除。請參閱如果已啟用刪除保護，您就無法刪除資料庫執行個體。	2020 年 1 月 20 日
海王星在中國（寧夏）發射 (p. 3)	亞馬遜海王星現已在中國（寧夏）（CN-西北 -1）上市。	2020 年 1 月 15 日
第四版引擎 (p. 973)	適用於引擎版本 1.0.2.1 的修補程式 R4 已正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.1.R4。	2019 年 12 月 20 日

发动机版本 (p. 975)	適用於引擎 1.0.2.1 版的修補程式 R3 已正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.1.R3。	2019 年 12 月 12 日
有關使用 Neptune 分析社交媒體提要的博客文章 (p. 3)	請參閱使用 Amazon Neptune 分析社交媒體摘要。	2019 年 11 月 27 日
发动机版本 (p. 976)	適用於引擎版本 1.0.2.1 的修補程式 R2 已正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.1.R2。	2019 年 11 月 25 日
发动机版本 (p. 969)	亞馬遜海王星引擎 1.0.2.1.R1 版本正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.1。	2019 年 11 月 22 日
发动机版本 (p. 981)	引擎版本 1.0.2.0 的修補程式 R2 已正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.0.R2。	2019 年 11 月 21 日
關於海王星會議和研討會的博客文章：發明 2019 (p. 3)	在 AWS re: Invent 2019 上查看您的亞馬遜海王星會議、研討會和粉筆演講指南。	2019 年 11 月 20 日
发动机版本 (p. 978)	亞馬遜海王星引擎 1.0.2.0.R1 版本正式推出。如需詳細資訊，請參閱海王星引擎版本 1.0.2.0。	2019 年 11 月 8 日
有關使用海王星流捕獲圖形更改的博客文章 (p. 3)	請參閱使用 Neptune 串流擷取圖形變更。	2019 年 11 月 6 日
引擎版本 (p. 984)	亞馬遜海王星引擎版本 1.0.0.0.200502.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200502.0。	2019 年 10 月 31 日
海王星在中東（巴林）發射 (p. 3)	亞馬遜海王星現已在中東（巴林）（我南 -1）上市。	2019 年 10 月 30 日
海王星在加拿大（中部）發射 (p. 3)	亞馬遜海王星現已在加拿大（中部）（CA-中部 -1）上市。	2019 年 10 月 30 日
關於海王星新的 SPARQL 串流功能和 SPARQL 聯合查詢支援的部落格文章 (p. 3)	請參閱 Amazon Neptune 發布圖表的串流、SPARQL 聯合查詢等。	2019 年 10 月 17 日
引擎版本 (p. 985)	亞馬遜海王星引擎版本 1.0.0.0.200463.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200463.0。	2019 年 10 月 15 日
引擎版本 (p. 985)	亞馬遜海王星引擎版本 1.0.1.0.200457.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200457.0。	2019 年 9 月 19 日
有關海王星新 SPARQL 說明功能的部落格文章 (p. 3)	請參閱使用 SPARQL Explain 以了解 Amazon Neptune 中的查詢執行。	2019 年 9 月 17 日

有關海王星支持 TinkerPop 3.4 的博客文章 (p. 3)	請參閱亞馬遜海王星現在支持 TinkerPop 3.4 功能。	2019 年 9 月 6 日
關於在亞馬遜PyTorch上使用海王星的博客文章 SageMaker (p. 3)	查看在亞馬遜SageMaker和亞馬遜海王星PyTorch上使用的個性化 shop-by-style「」體驗。	2019 年 8 月 22 日
關於使用海王星AWSAppSync和亞馬遜彈性疼的博客文章 (p. 3)	請參閱整合替代資料來源 AWSAppSync：亞馬遜海王星和亞馬遜ElastiCache。	2019 年 8 月 22 日
海王星在 AWSGovCloud（美國東部）發射 (p. 3)	亞馬遜海王星現已在 AWSGovCloud（美國東部）（us-gov-east-1）上市。	二零一九年八月二十
海王星在 AWSGovCloud（美國西部）發射 (p. 3)	亞馬遜海王星現已在 AWSGovCloud（美國西部）（us-gov-west-1）上市。	2019 年 8 月 14 日
发动机版本 (p. 986)	亞馬遜海王星引擎版本 1.0.1.0.200369.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200369.0 版本。	2019 年 8 月 13 日
引擎版本 (p. 987)	亞馬遜海王星引擎版本 1.0.0.0.200366.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.0.366.0。	2019 年 7 月 26 日
關於在亞馬遜PyTorch上使用海王星的博客文章 SageMaker (p. 3)	查看在亞馬遜SageMaker和亞馬遜海王星PyTorch上使用的個性化 shop-by-style「」體驗。	2019 年 7 月 3 日
引擎版本 (p. 988)	亞馬遜海王星引擎版本 1.0.0.200348.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200348.0。	2019 年 7 月 2 日
海王星在歐洲發射（斯德哥爾摩）(p. 3)	亞馬遜海王星現已在歐洲（斯德哥爾摩）（歐洲北部 -1）上市。	2019 年 6 月 27 日
Neptune 現在可以將審核日誌發佈到CloudWatch日誌 (p. 3)	如需詳細資訊，請參閱將海王星日誌發佈到 Amazon CloudWatch 日誌。	2019 年 6 月 18 日
引擎版本 (p. 988)	亞馬遜海王星引擎版本 1.0.1.0.200310.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200310.0。	2019 年 6 月 12 日
關於LifeOmic的博客文章 JupiterOne (p. 3)	請參閱如何LifeOmic使用 Amazon Neptune JupiterOne 簡化安全性和合規操作。	2019 年 5 月 2 日
海王星在亞太地區（首爾）啟動 (p. 3)	亞馬遜海王星現已在亞太區域（首爾）（AP-東北 -2）上市。	2019 年 5 月 1 日

引擎版本 (p. 989)	亞馬遜海王星引擎版本 1.0.0.0.200296.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.0.296.0。	2019 年 5 月 1 日
海王星在亞太區域（孟買）啟動 (p. 3)	亞馬遜海王星現已在亞太區域（孟買）（4 月南 1 號）上市。	2019 年 3 月 6 日
關於小精靈查詢提示的部落格文章 (p. 3)	請參閱適用於 Amazon Neptune 的 Gremlin 查詢提示簡介。	2019 年 2 月 26 日
海王星在亞太地區（東京）啟動 (p. 3)	亞馬遜海王星現已在亞太區域（東京）（AP-東北 -1）上市。	2019 年 1 月 23 日
AWS CloudFormation用於創建訪問海王星的AWS Lambda函數的模板 (p. 54)	已更新入門區段，並新增AWS CloudFormation範本以建立 Lambda 函數以搭配 Neptune 使用。如需詳細資訊，請參閱 Neptune 入門。	2019 年 1 月 23 日
引擎版本 (p. 990)	亞馬遜海王星引擎版本 1.0.0.200267.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200267.0。	2019 年 1 月 21 日
海王星在亞太地區（悉尼）啟動 (p. 3)	亞馬遜海王星現已在亞太地區（悉尼）（AP-東南 -2）上市。	2019 年 1 月 9 日
關於使用隱喻的部落格文章 (p. 3)	請參閱使用 Metaphactory 在 Amazon Neptune 上探索知識圖形。	2019 年 1 月 9 日
海王星在亞太區（新加坡）推出 (p. 3)	亞馬遜海王星現已在亞太地區（新加坡）（AP-東南 -1）上市。	2018 年 12 月 13 日
引擎版本 (p. 990)	亞馬遜海王星引擎版本 1.0.1.0.200264.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.264.0。	2018 年 11 月 19 日
亞馬遜海王星 SSL 支持 (p. 154)	海王星現在支援 SSL 連線。	2018 年 11 月 19 日
合併錯誤主題 (p. 849)	所有錯誤訊息和代碼資訊現在都放在單一主題中。	2018 年 11 月 15 日
更新的開始使用主題 (p. 52)	以其他連結和重新整理過的文件更新入門主題。	2018 年 11 月 14 日
引擎版本 (p. 991)	亞馬遜海王星引擎版本 1.0.0.200258.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200258.0。	2018 年 11 月 8 日
海王星在歐洲推出（法蘭克福）(p. 3)	亞馬遜海王星現已在歐洲（法蘭克福）（歐盟中部 -1）上市。	2018 年 11 月 7 日
博客文章 #1 在一系列 (p. 3)	請參閱讓我幫您畫圖 - 第一部分 - Air Routes。	2018 年 11 月 7 日

有關使用亞馬遜 SageMaker Jupyter 筆記本的博客文章 (p. 3)	請參閱使用亞馬遜 SageMaker Jupyter 筆記本分析亞馬遜海王星圖。	2018 年 11 月 1 日
引擎版本 (p. 992)	亞馬遜海王星引擎版本 1.0.1.0.200255.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200255.0。	2018 年 10 月 29 日
海王星在歐洲（倫敦）發射 (p. 3)	亞馬遜海王星現已在歐洲（倫敦）（歐盟-西部 -2）上市。	2018 年 10 月 3 日
引擎版本 (p. 992)	亞馬遜海王星引擎版本 1.0.1.0.200237.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200237.0。	2018 年 9 月 6 日
引擎版本 (p. 993)	亞馬遜海王星引擎版本 1.0.1.0.200236.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200236.0。	2018 年 7 月 24 日
引擎版本 (p. 994)	亞馬遜海王星引擎版本 1.0.0.200233.0 正式推出。如需詳細資訊，請參閱更新 1.0.0.200233.0。	2018年6月22日
新海王星快速入門 (p. 54)	更新 AWS CloudFormation 快速入門和 Gremlin 主控台教學。如需詳細資訊，請參閱 Amazon 海王星快速入門使用 AWS CloudFormation。	2018 年 6 月 19 日
亞馬遜海王星初始版 (p. 3)	此為《Neptune 使用者指南》的初始版本。另請參閱發行部落格文章 Amazon Neptune 已正式上市。	2018 年 5 月 30 日
介紹性海王星博客文章 (p. 3)	請參閱 Amazon Neptune - 全受管圖形資料庫服務。	2017 年 11 月 29 日

Amazon Neptune 入門

Amazon Neptune 是一種全受管的圖形資料庫服務，可擴展以處理數十億個關係，並讓您以毫秒的延遲進行查詢，而這類容量卻以低廉的成本進行查詢。

如果您想要有關 Neptune 的更多詳細資訊，請參閱亞馬遜海王星功能概述 (p. 82)。

如果您已經知道圖表，請跳到使用圖形筆記本 (p. 30)。或者，如果您想要立即建立 Neptune 資料庫，請參閱使用AWS CloudFormation堆疊建立海王星資料庫叢集 (p. 59)。

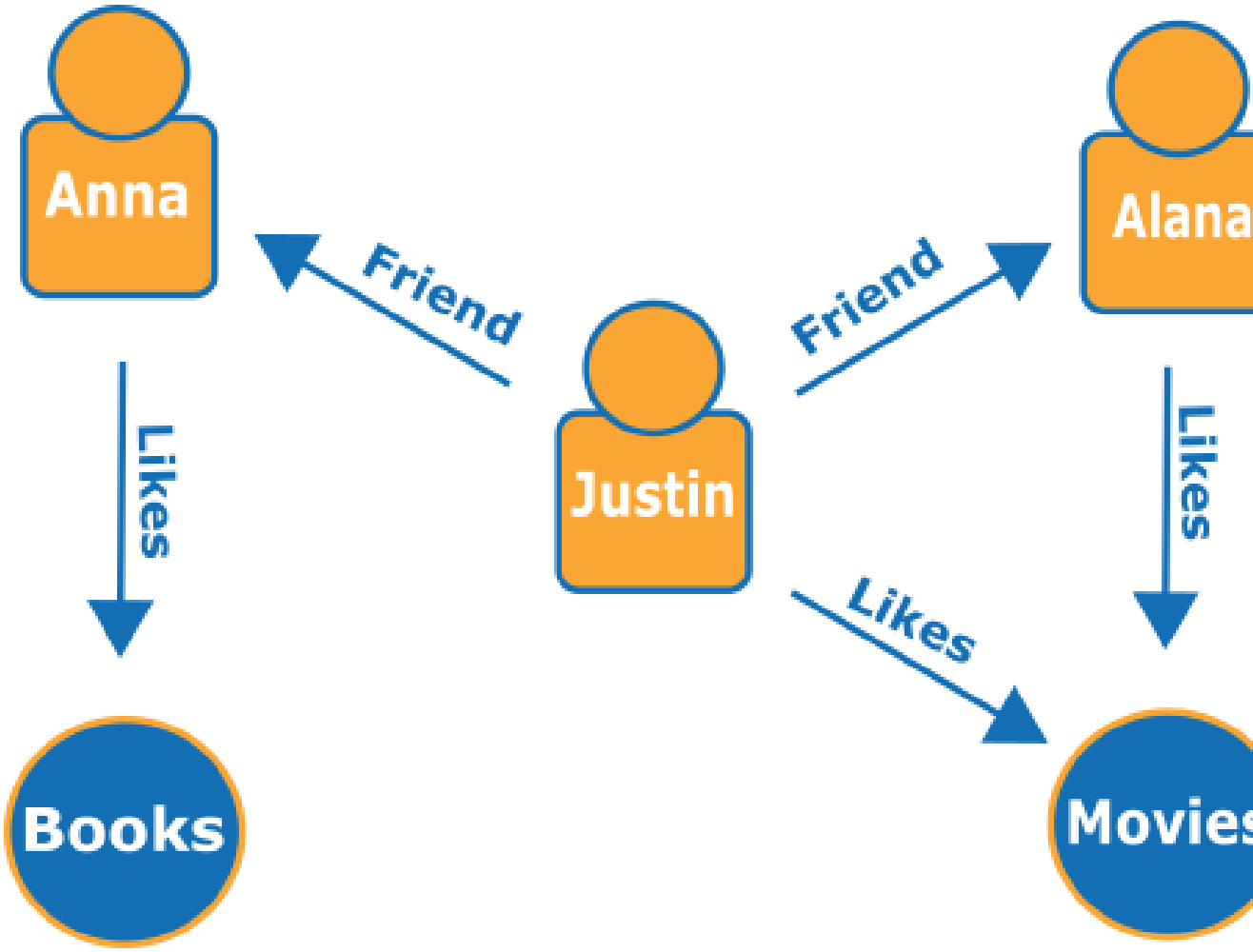
否則，在開始之前，您可能想要了解更多有關圖形數據庫的信息。

圖形數據庫到底是什麼？

圖形資料庫經過最佳化，可儲存和查詢資料項目之間的關係。

它們存儲數據項本身作為圖形的頂點，並且它們之間的關係作為邊緣。每個邊都有一個類型，並且從一個頂點（開始）導向到另一個（終點）。關係可以被稱為謂詞以及邊緣，和頂點有時也被稱為節點。在所謂的屬性圖，頂點和邊緣都可以具有與它們相關聯的附加屬性了。

這是一個代表社交網絡中的朋友和愛好的小圖：



邊緣顯示為命名箭頭，頂點表示它們連接的特定人物和愛好。

僅需簡單地周遊此圖形，即可了解 Justin 朋友的喜好。

為什麼要使用圖形數據庫？

每當實體之間的連線或關係是您嘗試建模的資料核心時，圖形資料庫就是您自然的選擇。

首先，很容易將資料互連建模為圖形，然後撰寫複雜的查詢，從圖形中擷取真實世界的資訊。

使用關聯式資料庫建置對等的應用程式需要您建立多個外部索引鍵的資料表，然後撰寫巢狀 SQL 查詢和複雜的聯結。從編碼的角度來看，這種方法不僅迅速變得笨拙，而且隨著數據量的增加，其性能會迅速下降。

相比之下，像 Neptune 這樣的圖形資料庫可以查詢數十億個頂點之間的關係，而不會陷入困境。

你可以用圖形數據庫做什麼？

圖形可以通過許多方式表示真實世界實體的相互關係，例如行動，所有權，父母身份，購買選擇，個人關係，家庭關係等。

以下是使用圖形資料庫的一些最常見的區域：

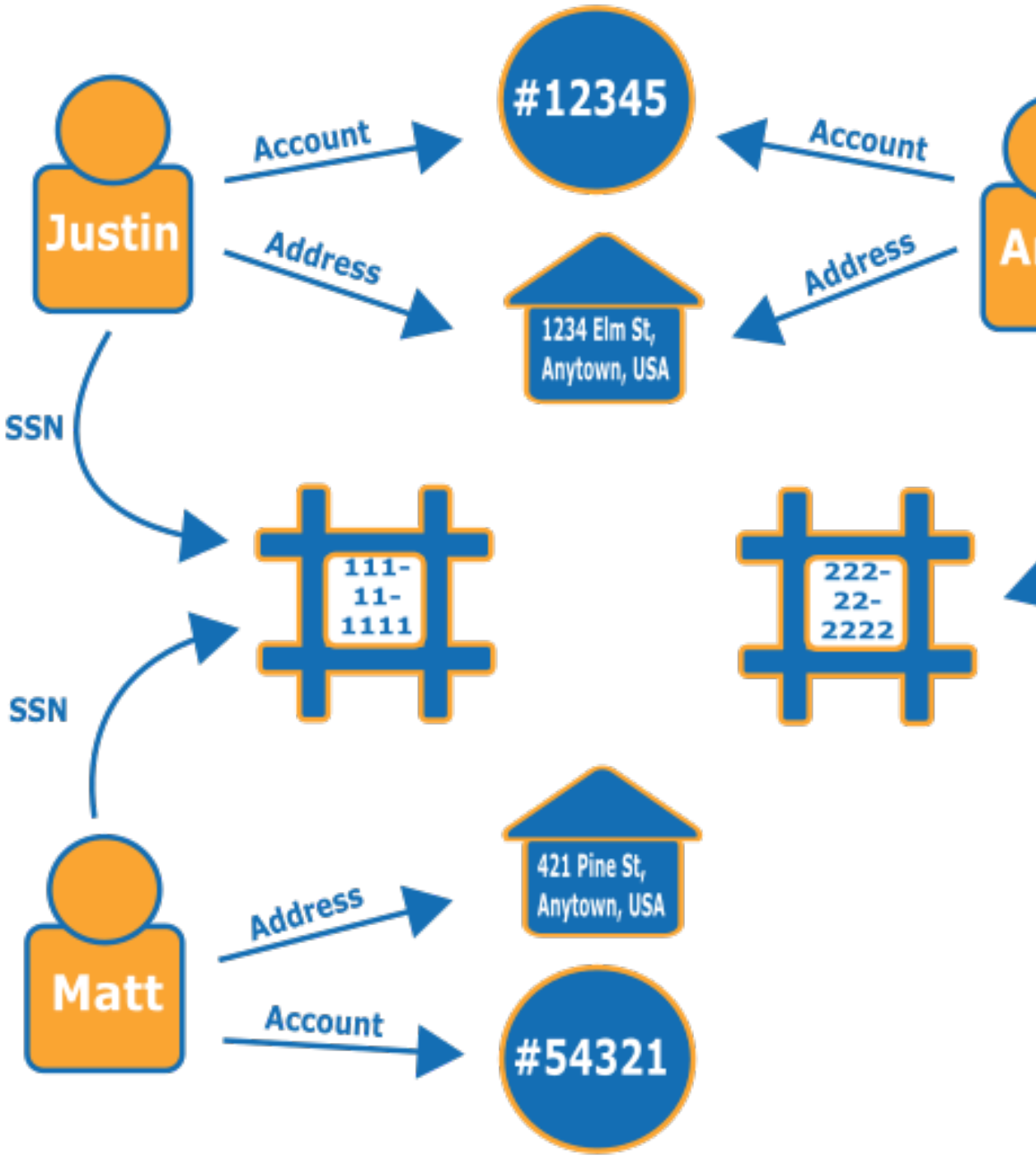
- 知識圖 — 知識圖可讓您組織和查詢各種連接信息以回答一般問題。使用知識圖，您可以在產品目錄中添加主題信息，並為維基數據中包含的各種信息建模。

  若要深入瞭解知識圖表的運作方式以及使用位置，請參閱上的知識圖AWS。
- 身分圖 — 在圖形資料庫中，您可以儲存客戶興趣、朋友和購買記錄等資訊類別之間的關係，然後查詢該資料以提供個人化且相關的建議。

  例如，您可以使用圖形資料庫，根據其他遵循相同運動並具有類似購買歷史記錄的其他人購買哪些產品，向使用者提供產品推薦。或者，您可以識別有共同朋友但還不認識對方的人，並提出友誼推薦。

  這種圖形被稱為身份圖，被廣泛用於個性化與用戶的互動。若要深入瞭解，請參閱上的識別圖AWS。若要開始建立自己的身分圖，您可以從使用 Amazon Neptune 的身分圖範例開始。
- 欺詐圖-這是圖形數據庫的常見用途。他們可以幫助您跟踪信用卡購買和購買地點，以檢測不尋常的使用情況，或檢測購買者正在嘗試使用與已知欺詐案件中相同的電子郵件地址和信用卡。它們可以讓您檢查與個人電子郵件地址相關聯的多個人，或是在不同實體位置共用相同 IP 位址的多個人。

  請看下列圖表。它顯示了三個人的關係和他們的身份相關信息。每位使用者都有一個地址、銀行帳戶與身分證號碼。但是，我們可以看到 Matt 和 Justin 共享相同的社會安全號碼，這是不規則的，表明其中一個可能發生的欺詐行為。對欺詐圖表的查詢可以揭示此類連接，以便可以對其進行審查。



要了解更多有關欺詐圖表及其使用位置的更多信息，請參閱上的欺詐圖表AWS。

- 社交網絡 — 在社交網絡應用程序中使用圖形數據庫的第一個也是最常見的領域之一。

例如，假設您想要將一個社交動態消息。您可以輕鬆地使用後端的圖形數據庫向用戶提供結果，這些結果反映了來自家人，朋友，他們「喜歡」更新的人以及住在他們附近的人的最新更新。

- 行駛路線 — 根據當前交通和典型的交通模式，圖表可以幫助您找到從起點到目的地的最佳路線。
- 物流 — 圖形可以幫助識別最有效的方式來使用可用的運輸和配送資源來滿足客戶的需求。
- 診斷 — 圖表可以代表複雜的診斷樹狀結構，可查詢這些樹狀結構以識別觀察到的問題和故障的來源。
- 科學研究 — 使用圖形資料庫，您可以建置應用程式，使用靜態加密來儲存和瀏覽科學資料，甚至是敏感醫療資訊。例如，您可以儲存疾病和基因相互作用的模型。您可以搜尋蛋白質化學反應的圖形模式，尋找與疾病相關聯的其他基因。您可以將化學化合物建模為圖表，並查詢分子結構中的模式。您可以關聯不同系統中醫療記錄中的患者資料。您可以依主題整理研究期刊，快速尋找相關資訊。
- 法規規則 — 您可以將複雜的法規要求儲存為圖形，並加以查詢，以偵測這些要求可能適用於您的 day-to-day 業務營運的情況。
- 網路拓撲與事件 — 圖形資料庫可協助您管理和保護 IT 網路。當您將網路拓撲儲存為圖形時，您也可以在網路上儲存和處理許多不同類型的事件。您可以回答問題，例如有多少主機正在執行指定的應用程式。您可以查詢可能顯示特定主機已遭惡意程式入侵的病毒碼，並查詢連線資料，這些資料可協助追蹤程式至下載該程式的原始主機。

你如何查詢一個圖表？

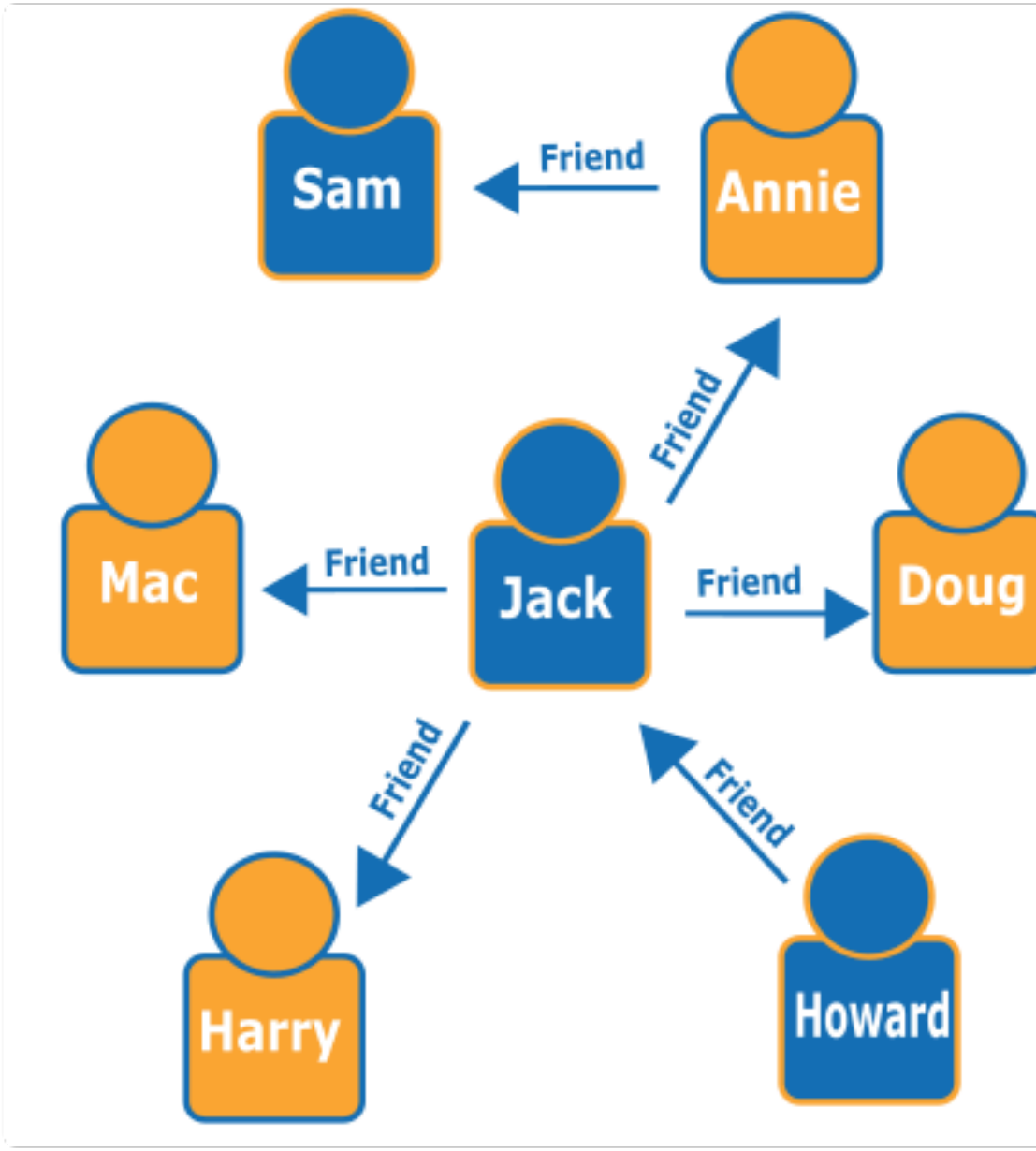
Neptune 支援三種特殊用途的查詢語言，專為查詢不同種類的圖形資料而設計。您可以使用下列語言來新增、修改、刪除和查詢 Neptune 圖形資料庫中的資料：

- Grimlin (p. 300) 是屬性圖的圖形遍歷語言。在 Grimlin 查詢是由離散步驟組成的遍歷，每個步驟都遵循一個邊緣到一個節點。如需詳細資訊，請參閱 Apache TinkerPop 3 的小鬼文件。

  Gremlin 的 Neptune 實現與其他實現有一些不同，特別是當您使用 Grimlin-Groovy（以序列化文本發送的 Gremlin 查詢）時。如需詳細資訊，請參閱亞馬遜海王星的小精靈標準合規 (p. 83)。
- 开放密码 (p. 382)-OpenCypher 是屬性圖的聲明式查詢語言，最初由 Neo4j 開發，然後在 2015 年開源，並在 Apache 2 開源許可證下為 OpenCypher 項目做出了貢獻。有關語言規範，請參閱 Cypher 查詢語言參考（版本 9），以及 Cypher 樣式指南以獲取更多信息。
- SPARQL (p. 422) 是 RDF 資料的宣告式查詢語言，根據全球資訊網協會 (W3C) 標準化，並在 SPARQL 1.1 概觀) 和 SPARQL 1.1 查詢語言規格中所述的圖形模式比對。亞馬遜海王星的 SPARQL 標準合規性 (p. 92)如需 SPARQL 的 Neptune 實作的特定詳細資訊，請參閱。

匹配小靈和 SPARQL 查詢的例子

鑑於以下圖表的人（節點）和他們的關係（邊緣），你可以找出誰的「朋友的朋友」一個特定的人 is— 例如，霍華德的朋友的朋友。



您可以從圖中發現 Howard 有一位朋友 Jack，而 Jack 則有四位朋友：Annie、Harry、Doug 和 Mac。此處採用的簡單範例是搭配簡易圖形使用，但您可以依複雜度、資料集大小與結果大小來擴大這些類型的查詢規模。

這是一個小鬼遍歷查詢，它返回霍華德朋友的朋友的姓名：

```
g.V().has('name', 'Howard').out('friend').out('friend').values('name')
```

這是一個 SPARQL 查詢，它返回霍華德朋友的朋友的姓名：

```
prefix : <#>

  select ?names where {
    ?howard :name "Howard" .
    ?howard :friend/:friend/:name ?names .
  }
```

Note

任何資源描述架構 (RDF) 三元組的每個部分都具備與其相關的 URI。在上面的範例中，URI 前綴是故意使用短。

參加使用 Amazon Neptune 的在線課程

如果您喜歡使用視頻學習，請在「在線技術講座」中AWS提供在AWS線課程，以幫助您開始學習。介紹圖形數據庫的一個是：

使用 Amazon Neptune 圖形資料庫簡介、深入探索和示範。

深入研究圖形參考架構

當您考慮圖形資料庫可以為您解決哪些問題，以及如何處理這些問題時，最好的起點之一就是 Neptune 圖形參考架構 GitHub 專案。

您可以在這裡找到圖形工作負載類型的詳細說明，以及三個區段，協助您設計有效的圖形資料庫：

- 資料模型和查詢語言 — 本節將引導您瞭解 Gremlin 和 SPARQL 之間的差異，以及如何在它們之間進行選擇。
- 圖形資料建模 — 這是如何做出圖形資料建模決策的全面討論，包括使用 Gremlin 進行屬性圖建模的詳細逐步解說，以及使用 SPARQL 進行 RDF 建模。
- 將其他數據模型轉換為圖形模型 — 在這裡您可以了解如何去關係數據模型轉換成圖形模型。

還有三個部分可引導您完成使用 Neptune 的特定步驟：

- 從 Neptune VPC 外部的用戶端連線至 Amazon Neptune — 本節顯示從資料庫叢集所在的 VPC 外部連線到 Neptune 的幾個選項。
- 從AWS Lambda 函數存取 Amazon Neptune — 在這裡您將了解如何從 Lambda 函數可靠地連接到 Neptune。
- 從 Amazon Kinesis 資料串流寫入 Amazon 海王星 — 本節可協助您使用 Neptune 處理高寫入輸送量案例。

使用 Neptune 圖形筆記本快速開始使用

您不必使用 Neptune 圖形筆記本來處理 Neptune 圖形，因此，如果您願意，可以繼續使用AWS CloudFormation範 (p. 59)本立即建立新的 Neptune 資料庫。

同時，無論您是圖形新手，想要學習和實驗，還是有經驗且想要精簡查詢，Neptune Workbench (p. 31) 都提供了一個互動式開發環境 (IDE)，可在您建置圖形應用程式時提高生產力。

海王星在開源海王星圖形筆記本項目和JupyterLab海王星工作台中提供 Jupyter 和筆記本。GitHub這些筆記本在互動式編碼環境中提供範例應用程式教學課程和程式碼片段，您可以在此瞭解圖形技術和 Neptune。您可以使用它們來逐步使用不同的查詢語言、不同的資料集，甚至是後端的不同資料庫來逐步完成設定、設定、填入和查詢圖形。

您可以使用數種不同的方式託管這些筆記本：

- Neptune 工作台 (p. 31)可讓您在完全受管的環境中執行 Jupyter 筆記本 (託管於 Amazon)SageMaker，並自動為您載入最新版本的 Neptune 圖形筆記本專案。建立新的 Neptune 資料庫時，很容易在 Neptune 主控台中設定工作台。
- 您也可以安裝 Jupyter 或JupyterLab (p. 34)本地。這可讓您從筆記型電腦執行筆記本，連線至 Neptune 或其中一個開放原始碼圖形資料庫的本機執行個體。在後一種情況下，您可以在花一分錢之前盡可能多地嘗試圖形技術。然後，當您準備就緒時，就可以順利移至 Neptune 提供的受管理生產環境。

使用 Neptune 工作台託管海王星筆記本

Neptune 提供項目T3和T4g執行個體類型可讓您以每小時不到 0.10 USD 的價格開始使用。您需要透過 Amazon 支付工作台資源的費用SageMaker，與您的 Neptune 帳單分開。請參閱海王星定價頁面。在海王星工作台上創建的 Jupyter 和JupyterLab筆記本全部使用亞馬遜 Linux 2 和 JupyterLab 3 環境。如需JupyterLab筆記型電腦支援的詳細資訊，請參閱 Amazon SageMaker 文件。

您可以使用以下兩種方式之一，使用 Neptune 工作台建立 Jupyter 或JupyterLab筆記本：AWS Management Console

- 建立新的 Neptune 資料庫叢集時，請使用筆記本組態功能表。若要這麼做，請依照中所述的步驟執行使用啟動 Neptune 資料庫叢集AWS Management Console (p. 534)。
- 建立資料庫叢集之後，請使用左側導覽窗格中的 [筆記本] 功能表。若要執行此操作，請按以下步驟進行。

使用記事本選單建立 Jupyter 或JupyterLab記事本

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在左側的導覽窗格中，選擇 Notebooks (筆記本)。
3. 選擇 Create notebook (建立筆記本)。
4. 在 [叢集] 清單中，選擇您的 Neptune DB 叢集。如果還沒有資料庫叢集，請選擇 Create cluster (建立叢集) 以建立叢集。
5. 選取記事本執行個體類型。
6. 為您的筆記本提供名稱，並選擇性提供描述。
7. 除非您已為筆記本建立 AWS Identity and Access Management (IAM) 角色，否則請選擇 [建立 IAM 角色]，然後輸入 IAM 角色名稱。

   Note

   如果您確實選擇重複使用 IAM 角色；為先前的筆記本建立，則角色政策必須包含正確的權限，才能存取您正在使用的 Neptune DB 叢集。您可以通過檢查`neptune-db:*`動作下的資源 ARN 中的組件是否與該集群匹配來驗證這一點。當您嘗試執行筆記本魔術指令時，不正確設定的權限會導致連線錯誤。
8. 選擇 Create notebook (建立筆記本)。創建過程可能需要 5 到 10 分鐘才能準備就緒。
9. 建立記事本之後，請選取它，然後選擇 [開啟 Jupyter] 或 [開啟]。JupyterLab

主控台可以為您的筆記本建立 AWS Identity and Access Management (IAM) 角色，也可以自行建立一個角色。此角色的政策應包括下列各項：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::aws-neptune-notebook-(AWS region)",
        "arn:aws:s3:::aws-neptune-notebook-(AWS region)/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "neptune-db:*",
      "Resource": [
        "arn:aws:neptune-db:(AWS region):(AWS account ID):(Neptune resource ID)/*"
      ]
    }
  ]
}
```

請注意，上述原則中的第二個陳述式會列出一或多個 Neptune 叢集資源 ID (p. 209)。

此外，角色也應該建立下列信任關係：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

同樣，準備好一切可能需要 5 到 10 分鐘。

您可以將新筆記本配置為與 Neptune ML 一起使用，如中所述為 Neptune ML 手動設定 Neptune 筆記本 (p. 660)。

使用 Python 將通用SageMaker筆記本連接到海王星

如果您已經安裝了 Neptune 魔術，將筆記本連接到 Neptune 很容易，但是即使您沒有使用 Neptune SageMaker 筆記本，也可以使用 Python 將筆記本連接到 Neptune。

在SageMaker筆記本電腦中連接到海王星的步驟

1. 安裝小精靈 Python 客戶端：

   ```
   !pip install gremlinpython
   ```

   海王星筆記本為您安裝 Gemlin Python 客戶端，因此只有在您使用普通SageMaker筆記本時，才需要執行此步驟。
2. 撰寫如下的程式碼來連線並發出 Gemlin 查詢：

```
from gremlin_python import statics
from gremlin_python.structure.graph import Graph
from gremlin_python.process.graph_traversal import __
from gremlin_python.process.strategies import *
from gremlin_python.driver.driver_remote_connection import DriverRemoteConnection
from gremlin_python.driver.aiohttp.transport import AiohttpTransport
from gremlin_python.process.traversal import *
import os

port = 8182
server = '(your server endpoint)'

endpoint = f'wss://{server}:{port}/gremlin'

graph=Graph()

connection = DriverRemoteConnection(endpoint,'g',
                 transport_factory=lambda:AiohttpTransport(call_from_event_loop=True))

g = graph.traversal().withRemote(connection)

results = (g.V().hasLabel('airport')
                .sample(10)
                .order()
                .by('code')
                .local(__.values('code','city').fold())
                .toList())

# Print the results in a tabular form with a row index
for i,c in enumerate(results,1):
    print("%3d %4s %s" % (i,c[0],c[1]))

connection.close()
```

Note

如果你碰巧正在使用一個版本的小鬼 Python 客戶端，那麼這一行：

```
connection = DriverRemoteConnection(endpoint,'g',

 transport_factory=lambda:AiohttpTransport(call_from_event_loop=True))
```

只會是：

```
connection = DriverRemoteConnection(endpoint,'g')
```

啟用 CloudWatch Neptune 筆記型電腦上

CloudWatchNeptune 筆記本現在預設為啟用記錄。如果您的筆記型電腦沒有產生記CloudWatch錄檔，請依照下列步驟手動啟用：

1. 登入 AWS Management Console 並開啟 SageMaker主控台。
2. 在左側的導覽窗格中，選擇記事本，然後選擇記事本執行個體。尋找您要啟用記錄檔的 Neptune 筆記本名稱。
3. 選取該筆記本執行個體的名稱，以前往詳細資訊頁面。
4. 如果記事本執行個體正在執行，請選取筆記本詳細資料頁面右上方的停止按鈕。

5. 在「權限和加密」下，存在 IAM 角色 ARN 欄位。選取此欄位中的連結，即可前往此筆記本執行個體所搭配執行的 IAM 角色。
6. 建立下列策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:Describe*",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
      "Resource": "*"
    }
  ]
}
```

7. 儲存此新政策並將其附加至步驟 4 中的 IAM 角色。
8. 按一下SageMaker筆記本執行個體詳細資訊頁面右上角的 [開始]。
9. 當記錄開始流動時，您應該會在詳細資料頁面的 [筆記本執行個體設定] 區段左下方附近，在標示為 [生命週期組態] 欄位下方看到 [檢視記錄] 連結。

如果筆記型電腦無法啟動，SageMaker主控台的筆記型電腦詳細資料頁面中會出現訊息，指出筆記型電腦執行個體需要 5 分鐘以上才能啟動。CloudWatch您可以在此名稱下找到與此問題相關的記錄檔：

```
(your-notebook-name)/LifecycleConfigOnStart
```

在本機電腦上設定圖形筆記本

圖形筆記本項目具有在本地計算機上設置 Neptune 筆記本的說明：

- 先決條件
- 重複和安裝 JupyterLab
- 連接到圖形數據庫。

您可以將本機筆記本連線到 Neptune DB 叢集，或連線至開放原始碼圖形資料庫的本機或遠端執行個體。

將 Neptune 筆記本與海王星叢集

如果您要連接到後端的 Neptune 叢集，則可能需要在 Amazon 中執行筆記本電腦SageMaker。從連接到 Neptune SageMaker 可以比從本地安裝筆記本電腦更方便，它可以讓您更輕鬆地使用 Neptune ML (p. 652) 工作。

如需有關如何在中設定筆記本的指示SageMaker，請參閱使用 Amazon 啟動圖形筆記本。SageMaker

如需有關如何設定和設定 Neptune 本身的指示，請參閱設定海王星 (p. 52)。

您也可以將 Neptune 筆記本的本機安裝連線到 Neptune 資料庫叢集。因為 Amazon Neptune 資料庫叢集只能在 Amazon 虛擬私人雲端 (VPC) 中建立，而這種情況可能會稍微複雜一些，因為 Amazon Neptune 資料庫叢集的設計與外界隔離。有很多方法可以從外部連接到 VPC。一種是使用負載平衡器。另一種方法是使用 VPC 對等互連 (請參閱 Amazon 虛擬私有雲端對等互連指南)。

但是，對於大多數人來說，最方便的方法是連接以在 VPC 中設置 Amazon EC2 代理服務器，然後使用 SSH 隧道（也稱為端口前向）連接到它。您可以在圖形筆記本專additional-databases/neptune案的資料夾中將圖形筆記本本機連接到 Amazon Neptune 中找到有關如何設定的指示。 GitHub

使用 Neptune 筆記本搭配開放原始碼

若要免費開始使用圖形技術，您也可以在後端使用 Neptune 筆記型電腦搭配各種開放原始碼資料庫。例如：TinkerPopGirmlin 伺服器和火焰圖資料庫。

要使用 Girmlin 服務器作為後端數據庫，請按照以下說明進行操作：

- 將圖形筆記本連接到 Gemlin 服務器文件夾。GitHub
- 圖形筆記本電腦的小鬼配置文件夾。GitHub

若要使用 Blazegraph 的本機執行個體做為後端資料庫，請依照下列指示執行：

- 大火圖快速入門說明
- 圖形筆記本電腦的配置文件夾。GitHub

將您的海王星筆記本從 Jupyter 遷移到 3 JupyterLab

在 2022 年 12 月 21 日之前建立的海王星筆記本使用亞馬遜 Linux 1 環境。您可以透過執行以下AWS部落格文章中所述的步驟，將在該日期之前建立的舊版 Jupyter 筆記本移轉到新的 Amazon Linux 2 環境 (使JupyterLab用 Amazon Linux 2) 將您的工作移轉到 Amazon SageMaker 筆記型電腦執行個體。

此外，還有一些專門適用於將 Neptune 筆記本遷移到新環境的步驟：

海王星特定的先決條件

在來源 Neptune 筆記本的 IAM 角色中，新增下列所有權限：

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket",
    "s3:CreateBucket",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::(your ebs backup bucket name)",
    "arn:aws:s3:::(your ebs backup bucket name)/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "sagemaker:ListTags"
  ],
  "Resource": [
    "*"
  ]
```

```
}
```

請務必為將用於備份的 S3 儲存貯體指定正確的 ARN。

海王星特定生命週期組態

建立第二個生命週期設定指令碼來還原備份 (自on-create.sh) 時，如部落格文章所述，生命週期名稱必須遵循下列aws-neptune-*格式，例如aws-neptune-sync-from-s3。如此可確保在 Neptune 主控台中建立筆記型電腦期間可以選取 LCC。

從快照到新執行個體的海王星特定同步

在部落格文章中描述的從快照同步到新執行個體的步驟中，以下是 Neptune 特定的變更：

- 在步驟 4 上，選擇筆記型電腦-al2-v2。
- 在步驟 5 中，重複使用來源 Neptune 筆記本的 IAM 角色。
- 在步驟 7 和 8 之間：
  - 在記事本執行個體設定中，設定使用該aws-neptune-*格式的名稱。
  - 開啟 [網路設定] 折疊式選單，然後選取與來源筆記本中相同的 VPC、子網路和安全性群組。

新筆記本創建後的海王星特定步驟

1. 選取記事本的 [開啟 Jupyter] 按鈕。一旦文SYNC_COMPLETE件顯示在主目錄中，繼續進行下一步。
2. 前往主控台中的筆記本執行個體頁SageMaker面。
3. 停止筆記本。
4. 選擇 Edit (編輯)。
5. 在筆記本執行個體設定中，選取來源 Neptune 筆記本的原始生命週期，以編輯生命週期組態欄位。請注意，這不是 EBS 備份生命週期。
6. 選擇更新筆記本設定。
7. 重新啟動筆記本。

透過部落格文章中所述步驟的修改，您的圖形筆記本現在應該移轉到使用 Amazon Linux 2 和 JupyterLab 3 環境的新 Neptune 筆記本執行個體上。它們會在中的 Neptune 頁面上顯示以供存取和管理AWS Management Console，您現在可以選取 [開啟 Jupyter] 或 [開啟]，繼續從中斷的地方繼續工作。JupyterLab

在筆記本中使用 Neptune 工作台魔術

Neptune 工作台在筆記本電腦中提供了許多所謂的魔術命令，這些命令可以節省大量的時間和精力。它們分為兩類：線魔術和細胞魔術。

線條魔術是指令前面有一個單一的百分比符號 (%)。它們只接受行輸入，而不是從細胞體的其餘部分輸入。Neptune 工作台提供以下線路神奇：

- % 種子 (p. 38)
- % 載入 (p. 39)
- % 載入識別碼 (p. 39)
- % 載入狀態 (p. 39)
- % 取消載入 (p. 39)
- % 狀態 (p. 39)
- %gremlin_ 狀態 (p. 39)

- % 打開狀態，或 %oc_ 狀態 (p. 39)
- %stream_viewer (p. 40)
- %sparql_ 狀態 (p. 40)
- %圖形/筆記本設定 (p. 40)
- %圖形筆記本主機 (p. 40)
- %圖形筆記本版本/版本 (p. 40)
- % 圖形筆記本瀏覽選項 (p. 40)
- % 統計數字 (p. 41)
- % 摘要 (p. 41)

細胞魔術前面有兩個百分號（%%）而不是一個，並使用單元格內容作為輸入，儘管它們也可以將行內容作為輸入。Neptune 工作台提供以下細胞魔術：

- %%sparql (p. 42)
- %% 麻黃 (p. 42)
- %% 開封密碼，或 %%oc (p. 43)
- %%圖形筆記本設定/設定 (p. 41)
- %%圖形筆記本_vis_選項 (p. 43)

還有兩種魔術，一個線條魔術和一個細胞魔術，用於工作Neptune Leptune (p. 652)：

- % 海王星 _ 毫升 (p. 43)
- % 海王星 _ 毫升 (p. 46)

> **Note**
>
> 使用 Neptune 魔術時，通常可以使用`--help`或`-h`參數來取得說明文字。使用細胞魔術，身體不能為空，因此在獲得幫助時，將填充文本甚至單個字符放在身體中。例如：

```
%%gremlin --help
  x
```

可變注入細胞或線魔術

在筆記本中定義的變數，可以在記事本中的任何儲存格或線條魔法中使用格式參考：`${VAR_NAME}`。

例如，假設您定義了這些變數：

```
c = 'code'
my_edge_labels = '{"route":"dist"}'
```

然後，這個 Gemlin 在單元格魔術中查詢：

```
%%gremlin -de $my_edge_labels
g.V().has('${c}','SAF').out('route').values('${c}')
```

相當於這個：

```
%%gremlin -de {"route":"dist"}
g.V().has('code','SAF').out('route').values('code')
```

適用於所有查詢語言的查詢引數

下列查詢引數可在 Neptune 工作台中使%%gremlin用%%opencypher、和%%sparql魔法：

常用查詢引數

- **--store-to**(或 **-s**) — 指定要儲存查詢結果的變數名稱。
- **--silent**— 如果存在，查詢完成後不會顯示輸出。
- **--group-by**(或 **-g**) — 指定用來分組節點的內容 (例如code或T.region)。頂點會根據其指定的群組著色。
- **--ignore-groups**— 如果存在，則忽略所有分組選項。
- **--display-property**(或 **-d**) — 指定應為每個頂點顯示其值的性質。

  每種查詢語言的預設值如下：
  - 對於小精靈：T.label。
  - 對於開放密碼：~labels。
  - 對於 SPARQL：type。
- **--edge-display-property**(或 **-t**) — 指定應針對每個邊顯示其值的性質。

  每種查詢語言的預設值如下：
  - 對於小精靈：T.label。
  - 對於開放密碼：~labels。
  - 對於 SPARQL：type。
- **--tooltip-property**(或 **-de**) — 指定屬性，其值應顯示為每個節點的工具提示。

  每種查詢語言的預設值如下：
  - 對於小精靈：T.label。
  - 對於開放密碼：~labels。
  - 對於 SPARQL：type。
- **--edge-tooltip-property**(或 **-te**) — 指定性質，其值應顯示為每個邊的工具提示。

  每種查詢語言的預設值如下：
  - 對於小精靈：T.label。
  - 對於開放密碼：~labels。
  - 對於 SPARQL：type。
- **--label-max-length** (或 **-l**) — 指定任何頂點標示的最大字元長度。預設值為 10。
- **--edge-label-max-length** (或 **-le**) — 指定任何邊緣標籤的最大字元長度。預設值為 10。

  在只有開放密碼的情況下，這是--rel-label-max-length或-rel相反。
- **--simulation-duration**(或 **-sd**) — 指定視覺化物理模擬的最長持續時間。預設值為 1500 毫秒。
- **--stop-physics**(或 **-sp**) — 在初始模擬穩定後停用視覺化物理。

這些引數的屬性值可以由單一屬性索引鍵組成，或可以為每個標籤類型指定不同屬性的 JSON 字串組成。JSON 字串只能使用變數注入 (p. 37)來指定。

%seed線條魔術

%seed線條魔術是將資料新增至 Neptune 端點的便捷方式，您可以使用這些端點來探索和試驗 Gremlin、OpenCypher 或 SPARQL 查詢。它提供了一個表單，您可以在其中選擇要瀏覽的數據模型（屬性圖或 RDF），然後從 Neptune 提供的許多不同的樣本數據集中進行選擇。

%load線條魔術

%load線條魔術會生成一個表單，您可以使用該表單向 Neptune 提交批量加載請求（請參閱海王星裝載程式 (p. 275)）。該來源的區域，必須與 Neptune 叢集所在同一個區域中的 Amazon S3 路徑。

%load_ids線條魔術

%load_ids線路神奇會擷取已提交至筆記本主機端點的載入 ID (請參閱海王星裝載機獲取狀態請求參數 (p. 283))。該請求採用以下形式：

```
GET https://your-neptune-endpoint:port/loader
```

%load_status線條魔術

%load_statusline magic 會擷取已提交至筆記本主機端點的特定載入工作的載入狀態，這些工作由線路輸入指定 (請參閱海王星裝載機獲取狀態請求參數 (p. 283))。該請求採用以下形式：

```
GET https://your-neptune-endpoint:port/loader?loadId=loadId
```

線條魔術類似於如下：

```
%load_status load id
```

%cancel_load線條魔術

%cancel_load線條魔術取消特定的加載作業（請參閱海王星裝載機取消工 (p. 291)）。該請求採用以下形式：

```
DELETE https://your-neptune-endpoint:port/loader?loadId=loadId
```

線條魔術類似於如下：

```
%cancel_load load id
```

%status線條魔術

從筆記本的主機端點擷取狀態資訊 (p. 770) (%圖形/筆記本設定 (p. 40)顯示主機端點)。

%gremlin_status線條魔術

檢索小鬼查詢狀態信息 (p. 324)。

%opencypher_status線條魔術（也%oc_status）

檢索開放密碼查詢的查詢狀態。這條線魔術採用以下可選參數：

- **--queryId**或 **-q**— 指定要顯示其狀態之特定執行中查詢的 ID。
- **--cancel_query**或 **-c**— 取消執行中的查詢。不需要一個值。
- **--silent**or **-s**— 如果`--silent`在取消查詢`true`時設定為，則會以 HTTP 回應碼取消執行中的查詢`200`。否則，HTTP 響應代碼將是`500`。

- **--store-to**— 指定要儲存查詢結果的變數名稱。

%sparql_status線條魔術

擷取 SPARQL 查詢狀態資訊 (p. 449)。

%stream_viewer線條魔術

如果在 Neptune 叢集上啟用串流，%stream_viewer線條魔術會顯示一個介面，允許以互動方式探索 Neptune 串流中記錄的項目。它接受以下可選參數：

- **language**— 串流資料的查詢語言：gremlin或sparql。預設值 (如果您不提供此引數)gremlin。
- **--limit**— 指定每頁要顯示的串流項目數目上限。如果您不提供此引數，則預設值為10。

> **Note**
>
> 只有 1.0.5.1 及更早版本的引擎才能完全支援%stream_viewer線條魔術。

%graph_notebook_config線條魔術

這行魔術會顯示 JSON 物件，其中包含筆記本用來與 Neptune 通訊的組態。配置包括：

- host：要連線並發出指令的端點。
- port：向 Neptune 發出命令時使用的連接埠。預設值為 8182。
- auth_mode：向 Neptune 發出命令時使用的驗證模式。IAM如果連線到已啟用 IAM 身份驗證的叢集，否則必須是DEFAULT。
- load_from_s3_arn：指定要使用的%load魔法風雲會的 Amazon S3 ARN。如果此值為空，則必須在指%load令中指定 ARN。
- ssl：表示是否使用 TLS 連接到 Neptune 的布林值。預設值為 true。
- aws_region：部署此筆記型電腦的區域。此資訊用於 IAM 身份驗證和請%load求。

您可以通過將%graph_notebook_config輸出複製到新的單元格中來更改配置，然後在該單元格中對其進行更改。然後，如果您在新%%圖形筆記本設定/設定 (p. 41)單元格上運行單元格魔術，則配置將相應地更改。

%graph_notebook_host線條魔術

將線路輸入設定為筆記本的主機。

%graph_notebook_version線條魔術

%graph_notebook_version線路神奇返回 Neptune 工作台筆記本發布號。例如，在版本中引入了圖形可視化1.27。

%graph_notebook_vis_options線條魔術

%graph_notebook_vis_options線條魔術會顯示筆記本目前正在使用的視覺效果設定。這些選項會在 vis.js 文件中進行說明。

您可以通過將輸出複製到新單元格中，進行所需的更改，然後在單元格上運行% %graph_notebook_vis_options單元格魔術來修改這些設置。

若要將視覺效果設定還原為預設值，您可以使用reset參數執行%graph_notebook_vis_options線條魔術。這會重設所有視覺效果設定：

```
%graph_notebook_vis_options reset
```

%statistics線條魔術

%statistics線路神奇用於擷取或管理 DFE 引擎統計資料 (請參閱管理要使用的海王星 DFE 的統計資料 (p. 131))。這種魔術也可以用來檢索圖形摘要 (p. 138)。

其接受下列參數：

- **--language**— 統計資料端點的查詢語言：或propertygraph (或pg) 或rdf。

  如果未提供，則預設值為propertygraph。
- **--mode**(或 **-m**) — 指定要提交的請求或動作類型：status、、、disableAutoCompute、enableAutoCompute、refresh、deletedetailed、或basic) 之一。

  如果未提供，則預設值為status除--summary非指定，在此情況下，預設值為basic。
- **--summary**— 從所選語言的統計資料摘要端點擷取圖形摘要。
- **--silent**— 如果存在，查詢完成後不會顯示輸出。
- **--store-to**— 用來指定要儲存查詢結果的變數。

%summary線條魔術

%summary線魔術用於檢索圖形摘要 (p. 138)信息。可從 Neptune 引擎版本開始使用1.2.1.0。

其接受下列參數：

- **--language**— 統計資料端點的查詢語言：或propertygraph (或pg) 或rdf。

  如果未提供，則預設值為propertygraph。
- **--detailed**— 在輸出中打開或關閉結構欄位的顯示。

  如果未提供，則預設為basic摘要顯示模式。
- **--silent**— 如果存在，查詢完成後不會顯示輸出。
- **--store-to**— 用來指定要儲存查詢結果的變數。

細%%graph_notebook_config胞魔法

%%graph_notebook_config儲存格魔術會使用包含組態資訊的 JSON 物件來修改筆記本用來與 Neptune 通訊的設定 (如果可能的話)。配置採用與%圖形/筆記本設定 (p. 40)線魔術返回的相同形式。

例如：

```
%%graph_notebook_config
{
  "host": "my-new-cluster-endpoint.amazon.com",
  "port": 8182,
```

```
  "auth_mode": "DEFAULT",
  "load_from_s3_arn": "",
  "ssl": true,
  "aws_region": "us-east-1"
}
```

細%%sparql胞魔法

%%sparql儲存格魔術會向 Neptune 端點發出 SPARQL 查詢。它接受以下可選行輸入：

- **-h**或**--help** — 傳回有關這些參數的說明文字。
- **--path**— 為 SPARQL 端點的路徑加上前置詞。例如，如果您指定，--path "abc/def"則呼叫的端點將是*host*:*port*/abc/def。
- **--expand-all**— 這是一個查詢視覺效果提示，告訴視覺化檢視在圖形圖表中包含所有?s ?p ?o結果，而不論繫結類型為何。

  依預設，SPARQL 視覺效果僅包含三重模式，其中o?為uri或bnode (空白節點)。所有其他?o繫結類型 (例如常值字串或整數) 都會被視為可使用「圖形」索引標籤中的「詳細資料」窗格檢視的?s節點屬性。

  如果您想要在視覺效果中包含這類常值做為頂點，請使用--expand-all查詢提示。

  請勿將此視覺化提示與解釋參數結合使用，因為解釋查詢不會被視覺化。
- **--explain-type**— 用於指定要使用的說明模式 (下列其中一個:dynamicstatic、、或details)。
- **--explain-format**— 用來指定說明查詢的回應格式 (text/csv或之一text/html)。
- --store-to — 用於指定要儲存查詢結果的變數。

explain查詢範例：

```
%%sparql explain

SELECT * WHERE {?s ?p ?o} LIMIT 10
```

具有視覺效果提示參數的--expand-all視覺化查詢範例 (請參閱可視化 (p. 50))：

```
%%sparql --expand-all

SELECT * WHERE {?s ?p ?o} LIMIT 10
```

細%%gremlin胞魔法

細%%gremlin胞魔術發出一個小鬼查詢到 Neptune 端點使用 WebSocket. 它接受切換至小精靈 explain (p. 347) /> mode 或的選擇性行輸入小精靈 API profile (p. 357)，以及一個單獨的選用視覺效果提示輸入來修改視覺效果輸出行為 (請參閱怪靈可視化 (p. 49))。

explain查詢範例：

```
%%gremlin explain

g.V().limit(10)
```

profile查詢範例：

```
%%gremlin profile
```

```
g.V().limit(10)
```

具有視覺化查詢提示的視覺化查詢範例：

```
%%gremlin -p v,outv

g.V().out().limit(10)
```

%%gremlin profile查詢的可選參數

- **--chop**— 指定設定檔結果字串的最大長度。如果您不提供此引數，則預設值為 250。
- **--serializer**— 指定用於結果的序列化程式。允許的值是任何有效的 MIME 類型或 TinkerPop 驅動程序「序列化器」枚舉值。如果您不提供此引數，則預設值為application.json。
- **--no-results**— 僅顯示結果計數。如果未使用，則依預設，所有查詢結果都會顯示在設定檔報告中。
- **--indexOps**— 顯示所有索引作業的詳細報告。

細%%opencypher胞魔術（也%%oc）

細%%opencypher胞魔術（也有縮寫%%oc形式），發出 OpenCypher 查詢到 Neptune 端點。它接受以下可選行輸入參數：

- mode — 查詢模式：query或bolt。如果您不提供此引數，則預設值為query。
- **--group-by**或 **-g**— 指定用來分組節點的內容。例如：code, ~id。如果您不提供此引數，則預設值為~labels。
- **--ignore-groups**— 如果存在，則忽略所有分組選項。
- **--display-propery**或 **-d**— 指定應為每個頂點顯示其值的性質。如果您不提供此引數，則預設值為~labels。
- **--edge-display-propery**或 **-de**— 指定應針對每個邊顯示其值的性質。如果您不提供此引數，則預設值為~labels。
- **--label-max-length**或 **-l**— 指定要顯示的頂點標示的最大字元數。如果您不提供此引數，則預設值為10。
- **--store-to**或 **-s**— 指定要儲存查詢結果的變數名稱。

細%%graph_notebook_vis_options胞魔法

%%graph_notebook_vis_options儲存格魔術可讓您設定筆記本的視覺化選項。您可以將%graph-notebook-vis-options線條魔術返回的設置複製到新單元格中，對其進行更改，然後使用%%graph_notebook_vis_options單元格魔術來設置新值。

這些選項會在 vis.js 文件中進行說明。

若要將視覺效果設定還原為預設值，您可以使用reset參數執行%graph_notebook_vis_options線條魔術。這會重設所有視覺效果設定：

```
%graph_notebook_vis_options reset
```

%neptune_ml線條魔術

您可以使用%neptune_ml線路神奇來啟動和管理各種 Neptune ML 作業。

Note

您也可以使用% 海王星 _ 毫升 (p. 46)儲存格魔術來啟動和管理某些 Neptune ML 作業。

- **%neptune_ml export start**— 啟動新的匯出工作。

  參數

  - **--export-url**匯*###-(##) ######*的 Amazon API Gateway 端點。
  - **--export-iam**— (選用) 指出對匯出網址的請求必須使用 SIGv4 簽署的旗標。
  - **--export-no-ssl**— (選用) 指出連線至匯出程式時不應使用 SSL 的旗標。
  - **--wait**— (選用) 指出作業應等到匯出完成之前的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定匯出狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待匯出工作完成的時間 (以秒為單位) (預設值：3,600)。
  - **--store-to***location-to-store-result*— (選用) 用於儲存匯出結果的變數。如果`--wait`指定，最終狀態將存儲在那裡。
- **%neptune_ml export status**— 擷取匯出工作的狀態。

  參數

  - **--job-id***#### ID* — 要擷取其狀態之匯出工作的 ID。
  - **--export-url**匯*###-(##) ######*的 Amazon API Gateway 端點。
  - **--export-iam**— (選用) 指出對匯出網址的請求必須使用 SIGv4 簽署的旗標。
  - **--export-no-ssl**— (選用) 指出連線至匯出程式時不應使用 SSL 的旗標。
  - **--wait**— (選用) 指出作業應等到匯出完成之前的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定匯出狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待匯出工作完成的時間 (以秒為單位) (預設值：3,600)。
  - **--store-to***location-to-store-result*— (選用) 用於儲存匯出結果的變數。如果`--wait`指定，最終狀態將存儲在那裡。
- **%neptune_ml dataprocessing start**— 啟動 Neptune ML 資料處理步驟。

  參數

  - **--job-id***#### ID – (##)* 指派給此工作的 ID。
  - **--s3-input-uri***S3 URI* — (選用) 要在其中尋找此資料處理工作輸入的 S3 URI。
  - **--config-file-name***####* — (選擇性) 此資料處理工作的組態檔名稱。
  - **--store-to***location-to-store-result*— (選擇性) 用於儲存資料處理結果的變數。
  - **--instance-type***(######)* — (選擇性) 用於此資料處理工作的執行個體大小。
  - **--wait**— (選擇性) 指出作業應等到資料處理完成之前的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定資料處理狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待資料處理工作完成的時間 (以秒為單位) (預設值：3,600)。
- **%neptune_ml dataprocessing status**— 擷取資料處理工作的狀態。

  參數

  - **--job-id***## ID* — 要擷取其狀態之工作的 ID。
  - **--store-to***######* — (選用) 用來儲存模型訓練結果的變數。

- **--wait**— (選擇性) 指出作業應等到模型訓練完成為止的旗標。
- **--wait-interval***interval-to-wait*— (選擇性) 設定模型訓練狀態檢查之間的時間 (以秒為單位) (預設值：60)。
- **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待資料處理工作完成的時間 (以秒為單位) (預設值：3,600)。

- **%neptune_ml training start**— 啟動 Neptune ML 模型訓練程序。

  參數

  - **--job-id***#### ID – (##)* 指派給此工作的 ID。
  - **--data-processing-id***###### ID* — (選用) 建立要用於訓練之人工因素之資料處理工作的 ID。
  - **--s3-output-uri***S3 URI* — (選用) 用來存放此模型訓練工作輸出的 S3 URI。
  - **--instance-type***(######)* — (選擇性) 用於此模型訓練工作的執行個體大小。
  - **--store-to***location-to-store-result*— (選擇性) 用於儲存模型訓練結果的變數。
  - **--wait**— (選擇性) 指出作業應等到模型訓練完成為止的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定模型訓練狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待模型訓練工作完成的時間 (以秒為單位) (預設值：3,600)。
- **%neptune_ml training status**— 擷取 Neptune ML 模型訓練工作的狀態。

  參數

  - **--job-id***## ID* — 要擷取其狀態之工作的 ID。
  - **--store-to***######* — (選用) 用來儲存狀態結果的變數。
  - **--wait**— (選擇性) 指出作業應等到模型訓練完成為止的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定模型訓練狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待資料處理工作完成的時間 (以秒為單位) (預設值：3,600)。
- **%neptune_ml endpoint create**— 建立 Neptune ML 模型的查詢端點。

  參數

  - **--job-id***#### ID – (##)* 指派給此工作的 ID。
  - **--model-job-id***###### ID* — (選用) 要為其建立查詢端點之模型訓練工作的 ID。
  - **--instance-type***(######)* — (選用) 用於查詢端點的執行個體大小。
  - **--store-to***location-to-store-result*— (選用) 要儲存端點建立結果的變數。
  - **--wait**— (選用) 指出作業應等到端點建立完成的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定狀態檢查之間的時間 (以秒為單位) (預設值：60)。
  - **--wait-timeout***###* — (選擇性) 設定在傳回最新狀態之前等待端點建立工作完成的時間 (以秒為單位) (預設值：3,600)。
- **%neptune_ml endpoint status**— 擷取 Neptune ML 查詢端點的狀態。

  參數

  - **--job-id***#### ID* — (選用) 要報告其狀態之端點建立工作的 ID。
  - **--store-to***location-to-store-result*— (選擇性) 用來儲存狀態結果的變數。
  - **--wait**— (選用) 指出作業應等到端點建立完成的旗標。
  - **--wait-interval***interval-to-wait*— (選擇性) 設定狀態檢查之間的時間 (以秒為單位) (預設值：60)。

- **`--wait-timeout`***###* — (選擇性) 設定在傳回最新狀態之前等待端點建立工作完成的時間 (以秒為單位) (預設值：3,600)。

細%%neptune_ml胞魔法

%%neptune_ml儲存格魔術會忽略線條輸入，例如--job-id或--export-url。相反，它可以讓您在儲存格體內提供這些輸入和其他輸入。

您還可以將此類輸入保存在另一個單元格中，分配給 Jupyter 變量，然後使用該變量將它們注入到單元格體中。這樣，您可以一遍又一遍地使用此類輸入，而無需每次都重新輸入它們。

這只有在注入變量是單元格的唯一內容時才有效。您不能在一個儲存格中使用多個變數，也不能在文字和變數的組合中使用。

例如，%%neptune_ml export start儲存格魔術可以使用包含中所述之所有參數的儲存格主體中的 JSON 文件用於控制 Neptune 匯出程序的參數 (p. 500)。

在 N eptune-ML-01 簡介至節點分類-Gremlin 筆記本的「匯出資料和模型組態」區段的「配置功能」下，您可以看到下列儲存格如何在指派給 Jupyter 變數的文件中保存匯出參數export-params：

```
export_params = {
  "command": "export-pg",
  "params": {
    "endpoint": neptune_ml.get_host(),
    "profile": "neptune_ml",
    "useIamAuth": neptune_ml.get_iam(),
    "cloneCluster": False
   },
  "outputS3Path": f'{s3_bucket_uri}/neptune-export',
  "additionalParams": {
    "neptune_ml": {
      "targets": [
        {
          "node": "movie",
          "property": "genre"
        }
      ],
      "features": [
        {
          "node": "movie",
          "property": "title",
          "type": "word2vec"
        },
        {
          "node": "user",
          "property": "age",
          "type": "bucket_numerical",
          "range" : [1, 100],
          "num_buckets": 10
        }
      ]
    }
  },
  "jobSize": "medium"}
```

當您執行此儲存格時，Jupyter 會以該名稱儲存參數文件。然後，您可以使${export_params}用將 JSON 文檔注入到 a 的主體中%%neptune_ml export start cell，如下所示：

```
%%neptune_ml export start --export-url {neptune_ml.get_export_service_host()} --export-iam
 --wait --store-to export_results
```

```
${export_params}
```

可用形式的%%neptune_ml細胞魔術

細%%neptune_ml胞魔術可以用於以下形式：

- **%%neptune_ml export start**— 啟動 Neptune ML 匯出程序。
- **%%neptune_ml dataprocessing start**— 啟動 Neptune ML 資料處理工作。
- **%%neptune_ml training start**— 啟動 Neptune ML 模型訓練工作。
- **%%neptune_ml endpoint create**— 為模型建立 Neptune ML 查詢端點。

Neptune 工作台中的圖形可視化

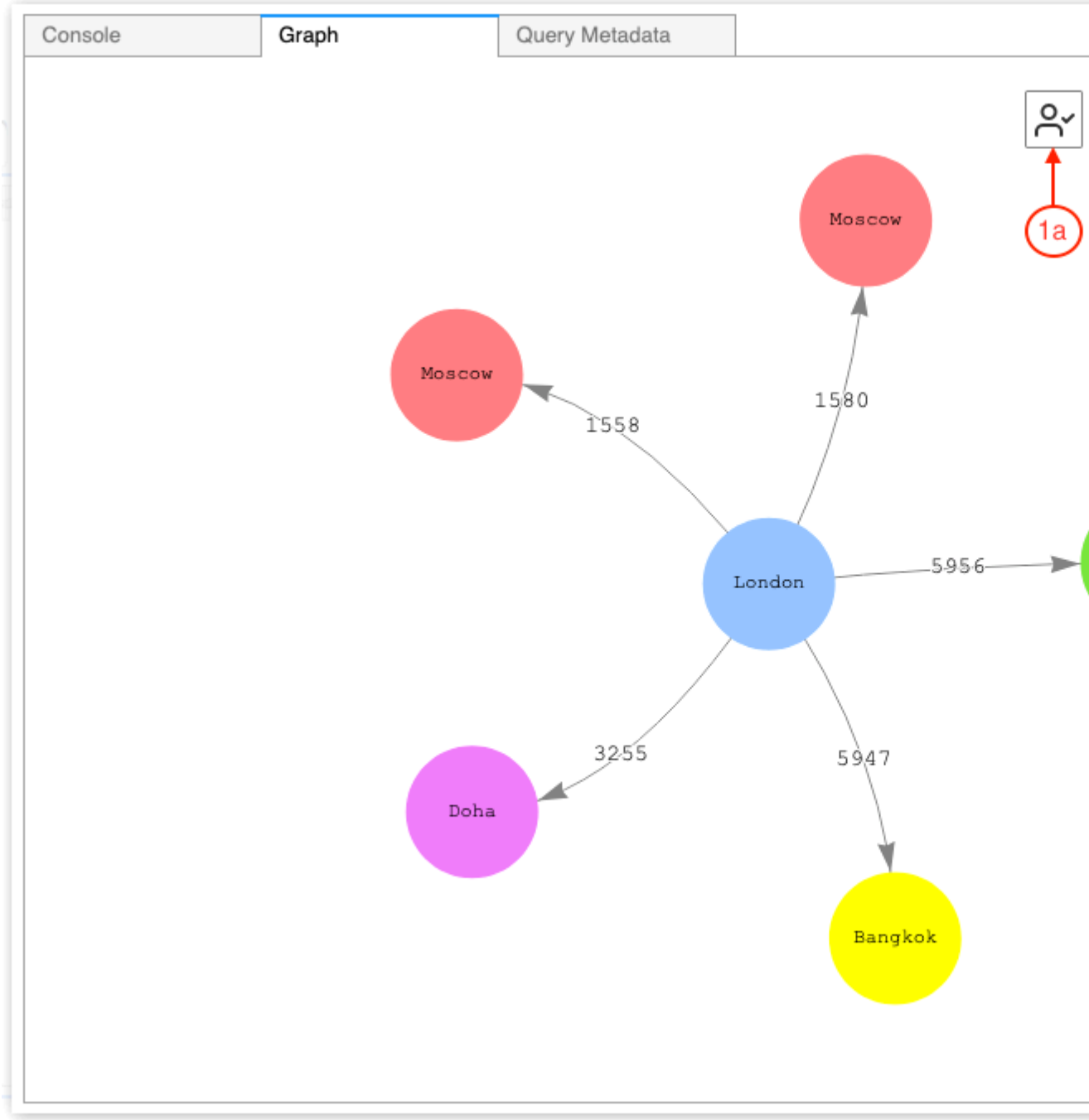
在許多情況下，Neptune 工作台可以創建查詢結果的可視圖表，以及以表格形式返回它們。只要有可能進行視覺化，就可以在查詢結果的「圖形」標籤中使用圖形視覺效果。

除了此處描述的內建視覺化功能之外，您還可以搭配 Neptune 圖形筆記本使用更進階的視覺化工具 (p. 476)。

> Note
>
> 若要存取您正在使用的筆記本中最近新增的功能和修正程式，請先停止再重新啟動筆記本執行個體。

圖表標籤介面概觀

此圖表識別「圖表」標籤中存在的使用者介面元素：



1. 圖形搜尋
   a. UUID 切換：切換圖形搜尋中包含 ID 屬性值的內容。根據預設，ID 包含為啟用狀態。如果停用，則符合 ID 性質 (包括參考節點 ID 的邊屬性) 不會導致元素亮顯。
   b. 搜尋文字欄位：亮顯包含您在此處指定的文字字串的所有頂點和邊性質值。
2. 圖形重設 — 重新執行圖形物理模擬，並將縮放設定為符合視窗中的圖形。
3. 切換圖形物理 — 切換圖形物理模擬的運行。物理預設為啟用狀態，讓圖形動態變更。如果停用，則移動其他頂點時，頂點會保持鎖住的位置。
4. 詳細資料檢視 — 選取節點或邊緣時，會顯示元素的內容索引鍵和值清單 (如果查詢結果中有的話)。

5. 全螢幕檢視 — 展開圖表索引標籤視窗以符合螢幕大小。再按一下可最小化圖形標籤。
6. 縮放選項
   a. 放大
   b. 縮放重置：將縮放設定為佈滿圖表標籤視窗內的所有頂點。
   c. 縮小

可視化小精靈查詢結果

Neptune 工作台創建查詢結果的可視化任何 Gemlin 查詢返回path. 若要查看視覺效果，請在執行查詢後選取「主控台」索引標籤右側的「圖形」索引標籤。

您可以使用查詢視覺效果提示來控制視覺效果檢視器圖表查詢輸出的方式。這些提示遵循%%gremlin儲存格魔法，並在--path-pattern (或其簡短格式-p) 參數名稱前面加上：

```
%%gremlin -p comma-separated hints
```

您還可以使用--group-by（或-g）標誌來指定頂點的屬性，以將它們分組。這允許為不同的頂點組指定顏色或圖標。

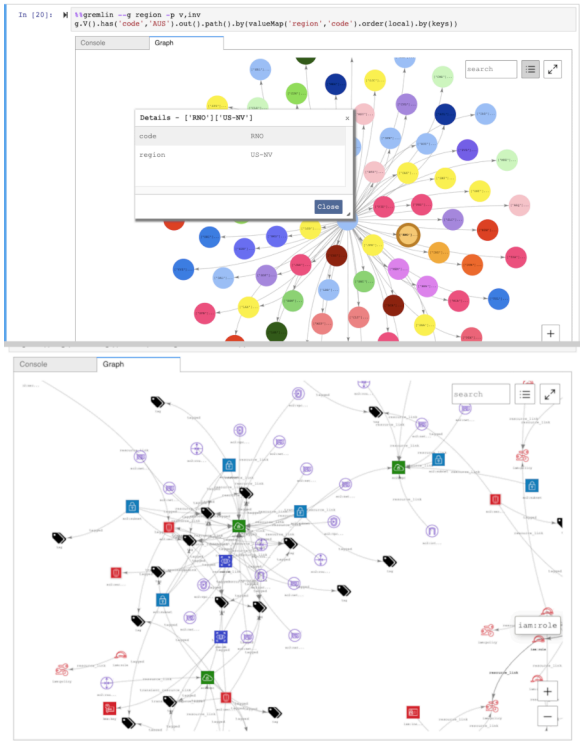
提示的名稱反映了在頂點之間遍歷時常用的 Grimlin 步驟，並且它們的行為相應。多個提示可以組合使用，用逗號分隔，它們之間沒有任何空格。使用的提示應符合要視覺化的查詢中對應的 Gemlin 步驟。請見此處範例：

```
%%gremlin -p v,oute,inv
g.V().hasLabel('airport').outE().inV().path().by('code').by('dist').limit(5)
```

可用的視覺效果提示如下：

```
v
inv
outv
e
ine
oute
```

以下是圖表視覺化的一些範例：



視覺化 SPARQL 查詢結果

Neptune 工作台會為採用下列其中一種形式之一的任何 SPARQL 查詢建立查詢結果的視覺效果：

- `SELECT ?subject ?predicate ?object`
- `SELECT ?s ?p ?o`

若要查看視覺效果，請在執行查詢之後，選取查詢下「表格」(Table) 標籤右側的「圖形」(Graph) 標籤。

依預設，SPARQL 視覺效果僅包含三重模式，其中`o?`為`uri`或`bnode` (空白節點)。所有其他`?o`繫結類型 (例如常值字串或整數) 都會被視為可使用「圖形」標籤中的「詳細資料」窗格檢視的`?s`節點屬性。

但是，在許多情況下，您可能想要在視覺效果中包含這些常值做為頂點。為此，請在`%%sparql`單元格魔術後使用`--expand-all`查詢提示：

```
%%sparql --expand-all
```

這告訴可視化器包括圖形圖中的所有`?s ?p ?o`結果，而不管綁定類型。

您可以看到整個`Air-Routes-SPARQL.ipynb`筆記本中使用的這個提示，您可以通過使用或不使用提示運行查詢來實驗，以查看它在視覺效果中的區別。

在 Neptune 工作台中存取視覺化教學課程

Neptune 工作台隨附的兩個視覺化教程筆記本提供了大量的例子在 Gremlin 和 SPARQL 如何有效地查詢圖形數據和可視化的結果。

請前往視覺化筆記本

1. 在左側的導覽窗格中，選擇開啟記事本。

2. 一旦 Neptune 工作台打開，運行 Jupyter，您將在頂層看到一個 Neptune 文件夾。選擇此資料夾予以開啟。
3. 在下一個級別是名為 02-可視化的文件夾。開啟此資料夾。裡面有幾個筆記本，引導您通過不同的方式來查詢您的圖形數據，在 Gremlin 和 SPARQL 中，以及如何可視化查詢結果：

   - 空氣路線小精靈
   - 航空路線
   - 工作台視覺效果
   - EPL-小鬼
   - EPL-分析

   選取要試驗其包含的查詢的記事本。

設定海王星

歡迎來到亞馬遜海王星。本節可協助您建立新的 Neptune 資料庫叢集，並在 Neptune 文件中尋找您要尋找的內容。

Note

如需AWS圖形資料庫參考架構和參考部署架構，請參閱 Amazon Neptune 資源。這些資源可協助您了解所選的圖形資料模型及查詢語言，加速您的部署過程。

主題

- 選擇正確的 Neptune 資料庫執行個體 (p. 52)
- 建立新的海王星資料庫叢集 (p. 54)
- 設定亞馬遜海王星資料庫叢集所在的亞馬遜 VPC (p. 60)
- 連接到亞馬遜海王星圖 (p. 63)
- 在亞馬遜海王星保護您的資料 (p. 65)
- 開始存取您的海王星圖 (p. 67)
- 將資料載入海王星 (p. 71)
- 亞馬遜海王星 (p. 72)
- Neptune 的疑難排解和最佳做法 (p. 72)

選擇正確的 Neptune 資料庫執行個體

Amazon Neptune 提供許多不同的執行個體大小和系列，提供適用於不同圖形工作負載的不同功能。本節旨在協助您選擇最符合需求的執行個體類型。

如需這些系列中每個執行個體類型的定價，請參閱 Neptune 定價頁面。

執行處理資源配置概要

Neptune 中使用的每個 Amazon EC2 執行個體類型和大小都提供定義數量的運算 (vCPU) 和系統記憶體。Neptune 的主要儲存裝置位於叢集中的資料庫執行個體外部，可讓運算和儲存容量彼此獨立擴充。

本節重點介紹如何擴展運算資源，以及每個執行個體系列之間的差異。

在所有執行個體系列中，vCPU 資源都會分配給每個 vCPU 支援兩 (2) 個查詢執行緒。此支援由執行個體大小決定。判斷指定 Neptune 資料庫執行個體的適當大小時，您需要考慮應用程式可能的並行性，以及查詢的平均延遲。您可以按照以下方式估算所需的 vCPU 數量，其中延遲是以秒為單位的平均查詢延遲來衡量，並行是以每秒查詢的目標數量來衡量：

$$vCPUs = \frac{latency \times concurrency}{2}$$

Note

SPARQL 查詢、OpenCypher 查詢，以及使用 DFE 查詢引擎的 Gremlin 讀取查詢，在某些情況下，每個查詢可以使用一個以上的執行緒。在最初調整資料庫叢集的大小時，請先假設每個查詢每次執行都會使用單一執行緒，並且如果您觀察到查詢佇列的背壓，請向上擴充。這可以透過使用`/gremlin/status/oc/status`、或 `/sparql/status` API 來觀察，也可以使用`MainRequestsPendingRequestsQueue`CloudWatch量度來觀察。

每個執行個體上的系統記憶體分為兩個主要配置：緩衝集區快取和查詢執行緒記憶體。

執行個體中約有三分之二的可用記憶體會分配給緩衝池快取。緩衝池緩存用於緩存圖形的最近使用的組件，以便在重複訪問這些組件的查詢更快地訪問。具有較大系統記憶體的執行個體擁有較大的緩衝集區快取，可在本機儲存更多圖形。使用者可以監控中提供的緩衝區快取命中和遺漏測量結果，以調整適當的緩衝區集區快取數量。CloudWatch

如果快取命中率在一段時間內降至 99.9% 以下，您可能會想要增加執行個體的大小。這表明緩衝區集區不夠大，而且引擎必須從基礎儲存磁碟區擷取資料的頻率高於效率。

剩餘的三分之一的系統記憶體會平均分佈在查詢執行緒之間，還有一些記憶體剩餘供作業系統使用，以及一個小型動態集區供執行緒視需要使用。每個執行緒可用的記憶體會從一個執行個體大小略有增加到下一個執行個體類型，直到8xl執行個體類型，每個執行緒配置的記憶體大小會達到上限。

新增更多執行緒記憶體的時間是當您遇到 OutOfMemoryException (OOM) 時。當一個執行緒需要超過分配給它的最大記憶體時，就會發生 OOM 例外狀況 (這與整個執行個體耗盡記憶體不足)。

t3和t4g執行個體類型

執行個體t3和t4g系列提供低成本的選項，可讓您開始使用圖形資料庫，也可用於初始開發和測試。這些執行個體符合 Neptune 免費方案的資格，可讓新客戶在獨立AWS帳戶中使用的前 750 個執行個體小時免費使用 Neptune，或在具有合併帳單 (付款人帳戶) 的AWS組織下彙整。

t3和t4g執行個體僅提供中型大小組態 (t3.medium和t4g.medium)。

由於這些執行個體的資源非常有限，因此不建議您使用它們來測試查詢執行時間或整體資料庫效能。若要評估查詢效能，請升級至其他執行個體系列之一。

r4例證類型族群

已取代 — Neptune 於 2018 年推出時提供此r4系列產品，但現在較新的執行個體類型可提供更好的價格/效能。從引擎版本 1.1.0.0 (p. 915) 開始，Neptune 不再支援r4執行個體類型。

r5例證類型族群

此r5系列包含記憶體最佳化執行個體類型，適用於大多數圖形使用案例。r5族群包含r5.large最多的例證類型r5.24xlarge。隨著大小的增加，它們會以線性方式擴展運算效能。例如，一個 r5.xlarge (4 個 vCPU 和 32GiB 記憶體) 的 vCPU 和記憶體是 r5.large (2 個 vCPU 和 16GiB 記憶體) 的兩倍，而一個 r5.2xlarge (8 個 vCPU 和 64GiB 記憶體) 的 vCPU 和記憶體則是. r5.xlarge 您可以預期查詢效能會隨著運算容量直接擴充至r5.12xlarge執行個體類型。

r5執行個體系列具有雙插槽 Intel CPU 架構。較小r5.12xlarge的類型使用單插槽和該單插槽處理器所擁有的系統記憶體。r5.16xlarge和r5.24xlarge類型同時使用通訊端和可用記憶體。由於雙插槽架構中的兩個實體處理器之間需要一些記憶體管理額外負荷，因此從 a r5.12xlarge 擴展到r5.16xlarge或r5.24xlarge執行個體類型的效能提升並不像您以較小的大小擴展那樣線性。

r5d例證類型族群

Neptune 具有查找緩存功 (p. 107)能，可用於改善需要獲取和返回大量屬性值和文字的查詢的性能。查詢需要傳回許多屬性的客戶主要使用此功能。查閱快取會藉由本機擷取這些屬性值，而不是在 Neptune 索引儲存區中反覆查詢每個屬性值，藉此提升這些查詢的效能。

查閱快取是使用執行個r5d體類型上的 NVMe 連接 EBS 磁碟區來實作。它是使用叢集的參數群組來啟用。由於資料是從 Neptune 索引儲存區擷取，屬性值和 RDF 常值會在此 NVMe 磁碟區中快取。

如果您不需要查閱快取功能，請使用標準r5執行個體類型而非標準執行個體類型r5d，以避免r5d.

r5d族群具有與r5族群大小相同的例證類型 (從r5d.large到) r5d.24xlarge。

r6g例證類型族群

AWS已經開發了自己的基於 ARM 的處理器，稱為引力，提供比英特爾和 AMD 等價物更好的價格/性能。此系r6g列使用重力敦 2 處理器。在我們的測試中，Graviton2 處理器可為 OLTP 樣式（受限）的圖形查詢提供 10-20% 更好的性能。然而，由於記憶體分頁效能的效能略低，因此使用 Graviton2 處理器時，較大的 OLAP-ish 查詢效能可能會略低於 Intel 處理器的效能。

同樣重要的是要注意，該系r6g列具有單插槽架構，這意味著效能會隨著運算容量從一個線性擴展r6g.large到一個r6g.16xlarge（系列中最大的類型）。

r6i例證類型族群

Amazon R6i 執行個體由第三代 Intel Xeon 可擴展處理器 (代碼名為 Ice Lake) 提供支援，非常適合需要大量記憶體的工作負載。一般而言，與同類 R5 執行個體類型相比，每個 vCPU 可提供高達 15% 的運算價格效能，以及高達 20% 的記憶體頻寬。

x2g例證類型族群

當實例具有較大的緩衝池緩存時，某些圖形用例可以看到更好的性能。該x2g系列的啟動是為了更好地支持這些用例。此x2g系列的 memory-to-v CPU 比例大於r5或r6g系列。這些x2g執行個體也使用 Graviton2 處理器，並且具有許多與r6g執行個體類型相同的效能特性，以及較大的緩衝池快取記憶體。

如果您是 CPU 使r5用率低且緩衝池快取遺漏率高的執行個r6g體類型，請嘗試改用x2g系列。這樣，您將獲得所需的額外內存，而無需支付更多 CPU 容量。

serverless例證類型

Neptune 無伺服器 (p. 577)功能可以根據工作負載的資源需求動態擴展執行個體大小。Neptune Serverless 可讓您為資料庫叢集中的執行個體設定運算容量的下限和上限 (以 Neptune 容 (p. 578)量單位計算)，而不是計算應用程式所需的 vCPU 數量。使用不同使用率的工作負載可以使用無伺服器而不是經過預設的執行個體進行成本最佳化

您可以在相同的資料庫叢集中設定佈建執行個體和無伺服器執行個體，以達到最佳的成本效能組態。

建立新的海王星資料庫叢集

建立新的 Amazon Neptune 資料庫叢集最簡單的方法是使用可為您建立所有必要資源的AWS CloudFormation範本，而無需手動執行所有操作。AWS CloudFormation範本會為您執行大部分的設定，包括建立 Amazon 彈性運算雲端 (Amazon EC2) 執行個體：

使用AWS CloudFormation範本啟動新的 Neptune 資料庫叢集

1. 建立具有使用 Neptune 資料庫叢集所需許可的新 IAM 使用者，如中所述IAM 使用者許可 (p. 525)。
2. 設定使用AWS CloudFormation範本所需的其他必要條件，如中所述使用AWS CloudFormation設定海王星的先決條件 (p. 56)。
3. 呼叫AWS CloudFormation堆疊，如中所述使用AWS CloudFormation堆疊建立海王星資料庫叢集 (p. 59)。

您也可以建立橫跨多個 Neptune 全域資料庫 (p. 73)AWS 區域，以啟用低延遲的全域讀取，並在中斷影響整AWS 區域體的極少數情況下提供快速復原。

如需使用手動建立 Amazon Neptune 叢集的相關資訊AWS Management Console，請參閱使用啟動 Neptune 資料庫叢集AWS Management Console (p. 534)。

您也可以使用AWS CloudFormation範本建立 Lambda 函數以搭配 Neptune 使用 (請參閱用於建立AWS CloudFormation要在海王星中使用的 Lambda 函數 (p. 640))。

如需在 Neptune 中管理叢集和執行個體的一般資訊，請參閱管理亞馬遜海王星數據庫 (p. 515)。

使用AWS CloudFormation設定海王星的先決條件

使用AWS CloudFormation範本建立 Amazon Neptune 叢集之前，您必須具備下列項目：

- 一個亞馬遜 EC2 密鑰對。
- 使用所需的權限AWS CloudFormation。

建立 Amazon EC2 金鑰配對，以便使用下列方式啟動海王星叢集 AWS CloudFormation

若要使用AWS CloudFormation範本啟動 Neptune 資料庫叢集，您必須在建立堆疊的區域中提供 Amazon EC2Key 配對 (及其關聯的 PEM 檔案)。AWS CloudFormation

如果您需要建立金鑰對，請參閱 Amazon EC2 Linux 執行個體使用者指南中的使用 Amazon EC2 建立金鑰對，或在 Amazon EC2 Windows 執行個體使用者指南中的使用 Amazon EC2 建立金鑰配對以取得指示。

新增 IAM 政策以授予使用AWS CloudFormation範本所需的許可

首先，您需要讓 IAM 使用者設定使用 Neptune 所需的許可，如中所述建立具有 Neptune 許可的 IAM 使用者使用者 (p. 525)。

然後，`AWSCloudFormationReadOnlyAccess`您需要將AWS受管理的策略新增至該使用者。

最後，您需要建立下列客戶管理的政策，並將其新增至該使用者：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "rds:CreateDBCluster",
          "rds:CreateDBInstance"
      ],
      "Resource": [
          "arn:aws:rds:*:*:*"
      ],
      "Condition": {
          "StringEquals": {
              "rds:DatabaseEngine": ["graphdb","neptune"]
          }
      }
    },
    {
      "Action": [
        "rds:AddRoleToDBCluster",
        "rds:AddSourceIdentifierToSubscription",
        "rds:AddTagsToResource",
        "rds:ApplyPendingMaintenanceAction",
        "rds:CopyDBClusterParameterGroup",
        "rds:CopyDBClusterSnapshot",
        "rds:CopyDBParameterGroup",
        "rds:CreateDBClusterParameterGroup",
        "rds:CreateDBClusterSnapshot",
        "rds:CreateDBParameterGroup",
        "rds:CreateDBSubnetGroup",
        "rds:CreateEventSubscription",
        "rds:DeleteDBCluster",
        "rds:DeleteDBClusterParameterGroup",
```

```
        "rds:DeleteDBClusterSnapshot",
        "rds:DeleteDBInstance",
        "rds:DeleteDBParameterGroup",
        "rds:DeleteDBSubnetGroup",
        "rds:DeleteEventSubscription",
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSecurityGroups",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEngineDefaultClusterParameters",
        "rds:DescribeEngineDefaultParameters",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeOptionGroups",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DescribeValidDBInstanceModifications",
        "rds:DownloadDBLogFilePortion",
        "rds:FailoverDBCluster",
        "rds:ListTagsForResource",
        "rds:ModifyDBCluster",
        "rds:ModifyDBClusterParameterGroup",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyDBSubnetGroup",
        "rds:ModifyEventSubscription",
        "rds:PromoteReadReplicaDBCluster",
        "rds:RebootDBInstance",
        "rds:RemoveRoleFromDBCluster",
        "rds:RemoveSourceIdentifierFromSubscription",
        "rds:RemoveTagsFromResource",
        "rds:ResetDBClusterParameterGroup",
        "rds:ResetDBParameterGroup",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:RestoreDBClusterToPointInTime"
      ],
      "Effect": "Allow",
      "Resource": [
          "*"
      ]
    },
    {
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs",
        "kms:ListAliases",
        "kms:ListKeyPolicies",
        "kms:ListKeys",
```

```
        "kms:ListRetirableGrants",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "sns:ListSubscriptions",
        "sns:ListTopics",
        "sns:Publish"
      ],
      "Effect": "Allow",
      "Resource": [
          "*"
      ]
    },
    {
      "Action": "iam:PassRole",
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "iam:passedToService": "rds.amazonaws.com"
        }
      }
    },
    {
      "Action": "iam:CreateServiceLinkedRole",
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
      "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "rds.amazonaws.com"
        }
      }
    }
  ]
}
```

Note

刪除堆疊僅需下列許
可：iam:DeleteRole、iam:RemoveRoleFromInstanceProfile、iam:DeleteRolePolicy、iam:DeleteI
及 ec2:DeleteVpcEndpoints。
亦請注意 ec2:*Vpc授予 ec2:DeleteVpc 許可。

使用AWS CloudFormation堆疊建立海王星資料庫叢集

您可以使用AWS CloudFormation範本來設定 Neptune 資料庫叢集。

1. 若要從 AWS CloudFormation 主控台啟動 AWS CloudFormation 堆疊，請選擇下表中的 Launch Stack (啟動堆疊) 按鈕之一。

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
亞太區域 (雪梨)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國東部)	檢視	在設計工具中檢視	Launch Stack

2. 在 Select Template (選取範本) 頁面上，請選擇 Next (下一步)。
3. 在 [指定詳細資訊] 頁面上，選擇 EC2SSH 的金鑰配對KeyPairName。

   存取 EC2 執行個體需要用到這個金鑰對。確認您具有所選金鑰對的 PEM 檔案。
4. 選擇 下一步。
5. 在 Options (選項) 頁面上，選擇 Next (下一步)。
6. 在 [檢閱] 頁面上，選取第一個核取方塊，以確認AWS CloudFormation將建立 IAM 資源。選取第二個核取方塊，確認新堆疊 `CAPABILITY_AUTO_EXPAND`。

   Note

   `CAPABILITY_AUTO_EXPAND` 明確確認在建立堆疊時，無需事先檢閱，即可擴充巨集。使用者通常會在處理過的範本中建立變更集，以便在實際建立堆疊之前，檢閱巨集所做的變更。如需詳細資訊，請參閱 AWS CloudFormationCreateStack API。

   然後選擇 Create (建立)。

   Note

   您也可以使用AWS CloudFormation範本升級資料庫叢集的引擎版本。 (p. 556)

設定亞馬遜海王星資料庫叢集所在的亞馬遜 VPC

Amazon Neptune 資料庫叢集只能在 Amazon 虛擬私有雲端 (Amazon VPC) 中建立，而且只能在該 VPC 內存取其端點。根據您想要存取資料庫叢集的方式，有許多不同的方法可以設定 VPC。

如果您在 2013 年 12 月 4 日之後建立AWS帳戶，則每個AWS區域都已有預設的虛擬私人雲端 (VPC)。如果您不確定是否有預設 VPC，請參閱 Amazon VPC 使用者指南中的偵測您是否擁有預設 VPC。有很多方法可以設定 VPC 或多個 VPC。如需預設 VPC 的詳細資訊，請參閱預設 VPC 與預設子網路。如需有關建立您自己的 VPC 的資訊，請參閱《Amazon VPC 使用者指南》。

如果您使用 Neptune 主控台建立資料庫叢集，則可以讓 Neptune 自動為您建立 VPC。Neptune 也可以為您建立 VPC 安全群組和資料庫子網路群組，或者您可以自行建立安全群組和/或子網路群組，如下列各節所述。如何設定 VPC 將取決於您打算連線到 Neptune 資料庫叢集的 (p. 63)方式。

設定 Neptune 資料庫叢集所在的 VPC 時，請注意以下幾點：

- 您的 VPC 必須至少有兩個子 (p. 61)網路。這些子網路必須位於兩個不同的可用區域 (AZ) 中。Neptune 將叢集執行個體分配至至少兩個 AZ，即使在可用區域故障不太可能發生故障的情況下，Neptune 也能確保資料庫叢集中始終有可用的執行個體。Neptune DB 叢集的叢集磁碟區永遠跨越三個 AZ，以提供耐用的儲存裝置，同時極低的資料遺失可能性。
- 每個子網路中的 CIDR 區塊必須足夠大，才能提供 Neptune 在維護活動、容錯移轉和調整期間可能需要的 IP 位址。
- VPC 必須具有包含您已建立子網路的資料庫子網路群組。Neptune 會選擇子網路群組中的其中一個子網路，以及該子網路內的 IP 位址，以便與資料庫叢集中的每個資料庫執行個體建立關聯。然後資料庫執行個體位於與子網路相同的 AZ 中。
- 虛擬私人雲端應該已啟用 DNS (p. 62) (DNS 主機名稱和 DNS 解析)。
- VPC 必須具有 VPC 安全性群組 (p. 62)，才能允許存取您的資料庫叢集。
- Neptune VPC 中的租用應設定為預設值。

將子網路新增至 Neptune 資料庫叢集所在的 VPC

子網是您的 VPC 中的 IP 地址範圍。您可以將 Neptune 資料庫叢集或 EC2 執行個體等資源啟動至特定子網路。建立子網時，您需為該子網指定 IPv4 CIDR 區塊，其即是 VPC CIDR 區塊的子網。每個子網路必須完全位於一個可用區域 (AZ) 內，而且不能跨越區域。透過在不同的可用區域中啟動執行個體，您可以保護應用程式免於其中一個區域發生故障。如需詳細資訊，請參閱 VPC 子網路文件。

Neptune 資料庫叢集至少需要兩個 VPC 子網路。

將子網路新增至 VPC

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中，選擇 Subnets (子網)。
3. 在 VPC 儀表板中，選擇「子網路」，然後選擇「建立子網路」。
4. 在 [建立子網路] 頁面上，選擇要在其中建立子網路的 VPC。
5. 在子網路設定下，進行下列選擇：
   a. 在子網路名稱下輸入新子網路的名稱。
   b. 為子網路選擇可用區域 (AZ)，或將選項保留為無偏好設定。
   c. 在 IPv4 CIDR 區塊下輸入子網路的 IP 位址區塊。
   d. 如果需要，請將標籤新增至子網路。
   e. 選擇
6. 如果您要同時建立其他子網路，請選擇 [新增子網路]。
7. 選擇 [建立子網路] 以建立新的子網路。

在 VPC 中建立子網路群組

建立子網路群組。

建立海王星子網路群組

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 選擇 Subnet groups (子網路群組)，然後選擇 Create DB Subnet Group (建立資料庫子網路群組)。
3. 輸入新子網路群組的名稱和說明 (需要說明)。

4. 在 VPC 下，選擇要放置此子網路群組的 VPC。
5. 在「可用性區域」下，選擇您要放置此子網路群組的 AZ。
6. 在子網路下，將此 AZ 中的一或多個子網路新增至此子網路群組。
7. 選擇 [建立] 以建立新的子網路群組。

使用 VPC 主控台建立安全群組

安全群組可讓您存取 VPC 中的 Neptune 資料庫叢集。它們充當關聯資料庫叢集的防火牆，可在執行個體層級控制輸入和輸出流量。根據預設，資料庫執行個體是使用防火牆和預設安全群組建立的，以防止任何人存取該執行個體。若要啟用存取權，您必須擁有具有其他規則的 VPC 安全群組。

下列程序說明如何新增自訂 TCP 規則，以指定 Amazon EC2 執行個體用於存取 Neptune 資料庫叢集的連接埠範圍和 IP 地址。您可以使用指派給 EC2 執行個體的 VPC 安全群組，而不是使用其 IP 位址。

在主控台上為 Neptune 建立 VPC 安全性群組

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在主控台的右上角，選擇您要為 Neptune 建立 VPC 安全性群組的AWS區域。在該區域的 Amazon VPC 資源清單中，應顯示您至少有一個 VPC 和多個子網路。如果沒有，表示您在該區域中沒有預設 VPC。
3. 在 [安全性] 下的功能窗格中，選擇 [安全性群組]
4. 選擇 Create Security Group (建立安全群組)。在 [建立安全性群組] 視窗中，輸入安全性群組名稱、說明，以及 Neptune DB 叢集所在之 VPC 的識別碼。
5. 為要連接到 Neptune 資料庫叢集之 Amazon EC2 執行個體的安全群組新增輸入規則：
   a. 在「輸入規則」區域中，選擇「新增規則」。
   b. 在 [類型] 清單中，保持選取 [自訂 TCP]。
   c. 在 [連接埠範圍] 方塊中，輸入 8182 (Neptune 的預設連接埠值)。
   d. 在來源下，輸入您要存取 Neptune 的 IP 位址範圍 (CIDR 值)，或選擇現有的安全性群組名稱。
   e. 如果您需要新增更多 IP 位址或不同的連接埠範圍，請再次選擇 [新增規則]。
6. 在 [輸出規則] 區域中，您也可以視需要新增一或多個輸出規則。
7. 完成後，請選擇 Create security group (建立安全群組)。

您可以在建立新的 Neptune 資料庫叢集時使用這個新的 VPC 安全性群組。

如果您使用預設 VPC，系統已經為您建立橫跨所有 VPC 子網路的預設子網路群組。當您在 Neptune 主控台中選擇 [建立資料庫] 時，除非您指定不同的 VPC，否則會使用預設 VPC。

請確定您的虛擬私人雲端中有 DNS 支援

網域名稱系統 (DNS) 是一種在網際網路上用於解析為對應 IP 地址的標準名稱。DNS 主機名稱為獨特的電腦名稱並由主機名稱和網域名稱組成。DNS 伺服器會將 DNS 主機名稱解析為對應的 IP 地址。

請檢查以確定 VPC 中已同時啟用 DNS 主機名稱和 DNS 解析。VPC 網路屬性 `enableDnsHostnames` 和 `enableDnsSupport` 必須設定為 `true`. 要查看和修改這些屬性，請移至位於 https://console.aws.amazon.com/vpc/ 的 VPC 主控台。

如需詳細資訊，請參閱以 VPC 使用 DNS。

> Note
>
> 如果您使用的是 Route 53，請確認您的組態未覆寫虛擬私人雲端中的 DNS 網路屬性。

連接到亞馬遜海王星圖

建立 Neptune DB 叢集之後，下一個步驟就是設定要連線到叢集的方式。

設定curl或 awscurl 以與您的海王星端點通訊

如本文件中的許多範例所示，擁有將查詢提交至 Neptune DB 叢集的命令列工具非常方便。curl 命令列工具是在未啟用 IAM 身份驗證時與 Neptune 端點通訊的絕佳選項。從 7.75.0 開始的版本支援在啟用 IAM 身份驗證時簽署請求的`--aws-sigv4`選項。

對於啟用 IAM 身份驗證的端點，您還可以使用 awscurl，該 awscurl 使用幾乎完全相同的語法，*`curl`*但支持 IAM 身份驗證所需的簽署請求。由於 IAM 身份驗證提供了額外的安全性，因此啟用它通常是個好主意。

如需有關如何使用 `curl` (或`awscurl`) 的詳細資訊，請參閱 curl 線上手冊和「所有內容捲曲」一書。

若要使用 HTTPS (海王星需要) 進行連線，`curl`需要存取適當的憑證。只要 `curl` 可以找到適當的憑證，它處理 HTTPS 連線的方式就跟 HTTP 連線一樣，無需額外參數。同樣是真實的`awscurl`. 此文件的範例是以該案例為基礎。

若要了解如何取得這類憑證，以及如何將這些憑證正確格式化為`curl`可以使用的憑證授權單位 (CA) 憑證存放區，請參閱`curl`文件中的 SSL 憑證驗證。

然後，您可以使用 `CURL_CA_BUNDLE` 環境變數來指定此 CA 憑證存放區的位置。在 Windows 上，`curl` 會自動在名為 `curl-ca-bundle.crt` 的檔案中尋找它。它會先在和 `curl.exe` 相同的目錄中尋找，然後再尋找路徑的其他位置。如需詳細資訊，請參閱 SSL Certificate Verification。

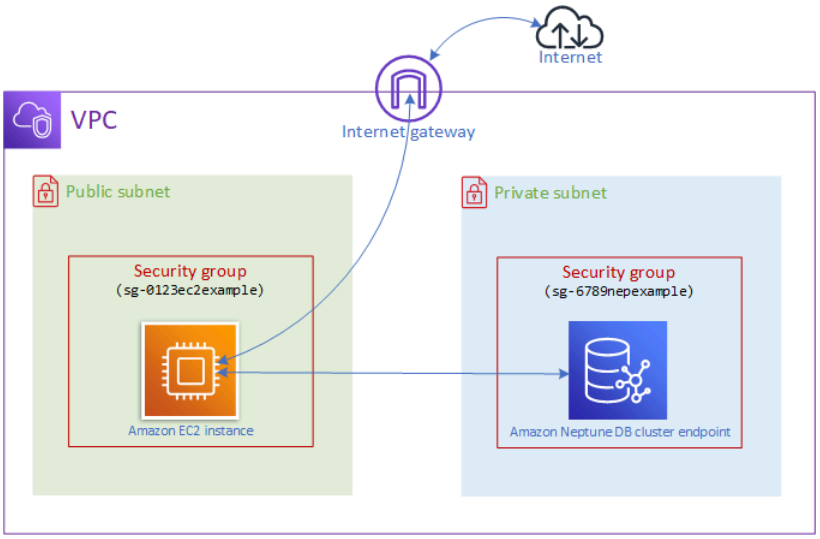
連接至 Neptune 資料庫叢集的不同方式

Amazon Neptune 資料庫叢集只能在 Amazon 虛擬私有雲端 (Amazon VPC) 中建立，而端點只能在該 VPC 內存取。有許多不同的方法可以在其 VPC 中設定 Neptune 資料庫叢集的存取權：

- 從相同 VPC 中的亞馬遜 EC2 實例進行連接 (p. 63)
- 從另一個 VPC 中的亞馬遜 EC2 實例連接 (p. 65)
- 從私人網路連線 (p. 65)

從相同 VPC 中的 Amazon EC2 執行個體連接到海王星資料庫叢集

連接到 Neptune 資料庫的最常見方式之一，就是從與 Neptune 資料庫叢集位於相同 VPC 中的 Amazon EC2 執行個體。例如，EC2 執行個體可能正在執行與網際網路連線的 Web 伺服器。在這種情況下，只有 EC2 執行個體可以存取 Neptune 資料庫叢集，而網際網路只能存取 EC2 執行個體：



若要啟用此設定，您必須設定正確的 VPC 安全性群組和子網路群組。Web 伺服器託管在公有子網路中，因此它可以連線到公用網際網路，而 Neptune 叢集執行個體則裝載在私有子網路中，以確保其安全。請參閱設定亞馬遜海王星資料庫叢集所在的亞馬遜 VPC (p. 60)。

為了讓 Amazon EC2 執行個體在上連接到您的 Neptune 端點 (例如連接埠)8182，您需要設定一個安全群組來執行此操作。例如，如果您的 Amazon EC2 執行個體使用名ec2-sg1為的安全群組ec2-sg1，則需要建立另一個具有連接埠輸入規則8182且具有其來源的 Amazon EC2 安全群組 (假設db-sg1)。然後，新增db-sg1至您的 Neptune 叢集以允許連線。

建立 Amazon EC2 執行個體之後，您可以使用 SSH 登入該執行個體，然後連線到 Neptune 資料庫叢集。如需使用安全殼層連線至 EC2 執行個體的詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的連線到 Linux 執行個體。

如果您使用 Linux 或 macOS 命令列連接到 EC2 執行個體，您可以從 堆疊 Outputs (輸出) 區段的 SSHAccessAWS CloudFormation 項目貼上 SSH 命令。您必須有目前目錄的 PEM 檔案，而且此 PEM 檔案許可必須設定為 400 (`chmod 400 keypair.pem`)。

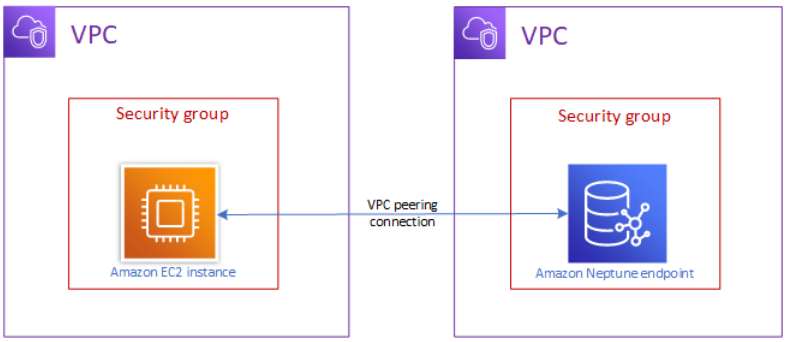
若要同時使用私有子網路和公用子網路建立 VPC

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在 AWS Management Console 的右上角，選擇要在其中建立 VPC 的區域。
3. 在 VPC 儀表板中，選擇啟動 VPC 精靈。
4. 完成 [建立 VPC] 頁面的 [VPC 設定] 區域：
    a. 在 Resources to create (建立資源) 下，選擇 VPC, subnets, etc. (VPC、子網路等)。
    b. 保持預設名稱標籤不變，或輸入您選擇的名稱，或取消勾選「自動產生」核取方塊以停用產生名稱標籤。
    c. 將 IPv4 CIDR 區塊值保留在。`10.0.0.0/16`
    d. 將 IPv6 CIDR 區塊值保留在沒有 IPv6 CIDR 區塊。
    e. 將租賃保留為預設值。
    f. 將可用區域 (AZ) 的數目保留為 2。
    g. 除非您需要一或多個 NAT 閘道，否則請將 NAT 閘道 ($) 保留為 「無」。
    h. 將虛擬私人雲端節點設定為 「無」，除非您將使用 Amazon S3。
    i. 啟用 DNS 主機名稱和啟用 DNS 解析都應該檢查。
5. 選擇 Create VPC (建立 VPC)。

從另一個 VPC 中的 Amazon EC2 執行個體存取您的資料庫叢集

Amazon Neptune 資料庫叢集只能在 Amazon 虛擬私有雲端 (Amazon VPC) 中建立，而且只能在該 VPC 內存取其端點，通常可從該 VPC 中執行的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體存取。

當您的資料庫叢集與您用來存取的 EC2 執行個體位於不同的 VPC 中時，您可以使用 VPC 對等互連來建立連線：



VPC 對等連線是兩個 VPC 之間的網路連線，以私密路由傳送流量，因此任一 VPC 中的執行個體都可以像在同一個網路中一樣進行通訊。您可以在帳戶中的 VPC 之間、帳戶中的 VPC 與其AWS他帳戶中的 VPC 之間建立 VPC 對等連線，或使用不同區域中的 VPC 建立 VPC 對等連線。AWS AWS

AWS 會使用現有的 VPC 基礎設施建立 VPC 對等互連。它既不是閘道器，也不是AWS站台對站台 VPN 連線，而且不依賴於單獨的實體硬體。它沒有單點的通信故障，也沒有帶寬瓶頸。

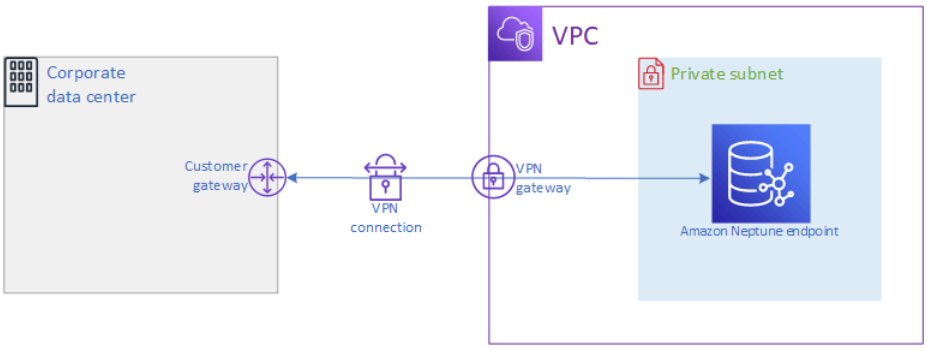
如需有關如何使用 VPC 對等互連的詳細資訊，請參閱 Amazon VPC 對等互連指南。

從私人網路存取資料庫叢集

您可以使用兩種不同的方式從私人網路存取 Neptune 資料庫叢集：

- 使用AWS網站對站台 VPN 連線。
- 使用AWS直接連接連接。

上面的鏈接包含有關這些連接方法以及如何設置它們的信息。站AWS台對站台連線的設定可能如下所示：



在亞馬遜海王星保護您的資料

您可以透過多種方式保護 Amazon 海王星叢集的安全。

使用 IAM 政策限制對 Neptune 資料庫叢集的存取

若要控制誰可以在 Neptune 資料庫叢集和資料庫執行個體上執行 Neptune 管理動作，請使用 AWS Identity and Access Management (IAM)。

當您使用 IAM 帳戶存取 Neptune 主控台時，必須先AWS Management Console使用您的 IAM 帳戶登入，然後才能在 https://console.aws.amazon.com/neptune/home 開啟 Neptune 主控台。

當您連線到AWS使用 IAM 登入資料時，您的 IAM 帳戶必須具有 IAM 政策，以授與執行 Neptune 管理作業所需的許可。如需詳細資訊，請參閱使用不同種類的 IAM 政策來控制對 Neptune 的存取 (p. 177)。

使用 VPC 安全群組限制對 Neptune 資料庫叢集的存取

海王星資料庫叢集必須在亞馬遜虛擬私有雲 (Amazon VPC) 中建立。若要控制哪些裝置和 EC2 執行個體可以在 VPC 中為 Neptune 資料庫叢集開啟與資料庫執行個體端點和連接埠的連線，請使用 VPC 安全群組。如需 VPCs 的詳細資訊，請參閱「使用 VPC 主控台建立安全群組 (p. 62)」。

使用 IAM 身分驗證限制對 Neptune 資料庫叢集的存取

如果您在 Neptune 資料庫叢集中啟用 AWS Identity and Access Management (IAM) 身份驗證，則存取資料庫叢集的任何人都必須先通過驗證。亞馬遜海王星AWS Identity and Access Management (IAM) 概述 (p. 157)如需設定 IAM 身份驗證的相關資訊，請參閱。

如需使用臨時登入資料進行驗證的相關資訊，包括AWS CLIAWS Lambda、和 Amazon EC2 的範例，請參閱the section called “暫時登入資料” (p. 218)。

下列連結提供有關透過個別查詢語言使用 IAM 身份驗證連線至 Neptune 的其他資訊：

使用帶有 IAM 身份驗證的小鬼

- the section called “Gremlin 主控台” (p. 163)
- the section called “Gremlin Java” (p. 165)
- the section called “Python 範例” (p. 169)

  Note

  此範例適用於 Gremlin 和 SPARQL。

使用帶有 IAM 身份驗證的開放密碼

- the section called “Gremlin 主控台” (p. 163)
- the section called “Gremlin Java” (p. 165)
- the section called “Python 範例” (p. 169)

  Note

  此範例適用於 Gremlin 和 SPARQL。

搭配身分驗證使用 SPARQL

- the section called “SPARQL Java (RDF4J 和 Jena)” (p. 167)
- the section called “Python 範例” (p. 169)

  Note

  此範例適用於 Gremlin 和 SPARQL。

開始存取您的海王星圖

一旦你已經創建了一個海王星數據庫集群和設置連接到它，下一步是與它進行通信，以便加載數據，進行查詢等。為此，大多數人都使用curl或命awscurl令列工具。

設定curl與您的海王星端點通訊

如本文件中許多範例所示，curl 命令列工具是與 Neptune 端點通訊的便利選項。如需工具資訊，請參閱 curl man 頁面和 Everything curl 一書。

若要使用 HTTPS 進行連線 (如我們所建議，以及 Neptune 在大多數區域中所需)，curl需要存取適當的憑證。若要了解如何取得這些憑證，以及如何將它們正確格式化為curl可以使用的憑證授權單位 (CA) 憑證存放區，請參閱說明curl文件中的 SSL 憑證驗證。

然後，您可以使用 CURL_CA_BUNDLE 環境變數來指定此 CA 憑證存放區的位置。在 Windows 上，curl 會自動在名為 curl-ca-bundle.crt 的檔案中尋找它。它會先在和 curl.exe 相同的目錄中尋找，然後再尋找路徑的其他位置。如需詳細資訊，請參閱 SSL Certificate Verification。

只要 curl 可以找到適當的憑證，它處理 HTTPS 連線的方式就跟 HTTP 連線一樣，無需額外參數。此文件的範例是以該案例為基礎。

使用查詢語言存取 Neptune 資料庫叢集中的圖形資料

一旦你連接，你可以使用 Gremlin 和 OpenCypher 查詢語言來創建和查詢屬性圖，或者 SPARQL 查詢語言來創建和查詢包含 RDF 數據的圖形。

海王星支援的圖形查詢語言

- 小鬼 (p. 300)是屬性圖的圖遍歷語言。在 Grimlin 查詢是由離散步驟組成的遍歷，每個步驟都遵循一個邊緣到一個節點。如需詳細資訊，請參閱 Apache TinkerPop 3 的小鬼文件。

  Gremlin 的海王星實現與其他實現有一些不同，特別是當您使用 Grimlin-Groovy (以序列化文本發送的 Gremlin 查詢) 時。如需詳細資訊，請參閱亞馬遜海王星的小精靈標準合規 (p. 83)。
- OpenCypher (p. 382) 是最初由 Neo4j 的，然後開源於 2015 年開發的屬性圖的聲明，並在一個阿帕奇 2 開源許可證下的 OpenCypher 項目做出了貢獻。它的語法記錄在 Cypher 查詢語言參考版本 9 中。
- SPARQL (p. 422) 是 RDF 資料的宣告式查詢語言，根據全球資訊網協會 (W3C) 標準化，並在 SPARQL 1.1 概觀) 和 SPARQL 1.1 查詢語言規格中所述的圖形模式比對。

  Note

  您可以使用小鬼和 OpenCypher 訪問海王星屬性圖數據，但不使用 SPARQL。同樣，您只能使用 SPARQL，而不是小靈或開放密碼訪問 RDF 數據。

使用小精靈訪問亞馬遜海王星中的圖形

您可以使用 Grimlin 控制台在 REPL (read-eval-print循環) 環境中嘗試TinkerPop圖形和查詢。

下面的教程將引導您通過使用 Gemlin 控制台添加頂點, 邊緣, 屬性, 和更多的海王星圖, 突出顯示在海王星特定的小鬼實現一些差異.

Note

此範例假設您已完成下列各項：

- 您已使用 SSH 連線至亞馬遜 EC2 執行個體。
- 您已經建立了海王星叢集，如中所述建立資料庫叢集 (p. 54)。
- 您已如安裝小精靈控制台 (p. 301)所述安裝 Gremlin 主控台。

使用 Gremlin 主控台

1. 將目錄變更為 Gremlin 主控台檔案解壓縮的資料夾。

```
cd apache-tinkerpop-gremlin-console-3.6.2
```

2. 輸入下列命令以執行 Gremlin 主控台。

```
bin/gremlin.sh
```

您應該會看到下列輸出：

```
         \,,,/
         (o o)
-----oOOo-(3)-oOOo-----
plugin activated: tinkerpop.server
plugin activated: tinkerpop.utilities
plugin activated: tinkerpop.tinkergraph
gremlin>
```

您現在進入 `gremlin>` 提示。您會在這個提示下進入其餘的步驟。

3. 出現`gremlin>`提示時，輸入以下指令以連線至 Neptune 資料庫執行個體。

```
:remote connect tinkerpop.server conf/neptune-remote.yaml
```

4. 在 `gremlin>` 提示下，輸入以下內容以切換為遠端模式。這會傳送所有 Gremlin 查詢到遠端連線。

```
:remote console
```

5. 新增頂點以及標籤和屬性。

```
g.addV('person').property('name', 'justin')
```

頂點會獲指派一個 `string` ID，包含 GUID。所有頂點 ID 都是海王星中的字串。

6. 新增頂點與自訂 ID。

```
g.addV('person').property(id, '1').property('name', 'martin')
```

`id` 屬性不使用括號括住。這是頂點 ID 的關鍵字。此處的頂點 ID 是一個字串，其中有數字 1。

正常屬性名稱必須包含在引號中。

7. 如果屬性不存在，請變更屬性或新增屬性。

```
g.V('1').property(single, 'name', 'marko')
```

您在這裡變更前一個步驟頂點的 `name` 屬性。這會移除 `name` 屬性所有現有的值。

如果您未指定 `single`，則會改為附加值到 `name` 屬性 (如果尚未這樣做)。

8. 新增屬性，但如果屬性已有值則附加屬性。

```
g.V('1').property('age', 29)
```

Neptune 使用設定基數作為預設動作。

此命令會新增 age 屬性以及值 29，但不會取代任何現有的值。

如果 age 屬性已有值，這個命令會附加 29 到屬性。例如，如果 age 屬性原本是 27，新的值會是 [27, 29]。

9. 新增多個頂點。

```
g.addV('person').property(id, '2').property('name', 'vadas').property('age',
 27).iterate()
g.addV('software').property(id, '3').property('name', 'lop').property('lang',
 'java').iterate()
g.addV('person').property(id, '4').property('name', 'josh').property('age',
 32).iterate()
g.addV('software').property(id, '5').property('name', 'ripple').property('lang',
 'java').iterate()
g.addV('person').property(id, '6').property('name', 'peter').property('age', 35)
```

您可以同時將多個語句發送到海王星。

陳述式可以使用換行符號 ('\n')、空格 (' ')、分號 ('; ') 來隔開，也可以不使用任何分隔符號 (例如：g.addV(‘person’).iterate()g.V() 為有效)。

> **Note**
>
> Gremlin 主控台會在每個新行 ('\n') 處傳送個別命令，所以在這種情況下每個都是單獨的交易。此範例將所有命令放在不同的行以方便閱讀。移除新行 ('\n') 字元可透過 Gremlin 主控台將其做為單一命令傳送。

最後一個陳述式除外的所有陳述式都必須結束在終止步驟，例如 .next() 或 .iterate()，否則不會執行。Gremlin 主控台不需要這些終止步驟。.iterate每當您不需要序列化結果時使用。

一起傳送的所有陳述式會包含在單一交易中，一起成功或一起失敗。

10. 新增邊緣。

```
g.V('1').addE('knows').to(__.V('2')).property('weight', 0.5).iterate()
g.addE('knows').from(__.V('1')).to(__.V('4')).property('weight', 1.0)
```

以下是新增邊緣的兩個不同方式。

11. 新增其他現代圖形。

```
g.V('1').addE('created').to(__.V('3')).property('weight', 0.4).iterate()
g.V('4').addE('created').to(__.V('5')).property('weight', 1.0).iterate()
g.V('4').addE('knows').to(__.V('3')).property('weight', 0.4).iterate()
g.V('6').addE('created').to(__.V('3')).property('weight', 0.2)
```

12. 刪除頂點。

```
g.V().has('name', 'justin').drop()
```

移除具有 name 屬性的頂點等於 justin。

> **Important**
>
> 停在這裡，你有完整的 Apache TinkerPop 現代圖。TinkerPop文件 「遍歷」區段中的範例使用「現代」圖表。

13. 執行周遊。

```
g.V().hasLabel('person')
```

傳回所有 person 頂點。

14. 使用值 (valueMap ()) 執行周遊。

```
g.V().has('name', 'marko').out('knows').valueMap()
```

傳回 marko「知道」的所有頂點的索引鍵值組。

15. 指定多個標籤。

```
g.addV("Label1::Label2::Label3")
```

海王星支持一個頂點的多個標籤。當您建立標籤時，您可以指定多重標籤並用 :: 分隔。

此範例新增有三種不同標籤的頂點。

hasLabel 步驟將比對此頂點和這三個標籤：hasLabel("Label1")、hasLabel("Label2") 和 hasLabel("Label3")。

:: 分隔符號僅針對本用途保留。

您不能在 hasLabel 步驟中指定多重標籤。例如，hasLabel("Label1::Label2") 不符合任何內容。

16. 指定時間/日期。

```
g.V().property(single, 'lastUpdate', datetime('2018-01-01T00:00:00'))
```

海王星不支援 Java 日期。請改用 datetime() 函數。datetime() 接受 ISO8061 相容 datetime 字串。

支援格式如下：YYYY-MM-DD, YYYY-MM-DDTHH:mm、YYYY-MM-DDTHH:mm:SS 和 YYYY-MM-DDTHH:mm:SSZ

17. 刪除頂點、屬性或邊緣。

```
g.V().hasLabel('person').properties('age').drop().iterate()
g.V('1').drop().iterate()
g.V().outE().hasLabel('created').drop()
```

以下是數個 drop 範例。

> **Note**
>
> .next() 步驟不適用於 .drop()。請改用 .iterate()。

18. 完成後，輸入以下內容以退出 Gremlin 主控台。

```
:exit
```

Note

使用分號 (;) 或換行符號字元 (\n) 來分隔每個陳述式。
最終周遊之前的每個周遊節尾必須為 iterate()，才能執行。但只有最後的周遊資料會傳回。

使用開放密碼訪問亞馬遜海王星圖

若要開始使用 OpenCypher，請參閱开放密码 (p. 382)或使用海王星圖形筆記本存放庫中的 OpenCypher 筆記本。GitHub

使用 RDF 和 SPARQL 訪問亞馬遜海王星中的圖形

SPARQL 是資源描述框架（RDF）的查詢語言，它是為網絡設計的圖形數據格式。亞馬遜海王星與 SPARQL 1.1 相容。這表示您可以連線到 Neptune 資料庫執行個體，並使用 SPARQL 1.1 查詢語言規格中所述的查詢語言查詢圖形。

SPARQL 的查詢包含 SELECT 子句，用於指定要傳回的變數，和 WHERE 子句，用於指定要比對圖形中的哪些資料。如果您不熟悉 SPARQL 查詢，請參閱 SPARQL 1.1 查詢語言中的編寫簡易查詢。

對海王星資料庫執行個體進行 SPARQL 查詢的 HTTP 端點為。`https://your-neptune-endpoint:port/sparql`

若要連接到 SPARQL

1. 您可以從AWS CloudFormation堆疊的 [輸出] 區段中的SparqlEndpoint項目取得 Neptune 叢集的 SPARQL 端點。
2. 輸入以下命令，以**UPDATE**使用 HTTP POST 和 curl 命令提交 SPARQL。

   ```
   curl -X POST --data-binary 'update=INSERT DATA { <https://test.com/s> <https://
   test.com/p> <https://test.com/o> . }' https://your-neptune-endpoint:port/sparql
   ```

   上述範例插入以下三元組到 SPARQL 預設圖形：<https://test.com/s> <https://test.com/p> <https://test.com/o>
3. 輸入以下命令，以**QUERY**使用 HTTP POST 和 curl 命令提交 SPARQL。

   ```
   curl -X POST --data-binary 'query=select ?s ?p ?o where {?s ?p ?o} limit 10'
    https://your-neptune-endpoint:port/sparql
   ```

   上述範例會使用限制為 10 的?s ?p ?o查詢，最多可傳回圖表中三元組 (subject-predicate-object) 的 10 個。若要查詢其他項目，請將查詢換成其他 SPARQL 查詢。

   Note

   SELECT 和 ASK 查詢回應的預設 MIME 類型為 application/sparql-results+json。
   CONSTRUCT 和 DESCRIBE 查詢回應的預設 MIME 類型為 application/n-quads。
   如需所有可用 MIME 類型的清單，請參閱 SPARQL HTTP API (p. 428)。

將資料載入海王星

Amazon Neptune 提供將資料直接從外部檔案載入 Neptune 資料庫執行個體的程序。您可以使用此程序，而非執行大量 INSERT 陳述式、addV 和 addE 步驟，或其他 API 呼叫。

以下是額外載入資訊的連結。

- 加載數據的方法 — 載入資料 (p. 255)
- 大量載入程式支援的資料格式 — the section called “資料格式” (p. 261)
- 加載示例-the section called “載入範例” (p. 270)

亞馬遜海王星

亞馬遜海王星支援下列監控方法。

- 亞馬遜 CloudWatch — Amazon Neptune 會自動將指標傳送到警報，CloudWatch並支援CloudWatch警報。如需詳細資訊，請參閱the section called “使用 CloudWatch” (p. 772)。
- AWS CloudTrail— 亞馬遜海王星支持使用CloudTrail. 如需詳細資訊，請參閱the section called “使用記錄海王星 API 呼叫 AWS CloudTrail” (p. 789)。
- 標記 — 使用標籤將中繼資料新增至 Neptune 資源，並根據標籤追蹤使用情況。如需詳細資訊，請參閱the section called “標記海王星資源” (p. 799)。
- 稽核記錄檔 — 使用 Neptune 主控台檢視、下載或監視資料庫記錄檔。如需詳細資訊，請參閱the section called “使用海王星稽核記錄” (p. 781)。
- 執行個體狀態 — 檢查 Neptune 執行個體圖形資料庫引擎的健全狀況、找出已安裝的引擎版本，以及使用執行個體狀態 API 取得其他引擎狀態 (p. 770)資訊。

Neptune 的疑難排解和最佳做法

下列連結可能有助於解決 Amazon Neptune 的問題。

- 最佳做法 — 如需常見問題和效能建議的解決方案，請參閱最佳實務 (p. 822)。
- 服務錯誤 — 如需管理 API 和圖形資料庫連線的錯誤清單，請參閱Neptune 錯誤 (p. 849)。
- 服務限制 — 如需 Neptune 限制的相關資訊，請參閱海王星極限 (p. 843)。
- 引擎版本 — 如需有關圖形引擎發行版本的資訊，包括版本說明，請參閱引擎版本 (p. 856)。
- 支援論壇 — 若要加入有關海王星的討論，請參閱亞馬遜海王星論壇。
- 定價 — 如需使用 Amazon Neptune 成本的相關資訊，請參閱亞馬遜海王星定價。
- AWS支援 — 如需專家的說明和指導，請參閱AWS Support。

建立 Neptune 全域資料庫

Amazon Neptune 全域資料庫跨越多個AWS 區域，可實現低延遲的全域讀取，並在中斷影響整體的罕見情況下提供快速復原AWS 區域。

Neptune 全域資料庫包含一個區域中的主要資料庫叢集，以及不同區域中最多五個次要資料庫叢集。

寫入只能在主要區域中發生。次要區域僅支援讀取。每個次要區域最多可有 16 個讀取器執行個體。

主題

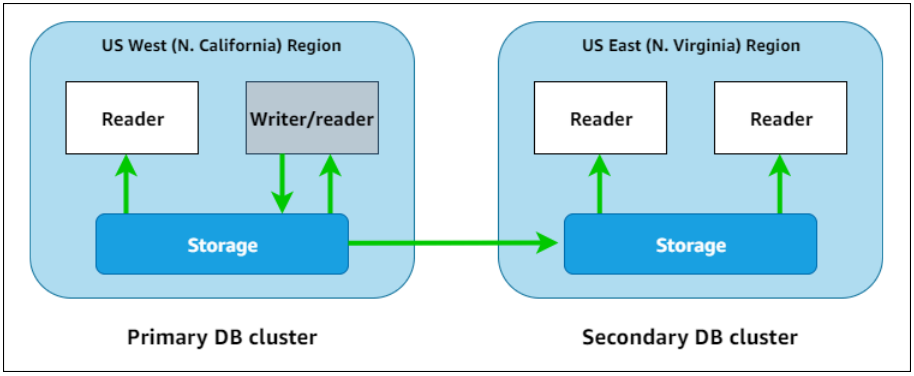
- Amazon Neptune 中的全域資料庫概觀 (p. 73)
- 在 Amazon Neptune 使用全域資料庫的優勢 (p. 74)
- Amazon Neptune 中的全域資料庫的限制 (p. 74)
- 在 Amazon Neptune 中設置全球數據庫 (p. 74)
- 管理 Amazon Neptune 全球資料庫 (p. 78)
- 在 Neptune 全域資料庫中使用容錯移 (p. 79)
- 使用度量監視 Neptune 全域 CloudWatch 資料庫 (p. 81)

Amazon Neptune 中的全域資料庫概觀

使用 Neptune 全域資料庫，您可以在跨多個的單一資料庫上執行全域分散式應用程式AWS 區域。

Neptune 全域資料庫包含一個寫入資料庫的主AWS 區域要資料庫叢集，以及次要資料庫叢集最多五個唯讀資料庫叢集AWS 區域。在主要資料庫叢集上執行寫入操作時，Neptune 會使用專用基礎設施將寫入的資料複寫到所有次要資料庫叢集，延遲通常不到一秒鐘。

下圖顯示跨越兩個的範例全域資料庫AWS 區域：



您可以透過新增一或多個僅供讀複本執行個體來獨立擴充每個次要叢集以處理唯讀工作負載。

若要執行寫入操作，您必須連接到主要資料庫叢集的資料庫叢集端。只有主要叢集可以執行寫入操作。然後，如上圖所示，複寫是由叢集儲存磁碟區 (p. 119)執行，而不是資料庫引擎。

Neptune 全域資料庫專為擁有全球足跡的應用程式所設計。唯讀次要資料庫叢集支援更接近應用程式使用者的讀取操作。

Neptune 全域資料庫支援兩種不同的容錯移轉方法：

- 若要從主要區域的中斷中斷中復原，請使用手動未計劃 (p. 79)的 detach-and-promote程序，在此處卸離其中一個次要叢集，將其轉換為獨立叢集，然後將其提升為新的主要叢集。
- 針對計劃的操作程序 (例如維護)，請使用受管的計劃容錯移轉 (p. 80)，在此將主要叢集重新放置到其中一個次要區域，而不會遺失資料。

在 Amazon Neptune 使用全域資料庫的優勢

使用全域資料庫，有下列優點：

- 以本機延遲提供全域讀取 — 如果您在世界各地設有辦公室，全域資料庫可讓次要區域的辦公室以本機延遲存取自己區域中的資料。
- 可擴充的次要 Neptune 資料庫叢集 — 您可以透過新增僅供讀取複本資料庫執行個體來擴 由於次要叢集是唯讀的，因此它們最多可支援 16 個僅供讀取複本，而不是通常的 15 個複本限制。
- 快速複寫到次要資料庫叢集 — 從主要資料庫叢集快速複寫到次要資料庫叢集 — 其延遲通常不到一秒鐘，對主要資料庫叢集的效能影響不大。由於複寫是在儲存層級執行，因此資料庫執行個體資源可用於應用程式讀取和寫入工作負載。
- 從全區域中斷中復原 — 次要資料庫叢集可讓您建立可用於新的區域，相較於傳統複寫解決方案，這個方法更快，RTO 率較低，且資料遺失率較低 (較低的 RPO)。

Amazon Neptune 中的全域資料庫的限制

全球資料庫目前有下列限制：

- Neptune 全域資料庫僅適用於下列項目AWS 區域：
  - 美國 (維吉尼亞)::::::us-east-1
  - 美國 (俄)::::::::us-east-2
  - 美國西部 (加州)::::::us-west-1
  - 美國 (奧)::::::::us-west-2
  - 歐 ()::::::::eu-west-1
  - 歐 (倫敦)::::::eu-west-2
  - 亞太 (()::::::::ap-northeast-1
- Neptune 全域資料庫不支援次要資料庫叢集的 auto-scaling。
- 在執行全域資料庫的主要版本升級時，無法為全域資料庫叢集套用自訂參數群組。而是在全域叢集的每個區域中建立自訂參數群組，然後在升級後手動將其套用至區域叢集。
- 您無法單獨停止或啟動全域資料庫中的資料庫叢集。
- 次要資料庫叢集中的僅供讀複本執行個體可在某些情況下重新啟動 如果主要叢集的寫入器執行個體重新啟動或容錯移轉，次要區域中的所有執行個體也會重新啟動。然後，次要叢集將無法使用，直到其所有執行個體都與主要資料庫叢集的寫入器執行個體同步。

在 Amazon Neptune 中設置全球數據庫

您可以使用下列其中一種方式建立 Neptune 全域資料庫：

- 使用所有新資料庫叢集和執行個體建立全域資料庫 (p. 75)。
- 使用現有的 Neptune 資料庫叢集作為主要叢集來建 (p. 76)立全域資料庫。

主題

- Amazon Neptune 中全域資料庫的組態需求 (p. 75)
- 使用在AWS CLI Amazon Neptune 中創建全局數據庫 (p. 75)
- 將現有的數據庫群集轉換為全局數據庫 (p. 76)
- 將次要全球資料庫區域新增至 Amazon Neptune 中的主要區域 (p. 77)
- 連線至 Neptune 全域資料庫 (p. 77)

Amazon Neptune 中全域資料庫的組態需求

Neptune 全域資料庫至少跨越兩個AWS 區域。主要AWS 區域包含具有一個寫入器執行個體的 Neptune 資料庫叢集。1 到 5AWS 區域 個次要每個次要叢集包含完全由僅供讀取複本組成的唯讀 Neptune 資料庫叢集。至少需要選取一AWS 區域個次要選項。

組成全域資料庫的 Neptune 資料庫的 Neptune 資料庫叢集具有下列特定需求：

- 資料庫執行個體類別需求 — 全域資料庫需要`r5`或`r6g`針對記憶體密集型工作負載最佳化的資料庫執行個體 (例如`db.r5.large`執行個體類型)。
- AWS 區域需求 — 全域資料庫至少需要一個中的主要 Neptune 資料庫叢集AWS 區域，而在不同區域中至少有一個次要 Neptune 資料庫叢集。您最多可以建立五個次要唯讀 Neptune 資料庫叢集，而且每個叢集必須位於不同的區域。換句話說，沒有兩個 Neptune 全域資料庫中沒有兩個 Neptune 資料庫中的 Neptune 資料庫中沒有兩AWS 區域個
- 引擎版本需求 — 全域資料庫中所有資料庫叢集所使用的 Neptune 引擎版本應相同，且必須大於或等於`1.2.0.0`。如果您在建立新的全域資料庫或叢集或執行個體時未指定引擎版本，則會使用最新的引擎版本。

> **Important**
>
> 雖然可以針對全域資料庫中的每個資料庫叢集獨立設定資料庫叢集參數群組，但是最好讓叢集之間的設定保持一致，以避免次要叢集升級為主叢集時發生意外的行為變更。例如，在所有資料庫叢集中的物件索引、串流等使用相同的設定。

使用在AWS CLI Amazon Neptune 中創建全局數據庫

> **Note**
>
> 本節中的範例遵循使用反斜線 (\) 作為行延伸字元的 UNIX 慣例。如果是 Windows，請使用插入符號 (^) 取代反斜線。

使用建立全域資料庫AWS CLI

1. 首先使用`create-global-cluster`AWS CLI命令（包裝CreateGlobalCluster (p. 1042) API）創建一個空的全局數據庫。指定您要作為主要AWS 區域的名稱，將 Neptune 設定為資料庫引擎，並選擇性地指定您要使用的引擎版本 (必須是版本 1.2.0.0 或更高版本)：

```
aws neptune create-global-cluster
  --region (primary region, such as us-east-1) \
  --global-cluster-identifier (ID for the global database) \
  --engine neptune \
  --engine-version (engine version; this is optional)
```

2. 全局數據庫可能需要幾分鐘的時間才能使用，因此在進行下一步之前，請使用`describe-global-clusters` CLI 命令（包裝DescribeGlobalClusters (p. 1047) API）來檢查全局數據庫是否可用：

```
aws neptune describe-global-clusters \
  --region (primary region) \
```

```
  --global-cluster-identifier (global database ID)
```

3. Neptune 全域資料庫可供使用之後，您可以建立新的 Neptune DB 叢集做為其主要叢集：

```
aws neptune create-db-cluster \
  --region (primary region) \
  --db-cluster-identifier (ID for the primary DB cluster) \
  --engine neptune \
  --engine-version (engine version; must be >= 1.2.0.0) \
  --global-cluster-identifier (global database ID)
```

4. 使用命describe-db-clustersAWS CLI令確認新的資料庫叢集已準備好新增其主要資料庫執行個體：

```
aws neptune describe-db-clusters \
  --region (primary region) \
  --db-cluster-identifier (primary DB cluster ID)
```

當回應顯示時"Status": "available"，請繼續下一個步驟。

5. 使用create-db-instanceAWS CLI指令建立主要叢集的主要資料庫執行個體。您必須使用其中一種記憶體最佳化r5或r6g執行個體類型，例如db.r5.large。

```
aws neptune create-db-instance \
  --region (primary region) \
  --db-cluster-identifier (primary cluster ID) \
  --db-instance-class (instance class) \
  --db-instance-identifier (ID for the DB instance) \
  --engine neptune \
  --engine-version (optional: engine version)
```

Note

如果您打算使用 Neptune 大量載入器將資料新增至新的主要資料庫叢集，請在新增次要區域之前執行此動作。這比在全域資料庫完全設定之後執行大量載入更快速且更具成本效益。

現在，將一或多個次要區域新增至新的全域資料庫，如中所述使用新增次要區域AWS CLI (p. 77)。

將現有的數據庫群集轉換為全局數據庫

若要將現有的資料庫叢集轉換為全域資料庫，請使用create-global-clusterAWS CLI指令在與現有資料庫叢集所在位置AWS 區域相同的位置建立新的全域資料庫，並將其--source-db-cluster-identifier參數設定為位於該處的現有叢集的 Amazon 資源名稱 (ARN)：

```
aws neptune create-global-cluster \
  --region (region where the existing cluster is located) \
  --global-cluster-identifier (provide an ID for the new global database) \
  --source-db-cluster-identifier (the ARN of the existing DB cluster) \
  --engine neptune \
  --engine-version (engine version; this is optional)
```

現在，將一或多個次要區域新增至新的全域資料庫，如中所述使用新增次要區域AWS CLI (p. 77)。

使用從快照還原的資料庫叢集做為主要叢集

您可以將從快照還原的資料庫叢集轉換為 Neptune 全域資料庫。還原完成之後，如上所述，將它建立的資料庫的資料庫的資料庫的資料庫的資料庫的資料庫的資料庫的資料庫的資料庫的

將次要全球資料庫區域新增至 Amazon Neptune 中的主要區域

Neptune 全域資料庫至少需要一個與主要資料庫在AWS 區域不同中的次要 Neptune 資料庫叢集。您最多可以將五個次要資料庫叢集連接至主要資料庫叢集。

您新增的每個次要資料庫叢集會減少一個您在主要叢集上可擁有的僅供讀取複本執行個體數目上限。例如，如果有 4 個次要叢集，則主要叢集上可擁有的僅供讀取複本執行個體數目上限為15 - 4 = 11。這表示如果主要資料庫叢集和 1 個次要叢集中的讀取器執行個體總數合計為 14，就無法新增另一個次要叢集。

使用將AWS CLI次要區域新增至 Neptune 中的全域資料庫

使用將次要資料庫新增AWS 區域至 Neptune 全域資料庫AWS CLI

1. 使用命create-db-clusterAWS CLI令在與主要叢集不同的區域中建立新的資料庫叢集，並設定其--global-cluster-identifier參數以指定全域資料庫的 ID：

```
aws neptune create-db-cluster \
  --region (the secondary region) \
  --db-cluster-identifier (ID for the new secondary DB cluster) \
  --global-cluster-identifier (global database ID)
  --engine neptune \
  --engine-version (optional: engine version)
```

2. 使用命describe-db-clustersAWS CLI令確認新的資料庫叢集已準備好新增其主要資料庫執行個體：

```
aws neptune describe-db-clusters \
  --region (primary region) \
  --db-cluster-identifier (primary DB cluster ID)
```

   當回應顯示時"Status": "available"，請繼續下一個步驟。

3. 使用r5或執行個體類別中的執行個體類型，使用create-db-instanceAWS CLI指令建立主要叢集的主要資料庫r6g執行個體：

```
aws neptune create-db-instance \
  --region (secondary region) \
  --db-cluster-identifier (secondary cluster ID) \
  --db-instance-class (instance class) \
  --db-instance-identifier (ID for the DB instance) \
  --engine neptune \
  --engine-version (optional: engine version)
```

   Note

   如果您不打算在次要區域中提供大量讀取請求，而且主要擔心將資料保持可靠地備份至該區域，則可以建立不含資料庫執行個體的次要資料庫叢集。這樣可以節省成本，因為您只需支付次要叢集的儲存費用，Neptune 將與主要資料庫叢集中的儲存保持同步。

連線至 Neptune 全域資料庫

連接到 Neptune 全域資料庫的方式取決於您是否需要寫入或從中讀取：

- 對於唯讀請求或查詢，請連線到中 Neptune 叢集的讀取器端點AWS 區域。
- 若要執行突變查詢，請連線至主要資料庫叢集的叢集端點，該端點可能與您的應用程式AWS 區域不同。

管理 Amazon Neptune 全球資料庫

除了受管計劃容錯移轉程序外，您還可在構成 Neptune 全球資料庫的個別叢集上執行大部分的管理操作。受管計劃容錯移轉程序僅適用於 Neptune 全球資料庫，不適用於個別 Neptune 資料庫叢集。如需進一步了解，請參閱 執行 Neptune 全域資料庫的受管計劃容錯移轉 (p. 80)。

若要從其主要區域的計劃外中斷復原 Neptune 全球資料庫，請參閱Detach-and-promote 在意外中斷的情況下，Neptune 全球數據庫 (p. 79)。

雖然您可以為全球資料庫中的每個 Neptune 叢集獨立設定資料庫叢集參數群組，但在所有叢集之間，最好保持一致的設定，以避免當次要叢集升級為主要叢集時發生非預期的行為變更。例如，在所有資料庫叢集中，對物件索引、串流等使用相同的設定。

從 Neptune 全域資料庫移除資料庫叢集

有幾個原因可促使您從全球資料庫移除資料庫叢集。例如：

- 如果主要叢集已降級或隔離，您可以將其從全域資料庫中移除，它就會變成可用來建立新全域資料庫的獨立佈建叢集 (請參閱Detach-and-promote 在意外中斷的情況下，Neptune 全球數據庫 (p. 79))。
- 如果想要刪除全球資料庫，首先必須從中移除 (分離) 所有關聯的叢集，將主要叢集設定為最後刪除 (請參閱刪除 Neptune 全域資料庫 (p. 78)。

您可以使用 remove-from-global-clusterCLI 命令 (包裝RemoveFromGlobalCluster (p. 1050) API) 從全域資料庫中分離 Neptune DB 叢集：

```
aws neptune remove-from-global-cluster \
  --region (region of the cluster to remove) \
  --global-cluster-identifier (global database ID) \
  --db-cluster-identifier (ARN of the cluster to remove)
```

然後，分離的資料庫叢集會變成獨立的資料庫叢集。

刪除 Neptune 全域資料庫

您無法在單一步驟中刪除全球資料庫和相關聯的叢集。相反，您必須逐個刪除其組件：

1. 如中所述，從全球資料庫中卸離所有次要資料庫叢集移除叢集 (p. 78)。如果想要，您現在可以單獨刪除它們。
2. 從全球資料庫中卸離主要資料庫叢集。
3. 從主要叢集刪除所有僅供讀取複本資料庫執行個體。
4. 從主要叢集刪除主要 (寫入器) 資料庫執行個體。如果您在控制台上執行此操作，它也會刪除數據庫集群。
5. 刪除全域資料庫本身。若要使用執行此操作AWS CLI，請使用 delete-global-clusterCLI 命令 (它會包裝DeleteGlobalCluster (p. 1044) API)，如下所示：

```
aws neptune delete-global-cluster \
  --region (region of the DB cluster to delete) \
  --global-cluster-identifier (global database ID)
```

修改 Neptune 全域資料庫

您可以針對全域資料庫中的每個 Neptune DB 叢集獨立設定資料庫叢集參數群組，但最好讓叢集之間的設定保持一致，以避免次要叢集升級為主叢集時發生意外的行為變更。

您可以使用 modify-global-clusterCLI 命令 (包裝ModifyGlobalCluster (p. 1045) API) 來修改全域資料庫本身的設定。例如，您可以變更全域資料庫識別碼，同時如下所示關閉刪除保護：

```
aws neptune modify-global-cluster \
  --region (region of the DB cluster to modify) \
  --global-cluster-identifier (current global database ID) \
  --new-global-cluster-identifier (new global database ID to assign) \
  --deletion-protection false
```

在 Neptune 全域資料庫中使用容錯移

Neptune 全域資料庫提供的容錯移轉功能比獨立的 Neptune 資料庫叢集更全面。使用全域資料庫，您可以相當快速地規劃並從災難復原。災難復原通常使用復原時間目標 (RTO) 和復原點目標 (RPO) 評估：

- 復原時間目標 (RTO) — 這是系統在災難後傳回工作狀態的快速。換言之，RTO 會測量停機時間。對於 Neptune 全域資料庫，RTO 可以按分鐘來排序。
- 復原點目標 (RPO) — 遺失資料的時間量。對於 Neptune 全域資料庫，RPO 通常以秒為單位進行測量 (請參閱執行 Neptune 全域資料庫的受管計劃容錯移轉 (p. 80))。

對於 Neptune 全域資料庫，存在兩種不同的容錯移轉方法：

- Detach-and-promote (手動計劃外復原) — 若要從計畫外的中斷情形復原或執行災難復原測試 (DR 測試)，可以針對全域資料庫中 detach-and-promote 的其中一個次要資料庫叢集執行跨區域。此手動程序的 RTO 取決於您在 分離並提升 (p. 79) 中執行所列任務的速度。RPO 通常為秒，但這取決於發生故障時網路上的儲存體複寫延遲。
- 受管理的計劃容錯移轉 — 此方法適用於作業維護和其他規劃的作業程序，例如將全域資料庫的主要資料庫叢集重新配置到其中一個次要區域。由於此程序會同步次要資料庫叢集與主要資料庫叢集，然後再做出任何其他變更，因此 RPO 實際上是 0 (也就是說，不會遺失資料)。請參閱 執行 Neptune 全域資料庫的受管計劃容錯移轉 (p. 80)。

Detach-and-promote 在意外中斷的情況下，Neptune 全球數據庫

在非常罕見的情況下，您的 Neptune 全域資料庫在其主要資料庫發生計劃時AWS 區域，您的主要 Neptune 資料庫叢集及其寫入器節點會變成無法使用，而主要叢集和次要叢集之間的複寫則會停止。若要將產生的停機時間 (RTO) 和資料遺失 (RPO) 減至最低，可以快速執行跨區域 detach-and-promote 來重建全域資料庫。

Tip

在使用此過程之前先了解這個過程是一個好主意，並制定一個計劃，以便在全區域問題的第一個跡象時快速進行。

- CloudWatch 定期使用 Amazon 追蹤次要叢集的延遲時間，以便在需要容錯移轉時，可以識別延遲時間最短的次要區域。
- 請務必測試您的計劃，以檢查程序是否完整且準確。
- 使用模擬環境來確保您的員工受過訓練，並準備好在必要時快速執行 DR 容錯移轉。

在主要區域發生計劃外中斷後容錯移轉至次要叢集

1. 停止在主資料庫叢集上發出突變查詢和其他寫入作業。
2. 識別次要資料庫叢集中的資料庫叢集，AWS 區域以用作全域資料庫的新主要資料庫叢集叢集。如果全域資料庫有兩個或更多次要叢集AWS 區域，請選擇具有最小延遲時間的次要叢集。

3. 卸離您從 Neptune 全域資料庫選擇的次要資料庫叢集。

   若從 Neptune 全域資料庫移除次要資料庫叢集，會立即停止從主要資料庫叢集複寫到該次要資料庫叢集，並將其提升為具有完整讀取/寫入功能的獨立資料庫叢集。全域資料庫中的任何其他次要叢集仍可供使用，而且可以接受應用程式的讀取呼叫。

   在重新建立 Neptune 全域資料庫之前，您還必須卸離其他次要叢集，以避免叢集之間的資料不一致 (請參閱移除叢集 (p. 78))。
4. 重新設定您的應用程式，以使用其新端點將所有寫入操作傳送至您選擇成為新主要叢集的獨立 Neptune 資料庫叢集，使用其新端點將所有寫入 如果您在建立 Neptune 全域資料庫時接受預設名稱，則可透過`-ro`從應用程式中叢集的端點字串移除來變更端點。

   例如，當該叢集從全域資料庫分離`my-global.cluster-`**`ro`**`-aaaaaabbbbbb.us-west-1.neptune.amazonaws.com``my-global.cluster-aaaaaabbbbbb.us-west-1.neptune.amazonaws.com`時，次要叢集的端點會變成。

   當您在下一個步驟中開始在下一個步驟中新增區域時，這個 Neptune 資料庫叢集會成為新的 Neptune 全域資料庫的主要叢集。
5. 將 AWS 區域 新增至資料庫叢集。當您執行這項操作時，從主要到次要的複寫程序即會開始。請參閱 將次要全球資料庫區域新增至 Amazon Neptune 中的主要區域 (p. 77)。
6. 視需新增更多 AWS 區域，以重新建立支援您的應用程式所需的拓撲。

請確定應用程式寫入會在做出這些變更之前、期間和之後，傳送至正確的 Neptune 資料庫叢集。這樣可以避免 Neptune 全域資料庫中資料庫叢集之間的資料庫叢集之間出現資料不一致的情況 (這些稱為分裂大腦問題)。

執行 Neptune 全域資料庫的受管計劃容錯移轉

受管計劃容錯移轉可讓您AWS 區域隨時將 Neptune 全域資料庫的主要叢集重新放置到不同的叢集。有些組織會想要定期輪換其主要叢集位置。

> Note
>
> 此處描述的受管計劃容錯移轉程序旨在用於運作狀態良好的 Neptune 全域資料庫上。若要從計畫外的中斷復原或執行災難復原 (DR) 測試，請遵循分離並提升 (p. 79)程序。

在受管計劃容錯移轉期間，您的主要叢集會容錯移轉至您選擇的次要區域，同時保留全域資料庫現有的複寫拓撲。在受管計劃容錯移轉程序開始之前，全域資料庫會將所有次要叢集與其主要叢集同步處理。確保所有叢集都已同步之後，受管計劃容錯移轉即會開始。主要區域中的資料庫叢集會變成唯讀，而選擇的次要叢集會將其中一個唯讀叢集提升為完整寫入器狀態，從而允許叢集擔任主要叢集的角色。由於所有次要叢集都在處理程序開始時與主要叢集同步，因此新的主要叢集會繼續執行全域資料庫的操作，而不會遺失任何資料。當主要叢集和選取的次要叢集會擔任其新角色時，資料庫只會在短時間內無法使用。

若要最佳化應用程式可用性，可以在非尖峰時段，在寫入主要資料庫叢集最少時，執行容錯移轉時，執行容錯移轉。此外，在開始容錯移轉之前，請遵循以下步驟：

- 盡可能讓應用程式離線，以減少對主要叢集的寫入。
- 檢查全域資料庫中所有次要 Neptune DB 叢集的延遲時間，並選擇整體延遲時間最少的次要叢集成為主要叢集。使用 Amazon CloudWatch 來檢視所有次要資料庫叢集的`NeptuneGlobalDBProgressLag`指標。此指標顯示次要資料庫叢集在主要資料庫叢集後面的距離 (以毫秒為單位)。其值與 Neptune 完成容錯移轉所需的時間成正比。換言之，延遲值越大，容錯移轉中斷就越長，容錯移轉中斷就越長，因此請選擇具有最小延遲的次要資料庫叢集。如需詳細資訊，請參閱 海王星CloudWatch度量 (p. 774)。

在受管計劃容錯移轉期間，選擇的次要資料庫叢集會提升為作為主要資料庫叢集的新角色，但不會繼承主要資料庫叢集的完整組態。組態不相符可能會導致效能問題、工作負載不相容，以及其他異常行為。若要避免此類問題，請在容錯移轉之前先解決全域資料庫叢集之間的下列組態差異：

- 在新的主要參數中設定參數以符合目前的主要參數。
- 設定監控工具、選項和警示 — 為使用與目前主要叢集相同的日誌記錄功能、警示等，為新的主要資料庫叢集設定與目前主要叢集的資料庫叢集設定相同的資料庫叢集。
- 設定與其他AWS服務的整合 — 如果您的 Neptune 全域資料庫與AWS Identity and Access Management (IAM)、Amazon S3 等AWS服務整合AWS Lambda，或者確定這些服務視需完成設定，以與新的主要資料庫叢集整合。

當容錯移轉程序完成且升級的資料庫叢集已準備好處理全域資料庫的寫入作業時，請務必變更您的應用程式，以使用新的主要端點。

使用起始受管理的AWS CLI計劃容錯移轉

使用 failover-global-clusterCLI 命令 (包裝FailoverGlobalCluster (p. 1048) API) 容錯移轉您的 Neptune 全域資料庫：

```
aws neptune failover-global-cluster \
  --region (the region where the primary cluster is located) \
  --global-cluster-identifier (global database ID) \
  --target-db-cluster-identifier (the ARN of the secondary DB cluster to promote)
```

Note

預覽中無法使用`failover-global-cluster` API。這將是 GA 發布的一部分。

使用度量監視 Neptune 全域 CloudWatch 資料庫

Neptune 支援下列可用來監視 Neptune 全域資料庫的 CloudWatch 度量：

- **`GlobalDbDataTransferBytes`**— Neptune 全域資料庫中從主要資料庫傳輸到次AWS 區域AWS 區域要重做日誌資料的位元組數目。
- **`GlobalDbReplicatedWriteIO`**— 從全域資料庫中的主要資料庫中複寫到輔助資料庫AWS 區域中的叢集磁碟區的叢集磁碟區的數目AWS 區域。

  Neptune ase 全域資料庫中每個資料庫叢集帳單計算使用該`VolumeWriteIOPS`指標來計算叢集內執行的寫入。對於主要資料庫叢集，計費計算用於將跨區域複寫`NeptuneGlobalDbReplicatedWriteIO`到次要資料庫叢集納入考量。
- **`GlobalDbProgressLag`**— 使用者交易和系統交易和系統交易和系統交易和系統交易和系統交易和系統交易和系統交易和系統交

指標	維度	出版於	個單位
`GlobalDbDataTransferBytes`	SourceRegion, 資料庫 ClusterIdentifier	Secondary	位元組
`GlobalDbReplicatedWriteIO`	SourceRegion, 資料庫 ClusterIdentifier	Secondary	計數
`GlobalDbProgressLag`	DBClusterIdentifier，SecondaryRegion ：在主數據庫中 ClusterIdentifier，SourceRegion ：在次要	主要、次要	毫秒

亞馬遜海王星功能概述

本節提供特定 Neptune 功能的概觀，包括：

- 海王星符合查詢語言 (p. 83)標準。
- 海王星的圖形資料模型 (p. 104)。
- 海王星交易語義的解釋 (p. 109)。
- Neptune 叢集和執行個體簡介 (p. 117)。
- Neptune 的儲存、可靠性和可用性 (p. 119)。
- 海王星端點的解釋 (p. 121)。
- 海王星的自訂查詢 ID 如何讓您檢查查詢狀態 (p. 126)。
- 使用 Neptune 的實驗室模式啟用實驗功能 (p. 127)。
- 海王星 DFE 引擎的描述 (p. 129)。
- 海王星的 JDBC 連線能力 (p. 148)。
- Neptune 引擎發行版本清單以及如何更新引擎 (p. 856)。

> **Note**
>
> 本節不涵蓋使用可用來存取 Neptune 圖形中資料的查詢語言。
> 如需如何使用 Grimlin 連線至執行中的 Neptune DB 叢集的詳細資訊，請參閱。訪問與小鬼海王星圖 (p. 300)
> 如需如何使用 OpenCypher 連線到執行中的海王星資料庫叢集的詳細資訊，請參閱。使用開放密碼訪問海王星圖 (p. 382)
> 如需如何使用 SPARQL 連線至執行中的 Neptune 資料庫叢集的詳細資訊，請參閱使用 SPARQL 存取海王星圖 (p. 422)。

主題

- 亞馬遜海王星標準合規的注意 (p. 83)
- 海王星圖數據模型 (p. 104)
- 海王星查閱快取可加速讀取查詢 (p. 107)
- 海王星中的事務語義 (p. 109)
- 亞馬遜海王星資料庫叢集和執 (p. 117)
- Amazon 海王星儲存、可靠性和可用性 (p. 119)
- 連接到亞馬遜海王星端點 (p. 121)
- 將自定義 ID 注入海王星小鬼或 SPARQL 查詢 (p. 126)
- 海王星實驗室 (p. 127)
- 亞馬遜海王星替代查詢引擎（DFE） (p. 129)
- 管理要使用的海王星 DFE 的統計資料 (p. 131)
- 獲取有關圖表的快速摘要報告 (p. 138)
- 亞馬遜海王星 JDBC 連接 (p. 148)
- 亞馬遜海神引擎更新 (p. 152)

亞馬遜海王星標準合規的注意

在大多數情況下，亞馬遜海王星在實施 Grinmlin 和 SPARQL 圖形查詢語言時都符合適用的標準。

這些部分描述了標準以及 Neptune 擴展或偏離標準的那些區域。

主題

- 亞馬遜海王星的小精靈標準合規 (p. 83)
- 亞馬遜海王星的 SPARQL 標準合規性 (p. 92)
- 亞馬遜海王星中的開放密碼標準合規性 (p. 95)

亞馬遜海王星的小精靈標準合規

以下各節提供了 Gremlin 的海王星實作概觀，以及它與 Apache TinkerPop 實作的不同之處。

海王星在其引擎中以原生方式實現了一些小鬼步驟，並使用 Apache TinkerPop Grimlin 實現來處理其他步驟（請參閱）。亞馬遜海王星的原生小精靈步進支持 (p. 377)

> Note
>
> 有關 Gramlin 控制台和 Amazon Neptune 中顯示的這些實施差異的一些具體示例，請參閱快速入門the section called “使用小精靈” (p. 67)章節。

主題

- Gremlin 的適用標準 (p. 83)
- 指令碼中的變數和參數 (p. 84)
- TinkerPop枚舉 (p. 84)
- Java 程式碼 (p. 85)
- 腳本執行 (p. 85)
- 工作階段 (p. 85)
- 交易 (p. 85)
- 頂點和邊 ID (p. 85)
- 使用者提供的 ID (p. 86)
- 頂點屬性 ID (p. 86)
- 頂點屬性的基數 (p. 86)
- 更新頂點屬性 (p. 86)
- 標籤 (p. 86)
- 跳脫字元 (p. 87)
- 常規局限性 (p. 87)
- 序列化 (p. 87)
- 拉姆達步驟 (p. 87)
- 不支援的小鬼方法 (p. 87)
- 不支援的小鬼步驟 (p. 88)
- 海王星中的小精靈圖特徵 (p. 88)

Gremlin 的適用標準

- 小鬼語言是由 Apache TinkerPop 文檔和阿帕奇TinkerPop實現的鬼魂，而不是由一個正式的規範來定義。

- 對於數字格式，格林遵循 IEEE 754 標準（IEEE 標準為浮點運算）。有關更多信息，另請參閱維基百科 IEEE 754 頁面）。

指令碼中的變數和參數

如果涉及預綁定變量，則遍歷對象在 Neptune 中g是預綁定的，並且不支持該graph對象。

雖然 Neptune 不支援指令碼中的 Gremlin 變數或參數化，但是您可能會在網際網路上遇到包含變數宣告的 Gemlin Server 範例指令碼，例如：

```
String query = "x = 1; g.V(x)";
List<Result> results = client.submit(query).all().get();
```

在提交查詢時，還有許多使用參數化（或綁定）的示例，例如：

```
Map<String,Object> params = new HashMap<>();
params.put("x",1);
String query = "g.V(x)";
List<Result> results = client.submit(query).all().get();
```

參數範例通常與關於在可能的情況下不參數化的效能損失的警告相關聯。有很多這樣的例TinkerPop子，你可能會遇到，他們都聽起來很有說服力的需要參數化。

但是，變數宣告功能和參數化功能 (以及警告) 都只適用於使用. TinkerPop `GremlinGroovyScriptEngine` 它們不適用於當小鬼伺服器使用 Grimlin 的 `gremlin-language` ANTLR 語法來解析查詢。ANTLR 語法不支持變量聲明或參數化，因此在使用 ANTLR 時，您不必擔心無法參數化。由於 ANTLR 語法是較新的元件 TinkerPop，因此您可能會在網際網路上遇到的較舊內容通常不會反映這種區別。

Neptune 在其查詢處理引擎中使用 ANTLR 語法，而不是使用`GremlinGroovyScriptEngine`，因此它不支援變數或參數化或屬性。`bindings`因此，與參數化失敗有關的問題不適用於 Neptune。使用 Neptune，只需按照通常參數化的位置提交查詢是完全安全的。因此，前面的示例可以簡化，而不會造成任何性能損失，如下所示：

```
String query = "g.V(1)";
List<Result> results = client.submit(query).all().get();
```

TinkerPop枚舉

Neptune 不支援列舉值的完整類別名稱。例如，您必須在 Groovy 請求中使用 `single`，而不是 `org.apache.tinkerpop.gremlin.structure.VertexProperty.Cardinality.single`。

列舉類型由參數類型決定。

下表顯示允許的列舉值和相關的TinkerPop完整名稱。

允許值	類別
`id`, `key`, `label`, `value`	org.apache.tinkerpop.gremlin.structure.T
`T.id`, `T.key`, `T. ##`, `t. #`	org.apache.tinkerpop.gremlin.structure.T
`set`, `##`	或者. 修補程式. 格林. VertexProperty. 基數
`ASC`, `##`, `##`	org.apache.tinkerpop.gremlin.process.traversal.Order
`##`, `#####`, `##. ##`	org.apache.tinkerpop.gremlin.process.traversal.Order

`global`, `##`	org.apache.tinkerpop.gremlin.process.traversal.Scope
`##. ##`, `##. ##`	org.apache.tinkerpop.gremlin.process.traversal.Scope
`##`, `##`, `##`, `###`	org.apache.tinkerpop.gremlin.process.traversal.Pop
`normSack`	或者. 修補程序. 格林. 進程遍歷. SackFunctions. 障礙物
`####`, `##`, `##`, `##`, `max`, `##`, `##`, `#`, `#`, `sum`, `###`	org.apache.tinkerpop.gremlin.process.traversal.Operator
`keys`, `values`	org.apache.tinkerpop.gremlin.structure.Column
`BOTH`, `IN`, `OUT`	org.apache.tinkerpop.gremlin.structure.Direction
`##`, `#`	或者. 修補程序. 格林. 進程. TraversalOptionParent 挑選。

Java 程式碼

海王星不支援呼叫任意 Java 或 Java 程式庫呼叫所定義的方法，而不是支援的小鬼 API。例如，不允許 `java.lang.*`、`Date()` 和 `g.V().tryNext().orElseGet()`。

腳本執行

所有查詢開頭必須為周遊物件 g。

在字串查詢提交中，您可以發出多重周遊，並以分號 (`;`) 或換行符號字元 (\n) 分隔。若要執行，除了最後一個以外的每個陳述式結尾必須是 `.iterate()` 步驟。只有最後的周遊資料會傳回。請注意，這不適用於 GLV ByteCode 查詢提交。

工作階段

海王星中的工作階段僅限於 10 分鐘的持續時間。如需詳細基於小鬼腳本的會話 (p. 326)資訊，請參閱〈TinkerPop工作階段參考〉。

交易

海王星在每個 Gemlin 遍歷的開始打開一個新的事務，並在遍歷的成功完成關閉事務。發生錯誤時交易將還原。

以分號 (`;`) 或換行符號字元 (\n) 分隔的多重陳述式包含在單一交易內。除了最後一個以外的每個陳述式結尾必須為要執行的 `next()` 步驟。只有最後的周遊資料會傳回。

使用 `tx.commit()` 和 `tx.rollback()` 的手動交易邏輯不受支援。

> **Important**
>
> 這僅適用於您以文本字符串形式發送 Gramlin 查詢的方法（請參閱小鬼交易 (p. 328)）。

頂點和邊 ID

海王星小鬼頂點和邊緣 ID 必須是類型的。`String`如果您新增頂點或邊緣時未提供 ID，將產生 UUID 並轉換為字串；例如，`"48af8178-50ce-971a-fc41-8c9a954cea62"`。這些識別碼字串支援 Unicode 字元，且大小不能超過 55 MB。

> **Note**
>
> 這表示使用者提供的 ID 受到支援，但他們在正常使用狀況下為選用。但是，海王星`Load`命令要求使用海王星 CSV 格式的 ~id 欄位來指定所有 ID。

使用者提供的 ID

海王星 Grimlin 允許使用者提供的 ID，且符合下列規定。

- 提供的 ID 是選用的。
- 僅支援頂點和邊緣。
- 僅支援 `String` 類型。

若要建立使用自訂 ID 的新頂點，請使用 `property` 步驟搭配 `id` 關鍵字：`g.addV().property(id, 'customid')`。

> **Note**
>
> 不要將引號放在 `id` 關鍵字旁邊。它指的是`T.id`.

所有的頂點 ID 必須各不相同，且所有的邊緣 ID 必須各不相同。但是，海王星確實允許頂點和邊緣具有相同的 ID。

如果您嘗試使用 `g.addV()` 建立新頂點，而使用該 ID 的頂點已存在，則操作會失敗。例外狀況為，如果您為頂點指定新的標籤，操作即會成功，但會將新標籤及指定的任何其他屬性新增至現有的頂點。不會覆寫任何項目。此舉不會建立新的頂點。頂點 ID 將維持不變，仍保有其唯一性。

例如，以下的 Gremlin 主控台命令會成功：

```
gremlin> g.addV('label1').property(id, 'customid')
gremlin> g.addV('label2').property(id, 'customid')
gremlin> g.V('customid').label()
==>label1::label2
```

頂點屬性 ID

Vertex 屬性 ID 將自動產生，且查詢時可顯示為正數或負數。

頂點屬性的基數

海王星支持設置基數和單基數。如未指定，則選取設定基數。這表示，如果您設定屬性值，它將新增新的值至屬性，但前提是其不能出現在值組內。此為 Set 的 Gremlin 列舉值。

不支援 `List`。如需有關屬性基數的詳細資訊，請參閱 Grimlin 中的頂點主題。JavaDoc

更新頂點屬性

若要更新屬性值，而不新增額外的值給值組，請在 `property` 步驟中指定 `single` 基數。

```
g.V('exampleid01').property(single, 'age', 25)
```

這會移除屬性所有現有的值。

標籤

海王星支持一個頂點的多個標籤。當您建立標籤時，您可以指定多重標籤並用 `::` 分隔。例如，`g.addV("Label1::Label2::Label3")` 將新增有三種不同標籤的頂點。`hasLabel` 步驟將比對此頂點和這三個標籤：`hasLabel("Label1")`、`hasLabel("Label2")` 和 `hasLabel("Label3")`。

> **Important**
>
> `::` 分隔符號僅針對本用途保留。您不能在 `hasLabel` 步驟中指定多重標籤。例如，`hasLabel("Label1::Label2")` 不符合任何內容。

跳脫字元

海王星解析所有轉義字符，如 Apache Groovy 語言文檔的轉義特殊字符部分中所述。

常規局限性

海王星不支持不開頭的 Groovy 命令g。這包括數學 (例如 1+1)、系統呼叫 (例如 `System.nanoTime()`) 和變數定義 (例如 1+1)。

> **Important**
>
> 海王星不支援完全限定的類別名稱。例如，您必須在 Groovy 請求中使用 `single`，而不是 `org.apache.tinkerpop.gremlin.structure.VertexProperty.Cardinality.single`。

序列化

海王星支持基於請求的 MIME 類型以下序列化。

MIME type	Serialization	Configuration
`application/vnd.gremlin-v1.0+gryo`	`GryoMessageSerializerV1D0`	`#######:[#. #####. TinkerIoRegistry1 # 0]`
`application/vnd.gremlin-v1.0+gryo-stringd`	`GryoMessageSerializerV1D0`	`serializeResultTo#####}`
`application/vnd.gremlin-v3.0+gryo`	`GryoMessageSerializerV3D0`	`#######:[#. #####. TinkerIoRegistry0 #`
`application/vnd.gremlin-v3.0+gryo-stringd`	`GryoMessageSerializerV3D0`	`serializeResultTo#####`
`application/vnd.gremlin-v1.0+json`	`### V MessageSerializerGremlin 1 # 0`	`#######:[#. #####. TinkerIoRegistry1 # 0]`
`application/vnd.gremlin-v2.0+json`	`### V2D0 MessageSerializer` (only works with WebSockets)	`#######:[#. #####. TinkerIoRegistry0 #]`
`application/vnd.gremlin-v3.0+json`	`GraphSONMessageSerializerV3d0`	
`application/json`	`### MessageSerializer`	`#######:[#. #####. TinkerIoRegistry0 #`
`application/vnd.graphbinary-v1.0`	`GraphBinaryMessageSerializerV1`	

拉姆達步驟

海王星不支援 Lambda 步驟。

不支援的小鬼方法

海王星不支持以下小鬼方法：

- org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversal.program(org.apac
- org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversal.sideEffect(java.
- org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversal.from(org.apache.
- org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversal.to(org.apache.ti

例如，不允許以下周
遊：g.V().addE('something').from(__.V().next()).to(__.V().next())。

> **Important**
>
> 這「只」適用於以「文字字串」傳送 Gremlin 查詢的方法。

不支援的小鬼步驟

海王星不支持以下小鬼步驟：

- 怪靈 io () 步驟僅在海王星部分支持。它可以在讀取上下文中使用，如在g.io(*url*).read()，但不能寫入。

海王星中的小精靈圖特徵

怪靈的海王星實現不會暴露該graph對象。下表列出了 Grimlin 功能，並指出海王星是否支持它們。

海王星對graph功能的支援

Neptune 圖形功能 (在支援的情況下) 與graph.features()指令傳回的相同。

圖表功能	已啟用？
##	true
ThreadedTransactions	false
Computer	false
Persistence	true
ConcurrentAccess	true

可變功能的海王星支援

可變特徵	已啟用？
Variables	false
SerializableValues	false
UniformListValues	false
BooleanArrayValues	false
DoubleArrayValues	false
IntegerArrayValues	false

StringArrayValues	false
BooleanValues	false
ByteValues	false
DoubleValues	false
FloatValues	false
IntegerValues	false
LongValues	false
MapValues	false
MixedListValues	false
StringValues	false
ByteArrayValues	false
FloatArrayValues	false
LongArrayValues	false

對頂點特徵的海王星支援

頂點特徵	已啟用？
MetaProperties	false
DuplicateMultiProperties	false
AddVertices	true
RemoveVertices	true
MultiProperties	true
UserSuppliedIds	true
AddProperty	true
RemoveProperty	true
NumericIds	false
StringIds	true
UuidIds	false
CustomIds	false
AnyIds	false

對頂點屬性功能的海王星支援

頂點屬性特徵	已啟用？

UserSuppliedIds	false
AddProperty	true
RemoveProperty	true
NumericIds	true
StringIds	true
UuidIds	false
CustomIds	false
AnyIds	false
##	true
SerializableValues	false
UniformListValues	false
BooleanArrayValues	false
DoubleArrayValues	false
IntegerArrayValues	false
StringArrayValues	false
BooleanValues	true
ByteValues	true
DoubleValues	true
FloatValues	true
IntegerValues	true
LongValues	true
MapValues	false
MixedListValues	false
StringValues	true
ByteArrayValues	false
FloatArrayValues	false
LongArrayValues	false

對邊緣功能的海王星支援

邊特徵	已啟用？
AddEdges	true
RemoveEdges	true

UserSuppliedIds	true
AddProperty	true
RemoveProperty	true
NumericIds	false
StringIds	true
UuidIds	false
CustomIds	false
AnyIds	false

邊緣屬性功能的海王星支援

邊屬性特徵	已啟用？
##	true
SerializableValues	false
UniformListValues	false
BooleanArrayValues	false
DoubleArrayValues	false
IntegerArrayValues	false
StringArrayValues	false
BooleanValues	true
ByteValues	true
DoubleValues	true
FloatValues	true
IntegerValues	true
LongValues	true
MapValues	false
MixedListValues	false
StringValues	true
ByteArrayValues	false
FloatArrayValues	false
LongArrayValues	false

亞馬遜海王星的 SPARQL 標準合規性

列出適用的 SPARQL 標準之後，以下各節提供有關 Neptune 的 SPARQL 實作如何從這些標準延伸或偏離的特定詳細資料。

主題

- SPARQL 的適用標準 (p. 92)
- 海王星 SPARQL 中的預設命名空間前綴 (p. 92)
- SPARQL 預設圖形和具名圖形 (p. 92)
- 海王星所支援的 SPARQL XPath 建構函式 (p. 93)
- 查詢和更新的預設基本 IRI (p. 93)
- XSD: 海王星中的日期時間值 (p. 94)
- 海王星處理特殊浮點值 (p. 94)
- 海王星任意長度值的限制 (p. 94)
- 海王星在 SPARQL 中擴展等於比較 (p. 94)
- 在海王星 SPARQL 中處理超出範圍的文字 (p. 95)

在實作 SPARQL 圖形查詢語言時，亞馬遜海王星符合下列標準。

SPARQL 的適用標準

- SPARQL 是依 2013 年 3 月 21 日的 W3C SPARQL 1.1 查詢語言建議定義。
- SPARQL 更新協定和查詢語言是依 W3C SPARQL 1.1 更新規格定義。
- 對於數字格式，SPARQL 遵循 W3C XML 結構描述定義語言（XSD）1.1 第二部分：數據類型規範，該規範與 IEEE 754 規範（IEEE 754-2019-用於浮點運算的 IEEE 標準）一致。 有關更多信息，另請參閱維基百科 IEEE 754 頁面）。不過，此規格不包含於 `IEEE 754-1985` 版本後引入的功能。

海王星 SPARQL 中的預設命名空間前綴

海王星定義默認情況下在 SPARQL 查詢中使用以下前綴。如需詳細資訊，請參閱 SPARQL 規格中的 Prefixed Names。

- `rdf` – `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
- `rdfs` – `http://www.w3.org/2000/01/rdf-schema#`
- `owl` – `http://www.w3.org/2002/07/owl#`
- `xsd` – `http://www.w3.org/2001/XMLSchema#`

SPARQL 預設圖形和具名圖形

亞馬遜海王星將每三個與一個命名的圖形相關聯 預設圖形是定義成所有具名圖形的聯集。

查詢的預設圖形

若您提交 SPARQL 查詢而未透過 `GRAPH` 關鍵字或 `FROM NAMED` 之類的建構式明確指定圖形，Neptune 一律會考慮資料庫執行個體中的所有三元組。例如，下列查詢會傳回來自海王星 SPARQL 端點的所有三元組：

```
SELECT * WHERE { ?s ?p ?o }
```

顯現於多個圖形中的三元組將僅傳回一次。

如需預設圖形規格的相關資訊，請參閱 SPARQL 1.1 查詢語言規格的 RDF 資料集一節。

指定具名圖形以進行載入、插入或更新

如果您在載入、插入或更新三元組時未指定具名圖形，Neptune 會使用 URI 定義的後援命名圖形。`http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`

當您使用三元格式發出 Neptune Load 要求時，您可以使用參數指定要用於所有三元組的具名圖形。`parserConfiguration: namedGraphUri`如需 Load 命令語法的相關資訊，請參閱 the section called “載入器命令” (p. 275)。

> **Important**
>
> 若您既沒有使用此參數也未指定具名圖形，則將使用備用 URI：`http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`。

如果透過 SPARQL UPDATE 載入三元組而未明確提供具名圖形目標，則亦將使用上述的備用具名圖形。

您可以使用四元組格式的 N-Quads，為資料庫中的每個三元組指定具名圖形。

> **Note**
>
> 使用 N-Quads 允許您將具名圖形留空不填。在此情況下即會使用 `http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`。
> 您可以使用 `namedGraphUri` 剖析器組態選項，覆寫 N-Quads 的預設具名圖形。

海王星所支援的 SPARQL XPath 建構函式

SPARQL 標準允許 SPARQL 引擎支援一組可擴充的 XPath 建構函式。Neptune 目前支持以下構造函數，其中前`xsd`綴被定義為`http://www.w3.org/2001/XMLSchema#`：

- `xsd:boolean`
- `xsd:integer`
- `xsd:double`
- `xsd:float`
- `xsd:decimal`
- `xsd:long`
- `xsd:unsignedLong`

查詢和更新的預設基本 IRI

由於 Neptune 叢集具有數個不同的端點，因此在解析相對 IRI 時，使用查詢或更新的要求 URL 做為基礎 IRI 可能會導致未預期的結果。

從發動機版本 1.2.1.0 (p. 861) 開始，如果明確的基礎 IRI 不是請求的一部分，海王星使用`http://aws.amazon.com/neptune/default/`作為基礎 IRI。

在以下請求中，基本 IRI 是請求的一部分：

```
BASE <http://example.org/default/>
INSERT DATA { <node1> <id> "n1" }

BASE <http://example.org/default/>
SELECT * { <node1> ?p ?o }
```

結果將是：

```
?p                                                  ?o
http://example.org/default/id                       n1
```

但是，在此要求中，不包括基本 IRI：

```
INSERT DATA { <node1> <id> "n1" }

SELECT * { <node1> ?p ?o }
```

在這種情況下，結果將是：

```
?p                                                  ?o
http://aws.amazon.com/neptune/default/id            n1
```

XSD: 海王星中的日期時間值

出於性能原因，Neptune 始終將日期/時間值存儲為協調世界時（UTC）。這能讓直接比較很有效率。

這也表示，如果您輸入指定特定時區的`dateTime`值，Neptune 會將值轉譯為 UTC，並捨棄該時區資訊。接下來，當您收到 `dateTime` 後，該執會以 UTC 呈現而非原始時區的時間，而且您也無法再判斷原始時區。

海王星處理特殊浮點值

海王星處理 SPARQL 中的特殊浮點值，如下所示。

海王星中的 NaN 處理

在海王星中，SPARQL 可以接受查詢`NaN`中的值。信號和靜音`NaN`值之間沒有區別。海王星將所有`NaN`值視為安靜。

在語意上，不可能進行 NaN 的比較，因為沒有任何值大於、小於或等於 NaN。這表示理論上，比較一端的 NaN 值和另一端的「任何內容」永遠不相符。

但是，XSD 規範確實將兩個`xsd:double`或`xsd:float`NaN值視為相等。Neptune 遵循此`IN`過濾器，過濾器表達式中的等於運算符以及完全匹配語義（`NaN`在三重模式的對象位置具有 a）。

海王星中的 SPARQL 無限值處理

在海王星中，SPARQL 可以接受`INF`或查詢`-INF`中的值。 `INF`比較大於任何其他數值，並`-INF`比較小於任何其他數值。

比較具有相符符號的兩個 INF 值時，無論其類型如何皆彼此相等 (例如，float `-INF` 等於 double `-INF`）。

當然，因為沒有任何值大於、小於或等於 NaN，所以不可能和 NaN 比較。

海王星中的 SPARQL 負零處理

海王星將負零值標準化為無符號零。負零值可用於查詢中，但不會原樣記錄在資料庫中，而且不等於不帶正負號的零。

海王星任意長度值的限制

海王星將 SPARQL 中 XSD 整數、浮點數和十進位值的儲存大小限制為 64 位元。使用較大的值會導致錯`InvalidNumericDataException`誤。

海王星在 SPARQL 中擴展等於比較

SPARQL 標準為值運算式定義了三元邏輯，其中，值運算式可評估為 `true`、`false` 或 `error`。SPARQL 1.1 規格中定義的術語相等的預設語意) 適用於條件中`=`並在`FILTER`條件中`!=`進行比較`error`時，會在比較規格中運算子資料表中未明確比較的資料類型時產生一個。

這種行為可能會導致不直觀的結果，如下列範例所示。

資料：

```
<http://example.com/Server/1> <http://example.com/ip> "127.0.0.1"^^<http://example.com/
datatype/IPAddress>
```

查詢 1：

```
SELECT * WHERE {
    <http://example.com/Server/1> <http://example.com/ip> ?o .
    FILTER(?o = "127.0.0.2"^^<http://example.com/datatype/IPAddress>)
}
```

查詢 2：

```
SELECT * WHERE {
    <http://example.com/Server/1> <http://example.com/ip> ?o .
    FILTER(?o != "127.0.0.2"^^<http://example.com/datatype/IPAddress>)
}
```

使用海王星在 1.0.2.1 版之前使用的預設 SPARQL 語意，這兩個查詢都會傳回空白結果。原因是 `?o = "127.0.0.2"^^<http://example.com/IPAddress>` 在評估 `?o := "127.0.0.1"^^<http://example.com/IPAddress>` 時會產生 `error`，而不是 `false`，因為沒有為自訂資料類型 `<http://example.com/IPAddress>` 指定明確的比較規則。因此，第二個查詢中的否定版本也會產生 `error`。在這兩個查詢中，`error` 會導致篩選出候選解決方案。

從 1.0.2.1 版開始，海王星已根據規範擴充 SPARQL 不等式運算子。請參閱 SPARQL 第 1.1 節，運算子可擴充性，讓引擎定義如何跨使用者定義和不可比較內建資料類型進行比較的其他相關規則。

使用此選項，Neptune 現在會將未在運算子對應表中明確定義的任何兩種資料類型的比較，視為評估常值和資料類型在語法上是`true`否相等，否則為 false。任何情況下都不會發生 `error`。

使用這些新的語意，第二個查詢就會傳回 `"127.0.0.1"^^<http://example.com/IPAddress>`，而不是空的結果。

在海王星 SPARQL 中處理超出範圍的文字

XSD 語義會定義每個數值類型的值空間，但 `integer` 和 `decimal` 除外。這些定義會將每種類型限制在某個範圍的值內。例如，範圍 `xsd:byte` 的範圍為 -128 到 +127 (頭尾皆含)。任何超出此範圍的值都視為無效。

如果您嘗試在類型的值空間之外指派常值 (例如，如果您嘗試將文字值設定`xsd:byte`為 999)，Neptune 會以原樣接受該out-of-range值，而不會四捨五入或截斷它。但因為指定的類型不能代表此值，所以不會保存為數值。

也就是說，Neptune 會接受，`"999"^^xsd:byte`即使它是定義的值範圍之外的`xsd:byte`值。但是，在資料庫中保存該值之後，此值只能用於三重模式物件位置的完全相符語義。沒有範圍篩選可以在其上執行，因為out-of-range常值不會被視為數值。

該 SPARQL 1.1 規範定義的形式 `numeric-運`算符-，-運算*`string`*符 *`literal-numeric string`*，等等的範圍運算符。`literal`海王星不能像 `invalid-literal` -operator-*`numeric-value`* 這樣執行範圍比較運算符。

亞馬遜海王星中的開放密碼標準合規性

OpenCypher 的亞馬遜海王星發行通常支援目前 OpenCypher 規格 (也就是密碼查詢語言參考版本 9) 中定義的子句、運算子、運算式、函數和語法。海王星對 OpenCypher 的支持的局限性和差異如下所示。

Note

密碼的當前 Neo4j 的實現包含不包含在上面提到的 OpenCypher 規範的功能。如果您要將目前的密碼移轉至 Neptune，請參閱Neptune 與 Neo4j 的兼容性 (p. 239)和以重寫密碼查詢以在 Neptune 上的 OpenCpher 中運行 (p. 242)取得更多資訊。

支持海王星中的開放密碼子句

Neptune 支援下列子句，除非另有說明：

- MATCH— 支援，但目*allShortestPaths()*前不支援*shortestPath()*和。
- OPTIONAL MATCH
- *MANDATORY MATCH*— 目前在海王星中不支援。但是，海王星確實支持MATCH查詢中的自定義 ID 值 (p. 102)。
- RETURN— 支援，與SKIP或LIMIT的非靜態值搭配使用時除外。例如，以下內容目前不起作用：

```
MATCH (n)
RETURN n LIMIT toInteger(rand())    // Does NOT work!
```

- WITH— 支援，與SKIP或LIMIT的非靜態值搭配使用時除外。例如，以下內容目前不起作用：

```
MATCH (n)
WITH n SKIP toInteger(rand())
WITH count() AS count
RETURN count > 0 AS nonEmpty    // Does NOT work!
```

- UNWIND
- WHERE
- ORDER BY
- SKIP
- LIMIT
- CREATE— 海王星可讓您在CREATE查詢中建立自訂 ID 值 (p. 102)。
- DELETE
- SET
- REMOVE
- MERGE— 海王星支援MERGE查詢中的自訂 ID 值 (p. 102)。
- *CALL[YIELD...]*— 目前在海王星中不支援。
- UNION, UNION ALL— 支援唯讀查詢，但目前不支援突變查詢。

支持海王星中的開放密碼運算符

Neptune 支援下列運算子，除非另有說明：

一般運算子

- DISTINCT
- 用於存取巢狀文字對應屬性的.運算子。

數學運算子

- +加法運算子。

- **-**減法運算子。
- ***乘法運算子。
- 除**/**法運算子。
- **%**模除法運算符。
- 指**^**數運算子。

比較運算子

- **=**加法運算子。
- **<>**不等式運算子。
- 除非其中一個引數是「路徑」、「清單」或「對映」，否則支援**<**小於運算子。
- 除非其中一**>**個引數是「路徑」、「清單」或「對映」，否則支援大於運算子。
- 除非其中一個引數是「路徑」、「清單」或「對映」，否則支援 **<=** less-than-or-equal-to 運算子。
- 除非其中一個引數是「路徑」、「清單」或「對映」，否則支援 **>=** greater-than-or-equal-to 運算子。
- `IS NULL`
- `IS NOT NULL`
- `STARTS WITH`如果要搜尋的資料是字串，則支援此選項。
- `ENDS WITH`如果要搜尋的資料是字串，則支援此選項。
- `CONTAINS`如果要搜尋的資料是字串，則支援此選項。

布林值運算子

- `AND`
- `OR`
- `XOR`
- `NOT`

字串運算子

- **+**串連運算子。

列表運算符

- **+**串連運算子。
- `IN`（檢查列表中的項目是否存在）

支持海王星中的開放密碼表達式

Neptune 支援下列運算式，除非另有說明：

- `CASE`
- Neptune 目前不支援此**[]**運算式，用於存取節點、關係或對映中的動態計算屬性索引鍵。例如，以下內容不起作用：

```
MATCH (n)
WITH [5, n, {key: 'value'}] AS list
RETURN list[1].name
```

支持海王星中的開放密碼函數

Neptune 支援下列功能，除非另有說明：

謂詞函數

- exists()

標量函數

- coalesce()
- endNode()
- epochmillis()
- head()
- id()
- last()
- length()
- randomUUID()
- properties()
- removeKeyFromMap
- size()-此重載方法目前僅適用於模式表達式，列表和字符串
- startNode()
- timestamp()
- toBoolean()
- toFloat()
- toInteger()
- type()

聚合函數

- avg()
- collect()
- count()
- max()
- min()
- percentileDisc()
- stDev()
- *percentileCont()*— 目前在海王星中不支援。
- *stDevP()*— 目前在海王星中不支援。
- sum()

列出函數

- join() (p. 101)（將列表中的字符串連接為單個字符串）
- keys()
- labels()
- nodes()

- range()
- relationships()
- reverse()
- tail()

數學函數 — 數字

- abs()
- ceil()
- floor()
- rand()
- round()
- sign()

數學函數 — 對數

- e()
- exp()
- log()
- log10()
- sqrt()

數學函數 — 三角函數

- acos()
- asin()
- atan()
- atan2()
- cos()
- cot()
- degrees()
- pi()
- radians()
- sin()
- tan()

字串函數

- join( ) (p. 101) (將列表中的字符串連接為單個字符串)
- left()
- lTrim()
- replace()
- reverse()
- right()
- rTrim()
- split()
- substring()

- toLower()
- toString()
- toUpper()
- trim()

使用者定義

Neptune 目前不支援#######。

海王星專用開放密碼實施細節

以下各節描述了 OpenCypher 的海王星實現可能不同於或超出 OpenCypher 規範的方式。

海王星中的可變長度路徑（VLP）評估

變數長度 path (VLP) 評估會探索圖形中節點之間的路徑。查詢中的路徑長度可以不受限制。為了防止循環，OpenCypher 規範指定每個邊緣必須每個解決方案最多穿越一次。

對於 VLP，海王星實作偏離 OpenCypher 規格，因為它只支援屬性相等篩選器的常數值。採取以下查詢：

```
MATCH (x)-[:route*1..2 {dist:33, code:x.name}]->(y) return x,y
```

由於x.name屬性相等篩選器值不是常數，因此此查詢會產生UnsupportedOperationException以下訊息：Property predicate over variable-length relationships with non-constant expression is not supported in this release.

海王星开放密码实施中的时间支持

海王星目前為 OpenCypher 中的時間函數提供了有限的支持。它支持時間類型的DateTime數據類型。

該datetime()函數可用於獲取當前的 UTC 日期和時間，如下所示：

```
RETURN  datetime() as res
```

日期和時間值可以從存儲在 Neptune 中的數據進行轉換，如下所示：

```
MATCH (n) RETURN datetime(n.createdDate)
```

日期和時間值可以從日"期時T間"格式的字符串中解析，其中日期和時間均以下面支持的形式之一表示：

支援的日期格式

- yyyy-MM-dd
- yyyyMMdd
- yyyy-MM
- yyyy-DDD
- yyyyDDD
- yyyy

支援的時間格式

- HH:mm:ssZ

- HHmmssZ
- HH:mm:ssZ
- HH:mmZ
- HHmmZ
- HHZ
- HHmmss
- HH:mm:ss
- HH:mm
- HHmm
- HH

例如：

```
RETURN datetime('2022-01-01T00:01')      // or another example:
RETURN datetime('2022T0001')
```

請注意，海王星 OpenCypher 中的所有日期/時間值都以 UTC 值的形式存儲和檢索。

海王星 OpenCypher 使用時statement鐘，這意味著在整個查詢的持續時間內使用相同的時間瞬間。同一交易中的不同查詢可能會使用不同的時間瞬間。

海王星不支持在調用中使用函數datetime()。例如，以下內容將不起作用：

```
CREATE (:n {date:datetime(tostring(2021))})  // ---> NOT ALLOWED!
```

海王星確實支持將 a 轉換為的epochmillis()功datetime能epochmillis。例如：

```
MATCH (n) RETURN epochMillis(n.someDateTime)
1698972364782
```

Neptune 目前不支持DateTime對象的其他功能和操作，例如加法和減法。

海王星開放密碼語言語義的差異

海王星表示節點和關係 ID 作為字符串，而不是整數。ID 等於透過資料載入器提供的 ID。如果該列有命名空間，則命名空間加上 ID。因此，該id函數返回一個字符串而不是整數。

資INTEGER料類型限制為 64 位元。使用TOINTEGER函數將較大的浮點數值或字串值轉換為整數時，負值會被截斷為，LLONG_MIN而正值會截斷為。LLONG_MAX

例如：

```
RETURN TOINTEGER(2^100)
>  9223372036854775807

RETURN TOINTEGER(-1 * 2^100)
>  -9223372036854775808
```

海王星join()特定功能

海王星實現了一個不存在於開放密碼規範中的join()函數。它從字符串文字和字符串分隔符的列表中創建一個字符串文字。它有兩個參數：

- 第一個參數是字符串文字列表。
- 第二個參數是分隔符串，它可以由零個，一個或多個字符組成。

範例：

```
join(["abc", "def", "ghi"], ", ")    // Returns "abc, def, ghi"
```

海王星removeKeyFromMap()特定功能

海王星實現了一個不存在於開放密碼規範中的removeKeyFromMap()函數。它從地圖中刪除指定的鍵，並返回生成的新映射。

該函數有兩個參數：

- 第一個參數是從中刪除鍵的映射。
- 第二個參數是從地圖中刪除的關鍵。

在您想要透過展開對應清單來設定節點或關係值的情況下，此removeKeyFromMap()函數特別有用。例如：

```
UNWIND [{`~id`: 'id1', name: 'john'}, {`~id`: 'id2', name: 'jim'}] as val
CREATE (n {`~id`: val.`~id`})
SET n = removeKeyFromMap(val, '~id')
```

節點和關係屬性的自訂 ID 值

從引擎版本 1.2.0.2 (p. 870) 開始，海王星已擴充 OpenCypher 規格，因此您現在可以在CREATE、MERGE和子句中指定節點和關係的id值。MATCH這使您可以分配用戶友好的字符串，而不是系統生成的 UUID 來識別節點和關係。

> Warning
>
> OpenCypher 規範的這個延伸模組是向後不相容的，因為現在~id被視為保留的屬性名稱。如果您已在資料和查詢中~id作為屬性使用，則需要將現有屬性移轉至新的屬性金鑰，並移除舊的屬性金鑰。請參閱 如果您目前使用~id作為物業，該怎麼辦 (p. 103)。

以下是示範如何建立具有自訂 IDS 的節點和關聯性的範例：

```
CREATE (n {`~id`: 'fromNode', name: 'john'})
  -[:knows {`~id`: 'john-knows->jim', since: 2020}]
  ->(m {`~id`: 'toNode', name: 'jim'})
```

如果您嘗試建立已在使用中的自訂 ID，Neptune 會擲回DuplicateDataException錯誤。

以下是在MATCH子句中使用自訂 ID 的範例：

```
MATCH (n {`~id`: 'id1'})
RETURN n
```

以下是在MERGE子句中使用自訂 ID 的範例：

```
MATCH (n {name: 'john'}), (m {name: 'jim'})
MERGE (n)-[r {`~id`: 'john->jim'}]->(m)
RETURN r
```

如果您目前使用~id作為物業，該怎麼辦

隨著引擎版本 1.2.0.2 (p. 870)，OpenCypher 子句中的密~id鑰現在被視為id而不是作為一個屬性。這意味著，如果你有一個名為的屬性~id，訪問它變得不可能。

如果您使用的是~id屬性，則在升級到引擎版本1.2.0.2或更高版本之前必須執行的操作是首先將現有~id屬性遷移到新的屬性密鑰，然後刪除該~id屬性。例如，下面的查詢：

- 為所有節點創建一個名為「NewID」的新屬性，
- 將 '~id' 屬性的值複製到 'NewID' 屬性中，
- 並從數據中刪除 '~id' 屬性

```
MATCH (n)
WHERE exists(n.`~id`)
SET n.newId = n.`~id`
REMOVE n.`~id`
```

對於具有~id屬性的數據中的任何關係，也需要執行相同的操作。

您還必須更改您使用該~id屬性引用的任何查詢。例如，這個查詢：

```
MATCH (n)
WHERE n.`~id` = 'some-value'
RETURN n
```

... 會改變為這個：

```
MATCH (n)
WHERE n.newId = 'some-value'
RETURN n
```

海王星開放密碼和密碼之間的其他區別

- 海王星僅支援 Bolt 通訊協定的 TCP 連線。WebSockets不支援螺栓的連接對。
- 海王星開放密碼刪除由 Unicode 中定義的空格trim()，ltrim()和函數。rtrim()
- 在海王星 OpenCypher 中，tostring(雙)不會自動切換到 E 表示法對於雙倍的大值。
- 雖然 OpenCypher 創建不會創建多值屬性，但它們可以存在於使用 Grimlin 創建的數據中。如果海王星 OpenCypher 遇到一個多值屬性，其中一個值被任意選擇，從而創建一個非確定性的結果。

海王星圖數據模型

Amazon 海王星圖表資料的基本單位是四位置 (四元) 元素，類似於資源描述框架 (RDF) 四元。以下是海王星四邊形的四個位置：

- `subject   (S)`
- `predicate (P)`
- `object    (O)`
- `graph     (G)`

每個 quad 是一個陳述式，可做出關於一個或多個資源的宣告。陳述式可以宣告兩個資源之間的關係，或者將屬性 (鍵/值組) 連接到資源。您一般可以將 quad 述詞值視為陳述式的動詞。它會描述正在定義的關係或屬性的類型。物件是關係的目標，或是屬性的值。範例如下：

- 可透過將來源頂點識別符存放在 S 位置中、將目標頂點識別符存放在 O 位置中，以及將邊緣標籤存放在 P 位置，來表示兩個頂點之間的關係。
- 可透過將元素識別符存放在 S 位置中、將屬性索引鍵存放在 P 位置中，以及將屬性值存放在 O 位置中，來表示屬性。

圖形位置 G 在不同堆疊中的使用方式不同。對於海王星中的 RDF 數據，該G位置包含一個命名的圖形標識符。對於 Gremlin 中的屬性圖，它用於在邊緣的情況下存放邊緣 ID 值。在所有其他情況下，它會預設為固定的值。

具有共用資源識別符的一組 quad 陳述式可建立圖形。

用戶面向值的字典

Neptune 不會將大多數面向使用者的值直接儲存在其維護的各種索引中。相反，它將它們分別存儲在字典中，並用 8 字節標識符在索引中替換它們。

- 將進入S，P或G索引的所有面向用戶的值都以這種方式存儲在字典中。
- 在索O引中，數值直接存儲在索引中（內聯）。這包括`date`和`datetime`值（以來自紀元的毫秒表示）。
- O索引中的所有其他面向用戶的值都存儲在字典中，並以 ID 表示在索引中。

字典包含使用者面向值的正向對應至索引中的 8 位元組 ID。`value_to_id`

它將 8 字節 ID 的反向映射存儲在兩個索引之一中的值，具體取決於值的大小：

- `id_to_value`索引會將 ID 對應至內部編碼後小於 767 個位元組的使用者面向值。
- `id_to_blob`索引會將 ID 對應至較大的使用者面向值。

如何在海王星中索引陳述式

當您查詢 quad 圖表，您可以對每個 quad 指定值限制，也可以不指定。查詢會傳回符合您指定之值限制的所有 quad。

海王星使用索引來解析查詢。Andreas Harth 和 Stefan Decker 在其 2005 年的文章 Optimized Index Structures for Querying RDF from the Web 中指出，四 quad 位置有 16 個 (2^4) 可能的存取模式。您可以有效率地查詢所有 16 個模式，而無需使用六個 quad 陳述式索引進行掃描和篩選。每個 quad 陳述式索引使用由以不同順序串連的四個位置值組成的索引鍵。

```
     Access Pattern                                          Index key order
-----------------------------------------------------    ---------------
 1.  ????  (No constraints; returns every quad)              SPOG
 2.  SPOG  (Every position is constrained)                   SPOG
 3.  SPO?  (S, P, and O are constrained; G is not)           SPOG
 4.  SP??  (S and P are constrained; O and G are not)        SPOG
 5.  S???  (S is constrained; P, O, and G are not)           SPOG
 6.  S??G  (S and G are constrained; P and O are not)        SPOG

 7.  ?POG  (P, O, and G are constrained; S is not)           POGS
 8.  ?PO?  (P and O are constrained; S and G are not)        POGS
 9.  ?P??  (P is constrained; S, O, and G are not)           POGS

10.  ?P?G  (P and G are constrained; S and O are not)        GPSO
11.  SP?G  (S, P, and G are constrained; O is not)           GPSO
12.  ???G  (G is constrained; S, P, and O are not)           GPSO

13.  S?OG  (S, O, and G are constrained; P is not)           OGSP
14.  ??OG  (O and G are constrained; S and P are not)        OGSP
15.  ??O?  (O is constrained; S, P, and G are not)           OGSP
16.  S?O?  (S and O are constrained; P and G are not)        OSGP
```

Neptune 默認情況下只創建和維護這六個索引中的三個：

- SPOG – 使用 Subject + Predicate + Object + Graph 組成的索引鍵。
- POGS – 使用 Predicate + Object + Graph + Subject 組成的索引鍵。
- GPSO – 使用 Graph + Predicate + Subject + Object 組成的索引鍵。

這三個索引會處理許多最常見的存取模式。只維護三個完整陳述式索引 (而非六個)，可大幅減少支援不經掃描和篩選而快速存取所需的資源。例如，每當頂點或頂點和屬性識別符等位置字首繫結時，SPOG 索引會允許有效查詢。只有存放於 P 位置的邊緣或屬性標籤繫結時，POGS 索引才會允許有效存取。

尋找陳述式的低層級 API 採用的陳述式模式，其中有些位置已知，而其餘則保留供索引搜尋探索。Neptune 會根據其中一個陳述式索引的索引鍵順序將已知位置組成索引鍵前置詞，以便執行範圍掃描，以擷取符合已知位置的所有陳述式。

不過，Neptune 預設不會建立的其中一個陳述式索引是反向遍歷索OSGP引，它可以跨物件和主旨收集述詞。相反，Neptune 預設會在用來執行聯集掃描的個別索引中追蹤不同的{all P x POGS}述詞。使用 Gremlin 時，述詞對應到屬性或邊緣標籤。

如果圖表中不同述詞的數量變大，預設的 Neptune 存取策略可能會變得效率低下。例如，在 Gremlin 中，未指定邊緣標籤的 in() 步驟，或內部使用 in() 的任何步驟 (例如 both() 或 drop()) 可能會變得沒有效率。

使用實驗室模式啟用 OSGP 索引建立

如果您的資料模型建立了大量不同的述詞，除了 Neptune 預設維護的三個索引之外，使用實驗室模式啟用 OSGP (p. 128) 索引之外，您可能會遇到效能降低和更高的營運成本。

> **Note**
>
> 此功能從海王星引擎版本 1.0.1.0. (p. 985) 200463.0 開始提供。

啟用 OSGP 索引可能會有幾個缺點：

- 插入速率最多可能減慢 23%。
- 儲存量增加高達 20%。
- 均等觸及所有索引的讀取查詢 (這是很少見的情況) 可能會增加延遲。

不過，一般而言，啟用具有大量不同述詞之資料庫叢集的 OSGP 索引是值得的。基於對象的搜索變得非常高效（例如，查找到頂點的所有傳入邊緣或與給定對象連接的所有主題），因此刪除頂點也變得更有效率。

Important

您只能在空白資料庫叢集中啟用 OSGP 索引，然後再將任何資料載入其中。

海王星數據模型中的小鬼語句

Gremlin 屬性圖數據在 SPOG 模型中使用三類語句來表示，即：

- 頂點標籤陳述式 (p. 348)
- 邊緣陳述式 (p. 348)
- 屬性陳述式 (p. 348)

有關如何在 Gemlin 查詢中使用這些內容的說明，請參閱。了解 Gemlin 查詢如何在海王星中工作 (p. 347)

海王星查閱快取可加速讀取查詢

Amazon Neptune 實作查閱快取，該快取會使用R5d執行個體的 NVMe 型 SSD，針對經常重複的屬性值或 RDF 常值查詢來改善查詢的讀取效能。查閱快取會暫時將這些值儲存在 NVMe SSD 磁碟區中，以便快速存取這些值。

此功能可從開始使用亚马逊海王星发动机 (p. 940)。

如果需要從叢集儲存體磁碟區而非記憶體擷取屬性值或常值，則傳回大量頂點和邊緣 (或許多 RDF 三元組) 屬性的讀取查詢可能會有較高的延遲。範例包括長時間執行的讀取查詢，這些查詢會從身分圖傳回大量全名，或從詐騙偵測圖傳回 IP 位址。隨著查詢傳回的屬性值或 RDF 常值數目增加，可用的記憶體會減少，而且您的查詢執行可能會大幅降低。

海王星查詢快取的使用案例

查找緩存只有在您的讀取查詢返回非常大量的頂點和邊，或 RDF 三元的屬性時才有幫助。

為了最佳化查詢效能，Amazon Neptune 會使用R5d執行個體類型為這類屬性值或常值建立大型快取。然後，從快取擷取它們的速度會比從叢集儲存磁碟區擷取它們快得多。

根據經驗法則，只有在滿足下列三個條件的情況下，才能啟用查閱快取：

- 您一直在觀察讀取查詢的延遲增加。
- 執行讀取查詢時，您也會觀察`BufferCacheHitRatio`CloudWatch量度 (p. 775)下降 (請參閱使用 Amazon 監控海王星 CloudWatch (p. 772))。
- 在呈現結果之前，您的讀取查詢會花費大量時間實現傳回值 (請參閱下面的 gremLIN-profile 範例，瞭解決查詢具體化多少屬性值的方法)。

> **Note**
>
> 此功能僅在上述特定案例中有用。例如，查找緩存根本沒有幫助聚合查詢。除非您正在執行會受益於查閱快取的查詢，否則沒有理由使用R5d執行個體類型，而非同等且成本較低的R5執行個體類型。

如果您使用的是 Gremlin，您可以使用. 小精靈 API `profile` (p. 357) 在「索引操作」下，它顯示了執行過程中實現的術語數：

```
Index Operations
Query execution:
    # of statement index ops: 3
    # of unique statement index ops: 3
    Duplication ratio: 1.0
    # of terms materialized: 5273
Serialization:
    # of statement index ops: 200
    # of unique statement index ops: 140
    Duplication ratio: 1.43
    # of terms materialized: 32693
```

具體化的非數字術語數量與 Neptune 必須執行的術語查詢數量成正比。

使用查閱快取

查詢快取僅適用於R5d執行個體類型，依預設會自動啟用該執行個體類型。Neptune R5d 執行個體具有與R5執行個體相同的規格，加上高達 1.8 TB 的本機 NVMe SSD 儲存體。查閱快取是執行個體特定的，受

益的工作負載可以專門導向 Neptune 叢集中的R5d執行個體，而其他工作負載則可導向至其他執行個體類型R5或其他執行個體類型。

若要在 Neptune 執行個體上使用查詢快取，只要將該執行個體升級為執行個R5d體類型即可。當您這麼做時，Neptune 會自動將海王星查找緩存 (p. 532)資料庫叢集參數設定為'enabled'，並在該特定執行個體上建立查閱快取。然後，您可以使用 執行個體狀態 (p. 770) API 來確認快取已啟用。

同樣地，若要停用指定執行個體上的查閱快取，請將執行個體從執行個R5d體類型縮減為相等的R5執行個體類型。

啟動R5d執行個體時，會啟用查閱快取並在冷啟動模式下，表示它為空白。Neptune 首先在查詢快取中檢查屬性值或 RDF 常值，同時處理查詢，並加入它們 (如果尚未存在)。這逐漸熱身緩存。

當您將需要屬性值或 RDF 常值查閱的讀取查詢導向至 R5d 讀取器執行個體時，讀取效能會在快取預熱時稍微降低。但是，當快取預熱時，讀取效能會大幅加快，而且您也可能會看到與查閱快取而非叢集儲存體相關的 I/O 成本下降。記憶體使用率也會改善。

如果您的作家實例是一個R5d，它會在每次寫入操作時自動加熱其查找緩存。此方法確實會稍微增加寫入查詢的延遲，但會更有效率地加熱查閱快取。然後，如果您將需要屬性值或 RDF 常值查詢的讀取查詢導向寫入器執行個體，則會立即開始獲得改善的讀取效能，因為這些值已經在該處快取。

此外，如果您正在R5d寫入器執行個體上執行大量載入程式，您可能會注意到其效能因為快取而略有降低。

由於查閱快取特定於每個節點，因此主機取代會將快取重設為冷啟動。

您可以將 DB 叢集參數設定為，暫時停用資料庫叢集中所有執行個體上的查閱快取'disabled'。海王星查找緩存 (p. 532)不過，一般而言，在特定執行個體上停用快取會更有意義，方法是將它們從執行個體類型縮減R5d為R5執行個體類型。

海王星中的事務語義

Amazon Neptune 旨在透過資料圖形支援高度並行線上交易處理 (OLTP) 工作負載。適用於 RDF 規格的 W3C SPARQL 查詢語言和 Apache TinkerPop 格林圖遍歷語言文件不會定義並行查詢處理的交易語意。由於 ACID 支持和明確定義的交易保證非常重要，因此我們強制執行嚴格的語義以幫助避免數據異常。

本節定義這些語意，並說明它們如何應用於 Neptune 中的一些常見使用案例。

主題

- 隔離層級的定義 (p. 109)
- 海王星中的事務隔離級別 (p. 109)
- 海王星事務語義的例子 (p. 113)
- 例外狀況處理和重試 (p. 115)

隔離層級的定義

ACID 中的 "I" 代表隔離。交易的隔離程度決定了其他並行交易可能影響其操作的資料有多少。

SQL:1992 標準 建立了描述隔離層級的詞彙。它定義了兩個並發事務之間可能發生的三種類型的交互（它稱之為現象），並Tx1且Tx2：

- Dirty read— 當Tx1修改項目，然後在提交變更之前Tx2讀取該項目時，就會Tx1發生這種情況。然後，如果 Tx1 從未成功遞交變更或將其回復，則 Tx2 已讀取從未進入資料庫的值。
- Non-repeatable read— 當Tx1讀取項目，然後Tx2修改或刪除該項目並提交變更，然後Tx1嘗試重新讀取該項目時，就會發生這種情況。 Tx1現在讀取的值與之前不同，或者發現該項目不再存在。
- Phantom read— 當Tx1讀取符合搜尋條件的一組項目，然後新Tx2增符合搜尋條件的新項目，然後Tx1重複搜尋時，就會發生這種情況。 Tx1現在獲得了一組與以前不同的項目。

這三種類型的互動，每一種都可能導致資料庫中產生的資料不一致。

SQL:1992 標準定義了四種隔離層級，在三種互動類型以及它們可能產生的不一致性方面具有不同的保證。在所有四個層級，可以保證交易完全執行或完全不執行：

- READ UNCOMMITTED— 允許所有三種交互（即髒讀，不可重複讀取和幻象讀取）。
- READ COMMITTED— 骯髒的讀取是不可能的，但不可重複和幻象讀取是.
- REPEATABLE READ-無論是骯髒的讀取也不可重複讀取是可能的，但幻象讀取仍然是。
- SERIALIZABLE— 三種類型的相互作用現象中的任何一種都不可能發生。

多版本並行控制 (MVCC) 允許另一種隔離，即 SNAPSHOT 隔離。這保證在交易開始時，交易可在存在之資料的快照上操作，而且其他交易都不能變更該快照。

海王星中的事務隔離級別

Amazon Neptune 針對唯讀查詢和變異查詢實作不同的交易隔離層級。SPARQL 和 Gremlin 查詢會根據下列條件分類為唯讀或變動：

- 在 SPARQL 中，讀取查詢 (SELECT、ASK、CONSTRUCT 和 DESCRIBE，如 SPARQL 1.1 查詢語言規格中所定義) 與變動查詢 (INSERT 和 DELETE，如 SPARQL 1.1 更新規格中所定義) 之間有明確的區別。

  請注意，Neptune 會將一起提交的多個突變查詢 (例如，在POST訊息中，以分號分隔) 視為單一交易。其為保證作用的原子單位，不是成功就是失敗，若是失敗則會轉返部分變更。

- 然而，在 Gremlin 中，Neptune 會根據查詢是否包含任何查詢路徑步驟 (例如、addE()、addV()或操作資料)，將查詢歸類為唯讀查詢或突變查詢。property() drop()如果查詢包含任何這類路徑步驟，則會將其分類並做為變動查詢執行。

也可以在 Gremlin 中使用常設工作階段。如需詳細資訊，請參閱基於小鬼腳本的會話 (p. 326)。在這些工作階段中，所有查詢 (包括唯讀查詢) 都會在與寫入器端點上的突變查詢相同的隔離下執行。

在 OpenCypher 中使用螺栓讀寫會話，包括只讀查詢在內的所有查詢都與突變查詢相同的隔離下執行，在寫入器端點上。

主題

- 海王星中的唯讀查詢隔離 (p. 110)
- 海王星突變查詢隔離 (p. 110)
- 使用鎖定等待逾時的衝突解決機制 (p. 111)
- 範圍鎖定和錯誤衝突 (p. 111)

海王星中的唯讀查詢隔離

Neptune 會評估快照隔離語意下的唯讀查詢。這意味著，只讀查詢評估開始時，邏輯上採取的數據庫的一致快照進行操作。然後，海王星可以保證以下現象都不會發生：

- Dirty reads— Neptune 中的唯讀查詢永遠不會從並行交易中看到未提交的資料。
- Non-repeatable reads— 多次讀取相同資料的唯讀交易將永遠取回相同的值。
- Phantom reads— 唯讀交易永遠不會讀取交易開始後新增的資料。

由於快照隔離是使用多版本並行控制 (MVCC) 來實現，因此唯讀查詢不需要鎖定資料，因此不會封鎖變動查詢。

僅供讀取複本僅接受唯讀查詢，因此針對僅供讀取複本的所有查詢都會依據 SNAPSHOT 隔離語意執行。

查詢僅供讀取複本時，唯一額外的考量是寫入器和僅供讀取複本之間可能會有較短的複寫延遲。這表示對寫入器所做的更新可能需要很短的時間傳播到您正在從中讀取的僅供讀取複本。實際複製時間取決於主要執行個體的寫入負載。Neptune 架構支援低延遲複寫，並在 Amazon CloudWatch 指標中測試複寫延遲。

儘管如此，由於 SNAPSHOT 隔離層級，讀取查詢永遠都會看到資料庫的一致狀態，即使不是最新的查詢也一樣。

如果您需要強力保證查詢會觀察先前更新的結果，請將查詢傳送到寫入器端點本身，而不是傳送到僅供讀取複本。

海王星突變查詢隔離

做為變動查詢一部分進行的讀取會在 READ COMMITTED 交易隔離下執行，以排除已變更讀取的可能性。超越了通常為READ COMMITTED交易隔離提供的保證，Neptune 提供了強有力的保證，NON-REPEATABLE既不能PHANTOM讀取也不會發生。

這些強力保證是透過讀取資料時鎖定記錄和記錄範圍來實現。這可防止並行交易在讀取後對索引範圍進行插入或刪除，因而保證可重複讀取。

> **Note**
>
> 不過，並行變動交易 Tx2 可能會在變動交易 Tx1 啟動之後開始，並可能在 Tx1 鎖定資料以讀取變更之前遞交變更。在此情況下，Tx1 會看到 Tx2 的變更，宛如在 Tx1 啟動之前，Tx2 就已完成。因為這只適用於遞交的變更，所以 dirty read 永遠不會發生。

要了解 Neptune 用於突變查詢的鎖定機制，它有助於首先了解 Neptune 圖形資料模型 (p. 104) 和索引策略 (p. 104). 海王星使用三個索引管理數據SPOG，即POGS，和GPSO。

為了實現READ COMMITTED交易級別的可重複讀取，Neptune 會在正在使用的索引中採用範圍鎖定。例如，如果變動查詢讀取名為 person1 之頂點的所有屬性和傳出邊緣，則節點會在讀取資料之前，鎖定 SPOG 索引中字首 S=person1 所定義的整個範圍。

使用其他索引時，也適用相同的機制。例如，當變動交易使用 POGS 索引，查詢所指定邊緣標籤的所有來源目標頂點時，P 位置中的邊緣標籤範圍將被鎖定。任何並行交易，無論它是唯讀或變動查詢，仍可在鎖定的範圍內執行讀取。不過，任何涉及在鎖定字首範圍內插入或刪除新記錄的變動都需要獨佔鎖定，而且將予以防止。

換言之，當變動交易讀取某個索引範圍時，強力保證在讀取交易結束之前，任何並行交易都不會修改此範圍。這可保證 non-repeatable reads 不會發生。

使用鎖定等待逾時的衝突解決機制

如果第二個交易嘗試修改第一個交易鎖定之範圍內的記錄，Neptune 會立即偵測到衝突，並封鎖第二個交易。

如果未偵測到相依性鎖死，Neptune 會自動套用鎖定等待逾時機制，在此機制中，封鎖的交易會等待最多 60 秒，讓擁有鎖定的交易完成並釋放鎖定。

- 如果鎖定等待逾時在釋出鎖定之前過期，則會復原封鎖的交易。
- 如果鎖定在鎖定等待逾時內釋出，則第二個交易會解除封鎖並可以成功完成，而不需要重試。

但是，如果 Neptune 偵測到兩個交易之間存在相依性鎖死，則無法自動調解衝突。在此情況下，Neptune 會立即取消並回復兩個交易中的其中一個，而不會啟動鎖定等待逾時。Neptune 會盡最大努力回滾插入或刪除記錄最少的交易。

範圍鎖定和錯誤衝突

海王星需要使用間隙鎖定範圍鎖定。間隙鎖定是鎖定索引記錄之間的間隙，或鎖定最後一個索引記錄之前或之後的間隙。

Neptune 使用所謂的字典表，將數值 ID 值與特定字串常值相關聯。下面是這樣一個海王星字典的示例狀態：table：

字串	ID
類型	1
默認圖形	2
人 _3	3
人 _1	5
知道	6
人 _2	7
年齡	8
邊緣 _1	9
生活 _	10

字串	ID
紐約	11
個人	12
地方	13
邊緣 _2	14

上面的字符串屬於一個屬性圖模型，但這些概念同樣適用於所有 RDF 圖形模型為好。

SPOG（主體-謂詞-對象 _ 圖形）索引的相應狀態如下所示在左側。右側會顯示對應的字串，以協助瞭解索引資料的含義。

秒 (識別碼)	P (身份證)	O (識別碼)	克 (身份證)		S（字符串）	P（字符串）	O（字符串）	G（字符串）
3	1	12	2		人 _3	類型	個人	默認圖形
5	1	12	2		人 _1	類型	個人	默認圖形
5	6	3	9		人 _1	知道	人 _3	邊緣 _1
5	8	40	2		人 _1	年齡	40	默認圖形
5	10	11	14		人 _1	生活 _	紐約	邊緣 _2
7	1	12	2		人 _2	類型	個人	默認圖形
11	1	13	2		紐約	類型	地方	默認圖形

現在，如果突變查詢讀取名為頂點的所有屬性和傳出邊緣`person_1`，則節點將鎖定由 SPOG 索引`S=person_1`中的前綴定義的整個範圍，然後再讀取數據。範圍鎖定會在所有相符的記錄和不相符的第一筆記錄上放置間隙鎖定。匹配的記錄將被鎖定，不匹配的記錄將不會被鎖定。海王星將放置間隙鎖，如下所示：

- `5 1 12 2` （間隙 1）
- `5 6 3 9` （間隙 2）
- `5 8 40 2` （間隙 3）
- `5 10 11 14` （間隙 4）
- `7 1 13 2` （間隙 5）

這會鎖定下列記錄：

- `5 1 12 2`
- `5 6 3 9`
- `5 8 40 2`
- `5 10 11 14`

在此狀態下，下列作業會被合法封鎖：

- 插入的新性質或邊`S=person_1`。不同於`type`或新邊緣的新性質必須位於間隙 2、間隙 3、間隙 4 或間隙 5 中，所有這些都已鎖住。

- 刪除任何現有記錄。

同時，一些並發操作將被錯誤地阻止（產生錯誤的衝突）：

- 任何屬性或邊緣插入S=person_3都會遭到封鎖，因為它們必須進入間隙 1。
- 在 3 和 5 之間分配 ID 的任何新頂點插入都將被阻止，因為它必須進入間隙 1。
- 在 5 到 7 之間分配 ID 的任何新頂點插入都將被阻止，因為它必須進入間隙 5。

間隙鎖定不夠精確，無法鎖定一個特定述詞的間隙 (例如，鎖定謂詞S=5的 gap5)。

範圍鎖定只會放置在發生讀取的索引中。在上述情況下，記錄僅鎖定在 SPOG 索引中，而不是在 POGS 或 GPSO 中。根據存取模式可以在所有索引上執行查詢的讀取，這些模式可以使用 `explain` API 列出 (針對 Sparql (p. 455) 和 Grem (p. 347) lin)。

> **Note**
>
> 也可以採用間隙鎖定來對基礎索引進行安全的並行更新，這也可能導致錯誤的衝突。這些間隙鎖定與交易執行的隔離等級或讀取作業無關。

錯誤衝突不僅可能發生在並行交易因間隙鎖定而發生衝突時，而且在某些情況下，在任何類型的失敗之後重試交易時也會發生。如果失敗觸發的回復仍在進行中，而先前針對交易採用的鎖定尚未完全釋放，則重試將會遇到錯誤的衝突並失敗。

在高負載下，您通常會發現 3-4% 的寫入查詢會因為錯誤的衝突而失敗。對於外部用戶端而言，這類錯誤的衝突很難預測，應該使用重試 (p. 115)來處理。

海王星事務語義的例子

下列範例說明 Amazon Neptune 中交易語意的不同使用案例。

主題

- 範例 1 — 只有在屬性不存在時才插入 (p. 113)
- 示例 2-斷言屬性值是全局唯一的 (p. 114)
- 範例 3 — 如果其他性質具有指定值，則變更性質 (p. 114)
- 實施例 4-替換現有屬性 (p. 115)
- 範例 5-避免懸置屬性或邊線 (p. 115)

範例 1 — 只有在屬性不存在時才插入

假設您想要確保屬性只設定一次。例如，假設多個查詢正在嘗試同時將信用分數指派給人員。您只想要插入一個屬性執行個體，而其他查詢會失敗，因為已設定屬性。

```
# GREMLIN:
g.V('person1').hasLabel('Person').coalesce(has('creditScore'), property('creditScore', 'AAA
+'))

# SPARQL:
INSERT { :person1 :creditScore "AAA+" .}
WHERE  { :person1 rdf:type :Person .
         FILTER NOT EXISTS { :person1 :creditScore ?o .} }
```

Gremlin `property()` 步驟會插入具有所指定索引鍵和值的屬性。`coalesce()` 步驟會在第一個步驟中執行第一個引數，如果失敗，則會執行第二個步驟：

在插入所指定 person1 頂點的 creditScore 屬性值之前，交易必須嘗試為 person1 讀取可能不存在的 creditScore 值。此嘗試讀取會鎖定 SPOG 索引中 S=person1 和 P=creditScore 的 SP 範圍，其中 creditScore 值存在或將被寫入。

採取此範圍鎖定可防止任何並行交易同時插入 creditScore 值。有多個平行交易時，一次最多一個平行交易可以更新該值。這會排除建立多個 creditScore 屬性的異常。

示例 2-斷言屬性值是全局唯一的

假設您想要插入社會安全號碼做為主索引鍵的人員。您想要變動查詢保證，在全域層級中，資料庫中沒有任何其他人具有相同的社會安全號碼：

```
# GREMLIN:
g.V().has('ssn', 123456789).fold()
  .coalesce(__.unfold(),
            __.addV('Person').property('name', 'John Doe').property('ssn', 123456789'))

# SPARQL:
INSERT { :person1 rdf:type :Person .
         :person1 :name "John Doe" .
         :person1 :ssn 123456789 .}
WHERE  { FILTER NOT EXISTS { ?person :ssn 123456789 } }
```

此範例與上述範例類似。主要差別在於範圍鎖定是針對 POGS 索引而非 SPOG 索引採取的。

執行查詢的交易必須讀取模式 (?person :ssn 123456789)，其中繫結 P 和 O 位置。範圍鎖定是針對 P=ssn 和 O=123456789 的 POGS 索引採取的。

- 如果模式的確存在，則不會採取任何動作。
- 如果不存在，鎖定會防止任何並行交易也插入該社會安全號碼

範例 3 — 如果其他性質具有指定值，則變更性質

假設遊戲中的各種事件將人物從第一層移到第二層，並將設為零的新 level2Score 屬性指派給他們。您需要確保這類交易的多個並行執行個體無法建立層級二分數屬性的多個執行個體。Gremlin 和 SPARQL 中的查詢可能看起來如下所示。

```
# GREMLIN:
g.V('person1').hasLabel('Person').has('level', 1)
 .property('level2Score', 0)
 .property(Cardinality.single, 'level', 2)

# SPARQL:
DELETE { :person1 :level 1 .}
INSERT { :person1 :level2Score 0 .
         :person1 :level 2 .}
WHERE  { :person1 rdf:type :Person .
         :person1 :level 1 .}
```

在 Gremlin 中，指定 Cardinality.single 時，property() 步驟會新增屬性，或將現有的屬性值取代為指定的新值。

屬性值的任何更新 (例如將 level 從 1 增加到 2) 會實作為刪除目前記錄，並使用新的屬性值插入新記錄。在此情況下，會刪除層級編號 1 的記錄，並重新插入層級編號 2 的記錄。

若要讓交易能夠新增 level，並將 level2Score 從 1 更新到 2，必須先驗證 level 值目前是否等於 1。執行此動作時，它會對 SPOG 索引中 S=person1、P=level 和 O=1 的 SPO 字首採取範圍鎖定。此鎖定可防止並行交易刪除版本 1 三元組，因此不可能發生起衝突的並行更新。

實施例 4-替換現有屬性

特定事件可能會將人員的信用分數更新為新值 (這裡指的是 BBB)。但您想要確保該類型的並行事件無法為人員建立多個信用分數屬性。

```
# GREMLIN:
g.V('person1').hasLabel('Person')
 .sideEffect(properties('creditScore').drop())
 .property('creditScore', 'BBB')

# SPARQL:
DELETE { :person1 :creditScore ?o .}
INSERT { :person1 :creditScore "BBB" .}
WHERE  { :person1 rdf:type :Person .
         :person1 :creditScore ?o .}
```

這種情況與示例 2 類似，不同之處在於 Neptune 不會鎖定SPO前綴，而是使用S=person1且P=creditScore僅鎖定SP前綴。這可防止並行交易插入或刪除任何具有 person1 主旨之 creditScore 屬性的三元組。

範例 5-避免懸置屬性或邊線

實體上的更新不應留下懸置元素，也就是與未輸入之實體相關聯的屬性或邊緣。這只是 SPARQL 中的問題，因為 Gremlin 具有內建限制條件來防止懸置元素。

```
# SPARQL:
tx1: INSERT { :person1 :age 23 } WHERE { :person1 rdf:type :Person }
tx2: DELETE { :person1 ?p ?o }
```

INSERT 查詢必須讀取和鎖定 SPOG 索引 O=Person 中具有 S=person1 和 P=rdf:type,的 SPO 字首。鎖定可防止 DELETE 查詢平行成功。

在這兩個查詢 (嘗試刪除 :person1 rdf:type :Person 記錄的 DELETE 查詢，以及讀取記錄並在 SPOG 索引中的 SPO 上建立範圍的 INSERT 查詢) 之間的競爭中，可能會產生以下結果：

- 如果 INSERT 查詢在 DELETE 查詢讀取並刪除 :person1 的所有記錄之前遞交，則 :person1 會完全從資料庫中移除，包括新插入的記錄。
- 如果 DELETE 查詢在 INSERT 查詢嘗試讀取 :person1 rdf:type :Person 記錄之前遞交，則讀取會觀察遞交的變更。也就是說，它找不到任何 :person1 rdf:type :Person 記錄，因此變成無操作。
- 如果 INSERT 查詢在 DELETE 查詢讀取之前讀取，則 :person1 rdf:type :Person 三元組會遭到鎖定，而且 DELETE 查詢會遭到封鎖，直到 INSERT 查詢遞交為止，如先前的第一個案例一樣。
- 如果 DELETE 在 INSERT 查詢之前讀取，而且 INSERT 查詢嘗試讀取，並對記錄的 SPO 字首採取鎖定，則會偵測到衝突。這是因為三元組已標記為移除，然後 INSERT 失敗。

在所有這些不同的可能事件序列中，不會建立懸置邊緣。

例外狀況處理和重試

由於無法解決的衝突或鎖定等待逾時而取消交易時，Amazon Neptune 會以. ConcurrentModificationException 如需詳細資訊，請參閱引擎錯誤代碼 (p. 849)。根據最佳實務，用戶端應一律截獲和處理這些例外狀況。

在許多情況下，當 ConcurrentModificationException 執行個體數量很低時，以指數退避為基礎的重試機制也很適合做為處理它們的方式。在這類重試方法中，重試次數和等待時間上限通常取決於交易的大小上限和持續時間。

不過， 如果您的應用程式具有高度並行更新工作負載，而且您觀察到大量 ConcurrentModificationException 事件，則您或許能夠修改應用程式，以減少發生衝突之並行修改的數量。

例如，考量有一個應用程式經常更新一組頂點，並使用多個並行執行緒進行這些更新，以最佳化寫入輸送量。如果每個執行緒持續執行更新一或多個節點屬性的查詢，則相同節點的並行更新可能會產生 ConcurrentModificationException。這樣一來，可能會降低寫入效能。

如果您可以序列化可能彼此衝突的更新，則可以大幅降低此類衝突的可能性。例如，如果您可以確保指定節點的所有更新查詢都是在同一個執行緒上進行 (可能使用雜湊型指派)，則可以確保它們逐一執行，而不是同時執行。雖然對相鄰節點採取的範圍鎖定仍有可能導致 ConcurrentModificationException，但您可以消除對相同節點的並行更新。

亞馬遜海王星資料庫叢集和執

Amazon Neptune 資料庫叢集會透過查詢來管理資料的存取。叢集包含：

- 一個主資料庫執行個體。 。
- 最多 15 個僅供讀取複本資料庫執行個體 。

叢集中的所有執行個體共用相同的基礎受管理儲存磁碟區 (p. 119)，這是專為提供可靠性和高可用性而設計的。

您可以透過 Neptune 端點 (p. 121)連線至資料庫叢集中的資料庫執行個體。

Neptune 資料庫叢集中的主要資料庫執行個體

主資料庫執行個體會將所有寫入作業協調至資料庫叢集的基礎儲存磁碟區。它還支持讀取操作。

Neptune 資料庫叢集中只能有一個主要資料庫執行個體。如果主要執行個體無法使用，Neptune 會自動容錯移轉至具有您可指定優先順序的其中一個僅供讀取複本執行個體。

Neptune 資料庫叢集中的僅供讀取複本資料庫

為資料庫叢集建立主要執行個體後，您可以在資料庫叢集中建立最多 15 個僅供讀取複本執行個體，以支援唯讀查詢。

Neptune 僅供讀取複本資料庫執行個體適用於擴展讀取容量，因為它們完全專用於叢集磁碟區上的讀取作業。所有寫入作業均由主要執行個體管理。每個僅供讀取複本資料庫執行個體都有自己的端

由於叢集中的所有執行個體之間共用叢集儲存磁碟區，因此所有僅供讀取複本執行個體都會針對查詢結果傳回相同的資料，而且複寫延遲很少。這種延遲通常在主要執行個體寫入更新之後不到 100 毫秒，但是當寫入作業的量非常大時，延遲可能會稍長一些。

在不同的可用區域中使用一或多個僅供讀取複本執行個體可提高可用性，因為僅供讀取複本可做為主要執行個體的容錯移轉目標。也就是說，如果主執行個體失敗，Neptune 會將僅供讀取複本執行個體提升為主執行個體。發生這種情況時，升級的執行個體重新啟動時會發生短暫中斷，在此期間，對主要執行個體所做的讀取和寫入要求會失敗，並發生例外狀況。

相反地，如果您的資料庫叢集不包含任何僅供讀取複本執行個體，則當主執行個體發生故障時，資料庫叢集仍然無法使用，直到重新建立為止。重新建立主要執行個體所花費的時間比提升僅供讀取複本要長得多。

為了確保高可用性，建議您建立一或多個僅供讀取複本執行個體，這些執行個體與主執行個體具有相同的資料庫執行個體類別，且位於與主執行個體不同的可用區域。請參閱 Neptune 資料庫叢集的容錯能力 (p. 807)。

透過主控台，您只要在建立資料庫叢集時指定 Multi-AZ (異地同步備份)，即可輕鬆建立異地同步備份部署。如果資料庫叢集位於單一可用區域中，您可以將其設為異地同步備份資料庫叢集，在不同的可用區域中新增 Neptune 複本。

> Note
>
> 您無法為未加密的 Neptune 資料庫叢集建立加密的僅供讀取複本執行個體，或為加密的 Neptune 資料庫叢集建立未加密的僅供讀取複本執行個體。

如需建立 Neptune 僅供讀取複本資料庫執行個體的詳細資訊，請參閱使用主控台建立 Neptune 複本 (p. 543)。

調整 Neptune 資料庫叢集中的資料庫執行

根據您的 CPU 和記憶體需求調整 Neptune 資料庫叢集中的執行個體大小。執行個體上的 vCPU 數目決定了處理傳入查詢的查詢執行緒數目。執行個體上的記憶體容量會決定緩衝區快取的大小，用來儲存從基礎儲存磁碟區擷取的資料頁面複本。

每個 Neptune 資料庫執行個體都具有相當於該執行個體上 2 x 個 vCPU 數量的查詢執行緒數目。例如`r5.4xlarge`，具有 16 個 vCPU，具有 32 個查詢執行緒，因此可以同時處理 32 個查詢。

在所有查詢執行緒佔用時到達的其他查詢會放入伺服器端佇列中，並在查詢執行緒可用時以 FIFO 的方式進行處理。此伺服器端佇列可以容納約 8000 個待處理的要求。一旦它已滿，Neptune 回應其他請求與`ThrottlingException`. 您可以使用`MainRequestQueuePendingRequests`CloudWatch指標來監視擱置的要求數目，或使用具有參數的 Gemlin 查詢狀態端點 (p. 324)。`includeWaiting`

從用戶端的角度來看，查詢執行時間除了實際執行查詢所花費的時間之外，還包括花在佇列中的任何時間。

持續的並行寫入負載，利用主資料庫執行個體上的所有查詢執行緒，理想情況下會顯示 90% 以上的 CPU 使用率，這表示伺服器上的所有查詢執行緒都在積極參與執行有用的工作。不過，即使在持續的並行寫入負載下，實際 CPU 使用率通常也會稍低一些。這通常是因為查詢執行緒正在等待基礎儲存磁碟區的 I/O 作業完成。Neptune 使用仲裁寫入，在三個可用區域複製六份資料，而這六個儲存節點中的四個必須確認寫入，才能將其視為持久性。當查詢執行緒從儲存磁碟區等待此仲裁時，它會停止，進而降低 CPU 使用率。

如果您的序列寫入負載正在執行一個接一個的寫入，並等待第一個寫入完成後再開始下一個，您可以預期 CPU 使用率仍會降低。確切的數量將是 vCPU 和查詢執行緒數目的函數 (查詢執行緒越多，每個查詢的整體 CPU 就越少)，而等待 I/O 會導致一些減少。

如需如何最佳調整資料庫執行個體大小的詳細資訊，請參閱選擇正確的 Neptune 資料庫執行個體 (p. 52)。如需每種執行個體類型的定價資訊，請參閱 Neptune 定價頁面。

在 Neptune 中監視資料庫執行個

您可以使用 Neptune 中的CloudWatch指標來監視資料庫執行個體的效能，並追蹤用戶端觀察到的查詢延遲。請參閱 用CloudWatch來監視 Neptune 中的資料庫執行個體效 (p. 774)。

Amazon 海王星儲存、可靠性和可用性

Amazon Neptune 使用分散式和共用儲存架構，該架構會隨著資料庫儲存需求的增長而自動擴展。

Neptune 資料儲存在叢集磁碟區中，這是使用非揮發性記憶體快速 (NVMe) SSD 磁碟機的單一虛擬磁碟區。叢集磁碟區由稱為區段的邏輯區塊集合所組成。這些區段中的每一個都會分配 10 GB 的儲存空間。每個區段中的資料會複寫成六個副本，然後分配到資料庫叢集所在區域中的AWS三個可用區域 (AZ)。

建立海王星資料庫叢集時，會配置 10 GB 的單一區段。當資料量增加並超過目前配置的儲存區時，Neptune 會透過新增區段來自動擴充叢集磁碟區。除了中國以外的所有支援區域，Neptune 叢集磁碟區的大小上限為 128 TB (TiB)GovCloud，且限制為 64 TiB。但是發行版本： (p. 960)，對於早於發行版本的引擎，所有區域中叢集磁碟區的大小限制為 64 TiB。

資料庫叢集磁碟區包含您所有的使用者資料、索引和字典 (請參閱海王星圖數據模型 (p. 104)本節所述，以及內部中繼資料 (例如內部交易記錄)。所有這些圖形資料 (包括索引和內部記錄) 都不能超過叢集磁碟區的大小上限。

海王星儲存分配

即使 Neptune 叢集磁碟區可以成長為 128 TiB (或少數區域中的 64 TiB)，您只需支付實際配置的空間費用。配置的總空間由儲存區高水位標記決定，這是在叢集磁碟區存在期間隨時配置給叢集磁碟區的最大容量。

這表示即使使用者資料已從叢集磁碟區移除，例如透過放置查詢`g.V().drop()`，總配置的空間仍保持不變。Neptune 會自動優化未使用的分配空間，以便將來重複使用。

除了使用者資料之外，另外兩種類型的內容會佔用內部儲存空間，即字典資料和內部交易記錄。雖然字典資料與圖形資料一起儲存，但即使刪除了它所支援的圖形資料，也就是說，如果重新引入資料，也就是說，可以重複使用項目。內部日誌數據存儲在具有自己高水位的單獨內部存儲空間中。當內部記錄檔到期時，它所佔用的儲存空間可以重複用於其他記錄檔，但不能用於圖形資料。已配置給記錄檔的內部空間量會包含在`VolumeBytesUsed`CloudWatch指標 (p. 772)報告的總空間中。

檢查將配置儲存最佳做法 (p. 119)的儲存裝置保持在最低限度並重複使用空間的方法。

海王星儲存計費

如上所述，儲存成本是根據儲存量高水位標記計費。雖然您的資料會複寫為六個副本，但您只需支付一份資料副本的費用。

您可以透過監視`VolumeBytesUsed`CloudWatch指標來判斷資料庫叢集目前的儲存高水位標記為何 (請參閱使用 Amazon 監控海王星 CloudWatch (p. 772))。

其他可能會影響 Neptune 儲存體成本的因素包括資料庫快照和備份，這些快照集和備份會以備份儲存體分別計費，並以 Neptune 儲存體成本為基礎 (請參閱CloudWatch 有助於管理 Neptune 備份儲存體的指標 (p. 808)。

但是，如果您建立資料庫的複 (p. 564)製，複製會指向資料庫叢集本身使用的相同叢集磁碟區，因此原始資料不會產生額外的儲存費用。複製後續變更會使用copy-on-write通訊協 (p. 565)定，並且會產生額外的儲存成本。

如需更多海王星定價資訊，請參閱 Amazon 海王星定價。

海王星儲存最佳做法

由於某些類型的資料會耗用 Neptune 中的永久儲存，因此請使用下列最佳做法來避免儲存體成長的大量尖峰：

- 在設計圖形資料模型時，請盡量避免使用本質上屬於暫時性質的屬性索引鍵和使用者面向值。

- 如果您打算變更資料模型，請在使用快速重設 API (p. 539) 清除該資料庫叢集中的資料之前，請勿使用新模型將資料載入到現有的資料庫叢集。最好的事情通常是將使用新模型的數據加載到新的數據庫集群中。
- 對大量資料進行操作的交易會產生相對應的大型內部記錄檔，這可能會永久增加內部記錄空間的高水位記。例如，刪除資料庫叢集中所有資料的單一交易可能會產生巨大的內部記錄檔，該記錄檔需要配置大量內部儲存空間，進而永久減少可用於圖形資料的空間。

  為了避免這種情況，請將大型交易拆分為較小的交易，並允許它們之間的時間，以便相關聯的內部日誌有機會過期並釋放其內部存儲空間以供後續日誌重複使用。
- 若要監視 Neptune 叢集磁碟區的成長，您可以在`VolumeBytesUsed`CloudWatch指標上設定CloudWatch警示。如果您的資料達到叢集磁碟區的最大大小，這會特別有用。如需詳細資訊，請參閱使用 Amazon CloudWatch 警示。

當您有大量未使用的配置空間時，縮小資料庫叢集使用的儲存空間的唯一方法是匯出圖形中的所有資料，然後將其重新載入到新的資料庫叢集中。如需從資料庫叢集匯出資料的簡單方法，請參閱 Neptune 的資料匯出服務和公用程 (p. 670)式，以及 Neptune 的大量載 (p. 255)入器，以便輕鬆將資料匯入 Neptune。

> **Note**
>
> 建立和還原快照 (p. 811)並不會減少為資料庫叢集配置的儲存空間量，因為快照會保留叢集基礎儲存體的原始映像。如果沒有使用大量已配置的儲存空間，縮小已配置儲存體數量的唯一方法是匯出圖形資料並將其重新載入到新的資料庫叢集中。

海王星儲存可靠性與高可用性

亞馬遜海王星的設計具有可靠、耐用和容錯能力。

在三個可用區域 (AZ) 之間維護六個 Neptune 資料副本的事實可確保資料的儲存具有高度耐用性，而且資料遺失的可能性極低。無論其中是否有資料庫執行個體，資料都會自動跨可用區域複寫，並且複寫量與叢集中的資料庫執行個體數量無關。

這表示您可以快速新增僅供讀取複本，因為 Neptune 不會建立圖形資料的新副本。僅供讀取複本會連線至已包含您資料的叢集磁碟區。同樣地，移除僅供讀取複本並不會移除任何基礎資料。

您只能在刪除叢集磁碟區的所有資料庫執行個體後刪除叢集磁碟區及其資料。

Neptune 也會自動偵測組成叢集磁碟區的區段中的故障。當區段中的資料副本損毀時，Neptune 會立即修復該區段，並使用相同區段內的其他資料副本，以確保修復的資料是最新的。因此，Neptune 可避免資料遺失，並減少執行point-in-time還原以從磁碟故障復原的需求。

連接到亞馬遜海王星端點

Amazon Neptune 使用的是資料庫執行個體叢集，而不是單一執行個體。每個 Neptune 連線都由特定的資料庫執行個體處理。當您連線到 Neptune 叢集時，您指定的主機名稱和連接埠會指向稱為端點的中繼處理常式。端點是包含主機位址和連接埠的 URL。Neptune 端點使用加密的傳輸層安全性/安全通訊端層 (TLS/SSL) 連線。

Neptune 使用端點機制來抽象這些連接，這樣您就不必對主機名進行硬編碼，或者在某些數據庫實例不可用時編寫自己的邏輯來重新路由連接。

您可以使用端點將每個連線對應至適當的執行個體或執行個體群組，視您的使用案例而定。自訂端點可讓您連線至資料庫執行個體的子集。Neptune 資料庫叢集中提供下列端點：

海王星叢集端點

叢集端點是 Neptune 資料庫叢集的端點，可連線至該資料庫叢集的目前主要資料庫執行個體。每個 Neptune 資料庫叢集都有叢集端點和一個主要資料庫執行個體。

叢集端點可為資料庫叢集的讀寫連接提供容錯移轉支援。對資料庫叢集上的所有寫入操作，包括插入、更新、刪除和資料定義語言 (DDL) 變更使用叢集端點。您也可以對讀取操作 (例如查詢) 使用叢集端點。

如果目前資料庫叢集的主要資料庫執行個體發生故障，Neptune 會自動容錯移轉到新的主要資料庫執行個體。容錯移轉期間，資料庫叢集會繼續從新的主要資料庫執行個體對叢集端點提供連接請求，將對服務的中斷降到最低。

下列範例說明 Neptune 資料庫叢集的叢集端點。

```
mydbcluster.cluster-123456789012.us-east-1.neptune.amazonaws.com:8182
```

海王星讀者端點

讀取器端點是 Neptune 資料庫叢集的端點，可連線至該資料庫叢集的其中一個可用 Neptune 複本。每個 Neptune DB 叢集都有一個讀取器端點。如果有多個 Neptune 複本，讀取器端點會將每個連線要求導向至其中一個 Neptune 複本。

讀取器端點可為資料庫叢集的唯讀連線提供循環配置資源路由。對讀取操作 (例如 查詢) 使用讀取器端點。

除非您有單一執行個體叢集 (沒有僅供讀取複本的叢集)，否則無法使用讀取器端點進行寫入作業。在這種情況下 (以及只有在這種情況下)，讀取器可以用於寫入操作以及讀取操作。

讀取器端點循環配置資源路由的運作方式是改變 DNS 項目指向的主機。每當您解析 DNS，就會取得不同的 IP，且對這些 IP 開放連線。建立連線後，所有對該連線的請求都會傳送到同一主機。用戶端必須再次建立新的連線和解析 DNS 記錄，以取得可能不同的僅供讀取複本的連線。

> **Note**
>
> WebSockets連接通常長時間保持活動狀態。若要獲得不同的僅供讀取複本，請執行下列動作：
>
> - 確保您的用戶端於每次連線時解析 DNS 項目。
> - 關閉連線並重新連線。

不同的用戶端軟體可能會以不同的方式解析 DNS。例如，如果您的用戶端解析 DNS，然後將 IP 用於每一個連線，則軟體會將所有請求導向單一主機。

用戶端或代理伺服器的 DNS 快取，會從快取將 DNS 名稱解析至同一個端點。這對循環配置資源路由和容錯移轉情況都構成問題。

Note

停用所有 DNS 快取設定，以強制每次執行 DNS 解析。

資料庫叢集會在可用 Neptune 複本之間將連線要求散佈至讀取器端點。如果資料庫叢集僅包含一個主要資料庫執行個體，讀取器端點會從主要資料庫執行個體提供連接請求。如果為該資料庫叢集建立了 Neptune 複本，則讀取器端點會繼續從新 Neptune 複本提供給讀取器端點的連線要求，而且服務中斷時間最小。

下列範例說明 Neptune 資料庫叢集的讀取器端點。

```
mydbcluster.cluster-ro-123456789012.us-east-1.neptune.amazonaws.com:8182
```

海王星實例端點

執行個體端點是 Neptune 資料庫叢集中資料庫執行個體的端點，連線至該特定資料庫執行個體。資料庫叢集中的每個資料庫執行個體，不論執行個體類型為何，都有自己的唯一執行個體端點。因此，資料庫叢集目前的主要資料庫執行個體有一個執行個體端點。資料庫叢集中的每個 Neptune 複本也有一個執行個體端點。

執行個體端點對資料庫叢集的連接提供直接控制，使用叢集端點或讀取器端點的案例可能不適用。例如，根據工作負載類型而定，您的用戶端應用程式可能需要精細負載平衡。在此情況下，您可以設定多個用戶端連線至資料庫叢集中的不同 Neptune 複本，以分配讀取工作負載。

下列範例說明 Neptune 資料庫叢集中資料庫執行個體的執行個體端點。

```
mydbinstance.123456789012.us-east-1.neptune.amazonaws.com:8182
```

海王星自訂端點

Neptune 叢集的自訂端點代表您選擇的一組資料庫執行個體。當您連線到端點時，Neptune 會選擇群組中的其中一個執行個體來處理連線。您可以定義此端點要參考的執行個體，並且決定端點的運作目的。

Neptune DB 叢集在您建立自訂端點之前沒有自訂端點，而且您可以為每個佈建的 Neptune 叢集建立最多五個自訂端點。

自訂端點可根據資料庫執行個體的唯讀或讀寫功能以外的準則來提供負載平衡的資料庫連線。由於連接可以連接到與端點關聯的任何資料庫執行個體，因此請確保該群組內的所有執行個體共用相同的效能和記憶體容量特性。使用自訂端點時，您一般不會對該叢集使用讀取器端點。

此功能適用於具有特殊工作負載類型的進階使用者，而這些工作負載不切實際保持叢集中的所有 Neptune 複本相同。透過自訂端點，您可以調整每個連線所使用之資料庫執行個體的容量。

例如，如果您定義數個自訂端點，這些端點連線到具有不同執行個體類別的執行個體群組，則可以將效能需求不同的使用者導向最適合其使用案例的端點。

下列範例說明 Neptune 資料庫叢集中資料庫執行個體的自訂端點：

```
myendpoint.cluster-custom-123456789012.us-east-1.neptune.amazonaws.com:8182
```

如需詳細資訊，請參閱 使用自訂端點 (p. 123)。

海王星端點考量

使用 Neptune 端點時，請考慮下列事項：

- 使用執行個體端點來連接至資料庫叢集中的特定資料庫執行個體之前，請考慮改為針對資料庫叢集使用叢集端點或讀取器端點。

  叢集端點和讀取器端點提供高可用性案例的支援。如果資料庫叢集的主要資料庫執行個體發生故障，Neptune 會自動容錯移轉到新的主要資料庫執行個體。它可以透過將現有 Neptune 複本升級到新的主要資料庫執行個體，或建立新的主要資料庫執行個體。如果發生容錯移轉，您可以使用叢集端點重新連

線至新升級或建立的主要資料庫執行個體，或使用讀取器端點重新連線至資料庫叢集中的其他 Neptune 複本。

如果您不採取此方法，您仍可以針對預期的操作確定已連接至資料庫叢集中的正確資料庫執行個體。若要這麼做，您可以手動或以程式設計方式探索資料庫叢集中可用資料庫執行個體的結果集，並在容錯移轉之後，於使用特定資料庫執行個體的執行個體端點之前確認其執行個體類型。

如需容錯移轉的詳細資訊，請參閱 Neptune 資料庫叢集的容錯能力 (p. 807)。

- 讀取器端點只會將連線導向至 Neptune 資料庫叢集中可用的 Neptune 複本。它不會將特定查詢導向。

  Important

  海王星不負載平衡。

  如果您想要平衡查詢負載以分配資料庫叢集的讀取工作負載，您必須管理您應用程式的負載。您必須使用執行個體端點直接連線至 Neptune 複本，以平衡負載。

- 讀取器端點循環配置資源路由的運作方式是改變 DNS 項目指向的主機。用戶端必須建立新的連線並再次解析 DNS 紀錄，以取得可能是新的僅供讀取複本的連線。

- 在容錯移轉期間，當 Neptune 複本升級到新的主要資料庫執行個體時，讀取器端點可能會在短時間內將連線導向至資料庫叢集的新主要資料庫執行個體。

在 Neptune 中使用自訂端點

新增資料庫執行個體至自訂端點或將它從自訂端點移除時，對該資料庫執行個體的任何現有連接會保持作用中。

您可以定義要包含在自訂端點 (靜態清單) 中的資料庫執行個體清單，也可以定義要從自訂端點中排除的資料庫執行個體清單 (. 排除列表) 。您可以使用包含/排除機制將資料庫執行個體細分為群組，並確保自訂端點涵蓋叢集中的所有資料庫執行個體。每個自訂端點只能包含這些清單類型的其中一個。

在AWS Management Console中，選項會以核取方塊 Attach future instances added to this cluster (連接新增至此叢集的未來執行個體) 呈現。將該核取方塊保持未選取時，自訂端點會使用僅包含對話方塊中所指定資料庫執行個體的靜態清單。當您核取此核取方塊時，自訂端點會使用例外清單。在這種情況下，自訂端點代表叢集中的所有資料庫執行個體 (包括您將來新增的任何資料庫執行個體)，但對話方塊中未選取的執行個體除外。

當資料庫執行個體因容錯移轉或升級而在主執行個體和 Neptune 複本之間變更角色時，Neptune 不會變更靜態或排除清單中指定的資料庫執行個體。

您可以將一個資料庫執行個體與多個自訂端點建立關聯。例如，假設您將新的資料庫執行個體新增至叢集。在這種情況下，資料庫執行個體會新增至符合資格的所有自訂端點。為其定義的靜態或例外清單決定了可以新增哪個資料庫執行個體。

如果端點包含資料庫執行個體的靜態清單，則不會將新增的 Neptune 複本新增至其中。相反地，如果端點具有排除清單，則只要新增的 Neptune 複本未在例外清單中命名，則會將新增的 Neptune 複本新增至該複本。

如果 Neptune 複本無法使用，它會保持與其自訂端點相關聯。無論由於其他原因而不健康，停止，重新啟動還是無法使用，都是如此。但是，只要它仍然不可用，您就無法通過任何端點連接到它。

由於新建立的 Neptune 叢集沒有自訂端點，因此您必須自行建立和管理它們。這也適用於從快照還原的 Neptune 叢集，因為自訂端點不包含在快照中。您必須在還原之後再次建立它們，如果還原的叢集與原始叢集位於相同的區域，請選擇新的端點名稱。

建立自訂端點

使用 Neptune 主控台管理自訂端點。請瀏覽至 Neptune 叢集的詳細資料頁面，並使用「自訂端點」區段中的控制項來執行此操作。

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 瀏覽叢集詳細資訊頁面。
3. 在「端點」區段中選擇`Create custom endpoint`處理行動。
4. 為您的使用者 ID 和地區唯一的自訂端點選擇一個名稱。名稱長度必須在 63 個字元以內，並採用下列格式：

   `endpointName``.cluster-custom-`*`customerDnsIdentifier.dnsSuffix`*

   由於自訂端點名稱未包含您的叢集名稱，如果將叢集重新命名，便不需變更這些名稱。但是，您無法對同一區域中的多個叢集重複使用相同的自訂端點名稱。提供每個自訂端點一個名稱，該名稱在您的使用者 ID 於特定區域內所擁有的叢集間都是唯一的。
5. 若要選取即使在叢集展開時仍保持相同的資料庫執行個體清單，請將核取方塊 Attach future instances added to this cluster (連接新增至此叢集的未來執行個體) 保持未選取。核取該核取方塊後，自訂端點會動態新增任何新增至叢集的執行個體。

檢視自訂端點

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 瀏覽至資料庫叢集的叢集詳細資訊頁面。
3. 「端點」區段僅包含自訂端點的相關資訊（有關內建端點的詳細資料會列在主要「詳細資料」區段中）。若要查看特定自訂端點的詳細資料，請選取其名稱以顯示該端點的詳細資料頁面。

編輯自訂端點

您可以編輯自訂端點的內容，以變更與其相關聯的資料庫執行個體。您也可以在靜態清單和例外清單之間切換。

您無法在編輯動作的變更進行中時連線至或使用自訂端點。在您進行變更之後，端點狀態可能需要幾分鐘的時間，才能將端點狀態返回「可用」，並且您可以再次連線

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 瀏覽叢集詳細資訊頁面。
3. 在「端點」區段中，選擇您要編輯的自訂端點名稱。
4. 在該端點的詳細資料頁面中，選擇 「編輯」 動作。

刪除自訂端點

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 瀏覽叢集詳細資訊頁面。
3. 在「端點」區段中，選擇要刪除的自訂端點名稱。
4. 在該端點的詳細資料頁面中，選擇「刪除」動作。

將自定義 ID 注入海王星小鬼或 SPARQL 查詢

依預設，Neptune 會為每個查詢指派唯一queryId值。您可以使用此 ID 來取得執行中查詢的相關資訊 (請參閱小鬼查詢狀態 API (p. 324) 或 查詢狀態 API (p. 449))，或取消它 (請參閱小精靈查詢取消 (p. 326)或取消查詢 (p. 450))。

海王星也可以讓您指定您自己的queryId值給 Grimlin 或 SPARQL 查詢，無論是在 HTTP 標頭，或使用查詢提示 SPARQL 查詢。queryId指派您自己的 queryID 可讓您輕鬆追蹤查詢，以取得狀態或取消該查詢。

Note

此功能可從開始使用版本 1.0.0.0.463.0 (p. 985)。

使用 HTTP 標頭來注入自訂 queryId 值

對於 Gremlin 和 SPARQL 二者，HTTP 標頭可用來將您自己的 queryId 值注入至查詢。

Gremlin 範例

```
curl -XPOST https://your-neptune-endpoint:port \
    -d "{\"gremlin\": \
        \"g.V().limit(1).count()\" , \
        \"queryId\":\"4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47\"  }"
```

SPARQL 範例

```
curl https://your-neptune-endpoint:port/sparql \
    -d "query=SELECT * WHERE { ?s ?p ?o } " \
       --data-urlencode \
       "queryId=4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47"
```

使用 SPARQL 查詢提示來注入自訂 queryId 值

以下範例說明如何使用 SPARQL queryId 查詢提示，將自訂 queryId 值注入至 SPARQL 查詢：

```
curl https://your-neptune-endpoint:port/sparql \
    -d "PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#> \
       SELECT * WHERE { hint:Query hint:queryId \"4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47\" \
       {?s ?p ?o}}"
```

使用 queryId 值檢查查詢狀態

Gremlin 範例

```
curl https://your-neptune-endpoint:port/gremlin/status \
    -d "queryId=4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47"
```

SPARQL 範例

```
curl https://your-neptune-endpoint:port/sparql/status \
    -d "queryId=4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47"
```

海王星實驗室

您可以使用 Amazon Neptune 實驗室模式來啟用目前 Neptune 引擎版本中的新功能，但尚未準備好用於生產環境，且預設情況下未啟用這些功能。這可讓您在開發和測試環境中試用這些功能。

Note

此功能可從開始使用版本 1.0.0.0.463.0 (p. 985)。

使用海王星實驗室

使用資neptune_lab_mode料庫叢集參數 (p. 531)來啟用或停用功能。做法是在資料庫叢集參數群組的 neptune_lab_mode 參數值中包含 *(feature name)*=enabled 或 *(feature name)*=disabled。

例如，在此引擎版本中，您可將 neptune_lab_mode 參數設為 Streams=disabled, ReadWriteConflictDetection=enabled。

如需如何針對您的資料庫編輯資料庫叢集參數群組的相關資訊，請參閱編輯參數群組 (p. 528)。請注意，您無法編輯預設資料庫叢集參數群組；如果您是使用預設群組，則必須建立新的資料庫叢集參數群組，才能設定 neptune_lab_mode 參數。

Note

當您變更靜態資料庫叢集參數 (例如) 時neptune_lab_mode，必須重新啟動叢集的主要 (寫入器) 執行個體，變更才會生效。之前已發行版本: (p. 884)，當主執行個體重新啟動時，資料庫叢集中的所有僅供讀取複本都會自動重新啟動。
從開始已發行版本: (p. 884)，重新啟動主執行個體不會導致任何複本重新啟動。這表示您必須個別重新啟動每個執行個體，才能取得資料庫叢集參數變更 (請參閱參數群組 (p. 527))。

Important

目前，如果您提供了錯誤的實驗室模式參數，或者您的請求因其他原因而失敗，則可能不會通知您失敗。您應該始終通過調用狀態 API (p. 770) 來驗證實驗室模式更改請求是否成功，如下所示：

```
curl -G https://your-neptune-endpoint:port/status
```

狀態結果包括實驗室模式資訊，這些資訊會顯示您要求的變更是否已進行：

```
{
  "status":"healthy",
  "startTime":"Wed Dec 29 02:29:24 UTC 2021",
  "dbEngineVersion":"development",
  "role":"writer",
  "dfeQueryEngine":"viaQueryHint",
  "gremlin":{"version":"tinkerpop-3.5.2"},
  "sparql":{"version":"sparql-1.1"},
  "opencypher":{"version":"Neptune-9.0.20190305-1.0"},
  "labMode":{
    "ObjectIndex":"disabled",
    "ReadWriteConflictDetection":"enabled"
  },
  "features":{
    "LookupCache":{"status":"Available"},
    "ResultCache":{"status":"disabled"},
    "IAMAuthentication":"disabled",
    "Streams":"disabled",
    "AuditLog":"disabled"
  },
  "settings":{"clusterQueryTimeoutInMs":"120000"}
```

```
}
```

目前使用實驗室模式存取下列功能：

OSGP 指數

Neptune 現在可以維護第四個索引，也就是 OSGP 索引，這對於具有大量述詞的資料集很有用 (請參閱啟用 OSGP 索引 (p. 105))。

> **Note**
>
> 此功能從海王星引擎版本 1.0. (p. 969) 2.1 開始提供。

您可以在資料庫叢集參數中設定，在新的空白海王星資料庫叢集`ObjectIndex=enabled`中啟用 OSGP 索引。`neptune_lab_mode`OSGP 索引只能在新的空白資料庫叢集中啟用。

根據預設，停用 OSGP 索引。

> **Note**
>
> 設定`neptune_lab_mode`資料庫叢集參數以啟用 OSGP 索引之後，您必須重新啟動叢集的寫入器執行個體，變更才會生效。

> **Warning**
>
> 如果您透過設定來停用已啟用的 OSGP 索引，`ObjectIndex=disabled`然後在新增更多資料後重新啟用該索引，則索引將無法正確建立。不支援依需求重建索引，因此您應該只在資料庫為空時啟用 OSGP 索引。

正規化的交易語意

Neptune 已更新並發交易的正式語意 (請參閱海王星中的事務語義 (p. 109))。

在啟用或停用格式化交易語義的 `neptune_lab_mode` 參數中，使用 `ReadWriteConflictDetection` 做為名稱。

根據預設，已啟用正規化的交易語意。如果您想要還原為先前行為，請將 `ReadWriteConflictDetection=disabled` 併入針對資料庫叢集 `neptune_lab_mode` 參數設定的值中。

亞馬遜海王星替代查詢引擎（DFE）

Amazon Neptune 具有稱為 DFE 的替代查詢引擎，它比原始 Neptune 引擎更有效地使用資料庫執行個體資源 (例如 CPU 核心、記憶體和 I/O)。

> Note
>
> 對於大型資料集，DFE 引擎可能無法在 t3 執行個體上順利執行。

DFE 引擎可執行 SPARQL、小靈和 OpenCypher 查詢，並支援多種計畫類型，包括左深、濃密和混合式計畫類型。計劃運算子可以叫用在一組保留的運算核心上執行的運算作業，以及 I/O 作業，每個作業都會在 I/O 執行緒集區的自己執行緒上執行。

DFE 使用有關 Neptune 圖形資料的預先產生統計資料，做出有關如何結構查詢的明智決策。請參DFE 統計資料 (p. 131)閱，以取得如何產生這些統計資料的資訊。

系統會根據預先產生的統計資料和 Neptune 頭節點中可用的資源，自動選擇計劃類型和使用的運算執行緒數目。結果的順序不是針對具有內部計算平行處理原則的計劃預先確定。

控制海王星 DFE 引擎的使用位置

根據預設，海王星查詢引擎 (p. 533)執行個體的執行個體參數設定為`viaQueryHint`，這會導致 DFE 引擎僅用於 OpenCypher 查詢，以及明確包含設定之查詢提示的 Gremlin 和 SPARQL 查詢。`useDFE true`

您可以透過將`neptune_dfe_query_engine`啟用例證參數設定為來完全啟用 DFE 引擎，以便盡可能使用 DFE 引擎。

您也可以透過包含特定小精靈`useDFE` (p. 322)查詢或 SPARQL 查詢的查詢提示來停用 DFE。 (p. 444)此查詢提示可讓您防止 DFE 執行該特定查詢。

您可以使用執行個體狀態 (p. 770)呼叫判斷是否在執行個體中啟用 DFE，如下所示：

```
curl -G https://your-neptune-endpoint:port/status
```

然后，狀態響應會指定是否啟用 DFE：

```
{
  "status":"healthy",
  "startTime":"Wed Dec 29 02:29:24 UTC 2021",
  "dbEngineVersion":"development",
  "role":"writer",
  "dfeQueryEngine":"viaQueryHint",
  "gremlin":{"version":"tinkerpop-3.5.2"},
  "sparql":{"version":"sparql-1.1"},
  "opencypher":{"version":"Neptune-9.0.20190305-1.0"},
  "labMode":{
    "ObjectIndex":"disabled",
    "ReadWriteConflictDetection":"enabled"
  },
  "features":{
    "ResultCache":{"status":"disabled"},
    "IAMAuthentication":"disabled",
    "Streams":"disabled",
    "AuditLog":"disabled"
  },
  "settings":{"clusterQueryTimeoutInMs":"120000"}
}
```

Gremlin explain 和profile結果會告訴您 DFE 是否正在執行查詢。請小靈報告中包含的信息 explain (p. 351)參閱DFE profile 報告 (p. 360)的explain和profile。

同樣地，SPARQL 會explain告訴您是否正由 DFE 執行 SPARQL 查詢。DFENode運算子 (p. 468)如需詳細資訊啟用 DFE 時的 SPARQL explain 輸出範例 (p. 461)，請參閱和。

海王星 DFE 支援的查詢結構

目前，海王星 DFE 支援 SPARQL 和小鬼查詢結構的子集。

對於 SPARQL，這是結合基本圖形模式的子集。

對於 Gemlin 來說，它通常是查詢的子集，其中包含不包含一些更複雜的步驟的遍歷鏈。

您可以找出 DFE 是全部還是部分執行查詢，如下所示：

- 在 Gremlin 中，explainprofile結果會告訴您 DFE 正在執行查詢的哪些部分（如果有的話）。請小靈報告中包含的信息 explain (p. 351)參閱DFE profile 報告 (p. 360)的explain和profile。另請參閱使用和調整小鬼查詢 explainprofile (p. 364)。

  有關海王星引擎對個別 Gemlin 步驟支援的詳細資訊，請參閱中記錄。小精靈步支持 (p. 377)
- 同樣地，SPARQL 會explain告訴您是否正由 DFE 執行 SPARQL 查詢。DFENode運算子 (p. 468)如需詳細資訊啟用 DFE 時的 SPARQL explain 輸出範例 (p. 461)，請參閱和。

管理要使用的海王星 DFE 的統計資料

Note

對開放密碼的支持取決於海王星的 DFE 查詢引擎。
DFE 引擎首先在海王星引擎版本 1.0.3.0 (p. 954) 的實驗室模式中提供，而且從海王星引擎版本 1.0.5.0 (p. 931) 開始，預設為啟用，但僅適用於查詢提示和 OpenCypher 支援。
從 Neptune 引擎版本 1.1.1.0 (p. 894) 開始，DFE 引擎不再處於實驗室模式，現在可以使用執行個體資料庫參數群組中的海王星查詢引擎 (p. 533)執行個體參數來控制。

DFE 引擎會在規劃查詢執行時，使用 Neptune 圖表中的資料相關資訊，進行有效的權衡。這項資訊採用統計資料的形式，其中包括所謂的特徵集和述詞統計資料，可以引導查詢規劃。

從引擎版本 1.2.1.0 (p. 861) 開始，您可以使用 GetGraphSummary (p. 206)API 或端點從這些統計資料擷取有關圖表的摘要資訊 (p. 138)。`summary`

只要圖表中超過 10% 的資料變更，或最新的統計資料超過 10 天，就會重新產生這些 DFE 統計資料。但是，這些觸發器將來可能會發生變化。

您可以透過下列其中一個端點管理 DFE 統計資料的產生：

- `https://`*`your-neptune-host`*`:`*`port`*`/rdf/statistics` (適用於 SPARQL)。
- `https://`*`your-neptune-host`*`:`*`port`*`/propertygraph/statistics` (對於小精靈和開放密碼) ，以及它的替代版本：。`https://`*`your-neptune-host`*`:`*`port`*`/pg/statistics`

Note

從引擎版本 1.1.1.0 (p. 894) 開始，Grimlin 統計端點 (`https://`*`your-neptune-host`*`:`*`port`*`/gremlin/statistics`) 已被棄用以支持或端點。`propertygraph` pg它仍然支持向後兼容性，但可能會在未來的版本中刪除。
從引擎版本 1.2.1.0 (p. 861) 開始，SPARQL 統計資料端點 (`https://`*`your-neptune-host`*`:`*`port`*`/sparql/statistics`) 已被取代以支援端點。`rdf`它仍然支持向後兼容性，但可能會在未來的版本中刪除。

在以下範例中，`$STATISTICS_ENDPOINT`代表這些端點 URL 中的任何一個。

Note

如果 DFE 統計資料端點位於讀取器執行個體上，則唯一可以處理的要求是狀態要求 (p. 131)。其他請求將失敗，並顯示`ReadOnlyViolationException`.

Note

`T3`和`T4g`執行個體上的統計資料產生會停用，因為它可能超過這些執行個體類型的記憶體容量。

DFE 統計資料產生的大小限制

目前，如果達到下列任一大小限制，DFE 統計資料產生便會中止：

- 產生的特性集數不得超過 50,000 個。
- 產生的述詞統計資料數目不可超過一百萬個。

這些限制可能會改變。

DFE 統計資料的目前狀態

您可以使用下列`curl`要求來檢查 DFE 統計資料的目前狀態：

```
curl -G "$STATISTICS_ENDPOINT"
```

對狀態請求的響應包含以下字段：

- `status`— 要求的 HTTP 傳回程式碼。如果請求成功，則代碼為`200`。常見錯誤 (p. 136)如需常見錯誤的清單，請參閱。
- `payload`:
  - `autoCompute`— (布林值) 指出是否啟用自動產生統計資料。
  - `active`— (布林值) 指出是否完全啟用 DFE 統計資料產生。
  - `statisticsId` — 報告目前統計資料產生執行的 ID。的值 `-1` 表示尚未產生任何統計資料。
  - `date`— 最近產生 DFE 統計資料的世界標準時間，格式為 ISO 8601。

    Note

    在引擎版本 1.2.1.0 (p. 861) 之前，這是以微小的精確度表示，但從引擎版本 1.2.1.0 向前，它會以毫秒精確度表示 (例如，)。`2023-01-24T00:47:43.319Z`
  - `note`— 關於統計資料無效情況下的問題的注意事項。
  - `signatureInfo`— 包含統計資料中產生之特性集的相關資訊 (在引擎版本 1.2.1.0 (p. 861) 之前，此欄位已命名`summary`)。這些通常不是直接採取行動的：
    - `signatureCount`— 所有特徵集中的簽名總數。
    - `instanceCount`— 特徵集執行個體的總數。
    - `predicateCount`— 唯一述詞的總數。

未產生統計資料時，對狀態要求的回應如下所示：

```
{
  "status" : "200 OK",
  "payload" : {
    "autoCompute" : true,
    "active" : false,
    "statisticsId" : -1
   }
}
```

如果 DFE 統計資料可用，回應如下所示：

```
{
  "status" : "200 OK",
  "payload" : {
    "autoCompute" : true,
    "active" : true,
    "statisticsId" : 1588893232718,
    "date" : "2020-05-07T23:13Z",
    "summary" : {
      "signatureCount" : 5,
      "instanceCount" : 1000,
      "predicateCount" : 20
    }
  }
}
```

如果 DFE 統計資料產生失敗，例如因為超出統計資料大小限制 (p. 131)，回應如下所示：

```
{
```

```
  "status" : "200 OK",
  "payload" : {
    "autoCompute" : true,
    "active" : false,
    "statisticsId" : 1588713528304,
    "date" : "2020-05-05T21:18Z",
    "note" : "Limit reached: Statistics are not available"
  }
}
```

停用自動產生 DFE 統計資料

依預設，當您啟用 DFE 時，會啟用自動產生 DFE 統計資料。

您可以停用自動產生，如下所示：

```
curl -X POST "$STATISTICS_ENDPOINT" -d '{ "mode" : "disableAutoCompute" }'
```

如果請求成功，則 HTTP 響應代碼為200，響應為：

```
{
  "status" : "200 OK"
}
```

您可以發出狀態要求 (p. 131)並檢查回應中的autoCompute欄位是否設定為，以確認已停用自動產生false。

停用自動產生統計資料並不會終止進行中的統計資料計算。

如果您提出要求停用讀取器執行個體的自動產生功能，而不是資料庫叢集的寫入器執行個體，則要求會失敗，並顯示 HTTP 傳回碼 400，輸出如下所示：

```
{
  "detailedMessage" : "Writes are not permitted on a read replica instance",
  "code" : "ReadOnlyViolationException",
  "requestId":"8eb8d3e5-0996-4a1b-616a-74e0ec32d5f7"
}
```

常見錯誤 (p. 136)如需其他常見錯誤的清單，請參閱。

重新啟用自動產生 DFE 統計資料

依預設，當您啟用 DFE 時，已啟用自動產生 DFE 統計資料。如果您停用自動產生功能，您可以稍後重新啟用它，如下所示：

```
curl -X POST "$STATISTICS_ENDPOINT" -d '{ "mode" : "enableAutoCompute" }'
```

如果請求成功，則 HTTP 響應代碼為200，響應為：

```
{
  "status" : "200 OK"
}
```

您可以發出狀態請求 (p. 131)並檢查回應中的autoCompute欄位是否設定為，以確認已啟用自動產生true。

手動觸發 DFE 統計資料的產生

您可以手動啟動 DFE 統計資料產生，如下所示：

```
curl -X POST "$STATISTICS_ENDPOINT" -d '{ "mode" : "refresh" }'
```

如果請求成功，則輸出如下，HTTP 返回代碼為 200：

```
{
  "status" : "200 OK",
  "payload" : {
    "statisticsId" : 1588893232718
  }
}
```

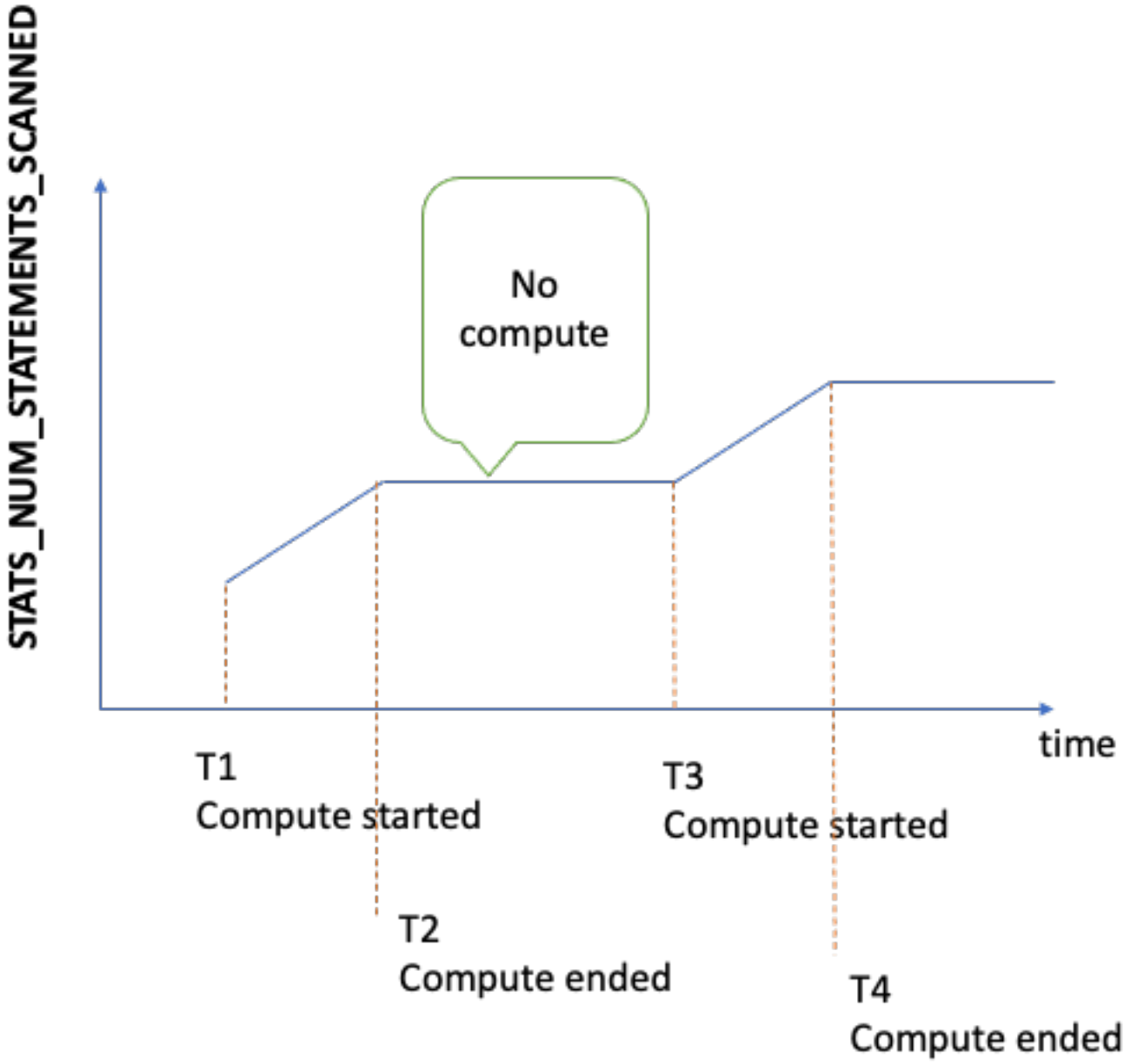
輸出`statisticsId`中的是目前正在執行的統計資料產生執行的識別碼。如果執行在要求時已經處理中，則要求會傳回該執行的 ID，而不是啟動新的執行 ID。一次只能執行一個統計資料產生。

如果在產生 DFE 統計資料時發生容錯移轉，則新寫入器節點會挑選上次處理的檢查點，並從該處繼續執行統計資料。

使用`StatsNumStatementsScanned`CloudWatch測量結果監督統計資料運算

此`StatsNumStatementsScanned`CloudWatch測量結果會傳回伺服器啟動後針對統計資料運算而掃描的敘述句總數。它在每個統計計算切片更新。

每次觸發統計計算時，這個數字都會增加，並且當沒有計算發生時，它保持不變。隨著時間的推移，查看`StatsNumStatementsScanned`值的情節，因此可以清楚地了解統計計算何時發生以及有多快：



進行計算時，圖形的斜率會顯示您的速度 (坡度越陡，計算的統計資料越快)。

如果圖表只是 0 的平線，表示統計功能已啟用，但根本沒有計算任何統計資料。如果統計資料功能已停用，或者您使用的引擎版本不支援統計資料運算，則`StatsNumStatementsScanned`不存在。

如前所述，您可以使用統計資料 API 停用統計資料運算，但是將其關閉可能會導致統計資料不是最新狀態，進而導致 DFE 引擎產生不良的查詢計畫。

如需有使用 Amazon 監控海王星 CloudWatch (p. 772)關如何使用的資訊，請參閱CloudWatch。

將 AWS Identity and Access Management (IAM) 身份驗證與 DFE 統計資料端點搭配使用

您可以使用 awscurl 或任何其他與 HTTPS 和 IAM 搭配使用的工具，透過 IAM 身份驗證安全地存取 DFE 統計資料端點。請參awscurl搭配臨時登入資料使用，安全地連線至啟用 IAM 身份驗證的資料庫叢集 (p. 162)閱以瞭解如何設定適當的認證。完成此操作後，您可以發出如下狀態請求：

```
awscurl "$STATISTICS_ENDPOINT" \
    --region (your region) \
    --service neptune-db
```

或者，例如，您可以建立名為的 JSON 檔案，其中包request.json含：

```
{ "mode" : "refresh" }
```

然後，您可以像這樣手動啟動統計信息生成：

```
awscurl "$STATISTICS_ENDPOINT" \
    --region (your region) \
    --service neptune-db \
    -X POST -d @request.json
```

刪除 DFE 統計資料

您可以向統計資料端點發出 HTTP DELETE 要求，以刪除資料庫中的所有統計資料：

```
curl -X "DELETE" "$STATISTICS_ENDPOINT"
```

有效的 HTTP 傳回代碼為：

- 200— 刪除成功。

  在這種情況下，典型的響應如下所示：

  ```
  {
    "status" : "200 OK",
    "payload" : {
        "active" : false,
        "statisticsId" : -1
    }
  }
  ```

- 204— 沒有要刪除的統計資料。

  在這種情況下，響應是空白的（沒有響應）。

如果您傳送刪除要求至讀取器節點上的統計資料端點，ReadOnlyViolationException就會擲回 a。

DFE 統計資料要求的常見錯誤碼

以下是當您向統計資料端點提出要求時可能發生的常見錯誤清單：

- AccessDeniedException-返回代碼：400。消息：Missing Authentication Token。

- BadRequestException (對於小精靈和開放密碼) -返回代碼：。 400消息：Bad route: /pg/statistics。
- BadRequestException(適用於 RDF 資料) — 傳回代碼：400。消息：Bad route: /rdf/statistics。
- InvalidParameterException-返回代碼：400。消息：Statistics command parameter 'mode' has unsupported value '*the invalid value*'。
- MissingParameterException-返回代碼：400。消息：Content-type header not specified.。
- ReadOnlyViolationException-返回代碼：400。消息：Writes are not permitted on a read replica instance。

例如，如果您在未啟用 DFE 和統計資料時提出要求，您會得到如下所示的回應：

```
{
  "code" : "BadRequestException",
  "requestId" : "b2b8f8ee-18f1-e164-49ea-836381a3e174",
  "detailedMessage" : "Bad route: /sparql/statistics"
}
```

獲取有關圖表的快速摘要報告

海王星圖形摘要 API 會擷取下列有關圖表的資訊：

- 對於屬性 (PG) 圖形，圖形摘要 API 會傳回節點和邊緣標籤以及屬性索引鍵的唯讀清單，以及節點、邊緣和屬性計數。
- 對於資源描述架構 (RDF) 圖形，圖形摘要 API 會傳回類別和述詞索引鍵的唯讀清單，以及四邊形、主旨和述詞的計數。

> **Note**
>
> 圖形摘要 API 是在海王星引擎版本 1.2.1.0 (p. 861) 中引入的。

使用圖形摘要 API，您可以快速了解圖形資料大小和內容。您也可以使用海王星工作台魔術，在海王星筆記本中以互動方式使用 API。%summary (p. 41)在圖形應用程式中，API 可以用來改善搜尋結果，方法是提供探索到的節點或邊緣標籤做為搜尋的一部分。

圖形摘要資料是從 Neptune DFE 引擎在執行階段所計 (p. 131)算的 DFE (p. 129) 統計資料中擷取，只要有 DFE 統計資料可用，即可使用。當您建立新的 Neptune 資料庫叢集時，預設會啟用統計資料。

您可以使用統計資料狀態 API 來檢查 DFE 統計資料的狀態 (p. 131)。只要沒有停用 (p. 133)自動產生統計資料，就會定期自動更新統計資料。

如果您想要在要求圖形摘要時確定統計資料盡可能是最新的，可以在擷取摘要之前，手動觸發統計資料更新 (p. 134)。如果圖形在計算統計數據時發生變化，則它們一定會稍微落後，但不會太大。

使用圖形摘要 API 擷取圖形摘要資訊

對於您使用 Grimlin 或 OpenCypher 進行查詢的屬性圖形，您可以從屬性圖摘要端點擷取圖形摘要。此端點既有長又短的 URI：

- `https://`*`your-neptune-host`*`:`*`port`*`/propertygraph/statistics/summary`
- `https://`*`your-neptune-host`*`:`*`port`*`/pg/statistics/summary`

對於您使用 SPARQL 查詢的 RDF 圖形，您可以從 RDF 摘要端點擷取圖形摘要：

- `https://`*`your-neptune-host`*`:`*`port`*`/rdf/statistics/summary`

這些端點為唯讀端點，且僅支援 HTTP GET 作業。如果 $ GRAPH_SUMPY_ENDPOINT 設置為要查詢的任何端點的地址，則可以使用和 HTTP 檢索摘要數據，如下所示：curlGET

```
curl -G "$GRAPH_SUMMARY_ENDPOINT"
```

如果在嘗試擷取圖形摘要時沒有可用的統計資料，回應如下所示：

```
{
  "detailedMessage": "Statistics are not available. Summary can only be generated after
 statistics are available.",
  "requestId": "48c1f788-f80b-b69c-d728-3f6df579a5f6",
  "code": "StatisticsNotAvailableException"
}
```

圖形摘要 API 的 mode URL 查詢參數

圖形摘要 API 接受名為的 URL 查詢參數mode，該參數可以採用兩個值之一，即basic（默認值）和detailed。對於 RDF 圖，detailed模式圖摘要響應包含一個附加subjectStructures字段。對於屬性圖，詳細的圖形摘要響應包含兩個附加字段，即nodeStructures和edgeStructures。

若要要求圖detailed形摘要回應，請包含mode參數，如下所示：

```
curl -G "$GRAPH_SUMMARY_ENDPOINT?mode=detailed"
```

如果mode參數不存在，則默認使用 basic mode，因此儘管可以?mode=basic明確指定，但這不是必需的。

屬性圖 (PG) 的圖形摘要回應

對於空的屬性圖，詳細的圖形摘要回應如下所示：

```
{
  "status" : "200 OK",
  "payload" : {
    "version" : "v1",
    "lastStatisticsComputationTime" : "2023-01-10T07:58:47.972Z",
    "graphSummary" : {
      "numNodes" : 0,
      "numEdges" : 0,
      "numNodeLabels" : 0,
      "numEdgeLabels" : 0,
      "nodeLabels" : [ ],
      "edgeLabels" : [ ],
      "numNodeProperties" : 0,
      "numEdgeProperties" : 0,
      "nodeProperties" : [ ],
      "edgeProperties" : [ ],
      "totalNodePropertyValues" : 0,
      "totalEdgePropertyValues" : 0,
      "nodeStructures" : [ ],
      "edgeStructures" : [ ]
    }
  }
}
```

屬性圖表 (PG) 摘要回應包含下列欄位：

- **status**— 要求的 HTTP 傳回程式碼。如果請求成功，則代碼為 200。

  常見的圖形摘要錯誤 (p. 146)如需常見錯誤的清單，請參閱。
- **payload**
  - **version**— 此圖形摘要回應的版本。
  - **lastStatisticsComputationTime** — Neptune 上次計算統計資料 (p. 131)的時間戳記 (ISO 8601 格式)。
  - **graphSummary**
    - **numNodes**— 圖形中的節點數目。
    - **numEdges**— 圖形中的邊數。
    - **numNodeLabels**— 圖形中不同節點標籤的數目。
    - **numEdgeLabels**— 圖形中不同邊緣標籤的數量。
    - **nodeLabels**— 圖形中不同節點標籤的清單。

- **edgeLabels**-圖形中不同邊緣標籤的清單。
- **numNodeProperties**— 圖形中不同節點屬性的數目。
- **numEdgeProperties**— 圖形中不同邊屬性的數目。
- **nodeProperties**— 圖形中不同節點屬性的清單，以及使用每個屬性的節點計數。
- **edgeProperties**— 圖形中不同邊緣屬性的清單，以及使用每個屬性的邊緣計數。
- **totalNodePropertyValues**— 所有節點屬性的使用總數。
- **totalEdgePropertyValues**— 所有邊屬性的使用總數。
- **nodeStructures**— 只有在請求中指定時，*mode=detailed*才會顯示此欄位。它包含節點結構的清單，每個結構都包含下列欄位：
  - **count**— 具有此特定結構的節點數目。
  - **nodeProperties**-存在於此特定結構中的節點屬性列表。
  - **distinctOutgoingEdgeLabels**-存在於此特定結構中的不同傳出邊標籤的清單。
- **edgeStructures**— 只有在請求中指定時，*mode=detailed*才會顯示此欄位。它包含邊結構的清單，每個邊結構都包含下列欄位：
  - **count**— 具有此特定結構的邊數。
  - **edgeProperties**-存在於此特定結構中的邊屬性清單。

RDF 圖形的圖形摘要響應

對於空的 RDF 圖，詳細的圖形摘要響應如下所示：

```
{
  "status" : "200 OK",
  "payload" : {
    "version" : "v1",
    "lastStatisticsComputationTime" : "2023-01-10T07:58:47.972Z",
    "graphSummary" : {
      "numDistinctSubjects" : 0,
      "numDistinctPredicates" : 0,
      "numQuads" : 0,
      "numClasses" : 0,
      "classes" : [ ],
      "predicates" : [ ],
      "subjectStructures" : [ ]
    }
  }
}
```

RDF 圖形摘要響應具有以下字段：

- **status**— 要求的 HTTP 傳回程式碼。如果請求成功，則代碼為 200。

  常見的圖形摘要錯誤 (p. 146)如需常見錯誤的清單，請參閱。
- **payload**
  - **version**— 此圖形摘要回應的版本。
  - **lastStatisticsComputationTime** — Neptune 上次計算統計資料 (p. 131)的時間戳記 (ISO 8601 格式)。
  - **graphSummary**
    - **numDistinctSubjects**— 圖表中不同主題的數量。
    - **numDistinctPredicates**— 圖形中不同述詞的數目。
    - **numQuads**— 圖表中的四邊形數量。
    - **numClasses**— 圖形中的類別數目。

- **classes**— 圖形中的類別清單。
- **predicates**— 圖形中述詞的清單，以及述詞計數。
- **subjectStructures**— 只有在請求中指定時，*mode=detailed*才會顯示此欄位。它包含主旨結構清單，每個結構都包含下列欄位：
  - **count**— 此特定結構的出現次數。
  - **predicates**— 此特定結構中存在的述詞清單。

屬性圖 (PG) 摘要回應範例

以下是屬性圖表的詳細摘要回應，其中包含範例屬性圖航空路線資料集：


```
{
  "status" : "200 OK",
  "payload" : {
    "version" : "v1",
    "lastStatisticsComputationTime" : "2023-03-01T14:35:03.804Z",
    "graphSummary" : {
      "numNodes" : 3748,
      "numEdges" : 51300,
      "numNodeLabels" : 4,
      "numEdgeLabels" : 2,
      "nodeLabels" : [
        "continent",
        "country",
        "version",
        "airport"
      ],
      "edgeLabels" : [
        "contains",
        "route"
      ],
      "numNodeProperties" : 14,
      "numEdgeProperties" : 1,
      "nodeProperties" : [
        {
          "desc" : 3748
        },
        {
          "code" : 3748
        },
        {
          "type" : 3748
        },
        {
          "country" : 3503
        },
        {
          "longest" : 3503
        },
        {
          "city" : 3503
        },
        {
          "lon" : 3503
        },
        {
          "elev" : 3503
        },
        {
          "icao" : 3503
        },
```

```
      {
        "region" : 3503
      },
      {
        "runways" : 3503
      },
      {
        "lat" : 3503
      },
      {
        "date" : 1
      },
      {
        "author" : 1
      }
    ],
    "edgeProperties" : [
      {
        "dist" : 50532
      }
    ],
    "totalNodePropertyValues" : 42773,
    "totalEdgePropertyValues" : 50532,
    "nodeStructures" : [
      {
        "count" : 3471,
        "nodeProperties" : [
          "city",
          "code",
          "country",
          "desc",
          "elev",
          "icao",
          "lat",
          "lon",
          "longest",
          "region",
          "runways",
          "type"
        ],
        "distinctOutgoingEdgeLabels" : [
          "route"
        ]
      },
      {
        "count" : 161,
        "nodeProperties" : [
          "code",
          "desc",
          "type"
        ],
        "distinctOutgoingEdgeLabels" : [
          "contains"
        ]
      },
      {
        "count" : 83,
        "nodeProperties" : [
          "code",
          "desc",
          "type"
        ],
        "distinctOutgoingEdgeLabels" : [ ]
      },
      {
        "count" : 32,
```

```
          "nodeProperties" : [
            "city",
            "code",
            "country",
            "desc",
            "elev",
            "icao",
            "lat",
            "lon",
            "longest",
            "region",
            "runways",
            "type"
          ],
          "distinctOutgoingEdgeLabels" : [ ]
        },
        {
          "count" : 1,
          "nodeProperties" : [
            "author",
            "code",
            "date",
            "desc",
            "type"
          ],
          "distinctOutgoingEdgeLabels" : [ ]
        }
      ],
      "edgeStructures" : [
        {
          "count" : 50532,
          "edgeProperties" : [
            "dist"
          ]
        }
      ]
    }
  }
}
```

RDF 圖形摘要回應範例

以下是 RDF 圖表的詳細摘要回應，其中包含 RDF 航空路線範例資料集：

```
{
  "status" : "200 OK",
  "payload" : {
    "version" : "v1",
    "lastStatisticsComputationTime" : "2023-03-01T14:54:13.903Z",
    "graphSummary" : {
      "numDistinctSubjects" : 54403,
      "numDistinctPredicates" : 19,
      "numQuads" : 158571,
      "numClasses" : 4,
      "classes" : [
        "http://kelvinlawrence.net/air-routes/class/Version",
        "http://kelvinlawrence.net/air-routes/class/Airport",
        "http://kelvinlawrence.net/air-routes/class/Continent",
        "http://kelvinlawrence.net/air-routes/class/Country"
      ],
      "predicates" : [
        {
          "http://kelvinlawrence.net/air-routes/objectProperty/route" : 50656
```

```
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/dist" : 50656
    },
    {
      "http://kelvinlawrence.net/air-routes/objectProperty/contains" : 7004
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/code" : 3747
    },
    {
      "http://www.w3.org/2000/01/rdf-schema#label" : 3747
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/type" : 3747
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/desc" : 3747
    },
    {
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#type" : 3747
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/icao" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/lat" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/region" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/runways" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/longest" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/elev" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/lon" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/country" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/city" : 3502
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/author" : 1
    },
    {
      "http://kelvinlawrence.net/air-routes/datatypeProperty/date" : 1
    }
  ],
  "subjectStructures" : [
    {
      "count" : 50656,
      "predicates" : [
        "http://kelvinlawrence.net/air-routes/datatypeProperty/dist"
      ]
    },
    {
      "count" : 3471,
      "predicates" : [
```

```
        "http://kelvinlawrence.net/air-routes/datatypeProperty/city",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/code",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/country",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/desc",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/elev",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/icao",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/lat",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/lon",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/longest",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/region",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/runways",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/type",
        "http://kelvinlawrence.net/air-routes/objectProperty/route",
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
        "http://www.w3.org/2000/01/rdf-schema#label"
      ]
    },
    {
      "count" : 238,
      "predicates" : [
        "http://kelvinlawrence.net/air-routes/datatypeProperty/code",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/desc",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/type",
        "http://kelvinlawrence.net/air-routes/objectProperty/contains",
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
        "http://www.w3.org/2000/01/rdf-schema#label"
      ]
    },
    {
      "count" : 31,
      "predicates" : [
        "http://kelvinlawrence.net/air-routes/datatypeProperty/city",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/code",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/country",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/desc",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/elev",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/icao",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/lat",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/lon",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/longest",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/region",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/runways",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/type",
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
        "http://www.w3.org/2000/01/rdf-schema#label"
      ]
    },
    {
      "count" : 6,
      "predicates" : [
        "http://kelvinlawrence.net/air-routes/datatypeProperty/code",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/desc",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/type",
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
        "http://www.w3.org/2000/01/rdf-schema#label"
      ]
    },
    {
      "count" : 1,
      "predicates" : [
        "http://kelvinlawrence.net/air-routes/datatypeProperty/author",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/code",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/date",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/desc",
        "http://kelvinlawrence.net/air-routes/datatypeProperty/type",
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
```

```
            "http://www.w3.org/2000/01/rdf-schema#label"
          ]
        }
      ]
    }
  }
}
```

搭配圖形摘要端點使用 AWS Identity and Access Management (IAM) 身份驗證

您可以使用 awscurl 或任何其他與 HTTPS 和 IAM 搭配使用的工具，透過 IAM 身份驗證安全地存取圖形摘要端點。請參awscurl搭配臨時登入資料使用，安全地連線至啟用 IAM 身份驗證的資料庫叢集 (p. 162)閱以瞭解如何設定適當的認證。完成此操作後，您可以發出這樣的請求：

```
awscurl "$GRAPH_SUMMARY_ENDPOINT" \
    --region (your region) \
    --service neptune-db
```

> **Important**
>
> 建立臨時登入資料的 IAM 身分或角色必須附加允許 IAM 動作的 GetGraphSummary (p. 206)IAM 政策。

如IAM 身分驗證錯誤 (p. 852)需您可能遇到的常見 IAM 錯誤清單，請參閱。

圖形摘要要求可能傳回的常見錯誤代碼

海王星服務錯誤代碼	HTTP 狀態	訊息	錯誤情境	缓解
AccessDeniedException	403	缺少身分驗證字符。	未簽署或未正確簽署的要求已傳送至 Neptune 資料庫，且啟用 IAM。	在發送之前使用 Sigv4 簽署請求（請參閱IAM 和圖表摘要 (p. 146)）。
	403	用戶：*### ARN#*未被授權執行：海王星數據庫：資源：*#GetGraphSummary* 資源 ARN）。	IAM 政策不允許在啟用 IAM 的情GetGraphSummary (p. 206)況下將圖形摘要請求傳送至 Neptune 資料庫時執行此動作。	確保附加到發出請求的使用者或角色的 IAM 政策允許GetGraphSummary執行動作。
BadRequestException	400	統計資料已停用，因此也會停用圖形摘要。	嘗試擷取已停用統計資料的高載執行個體類型 (t3或t4g) 摘要。	使用已啟用統計資料產生的執行個體類型 (除了t3和以外的所有支援執行個體t4g)。
	400	錯誤的路線：*/rdf /# #/#*要	請求發送到無效路徑。	針對圖形摘要端點使用正確的路由。
InvalidParameterException	400	請求包含未知參數：'*#########*'。	在請求中指定了無效的參數時。	請求中僅使用有效參數 (例如mode)。

海王星服務錯誤代碼	HTTP 狀態	訊息	錯誤情境	緩解
InvalidParameterException	400	URI 查詢參數 'mode' 具有不支援的值 '*(###)*'。	當請求中的 URL 參數「模式」後跟一個無效的值。	指定 URL 參數「模式」時，請使用有效值 (例如`basic`或`detailed`)。
MethodNotAllowedException	405	方法不允許。	使用 GET (例如`POST`或`DELETE`) 以外的任何 HTTP 方法呼叫摘要端點。	呼叫摘要端點時，請使用 HTTP GET 方法。
StatisticsNotAvailableException	400	統計資料尚未計算，統計資料計算完成後將可使用圖形摘要。	當要求傳送至摘要端點時，沒有可用的統計資料。	等到統計信息生成完成。您可以使用統計狀態 API 檢查統計資料產生的狀態 (p. 131)。
	400	已達統計資料限制，因此無法使用圖形摘要。	統計資料產生已停止，因為它達到統計資料大小限制 (p. 131)	此圖表中沒有圖表摘要。

例如，如果您在啟用 IAM 身份驗證的 Neptune 資料庫中提出圖形化摘要端點的請求，且要求者的 IAM 政策中沒有必要的許可，則您會得到如下所示的回應：

```
{
  "detailedMessage": "User: arn:aws:iam::(account ID):(user or user name) is not authorized
 to perform: neptune-db:GetGraphSummary on resource: arn:aws:neptune-db:(region):(account
 ID):(cluster resource ID)/*",
  "requestId": "7ac2b98e-b626-d239-1d05-74b4c88fce82",
  "code": "AccessDeniedException"
}
```

亞馬遜海王星 JDBC 連接

亞馬遜海王星已經發布了一個開源 JDBC 驅動程序，支持開放密碼，小鬼，SQL 格林和 SPARQL 查詢。JDBC 連線能力可讓您透過商業智慧 (BI) 工具 (例如 Tableau) 輕鬆連線至海王星。搭配 Neptune 使用 JDBC 驅動程式無需額外費用 — 您仍然只需為耗用的 Neptune 資源付費。

該驅動程序與 JDBC 4.2 兼容，並且至少需要 Java 8。如需如何使用 JDBC 驅動程式的相關資訊，請參閱 JDBC API 文件。

該GitHub項目，您可以在其中提交問題和打開功能請求，包含驅動程序的詳細文檔：

亞馬遜海王星的 JDBC 驅動程序

- 搭配 JDBC 驅動程式使用 SQL
- 在 JDBC 驅動程序中使用小鬼
- 搭配 JDBC 驅動程式使用開放密碼
- 搭配 JDBC 驅動程式使用

開始使用海王星 JDBC 驅動程式

若要使用 Neptune JDBC 驅動程式連接到 Neptune 執行個體，JDBC 驅動程式必須部署在與 Neptune 資料庫叢集相同 VPC 中的 Amazon EC2 執行個體上，或者執行個體必須透過安全殼層通道或負載平衡器提供使用。SSH 通道可以在驅動程序內部設置，也可以在外部設置。

您可以在此處下載驅動程序。該驅動程序打包為單個 JAR 文件，名稱類似`neptune-jdbc-1.0.0-all.jar`。若要使用它，請將 JAR 檔案放在應`classpath`用程式中。或者，如果您的應用程序使用 Maven 或搖籃，則可以使用適當的 Maven 或搖籃命令從 JAR 安裝驅動程序。

驅動程序需要一個 JDBC 連接 URL 來與海王星連接，形式如下：

```
jdbc:neptune:(connection type)://(host);property=value;property=value;...;property=value
```

GitHub專案中每個查詢語言的區段會說明您可以在該查詢語言的 JDBC 連線 URL 中設定的屬性。

如果 JAR 文件位於應用程序中`classpath`，則不需要其他配置。您可以使用 JDBC 接`DriverManager`口和海王星連接字符串連接驅動程序。例如，如果您的海王星資料庫叢集可以透過連接埠 8182 `neptune-example.com` 上的端點存取，您就可以像這樣與 OpenCypher 連線：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

void example() {
    String url = "jdbc:neptune:opencypher://bolt://neptune-example:8182";

    Connection connection = DriverManager.getConnection(url);
    Statement statement = connection.createStatement();

    connection.close();
}
```

GitHub專案中每種查詢語言的文件章節說明如何在使用該查詢語言時建構連接字串。

搭配海王星 JDBC 驅動程式使用畫面

若要將 Tableau 與海王星 JDBC 驅動程式搭配使用，請先下載並安裝最新版本的 Tableau 桌面版本。下載海王星 JDBC 驅動程式的 JAR 檔案，以及海王星 Tableau 連接器檔案（ .taco檔案 ）。

在 Mac 上連線到適用於海王星的 Tableau 的步驟

1. 將海王星 JDBC 驅動程式 JAR 檔案放在/Users/*(your user name)*/Library/Tableau/Drivers資料夾中。
2. 將海王星 Tableau 連接器.taco檔案放在資/Users/*(your user name)*/Documents/My Tableau Repository/Connectors料夾中。
3. 如果您已啟用 IAM 身份驗證，請為其設置環境。請注意.zprofile/，在、.zshenv/.bash_profile、等中設定的環境變數將不起作用。必須設置環境變量，以便它們可以由 GUI 應用程序加載。

   設置憑據的一種方法是將您的訪問密鑰和密鑰放在/Users/*(your user name)*/.aws/credentials文件中。

   設定服務區域的簡單方法是開啟終端機，並使用應用程式的區域輸入下列指令 (例如us-east-1)：

   ```
   launchctl setenv SERVICE_REGION region name
   ```

   還有其他方法可以設置在重新啟動後持續存在的環境變量，但是無論您使用哪種技術，都必須設置 GUI 應用程序可以訪問的變量。
4. 要在 Mac 上將環境變量加載到 GUI 中，請在終端機上輸入以下命令：

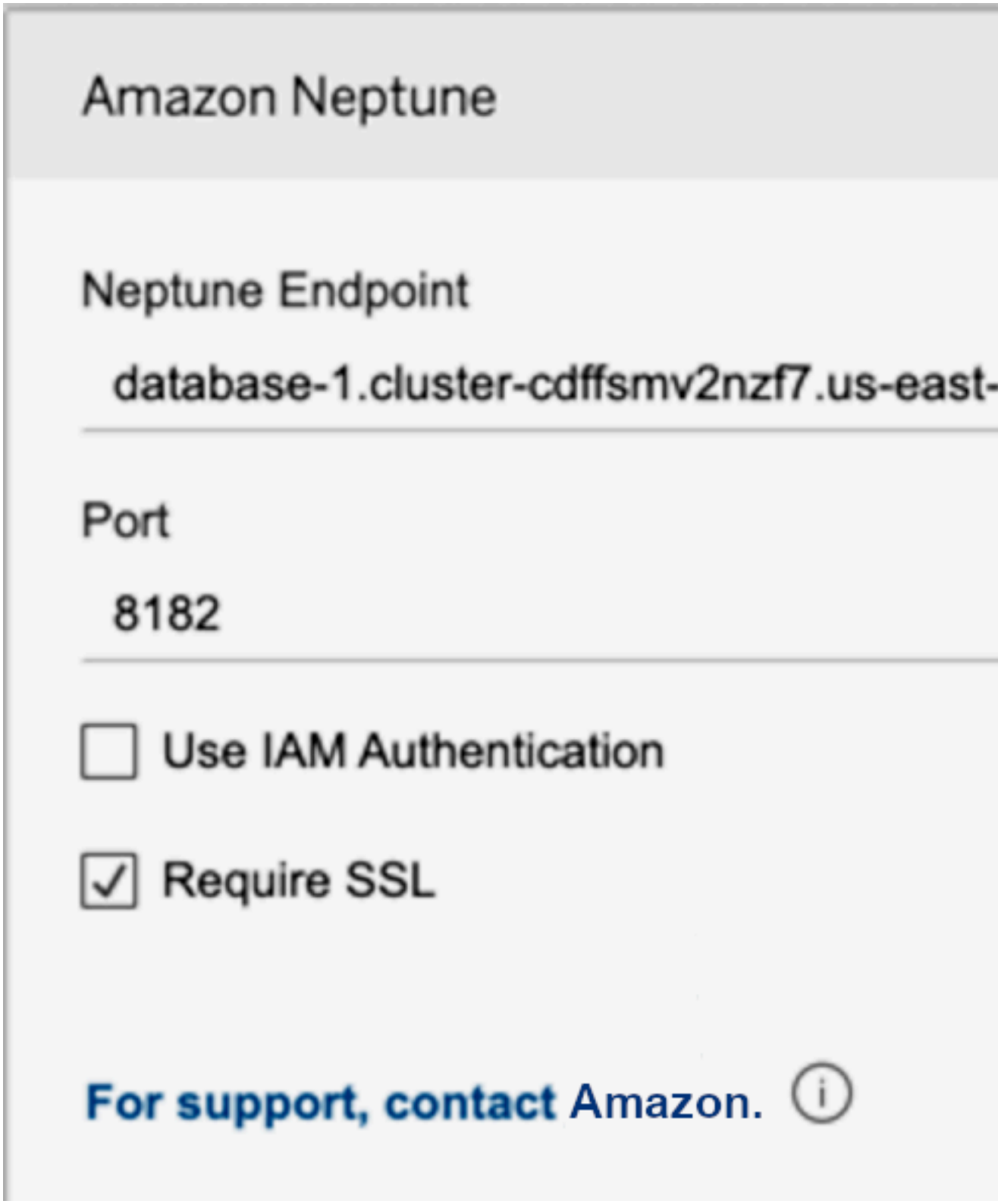
   ```
   /Applications/Tableau/Desktop/2021.1.app/Contents/MacOS/Tableau
   ```

在視窗電腦上連線到適用於海王星的 Tableau 的步驟

1. 將海王星 JDBC 驅動程式 JAR 檔案放在C:\Program Files\Tableau\Drivers資料夾中。
2. 將海王星 Tableau 連接器.taco檔案放在資C:\Users*(your user name)*\Documents\My Tableau Repository\Connectors料夾中。
3. 如果您已啟用 IAM 身份驗證，請為其設置環境。

   這可以像設定使用者ACCESS_KEYSECRET_KEY、和SERVICE_REGION環境變數一樣簡單。

在 Tableau 開啟的情況下，選取視窗左側的「更多」。如果 Tableau 連接器檔案位置正確，您可以在出現的清單AWS中選取 Amazon 海王星的方式：



您不需要編輯連接埠，也不需要新增任何連線選項。輸入您的海王星端點並設定 IAM 和 SSL 設定 (如果您使用的是 IAM，則必須啟用 SSL)。

當您選取 [登入] 時，如果圖表很大，連線可能需要 30 秒以上的時間。Tableau 正在收集頂點和邊緣表，並在邊緣連接頂點，以及創建可視化。

疑難排解 JDBC 驅動程式連線

如果驅動程序無法連接到服務器，請使用 JDBC Connection 對象的isValid功能來檢查連接是否有效。如果函數傳回 false (表示連線無效)，請檢查連線的端點是否正確，以及您位於 Neptune DB 叢集的 VPC 中，或者您是否擁有與叢集有效的 SSH 通道。

如果您收到來自DriverManager.getConnection呼叫的`No suitable driver found for (connection string)`回應，則可能在連接字串的開頭發生問題。確保您的連接字符串像這樣開始：

```
jdbc:neptune:opencypher://...
```

要收集有關連接的更多信息，可以像這樣LogLevel向連接字符串添加一個：

```
jdbc:neptune:opencypher://(JDBC URL):(port);logLevel=trace
```

或者，您也可以新增輸properties.put("logLevel", "trace")入內容以記錄追蹤資訊。

亞馬遜海神引擎更新

亞馬遜海王星定期發行引擎更新。您可以使用執行個體狀態 API (p. 770) 來判斷您目前安裝的引擎發行版本。

引擎版本列在 Amazon Neptune 的引擎版本 (p. 856)，而修補程式列在最新更新 (p. 3)。

您可以在以下位置找到有關如何發行更新以及如何升級資料庫中 Neptune 引擎的詳細資訊海王星引擎更新 (p. 857)。例如，引擎版本編號將在中說明海王星引擎版本編號 (p. 859)。

亞馬遜海王星的安全

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。

安全是 AWS 與您共同的責任。共同的責任模型 將此描述為雲端 本身 的安全和雲端內部的安全：

- 雲端本身的安全：AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 AWS 合規計劃中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon Neptune 的合規計劃，請參閱合規計劃範圍內的AWS服務。
- 雲端內部的安全：您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您瞭解如何在使用 Neptune 時套用共同的責任模型。下列主題說明如何設定 Neptune 以符合安全性和合規性目標。您也會學到如何使用其他可AWS協助您監控和保護 Neptune 資源的服務。

主題

- 亞馬遜海王星的資料保護 (p. 153)
- 亞馬遜海王星AWS Identity and Access Management（IAM）概述 (p. 157)
- 在海王星中啟用 IAM 資料庫身分 (p. 160)
- 使用簽AWS名版本 4 連接和簽名 (p. 160)
- 使用 IAM 政策管理存取 (p. 176)
- 針對 Neptune 使用服務連結角色 (p. 215)
- 使用暫時登入資料的 IAM 身分驗證 (p. 218)
- 記錄和監控亞馬遜海王星資源 (p. 223)
- 亞馬遜海王星的合規驗證 (p. 223)
- 亞馬遜海王星的彈性 (p. 224)

亞馬遜海王星的資料保護

AWS共同責任模型適用於 Amazon Neptune 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您必須負責維護在此基礎設施上託管之內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center (successor to AWS Single Sign-On) 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 憑證，並設定個別使用者。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶都使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務 內的所有預設安全控制項。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的詳細資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name (名稱) 欄位。這包括當您AWS 服務使用主控台、API 或 AWS SDK 與 Neptune 或其他AWS CLI人合作時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

您可以使用AWS已發佈的 API 呼叫透過網路管理海王星。用戶端必須使用強式加密套件支援傳輸層安全性 (TLS) 1.2 或更新版本，如中傳輸中加密 (p. 154)所述。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

以下各節將進一步說明 Neptune 資料的保護方式。

主題

- 每個亞馬遜海王星資料庫叢集都駐留在亞馬遜 VPC 中 (p. 154)
- 傳輸中加密：使用 SSL/HTTPS 連接到海王星 (p. 154)
- 在靜態時加密海王星資源 (p. 155)

每個亞馬遜海王星資料庫叢集都駐留在亞馬遜 VPC 中

Amazon Neptune 資料庫叢集只能在 Amazon 虛擬私有雲端 (Amazon VPC) 中建立，而且只能在該 VPC 內存取其端點，通常可從該 VPC 中執行的 Amazon 彈性運算雲端 (Amazon EC2) 執行個體存取。

您可以限制對 Neptune 資料庫叢集所在 VPC 的存取，以保護 Neptune 資料的安全，如中連接到亞馬遜海王星圖 (p. 63)所述。

傳輸中加密：使用 SSL/HTTPS 連接到海王星

從引擎版本 1.0.4.0 (p. 950) 開始，Amazon Neptune 只允許透過 HTTPS 連線到任何執行個體或叢集端點的安全通訊端層 (SSL)。

海王星需要使用以下強式密碼套件的 TLS 1.2 版：

- `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
- `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`
- `TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA`
- `TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA`

即使在舊版引擎中允許 HTTP 連線的情況下，任何使用新資料庫叢集參數群組的資料庫叢集依預設都必須使用 SSL。為了保護您的資料，引擎版本*1.0.4.0*及以上版本的 Neptune 端點僅支援 HTTPS 要求。如需詳細資訊，請參閱 使用 HTTP REST 端點連線至海王星資料庫執行個體 (p. 429)。

Neptune 會自動為海王星資料庫執行個體提供 SSL 憑證。您不需要請求任何憑證。憑證會在您建立新執行個體時提供。

Neptune 會為每個區域指派單一萬用字元 SSL 憑證給您帳戶中的執AWS行個體。此憑證為叢集端點、叢集唯讀端點、執行個體端點提供項目。

憑證詳細資訊

提供的憑證包含以下項目：

- 叢集端點 — `*.cluster-`*`a1b2c3d4wxyz.region`*`.neptune.amazonaws.com`

- 唯讀端點 — `*.cluster-ro-a1b2c3d4wxyz.region.neptune.amazonaws.com`
- 執行個體端點 — `*.a1b2c3d4wxyz.region.neptune.amazonaws.com`

只支援此處所列的項目。

代理連線

憑證只支援上一節所列的主機名稱。

如果您使用負載平衡器或代理伺服器 (例如 HAProxy)，您必須在代理伺服器上使用 SSL 終止，並有您自己的 SSL 憑證。

SSL 傳遞沒有作用，因為提供的 SSL 憑證與代理伺服器主機名稱不符。

根 CA 憑證

Neptune 執行個體的憑證通常會使用作業系統或 SDK (例如 Java SDK) 的本機信任存放區進行驗證。

如果您需要手動提供根憑證，可以從 Amazon 信任服務政策儲存庫下載 PEM 格式的 Amazon 根 CA 憑證。

詳細資訊

如需有關使用 SSL 連線到 Neptune 端點的詳細資訊，請參閱the section called “安裝小精靈控制台” (p. 301)和the section called “HTTP REST” (p. 429)。

在靜態時加密海王星資源

Neptune 加密執行個體可協助保護您的資料，防止未經授權存取基礎儲存體，進而提供額外的資料保護層。您可以使用 Neptune 加密來增加雲端中部署之應用程式的資料保護。您也可以使用它來滿足data-at-rest加密的合規要求。

若要管理用於加密和解密 Neptune 資源的金鑰，請使用 AWS Key Management Service() AWS KMS。AWS KMS結合安全、高可用性的硬體和軟體，提供專為雲端擴充的金鑰管理系統。您可以使用 AWS KMS 建立加密金鑰，並定義控制這些金鑰使用方式的政策。AWS KMS 支援 AWS CloudTrail，因此您可稽核金鑰使用情況，以確認金鑰的使用方式是否正確。您可以將AWS KMS金鑰與海王星和支援的AWS服務結合使用，例如亞馬遜簡單儲存服務 (Amazon S3)、亞馬遜彈性區塊存放區 (亞馬遜 EBS) 和亞馬遜 Redshift。如需支援的服務清單AWS KMS，請參閱AWS Key Management Service開發人員指南AWS KMS中的AWS服務使用方式。

Neptune 加密執行個體的所有記錄檔、備份和快照都會加密。

啟用 Neptune 資料庫執行個體的加密

若要啟用新 Neptune 資料庫執行個體的加密，請在 Neptune 主控台的 [啟用加密] 區段中選擇 [是]。如需建立 Neptune 資料庫執行個體的資訊，請參閱建立新的海王星資料庫叢集 (p. 54)。

建立加密的 Neptune 資料庫執行個體時，也可以提供加密AWS KMS金鑰的金鑰識別碼。如果您未指定AWS KMS金鑰識別碼，Neptune 會針對新的 Neptune 資料庫執行個體使用預設的 Amazon RDS 加密金鑰 (`aws/rds`)。 AWS KMS為您的AWS帳戶建立 Neptune 的預設加密金鑰。您的AWS帳戶每個AWS區域都有不同的預設加密金鑰。

建立加密的 Neptune 資料庫執行個體之後，就無法變更該執行個體的加密金鑰。因此，在建立加密的 Neptune 資料庫執行個體之前，請務必確定您的加密金鑰需求。

您可以使用來自另一個帳戶的金鑰的 Amazon 資源名稱 (ARN) 來加密 Neptune 資料庫執行個體。如果您建立 Neptune 資料庫執行個體的AWS帳戶與擁有用於AWS KMS加密該新 Neptune 資料庫執行個體的加密AWS KMS金鑰相同的帳戶，則您傳遞的金鑰識別碼可以是AWS KMS金鑰別名，而不是金鑰的 ARN。

Important

如果 Neptune 無法存取 Neptune 資料庫執行個體的加密金鑰 (例如，當 Neptune 對金鑰的存取權遭到撤銷時)，則加密的資料庫執行個體會置於終端機狀態，而且只能從備份還原。強烈建議您始終為加密的 NeptuneB 執行個體啟用備份，以防止資料庫中的加密資料遺失。

啟用加密時所需的金鑰權限

建立加密 Neptune 資料庫執行個體的 IAM 使用者或角色必須至少具有下列 KMS 金鑰的權限：

- "kms:Encrypt"
- "kms:Decrypt"
- "kms:GenerateDataKey"
- "kms:ReEncryptTo"
- "kms:GenerateDataKeyWithoutPlaintext"
- "kms:CreateGrant"
- "kms:ReEncryptFrom"
- "kms:DescribeKey"

以下是包含必要權限的金鑰原則範例：

```
{
  "Version": "2012-10-17",
  "Id": "key-consolepolicy-3",
  "Statement": [
    {
      "Sid": "Enable Permissions for root principal",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow use of the key for Neptune",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/NeptuneFullAccess"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:ReEncryptTo",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:CreateGrant",
        "kms:ReEncryptFrom",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "rds.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "Deny use of the key for non Neptune",
```

```
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/NeptuneFullAccess"
      },
      "Action": [
        "kms:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "kms:ViaService": "rds.us-east-1.amazonaws.com"
        }
      }
    }
  ]
}
```

- 此政策中的第一個聲明是可選的。它可以存取使用者的根主體。
- 第二個陳述式可讓您存取此角色的所有必要 AWS KMS API，範圍為 RDS 服務主體。
- 第三個陳述式強制執行此金鑰無法由此角色用於任何其他AWS服務，藉此加強安全性。

您還可以通過添加以下內容來進一步縮小createGrant權限：

```
"Condition": {
  "Bool": {
    "kms:GrantIsForAWSResource": true
  }
}
```

海王星加密的局限性

加密 Neptune 叢集時存在下列限制：

- 您無法將未加密的資料庫叢集轉換為加密的叢集。

  但是，您可以將未加密的 資料庫叢集快照還原至加密的 資料庫叢集。若要執行此作業，請在從未加密資料庫叢集快照還原時指定 KMS 加密金鑰。
- 您無法將未加密的資料庫執行個體轉換為加密的執行個體。您只能在建立資料庫執行個體時為其啟用加密。
- 此外，無法修改加密的資料庫執行個體以停用加密功能。
- 未加密資料庫執行個體不可以有加密僅供讀取複本，加密資料庫執行個體也不可以有未加密僅供讀取複本。
- 加密的讀取本必須使用與來源資料庫執行個體相同的索引鍵進行加密。

亞馬遜海王星AWS Identity and Access Management（IAM）概述

AWS Identity and Access Management (IAM) 是一種 AWS 服務，讓管理員能夠安全地控制對 AWS 資源的存取權限。IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 Neptune 資源。IAM 是一種您可以免費使用的 AWS 服務。

您可以使用 AWS Identity and Access Management (IAM) 向 Neptune 資料庫執行個體或資料庫叢集進行驗證。啟用 IAM 資料庫身份驗證後，每個請求都必須使用AWS簽名版本 4 簽署。

AWS簽名版本 4 將驗證信息添加到AWS請求中。為了安全起見，所有對已啟用 IAM 身份驗證的 Neptune DB 叢集的請求都必須使用存取金鑰簽署。此金鑰由存取金鑰 ID 和私密存取金鑰組成。身份驗證是使用 IAM 政策進行外部管理。

Neptune 會在連線上進行驗證，對於WebSockets連線，它會定期驗證權限，以確保使用者仍然具有存取權。

Note

- 不建議撤銷、刪除或輪換與 IAM 使用者相關聯的認證，因為它不會終止任何已開啟的連線。
- 每個資料庫執行個體的並行WebSocket連線數目有限制，以及連線可以保持開啟的時間長度。如需詳細資訊，請參閱WebSockets 限制 (p. 846)。

IAM 用法取決於您的角色

根據您在 Neptune 中所做的工作，使用方式 AWS Identity and Access Management (IAM) 會有所不同。

服務使用者 — 如果您使用 Neptune 服務執行工作，則系統管理員會為您提供使用 Neptune 資料平面所需的認證和權限。由於您需要更多存取權限來完成工作，因此瞭解資料存取的管理方式可協助您向管理員要求正確的權限。

服務管理員 — 如果您負責公司的 Neptune 資源，您可能可以存取 Neptune 管理動作，這些動作對應於 Neptune 管理 API (p. 996)。決定 Neptune 資料存取動作和資源服務使用者才能完成其工作，也可能是您的工作。然後 IAM 管理員可以套用 IAM 政策來變更服務使用者的許可。

IAM 管理員 — 如果您是 IAM 管理員，則需要撰寫 IAM 政策來管理 Neptune 的管理和資料存取。若要檢視您可以使用的 Neptune 身分型原則範例，請參閱。使用不同種類的 IAM 政策來控制對 Neptune 的存取 (p. 177)

使用身分來驗證

身分驗證是使用身分憑證登入 AWS 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者身分，或擔任 IAM 角色進行 驗證 (登入至 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分登入 AWS。AWS IAM Identity Center (successor to AWS Single Sign-On)(IAM Identity Center) 使用者、貴公司的單一登入身分驗證和您的 Google 或 Facebook 憑證都是聯合身分的範例。當您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您 AWS 藉由使用聯合進行存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入至 AWS 的詳細資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

如果您AWS以程式設計方式存取，請AWS提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，您必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來提高帳戶的安全。如需更多資訊，請參閱《AWS IAM Identity Center (successor to AWS Single Sign-On) 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的在 AWS 中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

如果是建立 AWS 帳戶，您會先有一個登入身分，可以完整存取帳戶中所有 AWS 服務 與資源。此身分稱為 AWS 帳戶 根使用者，使用建立帳戶時所使用的電子郵件地址和密碼即可登入並存取。強烈建議您不要以根使用者處理日常作業。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《AWS Account Management 參考指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者是您 AWS 帳戶 中的一種身分，具備單一人員或應用程式的特定許可。建議您盡可能依賴暫時性憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需詳細資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分登入。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的過程變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。若要進一步了解，請參閱《IAM 使用者指南》中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 中的一種身分，具備特定許可。它類似 IAM 使用者，但不與特定的人員相關聯。您可以在 AWS Management Console 中透過切換角色來暫時取得 IAM 角色。您可以透過呼叫 AWS CLI 或 AWS API 操作，或是使用自訂 URL 來取得角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色。

使用臨時性憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的詳細資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center (successor to AWS Single Sign-On) 使用者指南中的許可集。
- 暫時 IAM 使用者許可：使用者可以擔任 IAM 角色或角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶的資源。角色是授予跨帳戶存取的主要方式。但是，針對某些 AWS 服務，您可以將政策直接連接到資源 (而非使用角色作為代理)。若要了解跨帳戶存取角色和資源型政策間的差異，請參閱《IAM 使用者指南》中的 IAM 角色與資源類型政策的差異。
- 跨服務存取：有些 AWS 服務 會使用其他 AWS 服務 中的功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件存放在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫委託人的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 委託人許可：當您使用 IAM 使用者或角色在 AWS 中執行動作時，您會被視為委託人。政策能將許可授予委託人。當您使用某些服務時，您可能會執行一個動作，然後在不同的服務中觸發另一個動作。在此情況下，您必須具有執行這兩個動作的許可。若要查看某個動作是否需要原則中的其他相依動作，請參閱服務授權參考中的。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務服務。
  - 服務連結角色：服務連結角色是一種連結到 AWS 服務 的服務角色類型。服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的 AWS 帳戶 中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式：針對在 EC2 執行個體上執行並提出 AWS CLI 和 AWS API 請求的應用程式，您可以使用 IAM 角色來管理臨時性憑證。這是在 EC2 執行個體內存放存取金鑰的較好方式。若要指派 AWS 角色給 EC2 執行個體並提供其所有應用程式使用，您可以建立連接到執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得臨時性憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱《IAM 使用者指南》中的建立 IAM 角色 (而非使用者) 的時機。

在海王星中啟用 IAM 資料庫身分

根據預設，當您建立 Amazon Neptune 資料庫叢集時，IAM 資料庫身份驗證會停用。您可以使用 AWS Management Console 啟用 IAM 資料庫身分驗證 (或再次停用它)。

若要使用主控台建立具有 IAM 驗證的新 Neptune 資料庫叢集，請遵循中建立 Neptune 資料庫叢集的指示使用啟動 Neptune 資料庫叢集AWS Management Console (p. 534)。

在建立程序的第二頁，針對 Enable IAM DB Authentication (啟用 IAM 資料庫身分驗證)，選擇 Yes (是)。

為現有資料庫執行個體或叢集啟用或停用 IAM 身分驗證

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Clusters (叢集)。
3. 選擇您要修改的 Neptune DB 叢集，然後選擇 [叢集動作]。然後選擇 Modify cluster (修改叢集)。
4. 在 Database options (資料庫選項) 區段的 IAM DB Authentication (IAM 資料庫身分驗證)，選擇 Enable IAM DB authorization (啟用 IAM 資料庫授權) 或 No (否) (以進行停用)。然後選擇 Continue (繼續)。
5. 若要立即套用變更，請選擇 Apply immediately (立即套用)。
6. 選擇 Modify cluster (修改叢集)。

使用簽AWS名版本 4 連接和簽名

啟用 IAM 資料庫身份驗證的 Amazon 海王星資源需要使用簽AWS名版本 4 簽署所有 HTTP 請求。如需有關使用簽AWS名版本 4 簽署要求的一般資訊，請參閱簽名版本 4 簽署程序。

AWS簽名版本 4 是將驗證信息添加到AWS請求的過程。基於安全考量，大多數對於 AWS 的請求都必須以存取金鑰來簽署，而這組存取金鑰中包含存取金鑰 ID 和私密存取金鑰。

> Note
>
> 如果您使用臨時登入資料，登入資料會在指定的時間後過期，包括工作階段字符。
> 當您請求新的登入資料時，必須更新工作階段字符。如需詳細資訊，請參閱使用臨時安全登入資料來要求AWS資源的存取權。
>
> Important
>
> 使用以 IAM 為基礎的驗證存取 Neptune 時，您必須建立 HTTP 要求並自行簽署要求。

Signature 第 4 版的運作方式

1. 建立正式的請求。
2. 您可以使用規範要求和其他一些資訊來建立. string-to-sign
3. 您可以使用AWS密鑰訪問密鑰來導出簽名密鑰，然後使用該簽名密鑰和創建簽名。string-to-sign
4. 您可以對標頭中的 HTTP 請求新增所產生的簽章，或做為查詢字串參數。

Neptune 收到請求時，它會執行與計算簽名所做的相同步驟。然後，Neptune 將計算出的簽名與您隨請求發送的簽名進行比較。如果簽章相符，就會受理請求。如果簽章不相符，則會拒絕請求。

如需有關使用簽AWS名版本 4 簽署要求的一般資訊，請參閱 AWS 一般參考.

以下各節包含的範例說明如何將已簽署的要求傳送至啟用 IAM 身份驗證的 Neptune 資料庫執行個體的 Gramlin 和 SPARQL 端點。

主題

- Amazon Linux EC2 的事前準備 (p. 161)
- 使用命令列工具將查詢提交至 Neptune 資料庫叢集 (p. 161)
- 使用具有簽名版本 4 簽名的 Gimlin 控制台連接到海王星 (p. 163)
- 使用具有簽名版本 4 簽名的 Java 和小鬼連接到海王星 (p. 165)
- 使用具有簽名版本 4 簽名的 Java 和 SPARQL 連接到海王星（RDF4J 和耶拿）(p. 167)
- 透過簽章第 4 版簽署，使用 SPARQL 和 Node.js 連線至海王星 (p. 168)
- 示例：使用具有簽名版本 4 簽名的 Python 連接到海王星 (p. 169)

Amazon Linux EC2 的事前準備

以下是在亞馬遜 EC2 實例上安裝阿帕奇 Maven 和 Java 8 的說明。Amazon 海王星簽名版本 4 身份驗證範例需要這些資料。

若要在 EC2 執行個體上安裝 Apache Maven 和 Java 8。

1. 使用安全殼層用戶端連線到您的 Amazon EC2 執行個體。
2. 在 EC2 執行個體上安裝 Apache Maven。首先，輸入以下內容，新增包含 Maven 套件的儲存庫。

   ```
   sudo wget https://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
   ```

   輸入以下內容以設定套件的版本號碼。

   ```
   sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
   ```

   接著使用 yum 安裝 Maven。

   ```
   sudo yum install -y apache-maven
   ```
3. Gremlin 二進位碼需要 Java 8。輸入以下內容將 Java 8 安裝在您的 EC2 執行個體上。

   ```
   sudo yum install java-1.8.0-devel
   ```
4. 輸入以下內容將 Java 8 設定為 EC2 執行個體上預設的執行時間。

   ```
   sudo /usr/sbin/alternatives --config java
   ```

   出現提示時，輸入 Java 8 的數字。
5. 輸入以下內容，將 Java 8 設為 EC2 執行個體上的預設編譯器。

   ```
   sudo /usr/sbin/alternatives --config javac
   ```

   出現提示時，輸入 Java 8 的數字。

使用命令列工具將查詢提交至 Neptune 資料庫叢集

如本文件中的許多範例所示，擁有將查詢提交至 Neptune DB 叢集的命令列工具非常方便。如果未啟用 IAM 身份驗證，curl 工具是與 Neptune 端點通訊的絕佳選擇。

不過，為了確保資料安全，最好啟用 IAM 身份驗證。

啟用 IAM 身份驗證後，每個請求都必須使用簽名版本 4（Sig4）進行簽名。第三方 awscurl 命令列工具使用與相同的語法curl，並且可以使用 Sig4 簽名來簽署查詢。以下使用 awscurl (p. 162)章節說明如何awscurl安全地搭配臨時登入資料使用。

設定命令列工具以使用 HTTPS

海王星要求所有連線都使用 HTTPS。任何命令行工具，如curl或awscurl需要訪問適當的證書才能使用 HTTPS。只要curl或awscurl可以找到適當的證書，它們就像 HTTP 連接一樣處理 HTTPS 連接，而不需要額外的參數。此文件的範例是以該案例為基礎。

若要了解如何取得這類憑證，以及如何將這些憑證正確格式化為curl可以使用的憑證授權單位 (CA) 憑證存放區，請參閱curl文件中的 SSL 憑證驗證。

然後，您可以使用 CURL_CA_BUNDLE 環境變數來指定此 CA 憑證存放區的位置。在 Windows 上，curl 會自動在名為 curl-ca-bundle.crt 的檔案中尋找它。它會先在和 curl.exe 相同的目錄中尋找，然後再尋找路徑的其他位置。如需詳細資訊，請參閱 SSL Certificate Verification。

awscurl搭配臨時登入資料使用，安全地連線至啟用 IAM 身份驗證的資料庫叢集

awscurl 工具使用與語法相同curl，但也需要其他資訊：

- **--access_key**— 有效的存取金鑰。如果未使用此參數提供，則必須在AWS_ACCESS_KEY_ID環境變數或組態檔案中提供此參數。
- **--secret_key**— 對應於存取金鑰的有效密鑰。如果未使用此參數提供，則必須在AWS_SECRET_ACCESS_KEY環境變數或組態檔案中提供此參數。
- **--security_token**— 有效的工作階段權杖。如果未使用此參數提供，則必須在AWS_SECURITY_TOKEN環境變數或組態檔案中提供此參數。

在過去，常見的做法是將持續性登入資料搭配使用awscurl，例如 IAM 使用者登入資料，甚至是根登入資料，但不建議這麼做。而是使用其中一個AWS安全性權杖服務 (STS) API 或其中一個AWS CLI包裝函式來產生臨時認證。

最好將 STS 呼叫所傳回的AccessKeyIdSecretAccessKey、和SessionToken值放入 Shell 工作階段中的適當環境變數，而不是放入組態檔案中。然後，當 shell 終止時，認證將自動丟棄，這與配置文件不是這種情況。同樣地，請勿要求臨時憑證的持續時間長於您可能需要的時間。

下列範例顯示您可能在 Linux 命令介面中採取的步驟，以取得使用 sts assume-role 有效半小時的臨時認證，然後將它們放置在awscurl可以找到它們的環境變數中：

```
aws sts assume-role \
    --duration-seconds 1800 \
    --role-arn "arn:aws:iam::(account-id):role/(rolename)" \
    --role-session-name AWSCLI-Session > $output
AccessKeyId=$(cat $output | jq '.Credentials''.AccessKeyId')
SecretAccessKey=$(cat $output | jq '.Credentials''.SecretAccessKey')
SessionToken=$(cat $output | jq '.Credentials''.SessionToken')

export AWS_ACCESS_KEY_ID=$AccessKeyId
export AWS_SECRET_ACCESS_KEY=$SecretAccessKey
export AWS_SESSION_TOKEN=$SessionToken
```

然後，您可以使用awscurl向數據庫集群發出簽名的請求，如下所示：

```
awscurl (your cluster endpoint):8182/status \
    --region us-east-1 \
    --service neptune-db
```

使用具有簽名版本 4 簽名的 Gimlin 控制台連接到海王星

使用具有簽名版本 4 身份驗證的 Gemlin 控制台連接到 Amazon Neptune 的方式取決於您使用的是版本還是更高TinkerPop版本3.4.11，還是較早的版本。無論是哪一種情況，都必須符合下列先決條件：

- 您必須擁有簽署請求所需的 IAM 登入資料。請參閱AWS SDK for Java開發人員指南中的使用預設憑證提供者鏈結。
- 您必須安裝與資料庫叢集所使用的 Neptune 引擎版本相容的 Gemlin 主控台版本。

如果您使用的是臨時憑據，它們會在指定的間隔後過期，與會話令牌一樣，因此在請求新憑據時必須更新會話令牌。請參閱 IAM 使用者指南中的使用臨時安全登入AWS資料要求存取資源。

如需使用 SSL/TLS 連線的說明，請參閱。SSL/TLS 組態 (p. 310)

使用 TinkerPop 3.4.11 或更高版本通過 Sig4 簽名連接到海王星

使用 TinkerPop 3.4.11 或更高版本時，您將使用handshakeInterceptor()，它提供了一種將 Sigv4 簽署者插入命令建立的連接的方法。:remote與用於 Java 的方法一樣，它需要您手動配置對Cluster象，然後將其傳遞給命:remote令。

請注意，這與:remote命令需要配置文件形成連接的典型情況完全不同。配置文件方法將不起作用，因為handshakeInterceptor()必須以編程方式設置，並且無法從文件加載其配置。

使用 Sig4 簽名連接小鬼控制台（TinkerPop3.4.11 及更高版本）

1. 啟動小精靈控制台：

   ```
   $ bin/gremlin.sh
   ```

2. 在gremlin>提示符下，安裝amazon-neptune-sigv4-signer庫（這只需要為控制台完成一次）：

   ```
   :install com.amazonaws amazon-neptune-sigv4-signer 2.4.0
   ```

   如果您在此步驟中遇到問題，可能有助於查閱有關葡萄配置的TinkerPop文檔。

3. 將處理登錄所需的類導入到handshakeInterceptor()：

   ```
   :import com.amazonaws.auth.DefaultAWSCredentialsProviderChain
   :import com.amazonaws.neptune.auth.NeptuneNettyHttpSigV4Signer
   ```

4. 如果您使用臨時憑據，則還需要提供會話令牌，如下所示：

   ```
   System.setProperty("aws.sessionToken","(your session token)")
   ```

5. 如果您尚未建立帳戶憑據，則可以按照以下方式分配它們：

   ```
   System.setProperty("aws.accessKeyId","(your access key)")
   System.setProperty("aws.secretKey","(your secret key)")
   ```

6. 手動構造要連接到海王星的Cluster對象：

```
cluster = Cluster.build("(host name)")  \
                .enableSsl(true) \
                .handshakeInterceptor { r ->  \
                  def sigV4Signer = new NeptuneNettyHttpSigV4Signer("(Amazon region)",
 \
                                        new DefaultAWSCredentialsProviderChain()); \
                  sigV4Signer.signRequest(r); \
                  return r; } \
                .create()
```

如需尋找 Neptune 資料庫執行個體主機名稱的說明，請參閱連接到亞馬遜海王星端點 (p. 121)。

7. 在上一個步驟中，使用Cluster物件的變數名稱建立:remote連線：

```
:remote connect tinkerpop.server cluster
```

8. 輸入以下指令以切換至遠端模式。這將所有的 Gemlin 查詢發送到遠程連接：

```
:remote console
```

使用 3.4.11 之TinkerPop前的版本通過 Sig4 簽名連接到海王星

使用 TinkerPop 3.4.10 或更低版本時，請使用 Neptune 提供的程式amazon-neptune-gremlin-java-sigv4庫，透過 Sigv4 簽章將主控台連線到 Neptune，如下所述：

使用 Sig4 簽名連接小鬼控制台（3.4.11 之前的TinkerPop版本）

1. 啟動小精靈控制台：

```
$ bin/gremlin.sh
```

2. 在gremlin>提示符下，安裝amazon-neptune-sigv4-signer庫（這只需要為控制台完成一次）：

```
:install com.amazonaws amazon-neptune-sigv4-signer 2.4.0
```

如果您在此步驟中遇到問題，可能有助於查閱有關葡萄配置的TinkerPop文檔。

3. 在解壓縮conf目錄的子目錄中，建立一個名為neptune-remote.yaml的檔案。

如果您使用AWS CloudFormation範本建立 Neptune 資料庫叢集，則neptune-remote.yaml檔案將已存在。在這種情況下，您要做的就是編輯現有文件以包含如下所示的 channelizer 設置。

否則，請將下列文字複製到檔案中，並以 Neptune DB 執行個體的主機名稱或 IP 位址取代 *(####)*。請注意，包含主機名稱的方括號 ([]) 是必需的。

```
hosts: [(host name)]
port: 8182
connectionPool: {
  channelizer: org.apache.tinkerpop.gremlin.driver.SigV4WebSocketChannelizer,
  enableSsl: true
}
serializer: { className:
 org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1,
              config: { serializeResultToString: true }}
```

4. Important

   必須提供 IAM 登入資料才能簽署請求。輸入以下命令來將您的登入資料設為環境變數，並使用您的登入資料取代相關項目。

   ```
   export AWS_ACCESS_KEY_ID=access_key_id
   export AWS_SECRET_ACCESS_KEY=secret_access_key
   export SERVICE_REGION=us-east-1 or us-east-2 or us-west-1 or us-west-2 or ca-central-1 or
                         sa-east-1 or eu-north-1 or eu-west-1 or eu-west-2 or eu-west-3 or eu-central-1 or me-south-1 or me-south-2 or
                         af-south-1 or ap-east-1 or ap-northeast-1 or ap-northeast-2 or ap-southeast-1 or ap-southeast-2 or ap-south-1 or
                         cn-north-1 or cn-northwest-1 or
                         us-gov-east-1 or us-gov-west-1
   ```

   Neptune 第 4 版簽署者使用預設的認證提供者鏈結。如需提供認證的其他方法，請參閱AWS SDK for Java開發人員指南中的使用預設認證提供者鏈結。
   即使使用登入資料檔案，SERVICE_REGION 變數也仍是必要項目。

5. 使用以下.yaml檔案建立:remote連線：

   ```
   :remote connect tinkerpop.server conf/neptune-remote.yaml
   ```

6. 輸入以下命令以切換到遠程模式，該模式將所有 Gemlin 查詢發送到遠程連接：

   ```
   :remote console
   ```

使用具有簽名版本 4 簽名的 Java 和小鬼連接到海王星

使用 TinkerPop 3.4.11 或更高版本通過 Sig4 簽名連接到海王星

以下是一個示例，說明如何在使用 TinkerPop 3.4.11 或更高版本時使用帶有 Sig4 簽名的小鬼 Java API 連接到海王星（它假定有關使用 Maven 的一般知識）。首先，將依賴關係定義為pom.xml文件的一部分：

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>amazon-neptune-sigv4-signer</artifactId>
  <version>2.4.0</version>
</dependency>
```

然後，使用如下代碼：

```
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.neptune.auth.NeptuneNettyHttpSigV4Signer;
import com.amazonaws.neptune.auth.NeptuneSigV4SignerException;

 ...

System.setProperty("aws.accessKeyId","your-access-key");
System.setProperty("aws.secretKey","your-secret-key");

 ...

Cluster = Cluster.build((your cluster name))
                 .enableSsl(true)
                 .handshakeInterceptor( r ->
                  {
```

```
                    try {
                      NeptuneNettyHttpSigV4Signer sigV4Signer =
                        new NeptuneNettyHttpSigV4Signer("(your region)", new
 DefaultAWSCredentialsProviderChain());
                      sigV4Signer.signRequest(r);
                    } catch (NeptuneSigV4SignerException e) {
                      throw new RuntimeException("Exception occurred while signing the
 request", e);
                    }
                    return r;
                  }
                ).create();
try {
  Client client = cluster.connect();
  client.submit("g.V().has('code','IAD')").all().get();
} catch (Exception e) {
  throw new RuntimeException("Exception occurred while connecting to cluster", e);
}
```

Note

如果您要從中升級3.4.11，請移除對amazon-neptune-gremlin-java-sigv4資源庫的參考。使用時不再需要handshakeInterceptor()如上面的例子所示。不要嘗試與通道儀（SigV4WebSocketChannelizer.class）一起使用，因為它會產生錯誤。handshakeInterceptor()

使用 3.4.11 之TinkerPop前的版本通過 Sig4 簽名連接到海王星

TinkerPop之前的版本3.4.11不支援上一節 (p. 165)所示的handshakeInterceptor()組態，因此必須依賴amazon-neptune-gremlin-java-sigv4套件。這是一個包含該SigV4WebSocketChannelizer類的 Neptune 庫，它將標準TinkerPop通道器替換為可以自動注入 Sigv4 簽名的通道器。在可能的情況下，升級到 TinkerPop 3.4.11 或更高版本，因為該amazon-neptune-gremlin-java-sigv4庫已被棄用。

以下是使用 3.4.11 之前的TinkerPop版本（假設有關如何使用 Maven 的一般知識）時，如何使用具有 Sig4 簽名的 Gremlin Java API 連接到海王星的示例。

首先，將依賴關係定義為pom.xml文件的一部分：

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>amazon-neptune-gremlin-java-sigv4</artifactId>
  <version>2.4.0</version>
</dependency>
```

上面的依賴項將包括 Gemlin 驅動程序版本。3.4.10雖然可以使用較新的 Gemlin 驅動程式版本 (直到3.4.13)，但是升級到 3.4.10 之後的驅動程式應該包含使用上述handshakeInterceptor()模型的變更。 (p. 165)

然後，gremlin-driver群集對象應該在 Java 代碼中配置如下：

```
import org.apache.tinkerpop.gremlin.driver.SigV4WebSocketChannelizer;

 ...

Cluster cluster = Cluster.build(your cluster)
                      .enableSsl(true)
                      .channelizer(SigV4WebSocketChannelizer.class)
                      .create();
Client client = cluster.connect();
client.submit("g.V().has('code','IAD')").all().get();
```

使用具有簽名版本 4 簽名的 Java 和 SPARQL 連接到海王星（RDF4J 和耶拿）

本節介紹如何使用 RDF4J 或 Apache 耶拿簽名版本 4 身份驗證連接到海王星。

先決條件

- Java 8 或更高版本。
- Apache Maven 3.3 或更高版本。

  如需在執行 Amazon Linux 的 EC2 執行個體上安裝這些先決條件的詳細資訊，請參閱Amazon Linux EC2 的事前準備 (p. 161)。
- 用於簽署請求的 IAM 登入資料。如需詳細資訊，請參閱AWS SDK for Java開發人員指南中的使用預設認證提供者鏈結。

  > **Note**
  >
  > 如果您使用臨時登入資料，登入資料會在指定的時間後過期，包括工作階段字符。當您請求新的登入資料時，必須更新工作階段字符。如需詳細資訊，請參閱 IAM 使用者指南中的使用暫時安全登入資料來請求對 AWS 資源的存取。
- 將SERVICE_REGION變數設定為下列其中一項，表示 Neptune 資料庫執行個體的區域：
  - 美國東部 (維吉尼亞北部): `us-east-1`
  - 美國東部 (俄亥俄州): `us-east-2`
  - 美國西部 (加利佛尼亞北部): `us-west-1`
  - 美國西部 (奧勒岡): `us-west-2`
  - 加拿大 (中部): `ca-central-1`
  - 南美洲（聖保羅）：`sa-east-1`
  - 歐洲 (斯德哥爾摩): `eu-north-1`
  - 歐洲 (愛爾蘭): `eu-west-1`
  - 歐洲 (倫敦): `eu-west-2`
  - 歐洲（巴黎）：`eu-west-3`
  - 歐洲 (法蘭克福): `eu-central-1`
  - 中東（巴林）：`me-south-1`
  - 中東（阿聯酋）：`me-central-1`
  - 非洲（開普敦）：`af-south-1`
  - 亞太區域 (香港): `ap-east-1`
  - 亞太區域 (東京): `ap-northeast-1`
  - 亞太區域 (首爾): `ap-northeast-2`
  - 亞太區域 (新加坡)：`ap-southeast-1`
  - 亞太區域 (雪梨)：`ap-southeast-2`
  - 亞太區域 (孟買)：`ap-south-1`
  - 中國 (北京): `cn-north-1`
  - 中國 (寧夏): `cn-northwest-1`
  - AWSGovCloud(美國西部): `us-gov-west-1`
  - AWSGovCloud(美國東部): `us-gov-east-1`

使用 RDF4J 或阿帕奇耶拿簽名版本 4 簽名連接到海王星

1. 從複製範例存放庫GitHub。

```
git clone https://github.com/aws/amazon-neptune-sparql-java-sigv4.git
```

2. 變更到複製的目錄。

```
cd amazon-neptune-sparql-java-sigv4
```

3. 透過簽出具有 latest (最新) 標籤的分支，取得專案的最新版本。

```
git checkout $(git describe --tags `git rev-list --tags --max-count=1`)
```

4. 輸入以下其中一個命令來編譯及執行範例程式碼。

   將*########## Nep* tune 資料庫執行個體的主機名稱或 IP 位址。預設連接埠為 8182。

   > **Note**
   >
   > 如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

   Eclipse RDF4J

   輸入以下內容來執行 RDF4J 範例。

```
mvn compile exec:java \
    -Dexec.mainClass="com.amazonaws.neptune.client.rdf4j.NeptuneRdf4JSigV4Example" \
    -Dexec.args="https://your-neptune-endpoint:port"
```

   Apache Jena

   輸入以下內容來執行 Apache Jena 範例。

```
mvn compile exec:java \
    -Dexec.mainClass="com.amazonaws.neptune.client.jena.NeptuneJenaSigV4Example" \
    -Dexec.args="https://your-neptune-endpoint:port"
```

5. 若要查看範例的原始程式碼，請參閱 `src/main/java/com/amazonaws/neptune/client/` 目錄中的範例。

若要在您自己 Java 應用程式中使用 SigV4 簽署驅動程式，請將 `amazon-neptune-sigv4-signer` Maven 套件新增到您 `pom.xml` 的 `<dependencies>` 區段。我們建議您使用範例做為起點。

透過簽章第 4 版簽署，使用 SPARQL 和 Node.js 連線至海王星

以下是如何使用具有簽名版本 4 身份驗證的 Node.js 連接到海王星 SPARQL 的示例：

```
var AWS = require('aws-sdk');

var region = 'us-west-2'; // e.g. us-west-1
var neptune_endpoint = 'your-Neptune-cluster-endpoint';  // like: 'cluster-
id.region.neptune.amazonaws.com'
var query = `query=PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX class: <http://aws.amazon.com/neptune/csv2rdf/class/>
PREFIX resource: <http://aws.amazon.com/neptune/csv2rdf/resource/>
PREFIX prop: <http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/>
```

```
PREFIX objprop: <http://aws.amazon.com/neptune/csv2rdf/objectProperty/>

SELECT ?movies ?title WHERE {
    ?jel prop:name "James Earl Jones" .
    ?movies ?p2 ?jel .
    ?movies prop:title ?title
} LIMIT 10`;

runQuery(query);

function runQuery(q) {

    var endpoint = new AWS.Endpoint(neptune_endpoint);
    endpoint.port = 8182;
    var request = new AWS.HttpRequest(endpoint, region);
    request.path += 'sparql';
    request.body = encodeURI(query);
    request.headers['Content-Type'] = 'application/x-www-form-urlencoded';
    request.headers['host'] = neptune_endpoint;
    request.method = 'POST';

    var credentials = new AWS.CredentialProviderChain();
    credentials.resolve((err, cred)=>{
        var signer = new AWS.Signers.V4(request, 'neptune-db');
        signer.addAuthorization(cred, new Date());
    });

    var client = new AWS.HttpClient();
    client.handleRequest(request, null, function(response) {
        console.log(response.statusCode + ' ' + response.statusMessage);
        var responseBody = '';
        response.on('data', function (chunk) {
            responseBody += chunk;
        });
        response.on('end', function (chunk) {
            console.log('Response body: ' + responseBody);
        });
    }, function(error) {
        console.log('Error: ' + error);
    });
}
```

示例：使用具有簽名版本 4 簽名的 Python 連接到海王星

本節顯示使用 Python 撰寫的範例程式，說明如何使用亞馬遜海王星的簽名版本 4。此範例是根據 https://docs.aws.amazon.com/general/latest/gr/sigv4-signed-request-examples.html 中的 Amazon Web Services 一般參考Signature 第 4 版簽署程序部分。

若要使用此範例程式，您需要以下項目：

- 將 Python 3.x 安裝在電腦上，這個程式可從 Python 網站取得。這些程式已使用 Python 3.6 來測試。
- Python requests library，在範例指令碼中用來進行 web 請求。Python 套件安裝方便，只要使用 `pip` 即可從 Python 套件索引網站取得套件。然後，您可以在命令列執行 `pip install requests`，即可安裝 `requests`。
- 環境變數中的存取金鑰 (存取金鑰 ID 和私密存取金鑰) 名為 `AWS_ACCESS_KEY_ID` `AWS_SECRET_ACCESS_KEY`。根據最佳實務，建議您請勿 在程式碼中內嵌登入資料。如需詳細資訊，請參閱《AWS Account Management 參考指南》中的 AWS 帳戶的最佳實務。

  Neptune DB 叢集在名為的環境變數中的區域`SERVICE_REGION`。

如果您使用臨時的登入資料，除了指定
AWS_ACCESS_KEY_ID、AWS_SECRET_ACCESS_KEY、SERVICE_REGION，還必須指定
AWS_SESSION_TOKEN。

> **Note**
>
> 如果您使用臨時登入資料，登入資料會在指定的時間後過期，包括工作階段字符。
> 當您請求新的登入資料時，必須更新工作階段字符。如需詳細資訊，請參閱使用臨時安全登入資料來要求AWS資源的存取權。

下面的例子演示了如何使用 Python 向海王星發出簽名的請求。此請求提出 GET 或 POST 請求。身分驗證資訊會使用 Authorization 請求標頭來傳遞。

此範例也能做為 AWS Lambda 函數使用。如需詳細資訊，請參閱the section called “設定 Lambda” (p. 221)。

向小鬼和 SPARQL 海王星端點發出已簽署的要求

1. 建立名為 neptunesigv4.py 的新檔案，並在文字編輯器中開啟。
2. 複製以下程式碼，並在 neptunesigv4.py 檔案中貼上。

```
# Amazon Neptune version 4 signing example (version v3)

# The following script requires python 3.6+
#    (sudo yum install python36 python36-virtualenv python36-pip)
# => the reason is that we're using urllib.parse() to manually encode URL
#    parameters: the problem here is that SIGV4 encoding requires whitespaces
#    to be encoded as %20 rather than not or using '+', as done by previous/
#    default versions of the library.


# See: https://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html
import sys, datetime, hashlib, hmac
import requests  # pip3 install requests
import urllib
import os
import json
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
from botocore.credentials import ReadOnlyCredentials
from types import SimpleNamespace
from argparse import RawTextHelpFormatter
from argparse import ArgumentParser

# Configuration. https is required.
protocol = 'https'

# The following lines enable debugging at httplib level (requests->urllib3-
>http.client)
# You will see the REQUEST, including HEADERS and DATA, and RESPONSE with HEADERS but
 without DATA.
#
# The only thing missing will be the response.body which is not logged.
#
# import logging
# from http.client import HTTPConnection
# HTTPConnection.debuglevel = 1
# logging.basicConfig()
# logging.getLogger().setLevel(logging.DEBUG)
# requests_log = logging.getLogger("requests.packages.urllib3")
# requests_log.setLevel(logging.DEBUG)
```

```
# requests_log.propagate = True


# Read AWS access key from env. variables. Best practice is NOT
# to embed credentials in code.
access_key = os.getenv('AWS_ACCESS_KEY_ID', '')
secret_key = os.getenv('AWS_SECRET_ACCESS_KEY', '')
region = os.getenv('SERVICE_REGION', '')

# AWS_SESSION_TOKEN is optional environment variable. Specify a session token only if
 you are using temporary
# security credentials.
session_token = os.getenv('AWS_SESSION_TOKEN', '')

### Note same script can be used for AWS Lambda (runtime = python3.6).
## Steps to use this python script for AWS Lambda
# 1. AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY and AWS_SESSION_TOKEN and AWS_REGION
 variables are already part of Lambda's Execution environment
#    No need to set them up explicitly.
# 3. Create Lambda deployment package https://docs.aws.amazon.com/lambda/latest/dg/
lambda-python-how-to-create-deployment-package.html
# 4. Create a Lambda function in the same VPC and assign an IAM role with neptune
 access

def lambda_handler(event, context):
    # sample_test_input = {
    #     "host": "END_POINT:8182",
    #     "method": "GET",
    #     "query_type": "gremlin",
    #     "query": "g.V().count()"
    # }

    # Lambda uses AWS_REGION instead of SERVICE_REGION
    global region
    region = os.getenv('AWS_REGION', '')

    host = event['host']
    method = event['method']
    query_type = event['query_type']
    query =  event['query']

    return make_signed_request(host, method, query_type, query)

def validate_input(method, query_type):
    # Supporting GET and POST for now:
    if (method != 'GET' and method != 'POST'):
        print('First parameter must be "GET" or "POST", but is "' + method + '".')
        sys.exit()

    # SPARQL UPDATE requires POST
    if (method == 'GET' and query_type == 'sparqlupdate'):
        print('SPARQL UPDATE is not supported in GET mode. Please choose POST.')
        sys.exit()

def get_canonical_uri_and_payload(query_type, query, method):
    # Set the stack and payload depending on query_type.
    if (query_type == 'sparql'):
        canonical_uri = '/sparql/'
        payload = {'query': query}

    elif (query_type == 'sparqlupdate'):
        canonical_uri = '/sparql/'
        payload = {'update': query}

    elif (query_type == 'gremlin'):
        canonical_uri = '/gremlin/'
```

```
        payload = {'gremlin': query}
        if (method == 'POST'):
            payload = json.dumps(payload)

    elif (query_type == 'openCypher'):
        canonical_uri = '/openCypher/'
        payload = {'query': query}

    elif (query_type == "loader"):
        canonical_uri = "/loader/"
        payload = query

    elif (query_type == "status"):
        canonical_uri = "/status/"
        payload = {}

    elif (query_type == "gremlin/status"):
        canonical_uri = "/gremlin/status/"
        payload = {}

    elif (query_type == "openCypher/status"):
        canonical_uri = "/openCypher/status/"
        payload = {}

    elif (query_type == "sparql/status"):
        canonical_uri = "/sparql/status/"
        payload = {}

    else:
        print(
            'Third parameter should be from ["gremlin", "sparql", "sparqlupdate",
 "loader", "status] but is "' + query_type + '".')
        sys.exit()
    ## return output as tuple
    return canonical_uri, payload

def make_signed_request(host, method, query_type, query):
    service = 'neptune-db'
    endpoint = protocol + '://' + host

    print()
    print('+++++ USER INPUT +++++')
    print('host = ' + host)
    print('method = ' + method)
    print('query_type = ' + query_type)
    print('query = ' + query)

    # validate input
    validate_input(method, query_type)

    # get canonical_uri and payload
    canonical_uri, payload = get_canonical_uri_and_payload(query_type, query, method)

    # assign payload to data or params
    data = payload if method == 'POST' else None
    params = payload if method == 'GET' else None

    # create request URL
    request_url = endpoint + canonical_uri

    # create and sign request
    creds = SimpleNamespace(
        access_key=access_key, secret_key=secret_key, token=session_token,
 region=region,
    )
```

```
    request = AWSRequest(method=method, url=request_url, data=data, params=params)
    SigV4Auth(creds, service, region).add_auth(request)

    r = None

    # ************* SEND THE REQUEST *************
    if (method == 'GET'):

        print('++++ BEGIN GET REQUEST +++++')
        print('Request URL = ' + request_url)
        r = requests.get(request_url, headers=request.headers, verify=False,
 params=params)

    elif (method == 'POST'):

        print('\n+++++ BEGIN POST REQUEST +++++')
        print('Request URL = ' + request_url)
        if (query_type == "loader"):
            request.headers['Content-type'] = 'application/json'
        r = requests.post(request_url, headers=request.headers, verify=False,
 data=data)

    else:
        print('Request method is neither "GET" nor "POST", something is wrong here.')

    if r is not None:
        print()
        print('+++++ RESPONSE +++++')
        print('Response code: %d\n' % r.status_code)
        response = r.text
        r.close()
        print(response)

        return response

help_msg = '''
    export AWS_ACCESS_KEY_ID=[MY_ACCESS_KEY_ID]
    export AWS_SECRET_ACCESS_KEY=[MY_SECRET_ACCESS_KEY]
    export AWS_SESSION_TOKEN=[MY_AWS_SESSION_TOKEN]
    export SERVICE_REGION=[us-east-1|us-east-2|us-west-2|eu-west-1]

    python version >=3.6 is required.

    Examples: For help
    python3 program_name.py -h

    Examples: Queries
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q status
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q sparql/status
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q sparql -d
 "SELECT ?s WHERE { ?s ?p ?o }"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a POST -q sparql -d
 "SELECT ?s WHERE { ?s ?p ?o }"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a POST -q sparqlupdate -
d "INSERT DATA { <https://s> <https://p> <https://o> }"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q gremlin/status
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q gremlin -d
 "g.V().count()"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a POST -q gremlin -d
 "g.V().count()"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q openCypher/
status
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q openCypher -d
 "MATCH (n1) RETURN n1 LIMIT 1;"
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a POST -q openCypher -d
 "MATCH (n1) RETURN n1 LIMIT 1;"
```

```
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q loader -d
 '{"loadId": "68b28dcc-8e15-02b1-133d-9bd0557607e6"}'
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a GET -q loader -d '{}'
    python3 program_name.py -ho your-neptune-endpoint -p 8182 -a POST -q loader -d
 '{"source": "source", "format" : "csv", "failOnError": "fail_on_error", "iamRoleArn":
 "iam_role_arn", "region": "region"}'

    Environment variables must be defined as AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY
 and SERVICE_REGION.
    You should also set AWS_SESSION_TOKEN environment variable if you are using
 temporary credentials (ex. IAM Role or EC2 Instance profile).

    Current Limitations:
    - Query mode "sparqlupdate" requires POST (as per the SPARQL 1.1 protocol)
            '''

def exit_and_print_help():
    print(help_msg)
    exit()

def parse_input_and_query_neptune():


    parser = ArgumentParser(description=help_msg, formatter_class=RawTextHelpFormatter)
    group_host = parser.add_mutually_exclusive_group()
    group_host.add_argument("-ho", "--host", type=str)
    group_port = parser.add_mutually_exclusive_group()
    group_port.add_argument("-p", "--port", type=int, help="port ex. 8182,
 default=8182", default=8182)
    group_action = parser.add_mutually_exclusive_group()
    group_action.add_argument("-a", "--action", type=str, help="http action, default =
 GET", default="GET")
    group_endpoint = parser.add_mutually_exclusive_group()
    group_endpoint.add_argument("-q", "--query_type", type=str, help="query_type,
 default = status ", default="status")
    group_data = parser.add_mutually_exclusive_group()
    group_data.add_argument("-d", "--data", type=str, help="data required for the http
 action", default="")

    args = parser.parse_args()
    print(args)

    # Read command line parameters
    host = args.host
    port = args.port
    method = args.action
    query_type = args.query_type
    query = args.data

    if (access_key == ''):
        print('!!! ERROR: Your AWS_ACCESS_KEY_ID environment variable is undefined.')
        exit_and_print_help()

    if (secret_key == ''):
        print('!!! ERROR: Your AWS_SECRET_ACCESS_KEY environment variable is
 undefined.')
        exit_and_print_help()

    if (region == ''):
        print('!!! ERROR: Your SERVICE_REGION environment variable is undefined.')
        exit_and_print_help()

    if host is None:
        print('!!! ERROR: Neptune DNS is missing')
        exit_and_print_help()
```

```
    host = host + ":" + str(port)
    make_signed_request(host, method, query_type, query)


if __name__ == "__main__":
    parse_input_and_query_neptune()
```

3. 在終端機瀏覽到 neptunesigv4.py 檔案的位置。
4. 輸入以下命令，將存取金鑰、私有金鑰和區域取代成正確值。

```
export AWS_ACCESS_KEY_ID=MY_ACCESS_KEY_ID
export AWS_SECRET_ACCESS_KEY=MY_SECRET_ACCESS_KEY
export SERVICE_REGION=us-east-1 or us-east-2 or us-west-1 or us-west-2 or ca-central-1
 or
                       sa-east-1 or eu-north-1 or eu-west-1 or eu-west-2 or eu-west-3 or
 eu-central-1 or me-south-1 or me-south-2 or
                       af-south-1 or ap-east-1 or ap-northeast-1 or ap-northeast-2 or
 ap-southeast-1 or ap-southeast-2 or ap-south-1 or
                       cn-north-1 or cn-northwest-1 or
                       us-gov-east-1 or us-gov-west-1
```

如果您使用臨時的登入資料，除了指定 AWS_ACCESS_KEY_ID、AWS_SECRET_ACCESS_KEY、SERVICE_REGION，還必須指定 AWS_SESSION_TOKEN。

```
export AWS_SESSION_TOKEN=MY_AWS_SESSION_TOKEN
```

> **Note**
>
> 如果您使用臨時登入資料，登入資料會在指定的時間後過期，包括工作階段字符。當您請求新的登入資料時，必須更新工作階段字符。如需詳細資訊，請參閱使用臨時安全登入資料來要求AWS資源的存取權。

5. 輸入下列其中一個命令，將已簽署的要求傳送至 Neptune 資料庫執行個體。這些範例使用 Python 3.6 版。

端點狀態

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q status
```

Gremlin

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q gremlin -d
 "g.V().count()"

python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a POST -q gremlin -d
 "g.V().count()"
```

小精靈狀態

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q gremlin/status
```

SPARQL

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q sparql -d
 "SELECT ?s WHERE { ?s ?p ?o }"
```

SPARQL UPDATE

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a POST -q sparqlupdate -d
 "INSERT DATA { <https://s> <https://p> <https://o> }"
```

SPARQL 狀態

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q sparql/status
```

开放密码

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q openCypher -d
 "MATCH (n1) RETURN n1 LIMIT 1;"

python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a POST -q openCypher -d
 "MATCH (n1) RETURN n1 LIMIT 1;"
```

开放密码状态

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q openCypher/status
```

載入器

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q loader -d
 '{"loadId": "68b28dcc-8e15-02b1-133d-9bd0557607e6"}'

python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a GET -q loader -d '{}'

python3.6 neptunesigv4.py -ho your-neptune-endpoint -p 8182 -a POST -q loader -d
 '{"source": "source", "format" : "csv", "failOnError": "fail_on_error", "iamRoleArn":
 "iam_role_arn", "region": "region"}'
```

6. 執行 Python 指令碼的語法如下：

```
python3.6 neptunesigv4.py -ho your-neptune-endpoint -p port -a GET|POST -q gremlin|
sparql|sparqlupdate|loader|status -d "string0data"
```

SPARQL UPDATE 要求 POST。

使用 IAM 政策管理存取

IAM 政策是 JSON 物件，可定義使用動作和資源的許可。

您可以透過建立政策並將其連接到 AWS 身分或資源，在 AWS 中控制存取。政策是 AWS 中的一個物件，當其和身分或資源建立關聯時，便可定義其許可。AWS 會在主體 (使用者、根使用者或角色工作階段) 發出請求時評估這些政策。政策中的許可，決定是否允許或拒絕請求。大部分政策以 JSON 文件形式存放在 AWS 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱《IAM 使用者指南》中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

根據預設，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具備該政策的使用者便可以從 AWS Management Console、AWS CLI 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策則是獨立的政策，您可以將這些政策連接到 AWS 帳戶 中的多個使用者、群組和角色。受管政策包含 AWS 受管政策和由客戶管理之政策。若要了解如何在受管政策及內嵌政策間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

搭AWS配組織使用服務控制原則 (SCP)

服務控制策略（SCP）是 JSON 策略，用於指定中AWS Organizations組織或組織單位（OU）的最大權限。AWS Organizations是一種用於分組和集中管理您企業擁有的多個AWS帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中的實體許可，包括每個 AWS 帳戶根帳戶使用者。如需有關組織和 SCP 的詳細資訊，請參閱《AWS Organizations使用者指南》中的 SCP 如何運作。

在AWS組織內的AWS帳戶中部署 Amazon Neptune 的客戶可以利用 SCP 控制哪些帳戶可以使用 Neptune。若要確保存取成員帳戶內的 Neptune，請務必`neptune-db:*`分別允許存取控制平面和資料平面 IAM 動作。`neptune:*`

使用亞馬遜海王星主控台所需的許可

若要讓使用者使用 Amazon Neptune 主控台，該使用者必須擁有一組最低限度的許可。這些權限可讓使用者描述其AWS帳戶的 Neptune 資源，並提供其他相關資訊，包括 Amazon EC2 安全性和網路資訊。

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言， 主控台就無法如預期運作。若要確保這些使用者仍可使用 Neptune 主控台，請同時將受`NeptuneReadOnlyAccess`管理的原則附加至使用者，如中所述AWS亞馬遜海王星的受管 (預先定義) 政策 (p. 179)。

您不需要為僅對AWS CLI或 Amazon Neptune API 進行呼叫的使用者允許最低主控台許可。

將 IAM 政策附加到 IAM 使用者

若要套用受管或自訂政策，請將其附加到 IAM 使用者。如需此主題的教學，請參閱 IAM 使用者指南中的建立和連接您的第一個客戶受管政策。

在您進行教學課程時，可使用本節所示的其中一個政策範例做為起點，並依您的需求進行自訂。教學課程結束時，您將有一個連接政策的 IAM 使用者可使用 `neptune-db:*` 動作。

Important

- IAM 政策的變更最多需要 10 分鐘才能套用到指定的 Neptune 資源。
- 套用至 Neptune 資料庫叢集的 IAM 政策適用於該叢集中的所有執行個體。

使用不同種類的 IAM 政策來控制對 Neptune 的存取

若要提供 Neptune 管理動作或 Neptune 資料庫叢集中資料的存取權，您可以將政策附加至 IAM 使用者或角色。如需如何將 IAM 政策附加至使用者的相關資訊，請參閱將 IAM 政策附加到 IAM 使用者 (p. 177)。如需將政策附加至角色的相關資訊，請參閱 IAM 使用者指南中的新增和移除 IAM 政策。

對於 Neptune 的一般存取權，您可以使用 Neptune 的其中一個受管理原則 (p. 179)。對於更受限制的存取，您可以使用 Neptune 支援的系統管理動作 (p. 188)和資源 (p. 198)來建立自己的自訂原則。

在自訂 IAM 政策中，您可以使用兩種不同類型的政策陳述式來控制 Neptune DB 叢集的不同存取模式：

- 系統管理原則陳述 (p. 187)式 — 系統管理原則陳述式可讓您存取 Neptune 管理 API，這些 API (p. 996) 可用來建立、設定和管理資料庫叢集及其執行個體。

  由於 Neptune 與 Amazon RDS 共用功能，因此 Neptune 政策中的管理動作、資源和條件金鑰會依設計使用`rds:`前置詞。
- 資料存取原則陳述式 (p. 203) — 資料存取原則陳述式使用資料存取動作 (p. 205)、資源 (p. 209)和條件索引鍵 (p. 211)來控制資料庫叢集所包含之資料的存取。

  Neptune 資料存取動作、資源和條件金鑰使用`neptune-db:`前置詞。

在亞馬遜海王星中使用 IAM 條件內容金鑰

您可以在控制 Neptune 存取權的 IAM 政策聲明中指定條件。然後，政策聲明只有在條件成立時才會生效。

例如，您可能希望政策陳述式只在特定日期之後生效，或者只有在要求中有特定值時才允許存取。

若要表示條件，您可以在政策陳述式的`Condition`元素中使用預先定義的條件索引鍵，以及 IAM 條件政策運算子 (例如 equals 或小於)。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個金鑰，AWS 會使用邏輯 AND 操作評估他們。若您為單一條件索引鍵指定多個值，AWS 會使用邏輯 OR 操作評估條件。必須符合所有條件，才會授予陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

條件索引鍵的資料類型會決定您可以使用哪些條件運算子來比較要求中的值與原則陳述式中的值。如果您使用與該資料類型不相容的條件運算子，相符項目一律會失敗，而且不會套用原則陳述式。

Neptune 支援系統管理原則陳述式的條件金鑰集與資料存取原則陳述式不同的條件金鑰

- 管理原則陳述式的條件索引鍵 (p. 199)
- 資料存取原則陳述式的條件索引鍵 (p. 211)

支援亞馬遜海王星中的 IAM 政策和存取控制功能

下表顯示 Neptune 針對系統管理政策陳述式和資料存取政策陳述式支援的 IAM 功能：

可搭配海王星使用的 IAM 功能

IAM 功能	行政	資料存取
以身分為基礎的政策 (p. 177)	是	是
資源型政策	否	否
政策動作	是	是
政策資源	是	是
全域條件索引鍵	是	（一個子集）

IAM 功能	行政	資料存取
基於標籤的條件鍵 (p. 199)	是	否
存取控制清單 (ACL)	否	否
服務控制政策 (SCP) (p. 177)	是	是
服務連結角色 (p. 215)	是	否

IAM 政策限制

IAM 政策的變更最多需要 10 分鐘才能套用到指定的 Neptune 資源。

套用至 Neptune 資料庫叢集的 IAM 政策適用於該叢集中的所有執行個體。

Neptune 目前不支援跨帳戶存取控制。

AWS亞馬遜海王星的受管 (預先定義) 政策

AWS 透過提供獨立的 IAM 政策來解決許多常用案例，這些政策由 所建立與管理。AWS受管政策授與常見使用案例中必要的許可，讓您免於查詢需要哪些許可。如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

您可以將下列AWS受管政策附加到帳戶中的使用者，適用於使用 Amazon Neptune 管理 API：

- NeptuneReadOnlyAccess (p. 180)— 授予對所有 Neptune 資源的唯讀存取權，以用於根AWS帳戶中的管理和資料存取目的。
- NeptuneFullAccess (p. 181)— 授予對所有 Neptune 資源的完整存取權，以用於根AWS帳戶中的管理和資料存取目的。如果您需要從AWS CLI或 SDK 進行完整的 Neptune 存取權，但不能存AWS Management Console取，則建議使用此選項。
- NeptuneConsoleFullAccess (p. 184)— 授與根AWS帳戶中的所有 Neptune 管理動作和資源的完整存取權，但不授與任何資料存取動作或資源。它還包括其他許可，可簡化從主控台進行海王星存取，包括有限的 IAM 和 Amazon EC2 (VPC) 許可。

海王星 IAM 角色和政策授予某些對 Amazon RDS 資源的存取權，因為海王星與 Amazon RDS 共用特定管理功能的操作技術。這包括管理 API 權限，這就是 Neptune 管理動作具有`rds:`前置詞的原因。

Neptune 受AWS管原則的更新

下表追蹤 Neptune 管理原則的更新，從 Neptune 開始追蹤這些變更的時間開始：

政策	描述	日期
NeptuneReadOnlyAccess (p. 180) （添加權限）	為新的全域資料庫 API 新增資料存取權限和權限。	2022-08-03
NeptuneFullAccess (p. 181) （添加權限）	為新的全域資料庫 API 新增資料存取權限和權限。	2022-07-28
NeptuneConsoleFullAccess (p. 184) （添加權限）	新增全域資料庫 API 的權限。	2022-07-21
海王星開始追蹤變更	Neptune 開始追蹤其受AWS管理原則的變更。	2022-07-21

NeptuneReadOnlyAccess 受管 AWS 政策

以下NeptuneReadOnlyAccess受管理政策授予對所有 Neptune 動作和資源的唯讀存取權，以用於管理和資料存取目的。

> **Note**
>
> 此原則已於 2022-07-21 上更新，包含唯讀資料存取權限以及唯讀管理權限，並包含全域資料庫動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeGlobalClusters",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DownloadDBLogFilePortion",
        "rds:ListTagsForResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "kms:ListKeys",
        "kms:ListRetirableGrants",
        "kms:ListAliases",
        "kms:ListKeyPolicies"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogStreams",
        "logs:GetLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/rds/*:log-stream:*",
        "arn:aws:logs:*:*:log-group:/aws/neptune/*:log-stream:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:Read*",
        "neptune-db:Get*",
        "neptune-db:List*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

NeptuneFullAccess 受管 AWS 政策

以下NeptuneFullAccess受管理策略授予對所有 Neptune 動作和資源的完整存取權，以用於管理和資料存取目的。如果您需要從 SDK AWS CLI 或從 SDK 完整存取權，但不需要從AWS Management Console.

> **Note**
>
> 此原則已於 2022-07-21 上更新，包含完整的資料存取權限以及完整的管理權限，以及包含全域資料庫動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBCluster",
        "rds:CreateDBInstance"
      ],
      "Resource": [
        "arn:aws:rds:*:*:*"
      ],
      "Condition": {
        "StringEquals": {
          "rds:DatabaseEngine": ["graphdb","neptune"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:AddRoleToDBCluster",
```

```
        "rds:AddSourceIdentifierToSubscription",
        "rds:AddTagsToResource",
        "rds:ApplyPendingMaintenanceAction",
        "rds:CopyDBClusterParameterGroup",
        "rds:CopyDBClusterSnapshot",
        "rds:CopyDBParameterGroup",
        "rds:CreateDBClusterEndpoint",
        "rds:CreateDBClusterParameterGroup",
        "rds:CreateDBClusterSnapshot",
        "rds:CreateDBParameterGroup",
        "rds:CreateDBSubnetGroup",
        "rds:CreateEventSubscription",
        "rds:CreateGlobalCluster",
        "rds:DeleteDBCluster",
        "rds:DeleteDBClusterEndpoint",
        "rds:DeleteDBClusterParameterGroup",
        "rds:DeleteDBClusterSnapshot",
        "rds:DeleteDBInstance",
        "rds:DeleteDBParameterGroup",
        "rds:DeleteDBSubnetGroup",
        "rds:DeleteEventSubscription",
        "rds:DeleteGlobalCluster",
        "rds:DescribeDBClusterEndpoints",
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSecurityGroups",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEngineDefaultClusterParameters",
        "rds:DescribeEngineDefaultParameters",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeGlobalClusters",
        "rds:DescribeOptionGroups",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DescribeValidDBInstanceModifications",
        "rds:DownloadDBLogFilePortion",
        "rds:FailoverDBCluster",
        "rds:FailoverGlobalCluster",
        "rds:ListTagsForResource",
        "rds:ModifyDBCluster",
        "rds:ModifyDBClusterEndpoint",
        "rds:ModifyDBClusterParameterGroup",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyDBSubnetGroup",
        "rds:ModifyEventSubscription",
        "rds:ModifyGlobalCluster",
        "rds:PromoteReadReplicaDBCluster",
        "rds:RebootDBInstance",
        "rds:RemoveFromGlobalCluster",
        "rds:RemoveRoleFromDBCluster",
        "rds:RemoveSourceIdentifierFromSubscription",
        "rds:RemoveTagsFromResource",
```

```
        "rds:ResetDBClusterParameterGroup",
        "rds:ResetDBParameterGroup",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:RestoreDBClusterToPointInTime",
        "rds:StartDBCluster",
        "rds:StopDBCluster"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcs",
        "kms:ListAliases",
        "kms:ListKeyPolicies",
        "kms:ListKeys",
        "kms:ListRetirableGrants",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "sns:ListSubscriptions",
        "sns:ListTopics",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:passedToService": "rds.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "rds.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:*"
      ],
      "Resource": [
        "*"
      ]
    }
```

```
  ]
}
```

NeptuneConsoleFullAccess 受管 AWS 政策

出於NeptuneConsoleFullAccess管理目的，以下受管理策略授予對所有 Neptune 操作和資源的完整訪問權限，但不能用於數據訪問目的。它還包括其他許可，可簡化從主控台進行海王星存取，包括有限的 IAM 和 Amazon EC2 (VPC) 許可。

Note

此原則已於 2022-07-21 上更新，以包含全域資料庫動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:CreateDBCluster",
        "rds:CreateDBInstance"
      ],
      "Resource": [
        "arn:aws:rds:*:*:*"
      ],
      "Condition": {
        "StringEquals": {
          "rds:DatabaseEngine": ["graphdb","neptune"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:AddRoleToDBCluster",
        "rds:AddSourceIdentifierToSubscription",
        "rds:AddTagsToResource",
        "rds:ApplyPendingMaintenanceAction",
        "rds:CopyDBClusterParameterGroup",
        "rds:CopyDBClusterSnapshot",
        "rds:CopyDBParameterGroup",
        "rds:CreateDBClusterEndpoint",
        "rds:CreateDBClusterParameterGroup",
        "rds:CreateDBClusterSnapshot",
        "rds:CreateDBParameterGroup",
        "rds:CreateDBSubnetGroup",
        "rds:CreateEventSubscription",
        "rds:CreateGlobalCluster",
        "rds:DeleteDBCluster",
        "rds:DeleteDBClusterEndpoint",
        "rds:DeleteDBClusterParameterGroup",
        "rds:DeleteDBClusterSnapshot",
        "rds:DeleteDBInstance",
        "rds:DeleteDBParameterGroup",
        "rds:DeleteDBSubnetGroup",
        "rds:DeleteEventSubscription",
        "rds:DeleteGlobalCluster",
        "rds:DescribeDBClusterEndpoints",
        "rds:DescribeAccountAttributes",
        "rds:DescribeCertificates",
        "rds:DescribeDBClusterParameterGroups",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeDBClusterSnapshotAttributes",
        "rds:DescribeDBClusterSnapshots",
```

```
        "rds:DescribeDBClusters",
        "rds:DescribeDBEngineVersions",
        "rds:DescribeDBInstances",
        "rds:DescribeDBLogFiles",
        "rds:DescribeDBParameterGroups",
        "rds:DescribeDBParameters",
        "rds:DescribeDBSecurityGroups",
        "rds:DescribeDBSubnetGroups",
        "rds:DescribeEngineDefaultClusterParameters",
        "rds:DescribeEngineDefaultParameters",
        "rds:DescribeEventCategories",
        "rds:DescribeEventSubscriptions",
        "rds:DescribeEvents",
        "rds:DescribeGlobalClusters",
        "rds:DescribeOptionGroups",
        "rds:DescribeOrderableDBInstanceOptions",
        "rds:DescribePendingMaintenanceActions",
        "rds:DescribeValidDBInstanceModifications",
        "rds:DownloadDBLogFilePortion",
        "rds:FailoverDBCluster",
        "rds:FailoverGlobalCluster",
        "rds:ListTagsForResource",
        "rds:ModifyDBCluster",
        "rds:ModifyDBClusterEndpoint",
        "rds:ModifyDBClusterParameterGroup",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds:ModifyDBInstance",
        "rds:ModifyDBParameterGroup",
        "rds:ModifyDBSubnetGroup",
        "rds:ModifyEventSubscription",
        "rds:ModifyGlobalCluster",
        "rds:PromoteReadReplicaDBCluster",
        "rds:RebootDBInstance",
        "rds:RemoveFromGlobalCluster",
        "rds:RemoveRoleFromDBCluster",
        "rds:RemoveSourceIdentifierFromSubscription",
        "rds:RemoveTagsFromResource",
        "rds:ResetDBClusterParameterGroup",
        "rds:ResetDBParameterGroup",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:RestoreDBClusterToPointInTime",
        "rds:StartDBCluster",
        "rds:StopDBCluster"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "ec2:AllocateAddress",
        "ec2:AssignIpv6Addresses",
        "ec2:AssignPrivateIpAddresses",
        "ec2:AssociateAddress",
        "ec2:AssociateRouteTable",
        "ec2:AssociateSubnetCidrBlock",
        "ec2:AssociateVpcCidrBlock",
        "ec2:AttachInternetGateway",
        "ec2:AttachNetworkInterface",
        "ec2:CreateCustomerGateway",
        "ec2:CreateDefaultSubnet",
        "ec2:CreateDefaultVpc",
        "ec2:CreateInternetGateway",
```

```
        "ec2:CreateNatGateway",
        "ec2:CreateNetworkInterface",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSubnet",
        "ec2:CreateVpc",
        "ec2:CreateVpcEndpoint",
        "ec2:CreateVpcEndpoint",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeCustomerGateways",
        "ec2:DescribeInstances",
        "ec2:DescribeNatGateways",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribePrefixLists",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroupReferences",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcs",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifySubnetAttribute",
        "ec2:ModifyVpcAttribute",
        "ec2:ModifyVpcEndpoint",
        "iam:ListRoles",
        "kms:ListAliases",
        "kms:ListKeyPolicies",
        "kms:ListKeys",
        "kms:ListRetirableGrants",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "sns:ListSubscriptions",
        "sns:ListTopics",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:passedToService": "rds.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
      "Condition": {
        "StringLike": {
```

```
          "iam:AWSServiceName": "rds.amazonaws.com"
        }
      }
    }
  ]
}
```

亞馬遜海王星支援的 IAM 條件內容金鑰

您可以在 IAM 政策中指定控制 Neptune 管理動作和資源存取的條件。然後，政策聲明只有在條件成立時才會生效。

例如，您可能希望政策聲明僅在特定日期後生效，或者只有在 API 請求中存在特定值時才允許訪問。

若要表示條件，您可以在政策陳述式的`Condition`元素中使用預先定義的條件索引鍵，以及 IAM 條件政策運算子 (例如 equals 或小於)。

若您在陳述式中指定多個 `Condition` 元素，或是在單一 `Condition` 元素中指定多個金鑰，AWS 會使用邏輯 AND 操作評估他們。若您為單一條件索引鍵指定多個值，AWS 會使用邏輯 OR 操作評估條件。必須符合所有條件，才會授予陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

條件索引鍵的資料類型會決定您可以使用哪些條件運算子來比較要求中的值與原則陳述式中的值。如果您使用與該資料類型不相容的條件運算子，相符項目一律會失敗，而且不會套用原則陳述式。

海王星管理政策陳述式的 IAM 條件金鑰

- 全域條件金鑰 — 您可以在 Neptune 管理原則陳述式中使用大部分的AWS全域條件金鑰。
- 服務特定條件金鑰 (p. 199) — 這些是針對特定AWS服務定義的金鑰。中列出了 Neptune 支援的系統管理原則陳述式Neptune IAM 管理政策陳述式中提供的條件金鑰 (p. 199)。

Neptune 資料存取政策陳述式的 IAM 條件金鑰

- 全域條件索引鍵 — Neptune 在資料存取原則陳述式中所支援的這些金鑰子集列在中AWSNeptune 在資料存取原則陳述式中支援的全域條件內容金鑰 (p. 210)。
- Neptune 針對資料存取原則陳述式定義的服務特定條件金鑰列於中。條件索引鍵 (p. 210)

亞馬遜海王星的自訂 IAM 管理政策聲明

系統管理政策陳述式可讓您控制 IAM 使用者可以執行哪些動作來管理 Neptune 資料庫。

Neptune 系統管理原則陳述式會授與 Neptune 支援的一或多個管理動作 (p. 188)和系統管理資源 (p. 198)的存取權。您也可以使用條件索引鍵 (p. 199)讓系統管理權限更具體。

> **Note**
>
> 由於 Neptune 與 Amazon RDS 共用功能，因此管理政策陳述式中的管理動作、資源和服務特定條件金鑰會依設計使用`rds:`前置詞。

主題

- Neptune IAM 管理政策陳述式中可用的動作 (p. 188)
- IAM Neptune 管理政策聲明中提供的資源類型 (p. 198)
- Neptune IAM 管理政策陳述式中提供的條件金鑰 (p. 199)

- 海王星的 IAM 管理政策聲明範例 (p. 201)

Neptune IAM 管理政策陳述式中可用的動作

您可以使用 IAM 政策陳述式`Action`元素中列出的管理動作來控制 Neptune 管理 API (p. 996) 的存取。在政策中使用動作時，通常會允許或拒絕存取相同名稱的 API 操作或 CLI 命令。不過，在某些情況下，單一動作可控制對多個操作的存取。或者，某些操作需要多種不同的動作。

下列清單中的`Resource type`欄位指出每個動作是否支援資源層級權限。如果此欄位中沒有值，您必須在政策陳述式的`Resource`元素中指定所有資源（「*」）。如果資料行包含資源類型，則您可以使用該動作在陳述式中指定該類型的資源 ARN。此頁 (p. 198)面列出 Neptune 管理資源類型。

下列清單中會以星號 (*) 表示必要資源。如果您在使用此動作的陳述式中指定資源層級許可 ARN，則它必須屬於此類型。某些動作支援多種資源類型。如果資源類型是可選的（換句話說，沒有標記星號），則不必包含它。

如需此處所列欄位的詳細資訊，請參閱 IAM 使用者指南中的動作表。

rds：AddRoleTo數據庫集群

`AddRoleTo##### (p. 1031)`將 IAM 角色與海王星資料庫叢集建立關聯。

存取層級：`Write`。

相依動作：`iam:PassRole`。

資源類型:叢集 (p. 198) (必要)。

RDS：AddSourceIdentifierToSubscription

`AddSourceIdentifierToSubscription (p. 1142)`將來源識別碼新增至現有的 Neptune 事件通知訂閱。

存取層級：`Write`。

資源類型:es (p. 198) (必填)。

RDS：AddTagsToResource

`AddTagsToResource (p. 1151)`將 IAM 角色與海王星資料庫叢集建立關聯。

存取層級：`Write`。

資源類型：

- 資料庫 (p. 198)
- es (p. 198)
- PG (p. 199)
- 叢集快照 (p. 198)
- 副格柵 (p. 199)

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS：ApplyPendingMaintenanceAction

ApplyPendingMaintenanceAction (p. 1152)將擱置的維護動作套用至資源。

存取層級：Write。

資源類型:db (p. 198) (必要)。

RDS: 複製資料庫 ClusterParameterGroup

#####ClusterParameterGroup (p. 1088)複製指定的資料庫叢集參數群組。

存取層級：Write。

資源類型:叢集-pg (p. 198) (必要)。

RDS: 複製資料庫 ClusterSnapshot

ClusterSnapshot (p. 1115)複製一個數據庫集群的快照。

存取層級：Write。

資源類型:叢集-快照 (p. 198) (必要)。

RDS: 複製資料庫 ParameterGroup

#####ParameterGroup (p. 1087)複製指定的 DB 參數群組。

存取層級：Write。

資源類型:pg (p. 199) (必填).

RDS: 建立資料庫叢集

CreateDBCluster (p. 1005)建立新的海王星資料庫叢集。

存取層級：Tagging。

相依動作：iam:PassRole。

資源類型：

- 叢集 (p. 198) (必要)。
- 叢集-pg (p. 198) (必要)。
- 子格柵 (p. 199) (必需) 。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)
- 海王星-RDS_ DatabaseEngine (p. 199)

RDS: 建立資料庫 ClusterParameterGroup

#####ClusterParameterGroup (p. 1091)建立新的資料庫叢集參數群組。

存取層級：Tagging。

資源類型:叢集-pg (p. 198) (必要)。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 建立資料庫 ClusterSnapshot

ClusterSnapshot (p. 1110)創建一個數據庫集群的快照。

存取層級：Tagging。

資源類型：

- 叢集 (p. 198) (必要)。
- 叢集快照 (p. 198) (必要)。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 建立資料庫例證

CreateDBInstance (p. 1056)建立新的資料庫執行個體。

存取層級：Tagging。

相依動作：iam:PassRole。

資源類型：

- 分貝 (p. 198) (必需) 。
- pg (p. 199) (必填)。
- 子格柵 (p. 199) (必需) 。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 建立資料庫 ParameterGroup

#####ParameterGroup (p. 1090) 會建立新的資料庫參數群組。

存取層級：Tagging。

資源類型:pg (p. 199) (必填).

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 建立資料庫 SubnetGroup

#####SubnetGroup (p. 1104)建立新的資料庫子網路群組。

存取層級：Tagging。

資源類型:子格 (p. 199)式 (必要)。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS：CreateEventSubscription

CreateEventSubscription (p. 1136)建立海王星事件通知訂閱。

存取層級：Tagging。

資源類型:es (p. 198) (必填)。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 刪除資料庫叢集

DeleteDBCluster (p. 1012)刪除現有的海王星資料庫叢集。

存取層級：Write。

資源類型：

- 叢集 (p. 198) (必要)。
- 叢集快照 (p. 198) (必要)。

RD: 刪除資料庫 ClusterParameterGroup

#####ClusterParameterGroup (p. 1093)刪除指定的資料庫叢集參數群組。

存取層級：Write。

資源類型:叢集-pg (p. 198) (必要)。

RD: 刪除資料庫 ClusterSnapshot

ClusterSnapshot (p. 1113)刪除數據庫集群快照。

存取層級：Write。

資源類型:叢集-快照 (p. 198) (必要)。

RDS: 刪除資料庫例證

DeleteDBInstance (p. 1065)刪除指定的資料庫執行個體。

存取層級：Write。

資源類型:db (p. 198) (必要)。

RD: 刪除資料庫 ParameterGroup

#####ParameterGroup (p. 1092)刪除指定的數據庫ParameterGroup。

存取層級：Write。

資源類型:pg (p. 199) (必填).

RD: 刪除資料庫 SubnetGroup

#####SubnetGroup (p. 1105)刪除資料庫子網路群組。

存取層級：Write。

資源類型:子格 (p. 199)式 (必要)。

RDS：DeleteEventSubscription

DeleteEventSubscription (p. 1138)刪除事件通知訂閱。

存取層級：Write。

資源類型:es (p. 198) (必填)。

RDS: 描述 ClusterParameterGroups

BClusterParameterGroups (p. 1100)返回數據庫ClusterParameterGroup描述的列表。

存取層級：List。

資源類型:叢集-pg (p. 198) (必要)。

RDS: 描述 ClusterParameters

BClusterParameters (p. 1099)傳回特定資料庫叢集參數群組的詳細參數清單。

存取層級：List。

資源類型:叢集-pg (p. 198) (必要)。

RDS: 描述 ClusterSnapshotAttributes

B ClusterSnapshotAttributes (p. 1133)傳回手動資料庫叢集快照集的資料庫叢集快照屬性名稱和值的清單。

存取層級：List。

資源類型:叢集-快照 (p. 198) (必要)。

RDS: 描述 ClusterSnapshots

B ClusterSnapshots (p. 1131)返回有關數據庫集群快照的信息。

存取層級：Read。

rds:DescribeDBClusters

DescribeDBClusters (p. 1040)傳回已佈建 Neptune 資料庫叢集的相關資訊。

存取層級：List。

資源類型:叢集 (p. 198) (必要)。

RDS: 描述 EngineVersions

BEngineVersions (p. 1154)返回可用的數據庫引擎的列表。

存取層級：List。

資源類型:pg (p. 199) (必填).

RDS: 描述例證

DescribeDBInstances (p. 1082)返回有關數據庫實例的信息。

存取層級：List。

資源類型:es (p. 198) (必填)。

RDS: 描述 ParameterGroups

BParameterGroups (p. 1098)返回數據庫ParameterGroup描述的列表。

存取層級：List。

資源類型:pg (p. 199) (必填).

RDS: 描述參數

DescribeDBParameters (p. 1097)傳回特定資料庫參數群組的詳細參數清單。

存取層級：List。

資源類型:pg (p. 199) (必填).

RDS: 描述 SubnetGroups

BSubnetGroups (p. 1107)返回數據庫SubnetGroup描述的列表。

存取層級：List。

資源類型:子格 (p. 199)式 (必要)。

RDS：DescribeEventCategories

DescribeEventCategories (p. 1146)會傳回所有事件來源類型的類別清單，如果有指定，則傳回指定來源類型的類別清單。

存取層級：List。

RDS：DescribeEventSubscriptions

DescribeEventSubscriptions (p. 1141)列出客戶帳戶的所有訂閱說明。

存取層級：List。

資源類型:es (p. 198) (必填)。

RDS：DescribeEvents

DescribeEvents (p. 1144)傳回過去 14 天內與資料庫執行個體、資料庫安全群組和資料庫參數群組相關的事件。

存取層級：List。

資源類型:es (p. 198) (必填)。

RDS: DescribeOrderable 資料庫 InstanceOptions

DescribeOrderable###InstanceOptions (p. 1083)傳回指定引擎的可訂購資料庫執行個體選項清單。

存取層級：List。

RDS：DescribePendingMaintenanceActions

DescribePendingMaintenanceActions (p. 1153)傳回至少具有一個擱置維護動作的資源清單 (例如資料庫執行個體)。

存取層級：List。

資源類型:db (p. 198) (必要)。

RDS: DescribeValid 資料庫 InstanceModifications

DescribeValid###InstanceModifications (p. 1084)列出您可以對資料庫執行個體進行的可用修改。

存取層級：List。

資源類型:db (p. 198) (必要)。

RDS: 容錯移轉資料庫叢集

FailoverDBCluster (p. 1032)強制資料庫叢集的容錯移轉。

存取層級：Write。

資源類型:叢集 (p. 198) (必要)。

RDS：ListTagsForResource

ListTagsForResource (p. 1151)列出海王星資源上的所有標籤。

存取層級：Read。

資源類型：

- 叢集快照 (p. 198)
- 資料庫 (p. 198)
- es (p. 198)
- PG (p. 199)
- 副格柵 (p. 199)

RDS: 修改資料庫叢集

ModifyDBCluster (p. 1017)

修改 Neptune 資料庫叢集的設定。

存取層級：Write。

相依動作：iam:PassRole。

資源類型：

- 叢集 (p. 198) (必要)。
- 叢集-pg (p. 198) (必要)。

RDS: 修改資料庫 ClusterParameterGroup

#####ClusterParameterGroup (p. 1094)修改資料庫叢集參數群組的參數。

存取層級：Write。

資源類型:叢集-pg (p. 198) (必要)。

RDS: 修改資料庫 ClusterSnapshotAttribute

ClusterSnapshotAttribute (p. 1117)將屬性和值新增至手動資料庫叢集快照，或從中移除屬性和值。

存取層級：Write。

資源類型:叢集-快照 (p. 198) (必要)。

RDS: 修改資料庫例證

ModifyDBInstance (p. 1069)修改資料庫執行個體的設定。

存取層級：Write。

相依動作：iam:PassRole。

資源類型：

- 分貝 (p. 198) (必需) 。
- pg (p. 199) (必填)。

RDS: 修改資料庫 ParameterGroup

#####ParameterGroup (p. 1093)修改 DB 參數群組的參數。

存取層級：Write。

資源類型:pg (p. 199) (必填).

RDS: 修改資料庫 SubnetGroup

#####SubnetGroup (p. 1106)修改現有的資料庫子網路群組。

存取層級：Write。

資源類型:子格 (p. 199)式 (必要)。

RDS：ModifyEventSubscription

ModifyEventSubscription (p. 1139)修改現有的 Neptune 事件通知訂閱。

存取層級：Write。

資源類型:es (p. 198) (必填)。

RDS: 重新啟動資料庫例證

RebootDBInstance (p. 1078)重新啟動執行處理的資料庫引擎服務。

存取層級：Write。

資源類型:db (p. 198) (必要)。

rds：RemoveRoleFrom數據庫集群

RemoveRoleFrom##### (p. 1031)取消AWS身分識別和存取管理 (IAM) 角色與 Amazon Neptune 資料庫叢集的關聯性。

存取層級：Write。

相依動作：iam:PassRole。

資源類型:叢集 (p. 198) (必要)。

RDS：RemoveSourceIdentifierFromSubscription

RemoveSourceIdentifierFromSubscription (p. 1143)從現有 Neptune 事件通知訂閱中移除來源識別碼。

存取層級：Write。

資源類型:es (p. 198) (必填)。

RDS：RemoveTagsFromResource

RemoveTagsFromResource (p. 1152)從海王星資源中移除中繼資料標籤。

存取層級：Tagging。

資源類型：

- 叢集快照 (p. 198)
- 資料庫 (p. 198)
- es (p. 198)
- PG (p. 199)
- 副格柵 (p. 199)

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 重設資料庫 ClusterParameterGroup

#####ClusterParameterGroup (p. 1096)將資料庫叢集參數群組的參數修改為預設值。

存取層級：Write。

資源類型:叢集-pg (p. 198) (必要)。

RDS: 重設資料庫 ParameterGroup

#####ParameterGroup (p. 1095)將 DB 參數群組的參數修改為引擎/系統預設值。

存取層級：Write。

資源類型:pg (p. 199) (必填).

RDS: 恢復 ClusterFromSnapshot

ClusterFromSnapshot (p. 1119)從資料庫叢集快照建立新的資料庫叢集。

存取層級：Write。

相依動作：iam:PassRole。

資源類型：

- 叢集 (p. 198) (必要)。
- 叢集快照 (p. 198) (必要)。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 恢復 ClusterToPointInTime

ClusterToPointInTime (p. 1125)將數據庫集群恢復到任意時間點。

存取層級：Write。

相依動作：iam:PassRole。

資源類型：

- 叢集 (p. 198) (必要)。
- 子格柵 (p. 199) (必需) 。

條件鍵：

- AWS：RequestTag/### (p. 200)
- AWS：TagKeys (p. 200)

RDS: 起始資料庫叢集

StartDBCluster (p. 1023)啟動指定的數據庫集群。

存取層級：Write。

資源類型:叢集 (p. 198) (必要)。

RDS: 停止資料庫叢集

StopDBCluster (p. 1027)停止指定的數據庫集群。

存取層級：Write。

資源類型:叢集 (p. 198) (必要)。

IAM Neptune 管理政策聲明中提供的資源類型

Neptune 支援下表中的資源類型，以用於 IAM 管理政策陳述式的Resource元素。如需 Resource 元素的詳細資訊，請參閱 IAM JSON 政策元素：Resource。

Neptune 管理動作清單會識別可透過每個動作指定的 (p. 188)資源類型。資源類型也會決定您可以在策略中包含哪些條件索引鍵，如下表最後一欄所指定。

下表中的ARN欄指定 Amazon 資源名稱 (ARN) 格式，您必須使用這種格式來參考此類型的資源。前面有 a 的部分 $ 必須由您的案例的實際值所取代。例如，如果您在 ARN $user-name 中看到，則必須以實際的 IAM 使用者名稱或包含 IAM 使用者名稱的政策變數來取代該字串。如需有關 ARN 的詳細資訊，請參閱 IAM ARN 和. 使用亞馬遜海王星的管理 ARN (p. 803)

此 Condition Keys 欄會指定條件內容索引鍵，只有當這個資源和相容的支援動作都包含在陳述式中時，才能包含在 IAM 政策陳述式中。

資源類型	ARN	條件索引鍵
cluster （數據庫集群）	arn:*partition*:rds:*region*:*account-id*:cluster:*instance-name*	AWS：ResourceTag/### (p. 200) *RDS: ####/###* (p. 200)
cluster-pg (資料庫叢集參數群組)	arn:*partition*:rds:*region*:*account-id*:cluster-pg:*neptune-DBClusterParameterGroupName*	AWS：ResourceTag/### (p. 200)
cluster-snapshot (資料庫叢集快照)	arn:*partition*:rds:*region*:*account-id*:cluster-snapshot:*neptune-DBClusterSnapshotName*	AWS：ResourceTag/### (p. 200) rds：cluster-snapshot-tag/### (p. 200)
db (資料庫執行個體)	arn:*partition*:rds:*region*:*account-id*:db:*neptune-DbInstanceName*	AWS：ResourceTag/### (p. 200) RDS：DatabaseClass (p. 199) RDS：DatabaseEngine (p. 199) *RDS: DB-##/###* (p. 200)
es （活動訂閱）	arn:*partition*:rds:*region*:*account-id*:es:*neptune-CustSubscriptionId*	AWS：ResourceTag/### (p. 200) *RDS: E-##/###* (p. 200)

資源類型	ARN	條件索引鍵
pg (資料庫參數群組)	arn:*partition*:rds:*region*:*account-id*:pg:*neptune-ParameterGroupName*	AWS：ResourceTag/## # (p. 200) *RDS: PG-##/## # (p. 200)*
subgrp (資料庫子網路群組)	arn:*partition*:rds:*region*:*account-id*:subgrp:*neptune-DBSubnetGroupName*}	AWS：ResourceTag/## # (p. 200) *RDS: #####/## # (p. 201)*

Neptune IAM 管理政策陳述式中提供的條件金鑰

您可以使用條件索引鍵 (p. 178)在 IAM 政策陳述式中指定條件，以便陳述式只在條件為真時生效。您可以在 Neptune 系統管理原則陳述式中使用的條件索引鍵分為下列類別：

- 全域條件金鑰 — 這些金鑰由定義，以AWS供AWS服務一般使用。大多數可用於 Neptune 系統管理原則陳述式中。
- 管理資源特性條件索引 (p. 199)鍵 — 下面 (p. 199)列出的這些鍵是以管理資源的特性為基礎。
- 以標籤為基礎的存取條件金 (p. 199)鑰 — 下面 (p. 199)列出的這些金鑰是以附加至管理資源的AWS標籤為基礎。

海王星管理資源屬性條件鍵

條件索引鍵	描述	類型
rds:DatabaseClass	依資料庫執行個體類別類型篩選存取。	字串
rds:DatabaseEngine	依資料庫引擎來篩選存取權限。如需可能的值，請參閱建立資料庫執行個體 API 中的引擎參數	字串
rds:DatabaseName	依資料庫執行個體上的資料庫使用者定義名稱來篩選存取權限。	字串
rds:EndpointType	依端點類型篩選存取權限。READER、WRITER、CUSTOM 的其中一個	字串
rds:Vpc	依此值指定資料庫執行個體是否在 Amazon Virtual Private Cloud (Amazon VPC) 中執行來篩選存取權限。若要指出資料庫執行個體在 Amazon VPC 中執行，請指定true。	Boolean

基於管理標籤的條件鍵

Amazon Neptune 支援使用自訂標籤在 IAM 政策中指定條件，以透過Neptune 管理 API (p. 996).

例如，如果您將名為的標籤新增environment至資料庫執行個體，其值如betastaging、和production，則可以建立一個政策，根據該標記的值限制執行個體的存取。

> **Important**
>
> 如果您使用標記來管理 Neptune 資源的存取權，請務必確保對標籤的存取安全。您可以針對AddTagsToResource和動作建立政策，以限制對標籤的存RemoveTagsFromResource取。

例如，您可以使用下列策略來拒絕使用者為所有資源新增或移除標籤的能力。然後，您可以建立策略以允許特定使用者新增或移除標籤。

```
{ "Version": "2012-10-17",
  "Statement":[
    { "Sid": "DenyTagUpdates",
      "Effect": "Deny",
      "Action": [
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource":"*"
    }
  ]
}
```

下列以標籤為基礎的條件金鑰僅適用於系統管理原則陳述式中的管理資源。

基於標籤的管理條件鍵

條件索引鍵	描述	類型
aws:RequestTag/${TagKey}	根據要求中的標籤索引鍵值配對來篩選存取。	字串
aws:ResourceTag/${TagKey}	根據附加到資源的標籤鍵值對篩選存取。	字串
aws:TagKeys	根據要求中的標籤金鑰存在篩選存取。	字串
rds:cluster-pg-tag/${TagKey}	依附加至資料庫叢集參數群組的標籤篩選存取。	字串
rds:cluster-snapshot-tag/${TagKey}	依附加至資料庫叢集快照的標籤篩選存取。	字串
rds:cluster-tag/${TagKey}	依附加至資料庫叢集的標籤篩選存取。	字串
rds:db-tag/${TagKey}	依附加至資料庫執行個體的標籤篩選存取。	字串
rds:es-tag/${TagKey}	依附加至事件訂閱的標籤篩選存取。	字串
rds:pg-tag/${TagKey}	依附加至資料庫參數群組的標籤篩選存取。	字串
rds:req-tag/${TagKey}	依可用來標記資源的一組標籤鍵和值篩選存取。	字串
rds:secgrp-tag/${TagKey}	依附加至資料庫安全群組的標籤篩選存取。	字串
rds:snapshot-tag/${TagKey}	通過附加到數據庫快照的標籤過濾訪問。	字串

條件索引鍵	描述	類型
rds:subgrp-tag/${TagKey}	依附加至資料庫子網路群組的標籤來篩選存取權限	字串

海王星的 IAM 管理政策聲明範例

一般管理政策範例

下列範例顯示如何建立 Neptune 系統管理原則，以授與對資料庫叢集執行各種管理動作的權限。

防止 IAM 使用者刪除指定資料庫執行個體的政策

以下是防止 IAM 使用者刪除指定 Neptune 資料庫執行個體的範例政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyDeleteOneInstance",
      "Effect": "Deny",
      "Action": "rds:DeleteDBInstance",
      "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-instance-name"
    }
  ]
}
```

授予建立新資料庫執行個體權限的政策

以下是允許 IAM 使用者在指定的 Neptune 資料庫叢集中建立資料庫執行個體的範例政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateInstance",
      "Effect": "Allow",
      "Action": "rds:CreateDBInstance",
      "Resource": "arn:aws:rds:us-west-2:123456789012:cluster:my-cluster"
    }
  ]
}
```

授與使用特定資料庫參數群組建立新資料庫執行個體之權限的原則

以下範例政策允許 IAM 使用者僅使用指定的資料庫參數群組，在指定的 Neptune 資料庫叢集 (此處us-west-2) 中建立資料庫執行個體。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateInstanceWithPG",
      "Effect": "Allow",
      "Action": "rds:CreateDBInstance",
      "Resource": [
        "arn:aws:rds:us-west-2:123456789012:cluster:my-cluster",
        "arn:aws:rds:us-west-2:123456789012:pg:my-instance-pg"
      ]
    }
```

```
  ]
}
```

授予描述任何資源權限的策略

以下是允許 IAM 使用者描述任何 Neptune 資源的範例政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDescribe",
      "Effect": "Allow",
      "Action": "rds:Describe*",
      "Resource": *
    }
  ]
}
```

基於標籤的管理策略範例

下列範例顯示如何建立 Neptune 系統管理原則，以標記以篩選資料庫叢集上各種管理動作的權限。

範例 1：使用可採用多個值的自訂標籤，授予對資源動作的權限

下列政策允許在env標籤設定為CreateDBInstance或的任何資料庫執行個體上使用ModifyDBInstance、dev或 DeleteDBInstance APItest：

```
{ "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDevTestAccess",
      "Effect": "Allow",
      "Action": [
        "rds:ModifyDBInstance",
        "rds:CreateDBInstance",
        "rds:DeleteDBInstance"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "rds:db-tag/env": [
              "dev",
              "test"
          ],
          "rds:DatabaseEngine": "neptune"
        }
      }
    }
  ]
}
```

範例 2：限制可用來標記資源的標籤鍵和值集

此原則會使用Condition金鑰來允許將索引鍵env和值為testqa、或dev的標籤新增至資源：

```
{ "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTagAccessForDevResources",
      "Effect": "Allow",
      "Action": [
```

```
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "rds:req-tag/env": [
            "test",
            "qa",
            "dev"
          ],
          "rds:DatabaseEngine": "neptune"
        }
      }
    }
  ]
}
```

範例 3：允許根據以下項目對 Neptune 資源的完整存取 aws:ResourceTag

下列原則與上述第一個範例類似，但會aws:ResourceTag改用：

```
{ "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFullAccessToDev",
      "Effect": "Allow",
      "Action": [
        "rds:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/env": "dev",
          "rds:DatabaseEngine": "neptune"
        }
      }
    }
  ]
}
```

亞馬遜海王星的自訂 IAM 資料存取政策聲明

Neptune 資料存取原則陳述式使用資料存取動作 (p. 205)、資源 (p. 209)和條件索引鍵 (p. 211)，所有這些索引鍵前面都會加上前置詞。neptune-db:

主題

- 使用 Neptune 資料存取原則陳述式的查詢動作 (p. 203)
- Neptune IAM 資料存取政策陳述式中可用的動作 (p. 205)
- 在 Neptune IAM 資料存取政策陳述式中指定資源 (p. 209)
- Neptune IAM 資料存取政策陳述式中提供的條件金鑰 (p. 210)
- 海王星 IAM 資料存取政策的範例 (p. 211)

使用 Neptune 資料存取原則陳述式的查詢動作

有三個 Neptune 查詢動作可用於資料存取原則陳述式，亦即ReadDataViaQueryWriteDataViaQuery、和DeleteDataViaQuery。特定查詢可能需要執行多個這些動作的權限，而且可能並不總是很明顯，必須允許哪些動作組合才能執行查詢。

在執行查詢之前，Neptune 會決定執行查詢每個步驟所需的權限，並將這些權限合併到查詢所需的完整權限集中。請注意，這組完整的權限包括查詢可能會執行的所有動作，這不一定是查詢在資料上執行時實際執行的動作集。

這表示若要允許指定的查詢執行，您必須為查詢可能執行的每個動作提供權限，不論查詢是否實際執行這些動作。

以下是一些示例 Gemlin 查詢，其中更詳細地解釋了這一點：

- ```
  g.V().count()
  ```

  g.V()並且count()只需要讀取訪問權限，因此整體查詢只需要ReadDataViaQuery訪問權限。

- ```
  g.addV()
  ```

  addV()需要在插入新頂點之前檢查具有給定 ID 的頂點是否存在。這意味著它需要ReadDataViaQuery和WriteDataViaQuery訪問。

- ```
  g.V('1').as('a').out('created').addE('createdBy').to('a')
  ```

  g.V('1').as('a')並且out('created')只需要讀取訪問權限，但addE().from('a')需要讀取和寫入訪問權限，因為addE()需要讀取from和to頂點並檢查具有相同 ID 的邊緣是否已經存在，然後再添加新的邊緣。因此，查詢作為一個整體需要ReadDataViaQuery和WriteDataViaQuery訪問。

- ```
  g.V().drop()
  ```

  g.V()只需要讀取權限。 drop()需要讀取和刪除訪問權限，因為它需要在刪除頂點或邊緣之前讀取它，因此整體查詢需要ReadDataViaQuery和DeleteDataViaQuery訪問。

- ```
  g.V('1').property(single, 'key1', 'value1')
  ```

  g.V('1')只需要讀取權限，但property(single, 'key1', 'value1')需要讀取、寫入和刪除存取權限。在這裡，property()步驟插入鍵和值，如果它們不存在於頂點，但如果它們已經存在，它會刪除現有的屬性值，並在其位置插入一個新值。因此，整個查詢需要ReadDataViaQueryWriteDataViaQuery、和DeleteDataViaQuery存取。

  任何包含property()步驟的查詢都需要ReadDataViaQueryWriteDataViaQuery、和DeleteDataViaQuery權限。

以下是一些開放密碼的例子：

- ```
  MATCH (n)
  RETURN n
  ```

  此查詢讀取數據庫中的所有節點並返回它們，這只需要ReadDataViaQuery訪問。

- ```
  MATCH (n:Person)
  SET n.dept = 'AWS'
  ```

  此查詢需要ReadDataViaQueryWriteDataViaQuery、和DeleteDataViaQuery存取權。它讀取標籤為「Person」的所有節點，並AWS向其添加一個帶有鍵dept和值的新屬性，或者如果該dept屬性已經存在，則刪除舊值並插入AWS代替。此外，如果要設定的值為null，則會完全SET刪除屬性。

  由於子SET句在某些情況下可能需要刪除現有值，因此它始終需要DeleteDataViaQuery權限以及ReadDataViaQuery和WriteDataViaQuery權限。
  ```

- ```
  MATCH (n:Person)
  DETACH DELETE n
  ```

  此查詢需要ReadDataViaQuery和DeleteDataViaQuery權限。它發現與標籤的所有節點，Person並與連接到這些節點和任何相關的標籤和屬性的邊緣一起刪除它們。

- ```
  MERGE (n:Person {name: 'John'})-[:knows]->(:Person {name: 'Peter'})
  RETURN n
  ```

  此查詢需要ReadDataViaQuery和WriteDataViaQuery權限。該MERGE子句要么匹配指定的模式或創建它。因為，如果模式不匹配，則可能會發生寫入，因此需要寫入權限以及讀取權限。

Neptune IAM 資料存取政策陳述式中可用的動作

請注意，Neptune 資料存取動作具有前置詞neptune-db:，而 Neptune 中的系統管理動作則具有前置詞rds:。

IAM 中資料資源的 Amazon 資源名稱 (ARN) 與建立時指派給叢集的 ARN 不同。您必須建構 ARN，如指定資料資源 (p. 209)所示。此類資料資源 ARN 可以使用萬用字元來包含多個資源。

資料存取原則陳述式也可以包含 neptune-db: QueryLanguage (p. 211) 條件索引鍵，以便依據查詢語言限制存取。

從開始已發行版本: (p. 884)，Neptune 支援將權限限制為一或多個特定 Neptune 動作。這提供了比以前更精細的訪問控制。

Important

- IAM 政策的變更最多需要 10 分鐘才能套用到指定的 Neptune 資源。
- 套用至 Neptune 資料庫叢集的 IAM 政策適用於該叢集中的所有執行個體。

基於查詢的資料存取動作

Note

執行指定查詢需要哪些權限並不總是很明顯，因為查詢可能會根據處理的資料採取多個動作。如需詳細資訊，請參閱 使用查詢動作 (p. 203)。

neptune-db:ReadDataViaQuery

ReadDataViaQuery允許使用者透過提交查詢從 Neptune 資料庫讀取資料。

動作群組：唯讀、讀寫。

動作內容索引鍵：neptune-db:QueryLanguage。

所需資源：資料庫。

neptune-db:WriteDataViaQuery

WriteDataViaQuery允許使用者透過提交查詢將資料寫入 Neptune 資料庫。

動作群組：讀寫。

動作內容索引鍵：neptune-db:QueryLanguage。

所需資源：資料庫。

neptune-db:DeleteDataViaQuery

DeleteDataViaQuery允許使用者透過提交查詢來刪除 Neptune 資料庫中的資料。

動作群組：讀寫。

動作內容索引鍵：neptune-db:QueryLanguage。

所需資源：資料庫。

neptune-db:GetQueryStatus

GetQueryStatus允許使用者檢查所有使用中查詢的狀態。

動作群組：唯讀、讀寫。

動作內容索引鍵：neptune-db:QueryLanguage。

所需資源：資料庫。

neptune-db:GetStreamRecords

GetStreamRecords允許用戶從海王星獲取流記錄。

動作群組：讀寫。

動作內容索引鍵：neptune-db:QueryLanguage。

所需資源：資料庫。

neptune-db:CancelQuery

CancelQuery允許使用者取消查詢。

動作群組：讀寫。

所需資源：資料庫。

一般資料存取動作

neptune-db:GetEngineStatus

GetEngineStatus允許使用者檢查 Neptune 引擎的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:GetStatisticsStatus

GetStatisticsStatus可讓使用者檢查為資料庫收集之統計資料的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:GetGraphSummary

GetGraphSummary圖形摘要 API 可讓您擷取圖形的唯讀摘要。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:ManageStatistics

ManageStatistics可讓使用者管理資料庫的統計資料集合。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:DeleteStatistics

DeleteStatistics允許用戶刪除數據庫中的所有統計信息。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:ResetDatabase

ResetDatabase允許使用者取得重設所需的權杖，並重設 Neptune 資料庫。

動作群組：讀寫。

所需資源：資料庫。

大量載入器資料存取動作

neptune-db:StartLoaderJob

StartLoaderJob允許使用者啟動大量載入器工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:GetLoaderJobStatus

GetLoaderJobStatus可讓使用者檢查大量載入器工作的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:ListLoaderJobs

ListLoaderJobs允許使用者列出所有大量載入器工作。

動作群組：僅清單、唯讀、讀寫。

所需資源：資料庫。

neptune-db:CancelLoaderJob

CancelLoaderJob允許使用者取消載入器工作。

動作群組：讀寫。

所需資源：資料庫。

機器學習資料存取動作

neptune-db:StartMLDataProcessingJob

StartMLDataProcessingJob允許使用者啟動 Neptune ML 資料處理工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:StartMLModelTrainingJob

StartMLModelTrainingJob允許使用者啟動 ML 模型訓練工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:StartMLModelTransformJob

StartMLModelTransformJob允許使用者啟動 ML 模型轉換工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:CreateMLEndpoint

CreateMLEndpoint允許使用者建立海王星 ML 端點。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:GetMLDataProcessingJobStatus

GetMLDataProcessingJobStatus允許使用者檢查 Neptune ML 資料處理工作的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:GetMLModelTrainingJobStatus

GetMLModelTrainingJobStatus可讓使用者檢查 Neptune ML 模型訓練工作的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:GetMLModelTransformJobStatus

GetMLModelTransformJobStatus允許使用者檢查 Neptune ML 模型轉換工作的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:GetMLEndpointStatus

GetMLEndpointStatus允許使用者檢查 Neptune ML 端點的狀態。

動作群組：唯讀、讀寫。

所需資源：資料庫。

neptune-db:ListMLDataProcessingJobs

ListMLDataProcessingJobs允許使用者列出所有 Neptune ML 資料處理工作。

動作群組：僅清單、唯讀、讀寫。

所需資源：資料庫。

neptune-db:ListMLModelTrainingJobs

ListMLModelTrainingJobs允許使用者列出所有 Neptune ML 模型訓練工作。

動作群組：僅清單、唯讀、讀寫。

所需資源：資料庫。

neptune-db:ListMLModelTransformJobs

ListMLModelTransformJobs可讓使用者列出所有 ML 模型轉換工作。

動作群組：僅清單、唯讀、讀寫。

所需資源：資料庫。

neptune-db:ListMLEndpoints

ListMLEndpoints允許使用者列出所有 Neptune ML 端點。

動作群組：僅清單、唯讀、讀寫。

所需資源：資料庫。

neptune-db:CancelMLDataProcessingJob

CancelMLDataProcessingJob允許使用者取消 Neptune ML 資料處理工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:CancelMLModelTrainingJob

CancelMLModelTrainingJob允許使用者取消 Neptune ML 模型訓練工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:CancelMLModelTransformJob

CancelMLModelTransformJob允許使用者取消 Neptune ML 模型轉換工作。

動作群組：讀寫。

所需資源：資料庫。

neptune-db:DeleteMLEndpoint

DeleteMLEndpoint允許使用者刪除海王星 ML 端點。

動作群組：讀寫。

所需資源：資料庫。

在 Neptune IAM 資料存取政策陳述式中指定資源

數據資源（如數據操作）具有neptune-db:前綴。

在 Neptune 資料存取原則中，您可以使用下列格式在 ARN 中指定要授予存取權的資料庫叢集：

```
arn:aws:neptune-db:region:account-id:cluster-resource-id/*
```

這樣的資源 ARN 包含以下幾部分：

- *region*是亞馬遜海王星資料庫叢集的AWS區域。
- *account-id*是資料庫叢集的AWS帳號。
- *cluster-resource-id* 是資料庫叢集的資源 ID。

  Important

  `cluster-resource-id` 與叢集識別符不同。若要在 Neptune 中尋找叢集資源識別碼AWS Management Console，請查看相關資料庫叢集的 [組態] 區段。

Neptune IAM 資料存取政策陳述式中提供的條件金鑰

您可以使用條件索引鍵 (p. 178)在 IAM 政策陳述式中指定條件，以便陳述式只在條件為真時生效。

您可以在 Neptune 資料存取原則陳述式中使用的條件索引鍵分為下列類別：

- 全域條件索引鍵 — Neptune 在資料存取原則陳述式中支援的AWS全域條件金鑰子集如下 (p. 210)所列。
- 服務特定條件金鑰 (p. 211) — 這些是 Neptune 專門用於資料存取原則陳述式所定義的金鑰。目前只有一個，neptune-db: QueryLanguage (p. 211)，它只有在使用特定的查詢語言時才授予訪問權限。

AWSNeptune 在資料存取原則陳述式中支援的全域條件內容金鑰

下表列出 Amazon Neptune 支援在資料存取政策陳述式中使用的AWS全域條件內容金鑰子集：

您可以在資料存取原則陳述式中使用的全域條件索引鍵

條件索引鍵	描述	類型
aws:CurrentTime	依請求的目前日期和時間篩選存取。	String
aws:EpochTime	按請求的日期和時間篩選存取，以 UNIX 紀元值表示。	Numeric
aws:PrincipalAccount	依要求主體所屬帳戶篩選存取。	String
aws:PrincipalArn	依提出要求的主體 ARN 篩選存取。	String
aws:PrincipalIsAWSService	只有當AWS服務主體直接進行呼叫時，才允許存取。	Boolean
aws:PrincipalOrgID	依據請求主參與者所屬「組織」中AWS組織的識別碼篩選存取。	String
aws:PrincipalOrgPaths	依照提出請求之主參與者的「AWS組織」路徑篩選存取。	String
aws:PrincipalTag	依附加至提出要求之主參與者的標籤篩選存取。	String
aws:PrincipalType	依提出要求的主體類型篩選存取。	String
aws:RequestedRegion	依要求中呼叫的AWS區域篩選存取。	String
aws:SecureTransport	只有在使用 SSL 傳送要求時，才允許存取。	Boolean
aws:SourceIp	依要求者的 IP 位址篩選存取。	String

條件索引鍵	描述	類型
aws:TokenIssueTime	依照發出臨時安全登入資料的日期和時間來篩選存取。	String
aws:UserAgent	依請求者的用戶端應用程式來篩選存取權。	String
aws:userid	依要求者的主要識別碼篩選存取。	String
aws:ViaAWSService	只有在AWS服務代表您提出要求時，才允許存取。	Boolean

海王星服務特定條件金鑰

Neptune 支援 IAM 政策的下列服務特定條件金鑰：

海王星服務特定條件金鑰

條件索引鍵	描述	類型
neptune-db:QueryLanguage	依使用的查詢語言篩選資料存取。 有效值為：gremlin、sparql 和 opencypher。 支援的動作有ReadDataViaQueryWriteDataViaQueryDeleteDataViaQuery、GetQueryStatus、、和CancelQuery。	String

海王星 IAM 資料存取政策的範例

下列範例說明如何建立自訂 IAM 政策，這些政策使用 Neptune 引擎版本 1.2.0.0 版 (p. 884)中引入的資料平面 API 和動作的精確存取控制。

允許不受限制地存取 Neptune 資料庫叢集中的資料的原則範例

下列範例政策允許 IAM 使用者使用 IAM 資料庫身份驗證連線到 Neptune 資料庫叢集，並使用 "*" 字元來比對所有可用的動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "neptune-db:*",
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

上述範例包含 Neptune IAM 身份驗證特有格式的資源 ARN。若要建構 ARN，請參閱指定資料資源 (p. 209)。請注意，用於 IAM 授權Resource的 ARN 與建立時指派給叢集的 ARN 不同。

允許唯讀存取 Neptune 資料庫叢集的原則範例

下列原則會授與 Neptune 資料庫叢集中資料的完整唯讀存取權限：

```
{
  "Version":"2012-10-17",
```

```
  "Statement":[
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:Read*",
        "neptune-db:Get*",
        "neptune-db:List*"
      ],
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

拒絕對 Neptune 資料庫叢集的所有存取權的原則範例

預設的 IAM 動作是拒絕存取資料庫叢集，除非授與Allow效果。但是，下列原則會拒絕特定AWS帳戶和區域對資料庫叢集的所有存取，然後該帳戶和區域的優先順序高於任何Allow影響。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "neptune-db:*",
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

政策範例透過查詢授與讀取存取權

下列原則僅授與使用查詢從 Neptune 資料庫叢集讀取的權限：

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "neptune-db:ReadDataViaQuery",
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

只允許小鬼查詢的政策範例

下列原則會使用neptune-db:QueryLanguage條件金鑰來授與僅使用 Gremlin 查詢語言查詢 Neptune 的權限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:ReadDataViaQuery",
        "neptune-db:WriteDataViaQuery",
```

```
        "neptune-db:DeleteDataViaQuery"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "neptune-db:QueryLanguage": "Gremlin"
        }
      }
    }
  ]
}
```

允許除 Neptune ML 模型管理以外的所有存取的原則範例

下列原則會授與 Neptune 圖形作業的完整存取權，但 Neptune ML 模型管理功能除外：

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:CancelLoaderJob",
        "neptune-db:CancelQuery",
        "neptune-db:DeleteDataViaQuery",
        "neptune-db:DeleteStatistics",
        "neptune-db:GetEngineStatus",
        "neptune-db:GetLoaderJobStatus",
        "neptune-db:GetQueryStatus",
        "neptune-db:GetStatisticsStatus",
        "neptune-db:GetStreamRecords",
        "neptune-db:ListLoaderJobs",
        "neptune-db:ManageStatistics",
        "neptune-db:ReadDataViaQuery",
        "neptune-db:ResetDatabase",
        "neptune-db:StartLoaderJob",
        "neptune-db:WriteDataViaQuery"
      ],
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

允許存取 Neptune ML 模型管理的原則範例

此原則會授與 Neptune ML 模型管理功能的存取權：

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:CancelMLDataProcessingJob",
        "neptune-db:CancelMLModelTrainingJob",
        "neptune-db:CancelMLModelTransformJob",
        "neptune-db:CreateMLEndpoint",
        "neptune-db:DeleteMLEndpoint",
        "neptune-db:GetMLDataProcessingJobStatus",
        "neptune-db:GetMLEndpointStatus",
        "neptune-db:GetMLModelTrainingJobStatus",
        "neptune-db:GetMLModelTransformJobStatus",
```

```
        "neptune-db:ListMLDataProcessingJobs",
        "neptune-db:ListMLEndpoints",
        "neptune-db:ListMLModelTrainingJobs",
        "neptune-db:ListMLModelTransformJobs",
        "neptune-db:StartMLDataProcessingJob",
        "neptune-db:StartMLModelTrainingJob",
        "neptune-db:StartMLModelTransformJob"
      ],
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

授與完整查詢存取權限的政策

下列原則會授與 Neptune 圖形查詢作業的完整存取權，但不會授與快速重設、串流、大量載入器、Neptune ML 模型管理等功能的完整存取權：

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:ReadDataViaQuery",
        "neptune-db:WriteDataViaQuery",
        "neptune-db:DeleteDataViaQuery",
        "neptune-db:GetEngineStatus",
        "neptune-db:GetQueryStatus",
        "neptune-db:CancelQuery"
      ],
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

政策範例僅授與 Gemlin 查詢的完整存取權限

下列原則會使用 Gremlin 查詢語言授與 Neptune 圖形查詢作業的完整存取權，但不會授與其他語言的查詢，也不會授與快速重設、串流、大量載入器、Neptune ML 模型管理等功能的完整存取權：

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect": "Allow",
      "Action": [
        "neptune-db:ReadDataViaQuery",
        "neptune-db:WriteDataViaQuery",
        "neptune-db:DeleteDataViaQuery",
        "neptune-db:GetEngineStatus",
        "neptune-db:GetQueryStatus",
        "neptune-db:CancelQuery"
      ],
      "Resource": [
        "arn:aws:neptune-db:us-east-1:123456789012:cluster-ABCD1234EFGH5678IJKL90MNOP/*"
      ],
      "Condition": {
        "StringEquals": {
           "neptune-db:QueryLanguage":"Gremlin"
```

```
        }
      }
    }
  ]
}
```

授與完整存取權限 (快速重設除外) 的原則

下列原則會授與 Neptune 資料庫叢集的完整存取權，但使用快速重設除外：

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "neptune-db:*",
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    },
    {
      "Effect": "Deny",
      "Action": "neptune-db:ResetDatabase",
      "Resource": "arn:aws:neptune-db:us-east-1:123456789012:cluster-
ABCD1234EFGH5678IJKL90MNOP/*"
    }
  ]
}
```

針對 Neptune 使用服務連結角色

亞馬遜海王星使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 Neptune 的唯一 IAM 角色類型。Neptune 會預先定義服務連結角色，並包含服務代表您呼叫其他服AWS務所需的所有權限。

> **Important**
>
> 對於某些管理功能，亞馬遜海王星使用與 Amazon RDS 共用的操作技術。這包括服務連結角色和管理 API 許可。

服務連結角色可讓您更輕鬆地使用 Neptune，因為您不必手動新增必要的權限。Neptune 會定義其服務連結角色的權限，除非另有定義，否則只有 Neptune 可以擔任其角色。定義的許可包括信任政策和許可政策，並且該許可政策不能附加到任何其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除角色。這可以保護您的 Neptune 資源，因為您無法不小心移除存取資源的權限。

如需其他支援服務連結角色之服AWS務的相關資訊，請參閱搭配 IAM 使用的服務，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的 Yes (是)，以檢視該服務的服務連結角色文件。

Neptune 的服務連結角色權限

Neptune 使用AWSServiceRoleForRDS服務連結角色，允許 Neptune 和 Amazon RDS 代表您的資料庫執行個體呼叫AWS服務。AWSServiceRoleForRDS 服務連結角色信任 rds.amazonaws.com 服務來擔任該角色。

角色權限原則可讓 Neptune 在指定的資源上完成下列動作：

- 在 ec2 上的動作：

  - AssignPrivateIpAddresses
  - AuthorizeSecurityGroupIngress
  - CreateNetworkInterface
  - CreateSecurityGroup
  - DeleteNetworkInterface
  - DeleteSecurityGroup
  - DescribeAvailabilityZones
  - DescribeInternetGateways
  - DescribeSecurityGroups
  - DescribeSubnets
  - DescribeVpcAttribute
  - DescribeVpcs
  - ModifyNetworkInterfaceAttribute
  - RevokeSecurityGroupIngress
  - UnassignPrivateIpAddresses
- 在 sns 上的動作：
  - ListTopic
  - Publish
- 在 cloudwatch 上的動作：
  - PutMetricData
  - GetMetricData
  - CreateLogStream
  - PullLogEvents
  - DescribeLogStreams
  - CreateLogGroup

  Note

  您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。您可能會遇到下列錯誤訊息：
  無法建立資源。請確認您擁有建立服務連結角色的許可。否則請等待，然後再試一次。
  如果您看到此訊息，請確認您已啟用以下許可：

```
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
    }
}
```

  如需詳細資訊，請參閱《IAM 使用者指南》中的服務連結角色許可。

建立 Neptune 的服務連結角色

您不需要手動建立一個服務連結角色。當您建立執行個體或叢集時，Neptune 會為您建立服務連結角色。

Important

若要進一步了解，請參閱 IAM 使用者指南中的「我的 IAM 帳戶」中出現的新角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立執行個體或叢集時，Neptune 會再次為您建立服務連結角色。

編輯 Neptune 的服務連結角色

Neptune 不允許您編輯AWSServiceRoleForRDS服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需更多資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

刪除 Neptune 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。不過，您必須先刪除您的所有執行個體和叢集，才能刪除關聯的服務連結角色。

刪除之前先清除服務連結角色

您必須先確認服務連結角色沒有作用中的工作階段，並移除該角色使用的資源，之後才能使用 IAM 將其刪除。

檢查服務連結角色是否於 IAM 主控台有作用中的工作階段

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後選擇 AWSServiceRoleForRDS 角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 分頁中，檢閱服務連結角色的近期活動。

   Note

   如果您不確定 Neptune 是否正在使用此AWSServiceRoleForRDS角色，可以嘗試刪除角色。如果服務正在使用該角色，則刪除會失敗，而您可以檢視正在使用該角色的 區域。如果服務正在使用該角色，您必須先等到工作階段結束，才能刪除該角色。您無法撤銷服務連結角色的工作階段。

如果您想要移除 AWSServiceRoleForRDS 角色，則必須先刪除您的所有執行個體和叢集。

刪除您的所有執行個體

使用下列其中一個程序來刪除您的每一個執行個體。

刪除執行個體 (主控台)

1. 前往 https://console.aws.amazon.com/rds/，開啟 Amazon RDS 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 在 Instances (執行個體) 清單中，選擇您要刪除的執行個體。
4. 選擇 Instance actions (執行個體動作)，然後選擇 Delete (刪除)。
5. 若出現 Create final Snapshot? (是否建立最終快照？) 提示訊息，請選擇 Yes (是) 或 No (否)。
6. 如果您在前一個步驟中選擇 Yes (是)，則對於 Final snapshot name (最終快照名稱)，輸入您的最終快照名稱。
7. 選擇 刪除 。

刪除執行個體 (AWS CLI)

請參閱 AWS CLI 命令參考中的 delete-db-instance。

刪除執行個體 (API)

請參閱 DeleteDBInstance。

刪除您的所有叢集

使用以下其中一個程序來刪除單一叢集，然後為每個叢集重複此程序。

刪除叢集 (主控台)

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在 Clusters (叢集) 清單中，選擇您要刪除的叢集。
3. 選擇 Cluster Actions (叢集動作)，然後選擇 Delete (刪除)。
4. 選擇 刪除 。

刪除叢集 (CLI)

請參閱 AWS CLI 命令參考中的 delete-db-cluster。

刪除叢集 (API)

請參閱 DeleteDBCluster

您可以使用 IAM 主控台、IAM CLI 或 IAM API 刪除AWSServiceRoleForRDS服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的刪除服務連結角色。

使用暫時登入資料的 IAM 身分驗證

亞馬遜海王星使用臨時登入資料支援 IAM 身分

您可以使用假定的角色，並使用 IAM 身分驗證政策進行身分驗證，就像前面幾節的其中一個範例政策一樣。

如果您使用臨時的登入資料，除了指定
AWS_ACCESS_KEY_ID、AWS_SECRET_ACCESS_KEY、SERVICE_REGION，還必須指定
AWS_SESSION_TOKEN。

> **Note**
>
> 臨時登入資料會在指定的時間後過期，包括工作階段字符。
> 當您請求新的登入資料時，必須更新工作階段字符。如需詳細資訊，請參閱使用臨時安全登入資料來要求AWS資源的存取權。

以下各節說明如何允許存取和擷取臨時登入資料。

使用暫時登入資料進行身分驗證

1. 建立具有存取 Neptune 叢集權限的 IAM 角色。如需建立此角色的詳細資訊，請參閱the section called “IAM 政策的類型” (p. 177)。
2. 為角色新增一個允許存取登入資料的信任關係。

   擷取臨時登入資料，包括 AWS_ACCESS_KEY_ID、AWS_SECRET_ACCESS_KEY 和
   AWS_SESSION_TOKEN。

3. 連線至 Neptune 叢集，並使用臨時認證簽署要求。如需連線及簽署請求的詳細資訊，請參閱 the section called “連線與簽署” (p. 160)。

有多種方法可根據環境擷取臨時登入資料。

主題

- 使用 AWS CLI 取得臨時登入資料 (p. 219)
- 為海王星 IAM 身分驗證設定 AWS Lambda (p. 221)
- 為海王星 IAM 身份驗證設置亞馬遜 EC2 (p. 222)

使用 AWS CLI 取得臨時登入資料

若要使用 AWS Command Line Interface (AWS CLI) 取得認證，首先您需要新增信任關係，以授與將執行AWS CLI命令的使用AWS者承擔角色的權限。

將下列信任關係新增至 Neptune IAM 身份驗證角色。如果您沒有海王星 IAM 身份驗證角色，請參閱the section called “IAM 政策的類型” (p. 177)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需有關將信任關係新增至角色的資訊，請參閱《AWS Directory Service管理指南》中的編輯現有角色的信任關係。

如果 Neptune 原則尚未附加至角色，請建立新角色。附加 Neptune IAM 身份驗證政策，然後新增信任政策。如需建立新角色的資訊，請參閱建立新角色。

Note

下節假設您已安裝 AWS CLI。

手動執行 AWS CLI

1. 輸入以下命令，使用 AWS CLI 請求登入資料。將角色 ARN、工作階段名稱、描述檔取代為您自己的值。

```
aws sts assume-role  --role-arn  arn:aws:iam::123456789012:role/NeptuneIAMAuthRole --
role-session-name test --profile testprofile
```

2. 以下為此命令的範例輸出。`Credentials` 區段包含您需要的值。

Note

記錄 `Expiration` 值，因為在這之後您將需要取得新的登入資料。

```
{
```

```
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-
example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2/
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

3. 使用傳回的登入資料設定環境變數。

```
export AWS_ACCESS_KEY_ID=ASIAJEXAMPLEXEG2JICEA
export AWS_SECRET_ACCESS_KEY=9drTJvcXLB89EXAMPLELB8923FB892xMFI
export AWS_SESSION_TOKEN=AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/
qwjzP2iEXAMPLEbw/m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2/
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=

export SERVICE_REGION=us-east-1 or us-east-2 or us-west-1 or us-west-2 or ca-central-1
 or
                      sa-east-1 or eu-north-1 or eu-west-1 or eu-west-2 or eu-west-3 or
 eu-central-1 or me-south-1 or me-south-2 or
                      af-south-1 or ap-east-1 or ap-northeast-1 or ap-northeast-2 or
 ap-southeast-1 or ap-southeast-2 or ap-south-1 or
                      cn-north-1 or cn-northwest-1 or
                      us-gov-east-1 or us-gov-west-1
```

4. 使用以下其中一個方法連接。

   - the section called “Gremlin 主控台” (p. 163)
   - the section called “Gremlin Java” (p. 165)
   - the section called “SPARQL Java (RDF4J 和 Jena)” (p. 167)
   - the section called “Python 範例” (p. 169)

使用指令碼取得登入資料

1. 執行以下命令來安裝 jq 命令。指令碼會使用此命令來剖析 AWS CLI 命令的輸出。

```
sudo yum -y install jq
```

2. 在文字編輯器中建立名為 `credentials.sh` 的檔案，並加入下列文字。將服務區域、角色 ARN、工作階段名稱、描述檔取代為您自己的值。

```
#!/bin/bash

creds_json=$(aws sts assume-role  --role-arn  arn:aws:iam::123456789012:role/
NeptuneIAMAuthRole --role-session-name test --profile testprofile)
```

```
export AWS_ACCESS_KEY_ID=$(echo "$creds_json" | jq .Credentials.AccessKeyId |tr -d '"')
export AWS_SECRET_ACCESS_KEY=$(echo "$creds_json" | jq .Credentials.SecretAccessKey| tr
 -d '"')
export AWS_SESSION_TOKEN=$(echo "$creds_json" | jq .Credentials.SessionToken|tr -d '"')

export SERVICE_REGION=us-east-1 or us-east-2 or us-west-1 or us-west-2 or ca-central-1
 or
                      sa-east-1 or eu-north-1 or eu-west-1 or eu-west-2 or eu-west-3 or
 eu-central-1 or me-south-1 or me-south-2 or
                      af-south-1 or ap-east-1 or ap-northeast-1 or ap-northeast-2 or
 ap-southeast-1 or ap-southeast-2 or ap-south-1 or
                      cn-north-1 or cn-northwest-1 or
                      us-gov-east-1 or us-gov-west-1
```

3. 使用以下其中一個方法連接。

   - the section called “Gremlin 主控台” (p. 163)
   - the section called “Gremlin Java” (p. 165)
   - the section called “SPARQL Java (RDF4J 和 Jena)” (p. 167)
   - the section called “Python 範例” (p. 169)

為海王星 IAM 身分驗證設定 AWS Lambda

AWS Lambda每次執行 Lambda 函數時，都會自動包含認證。

首先，您要新增一個信任關係，以授與權限來擔任 Lambda 服務的角色。

將下列信任關係新增至 Neptune IAM 身份驗證角色。如果您沒有海王星 IAM 身份驗證角色，請參閱the section called “IAM 政策的類型” (p. 177)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需有關將信任關係新增至角色的資訊，請參閱《AWS Directory Service管理指南》中的編輯現有角色的信任關係。

如果 Neptune 原則尚未附加至角色，請建立新角色。附加 Neptune IAM 身份驗證政策，然後新增信任政策。如需有關建立新角色的資訊，請參閱《AWS Directory Service管理指南》中的建立新角色。

若要從 Lambda 存取海王星

1. 請登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/lambda/ 的 AWS Lambda 主控台。
2. 建立適用於 Python 3.6 版的新 Lambda 函數。
3. 將 `AWSLambdaVPCAccessExecutionRole` 角色指派給 Lambda 函數。這是訪問海王星資源所必需的，這些資源僅為 VPC。
4. 將海王星身份驗證 IAM 角色指派給 Lambda 函數。

如需詳細資訊，請參閱AWS Lambda開發人員指南中的 AWSLambda 許可。

5. 將 IAM 身分驗證 Python 範例複製到 Lambda 函數程式碼。

   如需範例及範本程式碼的詳細資訊，請參閱 the section called “Python 範例” (p. 169)。

為海王星 IAM 身份驗證設置亞馬遜 EC2

Amazon EC2 可讓您使用執行個體描述檔來自動提供登入資料。如需詳細資訊，請參閱 IAM 使用者指南中的使用執行個體設定檔。

首先，您要新增一個信任關係，該關係授予權限以擔任 Amazon EC2 服務的角色。

將下列信任關係新增至 Neptune IAM 身份驗證角色。如果您沒有海王星 IAM 身份驗證角色，請參閱the section called “IAM 政策的類型” (p. 177)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需有關將信任關係新增至角色的資訊，請參閱《AWS Directory Service管理指南》中的編輯現有角色的信任關係。

如果 Neptune 原則尚未附加至角色，請建立新角色。附加 Neptune IAM 身份驗證政策，然後新增信任政策。如需有關建立新角色的資訊，請參閱《AWS Directory Service管理指南》中的建立新角色。

使用指令碼取得登入資料

1. 執行以下命令來安裝 jq 命令。指令碼會使用此命令來剖析 curl 命令的輸出。

   ```
   sudo yum -y install jq
   ```

2. 在文字編輯器中建立名為 `credentials.sh` 的檔案，並加入下列文字。將服務區域取代成您自己的值。

   ```
   role_name=$( curl -s http://169.254.169.254/latest/meta-data/iam/security-
   credentials/ )
   creds_json=$(curl -s http://169.254.169.254/latest/meta-data/iam/security-credentials/
   ${role_name})

   export AWS_ACCESS_KEY_ID=$(echo "$creds_json" | jq .AccessKeyId |tr -d '"')
   export AWS_SECRET_ACCESS_KEY=$(echo "$creds_json" | jq .SecretAccessKey| tr -d '"')
   export AWS_SESSION_TOKEN=$(echo "$creds_json" | jq .Token|tr -d '"')

   export SERVICE_REGION=us-east-1 or us-east-2 or us-west-1 or us-west-2 or ca-central-1
    or
                         sa-east-1 or eu-north-1 or eu-west-1 or eu-west-2 or eu-west-3 or
    eu-central-1 or me-south-1 or me-south-2 or
   ```

```
                    af-south-1 or ap-east-1 or ap-northeast-1 or ap-northeast-2 or
ap-southeast-1 or ap-southeast-2 or ap-south-1 or
                    cn-north-1 or cn-northwest-1 or
                    us-gov-east-1 or us-gov-west-1
```

3. 使用 source 命令在 bash shell 中執行指令碼：

```
source credentials.sh
```

更好的是將此腳本中的命令添加到 EC2 執行個體上的 .bashrc 檔案中，以便在您登入時自動呼叫它們，讓 Gremlin 主控台可使用臨時的登入資料。

4. 使用以下其中一個方法連接。

- the section called “Gremlin 主控台” (p. 163)
- the section called “Gremlin Java” (p. 165)
- the section called “SPARQL Java (RDF4J 和 Jena)” (p. 167)
- the section called “Python 範例” (p. 169)

記錄和監控亞馬遜海王星資源

Amazon Neptune 支援各種監控效能和使用情況的方法：

- 叢集狀態 — 檢查 Neptune 叢集圖形資料庫引擎的健全狀況。如需詳細資訊，請參閱the section called “執行個體狀態” (p. 770)。
- 亞馬遜 CloudWatch — 海王星自動將指標發送到警報，CloudWatch並支持CloudWatch警報。如需詳細資訊，請參閱the section called “使用 CloudWatch” (p. 772)。
- 稽核記錄檔 — 使用 Neptune 主控台檢視、下載或監視資料庫記錄檔。如需詳細資訊，請參閱the section called “使用海王星稽核記錄” (p. 781)。
- 將日誌發佈到 Amazon CloudWatch 日誌 — 您可以設定 Neptune 資料庫叢集，將稽核日誌資料發佈到 Amazon CloudWatch 日誌中的日誌群組。使用CloudWatch日誌，您可以對日誌數據進行實時分析，用CloudWatch於創建警報和查看指標，以及使用CloudWatch日誌將日誌記錄存儲在高度耐用的存儲中。如需詳細資訊，請參閱海王星CloudWatch日誌 (p. 782)。
- AWS CloudTrail— 海王星支持使用 API 日誌記錄CloudTrail. 如需詳細資訊，請參閱the section called “使用記錄海王星 API 呼叫 AWS CloudTrail” (p. 789)。
- 標記 — 使用標籤將中繼資料新增至 Neptune 資源，並根據標籤追蹤使用情況。如需詳細資訊，請參閱the section called “標記海王星資源” (p. 799)。

亞馬遜海王星的合規驗證

要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱合規計劃範圍內的 AWS 服務，並選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS 合規計畫。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱 AWS Artifact 中的下載報告。

您使用 AWS 服務 時的合規責任取決於資料的敏感度、您的公司的合規目標，以及適用的法律和法規。AWS 提供以下資源協助您處理合規事宜：

- 安全與合規快速入門指南 – 這些部署指南討論在 AWS 上部署以安全及合規為重心的基準環境的架構考量和步驟。
- Amazon Web Services 的 HIPAA 安全與合規架構：本白皮書說明公司可如何運用 AWS 來建立符合 HIPAA 規定的應用程式。

Note

並非全部的 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源：這組手冊和指南可能適用於您的產業和位置。
- AWS Config 開發人員指南中的使用規則評估資源：AWS Config 服務可評估資源組態對於內部實務、業界準則和法規的合規狀態。
- AWS Security Hub – 此 AWS 服務 可供您全面檢視 AWS 中的安全狀態。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的合規是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub 控制參考。
- AWS Audit Manager – 此 AWS 服務 可協助您持續稽核 AWS 使用情況，以簡化管理風險與法規與業界標準的法規遵循方式。

亞馬遜海王星的彈性

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS區域提供多個分開且隔離的實際可用區域，它們以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

Amazon Neptune 資料庫叢集只能在至少兩個可用區域中具有至少兩個子網路的 Amazon VPC 中建立。Neptune 將您的叢集執行個體分配至至少兩個可用區域，有助於確保資料庫叢集中有可用的執行個體，以避免發生可用區域故障的可用性。Neptune DB 叢集的叢集磁碟區永遠跨越三個可用區域，以提供耐用的儲存裝置，減少資料遺失的可能性。

如需有關 AWS 區域與可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

將現有圖形遷移到 Amazon Neptune

有許多工具和技術可協助您將現有的圖形資料從其他資料存放區遷移到 Amazon Neptune。

簡單遷移需要下列步驟：

- 從 Amazon Simple Storage Service (Amazon S3) 中資料存放區。
- 清理並格式化以進行導入。
- 將其載入到 Neptune 資料庫叢集使用海王星散裝載機 (p. 255).
- 將您的小鬼或 SPARQL 應用程式設定為使用 Neptune 提供的對應端點。

> **Note**
>
> Neptune 叢集必須在應用程式可存取的 VPC 中執行。

有一些方法可以簡化和自動化其中一些步驟，具體取決於數據的存儲位置：

主題

- 從 Neo4j 遷移到 Amazon Neptune (p. 226)
- 將現有的圖形從阿帕奇 TinkerPop Gemlin 服務器遷移到 Amazon Neptune (p. 248)
- 將現有圖形從 RDF 三重商店遷移到 Amazon Neptune (p. 249)
- 使用AWS Database Migration Service (AWS DMS) 從關係或 NoSQL 數據庫遷移到 Amazon Neptune (p. 250)
- 從火焰圖遷移到 Amazon Neptune (p. 251)

從 Neo4j 遷移到 Amazon Neptune

Neo4j 和 Amazon Neptune 都是圖形資料庫，專為支援標籤屬性圖形資料模型的線上交易圖形工作負載而設計。這些相似性使得 Neptune 成為尋求遷移其當前 Neo4j 應用程序的客戶的共同選擇。不過，這些移轉不僅僅是提升和轉移，因為兩個資料庫之間的語言和功能支援、作業特性、伺服器架構和儲存功能都有所差異。

此頁面框架遷移過程，並提出了將 Neo4j 圖形應用程序遷移到 Neptune 之前需要考慮的事情。這些考量通常適用於任何 Neo4j 圖形應用程序，無論是由社區，企業或 Aura 數據庫提供支持。雖然每個解決方案都是獨一無二的，並且可能需要額外的程序，但所有遷移都遵循相同的

以下各節中描述的每個步驟都包含簡化移轉程序的考量和建議。此外，還有開源工具和描述該過程的博客文章 (p. 247)，以及帶有推薦架構選項的功能兼容性部分 (p. 239)。

主題

- 有關從 Neo4j 遷移到 Neptune 的一般信息 (p. 226)
- 準備從 Neo4j 遷移到 Neptune (p. 228)
- 從 Neo4j 遷移到 Neptune 時佈建基礎架構 (p. 231)
- 從 Neo4j 到 Neptune 的數據遷移 (p. 233)
- 從 Neo4j 到 Neptune 的應用程序遷移 (p. 237)
- Neptune 與 Neo4j 的兼容性 (p. 239)
- 重寫密碼查詢以在 Neptune 上的 OpenCpher 中運行 (p. 242)
- 從 Neo4j 遷移到 Neptune 的資源 (p. 247)

有關從 Neo4j 遷移到 Neptune 的一般信息

透過對 OpenCypher 查詢語言的 Neptune 支援 (p. 95)，您可以將使用螺栓通訊協定或 HTTPS 的大多數 Neo4j 工作負載移至 Neptune。然而，OpenCypher 是一個開源規範，其中包含大多數但不是所有的其他數據庫，如 Neo4j 的支持的功能。

儘管在許多方面兼容，Neptune 並不是 Neo4j 的立即替代品。Neptune 是一種完全受管的圖形資料庫服務，具有與 Neo4j 不同的高可用性和高耐久性等企業功能。Neptune 是以執行個體為基礎，具有單一主要寫入器執行個體和最多 15 個僅供讀取複本執行個體，可讓您水平擴展讀取容量。使用 Neptune 無伺服器 (p. 577)，您可以根據查詢磁碟區自動擴展和縮減運算容量。這與 Neptune 儲存無關，它會在您新增資料時自動擴展。

Neptune 支援開放原始碼的開放式密碼標準規格，第 9 版。在AWS，我們相信開放原始碼對每個人都有好處，我們致力於為客戶帶來開放原始碼的價值，同時為開放原始碼社群帶來卓越的AWS營運。

但是，在 Neo4j 上運行的許多應用程序也使用非開源且 Neptune 不支持的專有功能。例如，Neptune 不支援 APOC 程序、某些 Cyphers 特定的子句和函數，以及`Char`、`Date`、或`Duration`資料類型。Neptune 會自動將遺失的資料類型轉換為支援的資料類 (p. 268)型。

除了開放密碼，Neptune 還支持阿帕奇 TinkerPop 格林查詢語言的屬性圖（以及 SPARQL 的 RDF 數據）。怪靈可以在同一個屬性圖上與 OpenCypher 進行互操作，並且在許多情況下，您可以使用小鬼來提供 OpenCypher 不提供的功能。以下是兩種語言的快速比較：

	开放密码	Gremlin
Style (樣式)	宣告	勢在必行
語法	模式比對	基於遍歷
	`Match p=(a)-[:route]->(d)`	`g.V().has('code', 'ANC').`

	开放密码	Gremlin
	`WHERE a.code='ANC'` `RETURN p`	`out('route').path().` `by(elementMap())`
易於使用	受 SQL 啟發，可由非程序員讀取	更陡峭的學習曲線，類似於 Java 等編程語言
彈性	低	高
支援的查詢	以字串為基礎的查詢	用戶端程式庫支援的字串式查詢或內嵌程式碼
用戶端	HTTPS 和螺栓	HTTPS 和網路通訊端

一般來說，沒有必要更改數據模型即可從 Neo4j 遷移到 Neptune，因為 Neo4j 和 Neptune 都支持標記的屬性圖（LPG）數據。不過，Neptune 有一些架構和資料模型差異，您可以利用這些差異來最佳化效能。例如：

- Neptune ID 被視為頭等公民。
- Neptune 使用 AWS Identity and Access Management(IAM) 政策 (p. 157)以靈活且精細的方式保護對圖形資料的存取。
- Neptune 提供了幾種使用 Jupyter 筆記本 (p. 30)來執行查詢和視覺化結果的 (p. 47)方法。Neptune 也可搭配第三方視覺化工具 (p. 476)使用
- > 雖然 Neptune 沒有立即取代 Neo4j 圖形資料科學 (GDS) 程式庫，但是 Neptune 現在透過各種解決方案支援圖形分析。例如，數個範例筆記本示範如何在 Python 環境中利用 Neptune 與 PAWS andas SDK 整合，以對圖形資料執行分析。

如果您有任何疑問，請聯繫AWS支持或與您的客AWS戶團隊聯繫。我們會根據您的意見來排定符合您需求的新功能的優先順序。

準備從 Neo4j 遷移到 Neptune

遷移的方法

將 Neo4j 應用程序遷移到 Neptune 時，我們建議您使用以下兩種策略之一：重新平台化或重構/重構。如需有關移轉策略的詳細資訊，請參閱將應用程式移轉至雲端的 6 個策略，這是 Stephen Orban 所撰寫的部落格文章。

重新平台形成方法，有時稱為 lift-tinker-and-shift，包括以下步驟：

- 識別您的應用程式旨在滿足的使用案例。
- 使用 Neptune 的功能修改現有的圖形資料模型和應用程式架構，以充分滿足這些工作負載需求。
- 決定如何將來源應用程式的資料、查詢和其他部分移轉至目標模型和架構。

這種向後運作方法可讓您將應用程式遷移到您可能設計的 Neptune 解決方案，如果這是一個全新的專案。

相比之下，重構方法涉及：

- 識別現有實作的元件，包括基礎結構、資料、查詢和應用程式功能。
- 在 Neptune 中尋找可用來建置可比較實作的等價物。

這種向前工作的方法旨在將一個實現交換為另一個實現。

在實踐中，您可能會採用這兩種方法的混合。您可以從用例開始，設計目標 Neptune 體系結構，但然後轉到現有的 Neo4j 實現來確定您必須維護的約束和不變量。例如，您可能必須繼續與其他外部系統整合，或繼續向圖形應用程式的消費者提供特定的 API。有了這些資訊，您就可以判斷已存在哪些資料以移至目標模型，以及哪些資料必須在其他地方取得來源。

在其他方面，您可以從分析 Neo4j 實現的特定部分開始，作為有關應用程序打算執行的工作的最佳信息來源。現有應用程式中的這種考古學可以幫助定義一個使用案例，然後您可以使用 Neptune 的功能進行設計。

無論您是使用 Neptune 構建新的應用程序還是從 Neo4j 遷移現有的應用程序，我們建議您從用例向後工作以設計數據模型，一組查詢以及滿足您業務需求的應用程序架構。

Neptune 和 Neo4j 之間的架構差異

當客戶第一次考慮將應用程式從 Neo4j 遷移到 Neptune 時，通常很容易根據執行個體大小執行 like-to-like 比較。但是，Neo4j 和 Neptune 的體系結構有根本的區別。Neo4j 的基於一 all-in-one 種方法，其中數據加載，數據 ETL，應用程序查詢，數據存儲和管理操作都在同一組計算資源（例如 EC2 實例）中進行。

相比之下，Neptune 是 OLTP 集中的圖形資料庫，其中架構會分隔責任以及資源分離的位置，以便它們可以動態且獨立地擴充。

從 Neo4j 遷移到 Neptune 時，請確定應用程序的數據耐久性，可用性和可擴展性要求。Neptune 的叢集架構可簡化需要高耐用性、可用性和可擴充性之應用程式的設計。瞭解 Neptune 的叢集架構後，您就可以設計 Neptune 叢集拓撲來滿足這些需求。

集群架構

許多生產應用程序使用 Neo4j 的因果聚類來提供數據耐久性，高可用性和可擴展性。Neo4j 的聚類體系結構使用核心服務器和僅供讀取副本實例：

- 核心伺服器透過使用 Rraft 通訊協定複製資料，提供資料耐久性和容錯能力。
- 僅供讀取複本使用交易記錄傳送，以非同步方式複寫高讀取輸送量工作負載的資料。

叢集中的每個執行個體 (不論是核心伺服器或僅供讀取複本) 都包含圖形資料的完整副本。

海王星集群架構

Neptune 叢集 (p. 117)由主要寫入器執行個體和最多 15 個僅供讀取複本執行個體組成。叢集中的所有執行個體共用與執行個體分開的相同基礎分散式儲存服務。

- 主要寫入器執行個體會協調資料庫的所有寫入作業，並可垂直擴充，以針對不同的寫入工作負載提供彈性支援。它還支持讀取操作。
- 僅供讀取複本執行個體支援來自基礎儲存磁碟區的讀取作業，並可讓您水平擴展以支援高讀取工作負載。它們也會做為主要執行個體的容錯移轉目標，藉此提供高可用性。

  Note

  對於繁重的寫入工作負載，最好將僅供讀取複本執行個體擴展到與寫入器執行個體相同的大小，以確保讀取器能夠與資料變更保持一致。
- 基礎儲存體磁碟區會隨著資料庫中的資料增加，因此會高達 128 TiB 儲存體的儲存體。

執行個體大小是動態且獨立的。每個執行個體都可以在叢集執行時調整大小，而且可以在叢集執行時新增或移除僅供讀取複本。

Neptune 無伺服器 (p. 577)功能可以隨著需求增加和下降，自動擴展和縮減您的運算容量。這不僅可以降低管理額外負荷，還可以讓您設定資料庫來處理大量尖峰的需求，而不會降低效能或要求您過度佈建。

您可以停用 Neptune 叢集長達七天。

Neptune 還支持 auto-scaling (p. 549)，以根據工作負載自動調整讀取器實例大小。

使用 Neptune 的全域資料庫功能 (p. 73)，您最多可以在 5 個其他區域鏡像叢集。

Neptune 通過設計也是容錯 (p. 807)的：

- 叢集磁碟區為叢集中的所有執行個體提供資料儲存體的叢集磁碟區 (AZ) 中AWS 區域。每個 AZ 包含叢集資料的完整複本。
- 如果主執行個體無法使用，Neptune 會自動容錯移轉到現有的僅供讀取複本，資料遺失為零，通常在 30 秒內完成。如果叢集中沒有現有的僅供讀取複本，Neptune 會再次自動佈建新的主要執行個體，而不會遺失資料。

這一切意味著，從 Neo4j 因果叢集遷移到 Neptune 時，您不必明確地架構叢集拓撲，以達到高資料耐久性和高可用性。如此一來，您就可以透過幾種方式調整叢集的預期讀取和寫入工作負載，以及任何增加的可用性需求：

- 若要擴展讀取作業，請新增僅供讀取複本執行個體 (p. 117)或啟用 Neptune 無伺服器 (p. 577)功
- 若要提高可用性，請將叢集中的主要執行個體和僅供讀取複本分配到數個可用區域 (AZ) (AZ) 上。
- 若要縮短任何容錯移轉時間，請至少佈建一個僅供讀取複本執行個體，以做為主要執行個體的容錯移轉目標。您可以指派每個僅供讀取複本的優先順序，以決定失敗之後提升為主要執行個體的優先順 (p. 546)序。最佳做法是確保容錯移轉目標具有能夠處理應用程式寫入工作負載 (如果提升為主要) 的執行個體類別。

Neptune 和 Neo4j 之間的數據存儲區別

Neptune 使用基於本機四模型的圖形數 (p. 104)據模型。將資料移轉至 Neptune 時，資料模型和儲存層的架構有幾個差異，您應該注意這些差異，以便最佳利用 Neptune 提供的分散式和可擴充共用儲存裝置：

- Neptune 不使用任何明確定義的結構描述或條件約束。它使您可以動態添加節點，邊緣和屬性，而無需提前定義模式。Neptune 不會限制儲存資料的值和類型，除非 Neptune 限制 (p. 847)中所述。作為

Neptune 儲存體架構的一部分，資料也會以處理許多最常見存取模式的方式自動編製索引 (p. 104)。這種存儲架構消除了創建和管理數據庫模式和索引優化的操作開銷。

- Neptune 提供獨特的分散式和共用儲存體架構，可隨著資料庫的儲存需求成長自動擴充 10 GB 區塊，最高可達 128 TB (TiB)。此儲存層可靠、耐用且容錯，在 3 個可用區域中，資料複製 6 次，每次複製兩次。依預設，它為所有 Neptune 叢集提供高可用性和容錯的資料儲存層。Neptune 的儲存架構可降低成本，並免除佈建或過度佈建儲存裝置以處理 future 資料成長的需求。

在將資料遷移到 Neptune 之前，最好先熟悉 Neptune 的屬性圖資料模型 (p. 106)和交易語意 (p. 109)。

Neptune 和 Neo4j 之間的操作差異

Neptune 是一個完全託管的服務，可以自動執行使用內部部署或自我管理的數據庫（例如 Neo4j 企業版或社區版）時必須執行的許多正常操作任務：

- 自動備份 (p. 807) — Neptune 會自動備份您的叢集磁碟區，並在您指定的保留期間內保留備份 (從 1 到 35 天)。這些備份具備連續性和增量性，因此，您可以快速還原到保留期內的任何時間點。寫入備份資料時不會影響資料庫服務的效能或中斷服務。
- 手動快照 (p. 806) — Neptune 可以建立資料庫叢集的儲存體磁碟區快照，以備份整個資料庫叢集。然後可以使用這種快照集來還原資料庫、製作複本，以及跨帳戶共用。
- 複製 (p. 564) — Neptune 支援複製功能，可讓您快速建立具成本效益的資料庫複製。翻製會使用 copy-on-write 通訊協定，在建立之後只需要最少的額外空間。資料庫複製是試用新 Neptune 功能或升級的有效方法，而不會中斷原始叢集。
- 監控 (p. 770) — Neptune 提供多種方法來監控叢集的效能和使用狀況，包括：
  - 執行個體
  - 與亞馬遜 CloudWatch 和集成AWS CloudTrail
  - 稽核日誌功能
  - 事件通知
  - 標記
- 安全性 (p. 153) — Neptune 預設會提供安全的環境。叢集位於私有 VPC 內，可提供與其他資源的網路隔離。所有流量均通過 SSL 加密，所有數據均使用AWS KMS.

  此外，Neptune 還與AWS Identity and Access Management (IAM) 整合以提供身份驗證 (p. 157)。透過指定 IAM 條件金鑰 (p. 187)，您可以使用 IAM 政策對資料動作 (p. 203)提供更精細的存取控制。

Neptune 和 Neo4j 之間的工具和集成差異

Neptune 與 Neo4j 的集成和工具具有不同的架構，這可能會影響應用程序的體系結構。Neptune 會使用叢集的計算資源來處理查詢，但會運用其他 best-in-class AWS服務來執行全文檢索搜尋 (使用 OpenSearch)、ETL (使用 Glue) 等功能。如需這些整合的完整清單，請參閱將 Neptune (p. 848)。

從 Neo4j 遷移到 Neptune 時佈建基礎架構

Amazon Neptune 叢集的建置是以三個維度擴展：儲存、寫入容量和讀取容量。以下各節討論移轉時要考慮的特定選項。

佈建儲存體

任何 Neptune 叢集的儲存都會自動佈建，您不會產生任何管理額外負荷。隨著叢集的儲存需求增加，它會以 10 GB 區塊動態調整大小。因此，不需要估計和佈建或過度佈建儲存來處理 future 的資料成長。

佈建寫入容量

Neptune 提供單一寫入器執行個體，可垂直調整為 Neptune 定價頁面上提供的任何執行個體大小。讀取和寫入資料至寫入寫入器執行個體時，所有交易都符合 ACID 標準，且資料隔離如中所定義海王星中的事務隔離級別 (p. 109)。

為寫入器執行個體選擇最佳大小需要執行負載測試，以判斷工作負載的最佳執行個體大小。您可以修改資料庫執行個體類別，隨時調整 Neptune 中的任何執行個體 (p. 571)大小。您可以根據並行性和平均查詢延遲來估計起始執行個體大小，如下所述佈建叢集時估計最佳執行個體大小 (p. 231)。

佈建讀取容量

Neptune 專為水平擴展僅供讀取複本執行個體而建置，方法是在叢集內 (或在 Neptune 全域資料庫 (p. 73)中增加更多 15 個執行個體)，以及垂直擴充 N eptune 定價頁面上可用的任何執行個體大小。所有 Neptune 僅供讀取複本執行個體都使用相同的基礎儲存磁碟區，以最小的延遲啟用透明複寫資料

除了啟用 Neptune 叢集內的讀取要求水平擴展之外，僅供讀取複本也可做為寫入器執行個體的容錯移轉目標，以啟用高可用性。如Amazon Neptune 基本操作準則 (p. 823)需如何判斷叢集中僅供讀取複本的適當數目和放置位置的建議，請參閱。

對於無法預測連線和工作負載的應用程式，Neptune 也支援自動調整規模功能 (p. 549)，可根據您指定的準則自動調整 Neptune 複本的數目。

若要判斷僅供讀取複本執行個體的最佳大小和數目，需要執行負載測試以判斷其必須支援的讀取工作負載特性。您可以修改資料庫執行個體類別，隨時調整 Neptune 中的任何執行個體 (p. 571)大小。您可以根據並行性和平均查詢延遲來估計起始執行個體大小，如下一節 (p. 231)所述。

使用 Neptune 無伺服器視需要自動擴充讀取器和寫入器執行個體

雖然能夠估計預期工作負載所需的運算容量通常很有幫助，但您可以設定 Neptune Serverless (p. 577) 功能以自動擴展和縮減讀取和寫入容量。這可協助您滿足尖峰需求，同時也會在需求減少時自動縮減。

佈建叢集時估計最佳執行個體大小

估計最佳執行個體大小需要知道 Neptune 中的平均查詢延遲、工作負載執行時間，以及正在處理的並行查詢數目。執行個體大小的粗略估計可以計算為平均查詢延遲乘以並行查詢數目。這會提供處理工作負載所需的平均並行執行緒數。

Neptune 執行個體中的每個 vCPU 都可以支援兩個並行查詢執行緒，因此將執行緒除以 2，可提供所需的 vCPUs 數目，然後這些 vCPU 數目可以與 Neptune 定價頁面上的適當執行個體大小相關聯。例如：

```
Average Query Latency:          30ms (0.03s)
Number of concurrent queries:   1000/second

Number of threads needed:       0.03 x 1000 = 30 threads
Number of vCPUs needed:         30 / 2 = 15 vCPUs
```

`r5.4xlarge`將其與執行個體中的 vCPUs 數量相關聯，我們發現我們得到了一個粗略的估計，即建議使用此工作負載的執行個體。這個估計是粗略的，只是為了提供執行個體大小選擇的初始指引。任何應用程式都應該進行適當大小的練習，以確定適合工作負載的執行個體數目和類型。

記憶體需求也應考慮在內，以及處理需求。當查詢所存取的資料在主記憶體緩衝集區快取中可用時，Neptune 是最有效能的。佈建足夠的記憶體也可大幅降低 I/O 成本。

您可以在頁面上找到有關 Neptune 叢集中調整執行個體大小的其他詳細資訊和指調整 Neptune 資料庫叢集中的資料庫執行 (p. 118)引。

從 Neo4j 到 Neptune 的數據遷移

從 Neo4j 到 Amazon Neptune 執行遷移時，遷移數據是該過程中的主要步驟。有多種移轉資料的方法。正確的方法取決於應用程式的需求、資料大小和所需的移轉類型。但是，許多這些遷移都需要評估相同的考量，其中幾個在下面突出顯示。

Note

請參閱資料庫部落格中的全自動化公用程式將 Neo4j 圖形資料AWS庫遷移至 Neptune，以取得如何執行離線資料遷移範例的完整 step-by-step 逐步解說。

評估從 Neo4j 到 Neptune 的數據遷移

評估任何資料移轉的第一個步驟是決定如何移轉資料。這些選項取決於要移轉之應用程式的架構、資料大小，以及移轉期間的可用性需求。通常，遷移往往分為兩類之一：在線或離線。

離線移轉通常是最簡單的方法，因為應用程式在移轉期間不接受讀取或寫入流量。應用程式停止接受流量之後，可以匯出、最佳化、匯入資料，並在應用程式重新啟用之前測試應用程式。

線上移轉較為複雜，因為在移轉資料時，應用程式仍然需要接受讀取和寫入流量。每個線上移轉的確切需求可能會有所不同，但一般架構通常會與下列類似：

- 持續更改數據庫的飼料需要通過配置 Neo4j 的流作為源到卡夫卡集群啟用 Neo4j 的。
- 完成此操作後，可以按照中遷移到 Neptune 時從 Neo4j 導出數據 (p. 235)的說明進行導出，以及稍後與卡夫卡主題相關聯的時間進行操作。
- 然後，匯出的資料會依照中的指示匯入 Neptune遷移到 Neptune 時從 Neo4j 導入數據 (p. 236)。
- 然後，可以使用與從 Amazon Kinesis 資料串流寫入 Amazon Neptune 中所述的架構類似的架構，將 Kafka 串流中已變更的資料複製到海王星叢集。請注意，變更複寫可以 parallel 執行，以驗證新的應用程式架構和效能。
- 驗證數據遷移後，則應用程序流量可以被重定向到 Neptune 集群和 Neo4j 實例可以退役。

從 Neo4j 遷移到 Neptune 的數據模型優化

Neptune 和 Neo4j 的支持標記的屬性圖（LPG）。不過，Neptune 有一些架構和資料模型差異，您可以利用這些差異來最佳化效能：

最佳化節點和邊緣 ID

Neo4j 的自動生成數字長的 ID。使用 Cypher，您可以按 ID 引用節點，但通常不鼓勵通過索引屬性查找節點。

Neptune 允許您為頂點和邊提供自己的基於字符串的 ID (p. 86)。如果您不提供自己的 ID，Neptune 會自動為新的邊緣和頂點生成 UUID 的字符串表示。

如果您通過從 Neo4j 的導出遷移數據到 Neptune，然後批量導入到 Neptune，您可以保留 Neo4j 的 ID。由 Neo4j 生成的數值可以作為用戶提供的 ID 導入到 Neptune，在那裡他們被表示為字符串，而不是數值時。

但是，在某些情況下，您可能想要將頂點性質晉級為頂點 ID。就像使用索引屬性查找節點一樣是在 Neo4j 中查找節點的最快方法一樣，通過 ID 查找頂點是在 Neptune 中找到頂點的最快方法。因此，如果您可以識別包含唯一值的適當頂點屬性，則應考慮在批量載入 CSV 檔案中以指定的屬性值取代頂點 ~id。如果這樣做，您還必須在 CSV 文件中重寫任何相應的~從和~到邊緣值。

將數據從 Neo4j 遷移到 Neptune 時的架構約束

在 Neptune 中，唯一可用的結構描述條件約束是節點或邊緣識別碼的唯一性。我們鼓勵需要利用唯一性限制的應用程式來查看此方法，透過指定節點或邊緣 ID 來達成唯一性限制。如果應用程式使用多個資

料行做為唯一性條件約束，則 ID 可能會設定為這些值的組合。例如，`id=123, code='SEA'`可以表示為`ID='123_SEA'`）來實現複雜的唯一性約束。

將數據從 Neo4j 遷移到 Neptune 時的邊緣方向優化

當節點、邊緣或屬性新增至 Neptune 時，它們會以三種不同的方式自動建立索引 (p. 104)，並具有選擇性的第四個索引 (p. 128)。由於 Neptune 如何構建和使用索引 (p. 104)，遵循傳出邊緣的查詢比使用傳入邊緣的查詢更有效率。就 Neptune 的圖形資料儲存模型 (p. 104)而言，這些是使用 SPOG 索引的基於主題的搜尋。

如果在將資料模型和查詢移轉至 Neptune 時，您發現最重要的查詢依賴於遍歷高度散佈的傳入邊緣，您可能需要考慮更改模型，以便這些遍歷遵循傳出邊緣，尤其是當您無法指定要遍歷的邊緣時。若要這麼做，請反轉相關邊緣的方向，並更新邊緣標籤以反映此方向變更的語意。例如，您可以進行這些組別：

```
person_A – parent_of – person_B
   to:
person_B – child_of – person_A
```

若要在大量載入邊緣 CSV 檔案中進行 (p. 261)此變更，只要交換`~from`和`~to`欄標題，然後更新`~label`欄的值即可。

作為反轉邊緣方向的替代方法，您可以啟用第四個 Neptune 索引，即 OSGP 索引 (p. 105)，這使得遍歷傳入邊緣或基於對象的搜索更有效。但是，啟用此第四個索引將降低插入率，並且需要更多的存儲空間。

將數據從 Neo4j 遷移到 Neptune 時的過濾優化

Neptune 經過最佳化，以便在將屬性篩選為最可用的選擇性屬性時發揮最佳效果。當使用多個過濾器時，會為每個過濾器找到匹配的項目集合，然後計算所有匹配集的重疊。如果可能，將多個屬性合併為單一屬性，可將索引查閱次數降至最低，並減少查詢的延遲。

例如，此查詢使用兩個索引查詢和一個聯結：

```
MATCH (n) WHERE n.first_name='John' AND n.last_name='Doe' RETURN n
```

這個查詢會使用單一索引查詢擷取相同的資訊：

```
MATCH (n) WHERE n.name='John Doe' RETURN n
```

Neptune 支持不同的數據類型 (p. 268)比 Neo4j 的。

Neo4j 的數據類型映射到 Neptune 支持的數據類型

- 邏輯：`Boolean`

  將這個在 Neptune 映射到`Bool`或`Boolean`。
- 數字：`Number`

  將其映射到 Neptune 中最窄的以下可以支持有問題的數字屬性的所有值的 Neptune OpenCypher 類型：

  ```
  Byte
  Short
  Integer
  Long
  Float
  Double
  ```
- 文字：`String`

  將這在 Neptune 映射到`String`。

- 時間點：

```
Date
Time
LocalTime
DateTime
LocalDateTime
```

使用 Neptune 支援的下列其中一種 ISO-8601 格式，將 Neptune 中的這些對應`Date`為 UTC：

```
yyyy-MM-dd
yyyy-MM-ddTHH:mm
yyyy-MM-ddTHH:mm:ss
yyyy-MM-ddTHH:mm:ssZ
```

- 持續時間：`Duration`

  如有必要，請在 Neptune 中將其映射到日期算術的數值。
- 空間：`Point`

  將其映射到 Neptune 中的組件數值，每個數值然後成為一個單獨的屬性，或表示為要由客戶端應用程序解釋的字符串值。請注意，Neptune 的全文檢索搜尋 (p. 604)整合使用 OpenSearch 可讓您索引地理位置屬性。

將多值屬性從 Neo4j 遷移到 Neptune

Neo4j 的允許簡單類型的同質列表被存儲為兩個節點和邊的屬性。這些清單可以包含重複的值。

然而，Neptune 只允許為頂點屬性設定或單一基 (p. 86)數，以及屬性圖資料中邊緣屬性的單一基數。其結果是，有包含重複值到 Neptune 頂點屬性，或 Neo4j 的關係列表屬性到 Neptune 邊緣屬性 Neo4j 的節點列表屬性沒有直接的遷移。

將具有重複值的 Neo4j 多值節點屬性遷移到 Neptune 的一些可能策略如下：

- 丟棄重複的值，並將多值 Neo4j 的節點屬性轉換為設置基數 Neptune 頂點屬性。請注意，Neptune 集可能不會反映原始 Neo4j 多值屬性中項目的順序。
- 將多值 Neo4j 節點屬性轉換為 Neptune 頂點字符串屬性 JSON 格式列表的字符串表示。
- 將每個多值內容值擷取至具有 value 屬性的個別頂點，然後使用標有屬性名稱的邊將這些頂點連接到父頂點。

同樣，遷移 Neo4j 的多值關係屬性到 Neptune 可能的策略如下：

- 將多值 Neo4j 的關係屬性轉換為 JSON 格式的列表的字符串表示，並將其存儲為 Neptune 邊緣字符串屬性。
- 重構 Neo4j 的關係連接到中間頂點傳入和傳出 Neptune 邊緣。使用標有屬性名稱的邊，將每個多值關係屬性值擷取到具有 value 屬性的單獨頂點，並將這些頂點擷取到此中間頂點。

請注意，JSON 格式清單的字串表示對 OpenCypher 查詢語言來說是不透明的，雖然 OpenCypher 包含允許在字串值內進行簡單搜尋的`CONTAINS`述詞。

遷移到 Neptune 時從 Neo4j 導出數據

當從 Neo4j 的數據導出，使用 APOC 程序導出或者到 CSV 或 GraphML。雖然可以導出到其他格式，但有一些開源工具可以將從 Neo4j 導出的 CSV 數據轉換為 Neptune 批量加載格式，並且還可以將從 Neo4j 導出的 GraphML 數據轉換為 Neptune 批量加載格式的開源工具。

您也可以使用各種 APOC 程序將資料直接匯出到 Amazon S3。匯出至 Amazon S3 儲存貯體預設為停用狀態，但可以使用 Neo4j APOC 文件中匯出至 Amazon S3 中反白顯示的程序來啟用。

遷移到 Neptune 時從 Neo4j 導入數據

您可以使用 Nep tune 大量載 (p. 255)入器，或使用支援的查詢語言 (例如 OpenCypher (p. 382)) 中的應用程式邏輯，將資料匯入 Neptune。

Neptune 大量載入器是匯入大量資料的偏好方法，因為如果您遵循最佳實務，它可提供最佳 (p. 273)化的匯入效能。批量加載器支持兩種不同的 CSV 格式 (p. 261)，可以使用部分中上面提到的開源實用程序將從 Neo4j 導出的數據轉換為該格匯出資料 (p. 235)式。

您還可以使用 OpenCypher 導入具有自定義邏輯的數據，以進行解析，轉換和導入。您可以透過 HTTPS 端點 (p. 386) (建議使用) 或使用螺栓驅動程式 (p. 389)來提交 OpenCypher 查詢。

從 Neo4j 到 Neptune 的應用程序遷移

您已經從 Neo4j 的數據遷移到 Neptune 後，下一步是遷移應用程序本身。與資料一樣，根據您使用的工具、需求、架構差異等，有多種移轉應用程式的方法。下面概述了您通常需要在此過程中考慮的事項。

從 Neo4j 移動到 Neptune 時遷移連接

如果您目前未使用 Bolt 驅動程式，或想要使用替代驅動程式，您可以連線至 HTTPS 端點，該端點 (p. 386)可提供對傳回資料的完整存取權。

如果您確實有使用 Bolt 協議 (p. 389)的應用程序，則可以將這些連接遷移到 Neptune，並讓您的應用程序使用與在 Neo4j 中相同的驅動程序進行連接。若要連線到 Neptune，您可能需要對應用程式進行下列一或多項變更：

- 必須更新 URL 和連接埠，才能使用叢集端點和叢集連接埠 (預設值為 8182)。
- Neptune 要求所有連接都使用 SSL，因此您需要為其加密的每個連接指定。
- Neptune 透過指派 IAM 政策和角色 (p. 157)來管理驗證。IAM 政策和角色在應用程式內提供極為靈活的使用者管理層級，因此在設定叢集之前，務必先閱讀並瞭解 IAM 概觀 (p. 157)中的資訊。
- 螺栓連接在 Neptune 中的行為與 Neo4j 中的行為不同，如中所述海王星中的螺栓連接行為 (p. 402)。
- 您可以在中尋詳資訊和建議使用開放密碼和螺栓的 Neptune 最佳實踐 (p. 835)。

在中有一些常用語言 (例如 Java、Python、.NET 和 NodeJS) 的程式碼範例，以及如使用 IAM 身分驗證等連線案例的程式碼範例使用博爾特協議向海王星進行開放密碼查詢 (p. 389)。

從 Neo4j 移動到 Neptune 時將查詢路由到集群實例

Neo4j 的客戶端應用程序使用路由驅動程序並指定訪問模式，以將讀取和寫入請求路由到因果集群中的適當服務器。

將用戶端應用程式移轉至 Neptune 時，請使用 Neptune 端點 (p. 121)將查詢有效率地路由到叢集中的適當執行個體：

- 與 Neptune 的所有連接都應該使用`bolt://`而不是 URL`neo4j://` 中的`bolt+routing://`或。
- 叢集中目前的主要執行個體會連線到目前的主要執行個體。使用叢集端點將寫入要求路由到主要端點。
- 讀取器端點會將連線分散 (p. 825)到叢集中的僅供讀取複本執行個體。如果您的單一執行個體叢集會會連線到主要執行個體而支援寫入操作的主要執行個體。如果叢集確實包含一或多個僅供讀取複本執行個體，則傳送寫入要求至讀取器端點會產生例外狀況。
- 叢集中的每個執行個體也可以擁有自己的執行個體端點。如果您的用戶端應用程式需要傳送要求至叢集中的特定執行個體，請使用執行個體端點。

如需詳細資訊，請參閱海王星端點考量 (p. 122)。

Neptune 的資料一致性

使用 Neo4j 因果叢集時，僅供讀取複本最終與核心伺服器一致，但用戶端應用程式可以使用因果鏈結來確保因果一致性。因果鏈結需要在交易之間傳遞書籤，這允許用戶端應用程式寫入核心伺服器，然後從僅供讀取複本讀取自己的寫入。

在 Neptune 中，僅供讀取複本執行個體最終與寫入器一致，複本延遲通常少於 100 毫秒。但是，在複製變更之前，複本執行個體上看不到對現有邊和頂點的更新以及新增的邊和頂點。因此，如果您的應用程式需要透過讀取每個寫入內容在 Neptune 上立即一致性，請使用叢集端點進 read-after-write 行作業。這是唯一使用叢集端點進行讀取作業的時間。在所有其他情況下，請使用讀取器端點進行讀取。

將查詢從 Neo4j 遷移到 Neptune

儘管海王星對 OpenCypher 的支持顯著減少了從 Neo4j 遷移查詢所需的工作量，但遷移時仍需要評估一些差異：

- 如上所資料模型最佳化 (p. 233)述，您可能需要對資料模型進行修改，以便為 Neptune 建立最佳化的圖形資料模型，這反過來又需要變更您的查詢和測試。
- Neo4j 的提供了各種不包括在由 Neptune 實施的 OpenCypher 規範特定的語言擴展。根據使用案例和功能，OpenCypher 語言中可能存在解決方法，或使用 Gremlin 語言，或通過其他機制（如中所述）重寫密碼查詢以在 Neptune 上的 OpenCpher 中運行 (p. 242)。
- 應用程式通常會使用其他中介軟體元件與資料庫互動，而不是 Bolt 驅動程式本身。請檢Neptune 與 Neo4j 的兼容性 (p. 239)查您使用的工具或中介軟體是否受支援。
- 在容錯移轉的情況下，Bolt 驅動程式可能會繼續連線到先前的寫入器或讀取器執行個體，因為提供給連線的叢集端點已解析為 IP 位址。在您的應用程序中正確處理錯誤應該處理這個問題，如中所述容錯移轉後建立新連線 (p. 836)。
- 由於無法解決的衝突或鎖定等待逾時而取消交易時，Neptune 會以`ConcurrentModificationException`. 如需詳細資訊，請參閱引擎錯誤代碼 (p. 849)。根據最佳實務，用戶端應一律截獲和處理這些例外狀況。

  當多個執行緒或多個應用程式同時寫入系統時，偶爾會`ConcurrentModificationException`發生 A。由於事務隔離級別 (p. 110)，這些衝突有時可能是不可避免的。
- Neptune 支持在相同的數據上運行小鬼和 OpenCypher 查詢。這意味著在某些情況下，您可能需要考慮使用具有更強大的查詢功能的 Gemlin 來執行查詢的某些功能。

如上所佈建設施 (p. 231)述，每個應用程式都應該經過適當調整大小的練習，以確保執行個體數量、執行個體大小和叢集拓撲都針對應用程式的特定工作負載進行最佳化。

這裡討論的遷移應用程序的考慮因素是最常見的，但這不是一個詳盡的列表。每個應用都是唯一的。如果您還有其他問題，請聯繫AWS支持或與您的客戶團隊聯繫。

遷移特定於 Neo4j 的功能和工具

Neo4j 具有各種自定義功能和附加組件，具有您的應用程序可能依賴的功能。在評估遷移此功能的需求時，通常有助於調查內部是否有更好的方法AWS來實現相同的目標。考慮到 Neo4j 和 Neptune 之間的架構差異 (p. 228)，您通常可以找到利用其他AWS服務或集成 (p. 848)的有效替代方案。

如Neptune 與 Neo4j 的兼容性 (p. 239)需 Neo4J 特定功能和建議因應措施的清單，請參閱。

Neptune 與 Neo4j 的兼容性

Neo4j 有一種 all-in-one 架構方法，其中數據加載，數據 ETL，應用程序查詢，數據存儲和管理操作都發生在同一組計算資源中，例如 EC2 實例。Amazon Neptune 是以 OLTP 為重點的開放規格圖形資料庫，其架構可在其中分離操作並分離資源，以便動態擴展。

Neo4j 中有各種功能和工具，包括第三方工具，這些工具不屬於 OpenCypher 規範的一部分，與 OpenCypher 不兼容，或者與海王星的 OpenCypher 實現不兼容。下方清單中一些最常見的方式。

Neptune 中不存在的 NEO4J 特定功能

- **`LOAD CSV`**-與 Neo4j 相比，Neptune 具有不同的架構方法來加載數據。為了實現更好的擴展和成本最佳化，Neptune 實作了與資源相關的考量分離，並建議您使用其中一項AWS服務整合 (p. 848)，例如AWS Glue執行必要的 ETL 程序，以 Neptune 大量載入器 (p. 255)所支援的格式 (p. 261)準備資料。

  另一種選擇是使用在運AWS算資源上執行的應用程式程式碼，例如 Amazon EC2 執行個體、Lambda 函數、Amazon 彈性容器服務、任務等執行相同的AWS Batch作業。該代碼可以使用海王星的 HTTPS 端點 (p. 386)或博爾特端點 (p. 389)。
- 精細的存取控制 — Neptune 支援使用 IAM 條件金鑰對資料存取動作進 (p. 203)行精細的存取控制。可以在應用程序層實現其他細粒度訪問控制。
- Neo4j 的結構 — Neptune 確實支持跨數據庫查詢聯合使用 SPARQL `SERVICE` (p. 473)關鍵字的 RDF 工作負載。由於目前沒有針對屬性圖工作負載的查詢聯合的開放標準或規格，因此需要在應用程式層實作該功能。
- 角色型存取控制 (RBAC) — Neptune 透過指派 IAM 政策和角色 (p. 157)來管理驗證。IAM 政策和角色在應用程式內提供極為靈活的使用者管理層級，因此在設定叢集之前，值得先閱讀並瞭解 IAM 概觀 (p. 157)中的資訊。

- 書籤 — Neptune 叢集由單一寫入器執行個體和最多 15 個僅供讀取複本執行個體組成。寫入寫入器執行個體的資料符合 ACID 標準，並在後續讀取時提供強大的一致性保證。僅供讀取複本使用與寫入器執行個體相同的儲存磁碟區，並且最終保持一致，通常在寫入資料後不到 100 毫秒。如果您的使用案例立即需要保證新寫入的讀取一致性，則應將這些讀取導向至叢集端點，而不是讀取器端點。
- APOC 程序 — 由於 APOC 程序不包含在 OpenCypher 規範中，因此 Neptune 不提供對外部程序的直接支援。相反，Neptune 仰賴與其他AWS服務的整合 (p. 848)，以可擴充、安全且可靠的方式實現類似的終端使用者功能。有時候 APOC 程序可以在 OpenCypher 或小鬼中重寫，有些與 Neptune 應用程序無關。

  一般而言，APOC 程序分為以下類別：
  - 匯入 — Neptune 支援使用各種格式匯入資料，使用查詢語言、Neptune 大量載 (p. 255)入器或作為目標AWS Database Migration Service (p. 292)。對數據 ETL 操作可以使用AWS Glue和`neptune-python-utils`開源包來執行。
  - 匯出 — Neptune 支援使用公用`neptune-export` (p. 489)程式匯出資料，該公用程式支援多種常見的匯出格式和方法。
  - 資料庫整合 — Neptune 支援使用 ETL 工具 (例如) 與其他資料庫整合AWS Glue或移轉工具 (例如 AWS Database Migration Service (p. 292).
  - 圖形更新 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
  - 資料結構 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
  - 時間 (日期時間) — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
  - 數學 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
  - 進階圖形查詢 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。

  - 比較圖形 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
  - 密碼執行 — Neptune 透過對 OpenCypher 和 Gremlin 查詢語言的支援，支援更新屬性圖資料的豐富功能集。密碼重寫 (p. 242)如需常用程序重寫的範例，請參閱。
- 自訂程序 — Neptune 不支援使用者建立的自訂程序。此功能必須在應用程序層實現。
- 地理空間 — 儘管 Neptune 不提供對地理空間功能的原生支持，但可以通過與其他AWS服務集成來實現類似的功能，如本博客文章所示：將 Amazon Neptune 和亞馬遜 OpenSearch 服務結合在一起，以進行 Ross Gabay 和 Abhilash Vinod 的地理空間查詢（2022 年 2 月 1 日）。
- 圖形資料科學 — Neptune 現在透過各種解決方案支援圖形分析。Neptune 提供了與 P AWSandas SDK 和幾個示例筆記本的集成，這些筆記本示例顯示如何在 Python 環境中利用此集成來對圖形數據進行分析。

- 結構描述條件約束 — 在 Neptune 中，唯一可用的結構描述條件約束是節點或邊緣 ID 的唯一性。沒有功能可以指定任何其他結構描述條件約束，或是圖形中元素上的任何其他唯一性或值限制條件。Neptune 中的 ID 值是字符串，可以使用 Grimlin 進行設置，如下所示：

```
g.addV('person').property(id, '1') )
```

  我們鼓勵需要利用 ID 作為唯一性限制的應用程式，嘗試使用此方法來達成唯一性限制。如果應用程式使用多個資料行做為唯一性條件約束，則 ID 可能會設定為這些值的組合。例如，`id=123, code='SEA'`可以表示`ID='123_SEA'`為實現複雜的唯一性約束。
- 多租戶 — Neptune 僅支援每個叢集的單一圖形。若要使用 Neptune 建置多租用戶系統，請使用多個叢集，或在單一圖形中以邏輯方式分割租用戶，並使用應用程式端邏輯強制執行分隔。例如，添加一個屬性`tenantId`並將其包含在每個查詢中，如下所示：

```
MATCH p=(n {tenantId:1})-[]->({tenantId:1}) RETURN p LIMIT 5)
```

  Neptune Serverless (p. 577) 可讓您使用多個資料庫叢集來實作多租戶相對容易，每個叢集都可以根據需要獨立和自動調整規模。

對 Neo4j 工具的 Neptune 支持

Neptune 提供了以下替代 Neo4j 的工具：

- Neo4j 瀏覽器 — Neptune 提供開放原始碼圖形筆記本 (p. 30)，可提供以開發人員為中心的 IDE 來執行查詢和視覺化結果。
- Neo4j 的 Bloom — Neptune 支持使用第三方可視化解決方案 (p. 476)，如圖形資源管理器，湯姆索亞，劍橋情報，石墨學，隱喻和 G.V（）豐富的圖形可視化。
- GraphQL — Neptune 目前通過自定義AWS AppSync集成支持 GraphQL。請參閱使用 Amazon Neptune 和AWS Amplify 部落格文章建立圖形應用程式，以及使用和 Amazon Neptune 建立無伺服器卡路里追蹤器應用AWS AppSync程式範例專案。
- NeoSemantics— Neptune 原生支援 RDF 資料模型，因此建議希望執行 RDF 工作負載的客戶使用 Neptune 的 RDF 模型支援。
- Arrows.app — 使用匯出指令匯出模型時建立的密碼與 Neptune 相容。
- 鏈接奧格瑪-與林肯奧格瑪的樣本集成可以在這裡找到。
- 春季數據 Neo4j 的-這是目前不與 Neptune 兼容。
- Neo4j 的星火連接器-Neo4j 的火花連接器可以在一個火花 Job 中使用連接到 Neptune 使用 OpenCypher。以下是一些示例代碼和應用程序配置：

  範本程式碼：

```
SparkSession spark = SparkSession
            .builder()
```

```
            .config("encryption.enabled", "true")
            .appName("Simple Application").config("spark.master", "local").getOrCreate();

Dataset<Row> df = spark.read().format("org.neo4j.spark.DataSource")
            .option("url", "bolt://(your cluster endpoint):8182")
            .option("encryption.enabled", "true")
            .option("query", "MATCH (n:airport) RETURN n")
            .load();

System.out.println("TOTAL RECORD COUNT: " + df.count());
spark.stop();
```

應用程序配置：

```
<dependency>
    <groupId>org.neo4j</groupId>
    <artifactId>neo4j-connector-apache-spark_2.12-4.1.0</artifactId>
    <version>4.0.1_for_spark_3</version>
</dependency>
```

這裡未列出的 Neo4j 功能和工具

如果您使用的工具或功能未列於此處，我們不確定其與 Neptune 或其他服務的相容性AWS。如果您還有其他問題，請聯繫AWS支持或與您的客戶團隊聯繫。

重寫密碼查詢以在 Neptune 上的 OpenCpher 中運行

OpenCypher 語言是一種聲明式查詢語言，用於最初由 Neo4j 開發，然後在 2015 年開源，並在 Apache 2 開源許可證下為 OpenCypher 項目做出了貢獻。我們相信開放原始碼對每個人都有好處AWS，我們致力於為客戶帶來開放原始碼的價值，並為開放原始碼社群帶來卓越的AWS營運。

OpenCypher 語法記錄在密碼查詢語言參考版本 9 中。

由於 OpenCypher 包含 Cypher 查詢語言的語法和功能的子集，因此某些遷移案例需要以 OpenCypherr 相容的表單重寫查詢，或檢查替代方法以達到所需的功能。

本節包含處理常見差異的建議，但絕不是詳盡無遺的。您應該使用這些重寫徹底測試任何應用程式，以確保結果符合您的預期。

重寫NoneAll，和Any謂詞函數

這些功能不是開放密碼器規範的一部分。可比較的結果可以在 OpenCypher 中使用列表理解來實現。

例如，查找從節點Start到節點的所有路徑End，但不允許旅程通過具有 class 屬性的節點D：

```
# Neo4J Cypher code
match p=(a:Start)-[:HOP*1..]->(z:End)
where none(node IN nodes(p) where node.class ='D')
return p

# Neptune openCypher code
match p=(a:Start)-[:HOP*1..]->(z:End)
where size([node IN nodes(p) where node.class = 'D']) = 0
return p
```

列表理解可以實現這些結果，如下所示：

```
all  => size(list_comprehension(list)) = size(list)
any  => size(list_comprehension(list)) >= 1
none => size(list_comprehension(list)) = 0
```

在密碼器中重寫密碼reduce( )函數

該reduce( )函數不是開放密碼規範的一部分。它通常用於創建從列表中的元素數據的聚合。在許多情況下，您可以使用列表理解和UNWIND子句的組合來實現類似的結果。

例如，下列 Cypher 查詢會尋找在安克雷奇 (ANC) 和奧斯汀 (AUS) 之間有一到三個停靠點的路徑上的所有機場，並傳回每個路徑的總距離：

```
MATCH p=(a:airport {code: 'ANC'})-[r:route*1..3]->(z:airport {code: 'AUS'})
RETURN p, reduce(totalDist=0, r in relationships(p) | totalDist + r.dist) AS totalDist
ORDER BY totalDist LIMIT 5
```

您可以在 OpenCypher 為 Neptune 編寫相同的查詢，如下所示：

```
MATCH p=(a:airport {code: 'ANC'})-[r:route*1..3]->(z:airport {code: 'AUS'})
UNWIND [i in relationships(p) | i.dist] AS di
RETURN p, sum(di) AS totalDist
ORDER BY totalDist
```

```
LIMIT 5
```

在密碼中重寫密碼的 FOREACH 子句

FOREACH 子句不是開放密碼規範的一部分。它通常用於在查詢中間更新數據，通常是從路徑中的聚合或元素中更新數據。

作為一個路徑示例，找到安克雷奇（ANC）和奧斯汀（AUS）之間不超過兩個站點的路徑上的所有機場，並在每個機場上設置訪問過的屬性：

```
# Neo4J Example
MATCH p=(:airport {code: 'ANC'})-[*1..2]->({code: 'AUS'})
FOREACH (n IN nodes(p) | SET n.visited = true)

# Neptune openCypher
MATCH p=(:airport {code: 'ANC'})-[*1..2]->({code: 'AUS'})
WITH nodes(p) as airports
UNWIND airports as a
SET a.visited=true
```

另一個例子是：

```
# Neo4J Example
MATCH p=(start)-[*]->(finish)
WHERE start.name = 'A' AND finish.name = 'D'
FOREACH (n IN nodes(p) | SET n.marked = true)

# Neptune openCypher
MATCH p=(start)-[*]->(finish)
WHERE start.name = 'A' AND finish.name = 'D'
UNWIND nodes(p) AS n
SET n.marked = true
```

在 Neptune 中重寫 NEO4j 的 APOC 程序

下面的例子使用開放密碼來取代一些最常用的 APOC 程序。這些例子僅供參考，旨在提供一些有關如何處理常見情況的建議。實際上，每個應用程序都不同，您必須提出自己的策略來提供所需的所有功能。

重寫apoc.export程序

Neptune 使用海王星匯出公用程式，針對各種輸出格式 (例如 CSV 和 JSON) 的完整圖形和基於查詢的匯出提供了一系列選項 (請參閱從 Neptune 資料庫叢集匯出資料 (p. 489))。

重寫apoc.schema程序

Neptune 沒有明確定義的結構描述、索引或條件約束，因此不再需要許多apoc.schema程序。例子是：

- apoc.schema.assert
- apoc.schema.node.constraintExists
- apoc.schema.node.indexExists,
- apoc.schema.relationship.constraintExists
- apoc.schema.relationship.indexExists
- apoc.schema.nodes
- apoc.schema.relationships

Neptune OpenCypher 確實支持檢索與程序執行的相似值，如下所示，但可能會在較大的圖形上遇到性能問題，因為這樣做需要掃描圖形的大部分才能返回答案。

```
# openCypher replacement for apoc.schema.properties.distinct
MATCH (n:airport)
RETURN DISTINCT n.runways
```

```
# openCypher replacement for apoc.schema.properties.distinctCount
MATCH (n:airport)
RETURN DISTINCT n.runways, count(n.runways)
```

apoc.do程序的替代方案

這些程序是用來提供條件查詢執行，很容易使用其他 OpenCypher 子句來實現。在 Neptune 中，至少有兩種方法可以實現類似的行為：

- 一種方法是將 OpenCypher 的列表理解功能與UNWIND子句結合起來。
- 另一種方法是在小精靈中使用選擇（）和合併（）步驟。

這些方法的實例如下所示。

替代方法. 當

```
# Neo4J Example
MATCH (n:airport {region: 'US-AK'})
CALL apoc.do.when(
 n.runways>=3,
 'SET n.is_large_airport=true RETURN n',
 'SET n.is_large_airport=false RETURN n',
 {n:n}
) YIELD value
WITH collect(value.n) as airports
RETURN size([a in airports where a.is_large_airport]) as large_airport_count,
size([a in airports where NOT a.is_large_airport]) as small_airport_count


# Neptune openCypher
MATCH (n:airport {region: 'US-AK'})
WITH n.region as region, collect(n) as airports
WITH [a IN airports where a.runways >= 3] as large_airports,
[a IN airports where a.runways < 3] as small_airports, airports
UNWIND large_airports as la
SET la.is_large_airport=true
WITH DISTINCT small_airports, airports
UNWIND small_airports as la
    SET la.small_airports=true
WITH DISTINCT airports
RETURN size([a in airports where a.is_large_airport]) as large_airport_count,
size([a in airports where NOT a.is_large_airport]) as small_airport_count

#Neptune Gremlin using choose()
g.V().
  has('airport', 'region', 'US-AK').
  choose(
    values('runways').is(lt(3)),
    property(single, 'is_large_airport', false),
    property(single, 'is_large_airport', true)).
  fold().
  project('large_airport_count', 'small_airport_count').
    by(unfold().has('is_large_airport', true).count()).
    by(unfold().has('is_large_airport', false).count())
```

```
 #Neptune Gremlin using coalesce()
g.V().
  has('airport', 'region', 'US-AK').
  coalesce(
    where(values('runways').is(lt(3))).
    property(single, 'is_large_airport', false),
    property(single, 'is_large_airport', true)).
  fold().
  project('large_airport_count', 'small_airport_count').
    by(unfold().has('is_large_airport', true).count()).
    by(unfold().has('is_large_airport', false).count())
```

替代方案例

```
# Neo4J Example
MATCH (n:airport {region: 'US-AK'})
CALL apoc.case([
 n.runways=1, 'RETURN "Has one runway" as b',
 n.runways=2, 'RETURN "Has two runways" as b'
 ],
 'RETURN "Has more than 2 runways" as b'
) YIELD value
RETURN {type: value.b,airport: n}

# Neptune openCypher
MATCH (n:airport {region: 'US-AK'})
WITH n.region as region, collect(n) as airports
WITH [a IN airports where a.runways =1] as single_runway,
[a IN airports where a.runways =2] as double_runway,
[a IN airports where a.runways >2] as many_runway
UNWIND single_runway as sr
    WITH {type: "Has one runway",airport: sr} as res, double_runway, many_runway
WITH DISTINCT double_runway as double_runway, collect(res) as res, many_runway
UNWIND double_runway as dr
    WITH {type: "Has two runways",airport: dr} as two_runways, res, many_runway
WITH collect(two_runways)+res as res, many_runway
UNWIND many_runway as mr
    WITH {type: "Has more than 2 runways",airport: mr} as res2, res, many_runway
WITH collect(res2)+res as res
UNWIND res as r
RETURN r

#Neptune Gremlin using choose()
g.V().
  has('airport', 'region', 'US-AK').
  project('type', 'airport').
    by(
      choose(values('runways')).
        option(1, constant("Has one runway")).
        option(2, constant("Has two runways")).
        option(none, constant("Has more than 2 runways"))).
    by(elementMap())

 #Neptune Gremlin using coalesce()
 g.V().
  has('airport', 'region', 'US-AK').
  project('type', 'airport').
    by(
      coalesce(
        has('runways', 1).constant("Has one runway"),
        has('runways', 2).constant("Has two runways"),
        constant("Has more than 2 runways"))).
    by(elementMap())
```

基於列表的屬性的替代方案

Neptune 目前不支援儲存以清單為基礎的屬性。不過，您可以將清單值儲存為逗號分隔字串，然後使用 andsplit() 函數來建構join()和解建 list 屬性，以取得類似的結果。

例如，如果我們想要將標籤列表保存為屬性，我們可以使用示例 rewrite，該示例顯示瞭如何檢索逗號分隔屬性，然後使用帶有 List 理解的split()和join()函數來實現可比的結果：

```
# Neo4j Example (In this example, tags is a durable list of string.
MATCH (person:person {name: "TeeMan"})
WITH person, [tag in person.tags WHERE NOT (tag IN ['test1', 'test2', 'test3'])] AS newTags
SET person.tags = newTags
RETURN person

# Neptune openCypher
MATCH (person:person {name: "TeeMan"})
WITH person, [tag in split(person.tags, ',') WHERE NOT (tag IN ['test1', 'test2',
 'test3'])] AS newTags
SET person.tags = join(newTags,',')
RETURN person
```

從 Neo4j 遷移到 Neptune 的資源

Neptune 提供數種可協助遷移程序的工具和資源。

幫助從 Neo4j 遷移到 Neptune 的工具

- 開放密碼CheatSheet。
- neo4j-to-neptune-用於將數據從 Neo4j 遷移到 Neptune 的命令行實用程序。
- fully-automated-neo4j-to-neptune — 一個AWS CDK 應用程式，向您展示如何將簡單的 Neo4j 資料庫遷移到 Amazon Neptune。
- csv-to-neptune-bulk-format — 此工具採用基於配置的方法，將一個或多個 CSV 檔案重新格式化為支援的 Neptune 大量載入格式。

部落格文章

- 使用阿帕奇卡夫卡的亞馬遜託管流將數據捕獲從 Neo4j 更改為亞馬遜海王星（2020 年 6 月 22 日）
- 使用 Sanjeet Sahay 的全自動化公用程式將 Neo4j 圖形資料庫遷移到 Amazon Neptune (2020 年 4 月 13 日)

將現有的圖形從阿帕奇 TinkerPop Gemlin 服務器遷移到 Amazon Neptune

如果您想要遷移到 Amazon Neptune 的 Apache TinkerPop Gemlin 服務器中有圖形數據，則需要執行以下步驟：

1. 從 Amazon Simple Storage Service (Amazon S3) 中資料。
2. 將匯出的資料轉換為 Neptune 大量載入器可以匯入的 CSV 格式 (p. 262)。
3. 使用海王星散裝載機 (p. 255)，將資料匯入您已準備好的 Neptune 資料庫叢集。
4. 修改您現有的應用程式以連接到 Neptune 的 Gemlin 端點，並進行必要的變更以符合海王星 Gemlin 實作差異 (p. 83)。

將現有圖形從 RDF 三重商店遷移到 Amazon Neptune

如果您在 RDF/SPARQL 中有要遷移到 Amazon Neptune 的圖形資料，則需要執行以下步驟：

1. 從您的 RDF 三重存儲導出數據。
2. 將匯出的資料轉換為 Neptune 大量載入器可以匯入的格式 (p. 269)。
3. 將組建輸入存放在 Amazon Simple Storage Service (Amazon S3)。
4. 使用海王星散裝載機 (p. 255)，將資料從 Amazon S3 匯入您已準備好的 Neptune 資料庫叢集。
5. 修改您現有的應用程式以連線至海王星的 SPARQL 端點。

如果您想嘗試將屬性圖 CSV 資料遷移到 RDF，可以使用 Amazon Neptune CSV 到 RDF 轉換器。

使用AWS Database Migration Service (AWS DMS) 從關係或 NoSQL 數據庫遷移到 Amazon Neptune

AWS Database Migration Service (AWS DMS) 是一項雲端服務，可讓您輕鬆遷移關聯式資料庫、NoSQL 資料庫、資料倉儲和其他類型的資料存放區。如果您將圖形資料儲存在其中一個AWS DMS支援的關聯式或 NoSQL 資料庫中，AWS DMS可協助您快速安全地遷移至 Neptune，而不需要從目前的資料庫停機。如需詳細資訊，請參閱 用AWS Database Migration Service於從不同的資料存放區將資料載入 Amazon Neptune (p. 292)。

使AWS DMS用的遷移數據流如下：

- 建立AWS DMS表格對應物件。這個 JSON 對象指定哪些表應該從源數據庫讀取，以什麼順序，以及如何命名它們的列。它也可以篩選所複製的資料列，並提供簡單的值轉換，如轉換成小寫或四捨五入。
- 建立 Neptune，`GraphMappingConfig`以指定如何將從來源資料庫擷取的資料載入 Neptune。
  - 對於 RDF 資料 (使用 SPARQL 查詢得來)，`GraphMappingConfig` 是採用 W3 的標準 R2RML 對應語言編寫。
  - 對於屬性圖數據（使用 Gemlin 查詢），`GraphMappingConfig`是一個 JSON 對象，如中所述GraphMappingConfig屬性圖形/格林資料的佈局 (p. 293)
- 在與 Neptune 資料庫叢集相同的 VPC 中建立AWS DMS複寫執行個體，以執行移轉。
- 建立 Amazon S3 儲存貯體，用作暫存要遷移的資料的中介儲存體。
- 執行AWS DMS遷移任務。

用AWS Database Migration Service於從不同的資料存放區將資料載入 Amazon Neptune (p. 292)如需詳細資訊，以及 Chris Smith 的四篇部落格文章：「使用 Database Migration Service (DMS) 從關聯式資AWS料庫在 Amazon Neptune 中填入圖形：」

- 第 1 部分：設置舞台
- 第 2 部分：設計屬性圖模型
- 第 3 部分：設計 RDF 模型
- 整合任務

從火焰圖遷移到 Amazon Neptune

如果您在開放原始碼 Blazegraph RDF 三重存放區中有圖形，您可以使用下列步驟將圖形資料移轉到 Amazon Neptune：

- 提供AWS基礎設施。首先使用AWS CloudFormation 範本佈建所需的 Neptune 基礎結構 (請參閱建立資料庫叢集 (p. 54))。
- 從火焰圖導出數據。從 Blazegraph 導出數據有兩種主要方法，即使用 SPARQL 構造查詢或使用火焰圖導出實用程序。
- 將資料匯入 Neptune。然後，您可以使用 Neptune 工作台和將匯出的資料檔案載入到 Neptune (p. 30)海王星散裝載機 (p. 255)。

這種方法通常也適用於從其他 RDF 三存儲數據庫遷移。

大火圖到 Neptune 的兼容性

在將圖形數據遷移到 Neptune 之前，Blazegraph 和 Neptune 之間存在幾個顯著差異，您應該注意這些差異。這些差異可能需要變更查詢、應用程式架構或兩者，甚至使遷移變得不切實際：

- **`Full-text search`**-在 Blazegraph 中，您可以通過與 Apache Solr 的集成來使用內部全文搜索或外部全文搜索功能。如果您使用上述任一功能，請隨時瞭解 Neptune 支援的全文檢索搜尋功能的最新更新。請參閱 海王星全文搜索 (p. 604)。
- **`Query hints`**-大火圖和 Neptune 都使用查詢提示的概念擴展 SPARQL。在移轉期間，您需要移轉您使用的任何查詢提示。如需 Neptune 支援的最新查詢提示的相關資訊，請參閱查詢提示 (p. 437)。
- 推論 — Blazegraph 在三倍模式下支持推論作為可配置選項，但不支持四邊形模式。Neptune 尚不支持推論。
- 地理空間搜索 — Blazegraph 支持命名空間的配置，以支持地理空間。Neptune 中尚不提供此功能。
- 多租戶 — Blazegraph 支援單一資料庫內的多租戶。在 Neptune 中，透過將資料儲存在具名圖形中，並針對 SPARQL 查詢使用 USING NAMED 子句，或為每個租用戶建立個別的資料庫叢集，來支援多租用戶。
- 聯盟 — Neptune 目前支援 SPARQL 1.1 聯合，可供 Neptune 執行個體存取的位置 (例如私有 VPC 內、跨 VPC 或外部網際網路端點) 存取的位置。視特定設定和所需的同盟端點而定，您可能需要一些額外的網路組態。
- 大火圖標準擴展 — Blazegraph 包含 SPARQL 和 REST API 標準的多個擴展，而 Neptune 僅與標準規範本身兼容。這可能需要更改您的應用程序，或使遷移變得困難。

Neptune 佈建AWS基礎架構

雖然您可以透過AWS Management Console或手動建構所需的AWS基礎結構AWS CLI，但使用 CloudFormation 範本通常比較方便，如下所述：

使用 CloudFormation 範本佈建 Neptune：

1. 導覽至 使用AWS CloudFormation堆疊建立海王星資料庫叢集 (p. 59)。
2. 在您偏好的區域中選擇「啟動堆疊」。
3. 設置所需的參數（堆棧名稱和`EC2SSHKeyPairName`）。同時設定下列選用參數，以簡化移轉程序：
   - 將 `AttachBulkloadIAMRoleToNeptuneCluster` 設定為 true。此參數允許建立適當的 IAM 角色並將其附加到叢集，以便大量載入資料。
   - 設定`NotebookInstanceType`為您偏好的執行個體類型。此參數會建立 Neptune 工作簿，您可用來將大量載入執行至 Neptune 並驗證移轉。

4. 選擇 下一步。
5. 設置您想要的任何其他堆棧選項。
6. 選擇 下一步。
7. 檢閱您的選項並選取兩個核取方塊，以確認AWS CloudFormation 可能需要其他功能。
8. 選擇 Create Stack (建立堆疊)。

堆疊建立程序需要幾分鐘的時間才會完成。

從 Blazegraph 導出數據

下一步是以與 Neptune 批量加載器兼容的格式 (p. 269)從 Blazegraph 中導出數據。

根據資料在 Blazegraph 中的儲存方式 (三個或四邊形) 以及使用中的命名圖形數量，Blazegraph 可能會要求您多次執行匯出程序並產生多個資料檔案：

- 如果資料儲存為三元組，則需要為每個具名圖形執行一次匯出。
- 如果資料儲存為四邊形，您可以選擇以 N-Quads 格式匯出資料，或以三倍格式匯出每個已命名圖形。

下面我們假設您將單一命名空間匯出為 N-Quads，但是您可以針對其他命名空間或所需的匯出格式重複此程序。

如果您需要 Blazegraph 在移轉期間上線且可用，請使用 SPARQL 建構查詢。這需要您使用可存取的 SPARQL 端點來安裝、設定和執行 Blazegraph 執行個體。

如果您不需要 Blazegraph 上線，請使用「 BlazeGraph匯出」公用程式。為此，您必須下載 Blazegraph，並且需要可以訪問數據文件和配置文件，但服務器不需要運行。

使用 SPARQL 構造從火焰圖導出數據

SPARQL 構造是 SPARQL 的一個功能，它返回匹配指定的查詢模板的 RDF 圖形。對於此使用案例，您可以使用它一次將資料匯出一個命名空間，使用如下所示的查詢：

```
CONSTRUCT WHERE { hint:Query hint:analytic "true" . hint:Query hint:constructDistinctSPO
 "false" . ?s ?p ?o }
```

雖然存在其他 RDF 工具來導出此數據，但運行此查詢的最簡單方法是使用 Blazegraph 提供的 REST API 端點。以下腳本演示瞭如何使用 Python (3.6+) 腳本將數據導出為 N-四邊形：

```
import requests

# Configure the URL here: e.g. http://localhost:9999/sparql
url = "http://localhost:9999/sparql"
payload = {'query': 'CONSTRUCT WHERE { hint:Query hint:analytic "true" . hint:Query
 hint:constructDistinctSPO "false" . ?s ?p ?o }'}
# Set the export format to be n-quads
headers = {
'Accept': 'text/x-nquads'
}
# Run the http request
response = requests.request("POST", url, headers=headers, data = payload, files = [])
#open the file in write mode, write the results, and close the file handler
f = open("export.nq", "w")
f.write(response.text)
f.close()
```

如果數據存儲為三元組，則需要更改Accept頭參數，以使用 Blazegraph 存儲 GitHub 庫中指定的值以適當的格式（N-三倍，RDF/XML 或龜形）導出數據。

使用火焰圖導出實用程序導出數據

Blazegraph 包含了一個用於導出數據的實用方法，即ExportKB類。 ExportKB有助於從 Blazegraph 導出數據，但與以前的方法不同，要求在導出運行時服務器處於脫機狀態。這使得它成為您可以在遷移過程中離線使 Blazegraph 離線，或者從數據備份中進行遷移時使用它的理想方法。

您可以在已安裝 Blazegraph 但未執行的電腦上從 Java 命令列執行公用程式。執行此命令的最簡單方法是下載位於上的最新 blazegraph.jar 版本 GitHub。執行此命令需要數個參數：

- **log4j.primary.configuration**— log4j 內容檔案的位置。
- **log4j.configuration**— log4j 內容檔案的位置。
- **output**— 匯出資料的輸出目錄。檔案位於知識庫tar.gz中所記錄的子目錄中。
- **format**— 所需的輸出格式，後跟RWStore.properties檔案的位置。如果您正在使用三元組，則需要將-format參數更改為N-TriplesTurtle、或RDF/XML。

例如，如果您有 Blazegraph 日誌檔案和屬性檔案，請使用下列程式碼將資料匯出為 N-Quads：

```
java -cp blazegraph.jar \
    com.bigdata.rdf.sail.ExportKB \
    -outdir ~/temp/ \
    -format N-Quads \
    ./RWStore.properties
```

如果匯出成功，您會看到如下的輸出：

```
Exporting kb as N-Quads on /home/ec2-user/temp/kb
Effective output directory: /home/ec2-user/temp/kb
Writing /home/ec2-user/temp/kb/kb.properties
Writing /home/ec2-user/temp/kb/data.nq.gz
Done
```

建立 Amazon Simple Storage Service (Amazon Storage Service (Amazon S3) 儲存貯體，並將匯出的資料複製到

從 Blazegraph 匯出資料後，請在與目標 Neptune 資料庫叢集相同的區域中建立 Simple Storage Service (Amazon S3) 儲存貯體，以供 Neptune 大量載入器用於從中匯入資料。

如需如何建立 Amazon S3 儲存貯體，請參閱如何建立 S3 儲存貯體？ 在 Amazon 簡單儲存服務使用者指南中，以及 Amazon 簡單儲存服務使用者指南中有關建立儲存貯體的範例。

如需如何將已匯出到新 Amazon S3 儲存貯體的資料檔案複製指示，請參閱 Amazon 簡單儲存服務使用者指南中的將物件上傳到儲存貯體，或搭配AWS CLI 使用高級 (s3) 命令。您還可以使用如下所示的 Python 代碼逐個複製文件：

```
import boto3

region = 'region name'
bucket_name = 'bucket name'
s3 = boto3.resource('s3')
```

```
s3.meta.client.upload_file('export.nq', bucket_name, 'export.nq')
```

使用 Neptune 批量載入器將資料匯入 Neptune

從 Blazegraph 匯出資料並將其複製到 Amazon S3 儲存貯體之後，您就可以將資料匯入 Neptune 了。與使用 SPARQL 執行負載作業相比，Neptune 具有大量載入程式，可以更快地載入資料，且負荷更少。透過呼叫載入器端點 API，以將存放在已識別 S3 儲存貯體中的資料載入 Neptune 啟動批次載入器程序。

雖然您可以透過直接呼叫載入器 REST 端點來執行此操作，但您必須擁有執行目標 Neptune 執行個體所在的私有 VPC 的存取權。您可以設置防禦主機，SSH 進入該機器，然後運行 cURL 命令，但使用 Neptune 工作台 (p. 30)更容易。

Neptune 工作台是一個預先配置的 Jupyter 筆記本電腦作為亞馬遜 SageMaker筆記本運行，安裝了幾個海王星專用筆記本魔術。這些魔法可簡化常見的 Neptune 作業，例如檢查叢集狀態、執行 SPARQL 和 Gremlin 遍歷，以及執行大量載入作業。

要啟動批量加載過程，請使用`%load`魔術，它提供了一個接口來運行海王星裝載程式 (p. 275)：

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台登入管理主控台，開啟位於
2. 選擇 aws-neptune-blazegraph-to-海王星。
3. 選擇開啟筆記本。
4. 在 Jupyter 的執行個體中，選取現有的筆記本或使用 Python 3 核心建立新筆記本。
5. 在您的記事本中，開啟儲存格`%load`，輸入並執行儲存格。
6. 設定大量載入程式的參數：
   a. 在「來源」中，輸入要匯入的來源檔案位置：`s3://{bucket_name}/{file_name}`。
   b. 對於「格式」，請選擇適當的格式，在此範例中為`nquads`。
   c. 在「載入 ARN」中，輸入`IAMBulkLoad`角色的 ARN (此資訊位於 IAM 主控台的「角色」下)。
7. 選擇 Submit (提交)。

結果包含請求的狀態。批量負載通常是長時間運行的進程，因此響應並不意味著負載已經完成，只是它已經開始。此狀態資訊會定期更新，直到報告工作完成為止。

> Note
>
> 這些信息也可以在部落格文章「移轉到雲端：將 Blazegraph 遷移到 Amazon Neptune」中找到。

將資料載入亞馬遜海王星

有幾種不同的方式可以將圖形資料載入 Amazon 海王星：

- 如果您只須載入相對少量的資料，您可以使用查詢，例如 SPARQL `INSERT` 陳述式或 Gremlin `addV` 和 `addE` 步驟。
- 您可以利用 海王星散裝載機 (p. 255) 來擷取位於外部檔案中的大量資料。大量載入器命令的速度較快，而且其額外負荷比查詢語言命令少。它已針對大型資料集最佳化，且支援 RDF (資源描述架構) 資料和 Gremlin 資料。
- 您可以使用 AWS Database Migration Service (AWS DMS) 從其他資料倉庫匯入資料（用AWS Database Migration Service於從不同的資料存放區將資料載入 Amazon Neptune (p. 292)請參閱和使AWS Database Migration Service用指南）。
- 最後，您可以使用格林的`g.io(URL).read()`步驟來讀取 GraphML（XML 格式），Graph son（JSON 格式）和其他格式的數據文件。詳情請參閱TinkerPop文件。

主題

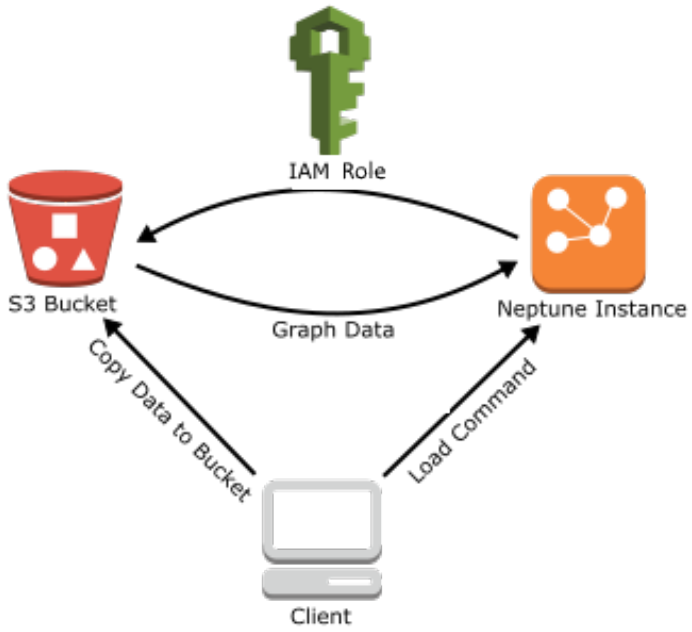
- 使用 Amazon 海王星大容量載入器擷取資料 (p. 255)
- 用AWS Database Migration Service於從不同的資料存放區將資料載入 Amazon Neptune (p. 292)

使用 Amazon 海王星大容量載入器擷取資料

Amazon Neptune 提供可將外部檔案中的資料直接載入 Neptune 資料庫叢集的`Loader`命令。您可以使用這個命令，而不是執行大量的`INSERT`陳述式`addV`、`addE`步驟或其他 API 呼叫。

海王星載入器命令速度更快，負荷更少，針對大型數據集進行了優化，並支持 Gremlin 數據和 SPARQL 使用的 RDF（資源描述框架）數據。

下圖顯示載入程序的概觀：



以下是載入程序的步驟：

1. 將資料檔案複製到亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體。
2. 建立具有讀取和列出儲存貯體的存取權的 IAM 角色。
3. 建立一個亞馬遜 S3 VPC 端點。
4. 透過 HTTP 傳送要求至 Neptune 資料庫執行個體，以啟動 Neptune 載入器。
5. Neptune 資料庫執行個體會擔任 IAM 角色，以便從儲存貯體載入資料。

> **Note**
>
> 如果使用 Amazon S3 SSE-S3 或SSE-KMS模式加密加密的資料，您就可以從 Amazon S3 載入加密的資料，前提是您用於批量載入的角色可以存取 Amazon S3 物件，而且在 SSE-KMS 的情況下也可以存取。`kms:decrypt`然後，Neptune 可以模擬您的憑據並代表您發出`s3:getObject`呼叫。但是，Neptune 目前不支援載入使用該SSE-C模式加密的資料。

以下各節提供準備資料並將資料載入 Neptune 的指示。

主題

- 先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)
- 載入資料格式 (p. 261)
- 範例：將資料載入 Neptune 資料庫執行個體 (p. 270)
- 優化亞馬遜海王星批量負載 (p. 273)
- 海王星載入程式 (p. 274)

先決條件：IAM 角色和亞馬遜 S3 存取

從 Amazon 簡單儲存服務 (Amazon S3) 儲存貯體載入資料需要具有存取儲存貯體的 AWS Identity and Access Management (IAM) 角色。亞馬遜海王星擔任此角色來載入資料。

> **Note**
>
> 如果使用 Amazon S3 SSE-S3 模式加密加密資料，您可以從 Amazon S3 載入加密的資料。在這種情況下，Neptune 能夠模擬您的憑據並代表您發出`s3:getObject`呼叫。
> 只要您的 IAM 角色包含必要的存取許可，您也可以從使用該SSE-KMS模式加密的 Amazon S3 載入加密的資料AWS KMS。若沒有適當的 AWS KMS 許可，大量載入操作會失敗，並傳回 `LOAD_FAILED` 回應。
> 海王星目前不支援載入使用該SSE-C模式加密的 Amazon S3 資料。

以下各節說明如何使用受管 IAM 政策建立 IAM 角色以存取 Amazon S3 資源，然後將該角色附加到 Neptune 叢集。

主題

- 建立 IAM 角色以允許亞馬遜海王星存取亞馬遜 S3 資源 (p. 257)
- 將 IAM 角色新增至亞馬遜海王星叢集 (p. 258)
- 建立亞馬遜 S3 VPC 端點 (p. 258)
- 在亞馬遜海王星鏈接 IAM 角色 (p. 259)

> **Note**
>
> 這些指示要求您必須能夠存取 IAM 主控台和許可，才能管理 IAM 角色和政策。如需詳細資訊，請參閱 IAM 使用者指南中的AWS管理主控台工作的許可。

Amazon Neptune 主控台要求使用者具有下列 IAM 許可，才能將角色附加到 Neptune 叢集：

```
iam:GetAccountSummary on resource: *
iam:ListAccountAliases on resource: *
iam:PassRole on resource: * with iam:PassedToService restricted to
 rds.amazonaws.com
```

建立 IAM 角色以允許亞馬遜海王星存取亞馬遜 S3 資源

使用受AmazonS3ReadOnlyAccess管 IAM 政策建立新的 IAM 角色，以允許 Amazon 海王星存取 Amazon S3 資源。

建立允許海王星存取 Amazon S3 的新 IAM 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 選擇 Create Role (建立角色)。
4. 在「AWS服務」下，選擇 S3。
5. 選擇 Next: Permissions (下一步：許可)。
6. 使用篩選方塊按 S3 一詞進行篩選，並勾選 Amazon ReadOnlyAccess S3 旁邊的方塊。

   Note

   此政策授予對所有儲存貯體的 s3:Get* 和 s3:List* 許可。稍後的步驟會使用信任政策來限制角色的存取權。
   載入程式只需要您從中載入的儲存貯體s3:Get*並獲得s3:List*許可，因此您也可以透過 Amazon S3 資源限制這些許可。
   如果您的 S3 儲存貯體已加密，則必須添加 kms:Decrypt 許可
7. 選擇 下一步：檢閱 。
8. 將「角色名稱」設定為 IAM 角色的名稱，例如：NeptuneLoadFromS3您也可以新增選用的角色描述值，例如「允許 Neptune 代表您存取 Amazon S3 資源」。
9. 選擇 Create Role (建立角色)。
10. 在導覽窗格中，選擇 Roles (角色)。
11. 在 Search (搜尋) 欄位中，輸入您建立的角色名稱，然後在該角色出現於清單時選擇它。
12. 在 Trust Relationships (信任關係) 索引標籤上選擇 Edit trust relationship (編輯信任關係)。
13. 在文字欄位中，貼上以下信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

14. 選擇 Update trust policy (更新信任政策)。
15. 完成「將 IAM 角色新增至亞馬遜海王星叢集 (p. 258)」中的步驟。

將 IAM 角色新增至亞馬遜海王星叢集

使用主控台將 IAM 角色新增至 Amazon 海王星叢集。這可讓叢集中的任何 Neptune 資料庫執行個體擔任該角色並從 Amazon S3 載入。

> **Note**
>
> Amazon Neptune 主控台要求使用者具有下列 IAM 許可，才能將角色附加到 Neptune 叢集：
>
> ```
> iam:GetAccountSummary on resource: *
> iam:ListAccountAliases on resource: *
> iam:PassRole on resource: * with iam:PassedToService restricted to
>  rds.amazonaws.com
> ```

若要將 IAM 角色新增至亞馬遜海王星叢集

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要修改之叢集旁的選項按鈕。
4. 在 Actions (動作) 之下選擇 Manage IAM roles (管理 IAM 角色)。
5. 選擇您在上一節建立的角色。
6. 選擇 Done (完成)。
7. 請等待 IAM 角色可供叢集存取，然後再使用它。

建立亞馬遜 S3 VPC 端點

海王星載入程式需要適用於 Amazon S3 的 VPC 端點。

設定 Amazon S3 的存取

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中選擇 Endpoints (端點)。
3. 選擇 Create Endpoint (建立端點)。
4. 選擇閘道類型端點的服務名稱`com.amazonaws.`*`region`*`.s3`。

   > **Note**
   >
   > 如果此處的區域不正確，請務必確保主控台區域正確。
5. 選擇包含 Neptune 資料庫執行個體的 VPC。
6. 選取與您叢集相關子網路關聯的路由表旁邊的核取方塊。如果您只有一個路由表，您必須選擇此方塊。
7. 選擇 Create Endpoint (建立端點)。

如需建立端點的相關資訊，請參閱 Amazon VPC 使用者指南中的 VPC 端點。如需 VPC 端點 (適用於 Amazon S3 的 VPC 端點) 限制的相關資訊。

後續步驟

現在您已授與 Amazon S3 儲存貯體的存取權，您可以準備載入資料。如需有關支援格式的資訊，請參閱 載入資料格式 (p. 261)。

在亞馬遜海王星鏈接 IAM 角色

將角色附加到叢集時，您的叢集可以扮演該角色來存取 Amazon S3 中存放的資料。從引擎版本 1.2.1.0.R3 (p. 865) 開始，如果該角色無法存取您需要的所有資源，您可以鏈結叢集可假設的一或多個其他角色，以存取其他資源。鏈結中的每個角色都會承擔鏈結中的下一個角色，直到叢集在鏈結末端擔任該角色為止。

若要鏈結角色，您必須在角色之間建立信任關係。例如，要鏈接RoleB到RoleA，RoleA必須具有允許它假設的權限策略RoleB，並且RoleB必須具有允許其將其權限傳回的信任策略RoleA。如需詳細資訊，請參閱使用 IAM 角色。

鏈結中的第一個角色必須附加至載入資料的叢集。

第一個角色，以及在鏈結中擔任下列角色的每個後續角色必須具有：

- 包含對動作Allow影響之特定陳述式的政sts:AssumeRole策。
- Resource元素中下一個角色的亞馬遜資源名稱（ARN）。

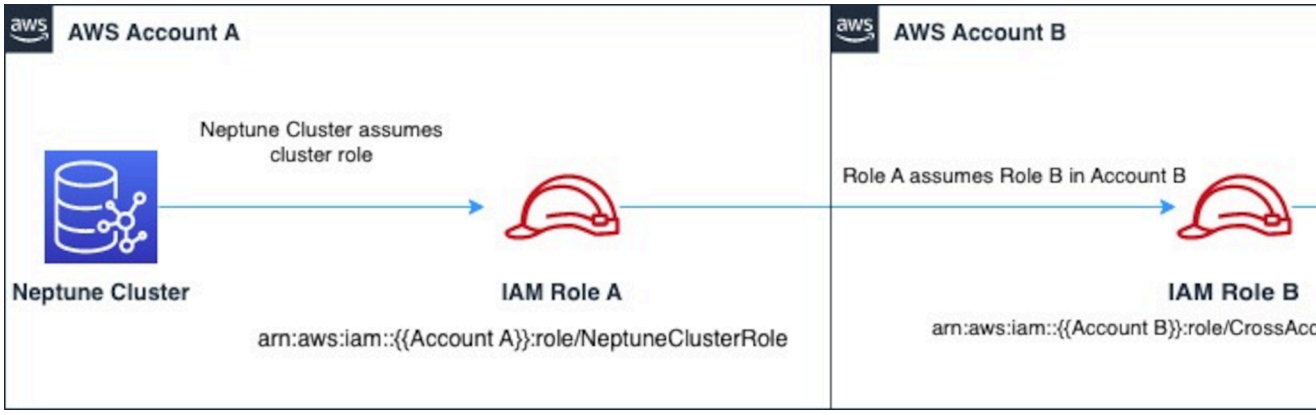
> **Note**
>
> 目標 Amazon S3 儲存貯體必須與叢集位於相同的AWS區域。

使用鏈結角色進行跨帳戶存取

您可以鏈結屬於另一個帳戶的一個或多個角色來授予跨帳戶存取權。當您的叢集暫時擔任屬於其他帳戶的角色時，它可以存取該帳戶的資源。

例如，假設帳戶 A 想要存取屬於帳戶 B 的 Amazon S3 儲存貯體中的資料：

- 帳戶 A 會為名為的 Neptune 建立AWS服務角色，RoleA並將其附加至叢集。
- 帳戶 B 會建立一個名為的角色，RoleB該角色已獲授權存取帳戶 B 值區中的資料。
- 帳戶 A 附加權限原則RoleA，以允許它假設RoleB。
- 帳戶 B 附加信任策略RoleB，允許它將其權限傳回RoleA。
- 若要存取帳戶 B 值區中的資料，帳戶 A 會使用鏈結RoleA和的iamRoleArn參數執行載入程式命令RoleB。在載入器作業期間，暫RoleA時假設存RoleB取帳戶 B 中的 Amazon S3 儲存貯體。



例如，RoleA會有一個與 Neptune 建立信任關係的信任原則：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
          "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

RoleA還會有一個允許它假設的權限策略RoleB，該策略由帳戶 B 擁有：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487639602000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": "arn:aws:iam::(Account B ID):role/RoleB"
    }
  ]
}
```

相反，RoleB將有一個信任政策與以下方面建立信任關係RoleA：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::(Account A ID):role/RoleA"
      }
    }
  ]
}
```

RoleB還需要獲得權限才能訪問位於帳戶 B 中的 Amazon S3 存儲桶中的數據。

建立 AWS Security Token Service (STS) 虛擬私人雲端端點

當您連結 IAM 角色以透過私有 IP 位址私有存取 AWS STS API AWS STS 時，Neptune 載入程式需要 VPC 端點。您可以以安全且可擴展的方式，直接從 Amazon VPC 連線到 AWS STS VPC 端點。當您使用介面 VPC 端點時，它會提供更好的安全性狀態，因為您不需要開啟輸出流量防火牆。它還提供使用 Amazon VPC 端點的其他好處。

使用 VPC 端點時，流量AWS STS不會透過網際網路傳輸，也不會離開 Amazon 網路。您的虛擬私人雲端會安全地連線至網路流量，AWS STS不會造成可用性風險或頻寬限制。如需詳細資訊，請參閱使用AWS STS 介面 VPC 端點。

若要設定 AWS Security Token Service (STS) 的存取權

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中選擇 Endpoints (端點)。
3. 選擇 Create Endpoint (建立端點)。
4. 選擇「服務名稱：」com.amazonaws.region.sts 做為「介面類型」端點。

5. 選擇包含您的 Neptune 資料庫執行個體和 EC2 執行個體的 VPC。
6. 選取 EC2 執行個體所在子網路旁邊的核取方塊。您無法在相同的可用區域內選取多個子網路。
7. 針對 IP address type (IP 地址類型)，從下列選項中選擇：
   - IPv4 - 將 IPv4 地址指派給您的端點網路介面。只有當所有選取的子網都具有 IPv4 地址範圍時，才支援此選項。
   - IPv6 - 將 IPv6 地址指派給您的端點網路介面。只有在所有選取的子網路都是僅 IPv4 子網路時，才支援此選項。
   - Dualstack - 將 IPv4 和 IPv6 地址指派給您的端點網路介面。只有當所有選取的子網都具有 IPv4 和 IPv6 地址範圍時，才支援此選項。
8. 對於 安全群組，選取要與 VPC 端點的端點網路界面建立關聯的安全群組。您需要選取連接到 Neptune 資料庫執行個體和 EC2 執行個體的所有安全群組。
9. 對於 Policy (政策)，選取 Full access (完整存取)，以允許 VPC 端點上所有資源的所有主體進行所有操作。否則，選取 Custom (自訂) 以連接 VPC 端點政策，該政策控制主體在 VPC 端點上對資源執行動作時所具有的許可。只有服務支援 VPC 端點政策時，此選項才可用。如需詳細資訊，請參閱端點策略。
10. (選擇性) 若要新增標籤，請選擇「新增標籤」，然後輸入您想要的標籤關鍵字和標籤值。
11. 選擇 Create endpoint (建立端點)。

如需建立端點的相關資訊，請參閱 Amazon VPC 使用者指南中的 VPC 端點。請注意，Amazon STS 虛擬私人雲端端點是 IAM 角色鏈結的必要先決條件。

現在您已授與AWS STS端點的存取權，您可以準備載入資料。如需支援格式的相關資訊，請參閱載入資料格式 (p. 261)。

在載入器命令中鏈結角色

您可以在執行載入器命令時指定角色鏈結，方法是在參數中加入逗號分隔的角色 ARN 清單。`iamRoleArn`

例如，像這樣的命令會鏈結三個角色：

```
curl -X POST https://localhost:8182/loader \
  -H 'Content-Type: application/json' \
  -d '{
        "source" : "s3://(the target bucket name)/(the target date file name)",
        "iamRoleArn" : "arn:aws:iam::(Account A ID):role/(RoleA),arn:aws:iam::(Account B
 ID):role/(RoleB),arn:aws:iam::(Account C ID):role/(RoleC)",
        "format" : "csv",
        "region" : "us-east-1"
      }'
```

載入資料格式

亞馬遜海王星 Load API 支援載入各種格式的資料。

屬性圖載入格式

然後可以使用小鬼和 OpenCypher 來查詢以下列屬性圖格式之一載入的資料：

- 小精靈加載數據格式 (p. 262)(`csv`)：逗號分隔值 (CSV) 格式。
- 開放密碼資料載入格式 (p. 267) (`opencypher`)：一種逗號分隔值 (CSV) 格式。

RDF 加載格式

若要載入您使用 SPARQL 查詢的資源描述架構 (RDF) 資料，您可以使用下列其中一個由全球資訊網聯盟 (W3C) 指定的標準格式：

- N-三元（`ntriples`）從規範在。https://www.w3.org/TR/n-triples/
- N-四邊形 (`nquads`) 來自規格的位置。https://www.w3.org/TR/n-quads/
- RDF/XML (`rdfxml`) 來自於規格的位置。https://www.w3.org/TR/rdf-syntax-grammar/
- 龜（`turtle`）從規範在https://www.w3.org/TR/turtle/。

載入資料必須使用 UTF-8 編碼

> **Important**
>
> 所有載入資料檔案必須以 UTF-8 格式編碼。如果一個文件不是 UTF-8 編碼，海王星嘗試將其加載為 UTF-8 無論如何。

在包含 Unicode 的 N-Quads 與 N-triples 資料中，支援 `\u`*`xxxxx`* 逸出序列。但是，海王星不支持正常化。如果存在需要標準化的值，則byte-to-byte在查詢期間將不匹配。如需有關正規化的詳細資訊，請參閱 Unicode.org 中的正規化頁面。

如果您的資料不是受支援的格式，您必須在載入資料之前對其進行轉換。

將圖形 ML 轉換為海王星 CSV 格式的工具可在上的專案中使用。GitHub

載入資料檔案的壓縮支援

Neptune 支援以`gzip`或`bzip2`格式壓縮個別檔案。

壓縮檔案必須具有`.gz`或`.bz2`副檔名，且必須是以 UTF-8 格式編碼的單一文字檔案。您可以載入多個檔案，但每個檔案都必須是個別的`.gz`.`bz2`、或未壓縮的文字檔案。不支援包含副檔名的封存檔案`.tar`.`tar.gz`，`.tgz`例如、和。

以下章節將更詳細說明格式。

主題

- 小精靈加載數據格式 (p. 262)
- 開放密碼數據的加載格式 (p. 267)
- RDF 載入資料格式 (p. 269)

小精靈加載數據格式

若要使用 CSV 格式載入 Apache 格TinkerPop林資料，您必須在不同的檔案中指定頂點和邊緣。

載入器可在單一載入工作中，從多點和多邊檔案進行載入。

對於每個載入命令，要載入的檔案集必須位於 Amazon S3 儲存貯體的相同資料夾中，而且您可以指定`source`參數的資料夾名稱。檔案名稱和副檔名不重要。

亞馬遜海王星 CSV 格式遵循 RFC 4180 CSV 規範。如需詳細資訊，請參閱 Internet Engineering Task Force (IETF) 網站上的 Common Format and MIME Type for CSV Files (CSV 檔案的常見格式與 MIME 類型)。

> **Note**
>
> 所有檔案都必須以 UTF-8 格式編碼。

每個檔案都有以逗號分隔的標頭資料列。此標頭資料列包含系統欄標題和屬性欄標頭。

系統欄標題

頂點檔案和邊緣檔案的必要和允許的系統欄標頭是不同的。

每個系統欄只能在標頭中出現一次。

所有標籤皆區分大小寫。

頂點標頭

- ~id - 必要

  頂點的 ID。
- ~label

  頂點的標籤。允許使用多個標籤值，並以分號 (;) 分隔。

  如果~label不存在，則提TinkerPop供帶有值的標籤vertex，因為每個頂點必須至少有一個標籤。

邊緣標頭

- ~id - 必要

  邊緣的 ID。
- ~from - 必要

  從頂點的頂點 ID。
- ~to - 必要

  至頂點的頂點 ID。
- ~label

  邊緣的標籤。邊緣只能有一個標籤。

  如果~label不存在，則提TinkerPop供帶有值的標籤edge，因為每個邊緣都必須有一個標籤。

屬性欄標題

您可以使用下列語法指定屬性的 column (:)。類型名稱不區分大小寫。但是請注意，如果屬性名稱中出現冒號，則必須在其前面加上反斜線:\:來逸出冒號。

```
propertyname:type
```

> **Note**
>
> 欄標題中不允許有空格，因此屬性名稱不可以包含空格。

您可以透過將 [] 新增至類型，來為陣列類型指定一個欄：

```
propertyname:type[]
```

> **Note**
>
> 邊緣屬性只能有一個值，如果指定了陣列類型或指定了第二個值，會導致錯誤。

以下範例顯示名為 age 的 Int 類型屬性的欄標頭。

```
age:Int
```

檔案中的每個資料列在該位置都必須有一個整數或保留空白。

允許使用字符串數組，但數組中的字符串不能包含分號（；）字符，除非使用反斜杠（如下所示）轉義它。\;

指定欄的基數

從 發行版本 1.0.0.200366.0 (2019 年 7 月 26 日) (p. 987) 開始，欄標頭可用來指定依欄識別之屬性的「基數」。這可讓大量載入器遵守 Gremlin 查詢執行的基數相似性。

您指定的欄基數類似這樣：

```
propertyname:type(cardinality)
```

##值可以是 single 或 set。預設值假設為 set，這表示欄可接受多個值。如果是邊緣檔案，基數一律為一個，指定任何其他基數會導致載入器擲出例外狀況。

如果基數是 single，則當載入一或多個值時，若已有舊值，載入器會拋出錯誤。您可以覆寫這種行為，使用 updateSingleCardinalityProperties 標記，讓載入的新值取代現有的值。請參閱 載入器命令 (p. 275)。

您可以使用基數設定與陣列類型，但這通常沒必要。以下是可能的組合：

- name:type— 基數是set, 和內容是單一值.
- name:type[]— 基數是set，並且內容是多值的。
- name:type(single)— 基數是single, 和內容是單一值.
- name:type(set)— 基數與預設值相同，且內容為單值。set
- name:type(set)[]— 基數是set，並且內容是多值的。
- name:type(single)[]-這是矛盾的，並導致錯誤被拋出。

下節列出所有可用的 Gremlin 資料類型。

Gremlin 資料類型

這是一個允許的屬性類型清單及每個類型的描述。

布林 (或布林值)

表示布林值欄位。允許的值：false、true

> **Note**
>
> 所有 true 以外的值會被視為 false。

整數類型

超出定義範圍的值將導致錯誤。

類型	範圍
Byte	-128 to 127
Short	-32768 to 32767
Int	-2^31 to 2^31-1
Long	-2^63 to 2^63-1

小數類型

支援小數符號或科學符號。還允許符號，例如（+/-）無窮大或 NaN。不支援 INF。

類型	範圍
Float	32-bit IEEE 754 floating point

Double	64-bit IEEE 754 floating point

浮點數和雙精確度值過長，將四捨五入到最接近 24 位元 (float) 和 53 位元 (double) 精確度的值並載入。位元層級的剩餘數中段值將四捨五入為 0。

字串

引號是選用的。逗號、新行及換行字元如果包含在由雙引號 (") 包圍的字串中，將會自動逸出。範例：`"Hello, World"`

若要在引號字串中包含引號，您可以在同一列使用兩個引號以逸出引號，例如：`"Hello ""World"""`

允許使用字符串數組，但數組中的字符串不能包含分號（`;`）字符，除非使用反斜杠（如下所示）轉義它。`\;`

如果您要使用引號包圍陣列中的字串，您必須以一組引號包圍整個陣列。範例：`"String one; String 2; String 3"`

日期

ISO-8601 格式的 Java 資料。支援格式如下：`yyyy-MM-dd`、`yyyy-MM-ddTHH:mm`、`yyyy-MM-ddTHH:mm:ss`、`yyyy-MM-ddTHH:mm:ssZ`

Gremlin 列格式

分隔符號

列中的欄位以逗號分隔。記錄以新行或新行後接換行符號加以分隔。

空白欄位

非必要欄允許使用空白欄位 (例如，使用者定義的屬性)。空白欄位仍然需要逗號分隔符號。下一節的範例在每個案例頂點中皆有空白欄位。

頂點 ID

`~id` 值必須是每個頂點檔案的所有頂點中的唯一值。具有相同 `~id` 值的多個頂點列將套用至圖形中的單一頂點。

邊緣 ID

此外，`~id` 值必須是每個邊緣檔案的所有邊緣中的唯一值。具有相同 `~id` 值的多個邊緣列將套用至圖形中的單一邊緣。

標籤

標籤有區分大小寫。

字串值

引號是選用的。逗號、新行及換行字元如果包含在由雙引號 (") 包圍的字串中，將會自動逸出。

CSV 格式規格

海王星 CSV 格式遵循 RFC 4180 CSV 規格，包括下列需求。

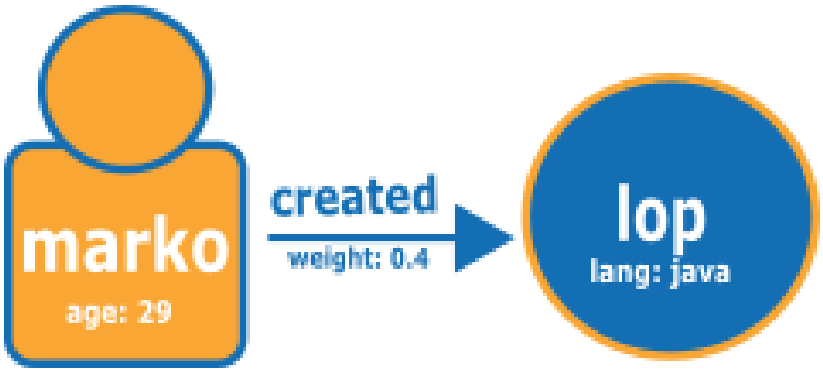
- 支援 Unix 和 Windows 樣式的行尾 (\n 或 \r\n)。
- 任何欄位皆可使用引號 (使用雙引號)。
- 內含分行符號、雙引號或逗號的欄位必須加上引號。(否則，將立即中止載入。)
- 欄位中的雙引號字元 (") 必須由兩個 (雙) 引號字元呈現。例如，`Hello "World"` 字串在資料中必須以 `"Hello ""World"""` 呈現。

- 分隔符號前後的空格將被忽略。如果資料列呈現為 `value1、value2`，它們將儲存為 `"value1"` 和 `"value2"`。
- 任何其他逸出字元將逐字儲存。例如，`"data1\tdata2"` 將儲存為 `"data1\tdata2"`。這些字元只要括在引號中，就不需要逸出。
- 允許空白欄位。空白欄位會視為空的值。
- 欄位中的多個值可在值之間使用分號 (`;`) 加以指定。

如需詳細資訊，請參閱 Internet Engineering Task Force (IETF) 網站上的 Common Format and MIME Type for CSV Files (CSV 檔案的常見格式與 MIME 類型)。

Gremlin 範例

下圖顯示了兩個頂點和從TinkerPop現代圖採取的邊緣的例子。



以下是海王星 CSV 負載格式的圖形。

頂點檔案：

```
~id, name:String, age:Int, lang:String, interests:String[], ~label
v1, "marko", 29, , "sailing;graphs", person
v2, "lop", , "java", , software
```

頂點檔案的表格式檢視：

~id	name:String	age:Int	lang:String	interests:String[]	~label
v1	"marko"	29		["sailing", "graphs"]	person
v2	"lop"		"java"		software

邊緣檔案：

```
~id, ~from, ~to, ~label, weight:Double
e1, v1, v2, created, 0.4
```

邊緣檔案的表格式檢視：

~id	~from	~to	~label	weight:Double
e1	v1	v2	created	0.4

後續步驟

現在您對於載入格式已有更深入的了解，請參閱 範例：將資料載入 Neptune 資料庫執行個體 (p. 270)。

開放密碼數據的加載格式

若要載入使用 OpenCypher CSV 格式的資料，您必須在不同的檔案中指定節點和關係。載入程式可以從單一載入工作中的多個節點檔案和關聯檔案載入。

對於每個載入命令，要載入的檔案集必須在 Amazon 簡單儲存服務值區中具有相同的路徑前綴。您可以在 source 參數中指定該前置詞。實際的檔案名稱和副檔名並不重要。

在亞馬遜海王星中，開放密碼 CSV 格式符合 RFC 4180 CSV 規範。如需詳細資訊，請參閱網際網路工程專責小組 (IETF) 網站上的 CSV 檔案的一般格式和 MIME 類型 (https://tools.ietf.org/html/rfc4180)。

> Note
>
> 這些文件必須以 UTF-8 格式進行編碼。

每個檔案都有一個以逗號分隔的標題列，其中包含系統欄標題和屬性欄標頭。

資料載入檔案中的系統欄標題

給定的系統列只能在每個文件中出現一次。所有系統欄標題標籤都區分大小寫。

OpenCypher 節點載入檔案和關係載入檔案所需和允許的系統欄標題不同：

節點檔案中的系統欄標題

- **`:ID`**— (必要) 節點的 ID。

  可以將可選的 ID 空間添加到節點`:ID`列標題，如下所示：`:ID(`*`ID Space`*`)`。例如，`:ID(movies)`。

  載入連接此檔案中節點的關係時，請在關係檔案`:START_ID`和/或`:END_ID`資料行中使用相同的 ID 空間。

  節點`:ID`資料行可以選擇性地儲存為表單中的屬性*`property name`*`:ID`。例如，`name:ID`。

  節點 ID 在目前和先前載入中的所有節點檔案中應該是唯一的。如果使用 ID 空間，則在目前和之前的載入中使用相同 ID 空間的所有節點檔案中，節點 ID 應該是唯一的。
- **`:LABEL`**— 節點的標籤。

  允許使用多個標籤值，並以分號 (`;`) 分隔。

關係檔案中的系統欄標題

- **`:ID`**— 關係的 ID。這是必需的時候`userProvidedEdgeIds`為 true（默認值），但無效的時候`userProvidedEdgeIds`是`false`。

  在目前和先前載入中的所有關係檔案中，關係 ID 應該是唯一的。
- **`:START_ID`**— (必要) 此關係起始之節點的節點 ID。

  或者，ID 空間可以與表單中的開始 ID 欄相關聯：`:START_ID(`*`ID Space`*`)`。指派給起始節點 ID 的 ID 空間應與指派給節點檔案中節點的 ID 空間相符。
- **`:END_ID`**— (必要) 此關係結束於之節點的節點 ID。

  或者，ID 空間可以與表單中的結束 ID 欄相關聯：`:END_ID(`*`ID Space`*`)`。指派給結束節點 ID 的 ID 空間應與指派給節點檔案中節點的 ID 空間相符。
- **`:TYPE`**-關係的類型。關係只能有單一類型。

Note

如正在載入開放密碼資料 (p. 280)需有關大量載入程序如何處理重複節點或關聯 ID 的資訊，請參閱。

資料載入檔案中的屬性欄標題

您可以使用下列格式的屬性欄標頭，指定資料欄包含特定屬性的值：

```
propertyname:type
```

欄標題中不允許使用空格，屬性名稱中不允許冒號 (:)。以下是名age為 type 的屬性的欄標題範例Int：

```
age:Int
```

age:Int作為列標題的列將必須在每一行中包含一個整數或一個空值。

海王星數據加載文件中的數據類型

- **Bool**or **Boolean**— 布林欄位。允許的值為 true 和 false。

  除true以外的任何值都會被視為false。
- **Byte**— 到範圍內的整-128數127。
- **Short**— 到範圍內的整-32,768數32,767。
- **Int**— 到範圍內的整-2^31數2^31 - 1。
- **Long**— 到範圍內的整-2^63數2^63 - 1。
- **Float**— 一個 32 位元 IEEE 754 浮點數。十進制記數法和科學記數法都支持。 Infinity、-Infinity、和NaN都被認可，但不INF是。

  具有太多位數的值會四捨五入為最接近的值 (對於位元層級的最後一個剩餘數字，中間值會四捨五入為0)。
- **Double**— 一個 64 位元 IEEE 754 浮點數。十進制記數法和科學記數法都支持。 Infinity、-Infinity、和NaN都被認可，但不INF是。

  具有太多位數的值會四捨五入為最接近的值 (對於位元層級的最後一個剩餘數字，中間值會四捨五入為0)。
- **String**— 引號是可選的。如果逗號，換行符和回車符包含在由雙引號（ " ）包圍的字符串中，則它們會自動轉義。"Hello, World"

  您可以在引號字串中包含引號，方法是連續使用兩個，例如"Hello ""World"""。
- **DateTime**— 採用下列其中一種 ISO-8601 格式的 Java 日期：
  - yyyy-MM-dd
  - yyyy-MM-ddTHH:mm
  - yyyy-MM-ddTHH:mm:ss
  - yyyy-MM-ddTHH:mm:ssZ

海王星 OpenCypher 資料載入檔案中的自動轉換資料類型

提供自動轉換資料類型以載入 Neptune 目前不支援的資料類型。這些列中的數據存儲為字符串，逐字存儲，沒有針對其預期格式進行驗證。允許使用下列自動轉換資料類型：

- **Char**— Char 欄位。儲存為字串。
- **Date**、**LocalDateLocalDateTime**、和 — 請參閱 Neo4j 暫時時間，以取得date、localdate和類型的說明。localdatetime這些值會逐字載入為字串，無需驗證。

- **`Duration`**-請參閱 Neo4j 的持續時間格式。這些值會逐字載入為字串，無需驗證。
- Point — 用於儲存空間資料的點欄位。請參閱 Neo4j 的點格式。這些值會逐字載入為字串，無需驗證。

開放密碼負載格式的示例

從TinkerPop現代圖表取下圖顯示了兩個節點和關係的一個例子：



以下是正常海王星 OpenCypher 負載格式的圖形。

節點檔案：

```
:ID,  name:String,  age:Int,  lang:String,  :LABEL
 v1,   "marko",      29,           ,        person
 v2,   "lop",          ,      "java",       software
```

關係檔案：

```
:ID,  :START_ID,  :END_ID,   :TYPE,      weight:Double
 e1,   v1,         v2,        created,    0.4
```

或者，您可以使用 ID 空格和 ID 作為屬性，如下所示：

第一個節點檔案：

```
name:ID(person),    age:Int,  lang:String,  :LABEL
 "marko",            29,           ,       person
```

第二個節點檔案：

```
name:ID(software),  age:Int,  lang:String,  :LABEL
 "lop",               ,        "java",       software
```

關係檔案：

```
:ID,  :START_ID,  :END_ID,   :TYPE,      weight:Double
 e1,   "marko",    "lop",     created,    0.4
```

RDF 載入資料格式

若要載入資源描述架構 (RDF) 資料，您可以使用以下其中一個標準格式，如全球資訊網協會 (W3C) 的規定：

- N-三元（`ntriples`）從規範在 https://www.w3.org/TR/n-triples/

- N-Quads (`nquads`) 取自 https://www.w3.org/TR/n-quads/ 的規格
- RDF/XML (`rdfxml`) 取自 https://www.w3.org/TR/rdf-syntax-grammar/ 的規格
- Turtle (`turtle`) 取自 https://www.w3.org/TR/turtle/ 的規格

> **Important**
>
> 所有檔案都必須以 UTF-8 格式編碼。
> 在包含 Unicode 的 N-Quads 與 N-triples 資料中，支援 \u*xxxxx* 逸出序列。但是，海王星不支持正常化。如果存在需要標準化的值，則byte-to-byte在查詢期間將不匹配。如需有關正規化的詳細資訊，請參閱 Unicode.org 中的正規化頁面。

後續步驟

現在您對於載入格式已有更深入的了解，請參閱 範例：將資料載入 Neptune 資料庫執行個體 (p. 270)。

範例：將資料載入 Neptune 資料庫執行個體

此範例顯示如何將資料載入亞馬遜海王星。除非另有說明，否則您必須從 Amazon 彈性運算雲端 (Amazon EC2) 執行個體與海王星資料庫執行個體位於相同 Amazon 虛擬私有雲端 (VPC) 中的執行個體執行下列步驟。

資料載入範例的先決條件

開始之前，您必須準備好以下項目：

- 海王星資料庫執行個體。

  如需啟動 Neptune 資料庫執行個體的資訊，請參閱建立新的海王星資料庫叢集 (p. 54)。
- 一個亞馬遜簡單存儲服務（亞馬遜 S3）存儲桶，用於將數據文件放入。

  您可以使用現有的儲存貯體。如果您沒有 S3 儲存貯體，請參閱 Amazon S3 入門指南中的建立儲存貯體。
- 要載入的圖形資料，採用 Neptune 載入程式支援的其中一種格式：

  如果您使用 Gemlin 來查詢圖形，Neptune 可以載入 comma-separated-values (CSV) 格式的資料，如中所述。小精靈加載數據格式 (p. 262)

  如果您使用 OpenCypher 來查詢您的圖形，海王星也可以載入特定於 OpenCyphere 的CSV格式的資料，如中所述。開放密碼數據的加載格式 (p. 267)

  如果您使用的是 SPARQL，海王星可以載入多種 RDF 格式的資料，如中所述。RDF 載入資料格式 (p. 269)
- Neptune 資料庫執行個體的 IAM 角色，假設具有允許存取 S3 儲存貯體中資料檔案的 IAM 政策。此政策必須授予讀取與列出許可。

  如需建立可存取 Amazon S3 的角色，然後將其與 Neptune 叢集建立關聯的相關資訊，請參閱先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)。

  > **Note**
  >
  > 海王星 Load API 只需要對資料檔案的讀取權限。IAM 政策不需要允許寫入存取權或存取整個值區。
- 一個亞馬遜 S3 VPC 端點。如需詳細資訊，請參閱 建立亞馬遜 S3 VPC 端點 (p. 270) 一節。

建立亞馬遜 S3 VPC 端點

海王星載入程式需要適用於 Amazon S3 的 VPC 端點。

設定 Amazon S3 的存取

1. 登入 AWS Management Console，並在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在左側導覽窗格中選擇 Endpoints (端點)。
3. 選擇 Create Endpoint (建立端點)。
4. 選擇 Service Name (服務名稱) `com.amazonaws.region.s3`。

   Note

   如果此處的區域不正確，請務必確保主控台區域正確。
5. 選擇包含 Neptune 資料庫執行個體的 VPC。
6. 選取與您叢集相關子網路關聯的路由表旁邊的核取方塊。如果您只有一個路由表，您必須選擇此方塊。
7. 選擇 Create Endpoint (建立端點)。

如需建立端點的相關資訊，請參閱 Amazon VPC 使用者指南中的 VPC 端點。如需 VPC 端點 (適用於 Amazon S3 的 VPC 端點) 限制的相關資訊。

將資料載入 Neptune 資料庫執行個體

1. 將資料檔案複製到 Amazon S3 儲存貯體。S3 儲存貯體必須與載入資料的叢集位於相同的AWS區域。

   您可以使用下列 AWS CLI 命令將檔案複製至儲存貯體。

   Note

   此命令不需要從 Amazon EC2 執行個體執行。

   ```
   aws s3 cp data-file-name s3://bucket-name/object-key-name
   ```

   Note

   在 Amazon S3 中，物件金鑰名稱是檔案的完整路徑，包括檔案名稱。
   範例：在命令 `aws s3 cp datafile.txt s3://examplebucket/mydirectory/datafile.txt` 中，物件鍵名稱為 **`mydirectory/datafile.txt`**。

   或者，您可以使用 AWS Management Console 將檔案上傳至 S3 儲存貯體。在 https://console.aws.amazon.com/s3/ 打開亞馬遜 S3 控制台，然後選擇一個存儲桶。在左上角，選擇 Upload (上傳) 以上傳檔案。
2. 在命令列視窗中，輸入以下內容以執行 Neptune 載入器，並使用適用於端點、Amazon S3 路徑、格式和 IAM 角色 ARN 的正確值。

   `format`參數可以是下列任何一個值：`csv`針對鬼靈、`opencypher`開放密碼、或`ntriples`、、和 RDF。`nquads` `turtle` `rdfxml`如需有關其他參數的資訊，請參閱 海王星裝載程式 (p. 275)。

   如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

   區域參數必須符合叢集和 S3 儲存貯體的區域。

   亞馬遜海王星在以下AWS區域提供：

   - 美國東部 (維吉尼亞北部): `us-east-1`
   - 美國東部 (俄亥俄州): `us-east-2`
   - 美國西部 (加利佛尼亞北部): `us-west-1`
   - 美國西部 (奧勒岡): `us-west-2`

- 加拿大 (中部): ca-central-1
- 南美洲 (聖保羅) : sa-east-1
- 歐洲 (斯德哥爾摩): eu-north-1
- 歐洲 (愛爾蘭): eu-west-1
- 歐洲 (倫敦): eu-west-2
- 歐洲 (巴黎) : eu-west-3
- 歐洲 (法蘭克福): eu-central-1
- 中東 (巴林) : me-south-1
- 中東 (阿聯酋) : me-central-1
- 非洲 (開普敦) : af-south-1
- 亞太區域 (香港): ap-east-1
- 亞太區域 (東京): ap-northeast-1
- 亞太區域 (首爾): ap-northeast-2
- 亞太區域 (新加坡) : ap-southeast-1
- 亞太區域 (雪梨) : ap-southeast-2
- 亞太區域 (孟買) : ap-south-1
- 中國 (北京): cn-north-1
- 中國 (寧夏): cn-northwest-1
- AWSGovCloud(美國西部): us-gov-west-1
- AWSGovCloud(美國東部): us-gov-east-1

```
curl -X POST \
    -H 'Content-Type: application/json' \
    https://your-neptune-endpoint:port/loader -d '
    {
      "source" : "s3://bucket-name/object-key-name",
      "format" : "format",
      "iamRoleArn" : "arn:aws:iam::account-id:role/role-name",
      "region" : "region",
      "failOnError" : "FALSE",
      "parallelism" : "MEDIUM",
      "updateSingleCardinalityProperties" : "FALSE",
      "queueRequest" : "TRUE",
      "dependencies" : ["load_A_id", "load_B_id"]
    }'
```

如需建立 IAM 角色並將其與 Neptune 叢集產生關聯的詳細資訊，請參閱先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)。

> **Note**
>
> 如需載入請求參數的詳細資訊，請參閱 海王星裝載機請求參 (p. 276))。簡而言之：
> 此source參數接受指向單一檔案或資料夾的 Amazon S3 URI。如果您指定資料夾，Neptune 會載入資料夾中的每個資料檔案。
> 資料夾可包含多個頂點檔案和多個邊緣檔案。
> URI 可以是下列任何格式。
>
> - s3://*bucket_name*/*object-key-name*
> - https://s3.amazonaws.com/*bucket_name*/*object-key-name*
> - https://s3-us-east-1.amazonaws.com/*bucket_name*/*object-key-name*

format 參數可為以下任一項：

- 小精靈 CSV 格式 (csv) 用於小精靈屬性圖
- 用於開放密碼屬性圖的開放密碼 CSV 格式 (opencypher)
- N -Triples (ntriples) 格式用於 RDF/SPARQL
- N-Quads (nquads) 格式用於 RDF/SPARQL
- RDF/XML (rdfxml) 格式用於 RDF/SPARQL
- Turtle (turtle) 格式用於 RDF/SPARQL

選用的 parallelism 參數可讓您限制大量載入程序所用的執行緒數目。它可以設定為 LOW、MEDIUM、HIGH 或 OVERSUBSCRIBE。
當 updateSingleCardinalityProperties 設為 "FALSE" 時，如果針對邊緣或單一基數頂點屬性所載入的來源檔案中提供了多個值，則載入器會傳回錯誤。
如果已有執行中的載入任務，則將 queueRequest 設為 "TRUE" 會造成載入請求置於佇列。
dependencies 參數會在已置於佇列的一或多個載入任務順利完成時執行載入請求。

3. Neptune 載入程式會傳回id可讓您檢查狀態或取消載入程序的工作；例如：

```
{
    "status" : "200 OK",
    "payload" : {
        "loadId" : "ef478d76-d9da-4d94-8ff1-08d9d4863aa5"
    }
}
```

4. 輸入下列指令以取得loadId從步驟 3 開始的負載狀態：

```
curl -G 'https://your-neptune-endpoint:port/loader/ef478d76-
d9da-4d94-8ff1-08d9d4863aa5'
```

如果載入狀態列出錯誤，您可以請求更詳細的狀態和錯誤清單。如需詳細資訊和範例，請參閱 海王星裝載機獲取狀態 API (p. 282)。

5. (選用) 取消 Load 工作。

將下列項目輸入至載Delete入程式工作，並id從步驟 3 開始的工作：

```
curl -X DELETE 'https://your-neptune-endpoint:port/loader/ef478d76-
d9da-4d94-8ff1-08d9d4863aa5'
```

成功取消時，DELETE 命令將傳回 HTTP 程式碼 200 OK。

來自已完成載入的載入工作的檔案資料不會還原。資料會保留在 Neptune 資料庫執行個體中。

優化亞馬遜海王星批量負載

使用下列策略將 Neptune 批量負載的載入時間保持在最短：

- 清理您的資料：
  - 載入之前，請務必將資料轉換為支援的資料格式 (p. 261)。
  - 刪除任何重複或已知錯誤。
  - 盡可能減少唯一謂詞 (例如邊緣和頂點的屬性) 的數量。
- 優化您的文件：

  - 如果您從 Amazon S3 儲存貯體載入大型檔案 (例如 CSV 檔案)，載入器會將並行載入的區塊解析成可以平行載入的區塊來為您管理並行處理。使用大量的小文件可能會減慢此過程。
  - 如果您從 Amazon S3 資料夾載入多個檔案，載入程式會先自動載入頂點檔案，然後再載入邊緣檔案。
  - 壓縮文件可減少傳輸時間。加載程序支持源文件的gzip壓縮。
- 檢查您的裝載機設置：
  - 如果您不需要在載入期間執行任何其他作業，請使用OVERSUBSCRIBEparallelism (p. 276)參數。此參數設定會使大量載入程式在執行時使用所有可用的 CPU 資源。通常需要 60%-70% 的 CPU 容量，才能保持作業在 I/O 限制允許的情況下執行。

    Note

    當parallelism設定為OVERSUBSCRIBE或 HIGH (預設設定) 時，載入 OpenCypher 資料時存在執行緒可能會遇到競爭條件和鎖死的風險，進而導致錯誤。LOAD_DATA_DEADLOCK在此情況下，請設定parallelism為較低的設定，然後重試載入。
  - 如果您的載入工作將包含多個載入要求，請使用queueRequest參數。設定TRUE可queueRequest讓 Neptune 將您的請求排入佇列，這樣您就不必等待一個要求完成後再發出另一個請求。
  - 如果您的載入要求正在佇列中，您可以使用dependencies參數設定相依性層級，如此一個工作失敗會導致相依工作失敗。這樣可以防止載入的資料中出現不一致的情況。
  - 如果載入工作將涉及更新先前載入的值，請務必將updateSingleCardinalityProperties參數設定為TRUE。如果不這樣做，載入程式會將嘗試更新現有的單基數值視為錯誤。對於 Gremlin 資料，也會在屬性欄標題中指定基數 (請參閱)。屬性欄標題 (p. 263)
  - 您可以使用failOnError參數來決定發生錯誤時，大量載入作業是否應該失敗或繼續執行。此外，您可以使用mode參數來確定載入工作會從前一個工作失敗的時間點繼續載入，而不是重新載入已載入的資料。
- 擴展 — 在大量載入之前，將資料庫叢集的寫入器執行個體設定為最大大小。請注意，如果執行此操作，您必須同時擴展資料庫叢集中的任何僅供讀取複本執行個體，或將其移除，直到完成載入資料為止。

  當大量載入完成時，請務必再次縮減寫入器執行個體。

  Important

  如果您因為大量載入期間的複寫延遲而遭遇重複僅供讀取複本重新啟動的週期，則您的複本可能無法跟上資料庫叢集中的寫入器。將讀取器縮放為大於寫入器，或在大量載入期間暫時將其移除，然後在完成後重新建立它們。

如需設定載入器要求參數請求參數 (p. 276)的詳細資訊，請參閱。

海王星載入程式

本節說明可從 Neptune 資料庫執行個體的 HTTP 端點取得的亞馬遜海王星 Loader API。

Note

如Neptune 載入器的錯誤和饋送訊息 (p. 854)需載入程式在發生錯誤時傳回的錯誤和摘要訊息清單，請參閱。

內容

- 海王星裝載程式 (p. 275)
  - 海王星加載器請求語 (p. 276)
  - 海王星裝載機請求參 (p. 276)
    - 載入開放密碼資料的特殊注意事項 (p. 280)
  - 海王星裝載機響應語 (p. 281)
  - 海王星裝載機錯 (p. 281)
  - 海王星裝載程式 (p. 282)

- 海王星裝載機獲取狀態 API (p. 282)
  - 海王星裝載機獲取狀態請求 (p. 283)
    - 載入程式取得狀態要求語法 (p. 283)
    - 海王星裝載機獲取狀態請求參數 (p. 283)
  - 海王星裝載機獲取狀態響應 (p. 284)
    - 海王星裝載機獲取狀態響應 JSON 佈局 (p. 284)
    - 海王星裝載機獲取狀態overallStatus和failedFeeds響應對象 (p. 285)
    - 海王星裝載機獲取狀態響應errors對象 (p. 286)
    - 海王星裝載機獲取狀態響應errorLogs對象 (p. 287)
  - 海王星裝載機取得狀態範例 (p. 288)
    - 負載狀態請求示例 (p. 288)
    - 對於 LoADID 的示例請求 (p. 288)
    - 詳細狀態的示例請求 (p. 288)
  - 海王星裝載機取得狀態範例 errorLogs (p. 289)
    - 發生錯誤時的詳細狀態響應示例 (p. 289)
    - 一個Data prefetch task interrupted錯誤的例子 (p. 290)
- 海王星裝載機取消工 (p. 291)
  - 取消工作請求語法 (p. 291)
  - 取消任務請求參數 (p. 291)
  - 取消任務回應語法 (p. 291)
  - 取消任務錯誤 (p. 291)
  - 取消任務錯誤訊息 (p. 291)
  - 取消任務範例 (p. 292)

海王星裝載程式

將資料從 Amazon S3 儲存貯體載入海王星資料庫執行個體。

若要載入資料，您必須將 HTTP POST 請求傳送至 `https://your-neptune-endpoint:port/loader` 端點。您可以用 POST 本文或以 URL 編碼參數來傳送 `loader` 請求的參數。

> **Important**
>
> MIME 類型必須是 `application/json`。

S3 儲存貯體必須與叢集位於相同的AWS區域。

> **Note**
>
> 如果使用 Amazon S3 `SSE-S3` 模式加密加密資料，您可以從 Amazon S3 載入加密的資料。在這種情況下，Neptune 能夠模擬您的憑據並代表您發出`s3:getObject`呼叫。
> 只要您的 IAM 角色包含必要的存取許可，您也可以從使用該`SSE-KMS`模式加密的 Amazon S3 載入加密的資料AWS KMS。若沒有適當的 AWS KMS 許可，大量載入操作會失敗，並傳回 `LOAD_FAILED` 回應。
> 海王星目前不支援載入使用該`SSE-C`模式加密的 Amazon S3 資料。

您不必等待一個載入工作完成，然後再啟動另一個工作。Neptune 可以一次將多達 64 個作業請求排入佇列，前提是它們的`queueRequest`參數都設定為`"TRUE"`。另一方面，如果您不想將載入工作排入佇列，則可以將其`queueRequest`參數設定為 `"FALSE"` (預設值)，如此一來，載入工作就會在另一個工作正在進行中時失敗。

針對只在且須在佇列中指定的先前任務順利完成後才能執行的任務，您可以使用 `dependencies` 參數將這些任務排入佇列。若這麼做，且任一指定的任務失敗的話，則您的任務將不會執行，且其狀態將設為 `LOAD_FAILED_BECAUSE_DEPENDENCY_NOT_SATISFIED`。

海王星加載器請求語

```
{
  "source" : "string",
  "format" : "string",
  "iamRoleArn" : "string",
  "mode": "NEW|RESUME|AUTO",
  "region" : "us-east-1",
  "failOnError" : "string",
  "parallelism" : "string",
  "parserConfiguration" : {
    "baseUri" : "http://base-uri-string",
    "namedGraphUri" : "http://named-graph-string"
  },
  "updateSingleCardinalityProperties" : "string",
  "queueRequest" : "TRUE",
  "dependencies" : ["load_A_id", "load_B_id"]
}
```

海王星裝載機請求參

- **source**— 一個亞馬遜 S3 URI。

  此SOURCE參數接受可識別單一檔案、多個檔案、資料夾或多個資料夾的 Amazon S3 URI。Neptune 會載入指定的任何資料夾中的每個資料檔案。

  URI 可以是下列任何格式。

  - s3://*bucket_name*/*object-key-name*
  - https://s3.amazonaws.com/*bucket_name*/*object-key-name*
  - https://s3.us-east-1.amazonaws.com/*bucket_name*/*object-key-name*

  URI 的object-key-name元素相當於 Amazon S3 ListObjectsAPI 呼叫中的前置詞參數。它會識別指定 Amazon S3 儲存貯體中名稱以該前置詞開頭的所有物件。可以是單個文件或文件夾，也可以是多個文件和/或文件夾。

  指定的一個或多個資料夾可以包含多個頂點檔案和多個邊緣檔案。
- **format**— 資料的格式。如需 Neptune Loader 命令之資料格式的詳細資訊，請參閱使用 Amazon 海王星大容量載入器擷取資料 (p. 255)。

  允許的值

  - **csv**為小鬼 CSV 資料 (p. 262)格式。
  - **opencypher**用於開放密碼 CSV 資料格式 (p. 267)。
  - **ntriples**對於 N-三倍 RDF 數據格式。
  - **nquads**用於 N 四邊形 RDF 資料格式。
  - **rdfxml**適用於 RDF\ XML RDF 資料格式。
  - **turtle**用於海龜 RDF 數據格式。
- **iamRoleArn**— 海王星資料庫執行個體假定 IAM 角色的 Amazon 資源名稱 (ARN) 以存取 S3 儲存貯體。如需建立可存取 Amazon S3 的角色，然後將其與 Neptune 叢集建立關聯的相關資訊，請參閱先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)。

  從引擎版本 1.2.1.0.R3 (p. 865) 開始，如果 Neptune 資料庫執行個體和 Amazon S3 儲存貯體位於不同的帳戶中，您也可以鏈結多個 IAM 角色。AWS在此情況下，會iamRoleArn包含以逗號分隔的角色 ARN 清單，如中所述。在亞馬遜海王星鏈接 IAM 角色 (p. 259)例如：

```
curl -X POST https://localhost:8182/loader \
  -H 'Content-Type: application/json' \
```

```
 -d '{
       "source" : "s3://(the target bucket name)/(the target date file name)",
       "iamRoleArn" : "arn:aws:iam::(Account A ID):role/(RoleA),arn:aws:iam::(Account B
ID):role/(RoleB),arn:aws:iam::(Account C ID):role/(RoleC)",
       "format" : "csv",
       "region" : "us-east-1"
     }'
```

- **region**— region 參數必須與叢集的AWS區域和 S3 儲存貯體相符。

  亞馬遜海王星在以下區域提供：

  - 美國東部 (維吉尼亞北部): us-east-1
  - 美國東部 (俄亥俄州): us-east-2
  - 美國西部 (加利佛尼亞北部): us-west-1
  - 美國西部 (奧勒岡): us-west-2
  - 加拿大 (中部): ca-central-1
  - 南美洲（聖保羅）：sa-east-1
  - 歐洲 (斯德哥爾摩): eu-north-1
  - 歐洲 (愛爾蘭): eu-west-1
  - 歐洲 (倫敦): eu-west-2
  - 歐洲（巴黎）：eu-west-3
  - 歐洲 (法蘭克福): eu-central-1
  - 中東（巴林）：me-south-1
  - 中東（阿聯酋）：me-central-1
  - 非洲（開普敦）：af-south-1
  - 亞太區域 (香港): ap-east-1
  - 亞太區域 (東京): ap-northeast-1
  - 亞太區域 (首爾): ap-northeast-2
  - 亞太區域 (新加坡)：ap-southeast-1
  - 亞太區域 (雪梨)：ap-southeast-2
  - 亞太區域 (孟買)：ap-south-1
  - 中國 (北京): cn-north-1
  - 中國 (寧夏): cn-northwest-1
  - AWSGovCloud(美國西部): us-gov-west-1
  - AWSGovCloud(美國東部): us-gov-east-1
- **mode**— 載入工作模式。

  允許的值：RESUME、NEW、AUTO。

  預設值：AUTO

  - RESUME— 在 RESUME 模式下，載入程式會從此來源尋找先前的載入，如果找到，則會繼續該載入工作。如果找不到先前的載入任務，載入器便會停止。

    載入程式可避免重新載入已順利載入先前任務的檔案。它只會嘗試處理失敗的檔案。如果您卸除先前從 Neptune 叢集載入的資料，則不會在此模式下重新載入該資料。如果先前的載入工作成功載入來自相同來源的所有檔案，則不會重新載入任何內容，且載入程式會傳回成功。
  - NEW— 在 NEW 模式中，不論先前的負載為何，都會建立新的載入要求。您可以使用此模式從 Neptune 叢集卸除先前載入的資料之後，從來源重新載入所有資料，或載入相同來源可用的新資料。
  - AUTO— 在 AUTO 模式下，載入程式會從相同來源尋找先前的載入工作，如果找到工作，就像在RESUME模式中一樣恢復該工作。

若載入器未能從相同來源找到先前的載入任務，則會從該來源載入所有資料，如同 NEW 模式那樣。

- **failOnError**— 用於在錯誤上切換完全停止的標誌。

  允許的值："TRUE"、"FALSE"。

  預設值："TRUE"。

  此參數設為 "FALSE" 時，載入器會嘗試載入指定位置中的所有資料，並略過任何有錯誤的項目。

  此參數設為 "TRUE" 時，載入器會在遇到錯誤時立即停止。到該時間點載入的資料仍然存在。

- **parallelism**— 這是選用參數，可設定此參數以減少大量載入程序使用的執行緒數目。

  允許的值：

  - LOW— 使用的執行緒數目是可用 vCPU 的數目除以 8。
  - MEDIUM— 使用的執行緒數目是可用 vCPU 的數目除以 2。
  - HIGH— 使用的執行緒數目與可用 vCPU 的數目相同。
  - OVERSUBSCRIBE— 使用的執行緒數目是可用 vCPU 數目乘以 2。如果使用此值，大量載入器會佔用所有可用的資源。

    不過，這並不表示OVERSUBSCRIBE設定會產生 100% 的 CPU 使用率。由於負載操作是 I/O 限制，因此預期的最高 CPU 使用率在 60% 到 70% 的範圍內。

  預設值：HIGH

  載入 OpenCypher 資料時，此parallelism設定有時會導致執行緒之間發生鎖死。發生這種情況時，海王星返回LOAD_DATA_DEADLOCK錯誤。通常，您可以通過設置parallelism為較低的設置並重試 load 命令來解決此問題。

- **parserConfiguration**— 具有其他剖析器組態值的選擇性物件。也可選用每個子參數：

名稱	範例值	描述
namedGraphUri	*http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph*	The default graph for all RDF formats when no graph is specified (for non-quads formats and NQUAD entries with no graph). The default is http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph
BaseURI	*http://aws.amazon.com/neptune/default*	The base URI for RDF/XML and Turtle formats. The default is http://aws.amazon.com/neptune/default.
allowEmptyStrings	*true*	Grimlin 使用者需要能夠在載入 CSV 資料時傳遞空字串值 (「」) 作為節點和邊緣屬性。如果設定allowEmptyStrings為 false (預設值)，則這類空字串會被視為空值而不會載入。 如果設定allowEmptyStrings為true，

	載入程式會將空字串視為有效的屬性值，並相應地載入它們。

如需詳細資訊，請參閱SPARQL 預設圖形和具名圖形 (p. 92)。

- **updateSingleCardinalityProperties**— 這是可選參數，可控制大量載入器如何處理單基數頂點或邊緣屬性的新值。載入開放密碼資料時不支援此功能 (請參閱正在載入開放密碼資料 (p. 280))。

  允許的值："TRUE"、"FALSE"。

  預設值："FALSE"。

  根據預設，或當 updateSingleCardinalityProperties 明確設定為 "FALSE" 時，載入器會將新值視為錯誤，因為它違反單一基數。

  另一方面，當 updateSingleCardinalityProperties 設為 "TRUE" 時，大量載入器會以新的值取代現有的值。如果在要載入的來源檔案中提供多個邊緣或單一基數頂點屬性值，則大量載入結尾的最終值可以是這些新值的其中之一。載入器只會保證現有的值已由其中一個新值取代。

- **queueRequest**— 這是選用的旗標參數，可指出載入要求是否可以排入佇列。

  您不需要等待一個載入工作完成，然後再發出下一個工作，因為 Neptune 一次可以佇列多達 64 個工作，前提是它們的queueRequest參數都設定為"TRUE"。

  如果 queueRequest 參數遭省略或設為 "FALSE"，且已有其他執行中的載入任務，則此載入請求將失敗。

  允許的值："TRUE"、"FALSE"。

  預設值："FALSE"。

- **dependencies**— 這是一個可選參數，可以使排隊的負載請求取決於隊列中的一個或多個先前作業的成功完成。

  如果 Neptune 的queueRequest參數設定為，Neptune 可以一次將多達 64 個載入請求排入佇列"TRUE"。dependencies 參數可讓您在一或多個佇列中指定的先前請求順利完成時，執行這類佇列請求。

  例如，如果載入 Job-A 和 Job-B 彼此獨立，但載入 Job-C 須先完成 Job-A 和 Job-B 才能開始，請依照指示執行：

  1. 以任何順序一個接著一個提交 load-job-A 和 load-job-B，並儲存其載入 ID。
  2. 使用兩任務其 dependencies 欄位中的載入 ID 提交 load-job-C：

  ```
  "dependencies" : ["job_A_load_id", "job_B_load_id"]
  ```

  由於 dependencies 參數的關係，大量載入器在 Job-A 和 Job-B 順利完成之前，將不會啟動 Job-C。如果其中任何一個失敗，則不執行 Job-C，且其狀態將設為 LOAD_FAILED_BECAUSE_DEPENDENCY_NOT_SATISFIED。

  您可以使用這種方式設定多個相依性層級，如此一來，一項任務若失敗，將導致直間接仰賴該任務的所有請求遭取消。

- **userProvidedEdgeIds**— 只有在載入包含關聯識別碼的 OpenCypher 資料時，才需要此參數。在載入資料中明確提供 OpenCypher 關係識別碼True時，它必須包含並設定為 (建議使用)。

  如果userProvidedEdgeIds不存在或設定為True，則載入中的每個關係檔案中都必須有:ID欄。

  當存userProvidedEdgeIds在且設定為時False，載入中的關係檔案不得包含:ID欄。相反地，Neptune 載入程式會自動為每個關聯產生一個 ID。

明確提供關係 ID 很有用，如此一來，載入器就可以在修正 CSV 資料中的錯誤後繼續載入，而不必重新載入任何已載入的關聯性。如果尚未明確指派關係 ID，則載入程式無法繼續失敗的載入 (如果必須更正任何關係檔案)，而且必須重新載入所有關係。

- `accessKey`— [已停用] 可存取 S3 儲存貯體和資料檔案之 IAM 角色的存取金鑰 ID。

  建議改用 `iamRoleArn` 參數。如需建立可存取 Amazon S3 的角色，然後將其與 Neptune 叢集建立關聯的相關資訊，請參閱先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)。

  如需詳細資訊，請參閱存取金鑰 (存取金鑰 ID 和私密存取金鑰)。
- `secretKey`— [已停用] 建議改為`iamRoleArn`參數。如需建立可存取 Amazon S3 的角色，然後將其與 Neptune 叢集建立關聯的相關資訊，請參閱先決條件：IAM 角色和亞馬遜 S3 存取 (p. 256)。

  如需詳細資訊，請參閱存取金鑰 (存取金鑰 ID 和私密存取金鑰)。

載入開放密碼資料的特殊注意事項

- 以 CSV 格式載入開啟密碼資料時，必須將格式參數設定為。`opencypher`
- 這個`updateSingleCardinalityProperties`參數不支援 OpenCypher 載入，因為所有 OpenCypher 屬性都有單一基數。OpenCypher 載入格式不支援陣列，如果 ID 值出現一次以上，則會將其視為重複或插入錯誤 (請參閱下文)。
- 海王星加載程序處理它在 OpenCypher 數據中遇到的重複項，如下所示：
  - 如果載入程式遇到具有相同節點 ID 的多個資料列，則會使用下列規則來合併這些資料列：
    - 在行中的所有標籤被添加到節點。
    - 對於每個屬性，只會載入其中一個屬性值。選取要載入的選項是不具決定性的。
  - 如果載入程式遇到具有相同關聯識別碼的多個資料列，則只會載入其中一個資料列。要加載的一個選擇是非清潔劑。
  - 如果載入程式遇到具有現有節點或關係 ID 的載入資料，則載入程式永遠不會更新資料庫中現有節點或關係的屬性值。但是，它會加載節點標籤和屬性不存在於現有的節點或關係。
- 雖然您不必將 ID 指派給關係，但這通常是個好主意 (請參閱上面的`userProvidedEdgeIds`參數)。如果沒有明確的關係 ID，則載入器必須在關聯檔案中發生錯誤時重新載入所有關聯，而不是從失敗處繼續載入。

  此外，如果載入資料不包含明確的關係 ID，則載入程式無法偵測重複的關係。

下面是一個開放密碼加載命令的例子：

```
curl -X POST https://your-neptune-endpoint:port/loader \
     -H 'Content-Type: application/json' \
     -d '
     {
       "source" : "s3://bucket-name/object-key-name",
       "format" : "opencypher",
       "userProvidedEdgeIds": "TRUE",
       "iamRoleArn" : "arn:aws:iam::account-id:role/role-name",
       "region" : "region",
       "failOnError" : "FALSE",
       "parallelism" : "MEDIUM",
     }'
```

加載器響應與正常相同。例如：

```
{
  "status" : "200 OK",
```

```
  "payload" : {
    "loadId" : "guid_as_string"
  }
}
```

海王星裝載機響應語

```
{
    "status" : "200 OK",
    "payload" : {
        "loadId" : "guid_as_string"
    }
}
```

200 OK

成功開始的載入任務會傳回 200 程式碼。

海王星裝載機錯

發生錯誤時，回應的 BODY 中會傳回 JSON 物件。message 物件包含此錯誤的描述。

錯誤類別

- Error 400— 語法錯誤會傳回 HTTP 400 錯誤要求錯誤。描述錯誤的訊息。
- Error 500— 無法處理的有效要求會傳回 HTTP 500 內部伺服器錯誤。描述錯誤的訊息。

以下是來自載入器的可能錯誤訊息及錯誤的描述。

載入器錯誤訊息

- Couldn't find the AWS credential for iam_role_arn (HTTP 400)

  找不到登入資料。對照 IAM 主控台或AWS CLI輸出驗證提供的登入資料。請確定您已iamRoleArn將在中指定的 IAM 角色新增至叢集。
- S3 bucket not found for source (HTTP 400)

  S3 儲存貯體不存在。檢查儲存貯體的名稱。
- The source *source-uri* does not exist/not reachable (HTTP 400)

  在 S3 儲存貯體找不到相符的檔案。
- Unable to connect to S3 endpoint. Provided source = *source-uri* and region = *aws-region* (HTTP 500)

  無法連接到亞馬遜 S3。區域必須符合叢集區域。確定您擁有 VPC 端點。如需有關建立 VPC 端點的資訊，請參閱 建立亞馬遜 S3 VPC 端點 (p. 270)。
- Bucket is not in provided Region (*aws-region*) (HTTP 400)

  儲存貯體必須與 Neptune 資料庫執行個體位於相同的AWS區域。
- Unable to perform S3 list operation (HTTP 400)

  提供的 IAM 使用者或角色沒有儲存貯體或資料夾的 List 許可。檢查儲存貯體的政策或存取控制清單 (ACL)。
- Start new load operation not permitted on a read replica instance (HTTP 405)

  載入是一種寫入操作。在讀取/寫入叢集端點上重試載入。

- `Failed to start load because of unknown error from S3 (HTTP 500)`

  亞馬遜 S3 返回一個未知的錯誤。請聯絡 AWS Support。
- `Invalid S3 access key (HTTP 400)`

  存取金鑰無效。檢查提供的登入資料。
- `Invalid S3 secret key (HTTP 400)`

  私密金鑰無效。檢查提供的登入資料。
- `Max concurrent load limit breached (HTTP 400)`

  如不使用 "queueRequest" : "TRUE" 提交載入請求 ，且某載入任務正在執行中，則此請求將失敗，並顯示此錯誤。
- `Failed to start new load for the source "source name". Max load task queue size limit breached. Limit is 64 (HTTP 400)`

  Neptune 支援一次排隊多達 64 個載入程式作業。若在佇列已包含 64 個任務時，再繼續提交其他載入請求到該佇列，則請求會失敗並顯示此訊息。

海王星裝載程式

Example 請求

以下是使用 `curl` 命令透過 HTTP POST 傳送的請求。它加載海王星 CSV 格式的文件。如需詳細資訊，請參閱小精靈加載數據格式 (p. 262)。

```
curl -X POST \
    -H 'Content-Type: application/json' \
    https://your-neptune-endpoint:port/loader -d '
    {
      "source" : "s3://bucket-name/object-key-name",
      "format" : "csv",
      "iamRoleArn" : "ARN for the IAM role you are using",
      "region" : "region",
      "failOnError" : "FALSE",
      "parallelism" : "MEDIUM",
      "updateSingleCardinalityProperties" : "FALSE",
      "queueRequest" : "FALSE"
    }'
```

Example 回應

```
{
    "status" : "200 OK",
    "payload" : {
        "loadId" : "ef478d76-d9da-4d94-8ff1-08d9d4863aa5"
    }
}
```

海王星裝載機獲取狀態 API

取得 `loader` 工作的狀態。

若要取得載入狀態，您必須將 HTTP GET 請求傳送至 `https://your-neptune-endpoint:port/loader` 端點。若要取得特定載入請求的狀態，您必須包含 `loadId` 做為 URL 參數，或將 `loadId` 連接到 URL 路徑。

Neptune 只會追蹤最近 1,024 個大量載入工作，而且每個作業僅儲存最後 10,000 個錯誤詳細資料。

如Neptune 載入器的錯誤和饋送訊息 (p. 854)需載入程式在發生錯誤時傳回的錯誤和摘要訊息清單，請參閱。

內容

- 海王星裝載機獲取狀態請求 (p. 283)
  - 載入程式取得狀態要求語法 (p. 283)
  - 海王星裝載機獲取狀態請求參數 (p. 283)
- 海王星裝載機獲取狀態響應 (p. 284)
  - 海王星裝載機獲取狀態響應 JSON 佈局 (p. 284)
  - 海王星裝載機獲取狀態overallStatus和failedFeeds響應對象 (p. 285)
  - 海王星裝載機獲取狀態響應errors對象 (p. 286)
  - 海王星裝載機獲取狀態響應errorLogs對象 (p. 287)
- 海王星裝載機取得狀態範例 (p. 288)
  - 負載狀態請求示例 (p. 288)
  - 對於 LoADID 的示例請求 (p. 288)
  - 詳細狀態的示例請求 (p. 288)
- 海王星裝載機取得狀態範例 errorLogs (p. 289)
  - 發生錯誤時的詳細狀態響應示例 (p. 289)
  - 一個Data prefetch task interrupted錯誤的例子 (p. 290)

海王星裝載機獲取狀態請求

載入程式取得狀態要求語法

```
GET https://your-neptune-endpoint:port/loader?loadId=loadId
```

```
GET https://your-neptune-endpoint:port/loader/loadId
```

```
GET https://your-neptune-endpoint:port/loader
```

海王星裝載機獲取狀態請求參數

- **loadId**— 載入工作的 ID。如果您未指定 loadId，會傳回載入 ID 的清單。
- **details**— 包含超出整體狀態的詳細資訊。

  允許的值：TRUE、FALSE。

  預設值：FALSE。
- **errors**— 包括錯誤清單。

  允許的值：TRUE、FALSE。

  預設值：FALSE。

  錯誤清單會分頁。page 和 errorsPerPage 參數可讓您翻閱所有錯誤頁面。
- **page**— 錯誤頁碼。僅適用於 errors 參數設為 TRUE 時。

  允許的值：正整數。

  預設值：1。
- **errorsPerPage**— 每個頁面的錯誤數。僅適用於 errors 參數設為 TRUE 時。

允許的值：正整數。

預設值：10。

- **`limit`**— 要列出的載入 ID 數目。僅適用於藉由傳送 `GET` 請求且未指定 `loadId` 以請求載入 ID 清單時。

  允許值：1 到 100 的正整數。

  預設值：100。
- **`includeQueuedLoads`**— 當要求載入 ID 清單時，可用來排除佇列載入要求的載入 ID 的選用參數。

  Note

  此參數可從海王星引擎版本 1.0. (p. 954) 3.0 開始使用。

  依預設，所有狀態為 `LOAD_IN_QUEUE` 的載入任務的載入 ID 都包含在這類清單中。它們會出現在其他任務的載入 ID 之前，依加入佇列的時間，從最近到最早進行排序。

  允許的值：`TRUE`、`FALSE`。

  預設值：`TRUE`。

海王星裝載機獲取狀態響應

海王星裝載機獲取狀態響應 JSON 佈局

載入程式狀態回應的一般佈局如下：

```
{
    "status" : "200 OK",
    "payload" : {
        "feedCount" : [
            {
                "LOAD_FAILED" : number
            }
        ],
        "overallStatus" : {
            "fullUri" : "s3://bucket/key",
            "runNumber" : number,
            "retryNumber" : number,
            "status" : "string",
            "totalTimeSpent" : number,
            "startTime" : number,
            "totalRecords" : number,
            "totalDuplicates" : number,
            "parsingErrors" : number,
            "datatypeMismatchErrors" : number,
            "insertErrors" : number,
        },
        "failedFeeds" : [
            {
                "fullUri" : "s3://bucket/key",
                "runNumber" : number,
                "retryNumber" : number,
                "status" : "string",
                "totalTimeSpent" : number,
                "startTime" : number,
                "totalRecords" : number,
                "totalDuplicates" : number,
                "parsingErrors" : number,
                "datatypeMismatchErrors" : number,
                "insertErrors" : number,
```

```
            }
        ],
        "errors" : {
            "startIndex" : number,
            "endIndex" : number,
            "loadId" : "string",
            "errorLogs" : [ ]
        }
    }
}
```

海王星裝載機獲取狀態overallStatus和failedFeeds響應對象

針對每個失敗摘要傳回的可能回應 (包括錯誤描述) 與回Get-Status應中的overallStatus物件相同。

所有負載的overallStatus物件中會顯示下列欄位，以及每個失敗摘要的failedFeeds物件：

- **fullUri**— 要載入的一個或多個檔案的 URI。

  類型：字串

  格式：s3://*bucket*/*key*。
- **runNumber**— 此負載或摘要的執行編號。它會在載入重新啟動時遞增。

  類型：unsigned long。
- **retryNumber**— 此負載或摘要的重試編號。它會在載入器自動重試饋送或載入時遞增。

  類型：unsigned long。
- **status**— 負載或摘要的傳回狀態。 LOAD_COMPLETED表示載入成功，沒有任何問題。如需其他載入狀態訊息的清單，請參閱。Neptune 載入器的錯誤和饋送訊息 (p. 854)

  類型：字串。
- **totalTimeSpent**— 剖析和插入負載或摘要資料所花費的時間 (以秒為單位)。這不包含擷取原始檔案清單所花費的時間。

  類型：unsigned long。
- **totalRecords**— 載入或嘗試載入的記錄總數。

  類型：unsigned long。

  請注意，從 CSV 檔案載入時，記錄計數並不是指載入的行數，而是指這些行中個別記錄的數目。例如，採取一個像這樣的小 CSV 文件：

  ```
  ~id,~label,name,team
  'P-1','Player','Stokes','England'
  ```

  海王星會認為這個文件包含 3 條記錄，即：

  ```
  P-1  label Player
  P-1  name  Stokes
  P-1  team  England
  ```
- **totalDuplicates**— 遇到的重複記錄數。

  類型：unsigned long。

  就像totalRecords計數一樣，此值包含 CSV 檔案中個別重複記錄的數目，而不是重複行的數目。以這個小的 CSV 文件為例：

```
~id,~label,name,team
P-2,Player,Kohli,India
P-2,Player,Kohli,India
```

加載後返回的狀態看起來像這樣，報告了 6 條記錄，其中 3 個是重複的：

```
{
  "status": "200 OK",
  "payload": {
    "feedCount": [
      {
        "LOAD_COMPLETED": 1
      }
    ],
    "overallStatus": {
      "fullUri": "(the URI of the CSV file)",
      "runNumber": 1,
      "retryNumber": 0,
      "status": "LOAD_COMPLETED",
      "totalTimeSpent": 3,
      "startTime": 1662131463,
      "totalRecords": 6,
      "totalDuplicates": 3,
      "parsingErrors": 0,
      "datatypeMismatchErrors": 0,
      "insertErrors": 0
    }
  }
}
```

對於開放密碼負載，在以下情況下會計算重複：

- 載入程式會偵測到節點檔案中的資料列具有沒有 ID 空間的 ID，與另一個沒有 ID 空間的 ID 值相同，無論是位於另一個資料列或屬於現有節點。
- 載入程式會偵測到節點檔案中的資料列具有 ID 空間的 ID 空間，該 ID 空間與另一個 ID 空間的 ID 值相同，無論是位於另一個資料列或屬於現有節點。

請參閱 載入開放密碼資料的特殊注意事項 (p. 280)。

- **`parsingErrors`**— 遇到的剖析錯誤數目。

  類型：unsigned long。
- **`datatypeMismatchErrors`**— 資料類型與指定資料不相符的記錄數。

  類型：unsigned long。
- **`insertErrors`**— 由於錯誤而無法插入的記錄數。

  類型：unsigned long。

海王星裝載機獲取狀態響應errors對象

錯誤屬於以下幾類：

- **`Error 400`**— 無效`loadId`傳回 HTTP 錯誤要求`400`錯誤。描述錯誤的訊息。
- **`Error 500`**— 無法處理的有效要求會傳回 HTTP `500` 內部伺服器錯誤。描述錯誤的訊息。

如Neptune 載入器的錯誤和饋送訊息 (p. 854)需載入程式在發生錯誤時傳回的錯誤和摘要訊息清單，請參閱。

發生錯誤時，會在回應中傳回 JSON `errors` 物件，並包含下列欄位：BODY

- **`startIndex`**— 第一個包含錯誤的索引。

  類型：unsigned long。
- **`endIndex`**— 最後一個包含錯誤的索引。

  類型：unsigned long。
- **`loadId`**— 負載的識別碼。您可以使用此 ID 並將 `errors` 參數設為 `TRUE` 以列印載入的錯誤。

  類型：字串。
- **`errorLogs`**— 錯誤清單。

  類型：清單。

海王星裝載機獲取狀態響應errorLogs對象

載入器 Get-Status 回應`errors`中的`errorLogs`物件包含使用下列欄位描述每個錯誤的物件：

- **`errorCode`**— 識別錯誤的性質。

  它可以採用下列其中一個值：
  - `PARSING_ERROR`
  - `S3_ACCESS_DENIED_ERROR`
  - `FROM_OR_TO_VERTEX_ARE_MISSING`
  - `ID_ASSIGNED_TO_MULTIPLE_EDGES`
  - `SINGLE_CARDINALITY_VIOLATION`
  - `FILE_MODIFICATION_OR_DELETION_ERROR`
  - `OUT_OF_MEMORY_ERROR`
  - `INTERNAL_ERROR`（當批量加載器無法確定錯誤類型時返回）。
- **`errorMessage`**— 描述錯誤的訊息。

  這可以是與錯誤代碼相關聯的一般訊息，也可以是包含詳細資訊的特定訊息，例如關於遺失自/到頂點或剖析錯誤的相關資訊。
- **`fileName`**— 摘要的名稱。
- **`recordNum`**— 在發生剖析錯誤的情況下，這是無法剖析的記錄檔案中的記錄編號。如果記錄編號不適用於錯誤，或無法確定，則其設定為零。

例如，如果在 RDF nquads 檔案中遇到如下錯誤的資料列，大量載入程式會產生剖析錯誤：

```
<http://base#subject> |http://base#predicate> <http://base#true> .
```

如您所見，上面一行http中的第二個應該在之前加上<而|不是。狀態響應errorLogs中的生成錯誤對象如下所示：

```
{
    "errorCode" : "PARSING_ERROR",
    "errorMessage" : "Expected '<', found: |",
    "fileName" : "s3://bucket/key",
    "recordNum" : 12345
},
```

海王星裝載機取得狀態範例

負載狀態請求示例

以下是使用 curl 命令透過 HTTP GET 傳送的請求。

```
curl -X GET 'https://your-neptune-endpoint:port/loader/loadId (a UUID)'
```

Example 回應

```
{
    "status" : "200 OK",
    "payload" : {
        "feedCount" : [
            {
                "LOAD_FAILED" : 1
            }
        ],
        "overallStatus" : {
            "datatypeMismatchErrors" : 0,
            "fullUri" : "s3://bucket/key",
            "insertErrors" : 0,
            "parsingErrors" : 5,
            "retryNumber" : 0,
            "runNumber" : 1,
            "status" : "LOAD_FAILED",
            "totalDuplicates" : 0,
            "totalRecords" : 5,
            "totalTimeSpent" : 3.0
        }
    }
}
```

對於 LoADID 的示例請求

以下是使用 curl 命令透過 HTTP GET 傳送的請求。

```
curl -X GET 'https://your-neptune-endpoint:port/loader?limit=3'
```

Example 回應

```
{
    "status" : "200 OK",
    "payload" : {
         "loadIds" : [
            "a2c0ce44-a44b-4517-8cd4-1dc144a8e5b5",
            "09683a01-6f37-4774-bb1b-5620d87f1931",
            "58085eb8-ceb4-4029-a3dc-3840969826b9"
        ]
    }
}
```

詳細狀態的示例請求

以下是使用 curl 命令透過 HTTP GET 傳送的請求。

```
curl -X GET 'https://your-neptune-endpoint:port/loader/loadId (a UUID)?details=true'
```

Example 回應

```
{
    "status" : "200 OK",
    "payload" : {
        "failedFeeds" : [
            {
                "datatypeMismatchErrors" : 0,
                "fullUri" : "s3://bucket/key",
                "insertErrors" : 0,
                "parsingErrors" : 5,
                "retryNumber" : 0,
                "runNumber" : 1,
                "status" : "LOAD_FAILED",
                "totalDuplicates" : 0,
                "totalRecords" : 5,
                "totalTimeSpent" : 3.0
            }
        ],
        "feedCount" : [
            {
                "LOAD_FAILED" : 1
            }
        ],
        "overallStatus" : {
            "datatypeMismatchErrors" : 0,
            "fullUri" : "s3://bucket/key",
            "insertErrors" : 0,
            "parsingErrors" : 5,
            "retryNumber" : 0,
            "runNumber" : 1,
            "status" : "LOAD_FAILED",
            "totalDuplicates" : 0,
            "totalRecords" : 5,
            "totalTimeSpent" : 3.0
        }
    }
}
```

海王星裝載機取得狀態範例 errorLogs

發生錯誤時的詳細狀態響應示例

這是一個通過 HTTP 發送的請求，GET使用curl：

```
curl -X GET 'https://your-neptune-endpoint:port/loader/0a237328-afd5-4574-a0bc-
c29ce5f54802?details=true&errors=true&page=1&errorsPerPage=3'
```

Example 發生錯誤時的詳細回應

這是您可能從上面的查詢中獲得的響應的示例，其中列出了遇到的加載錯誤的errorLogs對象：

```
{
    "status" : "200 OK",
    "payload" : {
        "failedFeeds" : [
            {
                "datatypeMismatchErrors" : 0,
                "fullUri" : "s3://bucket/key",
                "insertErrors" : 0,
                "parsingErrors" : 5,
```

```
                "retryNumber" : 0,
                "runNumber" : 1,
                "status" : "LOAD_FAILED",
                "totalDuplicates" : 0,
                "totalRecords" : 5,
                "totalTimeSpent" : 3.0
            }
        ],
        "feedCount" : [
            {
                "LOAD_FAILED" : 1
            }
        ],
        "overallStatus" : {
            "datatypeMismatchErrors" : 0,
            "fullUri" : "s3://bucket/key",
            "insertErrors" : 0,
            "parsingErrors" : 5,
            "retryNumber" : 0,
            "runNumber" : 1,
            "status" : "LOAD_FAILED",
            "totalDuplicates" : 0,
            "totalRecords" : 5,
            "totalTimeSpent" : 3.0
        },
        "errors" : {
            "endIndex" : 3,
            "errorLogs" : [
                {
                    "errorCode" : "PARSING_ERROR",
                    "errorMessage" : "Expected '<', found: |",
                    "fileName" : "s3://bucket/key",
                    "recordNum" : 1
                },
                {
                    "errorCode" : "PARSING_ERROR",
                    "errorMessage" : "Expected '<', found: |",
                    "fileName" : "s3://bucket/key",
                    "recordNum" : 2
                },
                {
                    "errorCode" : "PARSING_ERROR",
                    "errorMessage" : "Expected '<', found: |",
                    "fileName" : "s3://bucket/key",
                    "recordNum" : 3
                }
            ],
            "loadId" : "0a237328-afd5-4574-a0bc-c29ce5f54802",
            "startIndex" : 1
        }
    }
}
```

一個Data prefetch task interrupted錯誤的例子

有時候當您取得 LOAD_FAILED 狀態並接著請求更多詳細資訊時，這時傳回的錯誤可能為 PARSING_ERROR 並搭配 Data prefetch task interrupted 訊息，如下所示：

```
"errorLogs" : [
    {
        "errorCode" : "PARSING_ERROR",
        "errorMessage" : "Data prefetch task interrupted: Data prefetch task for 11467
 failed",
        "fileName" : "s3://some-source-bucket/some-source-file",
```

```
        "recordNum" : 0
    }
]
```

當資料載入作業發生通常原因不在於您的請求或資料的暫時中斷時，就會出現這個錯誤。通常只要再次執行大量上傳請求，就能解決這個問題。如果您使用的是預設設定，即 "mode":"AUTO" 和 "failOnError":"TRUE"，這時載入器會略過其已成功載入的檔案，並恢復其在中斷發生時尚未載入的檔案載入。

海王星裝載機取消工

取消載入工作。

若要取消工作，則必須傳送 HTTP DELETE請求至 https://*your-neptune-endpoint*:*port*/loader 端點。loadId 可以附加到 /loader URL 路徑，或包含它以做為 URL 中的變數。

取消工作請求語法

```
DELETE https://your-neptune-endpoint:port/loader?loadId=loadId
```

```
DELETE https://your-neptune-endpoint:port/loader/loadId
```

取消任務請求參數

loadId

載入工作的 ID。

取消任務回應語法

```
no response body
```

200 OK

成功刪除的載入任務會傳回 200 程式碼。

取消任務錯誤

發生錯誤時，回應的 BODY 中會傳回 JSON 物件。message 物件包含此錯誤的描述。

錯誤類別

- **Error 400**— 無效loadId傳回 HTTP 錯誤要求400錯誤。描述錯誤的訊息。
- **Error 500**— 無法處理的有效要求會傳回 HTTP 500 內部伺服器錯誤。描述錯誤的訊息。

取消任務錯誤訊息

以下是來自取消 API 的可能錯誤訊息及錯誤的描述。

- The load with id = *load_id* does not exist or not active(HTTP 404) — 找不到載入。檢查 id 參數值。
- Load cancellation is not permitted on a read replica instance.(HTTP 405) — 載入是寫入作業。在讀取/寫入叢集端點上重試載入。

取消任務範例

Example 請求

以下是使用 curl 命令透過 HTTP DELETE 傳送的請求。

```
curl -X DELETE 'https://your-neptune-endpoint:port/loader/0a237328-afd5-4574-a0bc-c29ce5f54802'
```

用AWS Database Migration Service於從不同的資料存放區將資料載入 Amazon Neptune

AWS Database Migration Service(AWS DMS) 可以快速安全地將資料從支援的來源資料庫載入 Neptune。來源資料庫在遷移期間仍然能夠維持完全正常運作，讓倚賴它的應用程式可以將停機時間縮到最短。

您可以在AWS Database Migration Service用戶指南和 AWS Database Migration ServiceAPI 參考AWS DMS中找到有關的詳細信息。特別是，您可以在使用 Amazon Neptune 作為目標中瞭解如何將 Neptune 叢集設定為移轉目標AWS Database Migration Service。

以下是使AWS DMS用以下方式將資料匯入 Neptune 的一些先決條

- 您將需要創建一個AWS DMS表映射對象來定義應如何從源數據庫中提取數據（有關詳細信息，請參閱 AWS DMS UserGuide 中的使用 JSON 通過表映射指定表選擇和轉換）。此資料表對應組態物件指定應依何種順序讀取哪些資料表，以及如何命名資料欄。它也可以篩選所複製的資料列，並提供簡單的值轉換，如轉換成小寫或四捨五入。
- 您將需要創建一個 Neptune，GraphMappingConfig以指定如何將從源數據庫中提取的數據加載到 Neptune 中。對於 RDF 資料 (使用 SPARQL 查詢得來)，GraphMappingConfig 是採用 W3 的標準 R2RML 對應語言編寫。對於屬性圖資料 (使用 Grimlin 查詢)，GraphMappingConfig是 JSON 物件，如中所述。GraphMappingConfig屬性圖形/格林資料的佈局 (p. 293)
- 您必須使AWS DMS用在與 Neptune 資料庫叢集相同的 VPC 中建立複寫執行個體，以調解資料的傳輸。
- 您還需要 Amazon S3 儲存貯體用作暫存遷移資料的中繼儲存。

建立海王星 GraphMappingConfig

您建立的GraphMappingConfig指定應如何將從來源資料存放區擷取的資料載入 Neptune DB 叢集。它的格式會因其用於載入 RDF 資料或載入屬性圖資料而有所不同。

若是 RDF 資料，您可以使用 W3 R2RML 語言將關聯式資料對應到 RDF。

如果您載入的是即將使用 Gremlin 查詢的屬性圖形資料，請為 GraphMappingConfig 建立 JSON 物件。

GraphMappingConfigRDF/SPARQL 資料的配置

如果您要載入要使用 SPARQL 查詢的 RDF 資料，您可以在 R2RML 中GraphMappingConfig編寫。
R2RML是用於將關係數據映射到 RDF 的標準 W3 語言。以下是一個範例：

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix ex: <http://example.com/ns#> .

<#TriplesMap1>
    rr:logicalTable [ rr:tableName "nodes" ];
    rr:subjectMap [
        rr:template "http://data.example.com/employee/{id}";
```

```
        rr:class ex:Employee;
    ];
    rr:predicateObjectMap [
        rr:predicate ex:name;
        rr:objectMap [ rr:column "label" ];
    ] .
```

以下是另一個範例：

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix ex: <http://example.com/#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<#TriplesMap2>
    rr:logicalTable [ rr:tableName "Student" ];
    rr:subjectMap   [ rr:template "http://example.com/{ID}{Name}";
                      rr:class foaf:Person ];
    rr:predicateObjectMap [
        rr:predicate ex:id ;
        rr:objectMap  [ rr:column "ID";
                        rr:datatype xsd:integer ]
    ];
    rr:predicateObjectMap [
        rr:predicate foaf:name ;
        rr:objectMap  [ rr:column "Name" ]
    ] .
```

R2RML: RDB to RDF Mapping Language 的 W3 建議中提供了語言的詳細資訊。

GraphMappingConfig屬性圖形/格林資料的佈局

適用於屬性圖形資料的比較 GraphMappingConfig 是一個 JSON 物件，為即將根據來源資料產生的每個圖形實體提供對應規則。下列範本顯示此物件中每個規則的外觀：

```
{
  "rules": [
    {
      "rule_id": "(an identifier for this rule)",
      "rule_name": "(a name for this rule)",
      "table_name": "(the name of the table or view being loaded)",
      "vertex_definitions": [
        {
          "vertex_id_template": "{col1}",
          "vertex_label": "(the vertex to create)",
          "vertex_definition_id": "(an identifier for this vertex)",
          "vertex_properties": [
            {
              "property_name": "(name of the property)",
              "property_value_template": "{col2} or text",
              "property_value_type": "(data type of the property)"
            }
          ]
        }
      ]
    },
    {
      "rule_id": "(an identifier for this rule)",
      "rule_name": "(a name for this rule)",
      "table_name": "(the name of the table or view being loaded)",
      "edge_definitions": [
        {
```

```
          "from_vertex": {
            "vertex_id_template": "{col1}",
            "vertex_definition_id": "(an identifier for the vertex referenced above)"
          },
          "to_vertex": {
            "vertex_id_template": "{col3}",
            "vertex_definition_id": "(an identifier for the vertex referenced above)"
          },
          "edge_id_template": {
            "label": "(the edge label to add)",
            "template": "{col1}_{col3}"
          },
          "edge_properties":[
            {
              "property_name": "(the property to add)",
              "property_value_template": "{col4} or text",
              "property_value_type": "(data type like String, int, double)"
            }
          ]
        }
      ]
    }
  ]
}
```

請注意，出現頂點標籤表示頂點是在此處建立，沒有頂點標籤則表示頂點是由不同來源建立的，而這項定義只是新增頂點屬性。

以下是員工記錄的規則範例：

```
{
  "rules": [
    {
      "rule_id": "1",
      "rule_name": "vertex_mapping_rule_from_nodes",
      "table_name": "nodes",
      "vertex_definitions": [
        {
          "vertex_id_template": "{emp_id}",
          "vertex_label": "employee",
          "vertex_definition_id": "1",
          "vertex_properties": [
            {
              "property_name": "name",
              "property_value_template": "{emp_name}",
              "property_value_type": "String"
            }
          ]
        }
      ]
    },
    {
      "rule_id": "2",
      "rule_name": "edge_mapping_rule_from_emp",
      "table_name": "nodes",
      "edge_definitions": [
        {
          "from_vertex": {
            "vertex_id_template": "{emp_id}",
            "vertex_definition_id": "1"
          },
          "to_vertex": {
            "vertex_id_template": "{mgr_id}",
            "vertex_definition_id": "1"
```

```
          },
          "edge_id_template": {
            "label": "reportsTo",
            "template": "{emp_id}_{mgr_id}"
          },
          "edge_properties":[
            {
              "property_name": "team",
              "property_value_template": "{team}",
              "property_value_type": "String"
            }
          ]
        }
      ]
    }
  ]
}
```

以 Neptune 為目標建立AWS DMS複寫工作

建立資料表對應和圖形對應組態之後，請使用下列程序將資料從來源存放區載入 Neptune。如需有關所討論 API 的詳細資訊，請參閱 AWS DMS 文件。

步驟 1：建立 AWS DMS 複寫執行個體

在執行 Neptune DB 叢集的 VPC 中建立AWS DMS複寫執行個體 (請參閱使用 AWS DMS 複寫執行個體和 AWS DMS使用指南CreateReplicationInstance中的)。您可以使用如下所示的 AWS CLI 命令來執行此操作：

```
aws dms create-replication-instance \
    --replication-instance-identifier (the replication instance identifier) \
    --replication-instance-class (the size and capacity of the instance, like
 'dms.t2.medium') \
    --allocated-storage (the number of gigabytes to allocate for the instance initially) \
    --engine-version (the DMS engine version that the instance should use) \
    --vpc-security-group-ids (the security group to be used with the instance)
```

步驟 2. 為來源資料庫建立 AWS DMS 端點

下一個步驟是為來源資料存放區建立 AWS DMS 端點。你可以AWS CLI像這樣使用 AWS DMS CreateEndpointAPI：

```
aws dms create-endpoint \
    --endpoint-identifier (source endpoint identifier) \
    --endpoint-type source \
    --engine-name (name of source database engine) \
    --username (user name for database login) \
    --password (password for login) \
    --server-name (name of the server) \
    --port (port number) \
    --database-name (database name)
```

步驟 3. 設定供海王星用於暫存資料的 Amazon S3 儲存貯體

如果您沒有可用於暫存資料的 Amazon S3 儲存貯體，請按照 Amazon S3 入門指南中建立儲存貯體中所述建立儲存貯體，或如何建立 S3 儲存貯體？ 在「主控台使用者指南」中。

如果您還沒有存儲桶 `GetObjectPutObject`，則需要為存儲桶創建一個授予`DeleteObject`和`ListObject`許可的 IAM 政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::(bucket-name)"
      ]
    },
    {
      "Sid": "AllObjectActions",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListObject"
       ],
      "Resource": [
        "arn:aws:s3:::(bucket-name)/*"
      ]
    }
  ]
}
```

如果您的 Neptune 資料庫叢集已啟用 IAM 身分驗證，您還需要包含下列政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "neptune-db:*",
      "Resource": "(the ARN of your Neptune DB cluster resource)"
    }
  ]
}
```

建立 IAM 角色做為信任文件，將政策附加到：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Sid": "neptune",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
```

將原則附加至角色後，將角色附加至 Neptune 資料庫叢集。這樣將使 AWS DMS 能夠使用儲存貯體暫存所載入的資料。

步驟 4. 在海王星 VPC 中建立亞馬遜 S3 端點

現在，在 Neptune 叢集所在的 VPC 中，為中繼 Amazon S3 儲存貯體建立 VPC 閘道端點。您可以使用 AWS Management Console 或 AWS CLI 執行這項操作，如建立閘道端點中所述。

步驟 5. 建立海王星的AWS DMS目標端點

為目標 Neptune 資料庫叢集建立AWS DMS端點。您可以將 AWS DMS CreateEndpointAPI 與NeptuneSettings參數一起使用，如下所示：

```
aws dms create-endpoint \
    --endpoint-identifier (target endpoint identifier) \
    --endpoint-type target \
    --engine-name neptune \
    --server-name (name of the server) \
    --port (port number) \
    --neptune-settings '{ \
      "ServiceAccessRoleArn": "(ARN of the service access role)", \
      "S3BucketName": "(name of S3 bucket to use for staging files when migrating)", \
      "S3BucketFolder": "(name of the folder to use in that S3 bucket)", \
      "ErrorRetryDuration": (number of milliseconds to wait between bulk-load retries), \
      "MaxRetryCount": (the maximum number of times to retry a failing bulk-load job), \
      "MaxFileSize": (maximum file size, in bytes, of the staging files written to S3), \
      "isIamAuthEnabled": (set to true if IAM authentication is enabled on the Neptune
 cluster) }'
```

在其 NeptuneSettings 參數中傳遞至 AWS DMS CreateEndpoint API 的 JSON 物件具有下列欄位：

- **ServiceAccessRoleArn**— (必要) IAM 角色的 ARN，允許對用於將資料移轉至 Neptune 的 S3 儲存貯體進行精細存取。如果啟用 IAM 授權，則此角色也應具有存取 Neptune 資料庫叢集的權限。
- **S3BucketName**— (必要) 對於完整負載移轉，複寫執行個體會將所有 RDS 資料轉換為 CSV 四個檔案，並將其上傳到 S3 中的此暫存儲貯體，然後將它們大量載入 Neptune。
- **S3BucketFolder**— (必要) 要在 S3 暫存貯體中使用的資料夾。
- **ErrorRetryDuration**— (選用) Neptune 要求失敗後等待的毫秒數，再發出重試要求。預設值為 250。
- **MaxRetryCount**— (選用) 可重試失敗之後，AWS DMS應發出的重試要求數目上限。預設值為 5。
- **MaxFileSize**— (選擇性) 移轉期間儲存至 S3 的每個暫存檔案的大小上限 (位元組)。預設值為 1,048,576 KB (1 GB)。
- **IsIAMAuthEnabled**— (選用) 設定為是`true`否在 Neptune 資料庫叢集上啟用 IAM 身份驗證，`false`如果未啟用，則設定為。預設值為 `false`。

步驟 6：測試與新端點的連線

您可以使用網址 AWS DMS =」https://docs.aws.amazon.com/dms/latest/APIReference/API_TestConnection .html "TestConnection> API 來測試與這些新端點的連接，如下所示：

```
aws dms test-connection \
    --replication-instance-arn (the ARN of the replication instance) \
```

```
    --endpoint-arn (the ARN of the endpoint you are testing)
```

步驟 7. 建立 AWS DMS 複寫任務

成功完成上述步驟後，請使用如下所示的 AWS DMS CreateReplicationTaskAPI 建立複寫工作，將來源資料存放區中的資料移轉至 Neptune：

```
aws dms create-replication-task \
    --replication-task-identifier (name for the replication task) \
    --source-endpoint-arn (ARN of the source endpoint) \
    --target-endpoint-arn (ARN of the target endpoint) \
    --replication-instance-arn (ARN of the replication instance) \
    --migration-type full-load \
    --table-mappings (table-mapping JSON object or URI like 'file:///tmp/table-
mappings,json') \
    --task-data (a GraphMappingConfig object or URI like 'file:///tmp/graph-mapping-
config.json')
```

`TaskData`參數提供指定GraphMappingConfig (p. 292)要複製的資料應該如何儲存在 Neptune 中。

步驟 8. 啟動 AWS DMS 複寫任務

現在您可以啟動複寫任務：

```
aws dms start-replication-task
    --replication-task-arn (ARN of the replication task started in the previous step)
    --start-replication-task-type start-replication
```

查詢海王星圖

Neptune 支持以下圖形查詢語言來訪問圖形：

- 小鬼，由 Apache 定義用TinkerPop於創建和查詢屬性圖。

  在 Gemlin 查詢是由離散步驟組成的遍歷，每個步驟都遵循一個邊緣到一個節點。

  請參閱訪問與小鬼海王星圖 (p. 300)以了解如何在海王星中使用 Grimlin，並亞馬遜海王星的小精靈標準合規 (p. 83)查找有關怪靈海王星實施的具體細節。
- 开放密码 (p. 382)是屬性圖的聲明式查詢語言，該語言最初由 Neo4j 開發，然後在 2015 年開源，並在 Apache 2 開源許可證下為 OpenCypher 項目做出了貢獻。它的語法記錄在開放密碼規範中。
- SPARQL 是一種基於圖形模式匹配的聲明式語言，用於查詢 RDF 數據。它是由萬維網聯盟的支持。

  請參閱使用 SPARQL 存取海王星圖 (p. 422)以了解如何在海王星中使用 SPARQL，亞馬遜海王星的 SPARQL 標準合規性 (p. 92)以及尋找有關 SPARQL 海王星實作的特定詳細資料。

  > **Note**
  >
  > Grinlin 和 OpenCypher 都可以用來查詢存儲在海王星中的任何屬性圖數據，而不管它是如何加載的。

主題

- 亞馬遜海王星中的查詢隊列 (p. 299)
- 訪問與小鬼海王星圖 (p. 300)
- 使用開放密碼訪問海王星圖 (p. 382)
- 使用 SPARQL 存取海王星圖 (p. 422)

亞馬遜海王星中的查詢隊列

開發和調校圖形應用程式時，了解資料庫將查詢排入佇列的方式會很有幫助。在亞馬遜海王星中，查詢佇列的發生方式如下：

- 無論執行個體大小為何，每個執行個體可以在佇列中等候的查詢數目上限為 8,192 個。超過該數目的任何查詢都會被拒絕，並且會失敗而產生 `ThrottlingException`。
- 一次可執行的查詢數目上限取決於指派的工作者執行緒數目，通常該數目設定為可用虛擬 CPU 核心 (vCPU) 數目的兩倍。
- 查詢延遲包含查詢花費在佇列中的時間，以及網路往返時間和實際執行時間。

判定佇列在指定時刻有多少個查詢

`MainRequestQueuePendingRequests` CloudWatch 指標會以五分鐘的精細程度記錄輸入佇列中等待的請求數目 (參閱 海王星CloudWatch度量 (p. 774))。

若是 Gremlin，您可以使用 小鬼查詢狀態 API (p. 324) 傳回的 `acceptedQueryCount` 值來取得佇列中目前的查詢計數。但請注意，查詢狀態 API (p. 449) 傳回的 `acceptedQueryCount` 值包含從伺服器啟動以來接受的所有查詢，包括完成的查詢。

查詢佇列如何影響逾時時間

如上所述，查詢延遲包含查詢花費在佇列中的時間，以及執行所花費的時間。

由於查詢的逾時期間通常是從查詢進入佇列起算，因此移動緩慢的佇列可能會使許多查詢一移出佇列就立即逾時。這個情況顯然不好，所以應避免將大量查詢排入佇列，除非它們可以快速執行。

訪問與小鬼海王星圖

亞馬遜海王星與阿帕奇 TinkerPop 3 和小鬼兼容。這表示您可以連接到海王星資料庫執行個體，並使用 Gremlin 遍歷語言來查詢圖形 (請參閱 Apache TinkerPop 3 文件中的圖形)。有關怪靈海王星實施的差異，請參閱。小鬼符合標準 (p. 83)

不同的海王星引擎版本支持不同的小鬼版本。檢查您正在運行的 Neptune 版本的引擎發布頁面 (p. 856)，以確定它支持哪個 Grinlin 發行版本。

Gremlin 中的「周遊」是一系列的鏈結步驟。它從頂點 (或邊緣) 開始。它先沿著每個頂點的外緣，再沿著這些頂點的外緣導出圖形。每個步驟皆是周遊的操作。如需詳細資訊，請參閱 TinkerPop 3 文件中的遍歷。

有 Gremlin 語言變體和各種程式設計語言中的 Gremlin 存取支援。如需詳細資訊，請參閱 TinkerPop 3 文件中的關於 Gemlin 語言變體。

本文件說明如何使用下列變體和程式設計語言存取 Neptune。

如中所述傳輸中加密：使用 SSL/HTTPS 連接到海王星 (p. 154)，連線至所有區域的 Neptune 時，您必須使用傳輸層安全性/安全通訊端層 (TLS/SSL)。AWS

Gremlin-Groovy

本節中的 Gremlin 主控台和 HTTP REST 範例將使用 Gremlin-Groovy 變體。如需有關 Gemlin 主控台和 Amazon 海王星的詳細資訊，請參閱快速入門the section called “使用小精靈” (p. 67)章節。

Gremlin-Java

Java 示例是使用官方的 TinkerPop 3 Java 實現編寫的，並使用小鬼 Java 變體。

Gremlin-Python

Python 示例是使用官方 TinkerPop 3 Python 實現編寫的，並使用小鬼 Python 變體。

以下各節將逐步說明如何透過 HTTPS 使用 Gremlin 主控台，REST以及各種程式設計語言連線到 Neptune 資料庫執行個體。

開始之前，您必須準備好以下事項：

- 海王星資料庫執行個體。如需建立 Neptune 資料庫執行個體的資訊，請參閱建立新的海王星資料庫叢集 (p. 54)。
- 與您的 Neptune 資料庫執行個體位於相同虛擬私有雲 (VPC) 中的 Amazon EC2 執行個體。

如需將資料載入 Neptune 的詳細資訊，包括必要條件、載入格式和載入參數，請參閱將資料載入亞馬遜海王星 (p. 255)。

主題

- 將 Gimlin 主控台設定為連線到海王星資料庫執行個體 (p. 301)
- 使用 HTTPS REST 端點連線至海王星資料庫執行個體 (p. 304)
- 基於 Java 的小鬼客戶端與亞馬遜海王星一起使用 (p. 305)
- 使用 Python 連接到海王星資料庫執行個體 (p. 313)
- 使用 .NET 連線至 Neptune 資料庫執行個體 (p. 314)
- 使用 Node.js 連線至海王星資料庫執行個體 (p. 316)
- 使用 Go 連線至 Neptune 資料庫執行個體 (p. 317)
- 小精靈查詢提示 (p. 319)

- 小鬼查詢狀態 API (p. 324)
- 小精靈查詢取消 (p. 326)
- 支持基於小鬼腳本的會話 (p. 326)
- 海王星小鬼交易 (p. 328)
- 在亞馬遜海王星中使用小鬼 API (p. 329)
- 在亞馬遜海王星小鬼中緩存查詢結果 (p. 330)
- 用小鬼mergeV()和步驟製作高效的升頻者 mergeE() (p. 334)
- 使用製作高效的小鬼提升者 fold()/coalesce()/unfold() (p. 338)
- 使用小靈分析海王星查詢執行 explain (p. 347)
- 使用小鬼與海王星 DFE 查詢引擎 (p. 381)

將 Gimlin 主控台設定為連線到海王星資料庫執行個體

Grimlin 控制台允許您在 REPL（read-eval-print循環）環境中嘗試TinkerPop圖形和查詢。

安裝 Girmlin 控制台並以通常的方式連接到它

您可以使用 Gremlin 主控台連接到遠端圖形資料庫。以下部分將逐步引導您如何安裝和設定 Gemlin 主控台，以便從遠端連線到 Neptune 資料庫執行個體。您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

如需使用 SSL/TLS 連線至海王星的說明 (這是必要的)，請參閱。SSL/TLS 組態 (p. 310)

> **Note**
>
> 如果您已在 Neptune 資料庫叢集上啟用 IAM 身份驗證 (p. 160)，請遵循中使用具有簽名版本 4 簽名的 Gimlin 控制台連接到海王星 (p. 163)的指示安裝 Gemlin 主控台，而不是此處的指示。

安裝小鬼控制台並連接到海王星

1. 小鬼控制台二進製文件需要 Java 8 或 Java 11。這些說明假設使用 Java 11。您可以在 EC2 實例上安裝 Java 11，如下所示：

   - 如果您使用的是亞馬遜 2（AL2）：

     ```
     sudo amazon-linux-extras install java-openjdk11
     ```

   - 如果您使用的是亞馬遜 AL2023

     ```
     sudo yum install java-11-amazon-corretto-devel
     ```

   - 對於其他發行版，請使用下列任一項適當：

     ```
     sudo yum install java-11-openjdk-devel
     ```

     或:

     ```
     sudo apt-get install openjdk-11-jdk
     ```

2. 輸入以下內容，將 Java 11 設定為 EC2 執行個體上的預設執行階段。

   ```
   sudo /usr/sbin/alternatives --config java
   ```

   出現提示時，請輸入 Java 11 的號碼。

3. 從 Apache 網站下載適當版本的小鬼控制台。您可以查看目前正在執行的 Neptune 引擎版本的引擎版本的引擎版本頁面 (p. 856)，以判斷它支援的 Gremlin 版本。例如，對於版本 3.6.2，您可以從 Apache Tinkerpop3 網站下載鬼靈控制台到您的 EC2 實例，如下所示：

```
wget https://archive.apache.org/dist/tinkerpop/3.6.2/apache-tinkerpop-gremlin-
console-3.6.2-bin.zip
```

4. 解壓縮 Gremlin 主控台 zip 檔。

```
unzip apache-tinkerpop-gremlin-console-3.6.2-bin.zip
```

5. 將目錄變更為解壓縮的目錄。

```
cd apache-tinkerpop-gremlin-console-3.6.2
```

6. 在解壓縮目錄的 conf 根資料夾中，使用下列文字建立名為 neptune-remote.yaml 的檔案。將*###### ###### Nep* tune 資料庫執行個體的主機名稱或 IP 位址。需使用方括號 ([ ])。

   Note

   如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

```
hosts: [your-neptune-endpoint]
port: 8182
connectionPool: { enableSsl: true }
serializer: { className:
 org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1,
              config: { serializeResultToString: true }}
```

7. 在終端機上，導覽至 Gremlin 主控台目錄 (apache-tinkerpop-gremlin-console-3.6.2)，然後輸入下列命令來執行 Gremlin 主控台。

```
bin/gremlin.sh
```

   您應該會看到下列輸出：

```
         \,,,/
         (o o)
-----oOOo-(3)-oOOo-----
plugin activated: tinkerpop.server
plugin activated: tinkerpop.utilities
plugin activated: tinkerpop.tinkergraph
gremlin>
```

   您現在進入 gremlin> 提示。您將在這個提示下輸入其餘的步驟。

8. 出現gremlin>提示時，輸入以下指令以連線至 Neptune 資料庫執行個體。

```
:remote connect tinkerpop.server conf/neptune-remote.yaml
```

9. 在 gremlin> 提示下，輸入以下內容以切換為遠端模式。這會傳送所有 Gremlin 查詢到遠端連線。

```
:remote console
```

10. 輸入以下內容以將查詢傳送到 Gremlin 圖形。

```
g.V().limit(1)
```

11. 完成後，輸入以下內容以退出 Gremlin 主控台。

```
:exit
```

Note

使用分號 (;) 或換行符號字元 (\n) 來分隔每個陳述式。
最終周遊之前的每個周遊節尾必須為 `next()`，才能執行。但只有最後的周遊資料會傳回。

有關 Gremlin 的海王星實現的更多信息，請參閱。the section called “小鬼符合標準” (p. 83)

連接到小鬼控制台的另一種方式

正常連接方法的缺點

連接到 Gemlin 控制台的最常用方法是上面解釋的方法，在提示符下使用類似這樣的`gremlin>`命令：

```
gremlin> :remote connect tinkerpop.server conf/(file name).yaml
gremlin> :remote console
```

這很好，並且可以讓您將查詢發送到海王星。但是，它需要 Groovy 腳本引擎出循環，所以海王星將所有查詢視為純粹的 Grimlin。這表示下列查詢表單會失敗：

```
gremlin> 1 + 1
gremlin> x = g.V().count()
```

以這種方式連接時，最接近使用變量的方法是使用控制台維護的`result`變量並使用發送查詢`:>`，如下所示：

```
gremlin> :remote console
==>All scripts will now be evaluated locally - type ':remote console' to return
 to remote mode for Gremlin Server - [krl-1-cluster.cluster-ro-cm9t6tfwbtsr.us-
east-1.neptune.amazonaws.com/172.31.19.217:8182]
gremlin> :> g.V().count()
==>4249

gremlin> println(result)
[result{object=4249 class=java.lang.Long}]

gremlin> println(result['object'])
[4249]
```

不同的連接方式

您也可以以不同的方式連接到 Gemlin 控制台，您可能會發現更好，如下所示：

```
gremlin> g = traversal().withRemote('conf/neptune.properties')
```

這裡`neptune.properties`採用這種形式：

```
gremlin.remote.remoteConnectionClass=org.apache.tinkerpop.gremlin.driver.remote.DriverRemoteConnection
gremlin.remote.driver.clusterFile=conf/my-cluster.yaml
gremlin.remote.driver.sourceName=g
```

該`my-cluster.yaml`文件應該如下所示：

```
hosts: [my-cluster-abcdefghijk.us-east-1.neptune.amazonaws.com]
port: 8182
serializer: { className:
 org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1,
              config: { serializeResultToString: false } }
connectionPool: { enableSsl: true }
```

像這樣配置 Gimlin 控制台連接可以讓您成功地進行以下類型的查詢：

```
gremlin> 1+1
==>2

gremlin> x=g.V().count().next()
==>4249

gremlin> println("The answer was ${x}")
The answer was 4249
```

您可以避免顯示結果，如下所示：

```
gremlin> x=g.V().count().next();[]
gremlin> println(x)
4249
```

所有常用的查詢方式（沒有終端步驟）繼續工作。例如：

```
gremlin> g.V().count()
==>4249
```

您甚至可以使用該`g.io().read()`步驟加載具有此類連接的文件。

使用 HTTPS REST 端點連線至海王星資料庫執行個體

亞馬遜海王星為小鬼查詢提供 HTTPS 端點。REST 接口與您的數據庫集群正在使用的任何 Gemlin 版本兼容（請參閱您正在運行的 Neptune 引擎版本的引擎發布頁 (p. 856)面，以確定它支持的 Gemlin 發行版本）。

> **Note**
>
> 如中所述傳輸中加密：使用 SSL/HTTPS 連接到海王星 (p. 154)，海王星現在要求您使用 HTTPS 而不是 HTTP 進行連接。

以下說明引導您使用 `curl` 命令和 HTTPS 連接到 Gremlin 端點。您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

用於向海王星資料庫執行個體進行 Grimlin 查詢的 HTTPS 端點為。`https://`*`your-neptune-endpoint`*`:`*`port`*`/gremlin`

> **Note**
>
> 如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)。

若要使用 HTTP REST 端點連線至海王星

以下範例使用 curl 透過 HTTP POST 提交 Gremlin 查詢。查詢是以 POST 本文中的 JSON 格式做為 `gremlin` 屬性提交。

```
curl -X POST -d '{"gremlin":"g.V().limit(1)"}' https://your-neptune-endpoint:port/gremlin
```

這個例子通過使用g.V().limit(1)遍歷返回圖形中的第一個頂點。您可以通過用另一個 Gremlin 遍歷替換它來查詢其他東西。

Important

根據預設，REST 端點會傳回單一 JSON 結果集中的所有結果。如果此結果集太大，Neptune 資料庫執行個體上可能會發生OutOfMemoryError例外狀況。
您可以啟用區塊回應 (以一系列個別回應傳回的結果) 來避免這種情況。請參閱 使用可選的 HTTP 尾隨標題來啟用多部分小鬼響應 (p. 305)。

雖然建議使用 HTTP POST 請求來發送小靈查詢，但也可以使用 HTTP GET 請求：

```
curl -G "https://your-neptune-endpoint:port?gremlin=g.V().count()"
```

Note

海王星不支持該bindings屬性。

使用可選的 HTTP 尾隨標題來啟用多部分小鬼響應

默認情況下，對格林查詢的 HTTP 響應在單個 JSON 結果集中返回。如果是非常大的結果集，這可能會導致資料庫執行個體發生OutOfMemoryError例外狀況。

不過，您可以啟用區塊回應 (在多個不同部分傳回的回應)。您可以在要求中加入轉移編碼 (TE) 預告片標頭 (te: trailers) 來達成此目的。如需 TE 標頭的詳細資訊，請參閱有關 TE 要求標頭的 MDN 頁面。

當回應分為多個部分傳回時，很難診斷接收到第一個零件之後發生的問題，因為第一部分送達的 HTTP 狀態碼為 200 (OK)。後續失敗通常會導致包含損毀回應的郵件內文，而 Neptune 會在此結尾附加錯誤訊息。

為了更輕鬆地檢測和診斷此類故障，Neptune 還在每個響應塊的尾隨標題中包含兩個新的標頭字段：

- X-Neptune-Status— 包含回應碼，後面接著簡短名稱。例如，在成功的情況下，尾隨標題將是：X-Neptune-Status: 200 OK。在發生故障的情況下，響應代碼將是 Neptune 引擎錯誤代碼 (p. 849)之一，例如X-Neptune-Status: 500 TimeLimitExceededException。
- X-Neptune-Detail— 對於成功的請求而言是空的。在錯誤的情況下，它包含 JSON 錯誤消息。由於 HTTP 標頭值中只允許使用 ASCII 字元，因此 JSON 字串會進行網址編碼。

Note

Neptune 目前不支援gzip壓縮區塊回應。如果用戶端同時要求區塊編碼和壓縮，Neptune 會略過壓縮。

基於 Java 的小鬼客戶端與亞馬遜海王星一起使用

您可以在亞馬遜海王星中使用兩個基於 Java 的開源 Gremlin 客戶端中的任何一個：Apache TinkerPop Java Gemlin 客戶端或亞馬遜海王星的 Gemlin 客戶端。

阿帕奇 TinkerPop Java 小鬼客戶端

如果可以的話，請始終使用引擎版本支持的最新版本的 Apache TinkerPop Java Gemlin 客戶端。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。

下表列出不同 Neptune 引擎版本所支援的最早和最新版本的TinkerPop用戶端：

海王星引擎版	最低 TinkerPop 版本	最大TinkerPop版本
1.2.1.0	3.6.2	3.6.2

海王星引擎版	最低 TinkerPop 版本	最大TinkerPop版本
1.2.0.2	3.5.2	3.5.5
1.2.0.1	3.5.2	3.5.5
1.2.0.0	3.5.2	3.5.5
1.1.1.0	3.5.2	3.5.5
1.1.0.0	3.4.0	3.4.13
1.0.5.1和更老	(已取代)	(已取代)

TinkerPop客戶端通常在一個系列中向後兼容（3.3.x例如，或3.4.x）。在特殊情況下，向後相容性必須中斷，因此最好在TinkerPop升級到新的用戶端版本之前檢查升級建議。

用戶端可能無法使用在伺服器支援之後的版本中引入的新步驟或新功能，但除非升級建議要求重大變更，否則您可以預期現有的查詢和功能可以運作。

> Note
>
> 從海王星引擎TinkerPop版本 1.1.1.0 (p. 894) 開始，請勿使用低於. 3.5.2
> Python 使用者應避免使用TinkerPop版本，3.4.9因為需要直接設定的預設逾時設定 (請參閱 TINKERPOP-2505)。

亞馬遜海王星的小精靈 Java 客戶端

亞馬遜海王星的 Gremlin 客戶端是一個基於 Java 的開源 Grimlin 客戶端，可作為標準 Java 客戶端的插入替代品。TinkerPop

海王星 Girmlin 用戶端已針對海王星叢集進行了最佳化。它可讓您管理叢集中多個執行個體的流量分佈，並在新增或移除複本時適應叢集拓撲中的變更。您甚至可以設定用戶端根據角色、執行個體類型、可用區域 (AZ) 或與執行個體相關聯的標籤，將要求分配到叢集中的執行個體子集。

海王星小鬼 Java 客戶端的最新版本可以在 Maven 中心找到。

有關海王星 Grimlin Java 客戶端的更多信息，請參閱此博客文章。有關代碼示例和演示，請查看客戶的 GitHub項目。

使用 Java 客戶端連接到海王星數據庫實例

以下章節將逐步引導您完成連線到 Neptune 資料庫執行個體並使用 Apache Gemlin 用戶端執行 Gemlin 周遊的完整 Java 範例的執行過程。TinkerPop

必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中的 Amazon EC2 執行個體執行這些指示。

若要使用 Java 連線至海王星

1. 在您的 EC2 實例上安裝阿帕奇 Maven。首先，輸入以下內容，以新增包含 Maven 套件的儲存庫：

```
sudo wget https://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-
maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
```

輸入以下內容以設定套件的版本編號：

```
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
```

接著使用 yum 安裝 Maven：

```
sudo yum install -y apache-maven
```

2. 安裝爪哇。小精靈庫需要 Java 8 或 11。您可以安裝 Java 11，如下所示：

   - 如果您使用的是亞馬遜 2（AL2）：

     ```
     sudo amazon-linux-extras install java-openjdk11
     ```

   - 如果您使用的是亞馬遜 AL2023

     ```
     sudo yum install java-11-amazon-corretto-devel
     ```

   - 對於其他發行版，請使用下列任一項適當：

     ```
     sudo yum install java-11-openjdk-devel
     ```

     或:

     ```
     sudo apt-get install openjdk-11-jdk
     ```

3. 將 Java 11 設置為 EC2 實例的默認運行時：輸入以下內容以將 Java 8 設置為 EC2 實例的默認運行時：

   ```
   sudo /usr/sbin/alternatives --config java
   ```

   出現提示時，請輸入 Java 11 的號碼。

4. 創建一個名為的新目錄**`gremlinjava`**：

   ```
   mkdir gremlinjava
   cd gremlinjava
   ```

5. 在 `gremlinjava` 目錄中，建立一個 `pom.xml` 檔案，然後在文字編輯器中開啟檔案：

   ```
   nano pom.xml
   ```

6. 將以下內容 `pom.xml` 檔案，然後儲存檔案：

   ```
   <project xmlns="https://maven.apache.org/POM/4.0.0"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://
   maven.apache.org/maven-v4_0_0.xsd">
     <properties>
       <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
     </properties>
     <modelVersion>4.0.0</modelVersion>
     <groupId>com.amazonaws</groupId>
     <artifactId>GremlinExample</artifactId>
     <packaging>jar</packaging>
     <version>1.0-SNAPSHOT</version>
     <name>GremlinExample</name>
     <url>https://maven.apache.org</url>
     <dependencies>
       <dependency>
         <groupId>org.apache.tinkerpop</groupId>
         <artifactId>gremlin-driver</artifactId>
         <version>3.6.2</version>
   ```

```
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.apache.tinkerpop/gremlin-groovy
      (Not needed for TinkerPop version 3.5.2 and up)
    <dependency>
      <groupId>org.apache.tinkerpop</groupId>
      <artifactId>gremlin-groovy</artifactId>
      <version>3.6.2</version>
    </dependency> -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-jdk14</artifactId>
      <version>1.7.25</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>2.5.1</version>
        <configuration>
          <source>11</source>
          <target>11</target>
        </configuration>
      </plugin>
        <plugin>
          <groupId>org.codehaus.mojo</groupId>
          <artifactId>exec-maven-plugin</artifactId>
          <version>1.3</version>
          <configuration>
            <executable>java</executable>
            <arguments>
              <argument>-classpath</argument>
              <classpath/>
              <argument>com.amazonaws.App</argument>
            </arguments>
            <mainClass>com.amazonaws.App</mainClass>
            <complianceLevel>1.11</complianceLevel>
            <killAfter>-1</killAfter>
          </configuration>
        </plugin>
    </plugins>
  </build>
</project>
```

Note

如果您要修改現有的 Maven 專案，所需的相依性將在先前的程式碼中反白顯示。

7. 若要為原始程式碼範例 (src/main/java/com/amazonaws/) 建立子目錄，在命令列輸入下列命令：

```
mkdir -p src/main/java/com/amazonaws/
```

8. 在 src/main/java/com/amazonaws/ 目錄中，建立一個名為 App.java 的檔案，然後在文字編輯器中開啟檔案。

```
nano src/main/java/com/amazonaws/App.java
```

9. 將以下內容複製到 App.java 檔案。將*########## Neptune* 資料庫執行個體的位址。請勿在addContactPoint方法中包含https://前置詞。

Note

如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)。

```
package com.amazonaws;
import org.apache.tinkerpop.gremlin.driver.Cluster;
import org.apache.tinkerpop.gremlin.driver.Client;
import org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversalSource;
import org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversal;
import static
 org.apache.tinkerpop.gremlin.process.traversal.AnonymousTraversalSource.traversal;
import org.apache.tinkerpop.gremlin.driver.remote.DriverRemoteConnection;
import org.apache.tinkerpop.gremlin.structure.T;

public class App
{
  public static void main( String[] args )
  {
    Cluster.Builder builder = Cluster.build();
    builder.addContactPoint("your-neptune-endpoint");
    builder.port(8182);
    builder.enableSsl(true);

    Cluster cluster = builder.create();

    GraphTraversalSource g =
 traversal().withRemote(DriverRemoteConnection.using(cluster));

    // Add a vertex.
    // Note that a Gremlin terminal step, e.g. iterate(), is required to make a request
 to the remote server.
    // The full list of Gremlin terminal steps is at https://tinkerpop.apache.org/docs/
current/reference/#terminal-steps
    g.addV("Person").property("Name", "Justin").iterate();

    // Add a vertex with a user-supplied ID.
    g.addV("Custom Label").property(T.id, "CustomId1").property("name", "Custom id
 vertex 1").iterate();
    g.addV("Custom Label").property(T.id, "CustomId2").property("name", "Custom id
 vertex 2").iterate();

    g.addE("Edge Label").from(__.V("CustomId1")).to(__.V("CustomId2")).iterate();

    // This gets the vertices, only.
    GraphTraversal t = g.V().limit(3).elementMap();

    t.forEachRemaining(
      e ->  System.out.println(t.toList())
    );

    cluster.close();
  }
}
```

如需使用 SSL/TLS 連線至海王星的說明 (這是必要的)，請參閱。SSL/TLS 組態 (p. 310)

10. 使用下列 Maven 命令編譯並執行範例：

```
mvn compile exec:exec
```

上述範例使用 g.V().limit(3).elementMap() 周遊傳回圖形中前兩個頂點的金鑰和每個屬性值的對應。若要查詢其他內容，將其換成其他使用其中一個適當之結束方法的 Gremlin 周遊。

Note

Gremlin 查詢最後的部分 .toList() 用來提交周遊至伺服器，以供進行評估。如果未包含該方法或其他對等方法，則不會將查詢提交至 Neptune 資料庫執行個體。
加入頂點或邊時，您還必須附加適當的端點，例如當您使用addV( )步驟時。

下列方法會將查詢提交至 Neptune 資料庫執行個體：

- toList()
- toSet()
- next()
- nextTraverser()
- iterate()

小鬼 Java 客戶端的 SSL/TLS 配置

海王星需要在預設情況下啟用 SSL/TLS。一般而言，如果 Java 驅動程式是使用設定的enableSsl(true)，它可以連線到 Neptune，而不需要設定trustStore()或keyStore()使用憑證的本機副本。早期版本的TinkerPop鼓勵使用keyCertChainFile()來配置本地存儲的.pem文件，但該文件已被棄用，並且在 3.5.x 之後不再可用。如果您將該安裝程式與公用憑證搭配使用SFSRootCAG2.pem，您現在可以移除本機副本。

不過，如果您連線的執行個體沒有網際網路連線來驗證公用憑證，或者您使用的憑證不是公開的，您可以採取下列步驟來設定本機憑證副本：

設定本機憑證副本以啟用 SSL/TLS

1. 從甲骨文下載並安裝密鑰工具。這將使設置本地密鑰存儲更加容易。
2. 下載 SFSRootCAG2.pem CA 證書（小靈 Java SDK 需要一個證書來驗證遠程證書）：

```
wget https://www.amazontrust.com/repository/SFSRootCAG2.pem
```

3. 以 JKS 或 PKCS12 格式建立金鑰存放區。此範例使用 JKS。回答提示時接下來的問題。稍後將需要您在此處創建的密碼：

```
keytool -genkey -alias (host name) -keyalg RSA -keystore server.jks
```

4. 將您下載的SFSRootCAG2.pem檔案匯入至新建立的金鑰存放區：

```
keytool -import -keystore server.jks -file .pem
```

5. 以編程方式配置Cluster對象：

```
Cluster cluster = Cluster.build("(your neptune endpoint)")
                         .port(8182)
                         .enableSSL(true)
                         .keyStore('server.jks')
                         .keyStorePassword("(the password from step 2)")
                         .create();
```

如果需要，您可以在配置文件中執行相同的操作，就像使用 Gemlin 控制台一樣：

```
hosts: [(your neptune endpoint)]
```

```
port: 8182
connectionPool: { enableSsl: true, keyStore: server.jks, keyStorePassword: (the
 password from step 2) }
serializer: { className:
 org.apache.tinkerpop.gremlin.driver.ser.GraphBinaryMessageSerializerV1, config:
 { serializeResultToString: true }}
```

使用重新連線邏輯連線至 Neptune 資料庫執行個體的 Java 範例

以下 Java 示例演示瞭如何使用重新連接邏輯連接到 Gremlin 客戶端以從意外斷開中恢復。

它具有以下依賴關係：

```
<dependency>
    <groupId>org.apache.tinkerpop</groupId>
    <artifactId>gremlin-driver</artifactId>
    <version>${gremlin.version}</version>
</dependency>

<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>amazon-neptune-sigv4-signer</artifactId>
    <version>${sig4.signer.version}</version>
</dependency>

<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>amazon-neptune-gremlin-java-sigv4</artifactId>
    <version>${sig4.signer.version}</version>
</dependency>

<dependency>
    <groupId>com.evanlennick</groupId>
    <artifactId>retry4j</artifactId>
    <version>0.15.0</version>
</dependency>
```

以下是示例代碼：

```
public static void main(String args[]) {
  boolean useIam = true;

  // Create Gremlin cluster and traversal source
  Cluster.Builder builder = Cluster.build()
         .addContactPoint(System.getenv("neptuneEndpoint"))
         .port(Integer.parseInt(System.getenv("neptunePort")))
         .enableSsl(true)
         .minConnectionPoolSize(1)
         .maxConnectionPoolSize(1)
         .serializer(Serializers.GRAPHBINARY_V1D0)
         .reconnectInterval(2000);

  if (useIam) {
    builder = builder.channelizer(SigV4WebSocketChannelizer.class);
  }

  Cluster cluster = builder.create();

  GraphTraversalSource g = AnonymousTraversalSource
      .traversal()
      .withRemote(DriverRemoteConnection.using(cluster));
```

```
  // Configure retries
  RetryConfig retryConfig = new RetryConfigBuilder()
      .retryOnCustomExceptionLogic(getRetryLogic())
      .withDelayBetweenTries(1000, ChronoUnit.MILLIS)
      .withMaxNumberOfTries(5)
      .withFixedBackoff()
      .build();

  @SuppressWarnings("unchecked")
  CallExecutor<Object> retryExecutor = new CallExecutorBuilder<Object>()
      .config(retryConfig)
      .build();

  // Do lots of queries
  for (int i = 0; i < 100; i++){
    String id = String.valueOf(i);

    @SuppressWarnings("unchecked")
    Callable<Object> query = () -> g.V(id)
        .fold()
        .coalesce(
            unfold(),
            addV("Person").property(T.id, id))
        .id().next();

    // Retry query
    // If there are connection failures, the Java Gremlin client will automatically
    // attempt to reconnect in the background, so all we have to do is wait and retry.
    Status<Object> status = retryExecutor.execute(query);

    System.out.println(status.getResult().toString());
  }

  cluster.close();
}

private static Function<Exception, Boolean> getRetryLogic() {

  return e -> {

    Class<? extends Exception> exceptionClass = e.getClass();

    StringWriter stringWriter = new StringWriter();
    String message = stringWriter.toString();


    if (RemoteConnectionException.class.isAssignableFrom(exceptionClass)){
      System.out.println("Retrying because RemoteConnectionException");
      return true;
    }

    // Check for connection issues
    if (message.contains("Timed out while waiting for an available host") ||
        message.contains("Timed-out waiting for connection on Host") ||
        message.contains("Connection to server is no longer active") ||
        message.contains("Connection reset by peer") ||
        message.contains("SSLEngine closed already") ||
        message.contains("Pool is shutdown") ||
        message.contains("ExtendedClosedChannelException") ||
        message.contains("Broken pipe") ||
        message.contains(System.getenv("neptuneEndpoint")))
    {
      System.out.println("Retrying because connection issue");
      return true;
    };
```

```
    // Concurrent writes can sometimes trigger a ConcurrentModificationException.
    // In these circumstances you may want to backoff and retry.
    if (message.contains("ConcurrentModificationException")) {
      System.out.println("Retrying because ConcurrentModificationException");
      return true;
    }

    // If the primary fails over to a new instance, existing connections to the old primary
 will
    // throw a ReadOnlyViolationException. You may want to back and retry.
    if (message.contains("ReadOnlyViolationException")) {
      System.out.println("Retrying because ReadOnlyViolationException");
      return true;
    }

    System.out.println("Not a retriable error");
    return false;
  };
}
```

使用 Python 連接到海王星資料庫執行個體

如果可以的話，請始終使用引擎版本支持的最新版本的 Apache TinkerPop Python 小鬼客戶端。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。要使用的gremlinpython版TinkerPop本通常會與 Java Gemlin 客戶端的表格中描述的 (p. 305)版本保持一致。

> **Note**
>
> 只要您在編寫的小鬼查詢中僅使用 TinkerPop 3.4.x 功能，3. gremlinpython 5.x 版本與 3.4.x 版本兼容。

下一節將逐步引導您完成連線到 Amazon Neptune 資料庫執行個體並執行 Gremlin 周遊的 Python 範例的執行過程。

您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

開始之前，請執行以下動作：

- 從我們的網站下載並安裝 Python 3.6 或更新版本。
- 確認已安裝 pip。如果您沒有pip或不確定，請參閱我是否需要安裝 pip？ 在文pip檔中。
- 如果 Python 安裝未提供 pip，請如下下載 `futures`：`pip install futures`

若要使用 Python 連線到海王星

1. 輸入以下內容以安裝 `gremlinpython` 套件：

   ```
   pip install --user gremlinpython
   ```

2. 建立名為 `gremlinexample.py` 的檔案，並在文字編輯器中開啟。
3. 將以下內容複製到 `gremlinexample.py` 檔案。將*########## Neptune* 資料庫執行個體的位址。

   如需尋找 Neptune 資料庫執行個體位址的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

   ```
   from __future__  import print_function  # Python 2/3 compatibility
   ```

```
from gremlin_python import statics
from gremlin_python.structure.graph import Graph
from gremlin_python.process.graph_traversal import __
from gremlin_python.process.strategies import *
from gremlin_python.driver.driver_remote_connection import DriverRemoteConnection

graph = Graph()

remoteConn = DriverRemoteConnection('wss://your-neptune-endpoint:8182/gremlin','g')
g = graph.traversal().withRemote(remoteConn)

print(g.V().limit(2).toList())
remoteConn.close()
```

4. 輸入下列命令以執行範例：

```
python gremlinexample.py
```

本範例結尾的 Gremlin 查詢將以清單傳回頂點 (`g.V().limit(2)`)。接著此清單將以標準 Python `print` 函數列印。

> **Note**
>
> Gremlin 查詢最後的部分 `toList()` 用來提交周遊至伺服器，以供進行評估。如果未包含該方法或其他對等方法，則不會將查詢提交至 Neptune 資料庫執行個體。

下列方法會將查詢提交至 Neptune 資料庫執行個體：

- `toList()`
- `toSet()`
- `next()`
- `nextTraverser()`
- `iterate()`

上述範例使用 `g.V().limit(2).toList()` 周遊傳回圖形中的前兩個頂點。若要查詢其他內容，將其換成其他使用其中一個適當之結束方法的 Gremlin 周遊。

使用 .NET 連線至 Neptune 資料庫執行個體

如果可以的話，請始終使用引擎版本支持的最新版本的 ApacheTinkerPop. NET Grim lin 客戶端。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。要使用的`Gremlin.Net`版TinkerPop本通常會與 Java Gemlin 客戶端的表格中描述的 (p. 305)版本保持一致。

下一節包含使用 C# 撰寫的程式碼範例，該範例會連線到 Neptune 資料庫執行個體並執行 Gemlin 周遊。

與 Amazon Neptune 的連線必須來自與海王星資料庫執行個體位於相同虛擬私有雲端 (VPC) 中的 Amazon EC2 執行個體。此範例程式碼已在執行 Ubuntu 的亞馬遜 EC2 執行個體上進行測試。

開始之前，請執行以下動作：

- 在亞馬遜 EC2 執行個體上安裝 .NET。如需在包括 Windows、Linux 和 macOS 等多重作業系統上安裝 .NET 的指示，請參閱開始使用 .NET。
- 請執行套件的 `dotnet add package gremlin.net`以安裝 Gremlin.NET。如需詳細資訊，請參閱說明文件中的 Gemlin.net。TinkerPop

若要使用綠色網路連線到海王星

1. 建立新的 .NET 專案。

```
dotnet new console -o gremlinExample
```

2. 將目錄變更為新的專案目錄。

```
cd gremlinExample
```

3. 將以下內容複製到 Program.cs 檔案。將*########## Neptune* 資料庫執行個體的位址。

   如需尋找 Neptune 資料庫執行個體位址的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

```
using System;
using System.Threading.Tasks;
using System.Collections.Generic;
using Gremlin.Net;
using Gremlin.Net.Driver;
using Gremlin.Net.Driver.Remote;
using Gremlin.Net.Structure;
using static Gremlin.Net.Process.Traversal.AnonymousTraversalSource;
namespace gremlinExample
{
  class Program
  {
    static void Main(string[] args)
    {
      try
      {
        var endpoint = "your-neptune-endpoint";
        // This uses the default Neptune and Gremlin port, 8182
        var gremlinServer = new GremlinServer(endpoint, 8182, enableSsl: true );
        var gremlinClient = new GremlinClient(gremlinServer);
        var remoteConnection = new DriverRemoteConnection(gremlinClient, "g");
        var g = Traversal().WithRemote(remoteConnection);
        g.AddV("Person").Property("Name", "Justin").Iterate();
        g.AddV("Custom Label").Property("name", "Custom id vertex 1").Iterate();
        g.AddV("Custom Label").Property("name", "Custom id vertex 2").Iterate();
        var output = g.V().Limit<Vertex>(3).ToList();
        foreach(var item in output) {
            Console.WriteLine(item);
        }
      }
      catch (Exception e)
      {
          Console.WriteLine("{0}", e);
      }
    }
  }
}
```

4. 輸入下列命令以執行範例：

```
dotnet run
```

本範例結尾的 Gremlin 查詢會傳回單一頂點的數量，以進行測試。接著會列印至主控台。

> **Note**
>
> Gremlin 查詢最後的部分 `Next()` 用來提交周遊至伺服器，以供進行評估。如果未包含該方法或其他對等方法，則不會將查詢提交至 Neptune 資料庫執行個體。

下列方法會將查詢提交至 Neptune 資料庫執行個體：

- `ToList()`
- `ToSet()`
- `Next()`
- `NextTraverser()`
- `Iterate()`

如`Next()`果您需要序列化並傳回查詢結果，或者`Iterate()`不需要，請使用此選項。

前面的例子通過使用`g.V().Limit(3).ToList()`遍歷返回一個列表。若要查詢其他內容，將其換成其他使用其中一個適當之結束方法的 Gremlin 周遊。

使用 Node.js 連線至海王星資料庫執行個體

如果可以的話，請始終使用引擎版本支持的最新版本的 Apache Gri TinkerPop JavaScript mlin 客戶端 gremlin。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。`gremlin`要使用的版 TinkerPop本通常會與 Java Gemlin 客戶端的表格中描述的 (p. 305)版本保持一致。

以下部分將逐步引導您完成 Node.js 範例的執行，該範例連線到 Amazon Neptune 資料庫執行個體並執行 Gramlin 遍歷。

您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

開始之前，請執行以下動作：

- 確認已安裝 Node.js 8.11 或更高版本。如果沒有，請從 Nodejs.org 網站下載並安裝 Node.js。

若要使用 Node.js 連線至海王星

1. 輸入以下內容以安裝 `gremlin-javascript` 套件：

```
npm install gremlin
```

2. 建立名為 `gremlinexample.js` 的檔案，並在文字編輯器中開啟。
3. 將以下內容複製到 `gremlinexample.js` 檔案。將*########## Neptune* 資料庫執行個體的位址。

   如需尋找 Neptune 資料庫執行個體位址的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

```
const gremlin = require('gremlin');
const DriverRemoteConnection = gremlin.driver.DriverRemoteConnection;
const Graph = gremlin.structure.Graph;

dc = new DriverRemoteConnection('wss://your-neptune-endpoint:8182/gremlin',{});

const graph = new Graph();
const g = graph.traversal().withRemote(dc);

g.V().limit(1).count().next().
    then(data => {
        console.log(data);
        dc.close();
    }).catch(error => {
        console.log('ERROR', error);
        dc.close();
```

```
    });
```

4. 輸入下列命令以執行範例：

```
node gremlinexample.js
```

前面的範例回傳使用 g.V().limit(1).count().next()周遊回傳圖表中的單一頂點計數。若要查詢其他內容，將其換成其他使用其中一個適當之結束方法的 Gremlin 周遊。

> **Note**
>
> Gremlin 查詢最後的部分 next() 用來提交周遊至伺服器，以供進行評估。如果未包含該方法或其他對等方法，則不會將查詢提交至 Neptune 資料庫執行個體。

下列方法會將查詢提交至 Neptune 資料庫執行個體：

- toList()
- toSet()
- next()
- nextTraverser()
- iterate()

如next()果您需要序列化並傳回查詢結果，或者iterate()不需要，請使用此選項。

> **Important**
>
> 這是一個獨立的 Node.js 範例。如果您打算在AWS Lambda函數中執行類似這樣的程式碼，請參閱以Lambda 函數範例 (p. 642)取得有關在 Neptune Lambda 函數中JavaScript有效使用的詳細資訊。

使用 Go 連線至 Neptune 資料庫執行個體

如果可以的話，請始終使用最新版本的 Apache TinkerPop Go 小鬼客戶端，gremlingo，您的引擎版本支持。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。

要使用的gremlingo版TinkerPop本通常會與 Java Gemlin 客戶端的表格中描述的 (p. 305)版本保持一致。

> **Note**
>
> 只要您在編寫的小鬼查詢中僅使用 3. TinkerPop 4.x 功能，gremlingo 3.5.x 版本與 3.4.x 版本向後兼容。

以下部分將逐步引導您執行 Go 範例，以連接至 Amazon Neptune 資料庫執行個體並執行 Gemlin 周遊。

您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

開始之前，請執行以下動作：

- 從 go.dev 網站下載並安裝 Go 1.17 或更新版本。

使用 Go 連接到海王星

1. 從空目錄開始，初始化一個新的 Go 模塊：

```
go mod init example.com/gremlinExample
```

2. 添加 gremlin-go 作為新模塊的依賴關係：

```
go get github.com/apache/tinkerpop/gremlin-go/v3/driver
```

3. 建立名為 gremlinExample.go 的檔案，然後在文字編輯器中開啟。
4. 將下列內容複製到gremlinExample.go檔案中，並以 Neptune 資料庫執行個體的位址取代 *(your neptune endpoint)*：

```
package main

import (
  "fmt"
  gremlingo "github.com/apache/tinkerpop/gremlin-go/v3/driver"
)

func main() {
  // Creating the connection to the server.
  driverRemoteConnection, err := gremlingo.NewDriverRemoteConnection("wss://(your
 neptune endpoint):8182/gremlin",
    func(settings *gremlingo.DriverRemoteConnectionSettings) {
      settings.TraversalSource = "g"
    })
  if err != nil {
    fmt.Println(err)
    return
  }
  // Cleanup
  defer driverRemoteConnection.Close()

  // Creating graph traversal
  g := gremlingo.Traversal_().WithRemote(driverRemoteConnection)

  // Perform traversal
  results, err := g.V().Limit(2).ToList()
  if err != nil {
    fmt.Println(err)
    return
  }
  // Print results
  for _, r := range results {
    fmt.Println(r.GetString())
  }
}
```

> **Note**
>
> 使用 macOS 的 Go 1.18+ 目前不支援海王星 TLS 憑證格式，因此在嘗試啟動連線時可能會出現 509 錯誤。對於本地測試，可以通過在導入中添加「crypto/tls」並修改DriverRemoteConnection設置來跳過，如下所示：
>
> ```
> // Creating the connection to the server.
> driverRemoteConnection, err := gremlingo.NewDriverRemoteConnection("wss://your-
> neptune-endpoint:8182/gremlin",
>   func(settings *gremlingo.DriverRemoteConnectionSettings) {
>       settings.TraversalSource = "g"
>       settings.TlsConfig = &tls.Config{InsecureSkipVerify: true}
>   })
> ```

5. 輸入下列命令以執行範例：

```
go run gremlinExample.go
```

在這個例子結束的 Grimlin 查詢返回切片(`g.V().Limit(2)`)的頂點。然後，該切片迭代並使用標準`fmt.Println`功能打印。

> **Note**
>
> Gremlin 查詢最後的部分 `ToList()` 用來提交周遊至伺服器，以供進行評估。如果未包含該方法或其他對等方法，則不會將查詢提交至 Neptune 資料庫執行個體。

下列方法會將查詢提交至 Neptune 資料庫執行個體：

- `ToList()`
- `ToSet()`
- `Next()`
- `GetResultSet()`
- `Iterate()`

上述範例使用 `g.V().Limit(2).ToList()` 周遊傳回圖形中的前兩個頂點。若要查詢其他內容，將其換成其他使用其中一個適當之結束方法的 Gremlin 周遊。

小精靈查詢提示

您可以使用查詢提示，為 Amazon Neptune 中的特定 Grinlin 查詢指定優化和評估策略。

使用以下語法在查詢中新增 `withSideEffect` 步驟，即可指定查詢提示。

```
g.withSideEffect(hint, value)
```

- hint — 識別要套用之提示的類型。
- value — 決定要考慮的系統方面的行為。

例如，以下示範如何在 Gremlin 周遊中包含 `repeatMode` 提示。

> **Note**
>
> 所有 Gremlin 查詢提示的副作用皆加上 `Neptune#` 字首。

```
g.withSideEffect('Neptune#repeatMode',
 'DFS').V("3").repeat(out()).times(10).limit(1).path()
```

前面的查詢指示海王星引擎遍歷圖形深度優先（DFS），而不是默認的海王星廣度優先（BFS）。

以下章節提供可用查詢提示及其用法的詳細資訊。

主題

- 小精靈重複模式查詢提示 (p. 319)
- 小鬼無序查詢提示 (p. 321)
- 小精靈類型促銷查詢提示 (p. 322)
- 小鬼使用 DFE 查詢提示 (p. 322)
- 使用結果快取的小鬼查詢提示 (p. 322)

小精靈重複模式查詢提示

Neptune `repeatMode` 查詢提示會指定 Neptune 引擎如何評估 Grimlin 遍歷中的`repeat()`步驟：廣度優先、深度優先或區塊深度優先。

當 repeat() 步驟的評估模式用於尋找或遵循路徑，而不只是在有限時間內重複單一步驟時，此模式很重要。

語法

將 withSideEffect 步驟加入查詢中，即可指定 repeatMode 查詢提示。

```
g.withSideEffect('Neptune#repeatMode', 'mode').gremlin-traversal
```

> **Note**
>
> 所有 Gremlin 查詢提示的副作用皆加上 Neptune# 字首。

可用模式

- BFS

  廣度優先搜尋

  repeat() 步驟的預設執行模式。會先取得所有同級節點，再進入更深的路徑。

  這個模式會佔用非常大量記憶體，領域可能變得非常龐大。Neptune 引擎會取消查詢記憶體不足的風險較高。這最符合其他 Gremlin 實作。
- DFS

  深度優先搜尋

  依循每個路徑到最大的深度，再移到下一個解決方案。

  這會使用較少的記憶體。在要從多個躍點中開始尋找單一路徑的情況下，這可以提供更好的效能。
- CHUNKED_DFS

  區塊深度優先搜尋

  一種混合的方法，會在 1,000 個節點的區塊中進行深度優先的圖形探索，而不是在1 個節點 (DFS) 或所有節點 (BFS)) 中探索。

  Neptune 引擎將在每個級別上獲得多達 1,000 個節點，然後再沿著路徑更深。

  這是在速度和記憶體使用量之間達到平衡的做法。

  如果您想要使用 BFS，但查詢使用過多記憶體，這很實用。

範例

以下區段說明 Gremlin 周遊重複模式的作用。

在 Neptune 中，repeat()步驟的預設模式是針對所有遍歷執行寬度優先 (BFS) 執行策略。

在大多數情況下，TinkerGraph實現使用相同的執行策略，但在某些情況下，它會改變遍歷的執行。

例如，TinkerGraph實作會修改下列查詢。

```
g.V("3").repeat(out()).times(10).limit(1).path()
```

此周遊的 repeat() 步驟會「展開」為以下周遊，導致深度優先 (DFS) 策略。

```
g.V(<id>).out().out().out().out().out().out().out().out().out().out().limit(1).path()
```

> **Important**
>
> Neptune 查詢引擎不會自動執行此操作。

廣度優先（BFS）是默認的執行策略，TinkerGraph在大多數情況下類似。不過，有些案例則偏好使用深度優先 (DFS) 策略。

BFS (預設值)

廣度優先 (BFS) 是 `repeat()` 運算子的預設執行策略。

```
g.V("3").repeat(out()).times(10).limit(1).path()
```

Neptune 引擎在找到十個跳躍的解決方案之前，全面探索了第一個九跳前沿。在許多情況下，這非常有效，例如短路徑查詢。

不過，在上述範例中，使用 `repeat()` 運算子的深度優先 (DFS) 模式周遊會更快。

DFS

以下查詢的 `repeat()` 運算子會使用深度優先 (DFS) 模式。

```
g.withSideEffect("Neptune#repeatMode", "DFS").V("3").repeat(out()).times(10).limit(1)
```

這會依循每個個別解決方案到最大的深度，再探索下一個解決方案。

小鬼無序查詢提示

當您提交 Grimlin 遍歷時，Neptune 查詢引擎會調查遍歷的結構，並重新排序查詢的各個部分，嘗試將評估和查詢回應時間所需的工作量降到最低。例如，有多個 `has()` 步驟等諸多限制的周遊，通常不會依指定的順序加以評估。它會在以靜態分析檢查查詢後重新排序。

Neptune 查詢引擎會嘗試識別哪個條件約束更具選擇性，然後先執行該條件約束。這通常會導致更好的效能，但 Neptune 選擇評估查詢的順序可能不一定是最佳的。

如果您知道資料的確切特性，並且想要手動指定查詢執行的順序，則可以使用 Neptune `noReordering` 查詢提示來指定按照給定的順序評估遍歷。

語法

將 `withSideEffect` 步驟加入查詢中，即可指定 `noReordering` 查詢提示。

```
g.withSideEffect('Neptune#noReordering', true or false).gremlin-traversal
```

> **Note**
>
> 所有 Gremlin 查詢提示的副作用皆加上 `Neptune#` 字首。

可用值

- `true`
- `false`

小精靈類型促銷查詢提示

當您提交篩選數值或範圍的 Gemlin 遍歷時，Neptune 查詢引擎通常必須在執行查詢時使用類型提升。這意味著它必須檢查可以保存您正在篩選的值的每種類型的值。

例如，如果您篩選的值等於 55，引擎必須尋找等於 55 的整數、等於 55L 的長整數、等於 55.0 的浮點數等於 55.0 等等。每個類型升級都需要對儲存空間進行額外查閱，這可能會導致顯然簡單的查詢需要非預期的長時間才能完成。

假設您正在搜索客戶年齡屬性大於 5 的所有頂點：

```
g.V().has('customerAge', gt(5))
```

若要徹底執行該遍歷，Neptune 必須展開查詢，以檢查您要查詢的值可以提升為的每個數值類型。在這種情況下，過gt濾器必須應用於任何超過 5 的整數，任何長度超過 5L，任何浮動超過 5.0，以及任何超過 5.0 的雙。由於這些類型的促銷活動都需要對儲存空間進行額外的查閱，因此當您小精靈 API profile (p. 357)針對此查詢執行時，您會看到每個數字篩選器的多個篩選器，而且完成所花費的時間會比預期的要長得多。

通常類型促銷是不必要的，因為你事先知道你只需要找到一個特定類型的值。在這種情況下，您可以使用typePromotion查詢提示來關閉類型升級，大幅加快查詢速度。

語法

將 withSideEffect 步驟加入查詢中，即可指定 typePromotion 查詢提示。

```
g.withSideEffect('Neptune#typePromotion', true or false).gremlin-traversal
```

> **Note**
>
> 所有 Gremlin 查詢提示的副作用皆加上 Neptune# 字首。

可用值

- true
- false

若要關閉上述查詢的類型促銷，您可以使用：

```
g.withSideEffect('Neptune#typePromotion', false).V().has('customerAge', gt(5))
```

小鬼使用 DFE 查詢提示

您可以使用此查詢提示來啟用 DFE 執行查詢。根據預設，海王星不會在未將此查詢提示設定為的情況下使用 DFEtrue，因為海王星 _dfe_query_ (p. 533) engine 例證參數預設為。viaQueryHint如果將該執行個體參數設定為enabled，則除了useDFE查詢提示設定為的查詢以false外，所有查詢都會使用 DFE 引擎。

為查詢啟用 DFE 的範例：

```
g.withSideEffect('Neptune#useDFE', true).V().out()
```

使用結果快取的小鬼查詢提示

啟用查詢結果快取時，可以使用下列查詢 (p. 330)提示。

小精靈enableResultCache查詢提示

值為的enableResultCache查詢提示true會導致查詢結果 (如果已經快取) 從快取傳回這些結果。如果沒有，它會返回新的結果並緩存它們，直到它們從緩存中清除。例如：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes')
```

稍後，您可以通過再次發出完全相同的查詢來訪問緩存的結果。

如果此查詢提示的值為false，或者不存在，則不會快取查詢結果。但是，將其設定為false不會清除現有的快取結果。若要清除快取的結果，請使用invalidateResultCache或invalidateResultCachekey提示。

小精靈enableResultCacheWithTTL查詢提示

enableResultCacheWithTTL查詢提示也會傳回快取結果 (如果有的話)，而不會影響快取中已有結果的 TTL。如果目前沒有快取的結果，查詢會傳回新的結果，並快取這些結果以供enableResultCacheWithTTL查詢提示指定的存留時間 (TTL)。該存留時間以秒為單位指定。例如，下列查詢會指定 60 秒的存留時間：

```
g.with('Neptune#enableResultCacheWithTTL', 60)
 .V().has('genre','drama').in('likes')
```

在 60 秒time-to-live結束之前，您可以使用與enableResultCache或查詢提示相同的enableResultCacheWithTTL查詢 (此處g.V().has('genre','drama').in('likes')) 來存取快取的結果。

> **Note**
>
> 使用指定的存留時間enableResultCacheWithTTL不會影響已快取的結果。
>
> - 如果之前使用快取結果enableResultCache，則必須先明確清除快取，然後才能enableResultCacheWithTTL產生新結果，並針對其指定的 TTL 快取它們。
> - 如果之前使用快取結果enableResultCachewithTTL，先前的 TTL 必須先到期，才能enableResultCacheWithTTL產生新的結果，並快取它們以其指定的 TTL。

在存留時間過後，會清除查詢的快取結果，然後相同查詢的後續執行個體會傳回新的結果。如enableResultCacheWithTTL果附加至該後續查詢，則會使用其指定的 TTL 快取新結果。

小精靈invalidateResultCacheKey查詢提示

查invalidateResultCacheKey詢提示可以採取true或false值。true值會導致清除所附加查詢的invalidateResultCacheKey快取結果。例如，下列範例會清除快取查詢索引鍵g.V().has('genre','drama').in('likes')的結果：

```
g.with('Neptune#invalidateResultCacheKey', true)
 .V().has('genre','drama').in('likes')
```

上面的示例查詢不會導致其新的結果被緩存。如果要在清除現有緩存的結果後緩存新結果，則可以在同一查詢中包含enableResultCache (或enableResultCacheWithTTL) ：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#invalidateResultCacheKey', true)
 .V().has('genre','drama').in('likes')
```

小精靈invalidateResultCache查詢提示

查invalidateResultCache詢提示可以採取true或false值。true值會清除結果快取中的所有結果。例如：

```
g.with('Neptune#invalidateResultCache', true)
 .V().has('genre','drama').in('likes')
```

上面的示例查詢不會導致其結果被緩存。如果要在完全清除現有緩存後緩存新結果，則可以在同一個查詢中包含enableResultCache（或enableResultCacheWithTTL）：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#invalidateResultCache', true)
 .V().has('genre','drama').in('likes')
```

小精靈numResultsCached查詢提示

numResultsCached查詢提示只能與包含的查詢搭配使用iterate()，而且它會針對其所附加的查詢指定要快取的結果數目上限。請注意，存在時緩存numResultsCached的結果不會返回，只有緩存。

例如，下列查詢會指定應快取其最多 100 個結果，但不會傳回任何快取的結果：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#numResultsCached', 100)
 .V().has('genre','drama').in('likes').iterate()
```

然後，您可以使用類似下面的查詢來檢索一系列緩存的結果（此處是前十個）：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#numResultsCached', 100)
 .V().has('genre','drama').in('likes').range(0, 10)
```

小精靈noCacheExceptions查詢提示

查noCacheExceptions詢提示可以採取true或false值。true值會導致隱藏與結果快取相關的任何例外狀況。例如：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#noCacheExceptions', true)
 .V().has('genre','drama').in('likes')
```

特別是，這會抑制QueryLimitExceededException，如果查詢的結果太大而無法放入結果快取，則會引發此選項。

小鬼查詢狀態 API

若要取得 Gremlin 查詢的狀態，請使用 HTTP GET 或 POST 向 https://*your-neptune-endpoint*:*port*/gremlin/status 端點提出請求。

小鬼查詢狀態請求參數

- QueryID (選擇性) — 執行中的小鬼查詢的識別碼。只顯示指定查詢的狀態。
- 包含等待中 (選用) — 傳回所有等待中查詢的狀態。

通常，響應中只包含運行中的查詢，但是當指定includeWaiting參數時，也會返回所有等待查詢的狀態。

小鬼查詢狀態響應語法

```
{
  "acceptedQueryCount": integer,
  "runningQueryCount": integer,
  "queries": [
    {
      "queryId":"guid",
      "queryEvalStats":
        {
          "waited": integer,
          "elapsed": integer,
          "cancelled": boolean
        },
      "queryString": "string"
    }
  ]
}
```

小鬼查詢狀態響應值

- acceptedQueryCount— 已接受但尚未完成的查詢數目，包括佇列中的查詢。
- runningQueryCount— 當前正在運行的小鬼查詢的數量。
- 查詢 — 目前的小鬼查詢清單。
- 查詢 ID — 查詢的 GUID 識別碼。Neptune 會自動將此 ID 值指派給每個查詢，或者您也可以指派自己的 ID (請參閱將自定義 ID 注入海王星小鬼或 SPARQL 查詢 (p. 126))。
- queryEvalStats— 此查詢的統計資料。
- 子查詢 — 此查詢中的子查詢數目。
- elapsed — 查詢到目前為止一直在執行的毫秒數。
- 已取消 — True 表示查詢已取消。
- 查詢字串 — 已提交的查詢。如果超過 1024 個字元即予截斷。
- 等候 — 指出查詢等待的時間長度 (以毫秒為單位)。

小鬼查詢狀態示例

下面是使用 `curl` 和 HTTP GET 的狀態命令範例。

```
curl https://your-neptune-endpoint:port/gremlin/status
```

此輸出會顯示單一執行中查詢。

```
{
  "acceptedQueryCount":9,
  "runningQueryCount":1,
  "queries": [
    {
      "queryId":"fb34cd3e-f37c-4d12-9cf2-03bb741bf54f",
      "queryEvalStats":
        {
          "waited": 0,
```

```
          "elapsed": 23,
          "cancelled": false
        },
      "queryString": "g.V().out().count()"
    }
  ]
}
```

小精靈查詢取消

若要取得 Gremlin 查詢的狀態，請使用 HTTP GET 或 POST 向 `https://`*your-neptune-endpoint*`:`*port*`/gremlin/status` 端點提出請求。

小鬼查詢取消請求參數

- 取消查詢 — 取消時必須填寫。此參數沒有對應的值。
- QueryID — 要取消的正在執行的小精靈查詢的識別碼。

小鬼查詢取消示例

以下為取消查詢的 `curl` 命令範例。

```
curl https://your-neptune-endpoint:port/gremlin/status \
  --data-urlencode "cancelQuery" \
  --data-urlencode "queryId=fb34cd3e-f37c-4d12-9cf2-03bb741bf54f"
```

成功取消會傳回 HTTP `200` OK。

支持基於小鬼腳本的會話

您可以在亞馬遜海王星中將 Gemlin 工作階段與隱含交易搭配使用。如需有關 Gemlin 工作階段的資訊，請參閱 Apache TinkerPop 文件中的考量工作階段。下面的章節描述了如何使用小鬼會話與 Java。

> **Note**
>
> 此功能從海王星引擎版本 1.0.1.0. (p. 985) 200463.0 開始提供。
> 從海王星引擎版本 1.1.1.0 (p. 894) 和TinkerPop版本 3.5.2 開始，您也可以使用。小鬼交易 (p. 328)

> **Important**
>
> 目前，Neptune 可以保持以指令碼為基礎的工作階段開啟的最長時間為 10 分鐘。如果不在此時前關閉工作階段，工作階段會逾時，並轉返其中所有內容。

主題

- 小鬼控制台上的小精靈會話 (p. 326)
- 小鬼語言變異中的小鬼會議 (p. 327)

小鬼控制台上的小精靈會話

如果您在 Gremlin 主控台建立遠端連線，而沒有 `session` 參數，則遠端連線會以「無工作階段」模式建立。在此模式中，每個提交至伺服器的請求本身都會視為完整交易，而且請求之間不會儲存狀態。如果請求失敗，只會轉返該請求。

如果您建立的遠端連線確實使用該*session*參數，則會建立以指令碼為基礎的工作階段，該工作階段會持續到您關閉遠端連線為止。每個工作階段都由一個主控台產生並傳回給您的唯一 UUID 識別。

以下是建立工作階段的主控台呼叫範例。提交查詢之後，另一個呼叫會關閉工作階段並遞交查詢。

> **Note**
>
> Gemlin 客戶端必須始終關閉才能釋放服務器端資源。

```
gremlin> :remote connect tinkerpop.server conf/neptune-remote.yaml session
  . . .
  . . .
gremlin> :remote close
```

如需詳細資訊和範例，請參閱TinkerPop文件中的工作階段。

您在工作階段期間執行的所有查詢都會形成單一交易，直到所有查詢成功且關閉遠端連線才會遞交。如果查詢失敗，或者您沒有在 Neptune 支援的最大工作階段存留期內關閉連線，則不會認可工作階段交易，而且其中的所有查詢都會復原。

小鬼語言變異中的小鬼會議

在 Gremlin 語言變體 (GLV) 中，您需要建立 `SessionedClient` 物件，以單一交易發出多個查詢，如下列範例所示。

```
try {                              // line 1
  Cluster cluster = Cluster.open();                    // line 2
  Client client = cluster.connect("sessionName");      // line 3
   ...
   ...
} finally {
  // Always close. If there are no errors, the transaction is committed; otherwise, it's
 rolled back.
  client.close();
}
```

上述範例中的第 3 行會根據針`SessionedClient`對相關叢集設定的組態選項來建立物件。您傳送到連線方法的 *`sessionName`* 字串會成為此工作階段的唯一名稱。若要避免衝突，名稱請使用 UUID。

用戶端會在初始化時啟動工作階段交易。只有在您呼叫 `client.close( )` 時，您在工作階段表單期間執行的所有查詢才會遞交。同樣地，如果單一查詢失敗，或者如果您沒有在 Neptune 支援的工作階段存留期內關閉連線，工作階段交易就會失敗，而且其中的所有查詢都會復原。

> **Note**
>
> Gemlin 客戶端必須始終關閉才能釋放服務器端資源。

```
GraphTraversalSource g = traversal().withRemote(conn);

Transaction tx = g.tx();

// Spawn a GraphTraversalSource from the Transaction.
// Traversals spawned from gtx are executed within a single transaction.
GraphTraversalSource gtx = tx.begin();
try {
  gtx.addV('person').iterate();
  gtx.addV('software').iterate();

  tx.commit();
} finally {
    if (tx.isOpen()) {
```

```
        tx.rollback();
    }
}
```

海王星小鬼交易

有在其中執行 Gemlin 事務 (p. 109)的幾個上下文。使用 Gremlin 時，了解您正在使用的上下文以及其含義非常重要：

- **Script-based**-請求是使用基於文本的 Gremlin 字符串進行的，如下所示：
  - 使用 Java 驅動程式和`Client.submit(string)`.
  - 使用小鬼控制台和. `:remote connect`
  - 使用 HTTP 應用程式介面。
- **Bytecode-based**-請求是使用典型的 Girmlin 語言變體（GLV）的序列化 Girm lin 字節碼進行的。

  例如，使用 Java 驅動程式、`g = traversal().withRemote(...)`.

對於上述任一上下文，存在被作為無會話或綁定到會話發送的請求的附加上下文。

> **Note**
>
> Gremlin 事務必須始終提交或回滾，以便服務器端資源可以被釋放。

無工作階段要求

如果沒有工作階段，請求等同於單一交易。

對於腳本而言，其含義是在單個請求中發送的一個或多個 Gremlin 語句將作為單個事務進行認可或回滾。例如：

```
Cluster cluster = Cluster.open();
Client client = cluster.connect(); // sessionless
// 3 vertex additions in one request/transaction:
client.submit("g.addV();g.addV();g.addV()").all().get();
```

對於字節碼，對於從以下產生和執行的每個遍歷都會發出無會話請求：g

```
GraphTraversalSource g = traversal().withRemote(...);

// 3 vertex additions in three individual requests/transactions:
g.addV().iterate();
g.addV().iterate();
g.addV().iterate();

// 3 vertex additions in one single request/transaction:
g.addV().addV().addV().iterate();
```

繫結至工作階段的要求

當綁定到一個會話，多個請求可以在單個事務的上下文中應用。

對於腳本，這意味著不需要將所有圖形操作連接在一起成單個嵌入字符串值：

```
Cluster cluster = Cluster.open();
Client client = cluster.connect(sessionName); // session
try {
```

```
    // 3 vertex additions in one request/transaction:
    client.submit("g.addV();g.addV();g.addV()").all().get();
} finally {
    client.close();
}

try {
    // 3 vertex additions in three requests, but one transaction:
    client.submit("g.addV()").all().get(); // starts a new transaction with the same
 sessionName
    client.submit("g.addV()").all().get();
    client.submit("g.addV()").all().get();
} finally {
    client.close();
}
```

對於字節碼，之後 TinkerPop3.5.x，可以明確控制事務並透明地管理會話。格林語言變體（GLV）支持小靈的語tx()法commit()或rollback()事務如下：

```
GraphTraversalSource g = traversal().withRemote(conn);

Transaction tx = g.tx();

// Spawn a GraphTraversalSource from the Transaction.
// Traversals spawned from gtx are executed within a single transaction.
GraphTraversalSource gtx = tx.begin();
try {
    gtx.addV('person').iterate();
    gtx.addV('software').iterate();

    tx.commit();
} finally {
    if (tx.isOpen()) {
        tx.rollback();
    }
}
```

雖然上面的例子是用 Java 編寫的，你也可以在 Python，JavaScript 和 .NET 中使用這個tx()語法。

> **Warning**
>
> 無工作階段唯讀查詢會在 SNAPSHOT (p. 109) 隔離下執行，但是在明確交易中執行的唯讀查詢會在 SERIAL (p. 109) IZABLE 隔離下執行。在SERIALIZABLE隔離下執行的唯讀查詢會產生更高的開銷，並且可能會阻止或被並發寫入阻止，這與在SNAPSHOT隔離下運行的那些不同。

在亞馬遜海王星中使用小鬼 API

> **Note**
>
> 亞馬遜海王星不支持該bindings屬性。

Gremlin HTTPS 請求全部使用單一端點：`https://your-neptune-endpoint:port/gremlin`。所有海王星連線都必須使用 HTTPS。

您可以通過直接將 Gimlin 控制台連接到海王星圖形。WebSockets

如需連接到 Gremlin 端點的詳細資訊，請參閱 訪問與小鬼海王星圖 (p. 300)。

Gremlin 的亞馬遜海王星實施具有您需要考慮的特定細節和差異。如需詳細資訊，請參閱亞馬遜海王星的小精靈標準合規 (p. 83)。

如需有關 Gremlin 語言和遍歷的詳細資訊，請參閱 Apache 文件中的穿越。TinkerPop

在亞馬遜海王星小鬼中緩存查詢結果

從引擎版本 1.0.5.1 (p. 924) 開始，亞馬遜海王星支援 Grimlin 查詢的結果快取。

您可以啟用查詢結果快取，然後使用查詢提示來快取 Gremlin 唯讀查詢的結果。

然後，任何重新執行查詢都會擷取快取的結果，而且只要它們仍在快取中，就會以低延遲且無 I/O 成本。這適用於在 HTTP 端點上提交和使用 Websockets 的查詢，無論是作為字節碼或字符串形式。

> Note
>
> 即使啟用查詢快取，傳送至設定檔端點的查詢也不會快取。

您可以透過數種方式控制 Neptune 查詢結果快取的行為。例如：

- 您可以在塊中獲取分頁緩存結果。
- 您可以為指定的查詢指定 time-to-live (TTL)。
- 您可以清除指定查詢的快取。
- 您可以清除整個快取。
- 您可以設定在結果超過快取大小時收到通知。

快取是使用 least-recently-used (LRU) 原則來維護，這表示一旦配置給快取的空間已滿，結果就會在快取新 least-recently-used結果時移除以騰出空間。

> Important
>
> `t3.medium`或`t4.medium`執行個體類型上無法使用查詢結果快取。

在 Neptune 中啟用查詢結果快取

若要在 Neptune 中啟用查詢結果快取，請使用主控台將`neptune_result_cache`資料庫執行個體參數設定為 1 (啟用)。

啟用結果快取之後，Neptune 會設定目前記憶體的一部分來快取查詢結果。您使用的執行個體類型越大，可用的記憶體就越多，Neptune 為快取預留的記憶體就越多。

如果結果快取記憶體已滿，Neptune 會自動捨棄 least-recently-used (LRU) 快取的結果，讓位給新的結果。

您可以使用執行個體狀態 (p. 770)指令檢查結果快取的目前狀態。

使用提示快取查詢結果

啟用查詢結果快取後，您可以使用查詢提示來控制查詢快取。以下所有示例都適用於相同的查詢遍歷，即：

```
g.V().has('genre','drama').in('likes')
```

使用 `enableResultCache`

啟用查詢結果快取後，您可以使用查詢提示快取 Grinlin 查詢的結果，如下`enableResultCache`所示：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes')
```

Neptune 接著會將查詢結果傳回給您，並快取它們。稍後，您可以通過再次發出完全相同的查詢來訪問緩存的結果：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes')
```

標識緩存結果的緩存鍵是查詢字符串本身，即：

```
g.V().has('genre','drama').in('likes')
```

使用 enableResultCacheWithTTL

您可以使用查詢提示來指定應快取enableResultCacheWithTTL查詢結果的時間長度。例如，下列查詢會指定查詢結果應在 120 秒後到期：

```
g.with('Neptune#enableResultCacheWithTTL', 120)
 .V().has('genre','drama').in('likes')
```

同樣地，識別快取結果的快取金鑰是基本查詢字串：

```
g.V().has('genre','drama').in('likes')
```

再次，您可以使用該查詢字符串與查enableResultCache詢提示訪問緩存的結果：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes')
```

如果自緩存結果以來已經過了 120 秒或更多秒，則該查詢將返回新的結果，並緩存它們，而不會有任何結果time-to-live。

您也可以使用查詢提示再次發出相同的查詢來存取快取的結果。enableResultCacheWithTTL例如：

```
g.with('Neptune#enableResultCacheWithTTL', 140)
 .V().has('genre','drama').in('likes')
```

在 120 秒過去之前 (也就是目前有效的 TTL)，這個使用查詢提示的新查enableResultCacheWithTTL詢會傳回快取的結果。120 秒後，它將返回新結果並緩存它們 140 秒。time-to-live

> **Note**
>
> 如果查詢索引鍵的結果已快取，則使用的相同查詢索引鍵enableResultCacheWithTTL不會產生新的結果，也不會影響目前快取time-to-live的結果。
>
> - 如果之前使用快取結果enableResultCache，則必須先清除快取，然後才能enableResultCacheWithTTL產生新結果，並針對其指定的 TTL 快取它們。
> - 如果之前使用快取結果enableResultCachewithTTL，先前的 TTL 必須先到期，才能enableResultCacheWithTTL產生新的結果，並快取它們以其指定的 TTL。

使用 invalidateResultCacheKey

您可以使用invalidateResultCacheKey查詢提示來清除某個特定查詢的快取結果。例如：

```
g.with('Neptune#invalidateResultCacheKey', true)
 .V().has('genre','drama').in('likes')
```

該查詢會清除查詢鍵的快取g.V().has('genre','drama').in('likes')，並傳回該查詢的新結果。

您也可以結invalidateResultCacheKey合enableResultCache或enableResultCacheWithTTL。例如，下列查詢會清除目前快取的結果、快取新結果，然後傳回結果：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#invalidateResultCacheKey', true)
 .V().has('genre','drama').in('likes')
```

使用 invalidateResultCache

您可以使用invalidateResultCache查詢提示來清除查詢結果快取中的所有快取結果。例如：

```
g.with('Neptune#invalidateResultCache', true)
 .V().has('genre','drama').in('likes')
```

該查詢會清除整個結果快取，並傳回查詢的新結果。

您也可以結invalidateResultCache合enableResultCache或enableResultCacheWithTTL。例如，下列查詢會清除整個結果快取、快取此查詢的新結果，然後傳回這些結果：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#invalidateResultCache', true)
 .V().has('genre','drama').in('likes')
```

分頁快取的查詢結果

假設你已經緩存了大量的結果，如下所示：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes')
```

現在假設您發出以下範圍查詢：

```
g.with('Neptune#enableResultCache', true)
 .V().has('genre','drama').in('likes').range(0,10)
```

海王星首先尋找完整的緩存鍵，即g.V().has('genre','drama').in('likes').range(0,10)。如果該鍵不存在，Neptune next 會查看是否有該查詢字符串的鍵沒有範圍（即g.V().has('genre','drama').in('likes')）。當它找到該鍵時，Neptune 然後從其緩存中獲取前十個結果，如範圍指定的那樣。

> **Note**
>
> 如果您將invalidateResultCacheKey查詢提示與結尾有範圍的查詢搭配使用，Neptune 會在找不到與範圍的查詢完全相符時，Neptune 會清除沒有範圍的查詢快取。

搭配使用 numResultsCached 與 .iterate()

您可以使用numResultsCached查詢提示填入結果快取，而不會傳回所有快取的結果，這在您偏好分頁大量結果時很有用。

查numResultsCached詢提示僅適用於結尾為的查詢iterate()。

例如，如果您想要快取範例查詢的前 50 個結果：

```
g.with("Neptune#enableResultCache", true)
 .with("Neptune#numResultsCached", 50)
```

```
 .V().has('genre','drama').in('likes').iterate()
```

在這種情況下，緩存中的查詢鍵是：`g.with("Neptune#numResultsCached", 50).V().has('genre','drama').in('likes')`。您現在可以使用此查詢擷取前十個快取結果：

```
g.with("Neptune#enableResultCache", true)
 .with("Neptune#numResultsCached", 50)
 .V().has('genre','drama').in('likes').range(0, 10)
```

而且，您可以從查詢中檢索接下來的十個結果，如下所示：

```
g.with("Neptune#enableResultCache", true)
 .with("Neptune#numResultsCached", 50)
 .V().has('genre','drama').in('likes').range(10, 20)
```

不要忘記包括提numResultsCached示！它是查詢鍵的重要組成部分，因此必須存在才能訪問緩存的結果。

使用時要記住的一些事情 **numResultsCached**

- 您提供的**numResultsCached**編號會在查詢結束時套用。 例如，這表示下列查詢實際上會快取範圍內的結果`(1000, 1500)`：

  ```
  g.with("Neptune#enableResultCache", true)
   .with("Neptune#numResultsCached", 500)
   .V().range(1000, 2000).iterate()
  ```

- 您提供的數目會**numResultsCached**指定要快取的結果數目上限。 例如，這表示下列查詢實際上會快取範圍內的結果`(1000, 2000)`：

  ```
  g.with("Neptune#enableResultCache", true)
   .with("Neptune#numResultsCached", 100000)
   .V().range(1000, 2000).iterate()
  ```

- 結尾為具**`.range().iterate()`**有自己的範圍的查詢快取的結果。 例如，假設您使用這樣的查詢快取結果：

  ```
  g.with("Neptune#enableResultCache", true)
   .with("Neptune#numResultsCached", 500)
   .V().range(1000, 2000).iterate()
  ```

  要從緩存中檢索前 100 個結果，您可以編寫如下查詢：

  ```
  g.with("Neptune#enableResultCache", true)
   .with("Neptune#numResultsCached", 500)
   .V().range(1000, 2000).range(0, 100)
  ```

  這百個結果將等同於範圍內基本查詢的結果`(1000, 1100)`。

用於查找緩存結果的查詢緩存鍵

快取查詢結果之後，具有相同查詢快取金鑰的後續查詢會從快取擷取結果擷取結果，而不是產生新的結果。查詢的查詢緩存鍵進行評估如下：

1. 查詢提示會被忽略，除了numResultsCached。
2. 最後一個`iterate()`步驟會被忽略。

3. 查詢的其餘部分根據其字節碼表示進行排序。

產生的字串會與快取中已存在的查詢結果索引進行比對，以判斷查詢是否有快取命中。

與結果快取相關的例外狀況

如果您嘗試快取的查詢結果太大而無法容納快取記憶體，即使在移除先前快取的所有項目之後，Neptune 就會引發`QueryLimitExceededException`錯誤。系統不會傳回任何結果，且例外狀況會產生下列錯誤訊息：

```
The result size is larger than the allocated cache,
      please refer to results cache best practices for options to rerun the query.
```

您可以使用`noCacheExceptions`查詢提示來禁止此訊息，如下所示：

```
g.with('Neptune#enableResultCache', true)
 .with('Neptune#noCacheExceptions', true)
 .V().has('genre','drama').in('likes')
```

用小鬼`mergeV()`和步驟製作高效的升頻者 `mergeE()`

upsert（或條件插入）會重複使用頂點或邊緣（如果它已經存在），或者如果不存在，則創建它。高效的升級者可以在 Gremlin 查詢的性能上產生顯著差異。

Upserts 允許您編寫冪等插入操作：無論您運行多少次這樣的操作，總體結果都是一樣的。這在高度並行寫入案例中非常有用，其中對圖形的相同部分進行並行修改可以強制一或多個交易使用 a 回滾`ConcurrentModificationException`，因此需要重試。

例如，下面的查詢 upserts 通過使用提供`Map`的，首先嘗試找到一個頂點與一個頂點。`T.id "v-1"`如果找到該頂點，則返回它。如果沒有找到，則通過`onCreate`子句創建具有該`id`和屬性的頂點。

```
g.mergeV([(id):'v-1').
  option(onCreate, [(label): 'PERSON', 'email': 'person-1@example.org'])
```

批處理提升者以提高吞吐量

對於高輸送量寫入案例，您可以鏈結`mergeV()`並逐`mergeE()`步處理，以批次形式提升頂點和邊緣。批次處理可減少提升大量頂點和邊緣的交易開銷。然後，您可以使用多個用戶端並行提升批次要求，進一步提升輸送量。

根據經驗，我們建議每個批次要求提升大約 200 筆記錄。記錄是單一頂點或邊緣標籤或屬性。例如，具有單一標籤和 4 個屬性的頂點會建立 5 筆記錄。具有標示和單一性質的邊緣會建立 2 筆記錄。如果你想 upsert 批次的頂點，每個頂點都有一個標籤和 4 個屬性，你應該從 40 的批次大小開始，因為。`200 / (1 + 4) = 40`

您可以嘗試批次大小。每批 200 筆記錄是一個很好的起點，但根據您的工作負載，理想的批次大小可能會更高或更低。但是請注意，海王星可能會限制每個請求的 Girmlin 步驟總數。此限制沒有記錄，但為了安全起見，請嘗試確保您的請求包含不超過 1,500 個 Gemlin 步驟。Neptune 可能會拒絕超過 1,500 個步驟的大批次要求。

若要增加輸送量，您可以使用多個用戶端並行提升批次 (請參閱建立有效率的多執行緒 Gremlin 寫入 (p. 828))。用戶端數目應與 Neptune 寫入器執行個體上的工作者執行緒數目相同，通常是伺服器上 vCPU 數目的 2 倍。例如，一個`r5.8xlarge`執行個體具有 32 個 vCPU 和 64 個工作者執行緒。對於使用的高輸送量寫入案例`r5.8xlarge`，您可以使用 64 個用戶端並行將批次更新伺服器寫入 Neptune。

每個客戶端都應提交批量請求並等待請求完成，然後再提交另一個請求。儘管多個客戶端並行運行，但每個單獨的客戶端以串行方式提交請求。這可確保伺服器提供穩定的要求串流，這些要求會佔用所有 Worker 執行緒，而不會淹沒伺服器端要求佇列 (請參閱調整 Neptune 資料庫叢集中的資料庫執行 (p. 118))。

嘗試避免產生多個遍歷器的步驟

當一個 Gemlin 步驟執行時，它需要一個傳入的遍歷器，並發出一個或多個輸出遍歷器。由一個步驟發出的遍歷數量決定了下一個步驟的執行次數。

通常，在執行批次作業時，您希望每個作業 (例如 upsert 頂點 A) 執行一次，以便作業順序看起來像這樣：upsert 頂點 A，接著是 upsert 頂點 B，然後是 upsert 頂點 C，依此類推。只要步驟只建立或修改一個元素，它就只會發出一個遍歷器，而代表下一個作業的步驟只會執行一次。另一方面，如果一個操作創建或修改了多個元素，它會發出多個遍歷，這反過來導致後續步驟在每個發射的遍歷器中執行一次多次。這可能會導致資料庫執行不必要的額外工作，並且在某些情況下可能會導致建立不需要的其他頂點、邊緣或屬性值。

事情可能會出錯的一個例子是類似的查詢`g.V().addV()`。這個簡單的查詢為圖中找到的每個頂點添加一個頂點，因為發`V()`射遍歷圖中的每個頂點和每個這些遍歷器觸發一個調用。`addV()`

如混合升級和插入 (p. 337)需處理可發出多個遍歷器之作業的方法，請參閱。

提升頂點

該`mergeV()`步驟是專門為提升頂點而設計的。它需要作為一個參數 `aMap`，表示圖中現有頂點匹配的元素，如果找不到一個元素，則使用它`Map`來創建一個新的頂點。該步驟還允許您在創建或匹配的情況下改變行為，其中`option()`調製器可以應用`Merge.onCreate`和`Merge.onMatch`令牌來控制這些相應的行為。如需有關如何使用此步驟的進一步資訊，請TinkerPop參閱參考文件。

您可以使用頂點 ID 來判斷特定頂點是否存在。這是首選的方法，因為 Neptune 針對圍繞 ID 的高度並發使用案例優化了 upserts。舉例來說，下列查詢會建立具有指定頂點 ID 的頂點 (如果尚未存在)，或重複使用它 (如果存在)：

```
g.mergeV([(T.id): 'v-1']).
    option(onCreate, [(T.label): 'PERSON', email: 'person-1@example.org', age: 21]).
    option(onMatch, [age: 22]).
  id()
```

請注意，此查詢以一個`id()`步驟結束。雖然對於提升頂點的目的並不是絕對必要的，但 upsert 查詢結束的`id()`步驟可確保伺服器不會將所有頂點屬性序列化回用戶端，這有助於降低查詢的鎖定成本。

或者，您可以使用頂點屬性來識別頂點：

```
g.mergeV([email: 'person-1@example.org']).
    option(onCreate, [(T.label): 'PERSON', age: 21]).
    option(onMatch, [age: 22]).
  id()
```

如果可能的話，請使用您自己的使用者提供的 ID 來建立頂點，並使用這些 ID 來判斷在 upsert 作業期間是否存在頂點。這可以讓海王星優化升級者。當並行修改很常見時，基於 ID 的 upsert 可以比基於屬性的 Upsert 更有效率。

鏈接頂點提升者

您可以將頂點上升端鏈接在一起以將其插入批次中：

```
g.V('v-1')
 .fold()
```

```
 .coalesce(unfold(),
           addV('Person').property(id, 'v-1')
                         .property('email', 'person-1@example.org'))
 .V('v-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-2')
                         .property('email', 'person-2@example.org'))
 .V('v-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-3')
                         .property('email', 'person-3@example.org'))
 .id()
```

或者，您也可以使用以下mergeV()語法：

```
g.mergeV([(T.id): 'v-1', (T.label): 'PERSON', email: 'person-1@example.org']).
  mergeV([(T.id): 'v-2', (T.label): 'PERSON', email: 'person-2@example.org']).
  mergeV([(T.id): 'v-3', (T.label): 'PERSON', email: 'person-3@example.org'])
```

不過，由於這種形式的查詢包含搜尋條件中的元素，這些元素對於基本查詢依據而言是多餘的id，所以效率不如先前的查詢。

上升邊緣

該mergeE()步驟是專門為提升邊緣而設計的。它需要一個Map作為一個參數，表示元素匹配在圖中現有的邊緣，如果沒有找到一個元素，使用它Map來創建一個新的邊緣。該步驟還允許您在創建或匹配的情況下改變行為，其中option()調製器可以應用Merge.onCreate和Merge.onMatch令牌來控制這些相應的行為。如需有關如何使用此步驟的進一步資訊，請TinkerPop參閱參考文件。

您可以使用邊 ID 以與使用自訂頂點 ID 提升頂點相同的方式來增加邊線。同樣，這是首選的方法，因為它允許 Neptune 優化查詢。例如，下列查詢會根據邊緣 ID 建立邊緣 (如果不存在)，或者如果存在，則會重複使用它。該查詢還使用Direction.from和Direction.to頂點的 ID，如果它需要創建一個新的邊緣：

```
g.mergeE([(T.id): 'e-1']).
    option(onCreate, [(from): 'v-1', (to): 'v-2', weight: 1.0]).
    option(onMatch, [weight: 0.5]).
  id()
```

請注意，此查詢以一個id()步驟結束。雖然對於提升邊緣的目的並不是絕對必要的，但是在 upsert 查詢結尾新增id()步驟可確保伺服器不會將所有邊緣屬性序列化回用戶端，這有助於降低查詢的鎖定成本。

許多應用程式使用自訂頂點 ID，但是請讓 Neptune 產生邊緣 ID。如果您不知道邊緣的 ID，但您確實知道from和to頂點 ID，則可以使用這種查詢來提升邊緣：

```
g.mergeE([(from): 'v-1', (to): 'v-2', (T.label): 'KNOWS']).
  id()
```

所有參照的頂點都mergeE()必須存在，才能建立邊的步驟。

鏈接邊緣上升

與頂點 upserts 一樣，將批處理請求的mergeE()步驟鏈接在一起很簡單：

```
g.mergeE([(from): 'v-1', (to): 'v-2', (T.label): 'KNOWS']).
  mergeE([(from): 'v-2', (to): 'v-3', (T.label): 'KNOWS']).
```

```
  mergeE([(from): 'v-3', (to): 'v-4', (T.label): 'KNOWS']).
  id()
```

結合頂點和邊緣上升

有時您可能想要提升頂點和連接它們的邊緣。您可以混合使用此處提供的批次範例。下面的例子提高了 3 個頂點和 2 個邊：

```
g.mergeV([(id):'v-1').
    option(onCreate, [(label): 'PERSON', 'email': 'person-1@example.org']).
  mergeV([(id):'v-2').
    option(onCreate, [(label): 'PERSON', 'email': 'person-2@example.org']).
  mergeV([(id):'v-3').
    option(onCreate, [(label): 'PERSON', 'email': 'person-3@example.org']).
  mergeE([(from): 'v-1', (to): 'v-2', (T.label): 'KNOWS']).
  mergeE([(from): 'v-2', (to): 'v-3', (T.label): 'KNOWS']).
 id()
```

混合升級和插入

有時您可能想要提升頂點和連接它們的邊緣。您可以混合使用此處提供的批次範例。下面的例子提高了 3 個頂點和 2 個邊：

Upserts 通常一次處理一個元素。如果您堅持使用此處顯示的 upsert 模式，則每個 upsert 操作都會發出一個遍歷器，這會導致後續操作僅執行一次。

但是，有時您可能想要將 upsert 與插入物混合在一起。例如，如果您使用 edge 來表示動作或事件的實例，則可能是這種情況。請求可能會使用 upserts 來確保所有必要的頂點都存在，然後使用插入來添加邊緣。對於這種請求，請注意從每個操作發出的潛在遍歷器數量。

請看下面的例子，它混合 upserts 和插入添加表示事件到圖形的邊緣：

```
// Fully optimized, but inserts too many edges
g.mergeV([(id):'v-1').
    option(onCreate, [(label): 'PERSON', 'email': 'person-1@example.org']).
  mergeV([(id):'v-2').
    option(onCreate, [(label): 'PERSON', 'email': 'person-2@example.org']).
  mergeV([(id):'v-3').
    option(onCreate, [(label): 'PERSON', 'email': 'person-3@example.org']).
  mergeV([(T.id): 'c-1', (T.label): 'CITY', name: 'city-1']).
  V('p-1', 'p-2').
  addE('FOLLOWED').to(V('p-1')).
  V('p-1', 'p-2', 'p-3').
  addE('VISITED').to(V('c-1')).
  id()
```

查詢應該插入 5 個邊緣：2 跟隨邊緣和 3 訪問的邊緣。然而，寫入查詢插入 8 個邊緣：2 跟隨和 6 訪問。這樣做的原因是，插入 2 ACKED 邊緣的操作會發出 2 個遍歷，從而導致後續插入操作（插入 3 個邊緣）被執行兩次。

修正方法是在每個可能發出多個遍歷器的作業之後新增一個`fold()`步驟：

```
g.mergeV([(T.id): 'v-1', (T.label): 'PERSON', email: 'person-1@example.org']).
  mergeV([(T.id): 'v-2', (T.label): 'PERSON', email: 'person-2@example.org']).
  mergeV([(T.id): 'v-3', (T.label): 'PERSON', email: 'person-3@example.org']).
  mergeV([(T.id): 'c-1', (T.label): 'CITY', name: 'city-1']).
  V('p-1', 'p-2').
  addE('FOLLOWED').
```

```
    to(V('p-1')).
  fold().
  V('p-1', 'p-2', 'p-3').
  addE('VISITED').
    to(V('c-1')).
  id()
```

在這裡，我們插入插入後的操作後的fold()步驟跟隨邊緣。這會導致單個遍歷器，然後導致後續操作僅執行一次。

這種方法的缺點是查詢現在沒有完全優化，因fold()為沒有優化。接下來的插入操作現在也不fold()會被優化。

如果您需fold()要使用代表後續步驟減少遍歷器的數量，請嘗試對操作進行排序，以便最便宜的操作佔用查詢的非優化部分。

使用製作高效的小鬼提升者 fold()/coalesce()/unfold()

upsert（或條件插入）會重複使用頂點或邊緣（如果它已經存在），或者如果不存在，則創建它。高效的升級者可以在 Gremlin 查詢的性能上產生顯著差異。

本頁顯示如何使用 fold()/coalesce()/unfold() Grimlin 模式來製作高效的提升者。但是，隨著引擎TinkerPop版本 1.2.1.0 (p. 861) 中 Neptune 中引入的 3.6.x 版本，在大多數情況下，新的mergeV()和mergeE()步驟更可取。此處描述的fold()/coalesce()/unfold()模式在某些複雜的情況下仍然有用，但在一般情況下mergeV()，mergeE()如果可以的話，如中所述用小鬼mergeV()和步驟製作高效的升頻者 mergeE() (p. 334)。

Upserts 允許您編寫冪等插入操作：無論您運行多少次這樣的操作，總體結果都是一樣的。這在高度並行寫入案例中非常有用，其中對圖形的相同部分進行並行修改可能會強制一或多個交易使用 a 回滾ConcurrentModificationException，因此需要重試。

例如，下列查詢會先尋找資料集中的指定頂點，然後將結果摺疊到清單中，來提升頂點。在提供給coalesce()步驟的第一個遍歷中，查詢然後展開此列表。如果展開的清單不是空的，則會從中發出結果。coalesce()但是，如果由於頂點目前不存在而unfold()返回空集合，則繼coalesce()續評估提供了該頂點的第二遍歷，並在第二次遍歷查詢中創建缺少的頂點。

```
g.V('v-1').fold()
          .coalesce(
             unfold(),
             addV('Person').property(id, 'v-1')
                           .property('email', 'person-1@example.org')
           )
```

使用優化的形式coalesce()為上升者

Neptune 可以優化fold().coalesce(unfold(), ...)習慣用法以進行高吞吐量更新，但只有在兩個部分都coalesce()返回頂點或邊緣但沒有其他內容時，此優化才有效。如果您嘗試從的任何部分傳回不同的項目 (例如屬性)coalesce()，則不會發生 Neptune 最佳化。查詢可能會成功，但它不會執行以及優化版本，特別是針對大型數據集。

由於未最佳化 upsert 查詢會增加執行時間並減少輸送量，因此值得使用 Gremlin explain 端點來判斷 upsert 查詢是否已完全最佳化。複查explain計劃時，請尋找以和開頭的明細+ not converted into Neptune steps行WARNING: >>。例如：

```
+ not converted into Neptune steps: [FoldStep, CoalesceStep([[UnfoldStep], [AddEdgeSte...
```

```
WARNING: >> FoldStep << is not supported natively yet
```

這些警告可協助您識別查詢中阻止其完全最佳化的部分。

有時無法完全優化查詢。在這些情況下，您應該嘗試在查詢結束時放置無法優化的步驟，從而允許引擎最佳化盡可能多的步驟。在某些批次 upsert 範例中使用此技術，在將任何其他可能未最佳化的修改套用至相同的頂點或邊之前，會先執行針對一組頂點或邊的所有最佳化 upsert。

批處理提升者以提高吞吐量

對於高輸送量寫入案例，您可以將 upsert 步驟鏈結在一起，以批次提升頂點和邊緣。批次處理可減少提升大量頂點和邊緣的交易開銷。然後，您可以使用多個用戶端並行提升批次要求，進一步提升輸送量。

根據經驗，我們建議每個批次要求提升大約 200 筆記錄。記錄是單一頂點或邊緣標籤或屬性。例如，具有單一標籤和 4 個屬性的頂點會建立 5 筆記錄。具有標示和單一性質的邊緣會建立 2 筆記錄。如果你想 upsert 批次的頂點，每個頂點都有一個標籤和 4 個屬性，你應該從 40 的批次大小開始，因為。`200 / (1 + 4) = 40`

您可以嘗試批次大小。每批 200 筆記錄是一個很好的起點，但根據您的工作負載，理想的批次大小可能會更高或更低。但是請注意，海王星可能會限制每個請求的 Girmlin 步驟總數。此限制沒有記錄，但為了安全起見，請嘗試確保您的請求包含不超過 1500 個 Gemlin 步驟。海王星可能會拒絕超過 1500 個步驟的大批次要求。

若要增加輸送量，您可以使用多個用戶端並行提升批次 (請參閱建立有效率的多執行緒 Gremlin 寫入 (p. 828))。用戶端數目應與 Neptune 寫入器執行個體上的工作者執行緒數目相同，通常是伺服器上 vCPU 數目的 2 倍。例如，一個`r5.8xlarge`執行個體具有 32 個 vCPU 和 64 個工作者執行緒。對於使用的高輸送量寫入案例`r5.8xlarge`，您可以使用 64 個用戶端並行將批次更新伺服器寫入 Neptune。

每個客戶端都應提交批量請求並等待請求完成，然後再提交另一個請求。儘管多個客戶端並行運行，但每個單獨的客戶端以串行方式提交請求。這可確保伺服器提供穩定的要求串流，這些要求會佔用所有 Worker 執行緒，而不會淹沒伺服器端要求佇列 (請參閱調整 Neptune 資料庫叢集中的資料庫執行 (p. 118))。

嘗試避免產生多個遍歷器的步驟

當一個 Gemlin 步驟執行時，它需要一個傳入的遍歷器，並發出一個或多個輸出遍歷器。由一個步驟發出的遍歷數量決定了下一個步驟的執行次數。

通常，在執行批次作業時，您希望每個作業 (例如 upsert 頂點 A) 執行一次，以便作業順序看起來像這樣：upsert 頂點 A，接著是 upsert 頂點 B，然後是 upsert 頂點 C，依此類推。只要步驟只建立或修改一個元素，它就只會發出一個遍歷器，而代表下一個作業的步驟只會執行一次。另一方面，如果一個操作創建或修改了多個元素，它會發出多個遍歷，這反過來導致後續步驟在每個發射的遍歷器中執行一次多次。這可能會導致資料庫執行不必要的額外工作，並且在某些情況下可能會導致建立不需要的其他頂點、邊緣或屬性值。

事情可能會出錯的一個例子是類似的查詢`g.V().addV()`。這個簡單的查詢為圖中找到的每個頂點添加一個頂點，因為發`V()`射遍歷圖中的每個頂點和每個這些遍歷器觸發一個調用。`addV()`

如混合升級和插入 (p. 344)需處理可發出多個遍歷器之作業的方法，請參閱。

提升頂點

您可以使用頂點 ID 來確定是否存在對應的頂點。這是首選的方法，因為 Neptune 針對圍繞 ID 的高度並發使用案例優化了 upserts。舉例來說，下列查詢會建立具有指定頂點 ID 的頂點 (如果尚未存在)，或重複使用它 (如果存在)：

```
g.V('v-1')
 .fold()
  .coalesce(unfold(),
```

```
            addV('Person').property(id, 'v-1')
                          .property('email', 'person-1@example.org'))
  .id()
```

請注意，此查詢以一個id()步驟結束。雖然對於提升頂點的目的並不是絕對必要的，但是在 upsert 查詢結尾新增一個id()步驟可確保伺服器不會將所有頂點屬性序列化回用戶端，這有助於降低查詢的鎖定成本。

或者，您可以使用頂點屬性來確定頂點是否存在：

```
g.V()
 .hasLabel('Person')
 .has('email', 'person-1@example.org')
 .fold()
 .coalesce(unfold(),
           addV('Person').property('email', 'person-1@example.org'))
 .id()
```

如果可能的話，請使用您自己的使用者提供的 ID 來建立頂點，並使用這些 ID 來判斷在 upsert 作業期間是否存在頂點。這可以讓海王星優化圍繞 ID 的提升者。在高度並行修改案例中，以 ID 為基礎的 upsert 可以比以屬性為基礎的 Upsert 更有效率。

鏈接頂點提升者

您可以將頂點上升端鏈接在一起以將其插入批次中：

```
g.V('v-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-1')
                         .property('email', 'person-1@example.org'))
 .V('v-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-2')
                         .property('email', 'person-2@example.org'))
 .V('v-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-3')
                         .property('email', 'person-3@example.org'))
 .id()
```

上升邊緣

您可以使用邊 ID 以與使用自訂頂點 ID 提升頂點相同的方式來增加邊線。同樣，這是首選的方法，因為它允許 Neptune 優化查詢。例如，下列查詢會根據邊緣 ID 建立邊緣 (如果不存在)，或者如果存在，則會重複使用它。查詢也會使用from和to頂點的 ID，如果它需要建立新的邊緣。

```
g.E('e-1')
 .fold()
 .coalesce(unfold(),
           addE('KNOWS').from(V('v-1'))
                        .to(V('v-2'))
                        .property(id, 'e-1'))
 .id()
```

許多應用程式使用自訂頂點 ID，但是請讓 Neptune 產生邊緣 ID。如果您不知道邊緣的 ID，但您確實知道from和to頂點 ID，則可以使用此公式來提升邊緣：

```
g.V('v-1')
 .outE('KNOWS')
 .where(inV().hasId('v-2'))
 .fold()
 .coalesce(unfold(),
           addE('KNOWS').from(V('v-1'))
                        .to(V('v-2')))
 .id()
```

請注意，where()子句中的頂點步驟應該是inV()（或者outV()如果您曾經用inE()來查找邊緣），而不是otherV()。不要在這裡使用otherV()，否則查詢將不會被優化，性能將受到影響。例如，Neptune 不會最佳化下列查詢：

```
// Unoptimized upsert, because of otherV()
g.V('v-1')
 .outE('KNOWS')
 .where(otherV().hasId('v-2'))
 .fold()
 .coalesce(unfold(),
           addE('KNOWS').from(V('v-1'))
                        .to(V('v-2')))
 .id()
```

如果您不知道前面的邊緣或頂點 ID，則可以使用頂點屬性來升級：

```
g.V()
 .hasLabel('Person')
 .has('name', 'person-1')
 .outE('LIVES_IN')
 .where(inV().hasLabel('City').has('name', 'city-1'))
 .fold()
 .coalesce(unfold(),
           addE('LIVES_IN').from(V().hasLabel('Person')
                                    .has('name', 'person-1'))
                           .to(V().hasLabel('City')
                                  .has('name', 'city-1')))
 .id()
```

與頂點上升端一樣，最好使用以 ID 為基礎的邊緣上升器，使用邊緣 ID 或fromto頂點 ID，而不是以屬性為基礎的上升端，以便 Neptune 可以完全最佳化 upsert。

檢查from和to頂點是否存在

注意創建一個新邊的步驟的構造：addE().from().to()。這種結構可確保查詢檢查頂點from和to頂點的存在。如果其中一個不存在，則查詢會傳回錯誤，如下所示：

```
{
  "detailedMessage": "Encountered a traverser that does not map to a value for child...
  "code": "IllegalArgumentException",
  "requestId": "..."
}
```

如果可能是頂點from或to頂點不存在，則應在提升它們之間的邊緣之前嘗試提升它們。請參閱 結合頂點和邊緣上升 (p. 343)。

還有一個替代的結構來創建一個你不應該使用的邊緣:V().addE().to(). 如果from頂點存在，它只會增加一個邊。如果to頂點不存在，則查詢會產生錯誤，如前所述，但是如果from頂點不存在，則無訊息地無法插入邊緣，而不會產生任何錯誤。例如，如果from頂點不存在，則下列 upsert 會在不升級邊緣的情況下完成：

```
// Will not insert edge if from vertex does not exist
g.V('v-1')
 .outE('KNOWS')
 .where(inV().hasId('v-2'))
 .fold()
 .coalesce(unfold(),
           V('v-1').addE('KNOWS')
                   .to(V('v-2')))
 .id()
```

鏈接邊緣上升

如果您想要將邊緣 Upsert 鏈結在一起以建立批次要求，則必須使用頂點查詢開始每個 upsert，即使您已經知道邊緣 ID 也是如此。

如果您已經知道要提升的邊緣的 ID 以及`from`和`to`頂點的 ID，則可以使用以下公式：

```
g.V('v-1')
 .outE('KNOWS')
 .hasId('e-1')
 .fold()
 .coalesce(unfold(),
           V('v-1').addE('KNOWS')
                   .to(V('v-2'))
                   .property(id, 'e-1'))
 .V('v-3')
 .outE('KNOWS')
 .hasId('e-2').fold()
 .coalesce(unfold(),
           V('v-3').addE('KNOWS')
                   .to(V('v-4'))
                   .property(id, 'e-2'))
 .V('v-5')
 .outE('KNOWS')
 .hasId('e-3')
 .fold()
 .coalesce(unfold(),
           V('v-5').addE('KNOWS')
                   .to(V('v-6'))
                   .property(id, 'e-3'))
 .id()
```

也許最常見的批處理邊緣 upsert 場景是您知道`from`和`to`頂點 ID，但不知道要更新的邊緣的 ID。在這種情況下，請使用以下公式：

```
g.V('v-1')
 .outE('KNOWS')
 .where(inV().hasId('v-2'))
 .fold()
 .coalesce(unfold(),
           V('v-1').addE('KNOWS')
                   .to(V('v-2')))

 .V('v-3')
 .outE('KNOWS')
 .where(inV().hasId('v-4'))
 .fold()
 .coalesce(unfold(),
           V('v-3').addE('KNOWS')
                   .to(V('v-4')))
 .V('v-5')
 .outE('KNOWS')
```

```
 .where(inV().hasId('v-6'))
 .fold()
 .coalesce(unfold(),
           V('v-5').addE('KNOWS').to(V('v-6')))
 .id()
```

如果你知道你想要 upsert 的邊緣的 ID，但不知道`from`和`to`頂點的 ID（這是不尋常的），你可以使用這個公式：

```
g.V()
 .hasLabel('Person')
 .has('email', 'person-1@example.org')
 .outE('KNOWS')
 .hasId('e-1')
 .fold()
 .coalesce(unfold(),
           V().hasLabel('Person')
              .has('email', 'person-1@example.org')
              .addE('KNOWS')
              .to(V().hasLabel('Person')
                     .has('email', 'person-2@example.org'))
                     .property(id, 'e-1'))
 .V()
 .hasLabel('Person')
 .has('email', 'person-3@example.org')
 .outE('KNOWS')
 .hasId('e-2')
 .fold()
 .coalesce(unfold(),
           V().hasLabel('Person')
              .has('email', 'person-3@example.org')
              .addE('KNOWS')
              .to(V().hasLabel('Person')
                     .has('email', 'person-4@example.org'))
              .property(id, 'e-2'))
 .V()
 .hasLabel('Person')
 .has('email', 'person-5@example.org')
 .outE('KNOWS')
 .hasId('e-1')
 .fold()
 .coalesce(unfold(),
           V().hasLabel('Person')
              .has('email', 'person-5@example.org')
              .addE('KNOWS')
              .to(V().hasLabel('Person')
                     .has('email', 'person-6@example.org'))
                     .property(id, 'e-3'))
 .id()
```

結合頂點和邊緣上升

有時您可能想要提升頂點和連接它們的邊緣。您可以混合使用此處提供的批次範例。下面的例子提高了 3 個頂點和 2 個邊：

```
g.V('p-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-1')
                         .property('email', 'person-1@example.org'))
 .V('p-2')
 .fold()
```

```
 .coalesce(unfold(),
           addV('Person').property(id, 'p-2')
                         .property('name', 'person-2@example.org'))
 .V('c-1')
 .fold()
 .coalesce(unfold(),
           addV('City').property(id, 'c-1')
                       .property('name', 'city-1'))
 .V('p-1')
 .outE('LIVES_IN')
 .where(inV().hasId('c-1'))
 .fold()
 .coalesce(unfold(),
           V('p-1').addE('LIVES_IN')
                   .to(V('c-1')))
 .V('p-2')
 .outE('LIVES_IN')
 .where(inV().hasId('c-1'))
 .fold()
 .coalesce(unfold(),
           V('p-2').addE('LIVES_IN')
                   .to(V('c-1')))
 .id()
```

混合升級和插入

有時您可能想要提升頂點和連接它們的邊緣。您可以混合使用此處提供的批次範例。下面的例子提高了 3 個頂點和 2 個邊：

Upserts 通常一次處理一個元素。如果您堅持使用此處顯示的 upsert 模式，則每個 upsert 操作都會發出一個遍歷器，這會導致後續操作僅執行一次。

但是，有時您可能想要將 upsert 與插入物混合在一起。例如，如果您使用 edge 來表示動作或事件的實例，則可能是這種情況。請求可能會使用 upserts 來確保所有必要的頂點都存在，然後使用插入來添加邊緣。對於這種請求，請注意從每個操作發出的潛在遍歷器數量。

請看下面的例子，它混合 upserts 和插入添加表示事件到圖形的邊緣：

```

// Fully optimized, but inserts too many edges
g.V('p-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-1')
                         .property('email', 'person-1@example.org'))
 .V('p-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-2')
                         .property('name', 'person-2@example.org'))
 .V('p-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-3')
                         .property('name', 'person-3@example.org'))
 .V('c-1')
 .fold()
 .coalesce(unfold(),
           addV('City').property(id, 'c-1')
                       .property('name', 'city-1'))
 .V('p-1', 'p-2')
 .addE('FOLLOWED')
 .to(V('p-1'))
```

```
 .V('p-1', 'p-2', 'p-3')
 .addE('VISITED')
 .to(V('c-1'))
 .id()
```

查詢應該插入 5 個邊緣：2 跟隨邊緣和 3 訪問的邊緣。然而，寫入查詢插入 8 個邊緣：2 跟隨和 6 訪問。原因是插入 2 ACKRED 邊緣的操作會發出 2 個遍歷，從而導致後續插入操作（插入 3 個邊緣）被執行兩次。

修正方法是在每個可能發出多個遍歷器的作業之後新增一個`fold()`步驟：

```
g.V('p-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-1')
                         .property('email', 'person-1@example.org'))
 .V('p-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-2').
                         .property('name', 'person-2@example.org'))
 .V('p-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'p-3').
                         .property('name', 'person-3@example.org'))
 .V('c-1')
 .fold().
 .coalesce(unfold(),
            addV('City').property(id, 'c-1').
                        .property('name', 'city-1'))
 .V('p-1', 'p-2')
 .addE('FOLLOWED')
 .to(V('p-1'))
 .fold()
 .V('p-1', 'p-2', 'p-3')
 .addE('VISITED')
 .to(V('c-1')).
 .id()
```

在這裡，我們插入插入後的操作後的`fold()`步驟跟隨邊緣。這會導致單個遍歷器，然後導致後續操作僅執行一次。

這種方法的缺點是查詢現在沒有完全優化，因`fold()`為沒有優化。接下來的插入操作現在不`fold()`會被優化。

如果您需`fold()`要使用代表後續步驟減少遍歷器的數量，請嘗試對操作進行排序，以便最便宜的操作佔用查詢的非優化部分。

修改現有頂點和邊的上升

有時候，如果頂點或邊緣不存在，您想要創建一個頂點或邊緣，然後向其添加或更新屬性，而不管它是新的還是現有的頂點或邊緣。

若要新增或修改屬性，請使用`property()`步驟。在步驟之外使用此`coalesce()`步驟。如果您嘗試修改`coalesce()`步驟內現有頂點或邊緣的屬性，Neptune 查詢引擎可能不會最佳化查詢。

下列查詢會在每個已上傳的頂點上新增或更新計數器屬性。每個`property()`步驟都有單一基數，以確保新值會取代任何現有值，而不是新增至一組現有值。

```
g.V('v-1')
```

```
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-1')
                         .property('email', 'person-1@example.org'))
 .property(single, 'counter', 1)
 .V('v-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-2')
                         .property('email', 'person-2@example.org'))
 .property(single, 'counter', 2)
 .V('v-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-3')
                         .property('email', 'person-3@example.org'))
 .property(single, 'counter', 3)
 .id()
```

如果您有適用於所有 Upserted 元素的屬性值 (例如`lastUpdated`時間戳記值)，您可以在查詢結束時新增或更新它：

```
g.V('v-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-1')
                         .property('email', 'person-1@example.org'))
 .V('v-2').
 .fold().
 .coalesce(unfold(),
           addV('Person').property(id, 'v-2')
                         .property('email', 'person-2@example.org'))
 .V('v-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-3')
                         .property('email', 'person-3@example.org'))
 .V('v-1', 'v-2', 'v-3')
 .property(single, 'lastUpdated', datetime('2020-02-08'))
 .id()
```

如果有其他條件決定是否應進一步修改頂點或邊，則可以使用`has()`步驟篩選將套用修改的元素。下列範例會使用一個`has()`步驟，根據其屬性的值篩選已上傳的`version`頂點。然後，查詢將更新為小於 3 `version`的任何頂點的 3：`version`

```
g.V('v-1')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-1')
                         .property('email', 'person-1@example.org')
                         .property('version', 3))
 .V('v-2')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-2')
                         .property('email', 'person-2@example.org')
                         .property('version', 3))
 .V('v-3')
 .fold()
 .coalesce(unfold(),
           addV('Person').property(id, 'v-3')
```

```
                        .property('email', 'person-3@example.org')
                        .property('version', 3))
.V('v-1', 'v-2', 'v-3')
.has('version', lt(3))
.property(single, 'version', 3)
.id()
```

使用小靈分析海王星查詢執行 explain

亞馬遜海王星已經添加了一個名為解釋的小鬼功能。此功能是一種自助服務工具，用於了解 Neptune 引擎採取的執行方法。您透過將 explain 參數新增到提交 Gremlin 查詢的 HTTP 呼叫，來進行叫用。

此 explain 功能可提供查詢執行計劃邏輯結構的相關資訊。您可以使用此資訊來識別潛在的評估和執行瓶頸，並調整查詢，如中所述。調整小精靈查詢 (p. 364)您也可以使用查詢提示 (p. 319)來改善查詢執行計畫。

> Note
>
> 此功能可從開始使用版本 1.0.0.0.463.0 (p. 985)。

主題

- 了解 Gemlin 查詢如何在海王星中工作 (p. 347)
- 在海王星中使用小鬼 explain API (p. 351)
- 海王星中的小鬼 profile API (p. 357)
- 使用和調整小鬼查詢 explainprofile (p. 364)
- 亞馬遜海王星的原生小精靈步進支持 (p. 377)

了解 Gemlin 查詢如何在海王星中工作

要充分利用 Gremlin explain 和亞馬遜海王星中的profile報告，了解有關 Gremlin 查詢的一些背景信息很有幫助。

主題

- 海王星中的小鬼语句 (p. 347)
- 海王星如何使用語句索引處理 Girmlin 查詢 (p. 349)
- 如何在海王星中處理小鬼查詢 (p. 350)

海王星中的小鬼语句

Amazon Neptune 中的屬性圖表資料是由四個位置 (四) 陳述式所組成。其中每個陳述式都代表屬性圖表資料的個別原子單位。如需詳細資訊，請參閱海王星圖數據模型 (p. 104)。與資源描述架構 (RDF) 資料模型類似，這四個位置如下：

- subject (S)
- predicate (P)
- object (O)
- graph (G)

每個陳述式都是有關一或多個資源的宣告。例如，陳述式可以宣告兩個資源之間的關係，或者將屬性 (鍵/值組) 連接到一些資源。

您可以將述詞視為陳述式的動詞，描述關係或屬性的類型。物件是關係的目標，或是屬性的值。圖形位置是選用的，可用於多種不同的方法。對於海王星屬性圖（PG）數據，它可以是未使用的（空圖），或者用於表示邊緣的標識符。具有共用資源識別符的一組陳述式可建立圖形。

海王星屬性圖數據模型中有三類語句：

主題

- Gremlin 頂點標籤陳述式 (p. 348)
- Gremlin 邊緣陳述式 (p. 348)
- Gremlin 屬性陳述式 (p. 348)

Gremlin 頂點標籤陳述式

Neptune 中的頂點標籤語句有兩個目的：

- 它們會追蹤頂點的標籤。
- 其中至少有一個陳述式暗示圖形中存在特定頂點。

這些陳述式的主題是頂點識別符，而物件是標籤，兩者都由使用者指定。您可以針對這些陳述式使用特殊的固定述詞 (顯示為 <~label>)，以及預設圖形識別符 (Null 圖形) (顯示為 <~>)。

例如，請考量下列 addV 周遊。

```
g.addV("Person").property(id, "v1")
```

此周遊會導致以下陳述式新增至圖形。

```
StatementEvent[Added(<v1> <~label> <Person> <~>) .]
```

Gremlin 邊緣陳述式

Grimlin 邊緣語句是什麼意味著在海王星圖中的兩個頂點之間的邊緣的存在。邊緣陳述式的主旨 (S) 是來源 from 頂點。述詞 (P) 是使用者提供的邊緣標籤。物件 (O) 是目標 to 頂點。圖形 (G) 是使用者提供的邊緣識別符。

例如，請考量下列 addE 周遊。

```
g.addE("knows").from(V("v1")).to(V("v2")).property(id, "e1")
```

此周遊會導致以下陳述式新增至圖形。

```
StatementEvent[Added(<v1> <knows> <v2> <e1>) .]
```

Gremlin 屬性陳述式

在海王星 Grimlin 屬性語句聲明的頂點或邊緣的單個屬性值。主旨是使用者提供的頂點或邊緣識別符。述詞是屬性名稱 (索引鍵)，而物件是個別屬性值。圖形 (G) 再次是預設圖形識別符 (Null 圖形)，顯示為 <~>。

請考量下列範例。

```
g.V("v1").property("name", "John")
```

此陳述式會產生下列結果。

```
StatementEvent[Added(<v1> <name> "John" <~>) .]
```

屬性陳述式與其他屬性不同，其物件是基本值 (string、date、byte、short、int、long、float 或 double)。其物件不是可做為另一個宣告之主旨的資源識別符。

對於多重屬性，集合中的每個個別屬性值都會收到自己的陳述式。

```
g.V("v1").property(set, "phone", "956-424-2563").property(set, "phone", "956-354-3692
 (tel:9563543692)")
```

成果如下所示。

```
StatementEvent[Added(<v1> <phone> "956-424-2563" <~>) .]
StatementEvent[Added(<v1> <phone> "956-354-3692" <~>) .]
```

海王星如何使用語句索引處理 Girmlin 查詢

陳述式在 Amazon Neptune 中透過三個陳述式索引的方式存取，如中所述如何在海王星中索引陳述式 (p. 104)。Neptune 從 Gremlin 查詢中提取語句模式，其中一些位置已知，其餘的則留給索引搜索進行發現。

Neptune 假設屬性圖結構描述的大小不大。這表示不同邊緣標示和性質名稱的數目相當低，導致不同述詞的總數較少。Neptune 會在單獨的索引中追蹤不同的述詞。它會使用此述詞快取來執行的 `{ all P x POGS }` 聯合掃描，而不是使用 OSGP 索引。避免需要反向周遊 OSGP 索引，可同時節省儲存空間和載入輸送量。

海王星 Grimlin 解釋/配置文件 API 使您可以在圖形中獲取謂詞計數。然後，您可以判斷您的應用程式是否使 Neptune 假設屬性圖結構描述很小的無效。

下列範例有助於說明海王星如何使用索引來處理 Gremlin 查詢。

問：什麼是頂點 **v1** 的標籤？

```
  Gremlin code:      g.V('v1').label()
  Pattern:           (<v1>, <~label>, ?, ?)
  Known positions:   SP
  Lookup positions:  OG
  Index:             SPOG
  Key range:         <v1>:<~label>:*
```

問：什麼是頂點 **v1** 的「已知」外邊緣？

```
  Gremlin code:      g.V('v1').out('knows')
  Pattern:           (<v1>, <knows>, ?, ?)
  Known positions:   SP
  Lookup positions:  OG
  Index:             SPOG
  Key range:         <v1>:<knows>:*
```

問：哪些頂點有 **Person** 頂點標籤？

```
  Gremlin code:      g.V().hasLabel('Person')
```

```
Pattern:           (?, <~label>, <Person>, <~>)
Known positions:   POG
Lookup positions:  S
Index:             POGS
Key range:         <~label>:<Person>:<~>:*
```

問：什麼是指定邊緣 **e1** 的來源/目標頂點？

```
Gremlin code:      g.E('e1').bothV()
Pattern:           (?, ?, ?, <e1>)
Known positions:   G
Lookup positions:  SPO
Index:             GPSO
Key range:         <e1>:*
```

海王星沒有的一個陳述式索引是反向遍歷 OSGP 索引。此索引可用來收集所有邊緣標籤的所有傳入邊緣，如下列範例所示。

問：什麼是傳入的相鄰頂點**v1**？

```
Gremlin code:      g.V('v1').in()
Pattern:           (?, ?, <v1>, ?)
Known positions:   O
Lookup positions:  SPG
Index:             OSGP  // <-- Index does not exist
```

如何在海王星中處理小鬼查詢

在 Amazon Neptune 中，更複雜的遍歷可以由一系列模式來表示，這些模式會根據具名變數的定義建立關係，這些模式可以跨模式共用以建立聯結。如以下範例所示。

問：什麼是頂點 **v1** 的雙躍點鄰近項？

```
Gremlin code:      g.V(‘v1’).out('knows').out('knows').path()
Pattern:           (?1=<v1>, <knows>, ?2, ?) X Pattern(?2, <knows>, ?3, ?)

The pattern produces a three-column relation (?1, ?2, ?3) like this:
                   ?1     ?2     ?3
                   ================
                   v1     v2     v3
                   v1     v2     v4
                   v1     v5     v6
```

透過在兩個模式之間共用 ?2 變數 (在第一個模式中的 O 位置和第二個模式的 S 位置)，您可以建立從第一個躍點鄰近項到第二個躍點鄰近項的聯結。每個 Neptune 解決方案都有三個命名變量的綁定，這些變量可用於重新創建TinkerPop遍歷器（包括路徑信息）。

在 Gramlin 查詢處理的第一步是將查詢解析為一個TinkerPop遍歷對象，由一系列步驟組成。TinkerPop這些步驟屬於開放原始碼 Apache TinkerPop 專案的一部分，是在參考實作中組成 Gemlin 遍歷的邏輯和實體運算子。它們都是用來代表查詢的模型。它們是可執行運算子，可根據所代表運算子的語意產生解決方案。例如，`.V()`由表示和執行TinkerPopGraphStep。

因為這些off-the-shelfTinkerPop步驟是可執行的，所以這樣的TinkerPop遍歷可以執行任何 Gemlin 查詢並產生正確的答案。但是，當對大圖執行時，TinkerPop步驟有時可能非常低效和緩慢。Neptune 嘗試將遍歷轉換為由模式群組組成的宣告式形式，如前所述，而不是使用它們。

海王星目前不支援其原生查詢引擎中的所有 Gemlin 運算子 (步驟)。因此，它會嘗試盡可能將多個步驟摺疊成單一 NeptuneGraphQueryStep，其中包含所有已轉換步驟的宣告式邏輯查詢計畫。理想情況下，所有步驟都會進行轉換。但是，當遇到無法轉換的步驟時，Neptune 會中斷原生執行，並將所有查詢執行從該點推遲到步TinkerPop驟。它不會嘗試將原生執行編入和編出。

將這些步驟轉譯成邏輯查詢計畫之後，Neptune 會執行一系列查詢最佳化工具，根據靜態分析和預估基數來重新撰寫查詢計畫。這些最佳化工具會執行如下動作：根據範圍計數重新排序運算子、刪除不必要或冗餘運算子、重新安排篩選條件、將運算子推送至不同的群組等等。

產生最佳化的查詢計畫之後，Neptune 會建立執行查詢工作的實體運算子管線。這包括從陳述式索引讀取資料、執行各種類型的聯結、篩選、排序等等。管線會產生解決方案資料流，然後再轉換回 TinkerPop Traverser 物件串流。

查詢結果的序列化

Amazon Neptune 目前依賴TinkerPop回應訊息序列化器將查詢結果 (TinkerPop遍歷器) 轉換為序列化資料，以透過線路傳送回用戶端。這些序列化格式往往相當冗長。

例如，若要序列化頂點查詢 (例如) 的結果`g.V().limit(1)`，Neptune 查詢引擎必須執行單一搜尋才能產生查詢結果。不過，GraphSON 序列化程式會執行大量的額外搜尋，將頂點封裝為序列化格式。它將必須執行一次搜尋以取得標籤、執行一次搜尋以取得屬性金鑰，以及針對頂點的每個屬性金鑰執行一次搜尋，以取得每個金鑰的所有值。

某些序列化格式更有效率，但都需要額外的搜尋。此外，TinkerPop序列化器不會嘗試避免重複的搜索，通常導致許多搜索不必要地重複。

這使得編寫查詢非常重要，以便它們只特別要求其所需的資訊。例如，`g.V().limit(1).id()` 只會傳回頂點 ID 並消除所有其他序列化程式搜尋。海王星中的小鬼 profile API (p. 357) 可讓您查看在查詢執行期間和序列化期間進行多少次搜尋呼叫。

在海王星中使用小鬼 explain API

亞馬遜海王星 Grimlin explain API 會傳回在執行指定查詢時將執行的查詢計畫。因為 API 並未實際執行查詢，所以幾乎會立即傳回計劃。

它不同於 TinkerPop .explain () 步驟，以便能夠報告特定於海王星引擎的資訊。

小靈報告中包含的信息 explain

explain報告包含下列資訊：

- 如請求的查詢字串。
- 原始遍歷。這是通過將查詢字符串解析成步驟生成TinkerPop的TinkerPop遍歷對象。它等同於針對查詢執行所產生.explain()的原始查詢TinkerPopTinkerGraph。
- 轉換後的遍歷。這是通過將遍歷轉換為海王星邏輯查詢計劃表示產生的海王星TinkerPop遍歷。在許多情況下，整個TinkerPop遍歷轉換為兩個 Neptune 步驟：一個執行整個查詢 (NeptuneGraphQueryStep) 和一個將 Neptune 查詢引擎輸出轉換回 TinkerPop Traversers ()。NeptuneTraverserConverterStep
- 優化的遍歷。這是 Neptune 查詢計畫透過一系列靜態工作減少最佳化器執行之後的最佳化版本，這些最佳化工作會根據靜態分析和估計基數來重新撰寫查詢。這些最佳化工具會執行如下動作：根據範圍計數重新排序運算子、刪除不必要或冗餘運算子、重新安排篩選條件、將運算子推送至不同的群組等等。
- 謂詞計數。由於先前所述的 Neptune 索引策略，因此具有大量不同的述詞可能會導致效能問題。這對於沒有邊緣標籤 (.in 或 .both) 的反向周遊運算子的查詢尤其適用。如果使用這類運算子且述詞計數夠高，則 explain 報告會顯示警告訊息。
- DFE 資訊。啟用 DFE 替代引擎時，下列遍歷元件可能會顯示在最佳化的遍歷中：
  - **DFEStep**— 海王星在包含子項的遍歷中最佳化 DFE 步驟。DFENode DFEStep代表在 DFE 引擎中執行的查詢計畫的一部分。

- **DFENode**— 包含作為一個或多個子項的中間表示DFEJoinGroupNodes。
- **DFEJoinGroupNode**— 代表一個或多個DFENode或DFEJoinGroupNode元素的連接。
- **NeptuneInterleavingStep**— 海王星在包含子項的遍歷中最佳化 DFE 步驟。DFEStep

  也包含包含遍歷相關資訊的stepInfo元素，例如邊界元素、使用的路徑元素等。此資訊用於處理子項DFEStep。

查明 DFE 是否正在評估查詢的一DFEStep個簡單方法是檢查explain輸出是否包含. 不屬於周遊的任何部分DFEStep將不會由 DFE 執行，並且將由引擎執行。TinkerPop

啟用 DFE 的範例 (p. 355)如需範例報告，請參閱。

小鬼語法 explain

explain API 的語法與用於查詢的 HTTP API 語法相同，差別在於它使用 /gremlin/explain 做為端點，而非 /gremlin，如下列範例所示。

```
curl -X POST https://your-neptune-endpoint:port/gremlin/explain -d
 '{"gremlin":"g.V().limit(1)"}'
```

上述查詢會產生下列輸出。

```
*******************************************************
                Neptune Gremlin Explain
*******************************************************

Query String
============
g.V().limit(1)

Original Traversal
==================
[GraphStep(vertex,[]), RangeGlobalStep(0,1)]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .]
        }, finishers=[limit(1)], annotations={path=[Vertex(?1):GraphStep], maxVarId=3}
    },
    NeptuneTraverserConverterStep
]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .],
 {estimatedCardinality=INFINITY}
        }, finishers=[limit(1)], annotations={path=[Vertex(?1):GraphStep], maxVarId=3}
    },
    NeptuneTraverserConverterStep
]

Predicates
```

```
==========
# of predicates: 18
```

未轉換的TinkerPop步驟

理想情況下，遍歷中的所有TinkerPop步驟都具有原生 Neptune 操作員覆蓋範圍。如果不是這種情況，Neptune 會退回其操作員覆蓋範圍中的差距的TinkerPop步驟執行。如果遍歷使用 Neptune 尚未具有原生涵蓋範圍的步驟，則explain報告會顯示警告，指出發生間隙的位置。

遇到沒有對應原生 Neptune 運算子的步驟時，即使後續步驟確實具有原生 Neptune 運算子，也會使用步TinkerPop驟執行從該點開始前進的整個遍歷。

除此之外是叫用 Neptune 全文檢索搜尋時。NeptuneSearchStep實作不含原生對等項目的步驟做為全文檢索搜尋步驟。

查詢中所有步驟都具有原生對等項目的explain輸出範例

以下是查詢的範例explain報表，其中所有步驟都具有原生對等項目：

```
*******************************************************
                Neptune Gremlin Explain
*******************************************************

Query String
============
g.V().out()

Original Traversal
==================
[GraphStep(vertex,[]), VertexStep(OUT,vertex)]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .]
            PatternNode[(?1, ?5, ?3, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .]
            PatternNode[(?3, <~label>, ?4, <~>) . project ask .]
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep], maxVarId=7}
    },
    NeptuneTraverserConverterStep
]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, ?5, ?3, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .],
 {estimatedCardinality=INFINITY}
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep], maxVarId=7}
    },
    NeptuneTraverserConverterStep
]

Predicates
==========
# of predicates: 18
```

查詢中的部分步驟沒有原生對等項目的範例

VertexStepNeptune 同時處理GraphStep和本地處理，但是如果引入FoldStep和UnfoldStep，則結果explain輸出不同：

```
***************************************************
                Neptune Gremlin Explain
***************************************************

Query String
============
g.V().fold().unfold().out()

Original Traversal
==================
[GraphStep(vertex,[]), FoldStep, UnfoldStep, VertexStep(OUT,vertex)]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .]
        }, annotations={path=[Vertex(?1):GraphStep], maxVarId=3}
    },
    NeptuneTraverserConverterStep
]
+ not converted into Neptune steps: [FoldStep, UnfoldStep, VertexStep(OUT,vertex)]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .],
 {estimatedCardinality=INFINITY}
        }, annotations={path=[Vertex(?1):GraphStep], maxVarId=3}
    },
    NeptuneTraverserConverterStep,
    NeptuneMemoryTrackerStep
]
+ not converted into Neptune steps: [FoldStep, UnfoldStep, VertexStep(OUT,vertex)]

WARNING: >> FoldStep << is not supported natively yet
```

在此情況下，FoldStep 會使您從原生執行中脫離。但即使由於後續 VertexStep 似乎在 Fold/Unfold 步驟的下游，而不再原生處理它也一樣。

為了節省效能和成本，請務必嘗試規劃周遊，以便在 Neptune 查詢引擎內部以原生方式完成可能的最大工作量，而不是由步驟實作。TinkerPop

使用海王星的查詢範例 full-text-search

下列查詢使用海王星全文檢索搜尋：

```
g.withSideEffect("Neptune#fts.endpoint", "some_endpoint")
  .V()
  .tail(100)
  .has("Neptune#fts mark*")
  -------
```

```
  .has("name", "Neptune#fts mark*")
  .has("Person", "name", "Neptune#fts mark*")
```

.has("name", "Neptune#fts mark*") 部分將搜尋限制為具有 name 的頂點，而 .has("Person", "name", "Neptune#fts mark*") 將搜尋限制為具有 name 和標籤 Person 的頂點。這會導致報告中的下列遍歷：explain

```
Final Traversal
[NeptuneGraphQueryStep(Vertex) {
    JoinGroupNode {
        PatternNode[(?1, termid(1,URI), ?2, termid(0,URI)) . project distinct ?1 .],
 {estimatedCardinality=INFINITY}
    }, annotations={path=[Vertex(?1):GraphStep], maxVarId=4}
}, NeptuneTraverserConverterStep, NeptuneTailGlobalStep(10),
 NeptuneTinkerpopTraverserConverterStep, NeptuneSearchStep {
    JoinGroupNode {
        SearchNode[(idVar=?3, query=mark*, field=name) . project ask .],
 {endpoint=some_endpoint}
    }
    JoinGroupNode {
        SearchNode[(idVar=?3, query=mark*, field=name) . project ask .],
 {endpoint=some_endpoint}
    }
}]
```

DFE explain 啟用時的使用範例

以下是啟用 DFE 替代查詢引擎時的explain報告範例：

```
********************************************************
                Neptune Gremlin Explain
********************************************************

Query String
============

g.V().as("a").out().has("name", "josh").out().in().where(eq("a"))


Original Traversal
==================
[GraphStep(vertex,[])@[a], VertexStep(OUT,vertex), HasStep([name.eq(josh)]),
 VertexStep(OUT,vertex), VertexStep(IN,vertex), WherePredicateStep(eq(a))]

Converted Traversal
===================
Neptune steps:
[
    DFEStep(Vertex) {
      DFENode {
        DFEJoinGroupNode[ children={
          DFEPatternNode[(?1, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, ?2,
 <http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph>) . project DISTINCT[?1]
 {rangeCountEstimate=unknown}],
          DFEPatternNode[(?1, ?3, ?4, ?5) . project ALL[?1, ?4] graphFilters=(!= <http://
aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph> . ), {rangeCountEstimate=unknown}]
        }, {rangeCountEstimate=unknown}
        ]
      } [Vertex(?1):GraphStep@[a], Vertex(?4):VertexStep]
    } ,
    NeptuneTraverserConverterDFEStep
]
+ not converted into Neptune steps: HasStep([name.eq(josh)]),
```

```
Neptune steps:
[
    NeptuneInterleavingStep {
      StepInfo[joinVars=[?7, ?1], frontierElement=Vertex(?7):HasStep,
 pathElements={a=(last,Vertex(?1):GraphStep@[a])}, listPathElement={}, indexTime=0ms],
      DFEStep(Vertex) {
        DFENode {
          DFEJoinGroupNode[ children={
            DFEPatternNode[(?7, ?8, ?9, ?10) . project ALL[?7, ?9]
 graphFilters=(!= <http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph> . ),
 {rangeCountEstimate=unknown}],
            DFEPatternNode[(?12, ?11, ?9, ?13) . project ALL[?9, ?12]
 graphFilters=(!= <http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph> . ),
 {rangeCountEstimate=unknown}]
          }, {rangeCountEstimate=unknown}
          ]
        } [Vertex(?9):VertexStep, Vertex(?12):VertexStep]
      }
    }
]
+ not converted into Neptune steps: WherePredicateStep(eq(a)),
Neptune steps:
[
    DFECleanupStep
]


Optimized Traversal
===================
Neptune steps:
[
    DFEStep(Vertex) {
      DFENode {
        DFEJoinGroupNode[ children={
          DFEPatternNode[(?1, ?3, ?4, ?5) . project ALL[?1, ?4] graphFilters=(!=
 defaultGraph[526] . ), {rangeCountEstimate=92233720368547758O7}]
        }, {rangeCountEstimate=unknown}
        ]
      } [Vertex(?1):GraphStep@[a], Vertex(?4):VertexStep]
    } ,
    NeptuneTraverserConverterDFEStep
]
+ not converted into Neptune steps: NeptuneHasStep([name.eq(josh)]),
Neptune steps:
[
    NeptuneMemoryTrackerStep,
    NeptuneInterleavingStep {
      StepInfo[joinVars=[?7, ?1], frontierElement=Vertex(?7):HasStep,
 pathElements={a=(last,Vertex(?1):GraphStep@[a])}, listPathElement={}, indexTime=0ms],
      DFEStep(Vertex) {
        DFENode {
          DFEJoinGroupNode[ children={
            DFEPatternNode[(?7, ?8, ?9, ?10) . project ALL[?7, ?9] graphFilters=(!=
 defaultGraph[526] . ), {rangeCountEstimate=92233720368547758O7}],
            DFEPatternNode[(?12, ?11, ?9, ?13) . project ALL[?9, ?12] graphFilters=(!=
 defaultGraph[526] . ), {rangeCountEstimate=92233720368547758O7}]
          }, {rangeCountEstimate=unknown}
          ]
        } [Vertex(?9):VertexStep, Vertex(?12):VertexStep]
      }
    }
]
+ not converted into Neptune steps: WherePredicateStep(eq(a)),
Neptune steps:
[
    DFECleanupStep
```

```
]


WARNING: >> [NeptuneHasStep([name.eq(josh)]), WherePredicateStep(eq(a))] << (or one of the
 children for each step) is not supported natively yet

Predicates
==========
# of predicates: 8
```

如中的資訊 explain (p. 351)需有關報告中 DFE 特定部份的說明，請參閱。

海王星中的小鬼 profile API

海王星 Grimlin profile API 會執行指定的 Gramlin 遍歷、收集有關執行的各種指標，並產生設定檔報告做為輸出。

> Note
>
> 此功能可從開始使用版本 1.0.0.0.463.0 (p. 985)。

它不同於 TinkerPop .profile () 步驟，以便能夠報告海王星引擎特定的資訊。

描述檔報告包含下列查詢計畫的相關資訊：

- 實體運算子管道
- 查詢執行和序列化的索引操作
- 結果的大小

profile API 使用 HTTP API 語法的延伸版本進行查詢，以 /gremlin/profile 做為端點，而不是 /gremlin。

海王星小鬼特有的參數 profile

- profile.results—boolean，允許的值：TRUE和FALSE，預設值：TRUE。

  如果為 true，則會收集查詢結果並顯示為profile報告的一部分。若為 false，則只會顯示結果計數。
- profile.chop—int，預設值：250。

  若為非零，則會在該字元數處截斷結果字串。這不會防止擷取所有結果。它只會限制描述檔報告中字串的大小。若設定為零，則字串會包含所有結果。
- profile.serializer—string，預設值：<null>。

  若為非 null，則會依此參數指定的格式，以序列化回應訊息傳回所收集的結果。這時會報告產生該回應訊息所需的索引操作數，以及要傳送到用戶端的大小 (以位元組為單位)。

  允許的值是<null>或任何有效的 MIME 類型或TinkerPop驅動程序「序列化器」枚舉值。

```
"application/json"                       or  "MIME_JSON"
"application/vnd.gremlin-v1.0+json"      or  "GRAPHSON_V1D0"
"application/vnd.gremlin-v2.0+json"      or  "GRAPHSON_V2D0"
"application/vnd.gremlin-v3.0+json"      or  "GRAPHSON_V3D0"
"application/vnd.gremlin-v1.0+gryo"      or  "GRYO_V1D0"
"application/vnd.gremlin-v3.0+gryo"      or  "GRYO_V3D0"
"application/vnd.gremlin-v1.0+gryo-lite" or  "GRYO_LITE_V1D0"
"application/vnd.graphbinary-v1.0"       or  "GRAPHBINARY_V1D0"
```

- profile.indexOps—boolean，允許的值：TRUE和FALSE，預設值：FALSE。

若為 true，則會顯示查詢執行和序列化期間所發生之所有索引操作的詳細報告。警告：此報告可能冗長。

海王星小精靈的樣本輸出 profile

以下是範例 profile 回應。

```
curl -X POST https://your-neptune-endpoint:port/gremlin/profile \
     -d '{"gremlin":"g.V().hasLabel(\"airport\")
                          .has(\"code\", \"AUS\")
                          .emit()
                          .repeat(in().simplePath())
                          .times(2)
                          .limit(100)",
         "profile.serializer":"application/vnd.gremlin-v3.0+gryo"}'
```

在博客文章中的航空路線示例圖表上執行時，此查詢生成以下profile報告，讓我為您繪製圖表-第 1 部分-航空路線。

```
*******************************************************
                Neptune Gremlin Profile
*******************************************************

Query String
==================
g.V().hasLabel("airport").has("code",
 "AUS").emit().repeat(in().simplePath()).times(2).limit(100)

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([~label.eq(airport), code.eq(AUS)]), RepeatStep(emit(true),
[VertexStep(IN,vertex), PathFilterStep(simple), RepeatEndStep],until(loops(2))),
 RangeGlobalStep(0,100)]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "AUS", ?) . project ?1 .], {estimatedCardinality=1,
 indexTime=84, hashJoin=true, joinTime=3, actualTotalOutput=1}
            PatternNode[(?1, <~label>, ?2=<airport>, <~>) . project ask .],
 {estimatedCardinality=3374, indexTime=29, hashJoin=true, joinTime=0, actualTotalOutput=61}
            RepeatNode {
                Repeat {
                    PatternNode[(?3, ?5, ?1, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .
 SimplePathFilter(?1, ?3)) .], {hashJoin=true, estimatedCardinality=50148, indexTime=0,
 joinTime=3}
                }
                Emit {
                    Filter(true)
                }
                LoopsCondition {
                    LoopsFilter([?1, ?3],eq(2))
                }
            }, annotations={repeatMode=BFS, emitFirst=true, untilFirst=false, leftVar=?1,
 rightVar=?3}
        }, finishers=[limit(100)], annotations={path=[Vertex(?1):GraphStep, Repeat[Vertex(?
3):VertexStep]], joinStats=true, optimizationTime=495, maxVarId=7, executionTime=323}
    },
```

```
    NeptuneTraverserConverterStep
]

Physical Pipeline
=================
NeptuneGraphQueryStep
    |-- StartOp
    |-- JoinGroupOp
        |-- SpoolerOp(100)
        |-- DynamicJoinOp(PatternNode[(?1, <code>, "AUS", ?) . project ?1 .],
 {estimatedCardinality=1, indexTime=84, hashJoin=true})
        |-- SpoolerOp(100)
        |-- DynamicJoinOp(PatternNode[(?1, <~label>, ?2=<airport>, <~>) . project ask .],
 {estimatedCardinality=3374, indexTime=29, hashJoin=true})
        |-- RepeatOp
            |-- <upstream input> (Iteration 0) [visited=1, output=1 (until=0, emit=1),
 next=1]
            |-- BindingSetQueue (Iteration 1) [visited=61, output=61 (until=0, emit=61),
 next=61]
                |-- SpoolerOp(100)
                |-- DynamicJoinOp(PatternNode[(?3, ?5, ?1, ?6) . project ?
1,?3 . IsEdgeIdFilter(?6) . SimplePathFilter(?1, ?3)) .], {hashJoin=true,
 estimatedCardinality=50148, indexTime=0})
            |-- BindingSetQueue (Iteration 2) [visited=38, output=38 (until=38, emit=0),
 next=0]
                |-- SpoolerOp(100)
                |-- DynamicJoinOp(PatternNode[(?3, ?5, ?1, ?6) . project ?
1,?3 . IsEdgeIdFilter(?6) . SimplePathFilter(?1, ?3)) .], {hashJoin=true,
 estimatedCardinality=50148, indexTime=0})
        |-- LimitOp(100)

Runtime (ms)
============
Query Execution:  392.686
Serialization:   2636.380

Traversal Metrics
=================
Step                                                               Count  Traversers
 Time (ms)    % Dur
-----------------------------------------------------------------------------------------------------------------
NeptuneGraphQueryStep(Vertex)                                        100         100
  314.162    82.78
NeptuneTraverserConverterStep                                        100         100
   65.333    17.22
                                           >TOTAL                      -           -
  379.495        -

Repeat Metrics
==============
Iteration  Visited   Output    Until     Emit     Next
------------------------------------------------------
        0        1        1        0        1        1
        1       61       61        0       61       61
        2       38       38       38        0        0
------------------------------------------------------
               100      100       38       62       62

Predicates
==========
# of predicates: 16

WARNING: reverse traversal with no edge label(s) - .in() / .both() may impact query
 performance

Results
```

```
=======
Count: 100
Output: [v[3], v[3600], v[3614], v[4], v[5], v[6], v[7], v[8], v[9], v[10], v[11], v[12],
 v[47], v[49], v[136], v[13], v[15], v[16], v[17], v[18], v[389], v[20], v[21], v[22],
 v[23], v[24], v[25], v[26], v[27], v[28], v[416], v[29], v[30], v[430], v[31], v[9...
Response serializer: GRYO_V3D0
Response size (bytes): 23566

Index Operations
================
Query execution:
    # of statement index ops: 3
    # of unique statement index ops: 3
    Duplication ratio: 1.0
    # of terms materialized: 0
Serialization:
    # of statement index ops: 200
    # of unique statement index ops: 140
    Duplication ratio: 1.43
    # of terms materialized: 393
```

除了呼叫 Neptune 所傳回的查詢計畫之外`explain`，`profile`結果還包括查詢執行周圍的執行階段統計資料。每個聯結操作都會標記執行聯結所花費的時間，以及傳遞它的實際解決方案數量。

`profile` 輸出包含核心查詢執行階段，以及序列化階段 (如果指定 `profile.serializer` 選項) 所花費的時間。

每個階段期間執行的索引操作明細也會包含在 `profile` 輸出底部。

請注意，連續執行相同查詢可能會由於快取而在執行時間和索引操作時顯示不同的結果。

對於使用 `repeat()` 步驟的查詢，如果 `repeat()` 步驟做為 `NeptuneGraphQueryStep` 的一部分下推，則每次反覆運算的邊界明細可供其使用。

啟用 DFE 時`profile`報表中的差異

啟用 Neptune DFE 替代查詢引擎時，`profile`輸出會有所不同：

優化遍歷：此部分類似於`explain`輸出中的部分，但包含其他信息。這包括在規劃中考慮的 DFE 運算子類型，以及相關的最壞情況和最佳案例成本估算。

實體管線：此段落會擷取用來執行查詢的運算子。 `DFESubQuery`元素摘要 DFE 用來執行其所負責之計劃部份的實體計劃。這些`DFESubQuery`元素會在下一節中展開，其中列出 DFE 統計資料。

DFE QueryEngine 統計值：此段落只有在 DFE 至少執行部分查詢時才會顯示。它概述了 DFE 特定的各種運行時統計信息，並包含查詢執行的各個部分所花費的時間的詳細明細，依`DFESubQuery`據。

在本節中，不同`DFESubQuery`元素中的巢狀子查詢會扁平化，唯一識別碼會以開頭的標頭`subQuery=`標記。

周遊測量結果：此段落顯示步驟層級遍歷測量結果，而當 DFE 引擎執行全部或部分查詢時，會顯示和/或的測量結果。`DFEStep` `NeptuneInterleavingStep`請參閱 使用和調整小鬼查詢 explainprofile (p. 364)。

> **Note**
>
> DFE 是在實驗室模式下發布的一項實驗功能，因此`profile`輸出的確切格式仍然可能會發生變化。

啟用海王星資料流引擎 (DFE) 時的範例`profile`輸出

當 DFE 引擎被用來運行 Grimlin 查詢，的輸出格式化如下小精靈 API `profile` (p. 357)面的例子。

查詢：

```
curl https://localhost:8182/gremlin/profile \
  -d "{\"gremlin\": \"g.withSideEffect('Neptune#useDFE', true).V().has('code',
 'ATL').out()\"}"
```

```
****************************************************
                Neptune Gremlin Profile
    ****************************************************

    Query String
    ==================
    g.withSideEffect('Neptune#useDFE', true).V().has('code', 'ATL').out()

    Original Traversal
    ==================
    [GraphStep(vertex,[]), HasStep([code.eq(ATL)]), VertexStep(OUT,vertex)]

    Optimized Traversal
    ===================
    Neptune steps:
    [
        DFEStep(Vertex) {
          DFENode {
            DFEJoinGroupNode[null](
              children=[
                DFEPatternNode((?1, vp://code[419430926], ?4, defaultGraph[526]) . project
 DISTINCT[?1] objectFilters=(in(ATL[452987149]) . ), {rangeCountEstimate=1},
                  opInfo=(type=PipelineJoin,
 cost=(exp=(in=1.00,out=1.00,io=0.00,comp=0.00,mem=0.00),wc=(in=1.00,out=1.00,io=0.00,comp=0.00,mem=0.0
                    disc=(type=PipelineScan,
 cost=(exp=(in=1.00,out=1.00,io=0.00,comp=0.00,mem=34.00),wc=(in=1.00,out=1.00,io=0.00,comp=0.00,mem=34
                DFEPatternNode((?1, ?5, ?6, ?7) . project ALL[?1, ?6] graphFilters=(!=
 defaultGraph[526] . ), {rangeCountEstimate=92233720368547758O7})],
              opInfo=[
                OperatorInfoWithAlternative[
                  rec=(type=PipelineJoin,
 cost=(exp=(in=1.00,out=27.76,io=0.00,comp=0.00,mem=0.00),wc=(in=1.00,out=27.76,io=0.00,comp=0.00,mem=0
                    disc=(type=PipelineScan,
 cost=(exp=(in=1.00,out=27.76,io=Infinity,comp=0.00,mem=295147905179352830000.00),wc=(in=1.00,out=27.76
                    alt=(type=PipelineScan,
 cost=(exp=(in=1.00,out=27.76,io=Infinity,comp=0.00,mem=295147905179352830000.00),wc=(in=1.00,out=27.76
          } [Vertex(?1):GraphStep, Vertex(?6):VertexStep]
        } ,
        NeptuneTraverserConverterDFEStep,
        DFECleanupStep
    ]


    Physical Pipeline
    =================
    DFEStep
        |-- DFESubQuery1

    DFEQueryEngine Statistics
    =================
    DFESubQuery1

 ###################################################################################################
    # ID # Out #1 # Out #2 # Name                   # Arguments
                                                                  # Mode # Units In #
 Units Out # Ratio  # Time (ms) #

 ###################################################################################################
```

```
   # 0  # 1      # -      # DFESolutionInjection  # solutions=[]
                                                                                  # -    # 0        #
 1         # 0.00   # 0.01      #
   #    #        #        #                       # outSchema=[]
                                                                               #      #          #
          #        #           #

 ##################################################################################################
   # 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/
neptune/vocab/v01/dfe/past/graph#089f43e3-4d71-4259-8d19-254ff63cee04/graph_1 # -    # 1
     # 1         # 1.00   # 0.02      #

 ##################################################################################################
   # 2  # 3      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/
neptune/vocab/v01/dfe/past/graph#089f43e3-4d71-4259-8d19-254ff63cee04/graph_2 # -    # 1
     # 242       # 242.00 # 0.02      #

 ##################################################################################################
   # 3  # 4      # -      # DFEMergeChunks        # -
                                                                                  # -    # 242      #
 242       # 1.00   # 0.01      #

 ##################################################################################################
   # 4  # -      # -      # DFEDrain              # -
                                                                                  # -    # 242      #
 0         # 0.00   # 0.01      #

 ##################################################################################################


   subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/
graph#089f43e3-4d71-4259-8d19-254ff63cee04/graph_1

 ##################################################################################################
   # ID # Out #1 # Out #2 # Name                 # Arguments
                  # Mode # Units In # Units Out # Ratio # Time (ms) #

 ##################################################################################################
   # 0  # 1      # -      # DFEPipelineScan      # pattern=Node(?1) with property 'code'
 as ?4 and label 'ALL' # -    # 0        # 1         # 0.00  # 0.22      #
   #    #        #        #                     # inlineFilters=[(?4 IN ["ATL"])]
                     #      #          #           #       #          #
   #    #        #        #                     # patternEstimate=1
                     #      #          #           #       #          #

 ##################################################################################################
   # 1  # 2      # -      # DFEMergeChunks       # -
                     # -    # 1        # 1         # 1.00  # 0.02      #

 ##################################################################################################
   # 2  # 4      # -      # DFERelationalJoin    # joinVars=[]
                     # -    # 2        # 1         # 0.50  # 0.09      #

 ##################################################################################################
   # 3  # 2      # -      # DFESolutionInjection # solutions=[]
                     # -    # 0        # 1         # 0.00  # 0.01      #
   #    #        #        #                     # outSchema=[]
                     #      #          #           #       #          #

 ##################################################################################################
   # 4  # -      # -      # DFEDrain              # -
                     # -    # 1        # 0         # 0.00  # 0.01      #

 ##################################################################################################
```

```
    subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/
graph#089f43e3-4d71-4259-8d19-254ff63cee04/graph_2

 ##########################################################################################################
    # ID # Out #1 # Out #2 # Name                # Arguments                          #
 Mode # Units In # Units Out # Ratio  # Time (ms) #

 ##########################################################################################################
    # 0  # 1      # -      # DFESolutionInjection # solutions=[]                      # -
    # 0        # 1         # 0.00   # 0.01      #
    #    #        #        #                     # outSchema=[?1]                   #
    #          #           #        #           #

 ##########################################################################################################
    # 1  # 2      # 3      # DFETee                # -                                # -
    # 1        # 2         # 2.00   # 0.01      #

 ##########################################################################################################
    # 2  # 4      # -      # DFEDistinctColumn     # column=?1                        # -
    # 1        # 1         # 1.00   # 0.21      #
    #    #        #        #                     # ordered=false                    #
    #          #           #        #           #

 ##########################################################################################################
    # 3  # 5      # -      # DFEHashIndexBuild     # vars=[?1]                        # -
    # 1        # 1         # 1.00   # 0.03      #

 ##########################################################################################################
    # 4  # 5      # -      # DFEPipelineJoin       # pattern=Edge((?1)-[?7:?5]->(?6))    # -
    # 1        # 242       # 242.00 # 0.51      #
    #    #        #        #                     # constraints=[]                   #
    #          #           #        #           #
    #    #        #        #                     # patternEstimate=9223372036854775807 #
    #          #           #        #           #

 ##########################################################################################################
    # 5  # 6      # 7      # DFESync               # -                                # -
    # 243      # 243       # 1.00   # 0.02      #

 ##########################################################################################################
    # 6  # 8      # -      # DFEForwardValue       # -                                # -
    # 1        # 1         # 1.00   # 0.01      #

 ##########################################################################################################
    # 7  # 8      # -      # DFEForwardValue       # -                                # -
    # 242      # 242       # 1.00   # 0.02      #

 ##########################################################################################################
    # 8  # 9      # -      # DFEHashIndexJoin      # -                                # -
    # 243      # 242       # 1.00   # 0.31      #

 ##########################################################################################################
    # 9  # -      # -      # DFEDrain              # -                                # -
    # 242      # 0         # 0.00   # 0.01      #

 ##########################################################################################################


    Runtime (ms)
    ============
    Query Execution: 11.744

    Traversal Metrics
    =================
    Step                                                               Count  Traversers
    Time (ms)    % Dur
```

```
----------------------------------------------------------------------------------------------------------------
   DFEStep(Vertex)                                                                242         242
      10.849    95.48
   NeptuneTraverserConverterDFEStep                                               242         242
       0.514     4.52
                                                >TOTAL                             -           -
      11.363        -

   Predicates
   ==========
   # of predicates: 18

   Results
   =======
   Count: 242


   Index Operations
   ================
   Query execution:
       # of statement index ops: 0
       # of terms materialized: 0
```

Note

由於 DFE 引擎是以實驗室模式發行的實驗性功能，因此`profile`輸出的確切格式可能會有所變更。

使用和調整小鬼查詢 explain`profile`

您通常可以使用從 Neptune 取得的報告中提供的資訊，在 Amazon Neptune 中調整 Gremlin 查詢，以取得更好的 (p. 351)效能。 (p. 357)要做到這一點，它有助於了解海王星如何處理 Girmlin 遍歷。

Important

在TinkerPop版本 3.4.11 中進行了更改，可提高查詢處理方式的正確性，但目前有時會嚴重影響查詢性能。
例如，此類查詢的執行速度可能會明顯變慢：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  out()
```

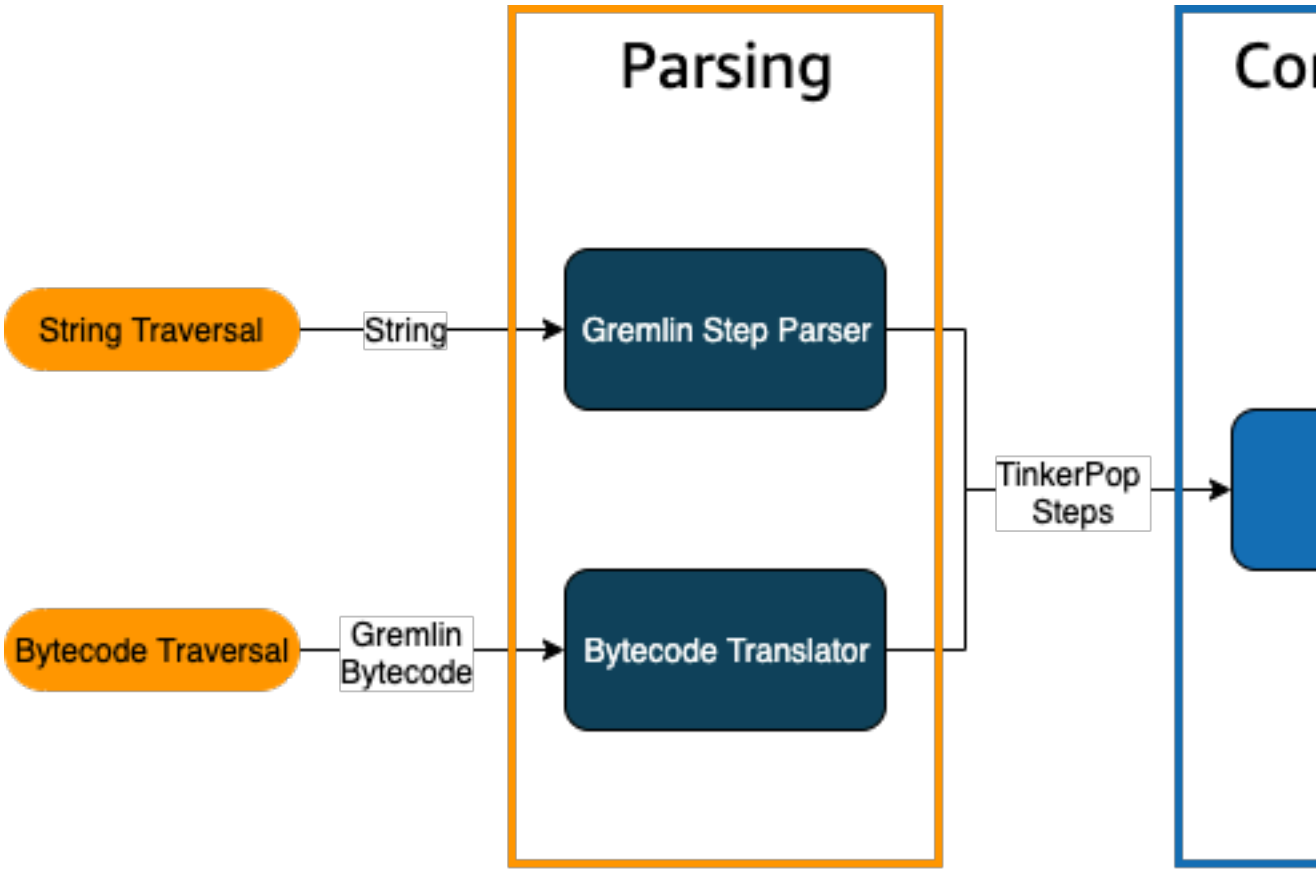
極限步驟之後的頂點現在會以 3.4.11 變更的非最佳方式擷取。TinkerPop若要避免這種情況，您可以在之後的任何點加入 barrier () 步驟來修改查詢`order().by()`。例如：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  barrier().
  out()
```

TinkerPop在海王星引擎版本 (p. 931) 1.0.5.0 中已啟用 3.4.11。

了解海王星中的小鬼遍歷處理

將 Grimlin 遍歷傳送至 Neptune 時，有三個主要處理程序會將遍歷轉換為要執行的基礎執行計畫。這些是解析，轉換和優化：



遍歷解析過程

處理遍歷的第一步是將其解析為通用語言。在海王星中，該通用語言是屬於 TinkerPopAPI 一部分的一組 TinkerPop步驟。這些步驟中的每個步驟都代表遍歷中的計算單位。

您可以將 Grimlin 遍歷發送到海王星作為字符串或字節碼。REST 端點和 Java 客戶端驅動程序`submit()`方法將遍歷作為字符串發送，如下例所示：

```
client.submit("g.V()")
```

使用 Gremlin 語言變體（GLV）的應用程序和語言驅動程序以字節碼發送遍歷。

遍歷轉換過程

處理遍歷的第二個步驟是將其步TinkerPop驟轉換為一組轉換和未轉換的 Neptune 步驟。Apache TinkerPop Gramlin 查詢語言中的大多數步驟都會轉換為海王星特定的步驟，這些步驟經過最佳化，以便在底層海王星引擎上執行。在遍歷中遇到沒有 Neptune 對等的步TinkerPop驟時，查詢引擎會處理該步驟和遍歷中的所有後續步驟。TinkerPop

如需在何種情況下可轉換哪些步驟的詳細資訊，請參閱小精靈步支持 (p. 377)。

遍歷優化過程

遍歷處理的最後一個步驟是通過最佳化器運行一系列轉換和未轉換的步驟，以嘗試確定最佳執行計劃。此最佳化的輸出是 Neptune 引擎所處理的執行計畫。

使用海王星小鬼 explain API 來調整查詢

海王星解釋 API 與小鬼explain()步驟不一樣。它返回海王星引擎執行查詢時將處理的最終執行計劃。因為它不會執行任何處理，所以不論使用哪些參數，都會傳回相同的計畫，而且其輸出不包含有關實際執行的統計資料。

考慮以下簡單的遍歷，找到安克雷奇的所有機場頂點：

```
g.V().has('code','ANC')
```

有兩種方法可以通過海王星 explain API 運行此遍歷。第一種方法是對解釋端點進行 REST 調用，如下所示：

```
curl -X POST https://your-neptune-endpoint:port/gremlin/explain -d
 '{"gremlin":"g.V().has('code','ANC')"}'
```

第二種方法是將海王星工作台的%% 麻黃 (p. 42)細胞魔術與explain參數一起使用。這會將儲存格主體中包含的遍歷傳遞至 Neptune explain API，然後在執行儲存格時顯示產生的輸出：

```
%%gremlin explain

g.V().has('code','ANC')
```

產生的 explain API 輸出描述了海王星針對遍歷的執行計劃。如下圖所示，該計劃包括處理管道中的 3 個步驟中的每個步驟：

```
Explain


**************************************************************
                  Neptune Gremlin Explain
**************************************************************

Query String
============

g.V().has('code','ANC')

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([code.eq(ANC)])]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project dist
            PatternNode[(?1, <code>, "ANC", ?) . project ask
        }, annotations={path=[Vertex(?1):GraphStep], maxVarId
    },
    NeptuneTraverserConverterStep
]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "ANC", ?) . project ?1 .
        }, annotations={path=[Vertex(?1):GraphStep], maxVarId
    },
    NeptuneTraverserConverterStep
]

Predicates
==========
# of predicates: 22
```

透過查看未轉換的步驟來調整遍歷

在海王星 explain API 輸出中尋找的第一件事是針對未轉換為海王星原生步驟的 Gremlin 步驟。在查詢計劃中，當遇到無法轉換為 Neptune 原生步驟的步驟時，它和計劃中的所有後續步驟都會由 Gramlin 伺服器處理。

在上面的例子中，遍歷中的所有步驟都被轉換。讓我們來看看這個遍歷的 explain API 輸出：

```
g.V().has('code','ANC').out().choose(hasLabel('airport'), values('code'), constant('Not an
 airport'))
```

如下圖所示，Neptune 無法轉換choose()步驟：

```
Explain
```

```
************************************************************
                Neptune Gremlin Explain
************************************************************

Query String
============

g.V().has('code','ANC').out().choose(hasLabel('airport'), val

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([code.eq(ANC)]), VertexStep(OU

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project dis
            PatternNode[(?1, <code>, "ANC", ?) . project ask
            PatternNode[(?1, ?5, ?3, ?6) . project ?1,?3 . Is
            PatternNode[(?3, <~label>, ?4, <~>) . project ask
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3
    },
    NeptuneTraverserConverterStep
]
+ not converted into Neptune steps: [ChooseStep([HasStep([~la

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "ANC", ?) . project ?1 .
            PatternNode[(?1, ?5, ?3, ?6) . project ?1,?3 . Is
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3
    },
    NeptuneTraverserConverterStep
]
+ not converted into Neptune steps: [ChooseStep([NeptuneHasSt
```

您可以做幾件事來調整遍歷的性能。首先是以這樣的方式重寫它，以消除無法轉換的步驟。另一種方法是將步驟移動到遍歷的末尾，以便所有其他步驟都可以轉換為本機步驟。

包含未轉換之步驟的查詢計劃並不一定需要調整。如果無法轉換的步驟位於遍歷的結尾，並且與輸出的格式化方式相關，而不是圖形的遍歷方式，它們可能對效能沒有什麼影響。

檢查來自 Neptune `explain` API 的輸出時要尋找的另一件事是不使用索引的步驟。以下遍歷查找所有有航班降落在安克雷奇的機場：

```
g.V().has('code','ANC').in().values('code')
```

此遍歷的解釋 API 的輸出是：

```
*******************************************************
                Neptune Gremlin Explain
*******************************************************

Query String
============

g.V().has('code','ANC').in().values('code')

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([code.eq(ANC)]), VertexStep(IN,vertex),
 PropertiesStep([code],value)]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(PropertyValue) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .]
            PatternNode[(?1, <code>, "ANC", ?) . project ask .]
            PatternNode[(?3, ?5, ?1, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .]
            PatternNode[(?3, <~label>, ?4, <~>) . project ask .]
            PatternNode[(?3, ?7, ?8, <~>) . project ?3,?8 . ContainsFilter(?7 in
 (<code>)) .]
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep, PropertyValue(?
8):PropertiesStep], maxVarId=9}
    },
    NeptuneTraverserConverterStep
]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(PropertyValue) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "ANC", ?) . project ?1 .], {estimatedCardinality=1}
            PatternNode[(?3, ?5, ?1, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .],
 {estimatedCardinality=INFINITY}
            PatternNode[(?3, ?7=<code>, ?8, <~>) . project ?3,?8 .],
 {estimatedCardinality=7564}
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep, PropertyValue(?
8):PropertiesStep], maxVarId=9}
    },
    NeptuneTraverserConverterStep
]
```

```
Predicates
==========
# of predicates: 26

WARNING: reverse traversal with no edge label(s) - .in() / .both() may impact query
 performance
```

因為無法使用 Neptune 維護的 3 個索引之一來處理遍歷中的`in()`步驟 (請參閱如何在海王星中索引陳述式 (p. 104)和海王星中的小鬼语句 (p. 347))，因此輸出底部的`WARNING`訊息會發生。因為`in()`步驟不包含邊緣篩選器，因此無法使用`SPOG`、`POGS`或`GPSO`索引來解析。相反，Neptune 必須執行聯合掃描以查找請求的頂點，效率要低得多。

在這種情況下，有兩種方法可以調整遍歷。第一個是將一或多個篩選條件新增至`in()`步驟，以便可以使用索引查詢來解析查詢。對於上面的例子，這可能是：

```
g.V().has('code','ANC').in('route').values('code')
```

來自海王星 `explain` API 的修訂遍歷的輸出不再包含該`WARNING`消息：

```
*******************************************************
                Neptune Gremlin Explain
*******************************************************

Query String
============

g.V().has('code','ANC').in('route').values('code')

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([code.eq(ANC)]), VertexStep(IN,[route],vertex),
 PropertiesStep([code],value)]

Converted Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(PropertyValue) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .]
            PatternNode[(?1, <code>, "ANC", ?) . project ask .]
            PatternNode[(?3, ?5, ?1, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .
 ContainsFilter(?5 in (<route>)) .]
            PatternNode[(?3, <~label>, ?4, <~>) . project ask .]
            PatternNode[(?3, ?7, ?8, <~>) . project ?3,?8 . ContainsFilter(?7 in
 (<code>)) .]
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep, PropertyValue(?
8):PropertiesStep], maxVarId=9}
    },
    NeptuneTraverserConverterStep
]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(PropertyValue) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "ANC", ?) . project ?1 .], {estimatedCardinality=1}
            PatternNode[(?3, ?5=<route>, ?1, ?6) . project ?1,?3 . IsEdgeIdFilter(?6) .],
 {estimatedCardinality=32042}
            PatternNode[(?3, ?7=<code>, ?8, <~>) . project ?3,?8 .],
 {estimatedCardinality=7564}
```

```
        }, annotations={path=[Vertex(?1):GraphStep, Vertex(?3):VertexStep, PropertyValue(?
8):PropertiesStep], maxVarId=9}
    },
    NeptuneTraverserConverterStep
]

Predicates
==========
# of predicates: 26
```

如果您正在執行許多此類遍歷，則另一個選項是在啟用選用OSGP索引的 Neptune DB 叢集中執行它們 (請參閱啟用 OSGP 索引 (p. 105))。啟用OSGP索引有缺點：

- 它必須在加載任何數據之前在數據庫集群中啟用。
- 頂點和邊的插入速率可能會降低多達 23%。
- 存儲使用量將增加 20% 左右。
- 讀取將要求分散到所有索引的查詢可能會增加延遲。

有一個OSGP索引對於一組受限制的查詢模式很有意義，但除非您經常運行這些模式，否則通常最好嘗試確保您編寫的遍歷可以使用三個主索引來解析。

使用大量謂詞

Neptune 會將圖形中的每個邊緣標籤和每個不同的頂點或邊屬性名稱視為述詞，並且預設情況下設計為使用相對較少數量的不同述詞。當您的圖形資料中有幾千個以上的述詞時，效能可能會降低。

如果是這種情況，海王星explain輸出將警告您：

```
Predicates
==========
# of predicates: 9549
WARNING: high predicate count (# of distinct property names and edge labels)
```

如果重工資料模型以減少標籤和屬性的數目，以及因此減少述詞的數目不方便，調整遍歷的最佳方法是在啟用OSGP索引的資料庫叢集中執行它們，如上所述。

使用海王星小鬼 profile API 來調整遍歷

海王星 profile API 與小鬼profile()步驟完全不同。與 explain API 一樣，其輸出包括 Neptune 引擎在執行遍歷時使用的查詢計劃。此外，根據其參數的設定方式，profile輸出還包括遍歷的實際執行統計資料。

再次，採取簡單的遍歷，找到安克雷奇的所有機場頂點：

```
g.V().has('code','ANC')
```

與 explain API 一樣，您可以使用 REST 調用來調用 profile API：

```
curl -X POST https://your-neptune-endpoint:port/gremlin/profile -d
 '{"gremlin":"g.V().has('code','ANC')"}'
```

您還可以使用海王星工作台的%% 麻黃 (p. 42)細胞魔術與profile參數。這會將儲存格主體中包含的遍歷傳遞至 Neptune profile API，然後在執行儲存格時顯示產生的輸出：

```
%%gremlin profile
```

```
g.V().has('code','ANC')
```

產生的 profile API 輸出包含 Neptune 的遍歷執行計劃和有關計劃執行的統計資料，如下圖所示：

```
Profile

************************************************************
                Neptune Gremlin Profile
************************************************************

Query String
==================

g.V().has('code','ANC')

Original Traversal
==================
[GraphStep(vertex,[]), HasStep([code.eq(ANC)])]

Optimized Traversal
===================
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <code>, "ANC", ?) . project ?1 .
        }, annotations={path=[Vertex(?1):GraphStep], joinStats
    },
    NeptuneTraverserConverterStep
]

Physical Pipeline
=================
NeptuneGraphQueryStep
    |-- StartOp
    |-- JoinGroupOp
        |-- SpoolerOp(1000)
        |-- DynamicJoinOp(PatternNode[(?1, <code>, "ANC", ?)

Runtime (ms)
============
Query Execution: 5.096

Traversal Metrics
=================
Step                                                               
-------------------------------------------------------------------
NeptuneGraphQueryStep(Vertex)
NeptuneTraverserConverterStep
```

在profile輸出中，執行計畫區段僅包含遍歷的最終執行計畫，而不包含中繼步驟。管線區段包含執行的實體管線作業，以及遍歷執行所花費的實際時間 (以毫秒為單位)。運行時指標在比較兩個不同版本的遍歷在優化時所採取的時間非常有用。

Note

遍歷的初始執行階段通常比後續執行階段長，因為第一個執行階段會導致快取相關資料。

profile輸出的第三個部分包含執行統計信息和遍歷的結果。要了解這些信息在調整遍歷時如何有用，請考慮以下遍歷，它會找到名稱以「Anchora」開頭的每個機場，以及從這些機場到達的所有機場，返回機場代碼，航線和距離：

```
%%gremlin profile

g.withSideEffect("Neptune#fts.endpoint", "{your-OpenSearch-endpoint-URL").
    V().has("city", "Neptune#fts Anchora~").
    repeat(outE('route').inV().simplePath()).times(2).
    project('Destination', 'Route').
        by('code').
        by(path().by('code').by('dist'))
```

海王星 profile API 輸出中的周遊度量

所有profile輸出中可用的第一組量度是遍歷量度。這些類似於 Gemlin profile() 步驟指標，有一些不同之處：

```
Traversal Metrics
=================
Step                                                               Count  Traversers
 Time (ms)    % Dur
-------------------------------------------------------------------------------------------------------------------
NeptuneGraphQueryStep(Vertex)                                       3856        3856
   91.701     9.09
NeptuneTraverserConverterStep                                       3856        3856
   38.787     3.84
ProjectStep([Destination, Route],[value(code), ...                  3856        3856
  878.786    87.07
  PathStep([value(code), value(dist)])                              3856        3856
  601.359
                                                >TOTAL                   -           -
 1009.274        -
```

遍歷測量結果表格的第一個資料欄列示遍歷所執行的步驟。前兩個步驟通常是海王星特定的步驟，和。NeptuneGraphQueryStep NeptuneTraverserConverterStep

NeptuneGraphQueryStep代表可由 Neptune 引擎以原生方式轉換和執行之遍歷整個部分的執行時間。

NeptuneTraverserConverterStep表示將那些轉換步驟的輸出轉換為TinkerPop遍歷器的過程，這些步驟允許無法被轉換的步驟（如果有的話）被處理，或以兼容的格式返回結果。TinkerPop

在上面的例子中，我們有幾個未轉換的步驟，因此我們看到每個TinkerPop步驟（ProjectStep，PathStep）然後在表中顯示為一行。

表格中的第二欄會報告通過步驟的代表遍歷數目，而第三欄則報告通過該步驟的遊覽者數目，如設定檔步驟文件中TinkerPop所述。CountTraversers

在我們的例子中，有 3,856 個頂點和 3,856 由返回的遍歷，並且這些數字在整個剩餘的處理過程中保持不變NeptuneGraphQueryStep，因為並且正在格式化結果，ProjectStep而不PathStep是過濾它們。

Note

不同於TinkerPop，Neptune 引擎不會通過在其NeptuneGraphQueryStep和NeptuneTraverserConverterStep步驟中膨脹來

優化性能。膨脹是將遍歷器結合在同一頂點以減少TinkerPop操作開銷的操作，這就是導致Count和Traversers數字不同的原因。因為膨脹只發生在 Neptune 委派的步驟中TinkerPop，而不是 Neptune 原生處理的步驟中，因此Count和Traverser資料行很少不同。

「時間」資料欄會報告步驟所花費的毫秒數，而% Dur資料欄則報告步驟所花費的總處理時間百分比。這些指標告訴您通過顯示花費最多時間的步驟來將調整工作集中在哪裡。

海王星 profile API 輸出中的索引操作指標

Neptune 設定檔 API 輸出中的另一組度量是索引作業：

```
Index Operations
================
Query execution:
    # of statement index ops: 23191
    # of unique statement index ops: 5960
    Duplication ratio: 3.89
    # of terms materialized: 0
```

這些報告：

- 索引查詢的總數。
- 執行的唯一索引查詢數目。
- 總索引查詢與唯一索引查詢的比率。較低的比例表示較少的冗餘。
- 從字詞字典實現的術語數目。

海王星 profile API 輸出中的重複量度

如果您的遍歷使用上述範例中的repeat()步驟，則輸出中會出現包含重複量度的區段：profile

```
Repeat Metrics
==============
Iteration  Visited   Output    Until     Emit      Next
-------------------------------------------------------
        0        2        0        0        0        2
        1       53        0        0        0       53
        2     3856     3856     3856        0        0
-------------------------------------------------------
              3911     3856     3856        0       55
```

這些報告：

- 列 (Iteration欄) 的迴圈計數。
- 循環 (Visited列) 訪問的元素的數量。
- 循環輸出的元素數目 (Output列) 。
- 由循環輸出的最後一個元素 (Until列) 。
- 循環 (Emit列) 發出的元素的數量。
- 從迴圈傳遞至後續迴圈 (Next資料行) 的元素數目。

這些重複指標對於理解遍歷的分支因素非常有幫助，以了解數據庫正在完成多少工作的感覺。您可以使用這些數字來診斷效能問題，尤其是當相同的遍歷在不同參數的情況下執行明顯不同時。

海王星 profile API 輸出中的全文檢索搜尋量度

當遍歷使用全文檢索搜尋 (p. 604)查閱 (如上述範例所示) 時，輸出中會顯示包含全文檢索搜尋 (FTS) 量度的區段：profile

```
FTS Metrics
==============
SearchNode[(idVar=?1, query=Anchora~, field=city) . project ?1 .],
    {endpoint=your-OpenSearch-endpoint-URL, incomingSolutionsThreshold=1000,
 estimatedCardinality=INFINITY,
    remoteCallTimeSummary=[total=65, avg=32.500000, max=37, min=28],
    remoteCallTime=65, remoteCalls=2, joinTime=0, indexTime=0, remoteResults=2}

    2 result(s) produced from SearchNode above
```

這會顯示傳送至 ElasticSearch (ES) 叢集的查詢，並報告與之互動有關的數個度量，ElasticSearch可協助您找出與全文檢索搜尋相關的效能問題：

- 有關ElasticSearch索引調用的摘要信息：
  - 所有 RemoteCall 滿足查詢所需的總毫秒數 () `total`。
  - 在遠端呼叫 () `avg` 花費的平均毫秒數。
  - 在 RemoteCall () `min` 中花費的最小毫秒數。
  - 用於遠端呼叫 () `max` 的最大毫秒數。
- 遠端呼叫 ElasticSearch (`remoteCallTime`) 使用的總時間。
- 對 ElasticSearch (`remoteCalls`) 所做的遠端呼叫數目。
- 結果 () 聯ElasticSearch結所花費的毫秒`joinTime`數。
- 用於索引查詢 (`indexTime`) 的毫秒數。
- () 傳回的結果ElasticSearch總`remoteResults`數。

亞馬遜海王星的原生小精靈步進支持

如中所述，Amazon Neptune 引擎目前並未對所有 Gremlin 步驟提供完整的原生支援。調整小精靈查詢 (p. 364)目前的支援分為四類：

- 總是可以轉換為原生海王星引擎操作的 gremlin 步驟 (p. 377)
- 在某些情況下，可以轉換為原生海王星引擎操作的 gremlin 步驟 (p. 378)
- 永遠不會轉換為原生海王星引擎作業的 Gremlin 步驟 (p. 380)
- 海王星根本不支持的小鬼步驟 (p. 381)

總是可以轉換為原生海王星引擎操作的 gremlin 步驟

只要符合下列條件，許多 Gemlin 步驟就可以轉換為原生 Neptune 引擎作業：

- 它們不會在查詢前面加上無法轉換的步驟。
- 他們的父步，如果有的話，可以轉換，
- 他們所有的子遍歷（如果有的話）都可以轉換。

如果下列 Gemlin 步驟符合這些條件，一律會轉換為原生 Neptune 引擎作業：

- 和 ( )
- 為 ( )
- 計數 ( )
- E ()
- 發射 ( )

- 解釋 ( )
- 組 ( )
- 群組數 ()
- 有 ( )
- 身份 ( )
- 是 ( )
- 鍵 ( )
- 標籤 ( )
- 限制 ( )
- 本地 ( )
- 循環 ( )
- 不 ( )
- 或 ( )
- 配置文件 ( )
- 屬性 ( )
- 子圖 ( )
- 直到 ( )
- V ()
- 值 ( )
- 值映射 ()
- 值 ( )

在某些情況下，可以轉換為原生海王星引擎操作的 gremlin 步驟

在某些情況下，某些 Gemlin 步驟可以轉換為原生 Neptune 引擎操作，但在其他情況下則不能轉換為：

- AdDe ()-addE( ) 步驟通常可以轉換為原生 Neptune 引擎作業，除非它緊接著包含遍歷作為索引鍵的property( )步驟。
- addV ()-addV( ) 步驟通常可以轉換為原生 Neptune 引擎作業，除非它緊接著包含遍歷作為索引鍵的property( )步驟，或者除非指派了多個標籤。
- aggregate ()-aggregate( ) 步驟通常可以轉換為本機 Neptune 引擎操作，除非該步驟用於子遍歷或子遍歷，或者除非要存儲的值是頂點，邊緣，ID，標籤或屬性值以外的值。

  在下面的例子中，不aggregate( )被轉換，因為它正在子遍歷中使用：

```
g.V().has('code','ANC').as('a')
     .project('flights').by(select('a')
     .outE().aggregate('x'))
```

  在此範例中，彙總 () 不會轉換，因為儲存min( )的是值的：

```
g.V().has('code','ANC').outE().aggregate('x').by(values('dist').min())
```

- barrier ()-barrier( ) 步驟通常可以轉換為原生 Neptune 引擎作業，除非步驟後續步驟未轉換。
- cap ()-轉換步驟的唯一情況是將cap( )步驟與返回頂點，邊緣，id 或 poperty 值的聚合的展開版本結合的unfold( )步驟結合。在這個例子中，cap( )將被轉換，因為它後面是.unfold( )：

```
g.V().has('airport','country','IE').aggregate('airport').limit(2)
```

```
    .cap('airport').unfold()
```

但是，如果您刪除.unfold()，cap()將不會被轉換：

```
g.V().has('airport','country','IE').aggregate('airport').limit(2)
    .cap('airport')
```

- coalesce () -轉換coalesce()步驟的唯一情況是遵循配方頁面上建議的 Upsert 模式。TinkerPop不允許使用其他合併 () 模式。轉換僅限於可以轉換所有子遍歷的情況下，它們都產生與輸出相同的類型（頂點，邊緣，ID，值，鍵或標籤），它們都遍歷到一個新元素，並且它們不包含步驟。repeat()
- constant ()-常量（）步驟當前只有在遍歷的一sack().by()部分中使用它來分配一個常量值時才被轉換，如下所示：

```
g.V().has('code','ANC').sack(assign).by(constant(10)).out().limit(2)
```

- CyClickPath ()-cyclicPath() 步驟通常可以轉換為原生海王星引擎操作，除非該步驟與by()from()、或to()調製器一起使用。例如，在以下查詢中，cyclicPath()不會轉換：

```
g.V().has('code','ANC').as('a').out().out().cyclicPath().by('code')
g.V().has('code','ANC').as('a').out().out().cyclicPath().from('a')
g.V().has('code','ANC').as('a').out().out().cyclicPath().to('a')
```

- drop ()-drop() 步驟通常可以轉換為原生 Neptune 引擎作業，除非步驟在 asideEffect() 或optional()步驟中使用。
- fold ()-只有兩種情況下 fold () 步驟可以轉換，即在TinkerPop食譜頁面上建議的 Upsert 模式中使用它時，以及在如下所示的上下group().by()文中使用時：

```
g.V().has('code','ANC').out().group().by().by(values('code', 'city').fold())
```

- id ()-除非在屬性上使用，否則會轉換id()步驟，如下所示：

```
g.V().has('code','ANC').properties('code').id()
```

- order ()-order() 步驟通常可以轉換為原生 Neptune 引擎作業，除非下列其中一項成立：
  - 這個order()步驟是在一個嵌套的子遍歷中，如下所示：

    ```
    g.V().has('code','ANC').where(V().out().order().by(id))
    ```

  - 正在使用本地排序，例如使用order(local)。
  - by()調製中正在使用自定義比較器以按順序進行排序。一個例子是這種使用sack()：

    ```
    g.withSack(0).
      V().has('code','ANC').
          repeat(outE().sack(sum).by('dist').inV()).times(2).limit(10).
          order().by(sack())
    ```

  - 同一元素上有多個訂單。
- project ()-project() 步驟通常可以轉換為本機 Neptune 引擎操作，除非之後的by()語句數量與指定的標籤數量project()不匹配，如下所示：

```
g.V().has('code','ANC').project('x', 'y').by(id)
```

- range () — 只有當有問題範圍的下端為零時，才會轉換range()步驟 (例如，range(0,3))。
- repeat ()-repeat() 步驟通常可以轉換為原生 Neptune 引擎操作，除非它嵌套在另一個repeat()步驟中，如下所示：

```
g.V().has('code','ANC').repeat(out().repeat(out()).times(2)).times(2)
```

- sack ()-該sack()步驟通常可以轉換為本機海王星引擎操作，但下列情況除外：
  - 如果正在使用非數字麻袋運算符。
  - 如果正在使用+，-mult，divmin和max以外的數字麻袋運算符。
  - 如果sack()在一個where()步驟中使用基於麻袋值進行過濾，如下所示：

```
g.V().has('code','ANC').sack(assign).by(values('code')).where(sack().is('ANC'))
```

- sum ()-該sum()步驟通常可以轉換為本地 Neptune 引擎操作，但不能在用於計算全局求和時，如下所示：

```
g.V().has('code','ANC').outE('routes').values('dist').sum()
```

- union ()-union() 步驟可以轉換為原生 Neptune 引擎操作，只要它是查詢中的最後一個步驟，除了終端步驟。
- 展開()-unfold() 步驟只有在TinkerPop配方頁面上建議的 Upsert 模式中使用時，才能將步驟轉換為原生 Neptune 引擎操作，並且與這cap()樣一起使用時：

```
g.V().has('airport','country','IE').aggregate('airport').limit(2)
     .cap('airport').unfold()
```

- where ()-where() 步驟通常可以轉換為原生 Neptune 引擎作業，但下列情況除外：
  - 當使用 by () 調製時，如下所示：

```
g.V().hasLabel('airport').as('a')
     .where(gt('a')).by('runways')
```

  - 使用eq、、neq和以外的比較運算without子時。within
  - 使用使用者提供的彙總時。

永遠不會轉換為原生海王星引擎作業的 Gremlin 步驟

Neptune 支援下列 Gemlin 步驟，但永遠不會轉換為原生海王星引擎作業。相反，它們是由 Gemlin 服務器執行的。

- 選擇 ( )
- 硬幣 ( )
- 注入 ( )
- 匹配 ( )
- 數學 ( )
- 最大 ( )
- 平均值 ( )
- 分鐘 ( )
- 選項 ( )
- 可選 ( )
- 路徑 ( )
- 屬性映射 ( )
- 樣品 ( )
- 跳過 ( )
- 尾巴 ( )

- 時間限制 ( )
- 樹 ( )

海王星根本不支持的小鬼步驟

海王星完全不支持以下小鬼步驟。在大多數情況下，這是因為他們需要一個GraphComputer，海王星目前不支持。

- 連接的組件 ( )
- io ( )
- 最短路徑 ( )
- 有電腦 ()
- 網頁排名 ()
- 對等壓力 ()
- 程序 ( )

該io()步驟實際上是部分支持的，因為它可以用於read()從 URL，但不能使用write()。

使用小鬼與海王星 DFE 查詢引擎

如果您在實驗室模式 (p. 127)中完全啟用稱為 DFE 的 Neptune 替代查詢引擎 (p. 129) (藉由將neptune_lab_mode資料庫叢集參數設定為DFEQueryEngine=enabled)，Neptune 會將唯讀的 Gremlin 查詢/遍歷轉譯為中繼邏輯表示法，並儘可能在 DFE 引擎上執行它們。

然而，DFE 尚未支援所有的小鬼步驟。當步驟無法在 DFE 上以原生方式執行時，Neptune 會回頭執TinkerPop行步驟。explain和profile報告包含發生這種情況時的警告。

> Note
>
> 從引擎版本 1.0.5.0 (p. 931) 開始，在沒有原生支援的情況下處理 Gemlin 步驟的預設 DFE 行為已變更。凡以前 DFE 發動機倒回海王星 Grimlin 發動機，現在它回落在香草發動機。TinkerPop

DFE 引擎原生支援的小精靈步驟

- **GraphStep**
- **VertexStep**
- **EdgeVertexStep**
- **IdStep**
- **TraversalFilterStep**
- **PropertiesStep**
- **HasStep**篩選對性質和 ID 和標籤上的頂點和邊緣的支援，但文字和Without述詞除外。
- **WherePredicateStep**使用Path範圍過濾器，但沒有ByModulation，SideEffect或Map查找支持
- **DedupGlobalStep**、例外ByModulationSideEffect、和Map查詢支援。

查詢計劃交錯

當翻譯程序遇到沒有對應的本機 DFE 運算子的 Gremlin 步驟時，在回到使用 Tinkerpop 之前，它會嘗試尋找其他可以在 DFE 引擎上以原生方式執行的中繼查詢組件。它通過將交錯邏輯應用到頂級遍歷來做到這一點。結果是，盡可能使用支援的步驟。

任何此類中繼、非前置詞查詢轉譯都會使用explain和profile輸出NeptuneInterleavingStep中表示。

為了進行效能比較，您可能想要關閉查詢中的交錯，同時仍使用 DFE 引擎來執行前置詞零件。或者，您可能只想使用TinkerPop引擎執行非前綴查詢。您可以通過使用disableInterleaving查詢提示來做到這一點。

正如具有值的使用 DFE (p. 322)查詢提示false可防止在 DFE 上執行查詢一樣，值為的disableInterleaving查詢提示會關true閉 DFE 交錯以進行查詢翻譯。例如：

```
g.with('Neptune#disableInterleaving', true)
 .V().has('genre','drama').in('likes')
```

更新了小精靈explain和輸出 profile

Gemlin 解釋 (p. 347)提供了有關海王星用來運行查詢的優化遍歷的詳細信息。請參閱 DFE explain 輸出範 (p. 355)例，以取得啟用 DFE 引擎時explain輸出的外觀範例。

會小精靈 API profile (p. 357)執行指定的 Gemlin 遍歷、收集有關執行的各種指標，並產生描述檔報告，其中包含有關最佳化查詢計畫和各種運算子的執行階段統計資料的詳細資料。如需啟用 DFE 引擎時profile輸profile出的範 (p. 360)例，請參閱 DFE 輸出範例。

> Note
>
> 由於 DFE 引擎是以實驗室模式發行的實驗性功能，因此explain和profile輸出的確切格式可能會有所變更。

使用開放密碼訪問海王星圖

海王星支持使用 OpenCypher 構建圖形應用程序，目前是使用圖形數據庫的開發人員最流行的查詢語言之一。開發人員、商務分析師和資料科學家，例如 OpenCypher 的 SQL 啟發語法，因為它提供了一個熟悉的結構來撰寫圖形應用程式的查詢。

OpenCypher 是最初由 Neo4j 的，然後開源於 2015 年開發的屬性圖的聲明式查詢語言，並在一個阿帕奇 2 開源許可證下的 OpenCypher 項目做出了貢獻。它的語法記錄在 Cypher 查詢語言參考版本 9 中。

如需海王星支援 OpenCypher 規格的限制和差異，請參閱。亞馬遜海王星中的開放密碼標準合規性 (p. 95)

> Note
>
> 密碼查詢語言的當前 Neo4j 的實現已經從 OpenCypher 規範的某些方面有所分歧。如果要將當前 Neo4j 的密碼遷移到海王星，請參閱Neptune 與 Neo4j 的兼容性 (p. 239)並重寫密碼查詢以在 Neptune 上的 OpenCpher 中運行 (p. 242)獲得幫助。

從發動機版本 1.1.1.0 開始，開放密碼可用於海王星的生產使用。

小精靈與開放密碼：相似之處和差異

小精靈和 OpenCypher 都是屬性圖查詢語言，並且它們在許多方面都是互補的。

Gremlin 旨在吸引程序員並無縫融入代碼中。其結果是，格林是必要的設計，而 OpenCypher 的聲明式語法可能會感覺更熟悉與 SQL 或 SPARQL 經驗的人。對於在 Jupyter 筆記本中使用 Python 的數據科學家來說，Gremlin 似乎更自然，而 OpenCypher 對於具有某些 SQL 背景的商業用戶來說似乎更直觀。

好處是，您不必在海王星的小鬼和開放密碼之間進行選擇。無論使用哪種語言輸入資料，任一語言的查詢都可以在相同的圖形上運作。您可能會發現將 Gremlin 用於某些事情更方便，對其他人使用 OpenCypher，具體取決於您正在做的事情。

Gremlin 使用命令式語法，可讓您透過一系列步驟控制您在圖形中移動的方式，每個步驟都需要資料串流，對其執行某些動作 (使用篩選器、對應等)，然後將結果輸出到下一個步驟。Girmlin 查詢通常採用表單`g.V()`，然後是額外的步驟。

在 OpenCypher 中，您可以使用由 SQL 啟發的聲明式語法，指定節點和關係的模式，以使用主題語法（如）在圖中查找。`()-[]->()`OpenCypher 查詢通常以`MATCH`子句開頭，接著是其他子句，例如`WHERE`和。`WITH``RETURN`

開始使用開放密碼

您可以使用 OpenCypher 在海王星中查詢屬性圖數據，而不管它是如何加載的，但是您不能使用 OpenCypher 來查詢加載為 RDF 的數據。

海王星批量加載程序 (p. 255)接受以 CSV 格式的屬性圖數據用於小精靈，並以 CSV 格式 (p. 262)接受 OpenCypher 的屬性圖數據。 (p. 267)另外，當然，您可以使用 Gremlin 和/或 OpenCypher 查詢將屬性數據添加到圖形中。

有許多在線教程可用於學習 Cypher 查詢語言。在這裡，OpenCypher 查詢的幾個簡單的例子可以幫助您了解語言，但到目前為止，開始使用 OpenCypher 查詢海王星圖的最好和最簡單的方法是通過在海王星工作台中使用 OpenCypher 筆記本電腦。 (p. 30)該工作台是開源的，並託管GitHub在 https://github.com/aws-samples/ 上amazon-neptune-samples。

您可以在GitHub海王星圖形筆記本存儲庫中找到 OpenCypher 筆記本電腦。特別是，檢查出的空路可視化，和英國總理團隊筆記本 OpenCypher。

由 OpenCypher 處理的數據採用無序系列鍵/值映射的形式。優化，操作和擴大這些映射的主要方法是使用子句來執行諸如模式匹配，插入，更新和刪除鍵/值對的任務。

在 OpenCypher 中有幾個子句用於查找圖形中的數據模式，其`MATCH`中最常見的。 `MATCH`可讓您指定要在圖形中尋找的節點、關係和篩選器模式。例如：

- 取得所有節點

```
MATCH (n) RETURN n
```

- 尋找連接的節點

```
MATCH (n)-[r]->(d) RETURN n, r, d
```

- 尋找路徑

```
MATCH p=(n)-[r]->(d) RETURN p
```

- 獲取帶有標籤的所有節點

```
MATCH (n:airport) RETURN n
```

請注意，上面的第一個查詢返回圖中的每個節點，接下來的兩個返回每個節點有一個 relationship— 這通常不推薦！在幾乎所有情況下，您都希望縮小返回的數據，您可以通過指定節點或關係標籤和屬性來縮小數據的範圍，如第四個示例所示。

您可以在海王星 github 示例存儲庫中找到 OpenCypher 語法的便捷作弊表。

海王星開放密碼狀態伺服器和狀態端點

OpenCypher 狀態端點可讓您存取目前在伺服器上執行或等待執行之查詢的相關資訊。它也可以讓您取消這些查詢。端點是：

```
https://(the server):(the port number)/openCypher/status
```

您可以使用 HTTP GET 和POST方法從服務器獲取當前狀態，或者取消查詢。您也可以使用該DELETE方法來取消正在運行或等待中的查詢。

狀態請求的參數

狀態查詢參數

- **includeWaiting**(true或false) — 當設定為true且其他參數不存在時，會導致傳回等待查詢以及執行查詢的狀態資訊。
- **cancelQuery**— 僅與GET和POST方法搭配使用，以表示這是取消請求。該DELETE方法不需要此參數。

  不使用cancelQuery參數的值，但如果存在，cancelQuery則需要queryId參數，以識別要取消的查詢。
- **queryId**— 包含特定查詢的 ID。

  如果與GET或方POST法搭配使用，且cancelQuery參數不存在，則queryId會針對其識別的特定查詢傳回狀態資訊。如果存在cancelQuery參數，則會取消queryId識別的特定查詢。

  與該DELETE方法一起使用時，queryId始終指示要取消的特定查詢。
- **silent**— 僅在取消查詢時使用。如果設定為true，則會導致取消以無訊息方式發生。

狀態請求響應字段

狀態響應字段（如果未提供特定查詢的 ID）

- acceptedQueryCount— 已接受但尚未完成的查詢數目，包括佇列中的查詢。
- runningQueryCount— 當前正在運行的小鬼查詢的數量。
- 查詢 — 目前的小鬼查詢清單。

特定查詢的狀態響應字段

- 查詢 ID — 查詢的 GUID 識別碼。Neptune 會自動將此 ID 值指派給每個查詢，或者您也可以指派自己的 ID (請參閱將自定義 ID 注入海王星小鬼或 SPARQL 查詢 (p. 126))。
- 查詢字串 — 已提交的查詢。如果超過 1024 個字元即予截斷。
- queryEvalStats— 此查詢的統計資料：
  - 等候 — 指出查詢等待的時間長度 (以毫秒為單位)。
  - elapsed — 查詢到目前為止一直在執行的毫秒數。
  - 已取消 — True 表示查詢已取消或False尚未取消。

狀態請求和響應的例子

- 請求所有查詢的狀態，包括等待中的查詢：

```
curl https://server:port/openCypher/status \
  --data-urlencode "includeWaiting=true"
```

回應：

```
{
```

```
  "acceptedQueryCount" : 0,
  "runningQueryCount" : 0,
  "queries" : [ ]
}
```

- 請求正在運行的查詢的狀態，不包括那些等待中：

```
curl https://server:port/openCypher/status
```

回應：

```
{
  "acceptedQueryCount" : 0,
  "runningQueryCount" : 0,
  "queries" : [ ]
}
```

- 請求單個查詢的狀態：

```
curl https://server:port/openCypher/status \
 --data-urlencode "queryId=eadc6eea-698b-4a2f-8554-5270ab17ebee"
```

回應：

```
{
  "queryId" : "eadc6eea-698b-4a2f-8554-5270ab17ebee",
  "queryString" : "MATCH (n1)-[:knows]->(n2), (n2)-[:knows]->(n3), (n3)-[:knows]->(n4),
 (n4)-[:knows]->(n5), (n5)-[:knows]->(n6), (n6)-[:knows]->(n7), (n7)-[:knows]->(n8),
 (n8)-[:knows]->(n9), (n9)-[:knows]->(n10) RETURN COUNT(n1);",
  "queryEvalStats" : {
    "waited" : 0,
    "elapsed" : 23463,
    "cancelled" : false
  }
}
```

- 取消查詢的請求

1. 使用POST：

```
curl -X POST https://server:port/openCypher/status \
  --data-urlencode "cancelQuery" \
  --data-urlencode "queryId=f43ce17b-db01-4d37-a074-c76d1c26d7a9"
```

回應：

```
{
  "status" : "200 OK",
  "payload" : true
}
```

2. 使用GET：

```
curl -X GET https://server:port/openCypher/status \
  --data-urlencode "cancelQuery" \
  --data-urlencode "queryId=588af350-cfde-4222-bee6-b9cedc87180d"
```

回應：

```
{
  "status" : "200 OK",
  "payload" : true
}
```

3. 使用DELETE：

```
curl -X DELETE \
  -s "https://server:port/openCypher/status?queryId=b9a516d1-d25c-4301-bb80-10b2743ecf0e"
```

回應：

```
{
  "status" : "200 OK",
  "payload" : true
}
```

亞馬遜海王星開放密碼 HTTPS 端點

主題

- 在 HTTPS 端點上讀取和寫入查詢 (p. 386)
- 默認的開放密碼 JSON 結果格式 (p. 387)

在 HTTPS 端點上讀取和寫入查詢

OpenCypher HTTPS 端點支援使用和方法的讀取GET和更新查詢。POST不支援DELETE和PUT方法。

下列指示會引導您完成使用curl指令和 HTTPS 連線至 OpenCypher 端點的步驟。您必須遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

語法是：

```
HTTPS://(the server):(the port number)/openCypher
```

以下是示例讀取查詢，一個使用POST和使用GET：

1. 使用POST：

```
curl HTTPS://server:port/openCypher \
  -d "query=MATCH (n1) RETURN n1;"
```

2. 使用GET（查詢字符串是 URL 編碼）：

```
curl -X GET \
  "HTTPS://server:port/openCypher?query=MATCH%20(n1)%20RETURN%20n1"
```

以下是示例寫/更新查詢，一個使用POST和使用：GET

1. 使用POST：

```
curl HTTPS://server:port/openCypher \
```

```
  -d "query=CREATE (n:Person { age: 25 })"
```

2. 使用GET（查詢字符串是 URL 編碼）：

```
curl -X GET \
  "HTTPS://server:port/openCypher?query=CREATE%20(n%3APerson%20%7B%20age%3A%2025%20%7D)"
```

默認的開放密碼 JSON 結果格式

預設會傳回下列 JSON 格式，或將要求標頭明確設定為Accept: application/json。這種格式被設計為使用大多數庫的本地語言功能輕鬆解析為對象。

傳回的 JSON 文件包含一個欄位results，其中包含查詢傳回值。下面的例子顯示了常見值的 JSON 格式。

值響應示例：

```
{
  "results": [
    {
      "count(a)": 121
    }
  ]
}
```

節點響應示例：

```
{
  "results": [
    {
      "a": {
        "~id": "22",
        "~entityType": "node",
        "~labels": [
          "airport"
        ],
        "~properties": {
          "desc": "Seattle-Tacoma",
          "lon": -122.30899810791,
          "runways": 3,
          "type": "airport",
          "country": "US",
          "region": "US-WA",
          "lat": 47.4490013122559,
          "elev": 432,
          "city": "Seattle",
          "icao": "KSEA",
          "code": "SEA",
          "longest": 11901
        }
      }
    }
  ]
}
```

關係響應示例：

```
{
  "results": [
```

```
    {
      "r": {
        "~id": "7389",
        "~entityType": "relationship",
        "~start": "22",
        "~end": "151",
        "~type": "route",
        "~properties": {
          "dist": 956
        }
      }
    }
  ]
}
```

路徑響應示例：

```
{
  "results": [
    {
      "p": [
        {
          "~id": "22",
          "~entityType": "node",
          "~labels": [
            "airport"
          ],
          "~properties": {
            "desc": "Seattle-Tacoma",
            "lon": -122.30899810791,
            "runways": 3,
            "type": "airport",
            "country": "US",
            "region": "US-WA",
            "lat": 47.4490013122559,
            "elev": 432,
            "city": "Seattle",
            "icao": "KSEA",
            "code": "SEA",
            "longest": 11901
          }
        },
        {
          "~id": "7389",
          "~entityType": "relationship",
          "~start": "22",
          "~end": "151",
          "~type": "route",
          "~properties": {
            "dist": 956
          }
        },
        {
          "~id": "151",
          "~entityType": "node",
          "~labels": [
            "airport"
          ],
          "~properties": {
            "desc": "Ontario International Airport",
            "lon": -117.600997924805,
            "runways": 2,
            "type": "airport",
            "country": "US",
            "region": "US-CA",
```

```
            "lat": 34.0559997558594,
            "elev": 944,
            "city": "Ontario",
            "icao": "KONT",
            "code": "ONT",
            "longest": 12198
          }
        }
      ]
    }
  ]
}
```

使用博爾特協議向海王星進行開放密碼查詢

Bolt 是最初由 Neo4j 開發的面向語句的客戶端/服務器協議，並根據知識共享 3.0 署名許可許可。ShareAlike它是客戶驅動的，這意味著客戶端始終啟動消息交換。

要使用 Neo4j 的 Bolt 驅動程序連接到海王星，只需使用 `bolt` URI 方案將 URL 和端口號替換為集群端點即可。如果您有一個正在運行的海王星實例，請使用 read_write 端點。如果有多個執行個體正在執行，則建議使用兩個驅動程式，一個用於寫入器，另一個用於所有僅供讀取複本。如果您只有預設的兩個端點，read_write 和 read_only 驅動程式就足夠了，但如果您也有自訂端點，請考慮為每個端點建立驅動程式執行個體。

> **Note**
>
> 儘管螺栓規範指出，Bolt 可以使用 TCP 進行連接WebSockets，或者海王星僅支持 Bolt 的 TCP 連接。

海王星允許多達 1000 個並發螺栓連接。

如需使用 Bolt 驅動程式之各種語言的 OpenCypher 查詢範例，請參閱 Neo4j 的驅動程式與語言指南說明文件。

> **Important**
>
> 適用於 Python，.NET 和 Golang 的 Neo4j 螺栓驅動程序目前不支持AWS簽名 v4 身份驗證令牌的自動更新。JavaScript這意味著簽名過期後（通常在 5 分鐘內），驅動程序無法進行身份驗證，並且後續請求失敗。下面的 PythonJavaScript，.NET 和 Go 示例都受到此問題的影響。
> 有關更多信息，請參閱 Neo4j 的 Python 驅動程序問題 #834，Neo4j 的 .NET 問題 #664，Neo4j 的 JavaScript驅動程序問題 #993 和 Neo4j 的黃金驅動程序問題 #429。

使用帶有 Java 的螺栓連接到海王星

您可以從 Maven MVN 存儲庫下載要使用的任何版本的驅動程序，也可以將此依賴項添加到您的項目中：

```
<dependency>
  <groupId>org.neo4j.driver</groupId>
  <artifactId>neo4j-java-driver</artifactId>
  <version>4.3.3</version>
</dependency>
```

然後，若要使用這些 Bolt 驅動程式之一在 Java 中連線到 Neptune，請使用如下所示的程式碼為叢集中的主要/寫入器執行個體建立驅動程式執行個體：

```
import org.neo4j.driver.Driver;
import org.neo4j.driver.GraphDatabase;

final Driver driver =
```

```
  GraphDatabase.driver("bolt://(your cluster endpoint URL):(your cluster port)",
    AuthTokens.none(),
    Config.builder().withEncryption()
                    .withTrustStrategy(TrustStrategy.trustSystemCertificates())
                    .build());
```

如果您有一個或多個讀取器複本，則可以使用類似的代碼為它們創建驅動程序實例，如下所示：

```
final Driver read_only_driver =              // (without connection timeout)
  GraphDatabase.driver("bolt://(your cluster endpoint URL):(your cluster port)",
      Config.builder().withEncryption()
                      .withTrustStrategy(TrustStrategy.trustSystemCertificates())
                      .build());
```

或者，超時：

```
final Driver read_only_timeout_driver =      // (with connection timeout)
  GraphDatabase.driver("bolt://(your cluster endpoint URL):(your cluster port)",
    Config.builder().withConnectionTimeout(30, TimeUnit.SECONDS)
                    .withEncryption()
                    .withTrustStrategy(TrustStrategy.trustSystemCertificates())
                    .build());
```

如果您有自定義端點，則也可能值得為每個端點創建一個驅動程序實例。

使用螺栓的 Python 開放密碼查詢示例

以下是如何使用螺栓在 Python 中進行打開密碼查詢：

```
python -m pip install neo4j
```

```
from neo4j import GraphDatabase
uri = "bolt://(your cluster endpoint URL):(your cluster port)"
driver = GraphDatabase.driver(uri, auth=("username", "password"), encrypted=True)
```

請注意，會忽略auth參數。

使用螺栓的 .NET 開放密碼查詢示例

要使用螺栓在. NET 中的 OpenCypher 查詢，第一步是使用安裝 Neo4j 的驅動程序。NuHet要進行同步調用，請使用該.Simple版本，如下所示：

```
Install-Package Neo4j.Driver.Simple-4.3.0
```

```
using Neo4j.Driver;

namespace hello
{
  // This example creates a node and reads a node in a Neptune
  // Cluster where IAM Authentication is not enabled.
  public class HelloWorldExample : IDisposable
  {
    private bool _disposed = false;
    private readonly IDriver _driver;
    private static string url = "bolt://(your cluster endpoint URL):(your cluster port)";
    private static string createNodeQuery = "CREATE (a:Greeting) SET a.message =
 'HelloWorldExample'";
```

```
   private static string readNodeQuery = "MATCH(n:Greeting) RETURN n.message";

   ~HelloWorldExample() => Dispose(false);

   public HelloWorldExample(string uri)
   {
     _driver = GraphDatabase.Driver(uri, AuthTokens.None, o =>
o.WithEncryptionLevel(EncryptionLevel.Encrypted));
   }

   public void createNode()
   {
     // Open a session
     using (var session = _driver.Session())
     {
        // Run the query in a write transaction
       var greeting = session.WriteTransaction(tx =>
       {
         var result = tx.Run(createNodeQuery);
         // Consume the result
         return result.Consume();
       });

       // The output will look like this:
       //   ResultSummary{Query=`CREATE (a:Greeting) SET a.message =
'HelloWorldExample".....
       Console.WriteLine(greeting);
     }
   }

   public void retrieveNode()
   {
     // Open a session
     using (var session = _driver.Session())
     {
       // Run the query in a read transaction
       var greeting = session.ReadTransaction(tx =>
       {
         var result = tx.Run(readNodeQuery);
         // Consume the result. Read the single node
         // created in a previous step.
         return result.Single()[0].As<string>();
       });
       // The output will look like this:
       //   HelloWorldExample
       Console.WriteLine(greeting);
     }
   }

   public void Dispose()
   {
     Dispose(true);
     GC.SuppressFinalize(this);
   }

   protected virtual void Dispose(bool disposing)
   {
     if (_disposed)
       return;
     if (disposing)
     {
       _driver?.Dispose();
     }
     _disposed = true;
   }
```

```
    public static void Main()
    {
      using (var apiCaller = new HelloWorldExample(url))
      {
        apiCaller.createNode();
        apiCaller.retrieveNode();
      }
    }
  }
}
```

使用帶有 IAM 身份驗證的博爾特的 Java 開放密碼查詢示例

下面的 Java 代碼演示了如何使用帶有 IAM 身份驗證的博爾特在 Java 中進行開放密碼查詢。該JavaDoc評論描述了它的用法。一旦驅動程序實例可用，您可以使用它來發出多個經過身份驗證的請求。

```
package software.amazon.neptune.bolt;

import com.amazonaws.DefaultRequest;
import com.amazonaws.Request;
import com.amazonaws.auth.AWS4Signer;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.http.HttpMethodName;
import com.google.gson.Gson;
import lombok.Builder;
import lombok.Getter;
import lombok.NonNull;
import org.neo4j.driver.Value;
import org.neo4j.driver.Values;
import org.neo4j.driver.internal.security.InternalAuthToken;
import org.neo4j.driver.internal.value.StringValue;

import java.net.URI;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

import static com.amazonaws.auth.internal.SignerConstants.AUTHORIZATION;
import static com.amazonaws.auth.internal.SignerConstants.HOST;
import static com.amazonaws.auth.internal.SignerConstants.X_AMZ_DATE;
import static com.amazonaws.auth.internal.SignerConstants.X_AMZ_SECURITY_TOKEN;

/**
 * Use this class instead of `AuthTokens.basic` when working with an IAM
 * auth-enabled server. It works the same as `AuthTokens.basic` when using
 * static credentials, and avoids making requests with an expired signature
 * when using temporary credentials. Internally, it generates a new signature
 * on every invocation (this may change in a future implementation).
 *
 * Note that authentication happens only the first time for a pooled connection.
 *
 * Typical usage:
 *
 * NeptuneAuthToken authToken = NeptuneAuthToken.builder()
 *     .credentialsProvider(credentialsProvider)
 *     .region("aws region")
 *     .url("cluster endpoint url")
 *     .build();
 *
 * Driver driver = GraphDatabase.driver(
 *     authToken.getUrl(),
 *     authToken,
 *     config
 * );
```

```
 */

public class NeptuneAuthToken extends InternalAuthToken {
  private static final String SCHEME = "basic";
  private static final String REALM = "realm";
  private static final String SERVICE_NAME = "neptune-db";
  private static final String HTTP_METHOD_HDR = "HttpMethod";
  private static final String DUMMY_USERNAME = "username";
  @NonNull
  private final String region;
  @NonNull
  @Getter
  private final String url;
  @NonNull
  private final AWSCredentialsProvider credentialsProvider;
  private final Gson gson = new Gson();

  @Builder
  private NeptuneAuthToken(
      @NonNull final String region,
      @NonNull final String url,
      @NonNull final AWSCredentialsProvider credentialsProvider
  ) {
      // The superclass caches the result of toMap(), which we don't want
      super(Collections.emptyMap());
      this.region = region;
      this.url = url;
      this.credentialsProvider = credentialsProvider;
  }

  @Override
  public Map<String, Value> toMap() {
    final Map<String, Value> map = new HashMap<>();
    map.put(SCHEME_KEY, Values.value(SCHEME));
    map.put(PRINCIPAL_KEY, Values.value(DUMMY_USERNAME));
    map.put(CREDENTIALS_KEY, new StringValue(getSignedHeader()));
    map.put(REALM_KEY, Values.value(REALM));

    return map;
  }

  private String getSignedHeader() {
    final Request<Void> request = new DefaultRequest<>(SERVICE_NAME);
    request.setHttpMethod(HttpMethodName.GET);
    request.setEndpoint(URI.create(url));
    // Comment out the following line if you're using an engine version older than 1.2.0.0
    request.setResourcePath("/opencypher");

    final AWS4Signer signer = new AWS4Signer();
    signer.setRegionName(region);
    signer.setServiceName(request.getServiceName());
    signer.sign(request, credentialsProvider.getCredentials());

    return getAuthInfoJson(request);
  }

  private String getAuthInfoJson(final Request<Void> request) {
    final Map<String, Object> obj = new HashMap<>();
    obj.put(AUTHORIZATION, request.getHeaders().get(AUTHORIZATION));
    obj.put(HTTP_METHOD_HDR, request.getHttpMethod());
    obj.put(X_AMZ_DATE, request.getHeaders().get(X_AMZ_DATE));
    obj.put(HOST, request.getHeaders().get(HOST));
    obj.put(X_AMZ_SECURITY_TOKEN, request.getHeaders().get(X_AMZ_SECURITY_TOKEN));

    return gson.toJson(obj);
  }
```

```
}
```

使用帶有 IAM 身份驗證的博爾特的 Python 開啟密碼查詢示例

下面的 Python 類可以讓你使用帶有 IAM 身份驗證的博爾特在 Python 中進行打開密碼查詢：

```
import json

from neo4j import Auth
from botocore.awsrequest import AWSRequest
from botocore.credentials import Credentials
from botocore.auth import (
  SigV4Auth,
  _host_from_url,
)

SCHEME = "basic"
REALM = "realm"
SERVICE_NAME = "neptune-db"
DUMMY_USERNAME = "username"
HTTP_METHOD_HDR = "HttpMethod"
HTTP_METHOD = "GET"
AUTHORIZATION = "Authorization"
X_AMZ_DATE = "X-Amz-Date"
X_AMZ_SECURITY_TOKEN = "X-Amz-Security-Token"
HOST = "Host"


class NeptuneAuthToken(Auth):
  def __init__(
    self,
    credentials: Credentials,
    region: str,
    url: str,
    **parameters
  ):
    # Do NOT add "/opencypher" in the line below if you're using an engine version older
 than 1.2.0.0
    request = AWSRequest(method=HTTP_METHOD, url=url + "/opencypher")
    request.headers.add_header("Host", _host_from_url(request.url))
    sigv4 = SigV4Auth(credentials, SERVICE_NAME, region)
    sigv4.add_auth(request)

    auth_obj = {
      hdr: request.headers[hdr]
      for hdr in [AUTHORIZATION, X_AMZ_DATE, X_AMZ_SECURITY_TOKEN, HOST]
    }
    auth_obj[HTTP_METHOD_HDR] = request.method
    creds: str = json.dumps(auth_obj)
    super().__init__(SCHEME, DUMMY_USERNAME, creds, REALM, **parameters)
```

您可以使用這個類來創建一個驅動程序，如下所示：

```
  authToken = NeptuneAuthToken(creds, REGION, URL)
  driver = GraphDatabase.driver(URL, auth=authToken, encrypted=True)
```

使用 IAM 身份驗證和螺栓的 Node.js 示例

下面的 Node.js 代碼使用JavaScript版本 3 和 ES6 語法的 AWS SDK 來創建一個驗證請求的驅動程序：

```
import neo4j from "neo4j-driver";
```

```
import { HttpRequest }  from "@aws-sdk/protocol-http";
import { defaultProvider } from "@aws-sdk/credential-provider-node";
import { SignatureV4 } from "@aws-sdk/signature-v4";
import crypto from "@aws-crypto/sha256-js";
const { Sha256 } = crypto;
import assert from "node:assert";

const region = "us-west-2";
const serviceName = "neptune-db";
const host = "(your cluster endpoint URL)";
const port = 8182;
const protocol = "bolt";
const hostPort = host + ":" + port;
const url = protocol + "://" + hostPort;
const createQuery = "CREATE (n:Greeting {message: 'Hello'}) RETURN ID(n)";
const readQuery = "MATCH(n:Greeting) WHERE ID(n) = $id RETURN n.message";

async function signedHeader() {
  const req = new HttpRequest({
    method: "GET",
    protocol: protocol,
    hostname: host,
    port: port,
    // Comment out the following line if you're using an engine version older than 1.2.0.0
    path: "/opencypher",
    headers: {
      host: hostPort
    }
  });

  const signer = new SignatureV4({
    credentials: defaultProvider(),
    region: region,
    service: serviceName,
    sha256: Sha256
  });

  return signer.sign(req, { unsignableHeaders: new Set(["x-amz-content-sha256"]) })
    .then((signedRequest) => {
      const authInfo = {
        "Authorization": signedRequest.headers["authorization"],
        "HttpMethod": signedRequest.method,
        "X-Amz-Date": signedRequest.headers["x-amz-date"],
        "Host": signedRequest.headers["host"],
        "X-Amz-Security-Token": signedRequest.headers["x-amz-security-token"]
      };
      return JSON.stringify(authInfo);
    });
}

async function createDriver() {
  let authToken = { scheme: "basic", realm: "realm", principal: "username", credentials:
 await signedHeader() };

  return neo4j.driver(url, authToken, {
      encrypted: "ENCRYPTION_ON",
      trust: "TRUST_SYSTEM_CA_SIGNED_CERTIFICATES",
      maxConnectionPoolSize: 1,
      // logging: neo4j.logging.console("debug")
    }
  );
}

function unmanagedTxn(driver) {
  const session = driver.session();
  const tx = session.beginTransaction();
```

```
  tx.run(createQuery)
  .then((res) => {
    const id = res.records[0].get(0);
    return tx.run(readQuery, { id: id });
  })
  .then((res) => {
    // All good, the transaction will be committed
    const msg = res.records[0].get("n.message");
    assert.equal(msg, "Hello");
  })
  .catch(err => {
    // The transaction will be rolled back, now handle the error.
    console.log(err);
  })
  .then(() => session.close());
}

createDriver()
.then((driver) => {
  unmanagedTxn(driver);
  driver.close();
})
.catch((err) => {
  console.log(err);
});
```

使用帶有 IAM 身份驗證的博爾特的 .NET 開密查詢示例

要在 .NET 中啟用 IAM 身份驗證，您需要在建立連接時簽署請求。下面的示例演示了如何創建一個NeptuneAuthToken幫助程序來生成一個身份驗證令牌：

```
using Amazon.Runtime;
using Amazon.Util;
using Neo4j.Driver;
using System.Security.Cryptography;
using System.Text;
using System.Text.Json;
using System.Web;

namespace Hello
{
  /*
   * Use this class instead of `AuthTokens.None` when working with an IAM-auth-enabled
 server.
   *
   * Note that authentication happens only the first time for a pooled connection.
   *
   * Typical usage:
   *
   * var authToken = new NeptuneAuthToken(AccessKey, SecretKey, Region).GetAuthToken(Host);
   * _driver = GraphDatabase.Driver(Url, authToken, o =>
 o.WithEncryptionLevel(EncryptionLevel.Encrypted));
   */

  public class NeptuneAuthToken
  {
    private const string ServiceName = "neptune-db";
    private const string Scheme = "basic";
    private const string Realm = "realm";
    private const string DummyUserName = "username";
    private const string Algorithm = "AWS4-HMAC-SHA256";
    private const string AWSRequest = "aws4_request";

    private readonly string _accessKey;
```

```
    private readonly string _secretKey;
    private readonly string _region;

    private readonly string _emptyPayloadHash;

    private readonly SHA256 _sha256;


    public NeptuneAuthToken(string awsKey = null, string secretKey = null, string region =
 null)
    {
      var awsCredentials = awsKey == null || secretKey == null
        ? FallbackCredentialsFactory.GetCredentials().GetCredentials()
        : null;

      _accessKey = awsKey ?? awsCredentials.AccessKey;
      _secretKey = secretKey ?? awsCredentials.SecretKey;
      _region = region ?? FallbackRegionFactory.GetRegionEndpoint().SystemName; //ex: us-
east-1

      _sha256 = SHA256.Create();
      _emptyPayloadHash = Hash(Array.Empty<byte>());
    }

    public IAuthToken GetAuthToken(string url)
    {
      return AuthTokens.Custom(DummyUserName, GetCredentials(url), Realm, Scheme);
    }

    /******************* AWS SIGNING FUNCTIONS ********************/
    private string Hash(byte[] bytesToHash)
    {
      return ToHexString(_sha256.ComputeHash(bytesToHash));
    }

    private static byte[] HmacSHA256(byte[] key, string data)
    {
      return new HMACSHA256(key).ComputeHash(Encoding.UTF8.GetBytes(data));
    }

    private byte[] GetSignatureKey(string dateStamp)
    {
      var kSecret = Encoding.UTF8.GetBytes($"AWS4{_secretKey}");
      var kDate = HmacSHA256(kSecret, dateStamp);
      var kRegion = HmacSHA256(kDate, _region);
      var kService = HmacSHA256(kRegion, ServiceName);
      return HmacSHA256(kService, AWSRequest);
    }

    private static string ToHexString(byte[] array)
    {
      return Convert.ToHexString(array).ToLowerInvariant();
    }

    private string GetCredentials(string url)
    {
      var request = new HttpRequestMessage
      {
        Method = HttpMethod.Get,
        RequestUri = new Uri($"https://{url}/opencypher")
      };

      var signedrequest = Sign(request);

      var headers = new Dictionary<string, object>
      {
```

```
      [HeaderKeys.AuthorizationHeader] =
signedrequest.Headers.GetValues(HeaderKeys.AuthorizationHeader).FirstOrDefault(),
      ["HttpMethod"] = HttpMethod.Get.ToString(),
      [HeaderKeys.XAmzDateHeader] =
signedrequest.Headers.GetValues(HeaderKeys.XAmzDateHeader).FirstOrDefault(),
      // Host should be capitalized, not like in Amazon.Util.HeaderKeys.HostHeader
      ["Host"] = signedrequest.Headers.GetValues(HeaderKeys.HostHeader).FirstOrDefault(),
    };

    return JsonSerializer.Serialize(headers);
  }

  private HttpRequestMessage Sign(HttpRequestMessage request)
  {
    var now = DateTimeOffset.UtcNow;
    var amzdate = now.ToString("yyyyMMddTHHmmssZ");
    var datestamp = now.ToString("yyyyMMdd");

    if (request.Headers.Host == null)
    {
      request.Headers.Host = $"{request.RequestUri.Host}:{request.RequestUri.Port}";
    }

    request.Headers.Add(HeaderKeys.XAmzDateHeader, amzdate);

    var canonicalQueryParams = GetCanonicalQueryParams(request);

    var canonicalRequest = new StringBuilder();
    canonicalRequest.Append(request.Method + "\n");
    canonicalRequest.Append(request.RequestUri.AbsolutePath + "\n");
    canonicalRequest.Append(canonicalQueryParams + "\n");

    var signedHeadersList = new List<string>();
    foreach (var header in request.Headers.OrderBy(a => a.Key.ToLowerInvariant()))
    {
      canonicalRequest.Append(header.Key.ToLowerInvariant());
      canonicalRequest.Append(':');
      canonicalRequest.Append(string.Join(",", header.Value.Select(s => s.Trim())));
      canonicalRequest.Append('\n');
      signedHeadersList.Add(header.Key.ToLowerInvariant());
    }
    canonicalRequest.Append('\n');

    var signedHeaders = string.Join(";", signedHeadersList);
    canonicalRequest.Append(signedHeaders + "\n");
    canonicalRequest.Append(_emptyPayloadHash);

    var credentialScope = $"{datestamp}/{_region}/{ServiceName}/{AWSRequest}";
    var stringToSign = $"{Algorithm}\n{amzdate}\n{credentialScope}\n"
      + Hash(Encoding.UTF8.GetBytes(canonicalRequest.ToString()));

    var signing_key = GetSignatureKey(datestamp);
    var signature = ToHexString(HmacSHA256(signing_key, stringToSign));

    request.Headers.TryAddWithoutValidation(HeaderKeys.AuthorizationHeader,
      $"{Algorithm} Credential={_accessKey}/{credentialScope},
SignedHeaders={signedHeaders}, Signature={signature}");

    return request;
  }

  private static string GetCanonicalQueryParams(HttpRequestMessage request)
  {
    var querystring = HttpUtility.ParseQueryString(request.RequestUri.Query);

    // Query params must be escaped in upper case (i.e. "%2C", not "%2c").
```

```
      var queryParams = querystring.AllKeys.OrderBy(a => a)
        .Select(key => $"{key}={Uri.EscapeDataString(querystring[key])}");
      return string.Join("&", queryParams);
    }
  }
}
```

以下是如何使用帶有 IAM 身份驗證的博爾特在 .NET 中進行 OpenCypher 查詢。下面的例子使用NeptuneAuthToken幫手：

```
using Neo4j.Driver;

namespace Hello
{
  public class HelloWorldExample
  {
    private const string Host = "(your hostname):8182";
    private const string Url = $"bolt://{Host}";
    private const string CreateNodeQuery = "CREATE (a:Greeting) SET a.message =
 'HelloWorldExample'";
    private const string ReadNodeQuery = "MATCH(n:Greeting) RETURN n.message";

    private const string AccessKey = "(your access key)";
    private const string SecretKey = "(your secret key)";
    private const string Region = "(your AWS region)"; // e.g. "us-west-2"

    private readonly IDriver _driver;

    public HelloWorldExample()
    {
      var authToken = new NeptuneAuthToken(AccessKey, SecretKey,
 Region).GetAuthToken(Host);

      // Note that when the connection is reinitialized after max connection lifetime
      // has been reached, the signature token could have already been expired (usually 5
 min)
      // You can face exceptions like:
      //   `Unexpected server exception 'Signature expired: XXXX is now earlier than YYYY
 (ZZZZ - 5 min.)`
      _driver = GraphDatabase.Driver(Url, authToken, o =>

 o.WithMaxConnectionLifetime(TimeSpan.FromMinutes(60)).WithEncryptionLevel(EncryptionLevel.Encrypted));
    }

    public async Task CreateNode()
    {
      // Open a session
      using (var session = _driver.AsyncSession())
      {
        // Run the query in a write transaction
        var greeting = await session.WriteTransactionAsync(async tx =>
        {
          var result = await tx.RunAsync(CreateNodeQuery);
          // Consume the result
          return await result.ConsumeAsync();
        });

        // The output will look like this:
        //   ResultSummary{Query=`CREATE (a:Greeting) SET a.message =
 'HelloWorldExample".....
        Console.WriteLine(greeting.Query);
      }
    }
```

```
    public async Task RetrieveNode()
    {
      // Open a session
      using (var session = _driver.AsyncSession())
      {
        // Run the query in a read transaction
        var greeting = await session.ReadTransactionAsync(async tx =>
        {
          var result = await tx.RunAsync(ReadNodeQuery);
          var records = await result.ToListAsync();

          // Consume the result. Read the single node
          // created in a previous step.
          return records[0].Values.First().Value;
        });
        // The output will look like this:
        //   HelloWorldExample
        Console.WriteLine(greeting);
      }
    }
  }
}
```

這個例子可以通過運行下面的代碼.NET 6或.NET 7使用以下軟件包來啟動：

- **Neo4j**.Driver=4.3.0
- **AWSSDK**.Core=3.7.102.1

```
namespace Hello
{
  class Program
  {
    static async Task Main()
    {
      var apiCaller = new HelloWorldExample();

      await apiCaller.CreateNode();
      await apiCaller.RetrieveNode();
    }
  }
}
```

使用帶有 IAM 身份驗證的博爾特的高朗開放式密碼查詢示例

下面的 Golang 軟件包顯示瞭如何使用帶 IAM 身份驗證的 Bolt 在 Go 語言中進行 OpenCypher 查詢：

```
package main

import (
  "context"
  "encoding/json"
  "fmt"
  "github.com/aws/aws-sdk-go/aws/credentials"
  "github.com/aws/aws-sdk-go/aws/signer/v4"
  "github.com/neo4j/neo4j-go-driver/v5/neo4j"
  "log"
  "net/http"
  "os"
  "time"
)
```

```
const (
  ServiceName   = "neptune-db"
  DummyUsername = "username"
)

// Find node by id using Go driver
func findNode(ctx context.Context, region string, hostAndPort string, nodeId string)
 (string, error) {
  req, err := http.NewRequest(http.MethodGet, "https://"+hostAndPort+"/opencypher", nil)

  if err != nil {
    return "", fmt.Errorf("error creating request, %v", err)
  }

  // credentials must have been exported as environment variables
  signer := v4.NewSigner(credentials.NewEnvCredentials())
  _, err = signer.Sign(req, nil, ServiceName, region, time.Now())

  if err != nil {
    return "", fmt.Errorf("error signing request: %v", err)
  }

  hdrs := []string{"Authorization", "X-Amz-Date", "X-Amz-Security-Token"}
  hdrMap := make(map[string]string)
  for _, h := range hdrs {
    hdrMap[h] = req.Header.Get(h)
  }

  hdrMap["Host"] = req.Host
  hdrMap["HttpMethod"] = req.Method

  password, err := json.Marshal(hdrMap)
  if err != nil {
    return "", fmt.Errorf("error creating JSON, %v", err)
  }
  authToken := neo4j.BasicAuth(DummyUsername, string(password), "")
  // +s enables encryption with a full certificate check
  // Use +ssc to disable client side TLS verification
  driver, err := neo4j.NewDriverWithContext("bolt+s://"+hostAndPort+"/opencypher",
 authToken)
  if err != nil {
    return "", fmt.Errorf("error creating driver, %v", err)
  }

  defer driver.Close(ctx)

  if err := driver.VerifyConnectivity(ctx); err != nil {
    log.Fatalf("failed to verify connection, %v", err)
  }

  config := neo4j.SessionConfig{}

  session := driver.NewSession(ctx, config)
  defer session.Close(ctx)

  result, err := session.Run(
    ctx,
    fmt.Sprintf("MATCH (n) WHERE ID(n) = '%s' RETURN n", nodeId),
    map[string]any{},
  )
  if err != nil {
    return "", fmt.Errorf("error running query, %v", err)
  }

  if !result.Next(ctx) {
```

```
    return "", fmt.Errorf("node not found")
  }

  n, found := result.Record().Get("n")
  if !found {
    return "", fmt.Errorf("node not found")
  }

  return fmt.Sprintf("+%v\n", n), nil
}

func main() {
  if len(os.Args) < 3 {
    log.Fatal("Usage: go main.go (region) (host and port)")
  }
  region := os.Args[1]
  hostAndPort := os.Args[2]
  ctx := context.Background()

  res, err := findNode(ctx, region, hostAndPort, "72c2e8c1-7d5f-5f30-10ca-9d2bb8c4afbc")
  if err != nil {
    log.Fatal(err)
  }
  fmt.Println(res)
}
```

海王星中的螺栓連接行為

以下是有關海王星 Bolt 連接的一些注意事項：

- 因為 Bolt 連線是在 TCP 層建立的，因此您無法像使用 HTTP 端點一樣，在它們前面使用應用程式負載平衡器。
- Neptune 用於 Bolt 連線的連接埠是資料庫叢集的連接埠。
- 海王星伺服器會根據傳遞給它的 Bolt 序頭訊號，選擇最高的 Bolt 版本 (1、2、3 或 4.0)。
- 用戶端可以在任何時間點開啟的 Neptune 伺服器的最大連線數目為 1,000。
- 如果客戶端在查詢後沒有關閉連接，則該連接可用於執行下一個查詢。
- 但是，如果連線閒置 20 分鐘，伺服器會自動關閉連線。
- 如果未啟用 IAM 身份驗證，您可以使用`AuthTokens.none()`而不是提供虛擬使用者名稱和密碼。例如，在 Java 中：

```
GraphDatabase.driver("bolt://(your cluster endpoint URL):(your cluster port)",
 AuthTokens.none(),

 Config.builder().withEncryption().withTrustStrategy(TrustStrategy.trustSystemCertificates()).build()
```

- 啟用 IAM 身份驗證時，如果 Bolt 連接因其他原因尚未關閉，則 Bolt 連接始終會在建立 10 天之後的幾分鐘內中斷連接。
- 如果用戶端傳送查詢以透過連線執行，而不耗用先前查詢的結果，則會捨棄新查詢。若要捨棄先前的結果，用戶端必須透過連線傳送重設訊息。
- 一次只能在給定的連接上創建一個事務。
- 如果在交易期間發生例外狀況，Neptune 伺服器會復原交易並關閉連線。在這種情況下，驅動程序為下一個查詢創建一個新的連接。
- 請注意，工作階段不是執行緒安全的。多個平行作業必須使用多個獨立工作階段。

開放密碼參數化查詢的例子

海王星支持參數化的開放密碼查詢。這可以讓您多次使用相同的查詢結構與不同的參數。由於查詢結構不會改變，Neptune 可以緩存其抽象語法樹（AST），而不必多次解析它。

使用 HTTPS 端點的開放式參數化查詢範例

以下是使用參數化查詢搭配海王星 OpenCypher HTTPS 端點的範例。該查詢是：

```
MATCH (n {name: $name, age: $age})
RETURN n
```

參數定義如下：

```
parameters={"name": "john", "age": 20}
```

使用GET，您可以像這樣提交參數化查詢：

```
curl -k \
  "https://localhost:8182/openCypher?query=MATCH%20%28n%20%7Bname:\$name,age:\$age%7D
%29%20RETURN%20n&parameters=%7B%22name%22:%22john%22,%22age%22:20%7D"
```

或者，您可以使用POST：

```
curl -k \
  https://localhost:8182/openCypher \
  -d "query=MATCH (n {name: \$name, age: \$age}) RETURN n" \
  -d "parameters={\"name\": \"john\", \"age\": 20}"
```

或者，使用DIRECT POST：

```
curl -k \
   -H "Content-Type: application/opencypher" \
  "https://localhost:8182/openCypher?parameters=%7B%22name%22:%22john%22,%22age%22:20%7D" \
  -d "MATCH (n {name: \$name, age: \$age}) RETURN n"
```

使用螺栓的開放密碼參數化查詢的例子

以下是使用螺栓協議的 OpenCypher 參數化查詢的 Python 示例：

```
from neo4j import GraphDatabase
uri = "bolt://[neptune-endpoint-url]:8182"
driver = GraphDatabase.driver(uri, auth=("", ""))

def match_name_and_age(tx, name, age):
  # Parameterized Query
  tx.run("MATCH (n {name: $name, age: $age}) RETURN n", name=name, age=age)

with driver.session() as session:
  # Parameters
  session.read_transaction(match_name_and_age, "john", 20)

driver.close()
```

以下是使用螺栓協議的 OpenCypher 參數化查詢的 Java 示例：

```
Driver driver = GraphDatabase.driver("bolt+s://(your cluster endpoint URL):8182");
HashMap<String, Object> parameters = new HashMap<>();
parameters.put("name", "john");
parameters.put("age", 20);
String queryString = "MATCH (n {name: $name, age: $age}) RETURN n";
Result result = driver.session().run(queryString, parameters);
```

开放密码数据模型

海王星 OpenCypher 引擎建立在與小精靈相同的屬性圖模型上。尤其是：

- 每個節點都有一個或多個標籤。如果插入沒有標示的節點，則會貼附名vertex為的預設標示。如果您嘗試刪除節點的所有標籤，則會擲回錯誤。
- 關聯是只有一種關係類型的實體，而且會在兩個節點之間形成單向連接 (也就是，從其中一個節點到另一個節點)。
- 節點和關係都可以具有屬性，但不必。海王星支持節點和零屬性的關係。
- 海王星不支持元屬性，這些元屬性也不包括在 OpenCypher 規範中。
- 如果圖形中的屬性是使用 Gremlin 創建的，則它們可以是多值的。這是一個節點或關係屬性可以有一組不同的值，而不是只有一個。海王星已經擴展了 OpenCypher 語義，以優雅地處理多值屬性。

支援的資料類型記錄在中开放密码数据格式 (p. 267)。但是，我們目前不建議將Array屬性值插入 OpenCypher 圖形中。雖然可以使用大量載入器插入陣列屬性值，但目前的 Neptune OpenCypher 發行版本會將它視為一組多值屬性，而非單一清單值值。

以下是此版本支持的數據類型列表：

- Bool
- Byte
- Short
- Int
- Long
- Float (包括加號和減去無窮大和 NaN，但不包括 INF)
- Double (包括加號和減去無窮大和 NaN，但不包括 INF)
- DateTime
- String

开放密码explain功能

OpenCypher explain 功能是亞馬遜海王星中的自助服務工具，可幫助您了解海王星引擎採取的執行方法。若要叫用說明，您可以將參數傳遞至 OpenCypher HTTPS (p. 386) 要求explain=*mode*，其中的*mode*值可以是下列其中一項：

- **static**— 在static模式中，僅explain列印查詢計畫的靜態結構。它實際上並不會執行查詢。
- **dynamic**— 在dynamic模式中，explain也會執行查詢，並包含查詢計畫的動態層面。這些可能包括流經操作符的中間綁定的數量，傳入綁定到傳出綁定的比率，以及每個運算符所花費的總時間。
- **details**— 在details模式中，會explain列印動態模式中顯示的資訊以及其他詳細資訊，例如實際 OpenCypher 查詢字串和連接運算子底層模式的估計範圍計數。

例如，使用POST：

```
curl HTTPS://server:port/openCypher \
  -d "query=MATCH (n) RETURN n LIMIT 1;" \
  -d "explain=dynamic"
```

或者，使用GET：

```
curl -X GET \
  "HTTPS://server:port/openCypher?query=MATCH%20(n)%20RETURN%20n%20LIMIT
%201&explain=dynamic"
```

海王星中開放密碼explain的局限性

當前版本的開放密碼解釋有以下限制：

- 說明計劃目前僅適用於執行唯讀作業的查詢。不支援執行任何類型變異的查詢 CREATE DELETE MERGESET，例如、、等。
- 特定計劃的運算子和輸出可能會在未來的版本中變更。

DFE 運算子在開放explain密碼器輸出

若要使用 OpenCypher explain 功能提供的資訊，您必須瞭解 DFE 查詢引擎運作方式的一些詳細資料 (DFE (p. 129)) 是海王星用來處理 OpenCypher 查詢的引擎)。

DFE 引擎會將每個查詢轉譯成運算子的管線。從第一個運算子開始，中間解會透過此運算子管線從一個運算子流向下一個運算子。解釋表中的每一行代表一個結果，直到評估點為止。

DFE 查詢計畫中可顯示的運算子如下：

DFEApply — 針對儲存在指定變數functor中的值，執行引數區段中指定的函數

DFE BindRelation-將具有指定名稱的變量綁定在一起

DFE ChunkLocalSubQuery-這是一個非阻塞操作，充當正在執行的子查詢的包裝。

DFE DistinctColumn — 根據指定的變數傳回輸入值的不同子集。

DFE DistinctRelation — 根據指定的變數傳回輸入解法的不同子集。

DFeDrain — 顯示在子查詢結尾處，做為該子查詢的終止步驟。解決方案的數目會記錄為Units In。Units Out總是為零。

DFE LoopSubQuery-這是一個非阻塞操作，充當一個子查詢允許它被重複運行以在循環中使用的包裝。

DFE OptionalJoin — 執行可選的接合A OPTIONAL B # (A JOIN B) UNION (A MINUS_NE B)。這是封鎖作業。

DFE PipelineJoin — 針對引數定義的元組模式加入輸入。pattern

DFE PipelineScan — 掃描資料庫中的指定pattern引數，無論是否有欄上的指定篩選器。

DFEProject — 接受多個輸入資料行和專案只需要想要的資料行。

DFEreduce — 對指定的變數執行指定的彙總函數。

DFE RelationalJoin — 使用合併聯結，根據指定的模式索引鍵，連接上一個運算子的輸入。這是封鎖作業。

DFE RelationalJoin — 使用合併聯結，根據指定的模式索引鍵，連接上一個運算子的輸入。這是封鎖作業。

DFEubQuery — 此運算子會出現在所有計畫的開頭，並封裝在 DFE 引擎 (p. 129)上執行的計畫部分，也就是 OpenCypher 的整個計畫。

DFE SymmetricHashJoin — 使用雜湊聯結，根據指定的模式索引鍵，連接上一個運算子的輸入。這是一個非阻塞操作。

DFETee-這是一個分支運算符，它將相同的一組解決方案發送給多個運營商。

SolutionInjection— 顯示在explain輸出中的所有其他項目之前，Units Out欄中的值為 1。但是，它提供無操作，並且實際上並沒有向 DFE 引擎注入任何解決方案。

TermResolution— 出現在計劃的結束和對象從海王星引擎翻譯成 OpenCypher 對象.

密碼輸出中explain的列

海王星產生為 OpenCypher 說明輸出的查詢計畫資訊包含每列有一個運算子的資料表。此表格包含下列欄：

ID — 此運算子在計劃中的數字 ID。

輸出 #1 (和 Out #2) — 從此運算子下游運算子的 ID。最多可以有兩個下游運營商。

名稱 — 此運算子的名稱。

引數 — 運算子的任何相關詳細資訊。這包括輸入結構描述、輸出結構描述、模式 (for PipelineScan和PipelineJoin) 等等。

模式 — 描述運算子基本行為的標籤。此列大多為空白 (-) 。一個例外是TermResolution，其中模式可以是id2value_opencypher，表示從 ID 到 OpenCypher 值的分辨率。

單位輸入 — 作為輸入傳遞給此運算子的解數。沒有上游運算子 (例如DFEPipelineScan、SolutionInjections、以及沒DFESubquery有插入靜態值的運算子) 將具有零值。

單位輸出 — 產生為此運算子輸出的解決方案數目。 DFEDrain是一種特殊情況，其中被排出的解決方案的數量記錄在中，Units In並且始終Units Out為零。

比率 — 與的Units Out比率Units In。

時間 (毫秒) — 此運算子使用的 CPU 時間，以毫秒為單位。

開放密碼的一個基本例子解釋輸出

下面是開密explain輸出的一個基本的例子。查詢是空中航線資料集中具有機場代碼的節點的單節點查詢，ATL該節點會以預設 ASCII 輸出格式explain使用details模式呼叫：

```
curl -d "query=MATCH (n {code: 'ATL'}) RETURN n" -k https://localhost:8182/openCypher -d
 "explain=details"
                            ~
Query:
MATCH (n {code: 'ATL'}) RETURN n

##################################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments          # Mode                #
 Units In # Units Out # Ratio # Time (ms) #
##################################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]     # -                   # 0
      # 1         # 0.00  # 0         #
```

```
##################################################################################################################
# 1  # 2      # -      # DFESubquery       # subQuery=subQuery1 # -                    # 0
      # 1         # 0.00  # 4.00      #
##################################################################################################################
# 2  # -      # -      # TermResolution    # vars=[?n]          # id2value_opencypher # 1
      # 1         # 1.00  # 2.00      #
##################################################################################################################


subQuery1
##################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                                                                  # Mode # Units In #
 Units Out # Ratio # Time (ms) #
##################################################################################################################
# 0  # 1      # -      # DFEPipelineScan       # pattern=Node(?n) with property 'code' as ?
n_code2 and label 'ALL'                                           # -    # 0        # 1
      # 0.00  # 0.21      #
#    #        #        #                       # inlineFilters=[(?n_code2 IN
 ["ATL"^^xsd:string])]                                                          #      #
         #           #       #           #
#    #        #        #                       # patternEstimate=1
                                                                  #      #          #
      #       #           #
##################################################################################################################
# 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#9d84f97c-c3b0-459a-98d5-955a8726b159/graph_1 # -    # 1        # 1
         # 1.00  # 0.04      #
##################################################################################################################
# 2  # 3      # -      # DFEProject            # columns=[?n]
                                                                  # -    # 1        # 1
      # 1.00  # 0.04      #
##################################################################################################################
# 3  # -      # -      # DFEDrain              # -
                                                                  # -    # 1        # 0
      # 0.00  # 0.03      #
##################################################################################################################


subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#9d84f97c-
c3b0-459a-98d5-955a8726b159/graph_1
##################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                # Mode # Units In # Units Out # Ratio # Time (ms) #
##################################################################################################################
# 0  # 1      # -      # DFESolutionInjection # outSchema=[?n, ?n_code2]
                # -    # 0        # 1         # 0.00  # 0.02      #
##################################################################################################################
# 1  # 2      # 3      # DFETee               # -
                # -    # 1        # 2         # 2.00  # 0.02      #
##################################################################################################################
# 2  # 4      # -      # DFEDistinctColumn    # column=?n
                # -    # 1        # 1         # 1.00  # 0.20      #
#    #        #        #                      # ordered=false
                #      #        #           #       #           #
##################################################################################################################
# 3  # 5      # -      # DFEHashIndexBuild    # vars=[?n]
                # -    # 1        # 1         # 1.00  # 0.04      #
##################################################################################################################
# 4  # 5      # -      # DFEPipelineJoin      # pattern=Node(?n) with property 'ALL' and
 label '?n_label1' # -    # 1        # 1         # 1.00  # 0.25      #
#    #        #        #                      # patternEstimate=3506
                #      #        #           #       #           #
##################################################################################################################
# 5  # 6      # 7      # DFESync              # -
                # -    # 2        # 2         # 1.00  # 0.02      #
```

```
##########################################################################################################
# 6  # 8      # -      # DFEForwardValue      # -
               # -    # 1        # 1         # 1.00  # 0.01      #
##########################################################################################################
# 7  # 8      # -      # DFEForwardValue      # -
               # -    # 1        # 1         # 1.00  # 0.01      #
##########################################################################################################
# 8  # 9      # -      # DFEHashIndexJoin     # -
               # -    # 2        # 1         # 0.50  # 0.35      #
##########################################################################################################
# 9  # -      # -      # DFEDrain             # -
               # -    # 1        # 0         # 0.00  # 0.02      #
##########################################################################################################
```

在頂層，出SolutionInjection現在其他所有項目之前，有 1 個單位。請注意，它實際上並沒有注入任何解決方案。您可以看到下一個運算子DFESubquery中有 0 個單位。

之後SolutionInjection在頂層是DFESubquery和TermResolution運算符。 DFESubquery封裝要推送至 DFE 引擎 (p. 129)的查詢執行計畫的部分 (對於 OpenCypher 查詢，整個查詢計畫會由 DFE 執行)。查詢計畫中的所有運算子都會嵌套在subQuery1所參考的內部DFESubquery。唯一的例外是TermResolution，它將內部 ID 實現為完全序列化的 OpenCypher 對象。

向下推送至 DFE 引擎的所有運算子都具有以DFE前置字元開頭的名稱。如上所述，整個 OpenCypher 查詢計劃是由 DFE 執行的，所以作為一個結果，除了最終的TermResolution操作符開始與所有運營商。DFE

在內部subQuery1，可以有零個或多個DFEChunkLocalSubQuery或DFELoopSubQuery運算子，封裝在有限記憶體的機制中執行的推送執行計畫的一部分。 DFEChunkLocalSubQuery這裡包SolutionInjection含一個用作子查詢的輸入。若要在輸出中尋找該子查詢的資料表，請搜尋DFEChunkLocalSubQuery或DFELoopSubQuery運算子Arguments欄中subQuery=*graph URI*指定的資料表。

在中subQuery1，如果DFEPipelineScan使用 ID 0，則會掃描資料庫中指定的pattern。該模式掃描具有在所有標籤上code保存為變量的屬性的實體（您可以通?n_code2過附加airport到特定標籤上進行過濾n:airport）。引inlineFilters數顯示等ATL於code屬性的篩選。

接下來，DFEChunkLocalSubQuery運算子加入包含DFEPipelineJoin的子查詢的中間結果。這可以確保實際上?n是一個節點，因為先前DFEPipelineScan掃描具有該code屬性的任何實體。

具有限制之關係查詢的explain輸出範例

此查詢查找具有類型的兩個匿名節點之間的關係route，並返回最多 10 個。同樣，explain模式是details，輸出格式是默認的 ASCII 格式。這是輸explain出：

在這裡，DFEPipelineScan掃描從匿名節點開始?anon_node7並在另一個匿名節點結束的邊緣?anon_node21，關係類型另存為?p_type1。有一個過濾器用於存?p_type1在el://route（其中el代表邊緣標籤），它對應[p:route]於查詢字符串中。

DFEDrain收集限制為 10 的輸出解決方案，如其Arguments欄所示。 DFEDrain一旦達到限制或產生所有解決方案，以先發生者為準，就會終止。

```
curl -d "query=MATCH ()-[p:route]->() RETURN p LIMIT 10" -k https://localhost:8182/
openCypher -d "explain=details"
                                    ~
Query:
MATCH ()-[p:route]->() RETURN p LIMIT 10

##########################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments          # Mode                #
 Units In # Units Out # Ratio # Time (ms) #
##########################################################################################################
```

```
# 0  # 1      # -      # SolutionInjection # solutions=[{}]     # -                   # 0
      # 1         # 0.00  # 0          #
#####################################################################################################
# 1  # 2      # -      # DFESubquery       # subQuery=subQuery1 # -                   # 0
      # 10        # 0.00  # 5.00      #
#####################################################################################################
# 2  # -      # -      # TermResolution    # vars=[?p]          # id2value_opencypher # 10
      # 10        # 1.00  # 1.00      #
#####################################################################################################


subQuery1
#####################################################################################################
# ID # Out #1 # Out #2 # Name             # Arguments
          # Mode # Units In # Units Out # Ratio # Time (ms) #
#####################################################################################################
# 0  # 1      # -      # DFEPipelineScan # pattern=Edge((?anon_node7)-[?p:?p_type1]->(?
anon_node21)) # -    # 0        # 1000      # 0.00  # 0.66      #
#    #        #        #                  # inlineFilters=[(?p_type1 IN [<el://route>])]
          #      #          #           #       #           #
#    #        #        #                  # patternEstimate=26219
          #      #          #           #       #           #
#####################################################################################################
# 1  # 2      # -      # DFEProject      # columns=[?p]
          # -    # 1000     # 1000      # 1.00  # 0.14      #
#####################################################################################################
# 2  # -      # -      # DFEDrain        # limit=10
          # -    # 1000     # 0         # 0.00  # 0.11      #
#####################################################################################################
```

值運算式函數的explain輸出範例

該功能是：

```
MATCH (a) RETURN DISTINCT labels(a)
```

在下面的explain輸出中，DFEPipelineScan(ID 0) 會掃描所有節點標籤。這對應於MATCH （a)。

DFEChunkLocalSubquery(ID 1) 彙總每個?a標籤?a的標籤。這對應於labels(a)。您可以看到，透過DFEApply和DFEReduce。

BindRelation (ID 2) 用於?__gen_labels0fa2將泛型列重命名為?labels(a)。

DFEDistinctRelation (ID 4) 僅檢索不同的標籤 (多個：機場節點會給出重複的標籤 (a) ：["機場"]) 。這對應於DISTINCT labels(a)。

```
curl -d "query=MATCH (a) RETURN DISTINCT labels(a)" -k https://localhost:8182/openCypher -d
 "explain=details"
                          ~
Query:
MATCH (a) RETURN DISTINCT labels(a)

#####################################################################################################
# ID # Out #1 # Out #2 # Name             # Arguments          # Mode               #
 Units In # Units Out # Ratio # Time (ms) #
#####################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]     # -                   # 0
      # 1         # 0.00  # 0          #
#####################################################################################################
# 1  # 2      # -      # DFESubquery       # subQuery=subQuery1 # -                   # 0
      # 5         # 0.00  # 81.00      #
#####################################################################################################
```

```
# 2  # -      # -      # TermResolution    # vars=[?labels(a)]  # id2value_opencypher # 5
      # 5         # 1.00  # 1.00      #
#################################################################################################################


subQuery1
#################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                                                                                  # Mode # Units In #
 Units Out # Ratio # Time (ms) #
#################################################################################################################
# 0  # 1      # -      # DFEPipelineScan       # pattern=Node(?a) with property 'ALL' and
 label '?a_label1'                                                    # -    # 0        #
 3750      # 0.00  # 26.77     #
#    #        #        #                       # patternEstimate=3506
                                                                          #      #          #
      #       #           #
#################################################################################################################
# 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#8b314f55-2cc7-456a-a48a-c76a0465cfab/graph_1 # -    # 3750     #
 3750      # 1.00  # 0.04      #
#################################################################################################################
# 2  # 3      # -      # DFEBindRelation       # inputVars=[?a, ?__gen_labelsOfa2, ?
__gen_labelsOfa2]                                                          # -    # 3750
 # 3750      # 1.00  # 0.08      #
#    #        #        #                       # outputVars=[?a, ?__gen_labelsOfa2, ?
labels(a)]                                                                 #      #
 #           #       #           #
#################################################################################################################
# 3  # 4      # -      # DFEProject            # columns=[?labels(a)]
                                                                          # -    # 3750     # 3750
      # 1.00  # 0.05      #
#################################################################################################################
# 4  # 5      # -      # DFEDistinctRelation   # -
                                                                          # -    # 3750     # 5
      # 0.00  # 2.78      #
#################################################################################################################
# 5  # -      # -      # DFEDrain              # -
                                                                          # -    # 5        # 0
      # 0.00  # 0.03      #
#################################################################################################################


subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#8b314f55-2cc7-456a-a48a-
c76a0465cfab/graph_1
#################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                  # Mode     # Units In # Units Out # Ratio # Time (ms) #
#################################################################################################################
# 0  # 1      # -      # DFESolutionInjection # outSchema=[?a]
                  # -        # 0        # 3750      # 0.00  # 0.02      #
#################################################################################################################
# 1  # 2      # 3      # DFETee               # -
                  # -        # 3750     # 7500      # 2.00  # 0.02      #
#################################################################################################################
# 2  # 4      # -      # DFEProject           # columns=[?a]
                  # -        # 3750     # 3750      # 1.00  # 0.04      #
#################################################################################################################
# 3  # 17     # -      # DFEOptionalJoin      # -
                  # -        # 7500     # 3750      # 0.50  # 0.44      #
#################################################################################################################
# 4  # 5      # -      # DFEDistinctRelation  # -
                  # -        # 3750     # 3750      # 1.00  # 2.23      #
#################################################################################################################
# 5  # 6      # -      # DFEDistinctColumn    # column=?a
                  # -        # 3750     # 3750      # 1.00  # 1.50      #
```

```
#    #        #        #                      # ordered=false
                #          #          #           #       #           #
#################################################################################################
# 6  # 7      # -      # DFEPipelineJoin      # pattern=Node(?a) with property 'ALL' and
 label '?a_label3' # -        # 3750     # 3750      # 1.00  # 10.58     #
#    #        #        #                      # patternEstimate=3506
                #          #          #           #       #           #
#################################################################################################
# 7  # 8      # 9      # DFETee               # -
                # -        # 3750     # 7500      # 2.00  # 0.02      #
#################################################################################################
# 8  # 10     # -      # DFEBindRelation      # inputVars=[?a_label3]
                # -        # 3750     # 3750      # 1.00  # 0.04      #
#    #        #        #                      # outputVars=[?100]
                #          #          #           #       #           #
#################################################################################################
# 9  # 11     # -      # DFEBindRelation      # inputVars=[?a, ?a_label3, ?100]
                # -        # 7500     # 3750      # 0.50  # 0.07      #
#    #        #        #                      # outputVars=[?a, ?a_label3, ?100]
                #          #          #           #       #           #
#################################################################################################
# 10 # 9      # -      # DFETermResolution    # column=?100
                # id2value # 3750     # 3750      # 1.00  # 7.60      #
#################################################################################################
# 11 # 12     # -      # DFEBindRelation      # inputVars=[?a, ?a_label3, ?100]
                # -        # 3750     # 3750      # 1.00  # 0.06      #
#    #        #        #                      # outputVars=[?a, ?100, ?a_label3]
                #          #          #           #       #           #
#################################################################################################
# 12 # 13     # -      # DFEApply             # functor=nodeLabel(?a_label3)
                # -        # 3750     # 3750      # 1.00  # 0.55      #
#################################################################################################
# 13 # 14     # -      # DFEProject           # columns=[?a, ?a_label3_alias4]
                # -        # 3750     # 3750      # 1.00  # 0.05      #
#################################################################################################
# 14 # 15     # -      # DFEMergeChunks       # -
                # -        # 3750     # 3750      # 1.00  # 0.02      #
#################################################################################################
# 15 # 16     # -      # DFEReduce            # functor=collect(?a_label3_alias4)
                # -        # 3750     # 3750      # 1.00  # 6.37      #
#    #        #        #                      # segmentationKey=[?a]
                #          #          #           #       #           #
#################################################################################################
# 16 # 3      # -      # DFEMergeChunks       # -
                # -        # 3750     # 3750      # 1.00  # 0.03      #
#################################################################################################
# 17 # -      # -      # DFEDrain             # -
                # -        # 3750     # 0         # 0.00  # 0.02      #
#################################################################################################
```

數學值表示式函數的explain輸出範例

在此範例中，RETURN abs(-10)執行簡單的評估，取得常數的絕對值-10。

DFEChunkLocalSubQuery（ID 1）對靜態值執行溶液注入-10，該值存儲在變量中?100。

DFEApply（ID 2）是對存儲在變量中的靜態值執行絕對值函abs()?100數的運算符。

以下是查詢和結果explain輸出：

```
curl -d "query=RETURN abs(-10)" -k https://localhost:8182/openCypher  -d "explain=details"

                            ~
```

```
Query:
RETURN abs(-10)

#################################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments             # Mode                #
 Units In # Units Out # Ratio # Time (ms) #
#################################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]        # -                   #
 0        # 1         # 0.00  # 0         #
#################################################################################################################
# 1  # 2      # -      # DFESubquery       # subQuery=subQuery1    # -                   #
 0        # 1         # 0.00  # 4.00      #
#################################################################################################################
# 2  # -      # -      # TermResolution    # vars=[?_internalVar1] # id2value_opencypher #
 1        # 1         # 1.00  # 1.00      #
#################################################################################################################


subQuery1
#################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                                                                  # Mode # Units In #
 Units Out # Ratio # Time (ms) #
#################################################################################################################
# 0  # 1      # -      # DFESolutionInjection  # outSchema=[]
                                                                  # -    # 0        # 1
      # 0.00  # 0.01      #
#################################################################################################################
# 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#c4cc6148-cce3-4561-93c0-deb91f257356/graph_1 # -    # 1        # 1
         # 1.00  # 0.03      #
#################################################################################################################
# 2  # 3      # -      # DFEApply              # functor=abs(?100)
                                                                  # -    # 1        # 1
      # 1.00  # 0.26      #
#################################################################################################################
# 3  # 4      # -      # DFEBindRelation       # inputVars=[?_internalVar2, ?_internalVar2]
                                                                  # -    # 1        # 1
      # 1.00  # 0.04      #
#    #        #        #                       # outputVars=[?_internalVar2, ?
_internalVar1]                                                                    #      #
       #           #       #           #
#################################################################################################################
# 4  # 5      # -      # DFEProject            # columns=[?_internalVar1]
                                                                  # -    # 1        # 1
      # 1.00  # 0.06      #
#################################################################################################################
# 5  # -      # -      # DFEDrain              # -
                                                                  # -    # 1        # 0
      # 0.00  # 0.05      #
#################################################################################################################

subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#c4cc6148-cce3-4561-93c0-
deb91f257356/graph_1
#################################################################################################################
# ID # Out #1 # Out #2 # Name                # Arguments                          # Mode
 # Units In # Units Out # Ratio # Time (ms) #
#################################################################################################################
# 0  # 1      # -      # DFESolutionInjection # solutions=[?100 -> [-10^^<LONG>]]   # -
 # 0        # 1         # 0.00  # 0.01      #
#    #        #        #                       # outSchema=[?100]                   #
 #          #           #       #           #
#################################################################################################################
# 1  # 3      # -      # DFERelationalJoin    # joinVars=[]                        # -
 # 2        # 1         # 0.50  # 0.18      #
#################################################################################################################
```

```
# 2  # 1      # -      # DFESolutionInjection # outSchema=[]                          # -
 # 0        # 1         # 0.00  # 0.01      #
#########################################################################################################
# 3  # -      # -      # DFEDrain             # -                                     # -
 # 1        # 0         # 0.00  # 0.02      #
#########################################################################################################
```

可變長度路徑 (VLP) 查詢的explain輸出範例

這是處理可變長度路徑查詢的更複雜查詢計畫的範例。為了清楚起見，此示例僅顯示部分explain輸出。

在中subQuery1，DFEPipelineScan(識別碼 0) 和 DFEChunkLocalSubQuery (ID 1) 會注入...graph_1子查詢，負責掃描含有程式碼的YPO節點。

在中subQuery1，DFEChunkLocalSubQuery(ID 2) 會注入...graph_2子查詢，負責掃描含有程式碼的LAX節點。

在中subQuery1，DFEChunkLocalSubQuery(ID 3) 會注入...graph3子查詢，該子查詢包含 DFELoopSubQuery (識別碼 17)，此子查詢又會插入子查詢。...graph5此操作負責解析兩個節點之間的查詢字符串中的-[*2]->可變長度模式。

```
curl -d "query=MATCH p=(a {code: 'YPO'})-[*2]->(b{code: 'LAX'}) return p" -k https://
localhost:8182/openCypher -d "explain=details"

                                        ~
Query:
MATCH p=(a {code: 'YPO'})-[*2]->(b{code: 'LAX'}) return p

#########################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments          # Mode                #
 Units In # Units Out # Ratio # Time (ms) #
#########################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]     # -                   # 0
      # 1         # 0.00  # 0         #
#########################################################################################################
# 1  # 2      # -      # DFESubquery       # subQuery=subQuery1 # -                   # 0
      # 0         # 0.00  # 84.00     #
#########################################################################################################
# 2  # -      # -      # TermResolution    # vars=[?p]          # id2value_opencypher # 0
      # 0         # 0.00  # 0         #
#########################################################################################################


subQuery1
#########################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                                                                  # Mode # Units In #
 Units Out # Ratio # Time (ms) #
#########################################################################################################
# 0  # 1      # -      # DFEPipelineScan       # pattern=Node(?a) with property 'code' as ?
a_code7 and label 'ALL'                                           # -    # 0        # 1
      # 0.00  # 0.68      #
#    #        #        #                      # inlineFilters=[(?a_code7 IN
 ["YPO"^^xsd:string])]                                                          #      #
         #           #       #           #
#    #        #        #                      # patternEstimate=1
                                                                  #      #          #
      #       #           #
#########################################################################################################
# 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#cc05129f-d07e-4622-bbe3-9e99558eca46/graph_1 # -    # 1        # 1
         # 1.00  # 0.03      #
```

```
################################################################################################################
# 2  # 3      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#cc05129f-d07e-4622-bbe3-9e99558eca46/graph_2 # -    # 1        # 1
         # 1.00  # 0.02      #
################################################################################################################
# 3  # 4      # -      # DFEChunkLocalSubQuery # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#cc05129f-d07e-4622-bbe3-9e99558eca46/graph_3 # -    # 1        # 0
         # 0.00  # 0.04      #
################################################################################################################
# 4  # 5      # -      # DFEBindRelation       # inputVars=[?__gen_path6, ?anon_rel26, ?
b_code8, ?b, ?a_code7, ?a, ?__gen_path6]                                  # -    # 0        # 0
         # 0.00  # 0.10      #
#    #        #        #                       # outputVars=[?__gen_path6, ?anon_rel26, ?
b_code8, ?b, ?a_code7, ?a, ?p]                                            #      #          #
        #       #           #
################################################################################################################
# 5  # 6      # -      # DFEProject            # columns=[?p]
                                                                                # -    # 0        # 0
      # 0.00  # 0.05      #
################################################################################################################
# 6  # -      # -      # DFEDrain              # -
                                                                                # -    # 0        # 0
      # 0.00  # 0.02      #
################################################################################################################


subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#cc05129f-d07e-4622-
bbe3-9e99558eca46/graph_1
################################################################################################################
# ID # Out #1 # Out #2 # Name                # Arguments
               # Mode # Units In # Units Out # Ratio # Time (ms) #
################################################################################################################
# 0  # 1      # -      # DFESolutionInjection # outSchema=[?a, ?a_code7]
               # -    # 0        # 1         # 0.00  # 0.01      #
################################################################################################################
# 1  # 2      # 3      # DFETee               # -
               # -    # 1        # 2         # 2.00  # 0.01      #
################################################################################################################
# 2  # 4      # -      # DFEDistinctColumn    # column=?a
               # -    # 1        # 1         # 1.00  # 0.25      #
#    #        #        #                      # ordered=false
               #      #          #           #       #           #
################################################################################################################
# 3  # 5      # -      # DFEHashIndexBuild    # vars=[?a]
               # -    # 1        # 1         # 1.00  # 0.05      #
################################################################################################################
# 4  # 5      # -      # DFEPipelineJoin      # pattern=Node(?a) with property 'ALL' and
 label '?a_label1' # -    # 1        # 1         # 1.00  # 0.47      #
#    #        #        #                      # patternEstimate=3506
               #      #          #           #       #           #
################################################################################################################
# 5  # 6      # 7      # DFESync              # -
               # -    # 2        # 2         # 1.00  # 0.04      #
################################################################################################################
# 6  # 8      # -      # DFEForwardValue      # -
               # -    # 1        # 1         # 1.00  # 0.01      #
################################################################################################################
# 7  # 8      # -      # DFEForwardValue      # -
               # -    # 1        # 1         # 1.00  # 0.01      #
################################################################################################################
# 8  # 9      # -      # DFEHashIndexJoin     # -
               # -    # 2        # 1         # 0.50  # 0.26      #
################################################################################################################
# 9  # -      # -      # DFEDrain             # -
               # -    # 1        # 0         # 0.00  # 0.02      #
################################################################################################################
```

```
subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#cc05129f-d07e-4622-
bbe3-9e99558eca46/graph_2
#########################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments
                       # Mode # Units In # Units Out # Ratio # Time (ms) #
#########################################################################################################
# 0  # 1      # -      # DFEPipelineScan       # pattern=Node(?b) with property 'code' as ?
b_code8 and label 'ALL' # -    # 0        # 1         # 0.00  # 0.38      #
#    #        #        #                       # inlineFilters=[(?b_code8 IN
 ["LAX"^^xsd:string])]                 #      #          #           #       #           #
#    #        #        #                       # patternEstimate=1
                       #      #          #           #       #           #
#########################################################################################################
# 1  # 2      # -      # DFEMergeChunks        # -
                       # -    # 1        # 1         # 1.00  # 0.02      #
#########################################################################################################
# 2  # 4      # -      # DFERelationalJoin     # joinVars=[]
                       # -    # 2        # 1         # 0.50  # 0.19      #
#########################################################################################################
# 3  # 2      # -      # DFESolutionInjection # outSchema=[?a, ?a_code7]
                       # -    # 0        # 1         # 0.00  # 0         #
#########################################################################################################
# 4  # -      # -      # DFEDrain              # -
                       # -    # 1        # 0         # 0.00  # 0.01      #
#########################################################################################################


subQuery=http://aws.amazon.com/neptune/vocab/v01/dfe/past/graph#cc05129f-d07e-4622-
bbe3-9e99558eca46/graph_3
#########################################################################################################
# ID # Out #1 # Out #2 # Name                   # Arguments
                                                                          # Mode     # Units In #
 Units Out # Ratio # Time (ms) #
#########################################################################################################
...
# 17 # 18     # -      # DFELoopSubQuery       # subQuery=http://aws.amazon.com/neptune/
vocab/v01/dfe/past/graph#cc05129f-d07e-4622-bbe3-9e99558eca46/graph_5 # -        # 1
 # 2         # 2.00  # 0.31      #
...
```

海王星开放密码

Amazon Neptune 中的 OpenCypher 實作使用海王星定義的交易語意 (p. 109)，但是，Bolt 驅動程式提供的隔離層級對 Bolt 交易語意有一些特定的含意，如以下各節所述。

唯讀的螺栓事務查詢

只讀查詢可以通過多種方式進行處理，具有不同的交易模型和隔離級別，如下所示：

隱含唯讀交易查詢

下面是一個只讀隱式事務的例子：

```
public void executeReadImplicitTransaction()
{
  // end point
  final String END_POINT = "(End Point URL)";

  // read query
```

```
  final String READ_QUERY = "MATCH (n) RETURN n limit 10";

  // create the driver
  final Driver driver = GraphDatabase.driver(END_POINT, AuthTokens.none(),
          Config.builder().withEncryption()
                          .withTrustStrategy(TrustStrategy.trustSystemCertificates())
                          .build());

  // create the session config
  SessionConfig sessionConfig = SessionConfig.builder()
                                             .withFetchSize(1000)
                                             .withDefaultAccessMode(AccessMode.READ)
                                             .build();

  // run the query as access mode read
  driver.session(sessionConfig).readTransaction(new TransactionWork<String>()
    {
      final StringBuilder resultCollector = new StringBuilder();

      @Override
      public String execute(final Transaction tx)
      {
        // execute the query
        Result queryResult = tx.run(READ_QUERY);

        // Read the result
        for (Record record : queryResult.list())
        {
          for (String key : record.keys())
          {
            resultCollector.append(key)
                           .append(":")
                           .append(record.get(key).asNode().toString());
          }
        }
        return resultCollector.toString();
      }

    }
  );

  // close the driver.
  driver.close();
}
```

由於僅供讀取複本只接受唯讀查詢，所以針對僅供讀取複本的所有查詢都會以讀取隱含交易的形式執行，而不論工作階段組態中設定的存取模式 Neptune 會在SNAPSHOT隔離語意下將讀取隱含交易評估為唯讀查詢 (p. 110)。

在失敗的情況下，默認情況下會重試讀隱式交易。

自動認可唯讀交易查詢

下面是一個只讀自動提交事務的例子：

```
public void executeAutoCommitTransaction()
{
  // end point
  final String END_POINT = "(End Point URL)";

  // read query
  final String READ_QUERY = "MATCH (n) RETURN n limit 10";
```

```
  // Create the session config.
  final SessionConfig sessionConfig = SessionConfig
    .builder()
    .withFetchSize(1000)
    .withDefaultAccessMode(AccessMode.READ)
    .build();

  // create the driver
  final Driver driver = GraphDatabase.driver(END_POINT, AuthTokens.none(),
    Config.builder()
          .withEncryption()
          .withTrustStrategy(TrustStrategy.trustSystemCertificates())
          .build());

  // result collector
  final StringBuilder resultCollector = new StringBuilder();

  // create a session
  final Session session = driver.session(sessionConfig);

  // run the query
  final Result queryResult = session.run(READ_QUERY);
  for (final Record record : queryResult.list())
  {
    for (String key : record.keys())
    {
      resultCollector.append(key)
                     .append(":")
                     .append(record.get(key).asNode().toString());
    }
  }

  // close the session
  session.close();

  // close the driver
  driver.close();
}
```

如果在工作階段組態READ中將存取模式設定為，Neptune 會在SNAPSHOT隔離語意下將自動認可交易查詢評估為唯讀查詢 (p. 110)。請注意，僅供讀取複本只接受唯讀查詢。

如果您沒有傳入工作階段設定，則預設會使用突變查詢隔離處理自動提交查詢，因此請務必傳入明確設定存取模式的工作階段組態。READ

在失敗的情況下，只讀自動提交查詢不會重試。

顯式唯讀交易查詢

下面是一個明確的只讀事務的例子：

```
public void executeReadExplicitTransaction()
{
  // end point
  final String END_POINT = "(End Point URL)";

  // read query
  final String READ_QUERY = "MATCH (n) RETURN n limit 10";

  // Create the session config.
  final SessionConfig sessionConfig = SessionConfig
    .builder()
    .withFetchSize(1000)
```

```
    .withDefaultAccessMode(AccessMode.READ)
    .build();

  // create the driver
  final Driver driver = GraphDatabase.driver(END_POINT, AuthTokens.none(),
    Config.builder()
          .withEncryption()
          .withTrustStrategy(TrustStrategy.trustSystemCertificates())
          .build());

  // result collector
  final StringBuilder resultCollector = new StringBuilder();

  // create a session
  final Session session = driver.session(sessionConfig);

  // begin transaction
  final Transaction tx = session.beginTransaction();

  // run the query on transaction
  final List<Record> list = tx.run(READ_QUERY).list();

  // read the result
  for (final Record record : list)
  {
    for (String key : record.keys())
    {
      resultCollector
        .append(key)
        .append(":")
        .append(record.get(key).asNode().toString());
    }
  }

  // commit the transaction and for rollback we can use beginTransaction.rollback();
  tx.commit();

  // close the driver
  driver.close();
}
```

如果在工作階段組態READ中將存取模式設定為，Neptune 會在SNAPSHOT隔離語意下將明確唯讀交易評估為唯讀查詢 (p. 110)。請注意，僅供讀取複本只接受唯讀查詢。

如果您沒有傳入工作階段設定，則會依預設使用突變查詢隔離處理明確的唯讀交易，因此請務必傳遞明確設定存取模式的工作階段組態。READ

如果發生失敗，預設情況下會重試唯讀顯式查詢。

突變螺栓交易查詢

與唯讀查詢一樣，可以使用不同的事務模型和隔離級別處理突變查詢的多種方式，如下所示：

隱式突變交易查詢

下面是一個隱式突變事務的例子：

```
public void executeWriteImplicitTransaction()
{
  // end point
  final String END_POINT = "(End Point URL)";
```

```
  // create node with label as label and properties.
  final String WRITE_QUERY = "CREATE (n:label {name : 'foo'})";

  // Read the vertex created with label as label.
  final String READ_QUERY = "MATCH (n:label) RETURN n";

  // create the driver
  final Driver driver = GraphDatabase.driver(END_POINT, AuthTokens.none(),
    Config.builder()
          .withEncryption()
          .withTrustStrategy(TrustStrategy.trustSystemCertificates())
          .build());

  // create the session config
  SessionConfig sessionConfig = SessionConfig
    .builder()
    .withFetchSize(1000)
    .withDefaultAccessMode(AccessMode.WRITE)
    .build();

  final StringBuilder resultCollector = new StringBuilder();

  // run the query as access mode write
  driver.session(sessionConfig).writeTransaction(new TransactionWork<String>()
  {
    @Override
    public String execute(final Transaction tx)
    {
      // execute the write query and consume the result.
      tx.run(WRITE_QUERY).consume();

      // read the vertex written in the same transaction
      final List<Record> list = tx.run(READ_QUERY).list();

      // read the result
      for (final Record record : list)
      {
        for (String key : record.keys())
        {
          resultCollector
            .append(key)
            .append(":")
            .append(record.get(key).asNode().toString());
        }
      }
      return resultCollector.toString();
    }
  }); // at the end, the transaction is automatically committed.

  // close the driver.
  driver.close();
}
```

作為突變查詢的一部分進行的讀取與 Neptune 突 (p. 110)變事務的通常保證READ COMMITTED隔離下執行。

無論您是否特別傳入工作階段組態，都會將交易視為寫入交易。

如需衝突，請參閱使用鎖定等待逾時的衝突解決機制 (p. 111)。

自動提交突變事務查詢

突變自動提交查詢繼承與突變隱式事務相同的行為。

如果您未傳入工作階段組態，則預設會將交易視為寫入交易。

在失敗的情況下，突變自動提交查詢不會自動重試。

顯式突變交易查詢

下面是一個明確的突變事務的例子：

```
public void executeWriteExplicitTransaction()
{
  // end point
  final String END_POINT = "(End Point URL)";

  // create node with label as label and properties.
  final String WRITE_QUERY = "CREATE (n:label {name : 'foo'})";

  // Read the vertex created with label as label.
  final String READ_QUERY = "MATCH (n:label) RETURN n";

  // create the driver
  final Driver driver = GraphDatabase.driver(END_POINT, AuthTokens.none(),
    Config.builder()
          .withEncryption()
          .withTrustStrategy(TrustStrategy.trustSystemCertificates())
          .build());

  // create the session config
  SessionConfig sessionConfig = SessionConfig
    .builder()
    .withFetchSize(1000)
    .withDefaultAccessMode(AccessMode.WRITE)
    .build();

  final StringBuilder resultCollector = new StringBuilder();

  final Session session = driver.session(sessionConfig);

  // run the query as access mode write
  final Transaction tx = driver.session(sessionConfig).beginTransaction();

  // execute the write query and consume the result.
  tx.run(WRITE_QUERY).consume();

  // read the result from the previous write query in a same transaction.
  final List<Record> list = tx.run(READ_QUERY).list();

  // read the result
  for (final Record record : list)
  {
    for (String key : record.keys())
    {
      resultCollector
        .append(key)
        .append(":")
        .append(record.get(key).asNode().toString());
    }
  }

  // commit the transaction and for rollback we can use tx.rollback();
  tx.commit();

  // close the session
  session.close();

  // close the driver.
  driver.close();
```

```
}
```

顯式突變查詢繼承與隱式突變交易相同的行為。

如果您未傳入工作階段組態，則預設會將交易視為寫入交易。

如需衝突，請參閱使用鎖定等待逾時的衝突解決機制 (p. 111)。

海王星开放密码限制

OpenCypher 的亞馬遜海王星版本仍然不支持密碼查詢語言參考版本 9 中指定的所有內容，如中所述。符合開放密碼標準 (p. 95)未來的版本預計將解決許多這些限制。

海王星开放密码

在亞馬遜海王星上使用 OpenCypher 時，可能會發生各種例外情況。以下是您可能從 HTTPS 端點或 Bolt 驅動程式收到的常見例外狀況 (Bolt 驅動程式的所有例外都會報告為伺服器狀態例外)：

HTTP 代碼	錯誤訊息	可擷取？	修正方式	
400	（語法錯誤，直接從 OpenCypher 解析器傳播）	否	請正確查詢語法，然後重試。	
500	`Operation terminated (out of memory)`	是	重新處理查詢以新增其他篩選條件，以減少所需的記憶體	
500	作業終止 (超過期限)	是	增加資料庫叢集參數群組中的查詢逾時，或重試要求。	
500	作業終止 (使用者取消)	是	重試 請求。	
500	資料庫重設正在進行中。請在叢集可用之後重試查詢。	是	重設完成時重試。	
500	作業失敗，因為並行作業發生衝突 (請重試)。交易目前正在倒回。	是	使用指數輪詢重試並重試策略 (p. 838)。	
400	########/功能不支持異常	否	不支援指定的操作。	
400	嘗試在唯讀複本上進行開啟密碼更新	否	將目標端點變更為寫入器端點。	
400	MalformedQueryException（海王星不顯示內部解析器狀態）	否	請更正查詢語法並重試。	
400	無法刪除節點，因為它仍然具有關係。若要刪除此節	否	而不是使`MATCH (n) DELETE`	

HTTP 代碼	錯誤訊息	可擷取？	修正方式	
	點，您必須先刪除其關係。		n用 MATCH(n) DETACH DELETE(n)	
400	無效的作業：嘗試移除節點的最後一個標籤。節點必須至少有一個標籤。	否	變更查詢和/或應用程式邏輯，以免刪除最後一個節點。	
500	已經違反的最大請求數，ConfiguredQueueCapacity={} 表示 connID = {}	是	目前無論堆棧和協議如何，都只能處理 8,192 個並發請求。	
500	突破了最大連接限制。	是	每個實例只允許 1000 個並發螺栓連接（對於 HTTP 沒有限制）。	
400	期望一個 [其中一個：節點，關係或路徑] 並得到一個文字	否	檢查您是否傳遞了正確的參數，正確的查詢語法，然後重試。	
400	屬性值必須是簡單的文字。或者：Set 屬性的預期地圖，但沒有找到一個。	否	SET 子句只接受簡單文字，而不接受複合類型。	
400	找不到傳遞要刪除的實體	否	檢查您嘗試刪除的實體是否存在於資料庫中。	
400	使用者沒有資料庫的存取權。	否	查看正在使用的 IAM 角色的政策。	
400	沒有令牌作為請求的一部分傳遞	否	在啟用 IAM 的叢集上，必須將正確簽署的權杖作為查詢請求的一部分傳遞。	
400	傳播錯誤訊息。	否	請透過要求識別碼聯絡AWS支援部門。	
500	作業終止 (內部錯誤)	是	請透過要求識別碼聯絡AWS支援部門。	

使用 SPARQL 存取海王星圖

SPARQL 是資源描述框架（RDF）的查詢語言，它是為網絡設計的圖形數據格式。亞馬遜海王星與 SPARQL 1.1 相容。這表示您可以連線到 Neptune 資料庫執行個體，並使用 SPARQL 1.1 查詢語言規格中所述的查詢語言查詢圖形。

SPARQL 的查詢包含 SELECT 子句，用於指定要傳回的變數，和 WHERE 子句，用於指定要比對圖形中的哪些資料。如果您不熟悉 SPARQL 查詢，請參閱 SPARQL 1.1 查詢語言中的編寫簡易查詢。

> **Important**
>
> 若要載入資料，SPARQL UPDATE INSERT 可能非常適用於小型資料集，但如果您需要從檔案載入大量資料，請參閱使用 Amazon 海王星大容量載入器擷取資料 (p. 255)。

如需有關海王星 SPARQL 實作細節的詳細資訊，請參閱。符合標準 (p. 92)

開始之前，您必須準備好以下事項：

- 海王星資料庫執行個體。如需建立 Neptune 資料庫執行個體的資訊，請參閱建立新的海王星資料庫叢集 (p. 54)。
- 與您的 Neptune 資料庫執行個體位於相同虛擬私有雲 (VPC) 中的 Amazon EC2 執行個體。

主題

- 使用 RDF4J 主控台連線至海王星資料庫執行個體 (p. 423)
- 使用 RDF4J 工作台連線至海王星資料庫執行個體 (p. 425)
- 使用 Java 連線至海王星資料庫執行個體 (p. 426)
- SPARQL HTTP API (p. 428)
- 查詢提示 (p. 437)
- SPARQL 描述相對於默認圖形的行為 (p. 447)
- 查詢狀態 API (p. 449)
- 取消查詢 (p. 450)
- 在亞馬遜海王星中使用 SPARQL 1.1 圖形存放區 HTTP 通訊協定 (GSP) (p. 452)
- 使用 SPARQL 分析海王星查詢執行 explain (p. 453)
- 在海王星中使用擴充功能的 SPARQL 聯合查詢 SERVICE (p. 473)

使用 RDF4J 主控台連線至海王星資料庫執行個體

RDF4J 主控台可讓您在 REPL (read-eval-print迴圈) 環境中試驗資源描述架構 (RDF) 圖形和查詢。

您可從 RDF4J 主控台新增遠端圖形資料庫做為儲存庫和查詢。本節將逐步引導您完成 RDF4J 主控台的設定，以便從遠端連線到 Neptune 資料庫執行個體。

若要使用 RDF4J 主控台連線至海王星

1. 從 RDF4J 網站的下載頁面下載 RDF4J SDK。
2. 解壓縮 RDF4J SDK zip 檔案。
3. 在終端機上，導覽至 RDF4J SDK 目錄，然後輸入下列命令來執行 RDF4J 主控台：

```
bin/console.sh
```

您應該會看到類似下列的輸出：

```
14:11:51.126 [main] DEBUG o.e.r.c.platform.PlatformFactory - os.name = linux
14:11:51.130 [main] DEBUG o.e.r.c.platform.PlatformFactory - Detected Posix platform
Connected to default data directory
RDF4J Console 3.6.1

3.6.1
Type 'help' for help.
```

```
>
```

您現在進入 > 提示。這是 RDF4J 主控台的一般提示。您可使用此提示設定儲存庫和其他操作。儲存庫有專用可執行查詢的提示。

4. 在>提示下，輸入下列命令，為 Neptune 資料庫執行個體建立 SPARQL 儲存庫：

```
create sparql
```

5. RDF4J 主控台會提示您輸入連接到 SPARQL 端點所需變數的值。

```
Please specify values for the following variables:
```

指定下列值：

變數名稱	值
SPARQL query endpoint	*https#//##########*
SPARQL update endpoint	*https#//##########*
Local repository ID [endpoint@localhost]	###
Repository title [SPARQL endpoint repository @localhost]	#######

如需尋找 Neptune 資料庫執行個體位址的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

如果操作成功，您會看到以下訊息：

```
Repository created
```

6. 出現>提示時，輸入下列命令以連線至 Neptune 資料庫執行個體：

```
open neptune
```

如果操作成功，您會看到以下訊息：

```
Opened repository 'neptune'
```

您現在進入 neptune> 提示。在此提示下，您可以針對 Neptune 圖形執行查詢。

> **Note**
>
> 現在您已新增存放庫，下次執行時bin/console.sh，您可以立即執行open neptune命令以連線到 Neptune 資料庫執行個體。

7. 在neptune>提示字元下，輸入下列命令以執行 SPARQL 查詢，此查詢最多可傳回圖形中的 10 個三元組 (subject-predicate-object)，方法是使用限制為 10 的?s ?p ?o查詢。若要查詢其他內容，請將 sparql 命令之後的文字換成其他的 SPARQL 查詢。

```
sparql select ?s ?p ?o where {?s ?p ?o} limit 10
```

使用 RDF4J 工作台連線至海王星資料庫執行個體

本節將逐步引導您使用 RDF4J 工作台和 RDF4J 伺服器連線到亞馬遜海王星資料庫執行個體的過程。RDF4J 服務器是必需的，因為它充當海王星 SPARQL HTTP 休息端點和 RDF4J 工作台之間的代理。

RDF4J Workbench 提供了簡單的介面以用來試驗圖形，包括可載入本機檔案。如需詳細資訊，請參閱 RDF4J 文件中的新增區段。

先決條件

開始之前，請執行以下動作：

- 安裝 Java 1.8 或更新版本。
- 安裝 RDF4J Server 和 RDF4J Workbench。如需詳細資訊，請參閱安裝 RDF4J Server 和 RDF4J Workbench。

使用 RDF4J 工作台連接到海王星

1. 在 Web 瀏覽器中，導覽到部署 RDF4J Workbench Web 應用程式的 URL。例如，如果您使用 Apache Tomcat，URL：https://*ec2_hostname*:8080/rdf4j-workbench/。
2. 若系統要求您 Connect to RDF4J Server (連接到 RDF4J 伺服器)，請確認 RDF4J 伺服器已安裝並執行中，且伺服器 URL 正確無誤。接著繼續進行下一個步驟。
3. 在左側窗格中，選擇 New repository (新增儲存庫)。

   在 New repository (新增儲存庫) 中：

   - 在 Type (類型) 下拉式清單中，選擇 SPARQL endpoint proxy (SPARQL 端點 Proxy)。
   - 在 ID 中，輸入 neptune。
   - 在標題中，輸入 Neptune 資料庫執行個體

   選擇 下一步。
4. 在 New repository (新增儲存庫) 中：

   - 在 SPARQL query endpoint URL (SPARQL 查詢端點 URL) 中，輸入 `https://`*`your-neptune-endpoint`*`:`*`port`*`/sparql`。
   - 在 SPARQL update endpoint URL (SPARQL 更新端點 URL) 中，輸入 `https://`*`your-neptune-endpoint`*`:`*`port`*`/sparql`。

   如需尋找 Neptune 資料庫執行個體位址的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

   選擇 建立 。
5. neptune 儲存庫現在會顯示在儲存庫清單中。可能需要幾分鐘的時間才能使用新的儲存庫。
6. 在表格的 Id 欄中，選擇 neptune 連結。
7. 從左側窗格選擇 Query (查詢)。

   > **Note**
   >
   > 如果 Explore (探索) 底下的功能表項目停用，您可能需要重新連線到 RDF4J 伺服器，並再次選擇 neptune 儲存庫。
   > 若要這樣做，您可以使用右上角的 [change] ([變更]) 連結。
8. 在查詢欄位中，輸入以下 SPARQL 查詢，然後選擇 Execute (執行)。

```
select ?s ?p ?o where {?s ?p ?o} limit 10
```

上述範例會使用限制為 10 的?s ?p ?o查詢，最多可傳回圖表中三元組 (subject-predicate-object) 的 10 個。

使用 Java 連線至海王星資料庫執行個體

本節將引導您完成連線至 Amazon Neptune 資料庫執行個體並執行 SPARQL 查詢的完整 Java 範例的執行過程。

請遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

若要使用 Java 連線至海王星

1. 在 EC2 執行個體上安裝 Apache Maven。首先，輸入以下內容，以新增包含 Maven 套件的儲存庫：

   ```
   sudo wget https://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-
   maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
   ```

   輸入以下內容以設定套件的版本編號：

   ```
   sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
   ```

   接著即可使用 yum 安裝 Maven：

   ```
   sudo yum install -y apache-maven
   ```
2. 本範例僅經過 Java 8 的測試。輸入以下內容將 Java 8 安裝在 EC2 執行個體上：

   ```
   sudo yum install java-1.8.0-devel
   ```
3. 輸入以下內容將 Java 8 設定為 EC2 執行個體上預設的執行時間：

   ```
   sudo /usr/sbin/alternatives --config java
   ```

   出現提示時，輸入 Java 8 的數字。
4. 輸入以下內容將 Java 8 設定為 EC2 執行個體上預設的編譯器：

   ```
   sudo /usr/sbin/alternatives --config javac
   ```

   出現提示時，輸入 Java 8 的數字。
5. 在新目錄中，建立一個 pom.xml 檔案，然後在文字編輯器中開啟檔案。
6. 將以下內容複製到pom.xml文件中並保存（您通常可以將版本號調整為最新的穩定版本）：

   ```
   <project xmlns="https://maven.apache.org/POM/4.0.0" xmlns:xsi="https://www.w3.org/2001/
   XMLSchema-instance"
     xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache.org/
   maven-v4_0_0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <groupId>com.amazonaws</groupId>
     <artifactId>RDFExample</artifactId>
   ```

```
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>RDFExample</name>
  <url>https://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>org.eclipse.rdf4j</groupId>
      <artifactId>rdf4j-runtime</artifactId>
      <version>3.6</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
          <groupId>org.codehaus.mojo</groupId>
          <artifactId>exec-maven-plugin</artifactId>
          <version>1.2.1</version>
          <configuration>
            <mainClass>com.amazonaws.App</mainClass>
          </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Note

如果您要修改現有的 Maven 專案，所需的相依性將在先前的程式碼中反白顯示。

7. 若要為範例來源碼 (`src/main/java/com/amazonaws/`) 建立子目錄，請在命令列輸入下列命令：

```
mkdir -p src/main/java/com/amazonaws/
```

8. 在 `src/main/java/com/amazonaws/` 目錄中，建立一個名為 `App.java` 的檔案，然後在文字編輯器中開啟檔案。
9. 將以下內容複製到 `App.java` 檔案。將*########## Neptune* 資料庫執行個體的位址。

Note

如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

```
package com.amazonaws;

import org.eclipse.rdf4j.repository.Repository;
import org.eclipse.rdf4j.repository.http.HTTPRepository;
import org.eclipse.rdf4j.repository.sparql.SPARQLRepository;

import java.util.List;
import org.eclipse.rdf4j.RDF4JException;
import org.eclipse.rdf4j.repository.RepositoryConnection;
import org.eclipse.rdf4j.query.TupleQuery;
import org.eclipse.rdf4j.query.TupleQueryResult;
import org.eclipse.rdf4j.query.BindingSet;
import org.eclipse.rdf4j.query.QueryLanguage;
```

```
import org.eclipse.rdf4j.model.Value;

public class App
{
    public static void main( String[] args )
    {
        String sparqlEndpoint = "https://your-neptune-endpoint:port/sparql";
        Repository repo = new SPARQLRepository(sparqlEndpoint);
        repo.initialize();

        try (RepositoryConnection conn = repo.getConnection()) {
           String queryString = "SELECT ?s ?p ?o WHERE { ?s ?p ?o } limit 10";

           TupleQuery tupleQuery = conn.prepareTupleQuery(QueryLanguage.SPARQL,
 queryString);

           try (TupleQueryResult result = tupleQuery.evaluate()) {
              while (result.hasNext()) {  // iterate over the result
                   BindingSet bindingSet = result.next();

                   Value s = bindingSet.getValue("s");
                   Value p = bindingSet.getValue("p");
                   Value o = bindingSet.getValue("o");

                   System.out.print(s);
                   System.out.print("\t");
                   System.out.print(p);
                   System.out.print("\t");
                   System.out.println(o);
              }
           }
        }
    }
}
```

10. 使用下列 Maven 命令編譯並執行範例：

```
mvn compile exec:java
```

上述範例會使用限制為 10 的`?s ?p ?o`查詢，最多可傳回圖表中三元組 (subject-predicate-object) 的 10 個。若要查詢其他項目，請將查詢換成其他 SPARQL 查詢。

範例中重複出現的結果將印出所傳回之各變數的值。`Value` 物件會轉換成 `String`，然後列印。如果您變更查詢的 `SELECT` 部分，您必須修改程式碼。

SPARQL HTTP API

SPARQL HTTP 請求在以下端點接受：`https://your-neptune-endpoint:port/sparql`

如需使用 SPARQL 連線到亞馬遜海王星的詳細資訊，請參閱使用 SPARQL 存取海王星圖 (p. 422)。

如需 SPARQL 通訊協定和查詢語言的詳細資訊，請參閱 SPARQL 1.1 通訊協定和 SPARQL 1.1 查詢語言規格。

下列主題提供 SPARQL RDF 序列化格式，以及如何搭配海王星使用 SPARQL HTTP API 的相關資訊。

內容

- 使用 HTTP REST 端點連線至海王星資料庫執行個體 (p. 429)
- 用於多部分 SPARQL 響應的可選 HTTP 尾隨標題 (p. 430)

- 海王星使用的 RDF 媒體類型 (p. 430)
  - 海王星 SPARQL 使用的 RDF 序列化格式 (p. 430)
  - 海王星 SPARQL 使用的 SPARQL 結果序列化格式 (p. 431)
  - 海王星可以用來匯入 RDF 資料的媒體類型 (p. 431)
  - Neptune 可用來匯出查詢結果的媒體類型 (p. 431)
- 使用 SPARQL 更新負載將數據導入海王星 (p. 432)
- 使用 SPARQL 更新卸載從海王星刪除數據 (p. 435)

使用 HTTP REST 端點連線至海王星資料庫執行個體

亞馬遜海王星為 SPARQL 查詢提供 HTTP 端點。REST 介面相容於 SPARQL 1.1 版。

> **Important**
>
> 發行版本： (p. 950)對於與亞馬遜海王星的所有連接，都必須強制使用 TLS 1.2 和 HTTPS。不再可能使用不安全的 HTTP 連接到海王星，或使用 HTTPS 與 1.2 之前的 TLS 版本一起使用。

以下說明引導您使用 curl 命令、透過 HTTPS 連線和使用 HTTP 語法連線到 SPARQL 端點。請遵循與 Neptune 資料庫執行個體位於相同虛擬私有雲端 (VPC) 中 Amazon EC2 執行個體的這些指示。

對海王星資料庫執行個體進行 SPARQL 查詢的 HTTP 端點為：`https://`*`your-neptune-endpoint`*`:`*`port`*`/sparql`。

> **Note**
>
> 如需尋找 Neptune 資料庫執行個體主機名稱的相關資訊，請參閱連接到亞馬遜海王星端點 (p. 121)章節。

使用 HTTP POST 執行 QUERY

下面的例子使用curl通**`QUERY`**過 HTTP POST 提交一個 SPARQL。

```
curl -X POST --data-binary 'query=select ?s ?p ?o where {?s ?p ?o} limit 10' https://your-neptune-endpoint:port/sparql
```

上述範例會使用限制為 10 的`?s ?p ?o`查詢，最多可傳回圖表中三元組 (subject-predicate-object) 的 10 個。若要查詢其他項目，請將查詢換成其他 SPARQL 查詢。

> **Note**
>
> SELECT 和 ASK 查詢的回應預設 MIME 類型為 `application/sparql-results+json`。
> CONSTRUCT 和 DESCRIBE 查詢回應的預設 MIME 類型為 `application/n-quads`。
> 如需 Neptune 用於序列化之媒體類型的清單，請參閱海王星 SPARQL 使用的 RDF 序列化格式 (p. 430)。

使用 HTTP POST 執行 UPDATE

下面的例子使用curl通**`UPDATE`**過 HTTP POST 提交一個 SPARQL。

```
curl -X POST --data-binary 'update=INSERT DATA { <https://test.com/s> <https://test.com/p>
 <https://test.com/o> . }' https://your-neptune-endpoint:port/sparql
```

上述範例插入以下三元組到 SPARQL 預設圖形：`<https://test.com/s> <https://test.com/p> <https://test.com/o>`

用於多部分 SPARQL 響應的可選 HTTP 尾隨標題

Note

此功能從海王星引擎版本 1.0. (p. 954) 3.0 開始提供。

SPARQL 查詢和更新的 HTTP 回應通常會傳回一個以上的部分或區塊。在查詢或更新開始傳送這些區塊之後，很難診斷發生的失敗，特別是因為第一個區塊附有的 HTTP 狀態碼200。

除非您明確要求結尾標頭，否則 Neptune 只會透過將錯誤訊息附加到訊息內文 (通常已損毀) 來報告此類失敗。

若要更容易偵測和診斷此類問題，您可以在要求中加入轉移編碼 (TE) 預告片標頭 (`te: trailers`) () (例如，請參閱 TE 要求標頭的 MDN 頁面)。這樣做會導致 Neptune 在響應塊的尾隨標題中包含兩個新的標題字段：

- `X-Neptune-Status`— 包含回應碼，後面接著簡短名稱。例如，在成功的情況下，尾隨標題將是：`X-Neptune-Status: 200 OK`。在發生故障的情況下，響應代碼將是 Neptune 引擎錯誤代碼 (p. 849)，例如`X-Neptune-Status: 500 TimeLimitExceededException`。
- `X-Neptune-Detail`— 對於成功的請求而言是空的。在錯誤的情況下，它包含 JSON 錯誤消息。由於 HTTP 標頭值中只允許使用 ASCII 字元，因此 JSON 字串會進行網址編碼。錯誤訊息仍會附加至回應訊息內文。

海王星使用的 RDF 媒體類型

資源描述架構 (RDF) 資料可以多種不同方法序列化，大部分都能為 SPARQL 取用或輸出：

海王星 SPARQL 使用的 RDF 序列化格式

- RDF/XML -RDF 的 XML 序列化，在 RDF 1.1 XML 語法中定義。媒體類型：`application/rdf+xml`。一般副檔名：`.rdf`。
- N 三元組 -一種基於行的純文本格式，用於編碼 RDF 圖形，RDF 1.1 N-三元組中定義。媒體類型：`application/n-triples`、`text/turtle` 或 `text/plain`。一般副檔名：`.nt`。
- N 四邊形 — 一種基於行的純文本格式，用於編碼 RDF 圖形，在 RDF 1.1 N 四邊形中定義。這是 N-Triples 的延伸。媒體類型：`application/n-quads`，若使用 7 位元 US-ASCII 編碼，則為 `text/x-nquads`。一般副檔名：`.nq`。
- 龜 — RDF 1.1 龜中定義的文本語法，允許一個 RDF 圖完全寫在一個緊湊和自然的文本形式，與常見的使用模式和數據類型的縮寫。Turtle 使用 N-Triples 及 SPARQL 三元組模式語法提供多層次的相容性。媒體類型：`text/turtle`一般副檔名：`.ttl`。
- triG — RDF 1.1 triG 中定義的 RDF 的文字語法，可讓 RDF 圖形完全以精簡且自然的文字形式撰寫，以及常見使用模式和資料類型的縮寫。TriG 是 Turtle 格式的延伸。媒體類型：`application/trig`。一般副檔名：`.trig`。
- N3（符號 3）— 在符號 3（N3）中定義的斷言和邏輯語言：可讀的 RDF 語法。N3 透過新增方程式 (為圖表本身的常值)、變數、邏輯暗示和功能述詞，擴充 RDF 資料模型，並提供 RDF/XML 的文字語法替代方案。媒體類型：`text/n3`。一般副檔名：`.n3`。
- JSO-LD-在 JSO-LD 1.0. 媒體類型中定義的數據序列化和消息傳遞格式：。`application/ld+json`一般副檔名：`.jsonld`。
- 特里克斯-RDF 在 XML 中的序列化，在特里克斯中定義：RDF 在 XML 中三元組。媒體類型：`application/trix`。一般副檔名：`.trix`。
- 結果-使用 SPAR QL 1.1 查詢結果 JSON 格式的 RDF 序列化。媒體類型：`application/sparql-results+json`。一般副檔名：`.srj`。
- RDF4J 二進位格式 — 用於編碼 RDF 資料的二進位格式，以 RDF4J 二進位 RDF 格式記錄。媒體類型：`application/x-binary-rdf`。

海王星 SPARQL 使用的 SPARQL 結果序列化格式

- SPARQL XML 結果 — SPARQL 查詢語言所提供的變數繫結和布林結果格式的 XML 格式，定義於 SPARQL 查詢結果 XML 格式 (第二版) 中。媒體類型：`application/sparql-results+xml`。一般副檔名：`.srx`。
- SPARQL CSV 和 TSV 結果 — 使用逗號分隔值和製表符分隔值來SELECT表示來自查詢的 SPARQL 查詢結果結果，定義於 SPAR QL 1.1 查詢結果 CSV 和 TSV 格式中。媒體類型：逗號分隔值為 `text/csv`，定位字元分隔值為 `text/tab-separated-values`。一般副檔名：逗號分隔值為 `.csv`，定位字元分隔值為 `.tsv`。
- 二進位結果表格 — 用來編碼 SPARQL 查詢輸出的二進位格式。媒體類型：`application/x-binary-rdf-results-table`。
- 結果-使用 SPAR QL 1.1 查詢結果 JSON 格式的 RDF 序列化。媒體類型：`application/sparql-results+json`。

海王星可以用來匯入 RDF 資料的媒體類型

Neptune 大量載入器支援的媒體類型 (p. 255)

- N-Triples
- N-Quads
- RDF/XML
- Turtle

SPARQL 更新載入可以匯入的媒體類型

- N-Triples
- N-Quads
- RDF/XML
- Turtle
- TriG
- N3
- JSON-LD

Neptune 可用來匯出查詢結果的媒體類型

若要指定 SPARQL 查詢回應的輸出格式，請以查詢請求傳送 "`Accept: media-type`" 標頭。例如：

```
curl -H "Accept: application/nquads" ...
```

SPARQL 選擇可以從海王星輸出的 RDF 媒體類型

- SPARQL JSON 結果 (這是預設值)
- SPARQL XML 結果
- 二進位結果表 (媒體類型：`application/x-binary-rdf-results-table`)
- 逗號分隔值 (CSV)
- 定位字元分隔值 (TSV)

SPARQL ASK 可以從海王星輸出的 RDF 媒體類型

- SPARQL JSON 結果 (這是預設值)

- SPARQL XML 結果
- 布林值 (媒體類型：`text/boolean`，表示 "true" 或 "false")

SPARQL 構造可以從海王星輸出的 RDF 媒體類型

- N-Quads (這是預設值)
- RDF/XML
- JSON-LD
- N-Triples
- Turtle
- N3
- TriX
- TriG
- SPARQL JSON 結果
- RDF4J 二進制格式

SPARQL 描述可以從海王星輸出的 RDF 媒體類型

- N-Quads (這是預設值)
- RDF/XML
- JSON-LD
- N-Triples
- Turtle
- N3
- TriX
- TriG
- SPARQL JSON 結果
- RDF4J 二進制格式

使用 SPARQL 更新負載將數據導入海王星

SPARQL 更新載入命令的語法是在 SPARQL 1.1 更新建議中指定：

```
LOAD SILENT (URL of data to be loaded) INTO GRAPH (named graph into which to load the data)
```

- **`SILENT`**— (選擇性) 即使在處理期間發生錯誤，也會導致作業傳回成功。

  當單個事務包含多個語句（例如，即使某些遠程數據無法處理，"`LOAD ...; LOAD ...; UNLOAD ...; LOAD ...;`"並且您希望事務完成）時，這很有用。
- *`###### URL`* — (必要) 指定包含要載入圖形之資料的遠端資料檔案。

  遠端檔案必須具有下列其中一個副檔名：
  - NTriples 為 `.nt`。
  - NQuads 為 `.nq`。
  - Trig 為 `.trig`。
  - RDF/XML 為 `.rdf`。
  - Turtle 為 `.ttl`。
  - N3 為 `.n3`。

  - JSON-LD 為 `.jsonld`。
- **`INTO GRAPH`***`(############)`*-(選擇性) 指定應載入資料的圖形。

  海王星將每三個與一個命名的圖形相關聯。您可以使用後援命名圖形 URI 來指定預設的具名圖形`http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`，如下所示：

```
INTO GRAPH <http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph>
```

  Note

  當您需要載入大量資料時，我們建議您使用 Neptune 大量載入器，而不是使用 UPDATE LOAD。如需大量載入器的詳細資訊，請參閱使用 Amazon 海王星大容量載入器擷取資料 (p. 255)。

您可以使用`SPARQL UPDATE LOAD`直接從 Amazon S3 載入資料，或從自我託管 Web 伺服器取得的檔案載入資料。要載入的資源必須與 Neptune 伺服器位於相同的區域，並且 VPC 中必須允許資源的端點。如需建立 Amazon S3 端點的相關資訊，請參閱建立亞馬遜 S3 VPC 端點 (p. 270)。

所有 `SPARQL UPDATE LOAD` URI 都必須以`https://`開頭。這包括亞馬遜 S3 網址。

與 Neptune 大量載入器不同，呼叫`SPARQL UPDATE LOAD`是完全交易的。

使用 SPARQL 更新負載將文件直接從亞馬遜 S3 加載到海王星

由於海王星不允許您在使用 SPARQL 更新負載時將 IAM 角色傳遞給 Amazon S3，因此有問題的 Amazon S3 儲存貯體必須是公有的，或者您必須在負載查詢中使用預先簽署的 Amazon S3 URL。

若要為 Amazon S3 檔案產生預先簽署的 URL，您可以使用如下AWS CLI命令：

```
aws s3 presign --expires-in (number of seconds) s3://(bucket name)/(path to file of data to
 load)
```

然後，您可以在`LOAD`命令中使用生成的預先簽名 URL：

```
curl https://(a Neptune endpoint URL):8182/sparql \
  --data-urlencode 'update=load (pre-signed URL of the remote Amazon S3 file of data to be
 loaded) \
                          into graph (named graph)'
```

如需詳細資訊，請參閱對請求進行身分驗證：使用查詢參數。Boto3 文件展示了如何使用 Python 指令碼來產生一個預先簽署的 URL。

此外，必須正確設定要載入的檔案內容類型。

1. 使用參數將檔案上傳到 Amazon S3 時，設定檔案的內容類`-metadata`型，如下所示：

   ```
   aws s3 cp test.nt s3://bucket-name/my-plain-text-input/test.nt --metadata Content-
   Type=text/plain
   aws s3 cp test.rdf s3://bucket-name/my-rdf-input/test.rdf --metadata Content-
   Type=application/rdf+xml
   ```

2. 確認媒體類型資訊真實存在。執行：

   ```
   curl -v bucket-name/folder-name
   ```

   此命令的輸出應該顯示您在上傳檔案時所設定的媒體類型資訊。

3. 然後，您可以使用SPARQL UPDATE LOAD命令將這些文件導入 Neptune：

```
curl https://your-neptune-endpoint:port/sparql \
  -d "update=LOAD <https://s3.amazonaws.com/bucket-name/my-rdf-input/test.rdf>"
```

上述步驟僅適用於公有 Amazon S3 儲存貯體，或在 LOAD 查詢中使用預先簽署的 Amazon S3 URL 存取的儲存貯體。

您也可以設定 Web 代理伺服器以從私有 Amazon S3 儲存貯體載入，如下所示：

使用網絡服務器將文件加載到海王星與 SPARQL 更新負載

1. 在託管 Neptune 的 VPC 內執行的電腦上安裝 Web 伺服器，以及要載入的檔案。例如，使用 Amazon Linux，您可能會以下列方式安裝 Apache：

```
sudo yum install httpd mod_ssl
sudo /usr/sbin/apachectl start
```

2. 定義您要載入之 RDF 檔案內容的 MIME 類型。SPARQL 使用 Web 伺服器傳送的 `Content-type` 標頭來決定內容的輸入格式，所以您必須定義 Web 伺服器的相關 MIME 類型。

   例如，假設您使用以下副檔名來識別檔案格式：

   - NTriples 為 `.nt`。
   - NQuads 為 `.nq`。
   - Trig 為 `.trig`。
   - RDF/XML 為 `.rdf`。
   - Turtle 為 `.ttl`。
   - N3 為 `.n3`。
   - JSON-LD 為 `.jsonld`。

   如果您使用 Apache 2 做為 Web 伺服器，您要編輯檔案 `/etc/mime.types` 並新增以下類型：

```
text/plain nt
application/n-quads nq
application/trig trig
application/rdf+xml rdf
application/x-turtle ttl
text/rdf+n3 n3
application/ld+json jsonld
```

3. 確認 MIME 類型映射有效。一旦啟動並執行 Web 伺服器，並以您選擇的格式託管 RDF 檔案，您就可以從本機主機向 Web 伺服器傳送請求，測試組態。

   例如，您可能會傳送類似以下的請求：

```
curl -v http://localhost:80/test.rdf
```

   然後，在 `curl` 的詳細輸出中，您應該會看到如下一行：

```
Content-Type: application/rdf+xml
```

   這會顯示已成功定義的內容類型映射。
4. 您現在可以使用 SPARQL 更新命令來載入資料：

```
curl https://your-neptune-endpoint:port/sparql \
    -d "update=LOAD <http://web_server_private_ip:80/test.rdf>"
```

Note

使用 SPARQL UPDATE LOAD 可以在載入大型來源檔案時於 Web 伺服器上觸發逾時。Neptune 會在檔案資料串流輸入的同時進行處理，對大型檔案而言，所需時間可能會比伺服器上設定的逾時時間長。這反過來可能會導致伺服器關閉連線，當 Neptune 在串流中遇到非預期的 EOF 時，可能會導致下列錯誤訊息：

```
{
  "detailedMessage":"Invalid syntax in the specified file",
  "code":"InvalidParameterException"
}
```

如果您收到此訊息，但不認為來源檔案包含無效的語法，請嘗試增加 Web 伺服器上的逾時時間設定。您也可以透過在伺服器上啟用偵錯日誌並尋找逾時時間來診斷問題。

使用 SPARQL 更新卸載從海王星刪除數據

Neptune 也提供自訂 SPARQL 作業UNLOAD，用於移除遠端來源中指定的資料。 UNLOAD可以被視為對應的LOAD操作。它的語法是：

Note

此功能從海王星引擎版本 1.0.4.1 (p. 943) 開始提供。

```
UNLOAD SILENT (URL of the remote data to be unloaded) FROM GRAPH (named graph from which to
 remove the data)
```

- **SILENT**— (選用) 即使在處理資料時發生錯誤，也會導致作業傳回成功。

  當單個事務包含多個語句（例如，即使某些遠程數據無法處理，"LOAD ...; LOAD ...; UNLOAD ...; LOAD ...;"並且您希望事務完成）時，這很有用。
- 要####### *URL* — (必要) 指定包含要從圖形中卸載之資料的遠端資料檔案。

  遠端檔案必須具有下列其中一個副檔名 (這些副檔名與 UPDATE-LOAD 支援的格式相同)：
  - NTriples 為 `.nt`。
  - NQuads 為 `.nq`。
  - Trig 為 `.trig`。
  - RDF/XML 為 `.rdf`。
  - Turtle 為 `.ttl`。
  - N3 為 `.n3`。
  - JSON-LD 為 `.jsonld`。

  該文件包含的所有數據將通過UNLOAD操作從數據庫集群中刪除。

  任何 Amazon S3 身份驗證都必須包含在 URL 中，才能卸載資料。您可以預先簽署 Amazon S3 檔案，然後使用產生的 URL 安全地存取該檔案。例如：

  ```
  aws s3 presign --expires-in (number of seconds) s3://(bucket name)/(path to file of data
   to unload)
  ```

  然後：

```
curl https://(a Neptune endpoint URL):8182/sparql \
  --data-urlencode 'update=unload (pre-signed URL of the remote Amazon S3 data to be
 unloaded) \
                           from graph (named graph)'
```

如需詳細資訊，請參閱對請求進行身分驗證：使用查詢參數。

- **FROM GRAPH** *(############)* — (選擇性) 指定應從中卸載遠端資料的具名圖形。

  海王星將每三個與一個命名的圖形相關聯。您可以使用後援命名圖形 URI 來指定預設的具名圖形http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph，如下所示：

```
FROM GRAPH <http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph>
```

以相同的方式LOAD對應於INSERT DATA { *(inline data)* }, UNLOAD 對應於DELETE DATA { *(inline data)* }. 就像DELETE DATA，UNLOAD不包含空白節點的數據工作。

例如，如果本地 Web 服務器提供的文件名為包data.nt含以下兩個三元組：

```
<http://example.org/resource#a> <http://example.org/resource#p> <http://example.org/
resource#b> .
<http://example.org/resource#a> <http://example.org/resource#p> <http://example.org/
resource#c> .
```

下面的UNLOAD命令將從命名的圖中刪除這兩個三元組：<http://example.org/graph1>

```
UNLOAD <http://localhost:80/data.nt> FROM GRAPH <http://example.org/graph1>
```

這與使用以下DELETE DATA命令具有相同的效果：

```
DELETE DATA {
  GRAPH <http://example.org/graph1> {
    <http://example.org/resource#a> <http://example.org/resource#p> <http://example.org/
resource#b> .
    <http://example.org/resource#a> <http://example.org/resource#p> <http://example.org/
resource#c> .
  }
}
```

UNLOAD命令拋出的異常

- **InvalidParameterException**— 資料中有空白節點。HTTP 狀態：400 錯誤的請求。

  訊息: Blank nodes are not allowed for UNLOAD

- **InvalidParameterException**-數據中的語法損壞。HTTP 狀態：400 錯誤的請求。

  訊息:Invalid syntax in the specified file.

- **UnloadUrlAccessDeniedException** — 訪問被拒絕. HTTP 狀態：400 錯誤的請求。

  訊息:Update failure: Endpoint *(Neptune endpoint)* reported access denied error. Please verify access.

- **`BadRequestException`** — 無法擷取遠端資料。HTTP 狀態：400 錯誤的請求。

  消息：（取決於 HTTP 響應）。

查詢提示

您可以使用查詢提示，為 Amazon Neptune 中的特定 SPARQL 查詢指定最佳化和評估策略。

查詢提示使用在 SPARQL 查詢中嵌入的額外三重模式表示，包含以下部分：

```
scope hint value
```

- scope — 決定查詢提示套用的查詢部分，例如查詢中的特定群組或完整查詢。
- hint — 識別要套用之提示的類型。
- value — 決定要考慮的系統方面的行為。

查詢提示和範圍在 Amazon Neptune 命名空間中會以預先定義的詞彙形式公開`http://aws.amazon.com/neptune/vocab/v01/QueryHints#`。本節範例包含命名空間做為 `hint` 前綴，此前綴已定義並包含在查詢中：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
```

例如，以下說明如何在 `SELECT` 查詢中包含 `joinOrder` 提示：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT ... {
 hint:Query hint:joinOrder "Ordered" .
 ...
}
```

上述查詢會指示 Neptune 引擎以指定順序評估查詢中的聯結，並停用任何自動重新排序。

使用查詢提示時請考慮以下情況：

- 您可以將不同的查詢提示合併在單一查詢中。例如，您可以使用 `bottomUp` 查詢提示來將子查詢標註為由下而上的評估，並使用 `joinOrder` 查詢提示來修復子查詢中的聯結順序。
- 您可以在不同的非重疊範圍中，多次使用相同的查詢提示。
- 查詢提示是提示。雖然查詢引擎通常的目標是要考慮給定的查詢提示，也可以予以忽略。
- 查詢提示為語意保留。新增查詢提示並不會變更查詢的輸出 (除了未提供任何排序保證時的潛在結果順序，也就是說，使用 ORDER BY 未明確強制執行結果順序時)。

下列各節提供有關可用查詢提示及其在 Neptune 中使用情況的詳細資訊。

主題

- 海王星中的 SPARQL 查詢提示範圍 (p. 438)
- 查joinOrder詢提示 (p. 438)
- 查evaluationStrategy詢提示 (p. 440)
- 查queryTimeout詢提示 (p. 442)
- 查rangeSafe詢提示 (p. 443)

- 查queryId詢提示 (p. 444)
- 查useDFE詢提示 (p. 444)
- 與描述一起使用的 SPARQL 查詢提示 (p. 444)

海王星中的 SPARQL 查詢提示範圍

下表顯示 Amazon Neptune 中 SPARQL 查詢提示的可用範圍、相關提示和說明。這些項目中的hint前置詞代表 Neptune 命名空間的提示：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
```

範圍	支援的提示	描述
hint:Query	joinOrder (p. 438)	查詢提示適用於整個查詢。
hint:Query	查詢超時 (p. 442)	逾時值適用於整個查詢。
hint:Query	安全範圍 (p. 443)	針對整個查詢停用類型促銷。
hint:Query	queryId (p. 444)	查詢 ID 值會套用至整個查詢。
hint:Query	使用 DFE (p. 444)	已啟用 (或停用) 整個查詢的 DFE 使用。
hint:Group	joinOrder (p. 438)	查詢提示適用於指定群組中的最上層元素，但不適用於巢狀元素 (如子查詢) 或父元素。
hint:SubQuery	evaluationStrategy (p. 440)	提示指定及套用到巢狀的 SELECT 子查詢。子查詢是獨立評估的，不會考慮在子查詢之前計算的解決方案。

查joinOrder詢提示

當您提交 SPARQL 查詢時，Amazon 海王星查詢引擎會調查查詢的結構。它會重新排序部分查詢，並嘗試將評估所需的工作量和查詢回應時間減到最低。

例如，一系列的連接三重模式通常不會依給定的順序評估。它使用啟發和統計資料 (例如個別模式的選擇性以及它們如何透過共用變數連接) 進行重新排序。此外，如果您的查詢包含更複雜的模式，例如子查詢、篩選器或複雜的 OPTION 或 MINS 區塊，Neptune 查詢引擎會在可能的情況下重新排序它們，以達到有效率的評估順序。

對於更複雜的查詢，Neptune 選擇評估查詢的順序可能不一定是最佳的。例如，Neptune 可能會遺漏在查詢評估期間出現的執行個體資料特性 (例如命中圖形中的功率節點)。

如果您知道資料的確切特性，並且想要手動指定查詢執行的順序，請使用 Neptune joinOrder 查詢提示來指定以指定順序評估查詢。

joinOrder提示語法

joinOrder 查詢提示是以三重模式指定，該模式包含在 SPARQL 查詢中。

為了清楚起見，下列語法會使用定義並包含在查詢中的hint前置詞來指定 Neptune 查詢提示命名空間：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
scope hint:joinOrder "Ordered" .
```

可用範圍

- hint:Query
- hint:Group

如需更多查詢提示範圍的詳細資訊，請參閱 海王星中的 SPARQL 查詢提示範圍 (p. 438)。

joinOrder提示範例

本節示範使用和不使用 joinOrder 查詢提示編寫的查詢以及相關的最佳化。

此範例假設資料集包含下列項目：

- 一個人名為 John，且 :likes 1,000 個人，包括 Jane。
- 一個人名為 Jane，且 :likes 10 個人，包括 John。

無查詢提示

以下 SPARQL 查詢會從一組社交網路資料中擷取所有名為 John 和 Jane 且彼此喜歡 (like) 的雙人組：

```
PREFIX : <https://example.com/>
SELECT ?john ?jane {
  ?person1 :name "Jane" .
  ?person1 :likes ?person2 .
  ?person2 :name "John" .
  ?person2 :likes ?person1 .
}
```

Neptune 查詢引擎可能會以不同於寫入的順序評估陳述式。例如，它可能選擇以下列順序評估：

1. 尋找名為 John 的所有人。
2. 尋找透過 :likes 邊緣連接到 John 的所有人。
3. 以人名 Jane 篩選這些人。
4. 以「透過 :likes 邊緣連接到 John」篩選這些人。

根據資料集，以此順序評估的結果是在第二個步驟會擷取 1,000 個實體。第三個步驟則將這些再縮減到單一節點 Jane。然後，最終步驟判斷 Jane 也 :likes John 節點。

查詢提示

從 Jane 節點開始是有利的，因為她只有 10 個傳出 :likes 邊緣。這可藉由避免在第二步驟擷取 1,000 個實體，減少查詢評估期間的工作量。

下列範例會使用 JoinOrder 查詢提示，藉由停用查詢的所有自動聯結重新排序，以確保Jane節點及其傳出邊緣會先處理：

```
PREFIX : <https://example.com/>
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT ?john ?jane {
  hint:Query hint:joinOrder "Ordered" .
  ?person1 :name "Jane" .
```

```
  ?person1 :likes ?person2 .
  ?person2 :name "John" .
  ?person2 :likes ?person1 .
}
```

適用的真實世界案例可以是在社交網路的應用，網路中的人被分類為具有許多連接的影響者或具有少量連接的普通使用者。在這類案例中，以先前範例中的查詢為例，可以確保先處理普通使用者 (Jane) 再處理影響者 (John)。

查詢提示和重新排序

這個範例可以再進一步處理。如果您知道單一節點的 `:name` 屬性是唯一的，可以透過重新排序和使用 `joinOrder` 查詢提示來加速查詢。此步驟可確保先擷取唯一的節點。

```
PREFIX : <https://example.com/>
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT ?john ?jane {
  hint:Query hint:joinOrder "Ordered" .
  ?person1 :name "Jane" .
  ?person2 :name "John" .
  ?person1 :likes ?person2 .
  ?person2 :likes ?person1 .
}
```

在這種情況下，您可以將查詢降低到每一個步驟只有單一動作，如下：

1. 尋找符合 `:name` Jane 的單一個人節點。
2. 尋找符合 `:name` John 的單一個人節點。
3. 確認第一個節點透過 `:likes` 邊緣連接到第二個節點。
4. 確認第二個節點透過 `:likes` 邊緣連接到第一個節點。

> **Important**
>
> 如果您選擇錯誤的順序，`joinOrder` 查詢提示可能導致效能大幅下降。例如，如果 `:name` 屬性不是唯一，前述範例的效率會很差。如果 100 個節點的名稱都是 Jane，且 1,000 個節點的名稱都是 John，則此查詢會造成對 `:likes` 邊緣檢查 1,000* 100 (共 100,000 個) 雙人組。

查evaluationStrategy詢提示

evaluationStrategy查詢提示會告訴 Amazon Neptune 查詢引擎，註解的查詢片段應從下而上進行評估，做為獨立單元。這表示之前評估步驟的解決方案不會用來計算查詢片段。查詢片段被當做獨立單位而評估，其生成的解決方案在計算後與查詢的其餘部分聯結。

使用 evaluationStrategy 查詢提示暗示區塊化 (非管道化的) 查詢計畫，這意味著使用查詢提示標註的片段解決方案，在主記憶體中會具體化和緩衝。使用此查詢提示可能會大幅增加評估查詢所需的主記憶體，尤其是在標註的查詢片段計算大量結果時。

evaluationStrategy提示語法

evaluationStrategy 查詢提示是以三重模式指定，該模式包含在 SPARQL 查詢中。

為了清楚起見，下列語法會使用定義並包含在查詢中的hint前置詞來指定 Neptune 查詢提示命名空間：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
hint:SubQuery hint:evaluationStrategy "BottomUp" .
```

可用範圍

- hint:SubQuery

  Note

  只有巢狀子查詢支援此查詢提示。

如需更多查詢提示範圍的詳細資訊，請參閱 海王星中的 SPARQL 查詢提示範圍 (p. 438)。

evaluationStrategy提示範例

本節示範使用和不使用 evaluationStrategy 查詢提示編寫的查詢以及相關的最佳化。

此範例假設資料集有下列特性：

- 包含 1,000 個有標記 :connectedTo 的邊緣。
- 每個 component 節點平均連接到 100 個其他 component 節點。
- 節點之間四躍點循環連接的數量通常是大約 100 個。

無查詢提示

以下 SPARQL 查詢會擷取透過四個躍點彼此循環連接的所有 component 節點：

```
PREFIX : <https://example.com/>
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT * {
  ?component1 :connectedTo ?component2 .
  ?component2 :connectedTo ?component3 .
  ?component3 :connectedTo ?component4 .
  ?component4 :connectedTo ?component1 .
}
```

Neptune 查詢引擎的方法是使用下列步驟評估此查詢：

- 擷取圖中的全部 connectedTo 邊緣，共 1,000 個。
- 擴大 100 倍 (從 component2 傳出的 connectedTo 邊緣數量)。

  中間結果：100,000 個節點。
- 擴大 100 倍 (從 component3 傳出的 connectedTo 邊緣數量)。

  中間結果：10,000,000 個節點。
- 掃描 10,000,000 個節點的循環結束。

這會得到串流查詢計畫，其主記憶體為不變數量。

查詢提示和子查詢

您可能想要犧牲主記憶體空間換取加速運算。使用 evaluationStrategy 查詢提示重寫查詢，您可以強制引擎計算兩個較小、具體化子集的聯結。

```
PREFIX : <https://example.com/>
          PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
```

```
SELECT * {
  {
    SELECT * WHERE {
      hint:SubQuery hint:evaluationStrategy "BottomUp" .
      ?component1 :connectedTo ?component2 .
      ?component2 :connectedTo ?component3 .
    }
  }
  {
    SELECT * WHERE {
      hint:SubQuery hint:evaluationStrategy "BottomUp" .
      ?component3 :connectedTo ?component4 .
      ?component4 :connectedTo ?component1 .
    }
  }
}
```

相較於反覆使用先前三重模式的結果做為即將發生模式的輸入時評估序列中的三重模式，evaluationStrategy 提示會使得兩個子查詢被獨立評估。兩種子查詢的中繼結果都是產生 100,000 個節點，然後聯結在一起形成最終輸出。

特別是，當您在較大的執行個體類型上執行 Neptune 時，將這兩個 100,000 個子集暫時儲存在主記憶體中，會增加記憶體使用量，進而大幅加快評估速度。

查queryTimeout詢提示

queryTimeout 查詢提示指定的逾時比資料庫參數群組中設定的 neptune_query_timeout 值短。

如果查詢因為此提示而終止，則會擲出 TimeLimitExceededException，並帶有 Operation terminated (deadline exceeded) 訊息。

queryTimeout提示語法

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT ... WHERE {
    hint:Query hint:queryTimeout 10 .
    # OR
    hint:Query hint:queryTimeout "10" .
    # OR
    hint:Query hint:queryTimeout "10"^^xsd:integer .
 ...
}
```

逾時值以毫秒表示。

逾時值必須小於資料庫參數群組中設定的 neptune_query_timeout 值。否則，會擲出 MalformedQueryException 例外狀況，並帶有 Malformed query: Query hint 'queryTimeout' must be less than neptune_query_timeout DB Parameter Group 訊息。

queryTimeout 查詢提示應該在主要查詢的 WHERE 子句中指定，或在其中一個子查詢的 WHERE 子句中指定，如下例所示。

它必須在所有查詢/子查詢和 SPARQL Updates 區段 (例如 INSERT 和 DELETE) 上設定一次。否則，會擲出 MalformedQueryException 例外狀況，並帶有 Malformed query: Query hint 'queryTimeout' must be set only once 訊息。

可用範圍

queryTimeout 提示可同時套用至 SPARQL 查詢和更新。

- 在 SPARQL 查詢中，它可以出現在主要查詢或子查詢的 WHERE 子句中。
- 在 SPARQL 更新中，可以在 INSERT、DELETE 或 WHERE 子句中設定。如果有多個更新子句，則只能在其中一個設定。

如需更多查詢提示範圍的詳細資訊，請參閱 海王星中的 SPARQL 查詢提示範圍 (p. 438)。

queryTimeout提示範例

下例在 UPDATE 查詢的主要 WHERE 子句中使用 `hint:queryTimeout`：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
INSERT {
    ?s ?p ?o
} WHERE {
    hint:Query hint:queryTimeout 100 .
    ?s ?p ?o .
}
```

在此，`hint:queryTimeout` 位在子查詢的 WHERE 子句中：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT * {
   ?s ?p ?o .
   {
      SELECT ?s WHERE {
         hint:Query hint:queryTimeout 100 .
         ?s ?p1 ?o1 .
      }
   }
}
```

查rangeSafe詢提示

使用此查詢提示可關閉 SPARQL 查詢的類型提升。

當您提交包含FILTER超過數值或範圍的 SPARQL 查詢時，Neptune 查詢引擎在執行查詢時，通常必須使用類型提升。這意味著它必須檢查可以保存您正在篩選的值的每種類型的值。

例如，如果您篩選的值等於 55，引擎必須尋找等於 55 的整數、等於 55L 的長整數、等於 55.0 的浮點數等於 55.0 等等。每個類型升級都需要對儲存空間進行額外查閱，這可能會導致顯然簡單的查詢需要非預期的長時間才能完成。

通常類型促銷是不必要的，因為你事先知道你只需要找到一個特定類型的值。在這種情況下，您可以使用rangeSafe查詢提示來關閉類型升級，大幅加快查詢速度。

rangeSafe提示語法

查rangeSafe詢提示會使用的值true來關閉類型提升。它也接受的值 `false` (預設值)。

範例。下列範例會示範如何在篩選o大於 1 的整數值時關閉類型提升：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT * {
   ?s ?p ?o .
   hint:Prior hint:rangeSafe 'true' .
```

```
  FILTER (?o > '1'^^<http://www.w3.org/2001/XMLSchema#int>)
```

查queryId詢提示

使用此查詢提示，將您自己的 queryId 值指派給 SPARQL 查詢。

範例：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT * WHERE {
  hint:Query hint:queryId "4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47"
  {?s ?p ?o}}
```

您指派的值在 Neptune DB 中的所有查詢中必須是唯一的。

查useDFE詢提示

您可以使用此查詢提示來啟用 DFE 執行查詢。根據預設，海王星不會在未將此查詢提示設定為的情況下使用DFE`true`，因為海王星 _dfe_query_ (p. 533) engine 例證參數預設為。`viaQueryHint`如果將該執行個體參數設定為`enabled`，則除了`useDFE`查詢提示設定為的查詢以`false`外，所有查詢都會使用 DFE 引擎。

啟用 DFE 進行查詢的範例：

```
PREFIX : <https://example.com/>
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>

SELECT ?john ?jane
{
  hint:Query hint:useDFE true .
  ?person1 :name "Jane" .
  ?person1 :likes ?person2 .
  ?person2 :name "John" .
  ?person2 :likes ?person1 .
}
```

與描述一起使用的 SPARQL 查詢提示

SPARQL `DESCRIBE` 查詢提供了一種靈活的機制來請求資源描述。不過，SPARQL 規格並未定義的精確語意。`DESCRIBE`

Neptune 從引擎版本 1.2.0.2 (p. 870) 開始，支援適用於不同情況的多`DESCRIBE`種不同模式和演算法。

此範例資料集可協助說明不同的模式：

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix : <https://example.com/> .

:JaneDoe :firstName "Jane" .
:JaneDoe :knows :JohnDoe .
:JohnDoe :firstName "John" .
:JaneDoe :knows _:b1 .
_:b1 :knows :RichardRoe .

:RichardRoe :knows :JaneDoe .
:RichardRoe :firstName "Richard" .

_:s1 rdf:type rdf:Statement .
```

```
_:s1 rdf:subject :JaneDoe .
_:s1 rdf:predicate :knows .
_:s1 rdf:object :JohnDoe .
_:s1 :knowsFrom "Berlin" .

:ref_s2 rdf:type rdf:Statement .
:ref_s2 rdf:subject :JaneDoe .
:ref_s2 rdf:predicate :knows .
:ref_s2 rdf:object :JohnDoe .
:ref_s2 :knowsSince 1988 .
```

下列範例假設使用 SPARQL 查詢要求資源:JaneDoe的描述，如下所示：

```
DESCRIBE <https://example.com/JaneDoe>
```

查describeMode詢提示

hint:describeModeSPARQL 查詢提示可用來選取下列其中一種海王星支援的 SPARQL DESCRIBE 模式：

描ForwardOneStep述模式

您可以使用像這樣的describeMode查詢提示調用ForwardOneStep模式：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
DESCRIBE <https://example.com/JaneDoe>
{
  hint:Query hint:describeMode "ForwardOneStep"
}
```

該ForwardOneStep模式只返回要描述的資源的屬性和轉發鏈接。在示例情況下，這意味著它返回具有的三元組:JaneDoe，要描述的資源，如主題：

```
:JaneDoe :firstName "Jane" .
:JaneDoe :knows :JohnDoe .
:JaneDoe :knows _:b301990159 .
```

請注意，描述查詢可能會與空白節點返回三元組，例如_:b301990159，與輸入數據集相比，每次都有不同的 ID。

描SymmetricOneStep述模式

SymmetricOneStep如果您未提供查詢提示，則為預設的「描述」模式。您也可以使用像這樣的describeMode查詢提示明確調用它：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
DESCRIBE <https://example.com/JaneDoe>
{
  hint:Query hint:describeMode "SymmetricOneStep"
}
```

在SymmetricOneStep語義下，DESCRIBE返回要描述的資源的屬性，正向鏈接和反向鏈接：

```
:JaneDoe :firstName "Jane" .
:JaneDoe :knows :JohnDoe .
:JaneDoe :knows _:b318767375 .
```

```
_:b318767631 rdf:subject :JaneDoe .

:RichardRoe :knows :JaneDoe .

:ref_s2 rdf:subject :JaneDoe .
```

簡潔有界描述（CBD）描述模式

使用如下所示的`describeMode`查詢提示叫用簡明界限描述 (CBD) 模式：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
DESCRIBE <https://example.com/JaneDoe>
{
  hint:Query hint:describeMode "CBD"
}
```

在`CBD`語義下，`DESCRIBE`返回要描述的資源的簡明有界描述（由 W3C 定義）：

```
:JaneDoe :firstName "Jane" .
:JaneDoe :knows :JohnDoe .
:JaneDoe :knows _:b285212943 .
_:b285212943 :knows :RichardRoe .

_:b285213199 rdf:subject :JaneDoe .
_:b285213199 rdf:type rdf:Statement .
_:b285213199 rdf:predicate :knows .
_:b285213199 rdf:object :JohnDoe .
_:b285213199 :knowsFrom "Berlin" .

:ref_s2 rdf:subject :JaneDoe .
```

RDF 資源的簡明界限描述（也就是 RDF 圖中的節點）是圍繞該節點可以單獨站立的最小子圖。實際上，這意味著如果您將此圖視為樹，並將指定的節點作為根，則沒有空白節點（bnodes）作為該樹的葉子。由於 bnodes 無法從外部定址或在後續查詢中使用，因此瀏覽圖形只是從當前節點中找到下一個單個躍點是不夠的。您還必須走得足夠遠才能找到可以在後續查詢中使用的東西（也就是說，bnode 以外的東西）。

計算中央商務區

給定源 RDF 圖中的特定節點（起始節點或根節點），該節點的 CBD 計算方式如下：

1. 在子圖中包括源圖中的所有語句，其中語句的主題是起始節點。
2. 遞歸地，對於子圖中迄今為止具有空白節點對象的所有語句，請在子圖中包含源圖中的所有語句，其中語句的主題是該空白節點，並且尚未包含在子圖中。
3. 遞歸地，對於到目前為止子圖中包含的所有語句，對於源圖中這些語句的所有具體說明，包括從每個具體化`rdf:Statement`節點開始的 CBD。

這會產生一個子圖，其中物件節點是 IRI 參考或常值，或空白節點不作為圖形中任何陳述式的主旨。請注意，CBD 無法使用單一 SPARQL 選取或建構查詢來計算。

對稱簡明有界描述 (SCBD) 描述模式

對稱簡明界限描述 (SCBD) 模式是使用如下所示的`describeMode`查詢提示來叫用：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
DESCRIBE <https://example.com/JaneDoe>
{
  hint:Query hint:describeMode "SCBD"
```

```
}
```

在 [SCBD語意] 下，DESCRIBE傳回資源的對稱簡明有界描述 (如 W3C 在使用 VoID 詞彙描述連結資料集中所定義)：

```
:JaneDoe :firstName "Jane" .
:JaneDoe :knows :JohnDoe .
:JaneDoe :knows _:b335544591 .
_:b335544591 :knows :RichardRoe .

:RichardRoe :knows :JaneDoe .

_:b335544847 rdf:subject :JaneDoe .
_:b335544847 rdf:type rdf:Statement .
_:b335544847 rdf:predicate :knows .
_:b335544847 rdf:object :JohnDoe .
_:b335544847 :knowsFrom "Berlin" .

:ref_s2 rdf:subject :JaneDoe .
```

CBD 和 SCBD 在ForwardOneStep和SymmetricOneStep模式上的優點是空白節點始終擴展以包含其表示形式。這可能是一個重要的優勢，因為您無法使用 SPARQL 查詢空白節點。此外，CBD 和 SCBD 模式還考慮了證明。

請注意，describeMode查詢提示也可以是WHERE子句的一部分：

```
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
DESCRIBE ?s
WHERE {
  hint:Query hint:describeMode "CBD" .
  ?s rdf:type <https://example.com/Person>
}
```

查describeIterationLimit詢提示

hint:describeIterationLimitSPARQL 查詢提示會針對要執行反覆運算式的「描述」演算法 (例如 CBD 和 SCBD) 的最大反覆運算擴充數提供選擇性限制。

描述限制在一起。因此，如果同時指定了迭代限制和語句限制，則必須滿足兩個限制，然後才能切斷 SHARECT 查詢。

此值的預設值為 5。您可以將其設定為 ZERO (0) 以指定反覆擴充的次數沒有限制。

查describeStatementLimit詢提示

hint:describeStatementLimitSPARQL 查詢提示會針對 SHARECT 查詢回應中可能存在的最大陳述式數目上限提供選擇性條件約束。它僅適用於迭代的描述算法，例如 CBD 和 SCBD。

描述限制在一起。因此，如果同時指定了迭代限制和語句限制，則必須滿足兩個限制，然後才能切斷 SHARECT 查詢。

此值的預設值為 5000。您可以將其設置為 ZERO（0）以指定返回的語句數沒有限制。

SPARQL 描述相對於默認圖形的行為

SPARQL DESCRIBE查詢表單可讓您擷取有關資源的資訊，而不需要知道資料的結構，也不必撰寫查詢。這些資訊的組合方式會留給 SPARQL 實作。Neptune 提供了幾個查詢提示 (p. 444)，以調用不同的模式和算法DESCRIBE以供使用。

在 Neptune 的實作中，無論模式為何，都DESCRIBE只會使用 SPARQL 預設圖形 (p. 92)中存在的資料。這與 SPARQL 處理資料集的方式一致 (請參閱 SPARQL 規格中的指定 RDF 資料集)。

在 Neptune 中，除非使用FROM和/或FROM NAMED子句指定特定的具名圖形，否則預設圖形包含資料庫中所有已命名圖形的聯集中的所有唯一三元組。在海王星的所有 RDF 數據存儲在一個命名的圖形。如果三重插入沒有命名圖上下文，Neptune 將其存儲在指定的命名圖形中。`http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`

當使用FROM子句指定一或多個具名圖形時，預設圖形是這些已命名圖形中所有唯一三元組的聯集。如果沒有FROM子句，並且有一個或多個子FROM NAMED句，則默認圖表是空的。

SPARQL 範例 DESCRIBE

考慮以下數據：

```
PREFIX ex: <https://example.com/>

GRAPH ex:g1 {
    ex:s ex:p1 "a" .
    ex:s ex:p2 "c" .
}

GRAPH ex:g2 {
    ex:s ex:p3 "b" .
    ex:s ex:p2 "c" .
}

ex:s ex:p3 "d" .
```

對於此查詢：

```
PREFIX ex: <https://example.com/>
DESCRIBE ?s
FROM ex:g1
FROM NAMED ex:g2
WHERE {
  GRAPH ex:g2 { ?s ?p "b" . }
}
```

海王星將返回：

```
ex:s ex:p1 "a" .
ex:s ex:p2 "c" .
```

在這裡，首先評估圖形模式`GRAPH ex:g2 { ?s ?p "b" }`，導致綁定?s，然後將DESCRIBE部分評估為默認圖形，現在只是ex:g1。

但是，對於此查詢：

```
PREFIX ex: <https://example.com/>
DESCRIBE ?s
FROM NAMED ex:g1
WHERE {
  GRAPH ex:g1 { ?s ?p "a" . }
}
```

Neptune 不會返回任何內容，因為當FROM NAMED子句存在沒有任何FROM子句時，默認圖形是空的。

在下面的查詢中DESCRIBE，使用沒有FROM或FROM NAMED子句存在：

```
PREFIX ex: <https://example.com/>
DESCRIBE ?s
WHERE {
  GRAPH ex:g1 { ?s ?p "a" . }
}
```

在這種情況下，默認圖形由數據庫中所有命名圖形的聯合中的所有唯一三元組成（正式地，RDF 合併），因此 Neptune 將返回：

```
ex:s ex:p1 "a" .
ex:s ex:p2 "c" .
ex:s ex:p3 "b" .
ex:s ex:p3 "d" .
```

查詢狀態 API

若要取得 SPARQL 查詢的狀態，請使用 HTTP GET 或 POST 向 `https://`*your-neptune-endpoint*`:`*port*`/sparql/status` 端點提出請求。

查詢狀態請求參數

queryId (選用)

執行中 SPARQL 查詢的 ID。只顯示指定查詢的狀態。

SPARQL 查詢狀態回應語法

```
{
    "acceptedQueryCount": integer,
    "runningQueryCount": integer,
    "queries": [
      {
        "queryId":"guid",
        "queryEvalStats":
          {
            "subqueries": integer,
            "elapsed": integer,
            "cancelled": boolean
          },
        "queryString": "string"
      }
    ]
}
```

SPARQL 查詢狀態回應值

acceptedQueryCount

自上次重新啟動 Neptune 引擎後接受的查詢數目。

runningQueryCount

目前執行中的 SPARQL 查詢數。

queries

目前 SPARQL 查詢的清單。

queryId

查詢的 GUID ID。Neptune 會自動將此 ID 值指派給每個查詢，或者您也可以指派自己的 ID (請參閱將自定義 ID 注入海王星小鬼或 SPARQL 查詢 (p. 126))。

queryEvalStats

此查詢的統計資訊。

subqueries

此查詢中的子查詢數。

elapsed

到目前為止查詢已執行的毫秒數。

cancelled

True 表示查詢已取消。

queryString

提交的查詢。

查詢狀態範例

下面是使用 `curl` 和 HTTP GET 的狀態命令範例。

```
curl https://your-neptune-endpoint:port/sparql/status
```

此輸出會顯示單一執行中查詢。

```
{
    "acceptedQueryCount":9,
    "runningQueryCount":1,
    "queries": [
        {
            "queryId":"fb34cd3e-f37c-4d12-9cf2-03bb741bf54f",
            "queryEvalStats":
                {
                    "subqueries": 0,
                    "elapsed": 29256,
                    "cancelled": false
                },
            "queryString": "SELECT ?s ?p ?o WHERE {?s ?p ?o}"
        }
    ]
}
```

取消查詢

若要取得 SPARQL 查詢的狀態，請使用 HTTP GET 或 POST 向 `https://your-neptune-endpoint:port/sparql/status` 端點提出請求。

SPARQL 查詢取消請求參數

cancelQuery

(必要) 告知狀態命令取消查詢。此參數不採用數值。

queryId

(必要) 要取消之執行中 SPARQL 查詢的 ID。

silent

(選用) 如果 `silent=true`，則執行中查詢會遭到取消，且 HTTP 回應代碼為 200。如果 `silent` 不存在或 `silent=false`，則會取消查詢並顯示 HTTP 500 狀態碼。

取消查詢範例

範例 1：描配 **`silent=false`** 進行取消

以下是使用狀態命令的範例，其使用 `curl` 搭配設定為 `false` 的 `silent` 參數來取消查詢：

```
curl https://your-neptune-endpoint:port/sparql/status \
  -d "cancelQuery" \
  -d "queryId=4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47" \
  -d "silent=false"
```

除非查詢已啟動串流結果，否則取消的查詢隨後將傳回 HTTP 500 代碼，回應如下：

```
{
  "code": "CancelledByUserException",
  "requestId": "4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47",
  "detailedMessage": "Operation terminated (cancelled by user)"
}
```

如果查詢已傳回 HTTP 200 代碼 (OK)，且在取消之前已啟動串流結果，則逾時例外狀況資訊會傳送至一般輸出串流。

範例 2：搭配 **`silent=true`** 進行取消

以下是與上述相同狀態命令的範例，但 `silent` 參數現在設定為 `true`：

```
curl https://your-neptune-endpoint:port/sparql/status \
  -d "cancelQuery" \
  -d "queryId=4d5c4fae-aa30-41cf-9e1f-91e6b7dd6f47" \
  -d "silent=true"
```

此命令會傳回與 `silent=false` 時相同的回應，但取消的查詢現在會傳回 HTTP 200 代碼，其回應如下：

```
{
  "head" : {
    "vars" : [ "s", "p", "o" ]
  },
  "results" : {
    "bindings" : [ ]
  }
}
```

在亞馬遜海王星中使用 SPARQL 1.1 圖形存放區 HTTP 通訊協定 (GSP)

在 SPARQL 1.1 圖形存放區 HTTP 通訊協定建議中，W3C 定義了用於管理 RDF 圖形的 HTTP 通訊協定。它定義了用於刪除，創建和替換 RDF 圖形內容，以及用於添加 RDF 語句到現有內容操作。

圖形存放區通訊協定 (GSP) 提供了一種方便的方式來操作整個圖形，而不必撰寫複雜的 SPARQL 查詢。

截至目前已發行版本: (p. 931)，海王星完全支持此協議。

圖形存放區通訊協定 (GSP) 的端點為：

```
https://your-neptune-cluster:port/sparql/gsp/
```

若要使用 GSP 存取預設圖形，請使用：

```
https://your-neptune-cluster:port/sparql/gsp/?default
```

要使用 GSP 訪問命名的圖形，請使用：

```
https://your-neptune-cluster:port/sparql/gsp/?graph=named-graph-URI
```

海王星 GSP 實施的特殊細節

海王星完全實現了定義 GSP 的 W3C 建議。但是，在某些情況下，該規格不涵蓋。

其中一種情況是，PUT或POST請求在請求主體中指定一個或多個具名圖形，這些圖形與請求 URL 指定的圖形不同。只有在要求主體 RDF 格式支援具名圖形時，才會發生這種情況，例如，使用Content-Type: application/n-quads或Content-Type: application/trig。

在這種情況下，海王星添加或更新所有存在於正文中的命名圖形，以及在 URL 中指定的命名圖形。

例如，假設從空白資料庫開始，您會將 upsert 投票的PUT要求傳送成三個圖形。其中一名urn:votes，包含所有選舉年份的所有選票。另外兩個人, 命名urn:votes:2005和urn:votes:2019, 包含來自特定選舉年份的投票. 請求及其有效負載如下所示：

```
PUT "http://your-Neptune-cluster:port/sparql/gsp/?graph=urn:votes"
  Host: example.com
  Content-Type: application/n-quads

  PAYLOAD:

  <urn:JohnDoe> <urn:votedFor> <urn:Labour> <urn:votes:2005>
  <urn:JohnDoe> <urn:votedFor> <urn:Conservative> <urn:votes:2019>
  <urn:JaneSmith> <urn:votedFor> <urn:LiberalDemocrats> <urn:votes:2005>
  <urn:JaneSmith> <urn:votedFor> <urn:Conservative> <urn:votes:2019>
```

執行請求之後，資料庫中的資料如下所示：

```
<urn:JohnDoe>   <urn:votedFor> <urn:Labour>           <urn:votes:2005>
<urn:JohnDoe>   <urn:votedFor> <urn:Conservative>     <urn:votes:2019>
<urn:JaneSmith> <urn:votedFor> <urn:LiberalDemocrats> <urn:votes:2005>
<urn:JaneSmith> <urn:votedFor> <urn:Conservative>     <urn:votes:2019>
<urn:JohnDoe>   <urn:votedFor> <urn:Labour>           <urn:votes>
<urn:JohnDoe>   <urn:votedFor> <urn:Conservative>     <urn:votes>
```

```
<urn:JaneSmith> <urn:votedFor> <urn:LiberalDemocrats> <urn:votes>
<urn:JaneSmith> <urn:votedFor> <urn:Conservative>     <urn:votes>
```

另一個不明確的情況是，在請求 URL 本身中使用任何PUTPOST、GET或。DELETE例如：

```
POST "http://your-Neptune-cluster:port/sparql/gsp/?
graph=urn:votes:2005&graph=urn:votes:2019"
```

或者：

```
GET "http://your-Neptune-cluster:port/sparql/gsp/?default&graph=urn:votes:2019"
```

在此情況下，海王星會傳回 HTTP 400，其中包含一則訊息，指出只能在要求 URL 中指定一個圖形。

使用 SPARQL 分析海王星查詢執行 explain

亞馬遜海王星增加了一個名為解釋的 SPARQL 功能。此功能是一種自助服務工具，用於了解 Neptune 引擎採取的執行方法。您透過將 explain 參數新增到提交 SPARQL 查詢的 HTTP 呼叫，來進行叫用。

此 explain 功能可提供查詢執行計劃邏輯結構的相關資訊。您可以使用此資訊來識別潛在的評估和執行瓶頸。然後，您可以使用查詢提示 (p. 437)，以改善您的查詢執行計劃。

主題

- SPARQL 查詢引擎在海王星中的運作方式 (p. 453)
- 如何使用 SPARQL explain 來分析海王星查詢執行 (p. 454)
- 在海王星中叫用 SPARQL explain 的範例 (p. 455)
- 海王星運營商 explain (p. 467)
- 海王星explain中的 SPARQL 的局限性 (p. 473)

SPARQL 查詢引擎在海王星中的運作方式

若要使用 SPARQL explain 功能提供的資訊，您需要瞭解 Amazon 海王星 SPARQL 查詢引擎如何運作的一些詳細資料。

此引擎會將每個 SPARQL 查詢轉換為運算子管道。從第一個運算子開始，名為繫結清單的中繼解決方案將套用至此運算子管道。您可以將繫結清單視為資料表，其中資料表標頭是查詢中所用的部分變數。因此，資料表中的每一列代表評估時間點之前的結果。

假設系統已為我們的資料定義兩個命名空間前綴：

```
  @prefix ex:   <http://example.com> .
  @prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

以下是此內容中簡單繫結清單的範例：

```
  ?person       | ?firstName
  ------------------------------------------------------
  ex:JaneDoe    | "Jane"
  ex:JohnDoe    | "John"
  ex:RichardRoe | "Richard"
```

對於三個人中的每一個人，清單會將 ?person 變數繫結至該人員的識別符，且將 ?firstName 變數繫結至該人員的名字。

一般情況下，變數可以維持在未繫結的狀態，例如，對於資料中不存在值的查詢中變數的 OPTIONAL 選項。

PipelineJoin運算子是目前在explain輸出中的 Neptune 查詢引擎運算子的範例。它會將輸入做為上個運算子的傳入繫結集，並針對三重模式來進行聯結，也就是 (?person, foaf:lastName, ?lastName)。這個運算子會在此輸入串流使用 ?person 變數的繫結，將這些繫結取代為三重模式，並從資料庫中查詢三元素。

當從上個資料表中在傳入繫結內容中執行時，PipelineJoin 會評估三個查詢，也就是下列項目：

```
(ex:JaneDoe,    foaf:lastName, ?lastName)
(ex:JohnDoe,    foaf:lastName, ?lastName)
(ex:RichardRoe, foaf:lastName, ?lastName)
```

這種方法稱為 as-bound 評估。系統會根據傳入解決方案重新聯結這個評估程序的解決方案，並在傳入解決方案中填補偵測到的 ?lastName。假設您找到所有三個人員的姓氏，運算子會產生一個傳出繫結清單，這會像下面這樣：

```
?person       | ?firstName | ?lastName
---------------------------------------
ex:JaneDoe    | "Jane"     | "Doe"
ex:JohnDoe    | "John"     | "Doe"
ex:RichardRoe | "Richard"  | "Roe"
```

這個傳出繫結清單則做為輸入，供管道中下一個運算子使用。最後，管道中最後一個運算子的輸出會定義查詢結果。

運算子管道通常是線性的，這是因為每個運算子會為單一連線的運算子發出解決方案。不過，在某些情況下，他們可以擁有更為複雜的結構。例如，SPARQL 查詢中的 UNION 運算子會映射到 Copy 操作。這個操作會複製繫結並將副本轉送到兩個子計劃，一個用於 UNION 的左側而另一個用於右側。

如需運算子的詳細資訊，請參閱 海王星運營商 explain (p. 467)。

如何使用 SPARQL explain 來分析海王星查詢執行

SPARQL explain 功能是 Amazon Neptune 中的自助服務工具，可協助您瞭解海王星引擎採用的執行方法。若要叫用 explain，您會以 explain=*mode* 的形式將參數傳遞到 HTTP 或 HTTPS 請求。

模式值可以是 static dynamic 或 details 的其中之一。

- 在靜態模式中，explain 只會列印查詢計劃的靜態結構。
- 在動態模式中，explain 還包括查詢計劃的動態層面。這些層面可能包含經由運算子流動的中繼繫結數和傳入繫結與傳出繫結的比率，以及運算子所耗費的時間。
- 在詳細資訊模式中，explain 會列印 dynamic 模式中顯示的資訊及其他詳細資訊，例如，實際 SPARQL 查詢字串，以及構成聯結運算子基礎之模式的預估範圍計數。

海王星支持explain與 W3C SPARQL 1.1 協議規範中列出的所有三個 SPARQL 查詢訪問協議一起使用，即：

1. HTTP GET
2. 使用 URL 編碼參數的 HTTP POST
3. 使用文字參數的 HTTP POST

如需 SPARQL 查詢引擎的詳細資訊，請參閱 SPARQL 查詢引擎在海王星中的運作方式 (p. 453)。

如需叫用 SPARQL explain 產生之輸出種類的詳細資訊，請參閱 在海王星中叫用 SPARQL explain 的範例 (p. 455)。

在海王星中叫用 SPARQL explain 的範例

本節中的範例顯示透過叫用 SPARQL explain 功能來分析 Amazon Neptune 中的查詢執行，可產生的各種輸出。

主題

- 了解 Explain 輸出 (p. 455)
- 詳細資訊模式輸出範例 (p. 457)
- 靜態模式輸出範例 (p. 458)
- 編碼參數的不同方式 (p. 459)
- 除文本/普通以外的其他輸出類型 (p. 459)
- 啟用 DFE 時的 SPARQL explain 輸出範例 (p. 461)

了解 Explain 輸出

在這個範例中，Jane Doe 知道兩個人，也就是 John Doe 和 Richard Roe：

```
@prefix ex: <http://example.com> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

ex:JaneDoe foaf:knows ex:JohnDoe .
ex:JohnDoe foaf:firstName "John" .
ex:JohnDoe foaf:lastName "Doe" .
ex:JaneDoe foaf:knows ex:RichardRoe .
ex:RichardRoe foaf:firstName "Richard" .
ex:RichardRoe foaf:lastName "Roe" .
.
```

若要判斷 Jane Doe 知道的所有人的名字，您可以編寫以下查詢：

```
 curl http(s)://your_server:your_port/sparql \
   -d "query=PREFIX foaf: <https://xmlns.com/foaf/0.1/> PREFIX ex: <https://
www.example.com/> \
       SELECT ?firstName WHERE { ex:JaneDoe foaf:knows ?person . ?person foaf:firstName ?
firstName }" \
   -H "Accept: text/csv"
```

此簡單查詢會傳回下列結果：

```
firstName
John
Richard
```

接著，透過新增 -d "explain=dynamic" 和使用預設的輸出類型 (而不是 text/csv)，來變更 curl 命令以叫用 explain：

```
 curl http(s)://your_server:your_port/sparql \
   -d "query=PREFIX foaf: <https://xmlns.com/foaf/0.1/> PREFIX ex: <https://
www.example.com/> \
       SELECT ?firstName WHERE { ex:JaneDoe foaf:knows ?person . ?person foaf:firstName ?
firstName }" \
   -d "explain=dynamic"
```

查詢現在會在適合列印 ASCII 格式 (HTTP 內容類型 text/plain) (也就是預設的輸出類型) 中傳回輸出：

```
##########################################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments
        # Mode     # Units In # Units Out # Ratio # Time (ms) #
##########################################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]
        # -        # 0        # 1         # 0.00  # 0         #
##########################################################################################################################
# 1  # 2      # -      # PipelineJoin      # pattern=distinct(ex:JaneDoe, foaf:knows, ?
person)     # -        # 1        # 2         # 2.00  # 1         #
#    #        #        #                   # joinType=join
        #          #          #           #       #           #
#    #        #        #                   # joinProjectionVars=[?person]
        #          #          #           #       #           #
##########################################################################################################################
# 2  # 3      # -      # PipelineJoin      # pattern=distinct(?person, foaf:firstName, ?
firstName) # -        # 2        # 2         # 1.00  # 1         #
#    #        #        #                   # joinType=join
        #          #          #           #       #           #
#    #        #        #                   # joinProjectionVars=[?person, ?firstName]
        #          #          #           #       #           #
##########################################################################################################################
# 3  # 4      # -      # Projection        # vars=[?firstName]
        # retain   # 2        # 2         # 1.00  # 0         #
##########################################################################################################################
# 4  # -      # -      # TermResolution    # vars=[?firstName]
        # id2value # 2        # 2         # 1.00  # 1         #
##########################################################################################################################
```

對於在 Name 欄及其引數中的運算子詳細資訊，請參閱 解釋運營商 (p. 467)。

以下以逐列說明輸出：

1. 主要查詢的第一步一律是使用 SolutionInjection 運算子來注入解決方案。然後，系統會透過評估過程將解決方案擴展至最終結果。

   在這種情況下，它會注入所謂的通用解決方案 { }。存在 VALUES 子句或 BIND 時，此步驟也可能注入更為複雜的變數繫結來開始使用。

   Units Out 欄表示這個單一解決方案會自此運算子傳出。Out #1 欄會指定此運算子將結果送入其中的運算子。在這個範例中，所有運算子會連接到此資料表中的以下運算子。
2. 第二個步驟是 PipelineJoin。它收到做為上個運算子 (Units In := 1) 產生的輸入單一 (完全無限制) 解決方案。並將其聯結至 pattern 引數定義的元組模式。此會對應到模式的簡單查詢。在這種情況下，會將三重模式定義為下列各項：

   ```
   distinct( ex:JaneDoe, foaf:knows, ?person )
   ```

   joinType := join 引數表示一般聯結操作 (其他類型包括 optional 聯結、existence check 聯結等)。

   distinct := true 引數表示您從資料庫 (非重複項目) 擷取的剛好是不同的符合項目，且您將不同的符合項目繫結至變數 joinProjectionVars := ?person，來去重複。

   事實上，Units Out 欄值為 2 表示有兩個傳出的解決方案。特別是，反映兩個人員的 ?person 變數繫結，此資料說明 Jane Doe 知道：

   ```
   ?person
   -------------
   ex:JohnDoe
   ```

```
 ex:RichardRoe
```

3. 兩種解決方案從階段 2 流程做為輸入 (Units In := 2) 傳入至第二個 PipelineJoin。此運算子會將兩個舊解決方案與以下三重模式聯結：

```
distinct(?person, foaf:firstName, ?firstName)
```

已知運算子傳入解決方案會將 ?person 變數繫結至 ex:JohnDoe 或 ex:RichardRoe。基於此，PipelineJoin 會擷取名字，John 和 Richard。兩個傳出解決方案 (Units Out := 2) 則如下所示：

```
?person       | ?firstName
--------------------------
ex:JohnDoe    | John
ex:RichardRoe | Richard
```

4. 下一個投影運算子使用階段 3 (Units In := 2) 兩個解決方案的輸入並投射到 ?firstName 變數。這樣就不需在映射中繫結所有其他變數並在兩個繫結 (Units Out := 2) 上進行傳遞：

```
?firstName
----------
John
Richard
```

5. 為了改善效能，Neptune 會盡可能在指派給諸如 URI 和字串常值之類的項目的內部識別碼上運作，而不是在字串本身上運作。最後一個運算子 TermResolution 會透過這些內部識別符執行映射，並傳回對應的字詞字串。

在一般 (非說明) 查詢評估中，會將最後一個運算子運算的結果序列化為請求的序列化格式並串流到用戶端。

詳細資訊模式輸出範例

Note

SPARQL 說明詳細資料模式可從海王星引擎版本 1.0. (p. 969) 2.1 開始使用。

假設您執行的查詢與先前在詳細資訊模式 (而不是動態模式) 中執行的一樣：

```
 curl http(s)://your_server:your_port/sparql \
   -d "query=PREFIX foaf: <https://xmlns.com/foaf/0.1/> PREFIX ex: <https://
www.example.com/> \
       SELECT ?firstName WHERE { ex:JaneDoe foaf:knows ?person . ?person foaf:firstName ?
firstName }" \
   -d "explain=details"
```

如本範例所示，輸出與其他一些詳細資訊相同，例如，輸出頂端的查詢字串，以及 PipelineJoin 運算子的 patternEstimate 計數：

```
Query:
PREFIX foaf: <https://xmlns.com/foaf/0.1/> PREFIX ex: <https://www.example.com/>
SELECT ?firstName WHERE { ex:JaneDoe foaf:knows ?person . ?person foaf:firstName ?
firstName }

################################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments
        # Mode     # Units In # Units Out # Ratio # Time (ms) #
################################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]
        # -        # 0        # 1         # 0.00  # 0         #
```

```
##################################################################################################
# 1  # 2      # -      # PipelineJoin      # pattern=distinct(ex:JaneDoe, foaf:knows, ?
person)     # -        # 1        # 2         # 2.00  # 13        #
#    #        #        #                   # joinType=join
        #          #          #           #       #           #
#    #        #        #                   # joinProjectionVars=[?person]
        #          #          #           #       #           #
#    #        #        #                   # patternEstimate=2
        #          #          #           #       #           #
##################################################################################################
# 2  # 3      # -      # PipelineJoin      # pattern=distinct(?person, foaf:firstName, ?
firstName) # -        # 2        # 2         # 1.00  # 3         #
#    #        #        #                   # joinType=join
        #          #          #           #       #           #
#    #        #        #                   # joinProjectionVars=[?person, ?firstName]
        #          #          #           #       #           #
#    #        #        #                   # patternEstimate=2
        #          #          #           #       #           #
##################################################################################################
# 3  # 4      # -      # Projection        # vars=[?firstName]
        # retain   # 2        # 2         # 1.00  # 1         #
##################################################################################################
# 4  # -      # -      # TermResolution    # vars=[?firstName]
        # id2value # 2        # 2         # 1.00  # 7         #
##################################################################################################
```

靜態模式輸出範例

假設您執行的查詢與先前在靜態模式 (預設) (而不是詳細資訊模式) 中執行的一樣：

```
 curl http(s)://your_server:your_port/sparql \
   -d "query=PREFIX foaf: <https://xmlns.com/foaf/0.1/> PREFIX ex: <https://
www.example.com/> \
       SELECT ?firstName WHERE { ex:JaneDoe foaf:knows ?person . ?person foaf:firstName ?
firstName }" \
   -d "explain=static"
```

如此範例所示，輸出都是相同的，但它省略最後三個欄位：

```
##################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments
        # Mode     #
##################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]
        # -        #
##################################################################################################
# 1  # 2      # -      # PipelineJoin      # pattern=distinct(ex:JaneDoe, foaf:knows, ?
person)     # -        #
#    #        #        #                   # joinType=join
        #          #
#    #        #        #                   # joinProjectionVars=[?person]
        #          #
##################################################################################################
# 2  # 3      # -      # PipelineJoin      # pattern=distinct(?person, foaf:firstName, ?
firstName) # -        #
#    #        #        #                   # joinType=join
        #          #
#    #        #        #                   # joinProjectionVars=[?person, ?firstName]
        #          #
##################################################################################################
# 3  # 4      # -      # Projection        # vars=[?firstName]
        # retain   #
##################################################################################################
```

```
# 4  # -      # -      # TermResolution    # vars=[?firstName]
       # id2value #
#########################################################################################################
```

編碼參數的不同方式

以下範例查詢說明叫用 SPARQL `explain` 時兩種不同的參數編碼方法。

使用 URL 編碼 — 此範例使用參數的 URL 編碼，並指定動態輸出：

```
curl -XGET "http(s)://your_server:your_port/sparql?query=SELECT%20*%20WHERE%20%7B%20%3Fs
%20%3Fp%20%3Fo%20%7D%20LIMIT%20%31&explain=dynamic"
```

直接指定參數 — 這與先前的查詢相同，不同之處在於它會直接透過 POST 傳遞參數：

```
curl http(s)://your_server:your_port/sparql \
  -d "query=SELECT * WHERE { ?s ?p ?o } LIMIT 1" \
  -d "explain=dynamic"
```

除文本/普通以外的其他輸出類型

上述範例使用預設`text/plain`輸出類型。海王星還可以將 SPARQL `explain` 輸出格式化為其他兩種 MIME 類型的格式，即和。`text/csv` `text/html`您透過設定 HTTP `Accept` 標頭 (也就是您在 `curl` 使用 `-H` 旗標可以做的事) 來進行叫用，如下所示：

```
  -H "Accept: output type"
```

以下是一些範例：

`text/csv` 輸出

此查詢透過指定 `-H "Accept: text/csv"` 來呼叫 CSV MIME 類型輸出：

```
curl http(s)://your_server:your_port/sparql \
  -d "query=SELECT * WHERE { ?s ?p ?o } LIMIT 1" \
  -d "explain=dynamic" \
  -H "Accept: text/csv"
```

CSV 格式 (匯入到試算表或資料庫的實用格式)，透過分號 (`;`) 將每個 `explain` 列中的欄位分隔，如下所示：

```
ID;Out #1;Out #2;Name;Arguments;Mode;Units In;Units Out;Ratio;Time (ms)
0;1;-;SolutionInjection;solutions=[{}];-;0;1;0.00;0
1;2;-;PipelineJoin;pattern=distinct(?s, ?p, ?o),joinType=join,joinProjectionVars=[?s, ?p, ?
o];-;1;6;6.00;1
2;3;-;Projection;vars=[?s, ?p, ?o];retain;6;6;1.00;2
3;-;-;Slice;limit=1;-;1;1;1.00;1
```

`text/html` 輸出

如果您指定 `-H "Accept: text/html"`，則 `explain` 會產生 HTML 資料表：

```
<!DOCTYPE html>
<html>
  <body>
```

```
<table border="1px">
  <thead>
    <tr>
      <th>ID</th>
      <th>Out #1</th>
      <th>Out #2</th>
      <th>Name</th>
      <th>Arguments</th>
      <th>Mode</th>
      <th>Units In</th>
      <th>Units Out</th>
      <th>Ratio</th>
      <th>Time (ms)</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>0</td>
      <td>1</td>
      <td>-</td>
      <td>SolutionInjection</td>
      <td>solutions=[{}]</td>
      <td>-</td>
      <td>0</td>
      <td>1</td>
      <td>0.00</td>
      <td>0</td>
    </tr>

    <tr>
      <td>1</td>
      <td>2</td>
      <td>-</td>
      <td>PipelineJoin</td>
      <td>pattern=distinct(?s, ?p, ?o)<br>
          joinType=join<br>
          joinProjectionVars=[?s, ?p, ?o]</td>
      <td>-</td>
      <td>1</td>
      <td>6</td>
      <td>6.00</td>
      <td>1</td>
    </tr>

    <tr>
      <td>2</td>
      <td>3</td>
      <td>-</td>
      <td>Projection</td>
      <td>vars=[?s, ?p, ?o]</td>
      <td>retain</td>
      <td>6</td>
      <td>6</td>
      <td>1.00</td>
      <td>2</td>
    </tr>

    <tr>
      <td>3</td>
      <td>-</td>
      <td>-</td>
      <td>Slice</td>
      <td>limit=1</td>
      <td>-</td>
      <td>1</td>
```

```
          <td>1</td>
          <td>1.00</td>
          <td>1</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

HTML 在瀏覽器中顯示如下：

ID	Out #1	Out #2	Name	Arguments	Mode	Units In	Units Out	Ratio	Time (ms)
0	1	-	SolutionInjection	solutions=[{}]	-	0	1	0.00	0
1	2	-	PipelineJoin	pattern=distinct(?s, ?p, ?o) joinType=join joinProjectionVars=[?s, ?p, ?o]	-	1	6	6.00	1
2	3	-	Projection	vars=[?s, ?p, ?o]	retain	6	6	1.00	2
3	-	-	Slice	limit=1	-	1	1	1.00	1

啟用 DFE 時的 SPARQL explain 輸出範例

以下是啟用海王星 DFE 替代查詢引擎時 SPARQL explain 輸出的範例：

```
##########################################################################################################################
# ID # Out #1 # Out #2 # Name              # Arguments

                                                                                                          # Mode     #
 Units In # Units Out # Ratio # Time (ms) #
##########################################################################################################################
# 0  # 1      # -      # SolutionInjection # solutions=[{}]

                                                                                                          # -        #
 0        # 1         # 0.00  # 0         #
##########################################################################################################################
# 1  # 2      # -      # HashIndexBuild    # solutionSet=solutionSet1

                                                                                                          # -        #
 1        # 1         # 1.00  # 22        #
#    #        #        #                   # joinVars=[]

                                                                                                          #          #
          #           #       #           #
#    #        #        #                   # sourceType=pipeline

                                                                                                          #          #
          #           #       #           #
##########################################################################################################################
# 2  # 3      # -      # DFENode           # DFE Stats=

                                                                                                                  # -
     # 101      # 100       # 0.99  # 32        #
#    #        #        #                   # ====> DFE execution time (measured by
 DFEQueryEngine)
                                                                                                                    #
        #          #           #       #           #
#    #        #        #                   # accepted [micros]=127

                                                                                                          #          #
          #           #       #           #
#    #        #        #                   # ready [micros]=2

                                                                                                          #          #
          #           #       #           #
#    #        #        #                   # running [micros]=5627
```

```
                                                                                #           #
          #           #       #           #
#    #        #        #                   # finished [micros]=0

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
#    #        #        #                   # ===> DFE execution time (measured in DFENode)

                                                                                #           #
          #           #       #           #
#    #        #        #                   # -> setupTime [ms]=1

                                                                                #           #
          #           #       #           #
#    #        #        #                   # -> executionTime [ms]=14

                                                                                #           #
          #           #       #           #
#    #        #        #                   # -> resultReadTime [ms]=0

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
#    #        #        #                   # ===> Static analysis statistics

                                                                                #           #
          #           #       #           #
#    #        #        #                   # --> 35907 micros spent in parser.

                                                                                #           #
          #           #       #           #
#    #        #        #                   # --> 7643 micros spent in range count
 estimation
                                                                                           #
         #          #           #       #           #
#    #        #        #                   # --> 2895 micros spent in value resolution

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
#    #        #        #                   # --> 39974925 micros spent in optimizer loop

                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                #           #
          #           #       #           #
```

```
#    #        #        #                   #

                                                                          #          #
          #           #       #           #
#    #        #        #                   # DFEJoinGroupNode[ children={

                                                                          #          #
          #           #       #           #
#    #        #        #                    #   DFEPatternNode[(?1, TERM[117442062], ?2, ?
3) . project DISTINCT[?1, ?2] {rangeCountEstimate=100},
                                                                            #
 #          #           #       #           #
#    #        #        #                   #     OperatorInfoWithAlternative[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #       rec=OperatorInfo[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #         type=INCREMENTAL_PIPELINE_JOIN,

                                                                          #          #
          #           #       #           #
#    #        #        #                   #         costEstimates=OperatorCostEstimates[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #
 costEstimate=OperatorCostEstimate[in=1.0000,out=100.0000,io=0.0002,comp=0.0000,mem=0],

                          #          #          #           #       #           #
#    #        #        #                   #
 worstCaseCostEstimate=OperatorCostEstimate[in=1.0000,out=100.0000,io=0.0002,comp=0.0000,mem=0]]],

                  #          #          #           #       #           #
#    #        #        #                   #       alt=OperatorInfo[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #         type=INCREMENTAL_HASH_JOIN,

                                                                          #          #
          #           #       #           #
#    #        #        #                   #         costEstimates=OperatorCostEstimates[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #
 costEstimate=OperatorCostEstimate[in=1.0000,out=100.0000,io=0.0003,comp=0.0000,mem=3212],

                          #          #          #           #       #           #
#    #        #        #                   #
 worstCaseCostEstimate=OperatorCostEstimate[in=1.0000,out=100.0000,io=0.0003,comp=0.0000,mem=3212]]]]],

             #          #          #           #       #           #
#    #        #        #                    #   DFEPatternNode[(?1, TERM[150997262], ?4, ?
5) . project DISTINCT[?1, ?4] {rangeCountEstimate=100},
                                                                            #
 #          #           #       #           #
#    #        #        #                   #     OperatorInfoWithAlternative[

                                                                          #          #
          #           #       #           #
#    #        #        #                   #       rec=OperatorInfo[
```

```
                                                                                                #           #
          #           #       #           #
#    #        #        #                   #         type=INCREMENTAL_HASH_JOIN,

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #         costEstimates=OperatorCostEstimates[

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #
 costEstimate=OperatorCostEstimate[in=100.0000,out=100.0000,io=0.0003,comp=0.0000,mem=6400],

                          #           #           #           #       #           #
#    #        #        #                   #
 worstCaseCostEstimate=OperatorCostEstimate[in=100.0000,out=100.0000,io=0.0003,comp=0.0000,mem=6400]]],

             #           #           #           #       #           #
#    #        #        #                   #       alt=OperatorInfo[

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #         type=INCREMENTAL_PIPELINE_JOIN,

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #         costEstimates=OperatorCostEstimates[

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #
 costEstimate=OperatorCostEstimate[in=100.0000,out=100.0000,io=0.0010,comp=0.0000,mem=0],

                            #           #           #           #       #           #
#    #        #        #                   #
 worstCaseCostEstimate=OperatorCostEstimate[in=100.0000,out=100.0000,io=0.0010,comp=0.0000,mem=0]]]]]

               #           #           #           #       #           #
#    #        #        #                   # },

                                                                                                #           #
          #           #       #           #
#    #        #        #                   # ]

                                                                                                #           #
          #           #       #           #
#    #        #        #                   #

                                                                                                #           #
          #           #       #           #
#    #        #        #                   # ===> DFE configuration:

                                                                                                #           #
          #           #       #           #
#    #        #        #                   # solutionChunkSize=5000

                                                                                                #           #
          #           #       #           #
#    #        #        #                   # ouputQueueSize=20

                                                                                                #           #
          #           #       #           #
#    #        #        #                   # numComputeCores=3

                                                                                                #           #
          #           #       #           #
```

```
#    #        #        #                  # maxParallelIO=10
                                                                                #          #
          #          #       #           #
#    #        #        #                  # numInitialPermits=12
                                                                                #          #
          #          #       #           #
#    #        #        #                  #
                                                                                #          #
          #          #       #           #
#    #        #        #                  #
                                                                                #          #
          #          #       #           #
#    #        #        #                  # ====> DFE configuration (reported back)
                                                                                #          #
          #          #       #           #
#    #        #        #                  # numComputeCores=3
                                                                                #          #
          #          #       #           #
#    #        #        #                  # maxParallelIO=2
                                                                                #          #
          #          #       #           #
#    #        #        #                  # numInitialPermits=12
                                                                                #          #
          #          #       #           #
#    #        #        #                  #
                                                                                #          #
          #          #       #           #
#    #        #        #                  # ===> Statistics & operator histogram
                                                                                #          #
          #          #       #           #
#    #        #        #                  # ==> Statistics
                                                                                #          #
          #          #       #           #
#    #        #        #                  # -> 3741 / 3668 micros total elapsed (incl.
 wait / excl. wait)
                                                                                  #
   #          #          #       #          #
#    #        #        #                  # -> 3741 / 3 millis total elapse (incl. wait /
 excl. wait)
                                                                                 #
 #          #          #       #           #
#    #        #        #                  # -> 3741 / 0 secs total elapsed (incl. wait /
 excl. wait)
                                                                                 #
 #          #          #       #           #
#    #        #        #                  # ==> Operator histogram
                                                                                #          #
          #          #       #           #
#    #        #        #                  # -> 47.66% of total time (excl. wait):
 pipelineScan (2 instances)
                                                                                       #
        #          #          #       #          #
#    #        #        #                      # -> 10.99% of total time (excl. wait): merge (1
 instances)
```

```
                                                
```

```
################################################################################################################
```

海王星運營商 explain

以下各節說明目前 Amazon 海王星中可用之 SPARQL `explain` 功能的運算子和參數。

> **Important**
>
> SPARQL `explain` 功能仍在進行改善，這裡記載的運算子和參數在未來的版本中可能會加以變更。

主題

- Aggregation運算子 (p. 467)
- ConditionalRouting運算子 (p. 467)
- Copy運算子 (p. 468)
- DFENode運算子 (p. 468)
- Distinct運算子 (p. 468)
- Federation運算子 (p. 468)
- Filter運算子 (p. 469)
- HashIndexBuild運算子 (p. 469)
- HashIndexJoin運算子 (p. 469)
- MergeJoin運算子 (p. 469)
- NamedSubquery運算子 (p. 470)
- PipelineJoin運算子 (p. 470)
- PipelineCountJoin運算子 (p. 470)
- PipelinedHashIndexJoin運算子 (p. 471)
- Projection運算子 (p. 471)
- PropertyPath運算子 (p. 471)
- TermResolution運算子 (p. 472)
- Slice運算子 (p. 472)
- SolutionInjection運算子 (p. 472)
- Sort運算子 (p. 472)
- VariableAlignment運算子 (p. 473)

Aggregation運算子

該運算子會執行一個或多個彙總，以實作 `count`、`max`、`min`、`sum` 等 SPARQL 彙總運算子的語意。

`Aggregation` 附帶使用 `groupBy` 子句的選用分組，以及選用的 `having` 限制。

引數

- `groupBy`— (選擇性) 提供子`groupBy`句，指定傳入解決方案所依據的分組運算式順序。
- `aggregates`— (必要) 指定彙總運算式的排序清單。
- `having`— (選擇性) 新增條件約束以篩選群組，如 SPARQL 查詢中的`having`子句所暗示。

ConditionalRouting運算子

該運算子會根據指定的條件路由傳入解決方案。符合該條件的解決方案會路由至 `Out #1` 所參考的運算子 ID，而不符合條件的解決方案則會路由至 `Out #2` 參考的運算子。

引數

- condition— (必要) 路由條件。

Copy運算子

該運算子會依指定模式的規定來委派解決方案串流。

模式

- forward— 將解決方案轉發給由Out #1識別的下游操作員。
- duplicate— 複製解決方案並將其轉發到由Out #1和識別的兩個運算子中的每一個。Out #2

Copy 並沒有任何引數。

DFENode運算子

這個運算子是 DFE 替代查詢引擎所執行之計畫的抽象。詳細的 DFE 計劃在此運算子的引數中概述。引數目前已多載，以包含 DFE 計劃的詳細執行階段統計資料。它包含在由 DFE 查詢執行的各個步驟花費的時間。

DFE 查詢計劃的邏輯最佳化抽象語法樹狀結構 (AST) 會列印出計劃時所考慮之運算子類型的相關資訊，以及執行運算子的相關最佳和最差成本。AST 目前由以下類型的節點組成：

- DFEJoinGroupNode— 代表一或多個聯結DFEPatternNodes。
- DFEPatternNode-封裝使用相符元組從基礎資料庫中投影出來的基礎模式。

子段落包含有關DataflowOp計劃執行時間以及每個操作員使用的 CPU 時間明細的詳細資訊。Statistics & Operator histogram下面有一個打印 DFE 執行的計劃的詳細運行時統計信息的表。

> Note
>
> 由於 DFE 是以實驗室模式發佈的實驗性功能，因此其explain輸出的確切格式可能會變更。

Distinct運算子

該運算子會運算變數子集上不同的投影，藉此消除重複項目。因此，流入的解決方案數量會大於或等於流出的解決方案數量。

引數

- vars-(必要) 要套用Distinct投影的變數。

Federation運算子

將指定的查詢傳遞至指定的遠端 SPARQL 端點。

引數

- endpoint— (必要) SPARQL SERVICE 陳述式中的端點網址。這可以是常數字串，或者如果查詢端點是根據相同查詢中的變數來決定，則它可以是變數名稱。
- query— (必要) 要傳送至遠端端點的重建查詢字串。即使用戶端未指定任何字首，引擎也會將預設字首新增至此查詢。

- silent— (必要) 布林值，指出SILENT關鍵字是否出現在關鍵字之後。 SILENT告訴引擎不要失敗整個查詢，即使遠程SERVICE部分失敗。

Filter運算子

該運算子會篩選傳入的解決方案。唯有符合篩選條件的解決方案可轉送至上游運算子，其他所有解決方案則會遭到捨棄。

引數

- condition— (必要) 篩選條件。

HashIndexBuild運算子

該運算子會接受繫結清單並將其多工緩衝處理至雜湊索引，該雜湊索引的名稱是由 solutionSet 引數所定義。後續的運算子通常會對這個解決方案集執行聯結操作，再用該名稱指向解決方案集。

引數

- solutionSet— (必要) 雜湊索引解決方案集的名稱。
- sourceType— (必要) 從中取得要儲存在雜湊索引之繫結的來源類型：
  - pipeline— 將運算子管線中下游運算子的傳入解決方案捲入雜湊索引。
  - binding set— 將引數指定的固定繫結集排入雜湊索sourceBindingSet引。
- sourceBindingSet— (選擇性) 如果sourceType引數值為binding set，此引數會指定要多工緩衝處理至雜湊索引的靜態繫結集。

HashIndexJoin運算子

該運算子會對 solutionSet 引數辨識的雜湊索引解決方案集執行傳入解決方案聯結操作。

引數

- solutionSet— (必要) 要加入之解決方案集的名稱。這必須是在先前步驟中使用 HashIndexBuild 運算子建構的雜湊索引解決方案。
- joinType— (必要) 要執行的連接類型：
  - join— 正常聯結，需要在所有共用變數之間完全相符。
  - optional— 使用 SPARQL OPTIONAL 運算子語意的optional聯結。
  - minus— minus 作業會使用 SPARQL MINUS 運算子語意，保留沒有聯結夥伴存在的對應。
  - existence check— 檢查是否存在聯結夥伴，並將existenceCheckResultVar變數繫結至此檢查的結果。
- constraints— (可選) 在連接期間考慮的其他接合約束。不符合這些限制的聯結皆會予以捨棄。
- existenceCheckResultVar— (選用) 僅用joinType於等於其中的連接 existence check (請參閱前面的joinType引數)。

MergeJoin運算子

該運算子會對 solutionSets 辨識的多個解決方案集執行合併聯結。

引數

- solutionSets— (必要) 要結合在一起的解決方案集。

NamedSubquery運算子

該運算子會針對 subQuery 引數識別的子查詢觸發評估作業，並將結果多工緩衝處理至 solutionSet 引數指定的解決方案集。運算子的傳入解決方案會轉送至該子查詢，接著轉送至下一個運算子。

引數

- subQuery— (必要) 要評估的子查詢名稱。輸出中會明確呈現該子查詢。
- solutionSet— (必要) 要儲存子查詢結果的解決方案集名稱。

PipelineJoin運算子

該運算子會接收先前運算子的輸出做為輸入，並將其聯結至 pattern 引數定義的元組模式。

引數

- pattern- (必填) 模式，它採用的形式subject-predicate-object，並可選擇-graph 元組的基礎連接。如果 distinct 為指定的模式值，則聯結僅會從 projectionVars 引數指定的投影變數中擷取不同解決方案，而不是所有相符的解決方案。
- inlineFilters— (選擇性) 要套用至模式中變數的一組篩選器。系統會一併評估模式與這些篩選條件。
- joinType— (必要) 要執行的連接類型：
  - join— 正常聯結，需要在所有共用變數之間完全相符。
  - optional— 使用 SPARQL OPTIONAL 運算子語意的optional聯結。
  - minus— minus 作業會使用 SPARQL MINUS 運算子語意，保留沒有聯結夥伴存在的對應。
  - existence check— 檢查是否存在聯結夥伴，並將existenceCheckResultVar變數繫結至此檢查的結果。
- constraints— (可選) 在連接期間考慮的其他接合約束。不符合這些限制的聯結皆會予以捨棄。
- projectionVars— (選擇性) 投影變數。此引數需與 distinct := true 結合使用，以在指定的一組變數內強制擷取不同投影。
- cutoffLimit— (選擇性) 擷取的加入合作伙伴數目的截止限制。雖然沒有任何預設限制，但您可以在執行聯結操作時將此引數設為 1 以實作 FILTER (NOT) EXISTS 子句，其足以證明或駁斥聯結夥伴的存在。

PipelineCountJoin運算子

該運算子為 PipelineJoin 的變體；其不會執行聯結操作，而是計算相符的聯合夥伴數量，並將計數繫結至 countVar 引數指定的變數。

引數

- countVar— (必要) 計數結果 (即連接夥伴的數目) 應該繫結的變數。
- pattern- (必填) 模式，它採用的形式subject-predicate-object，並可選擇-graph 元組的基礎連接。如果 distinct 為指定的模式值，則聯結僅會從 projectionVars 引數指定的投影變數中擷取不同解決方案，而不是所有相符的解決方案。
- inlineFilters— (選擇性) 要套用至模式中變數的一組篩選器。系統會一併評估模式與這些篩選條件。
- joinType— (必要) 要執行的連接類型：
  - join— 正常聯結，需要在所有共用變數之間完全相符。
  - optional— 使用 SPARQL OPTIONAL 運算子語意的optional聯結。
  - minus— minus 作業會使用 SPARQL MINUS 運算子語意，保留沒有聯結夥伴存在的對應。
  - existence check— 檢查是否存在聯結夥伴，並將existenceCheckResultVar變數繫結至此檢查的結果。

- constraints— (可選) 在連接期間考慮的其他接合約束。不符合這些限制的聯結皆會予以捨棄。
- projectionVars— (選擇性) 投影變數。此引數需與 distinct := true 結合使用，以在指定的一組變數內強制擷取不同投影。
- cutoffLimit— (選擇性) 擷取的加入合作伙伴數目的截止限制。雖然沒有任何預設限制，但您可以在執行聯結操作時將此引數設為 1 以實作 FILTER (NOT) EXISTS 子句，其足以證明或駁斥聯結夥伴的存在。

PipelinedHashIndexJoin運算子

這是一個all-in-one構建哈希索引和連接運算符。它需要一個綁定列表，將它們捲軸到哈希索引中，然後將傳入的解決方案與哈希索引聯接。

引數

- sourceType— (必要) 從中取得要儲存在雜湊索引之繫結的來源類型，以下其中一種：
  - pipeline— 導致將PipelinedHashIndexJoin來自運算子管線中下游運算子的傳入解決方案捲入雜湊索引。
  - binding set— 導PipelinedHashIndexJoin致將引sourceBindingSet數指定的固定繫結集排存到雜湊索引中。
- sourceSubQuery — (選擇性) 如果sourceType引數值為pipeline，此引數會指定評估並多工緩衝處理至雜湊索引的子查詢。
- sourceBindingSet — (選擇性) 如果sourceType引數值為binding set，此引數會指定要多工緩衝處理至雜湊索引的靜態繫結集。
- joinType— (必要) 要執行的連接類型：
  - join— 正常聯結，需要在所有共用變數之間完全相符。
  - optional— 使用 SPARQL OPTIONAL 運算子語意的optional聯結。
  - minus— minus 作業會使用 SPARQL MINUS 運算子語意，保留沒有聯結夥伴存在的對應。
  - existence check— 檢查是否存在聯結夥伴，並將existenceCheckResultVar變數繫結至此檢查的結果。
- existenceCheckResultVar— (選用) 僅用joinType於等於其中的連接 existence check (請參閱上面的 JoinType 引數)。

Projection運算子

該運算子會在變數子集上進行投影。流入的解決方案數量會等同於流出的解決方案數量，但解決方案的圖形會因模式設定而異。

模式

- retain— 僅在解決方案中保留由引數指定的變vars數。
- drop— 卸除引數所指定的所有變vars數。

引數

- vars— (必要) 要保留或刪除的變數，視模式設定而定。

PropertyPath運算子

啟用遞迴屬性路徑，例如+或*。Neptune 根據引iterationTemplate數指定的模板實現定點迭代方法。系統會針對每個固定點反覆運算項目將已知的左側或右側變數繫結至範本，直到找不到任何新的解決方案為止。

引數

- iterationTemplate-(必要) 用來實作定點版序的子查詢範本名稱。
- leftTerm— (必要) 屬性路徑左側的術語 (變數或常數)。
- rightTerm— (必要) 屬性路徑右側的術語 (變數或常數)。
- lowerBound-(必要) 定點反覆運算的下限 (0適用於*查詢或+查1詢)。

TermResolution運算子

該運算子會將內部字串識別符值轉換為各自對應的外部字串，或將外部字串轉換為內部字串識別符值，視模式而定。

模式

- value2id— 將字詞 (例如常值和 URI) 對應至對應的內部 ID 值 (編碼至內部值)。
- id2value— 將內部 ID 值對應至對應的字詞，例如常值和 URI (解碼內部值)。

引數

- vars— (必要) 指定應對應其字串或內部字串 ID 的變數。

Slice運算子

該運算子會使用 SPARQL LIMIT 和 OFFSET 子句的語意，在傳入解決方案串流上實作配量。

引數

- limit— (選用) 要轉送的解決方案的限制。
- offset— (選用) 評估解決方案以進行轉送的偏移量。

SolutionInjection運算子

該運算子不會接收任何輸入，其會將靜態解決方案插入查詢計劃，並在 solutions 引數中加以記錄。

查詢計劃一律會從插入的靜態項目開始執行。如果可以結合各種來源的靜態繫結，進而從查詢本身衍生取得 (例如從 VALUES 或 BIND 子句) 要插入的靜態解決方案，則 SolutionInjection 運算子就會插入這些衍生的靜態解決方案。在最簡單的情況下，這會反映出外部 VALUES 子句所指示的繫結。

如果不能自查詢衍生取得任何靜態解決方案，SolutionInjection 就會插入空白的解決方案，也就是所謂的通用解決方案。在整個查詢評估過程中，系統會擴展並倍增這個解決方案。

引數

- solutions— (必要) 操作員注入的解決方案的順序。

Sort運算子

該運算子會使用指定排序條件來排列解決方案集。

引數

- sortOrder— (必要) 已排序的變數清單，每個變數都包含 ASC (遞增) 或 DESC (遞減) 識別碼，依序用來排序解決方案集。

VariableAlignment運算子

該運算子會逐一檢查所有解決方案，並針對其中的兩個變數執行校準作業。此處的變數指的是指定的 sourceVar 值和指定的 targetVar 值。

如果解決方案中的 sourceVar 和 targetVar 值相同，系統就會將該變數視為已經過校準並轉送解決方案，然後投射出多餘的 sourceVar。

若變數繫結至不同值，解決方案就會完全遭到排除。

引數

- sourceVar— (必要) 要與目標變數進行比較的來源變數。若在解決方案中成功完成校準，表示兩個變數擁有相同值，且系統會投射出來源變數。
- targetVar— (必要) 與來源變數進行比較的目標變數。即使校準成功，該值仍會保留不變。

海王星explain中的 SPARQL 的局限性

海王星 SPARQL explain 功能的發行具有下列限制。

海王星目前僅支援 SPARQL 選取查詢中的說明

如需其他查詢形式評估程序的相關資訊，例如 ASK、CONSTRUCT、DESCRIBE 和 SPARQL UPDATE 查詢，您可以將這些查詢轉換為 SELECT 查詢。然後，改用 explain 來檢查對應的 SELECT 查詢。

例如，若要取得與 ASK WHERE {...} 查詢相關的 explain 資訊，使用 explain 執行對應的 SELECT WHERE {...} LIMIT 1。

同樣地，對於 CONSTRUCT {...} WHERE {...} 查詢，請捨棄 CONSTRUCT {...} 部分並使用 explain，在第二個 WHERE {...} 子句執行 SELECT 查詢。對第二個 WHERE 子句進行評估通常也揭露處理 CONSTRUCT 查詢的主要挑戰，因為從第二個 WHERE 傳到 CONSTRUCT 範本的解決方案通常只需要直接替換。

解釋運算子在未來版本可能變更

SPARQL explain 運算子及其參數可能在未來版本變更。

解釋輸出在未來版本可能變更

例如，欄標題可能會變更，且可能會將更多欄新增到資料表。

在海王星中使用擴充功能的 SPARQL 聯合查詢 SERVICE

亞馬遜海王星完全支援使用關鍵字的 SPARQL 聯合查詢延伸模組。SERVICE(如需詳細資訊，請參閱 SPARQL 1.1 聯合查詢。)

> **Note**
>
> 此功能從中開始提供版本 1.0.0.0.463.0 (p. 985)。

SERVICE 關鍵字會指示 SPARQL 查詢引擎對遠端 SPARQL 端點執行部分查詢，並撰寫最終查詢結果。只有READ操作是可能的。 WRITE和DELETE操作不受支持。Neptune 只能針對可在其虛擬私有雲 (VPC) 中存取的 SPARQL 端點執行聯合查詢。不過，您也可以在 VPC 中使用反向代理，使外部資料來源可在 VPC 內存取。

Note

使用 SPARQL SERVICE 將查詢聯合至相同 VPC 中的兩個或多個 Neptune 叢集時，必須將安全性群組設定為允許所有這些 Neptune 叢集彼此通訊。

Important

SPARQL 1.1 聯合會在將查詢和參數傳遞至外部 SPARQL 端點時，代您提出服務請求。您有責任驗證外部 SPARQL 端點是否滿足您應用程式的資料處理和安全需求。

海王星聯合查詢的範例

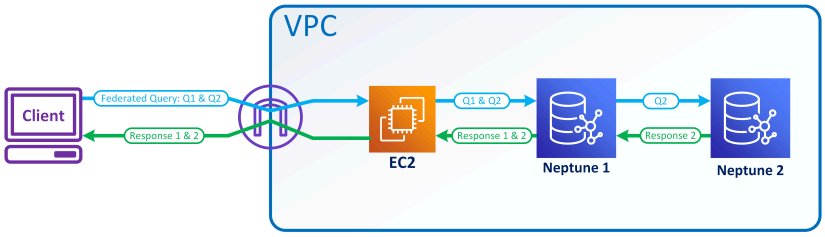
以下簡單範例說明 SPARQL 聯合查詢的運作方式。

假設客戶將下列查詢傳送至海王星 -1 的位置。`http://neptune-1:8182/sparql`

```
SELECT * WHERE {
   ?person rdf:type foaf:Person .
   SERVICE <http://neptune-2:8182/sparql> {
       ?person foaf:knows ?friend .
    }
}
```

1. 海王星 -1 評估第一個查詢模式（Q-1）也就是說*?person rdf:type foaf:Person*，使用結果*?person*在 Q-2（`?person foaf:knows ?friend`）中解析，並將生成的模式轉發到海王星 -2 處。`http://neptune-2:8182/sparql`
2. 海王星 2 評估 Q-2 並將結果發送回海王星 1。
3. Neptune 1 加入了這兩種模式的解決方案，並將結果發送回給客戶。

下圖顯示此流程。



Note

「默認情況下，優化器確定在查詢執行的時間點執行SERVICE指令。您可以使用joinOrder (p. 438)查詢提示覆寫此放置位置。

Neptune 中聯合查詢的存取控制

海王星使用 AWS Identity and Access Management (IAM) 進行身份驗證和授權。聯合查詢的存取控制可能涉及多個 Neptune 資料庫執行個體。這些執行個體可能會有不同的存取控制需求。在某些情況下，這會限制您進行聯合查詢的能力。

考慮上一節中介紹的簡單示例。海王星 1 號稱海王星 -2 與它被稱為相同的憑據。

- 如果 Neptune-1 需要 IAM 身份驗證和授權，但海王星 2 不需要，那麼您只需要適用於海王星 1 的 IAM 許可即可進行聯合查詢。
- 如果 Neptune-1 和 Neptune 2 都需要 IAM 身份驗證和授權，則需要為兩個數據庫附加 IAM 許可以進行聯合查詢。

- 但是，如果海王星 -1 不啟用 IAM，但是海王星 -2 是，則無法進行聯合查詢。原因是 Neptune-1 無法檢索您的 IAM 憑據並將其傳遞給 Neptune 2 以授權查詢的第二部分。

海王星的圖形可視化工具

除了 Neptune 圖形筆記本內建的視覺化功能之外，您還可以使用AWS合作夥伴和協力廠商所建置的解決方案，將 (p. 47)儲存在 Neptune 中的資料視覺化。

複雜的圖形視覺化可協助組織中的資料科學家、管理員和其他角色以互動方式探索圖形資料，而不必知道如何撰寫複雜的查詢。

主題

- 開放原始碼的圖形瀏覽器 (p. 476)
- 湯姆索亞軟件 (p. 480)
- 劍橋情報局 (p. 482)
- 石墨 (p. 484)
- 隱喻 (p. 485)
- G.V () (p. 488)

開放原始碼的圖形瀏覽器

圖形資源管理器是一種用於圖形數據的開源低代碼可視化探索工具，根據 Apehe- 2.0 許可證提供。它可讓您瀏覽圖形資料庫中的標籤屬性圖 (LPG) 或資源描述架構 (RDF) 資料，而不需要撰寫圖形查詢。Graph-Explorer 旨在幫助數據科學家，業務分析師和組織中的其他角色以互動方式探索圖形數據，而無需學習圖形查詢語言。

圖形資源管理器提供了一個基於反應的 Web 應用程序，可以部署為可視化圖形數據的容器。您可以連接到亞馬遜海王星或其他圖形資料庫，這些資料庫提供阿帕奇TinkerPop格林或 SPARQL 1.1 端點。

- 您可以使用多面過濾器快速查看數據的摘要，或通過在搜索欄中鍵入文本來搜索數據。
- 您也可以互動式探索節點和邊緣連接。您可以檢視節點鄰近項目以查看物件彼此之間的關聯性，然後向下鑽研以視覺化檢查邊緣和屬性。
- 您也可以自訂圖表配置、顏色、圖示，以及要針對節點和邊顯示的預設屬性。對於 RDF 圖，您也可以自訂資源 URI 的命名空間。
- 對於涉及圖形資料的報表和簡報，您可以設定和儲存您建立的檢視為高解析度 PNG 格式。您也可以將相關資料下載到 CSV 或 JSON 檔案中，以便進一步處理。

若要開始使用，您可以建立圖形資源管理器 Docker 映像，然後在本機電腦或託管服務 (例如 Amazon 彈性運算雲端 (Amazon EC2) 或 Amazon 彈性容器服務 (Amazon ECS) 上執行，如圖形資源管理器專案讀取我的入門章節所述。GitHub

下一節提供在 Amazon ECS 上執行圖形總管的step-by-step指示：AWS Fargate

如何在亞馬遜 ECS 上運行圖形瀏覽器AWS Fargate並連接到海王星

下列步驟可讓您在 Amazon ECS 的 Fargate 上設定圖形瀏覽器，並連線到執行中的海王星資料庫。

1. 建立新的 IAM 角色並將這些政策附加到該角色：

   - 亞馬遜 TaskExecutionRolePolicy
   - CloudWatchLogsFullAccess

   保持角色名稱方便在一分鐘內使用。
2. 建立具有基礎設施設定為 FARGATE 的 Amazon ECS 叢集，並使用下列聯網選項：

   - VPC：設定為您的海王星資料庫所在的 VPC。
   - Subnets：設置為該 VPC 的公共子網（刪除所有其他子網）。
3. 建立新的 JSON 工作定義，如下所示：

```
{
  "family": "explorer-test",
  "containerDefinitions": [
    {
      "name": "graph-explorer",
      "image": "public.ecr.aws/neptune/graph-explorer:latest",
      "cpu": 0,
      "portMappings": [
        {
          "name": "graph-explorer-80-tcp",
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp",
          "appProtocol": "http"
        },
        {
          "name": "graph-explorer-443-tcp",
          "containerPort": 443,
          "hostPort": 443,
          "protocol": "tcp",
          "appProtocol": "http"
        }
      ],
      "essential": true,
      "environment": [
        {
          "name": "HOST",
          "value": "localhost"
        }
      ],
      "mountPoints": [],
      "volumesFrom": [],
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-create-group": "true",
          "awslogs-group": "/ecs/graph-explorer",
          "awslogs-region": "{region}",
          "awslogs-stream-prefix": "ecs"
        }
      }
    }
  ],
  "taskRoleArn": "arn:aws:iam::{account_no}:role/{role_name_from_step_1}",
  "executionRoleArn": "arn:aws:iam::{account_no}:role/{role_name_from_step_1}",
  "networkMode": "awsvpc",
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "1024",
```

```
  "memory": "3072",
  "runtimePlatform": {
    "cpuArchitecture": "X86_64",
    "operatingSystemFamily": "LINUX"
  }
}
```

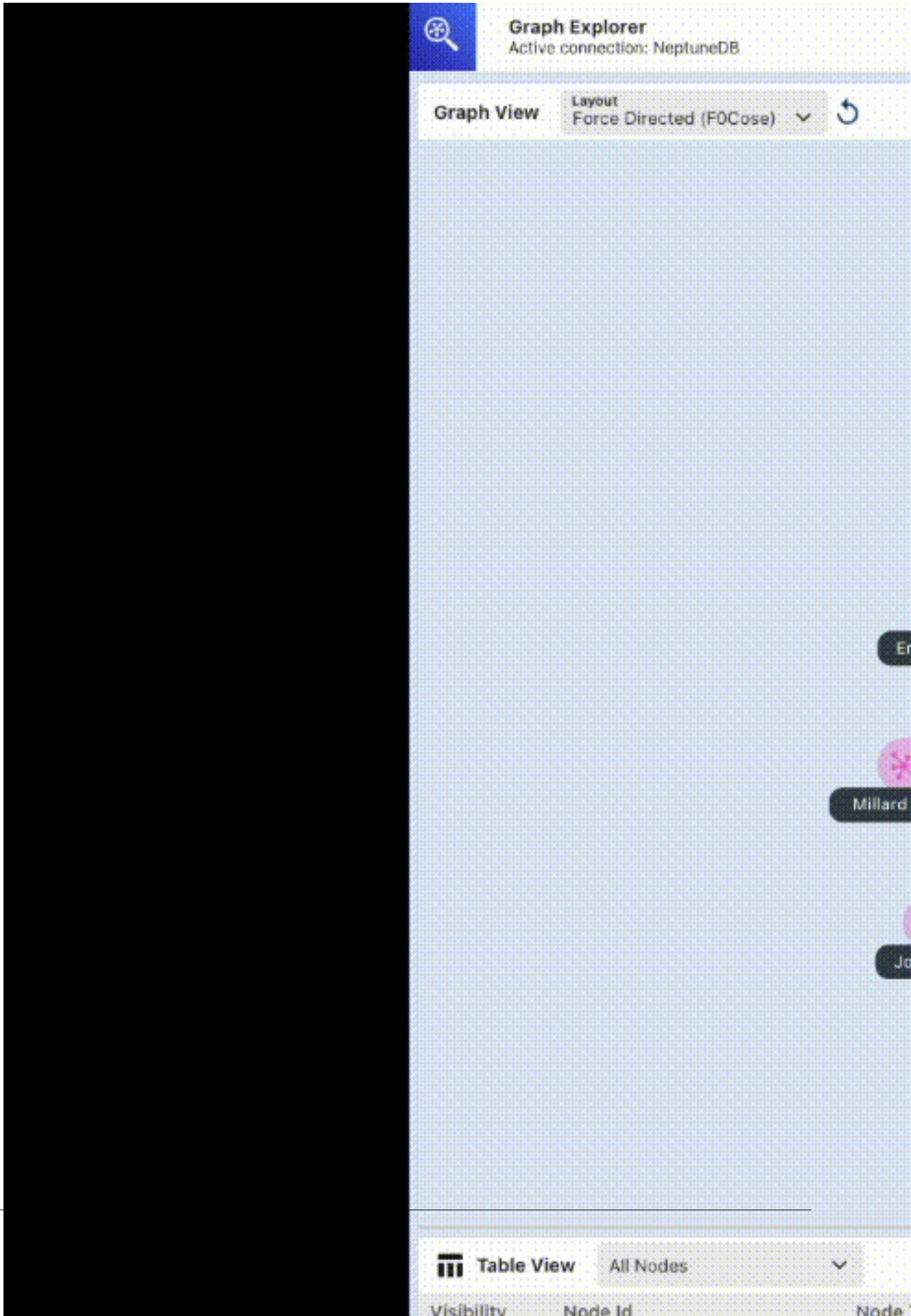
4. 使用預設設定啟動新工作，但下列欄位除外：
   - Environment (環境)
     - 計算選項 => 啟動類型
   - Deployment configuration (部署組態)
     - 應用程式類型 => 工作
     - 家庭 =>*#### JSON #####*
     - 修訂版 =>*####*
   - 聯網
     - VPC => *(######## VPC*)
     - 子網路 => *(# VPC ###### – #####*他子網路)
     - 安全組 => 創建一個新的安全組
     - 安全組名 => 圖形資源管理器
     - 安全性群組描述 = 用於存取圖形總管的安全性群組
     - 安全組的入站規則 =>
       1. 80 任何地方
       2. 443 任何地方
5. 選取 Create (建立)。
6. 工作開始後，複製執行中工作的公用 IP，然後瀏覽至：`https://`*(your public IP)*`/explorer`。
7. 接受使用已產生的無法辨識憑證的風險，或將其新增至您的金鑰鏈。
8. 現在您可以新增至海王星的連線。為屬性圖 (LPG) 或 RDF 建立新連線，並設定下列欄位：

```
Using proxy server => true
Public or Proxy Endpoint => https://(your public IP address)
Graph connection URL => https://(your Neptune endpoint):8182
```

您現在應該已連線。

圖形瀏覽器示範

這個簡短的視頻讓您了解如何使用圖形資源管理器輕鬆地將圖形數據視覺化：


Graph Explorer
Active connection: NeptuneDB
Graph View
Layout
Force Directed (FOCose)
Millard
Table View
All Nodes
Visibility
Node Id

湯姆索亞軟件

湯姆·索亞視角是一個低代碼的圖形和數據可視化和分析開發平台，用於存儲在亞馬遜海王星中的數據。整合式設計與預覽介面，以及廣泛的 API 程式庫，可讓您快速建立自訂的生產品質視覺化應用程式。透過設計 point-and-click師介面和 30 種內建分析演算法，您可以設計和開發應用程式，以深入瞭解來自數十個來源聯合的資料。

Tom Sawyer 圖形資料庫瀏覽器可讓您輕鬆視覺化和分析亞馬遜海王星中的資料。您可以在查詢語言或結構描述的情況下查看和理解資料中的連線。您可以在沒有技術知識的情況下與資料互動，只要載入所選節點的鄰近項目，並以任何您需要的方向建立視覺效果即可。您還可以利用五種獨特的圖形佈局，以提供最大意義的方式顯示圖形，並且可以應用集中性，聚類和路徑查找分析來顯示以前看不到的模式。若要查看圖形資料庫瀏覽器與 Neptune 整合的範例，請查看這篇部落格文章。若要開始免費試用圖形資料庫瀏覽器，請造訪AWS市集。


www.tom

劍橋情報局

劍橋智慧提供資料視覺化技術，用於探索和理解 Amazon Neptune 資料。圖形視覺化工具包（KeyLines適用於JavaScript開發人員和 ReGraphReact 開發人員）提供了一種簡單的方法來為 Web 應用程序構建高度互動和可定制的工具。這些工具包利用 WebGL 和 HTML5 Canvas 實現快速性能，它們支持高級圖形分析功能，並將靈活性和可擴展性與安全，強大的架構結合在一起。這些 SDK 與海王星小鬼和 RDF 數據的工作。

查看這些有關 Gremlin 資料、SPARQL 資料和海王星架構的整合教學課程。

以下是範例KeyLines視覺效果：

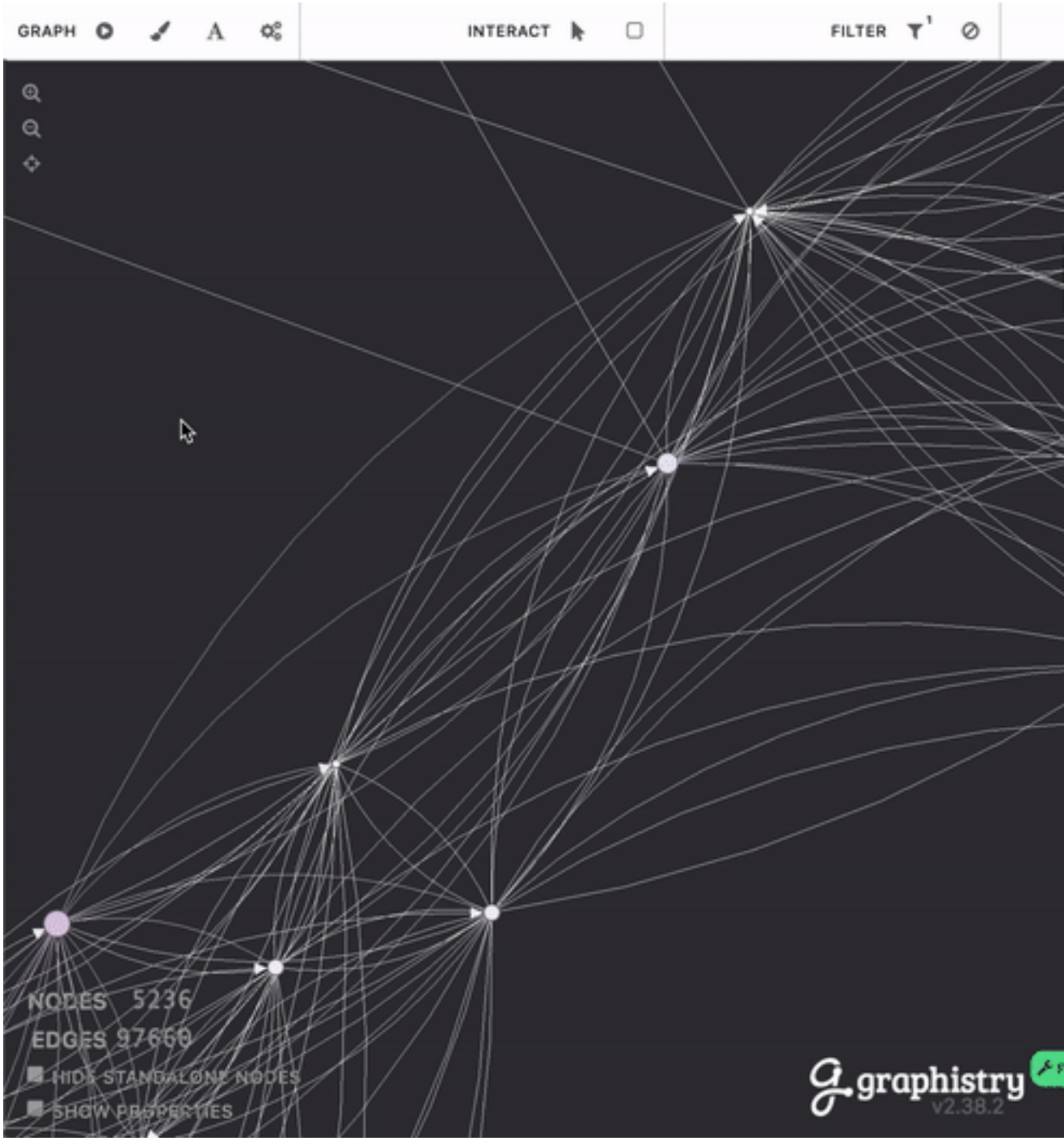


石墨

Graphstry 是一個視覺化圖形智慧平台，利用 GPU 加速提供豐富的視覺體驗。團隊可以使用各種功能在 Graphstry 上進行協作，從無程式碼探索檔案和資料庫，到共用 Jupyter 筆記本和 Streamlit 儀表板，到在您自己的應用程式中使用內嵌 API。

您只需設定和啟動並修改幾行程式碼，就可以開始使用低編碼的完全互動式儀表板。graph-app-kit查看此博客文章，了解使用石墨和海王星創建第一個儀表板的演練。您也可以嘗試海王星PyGraphistry示範。PyGraphistry是筆記本的 Python 視覺化圖形分析程式庫。查看此教程筆記本以獲取 Neptune PyGraphistry 演示。

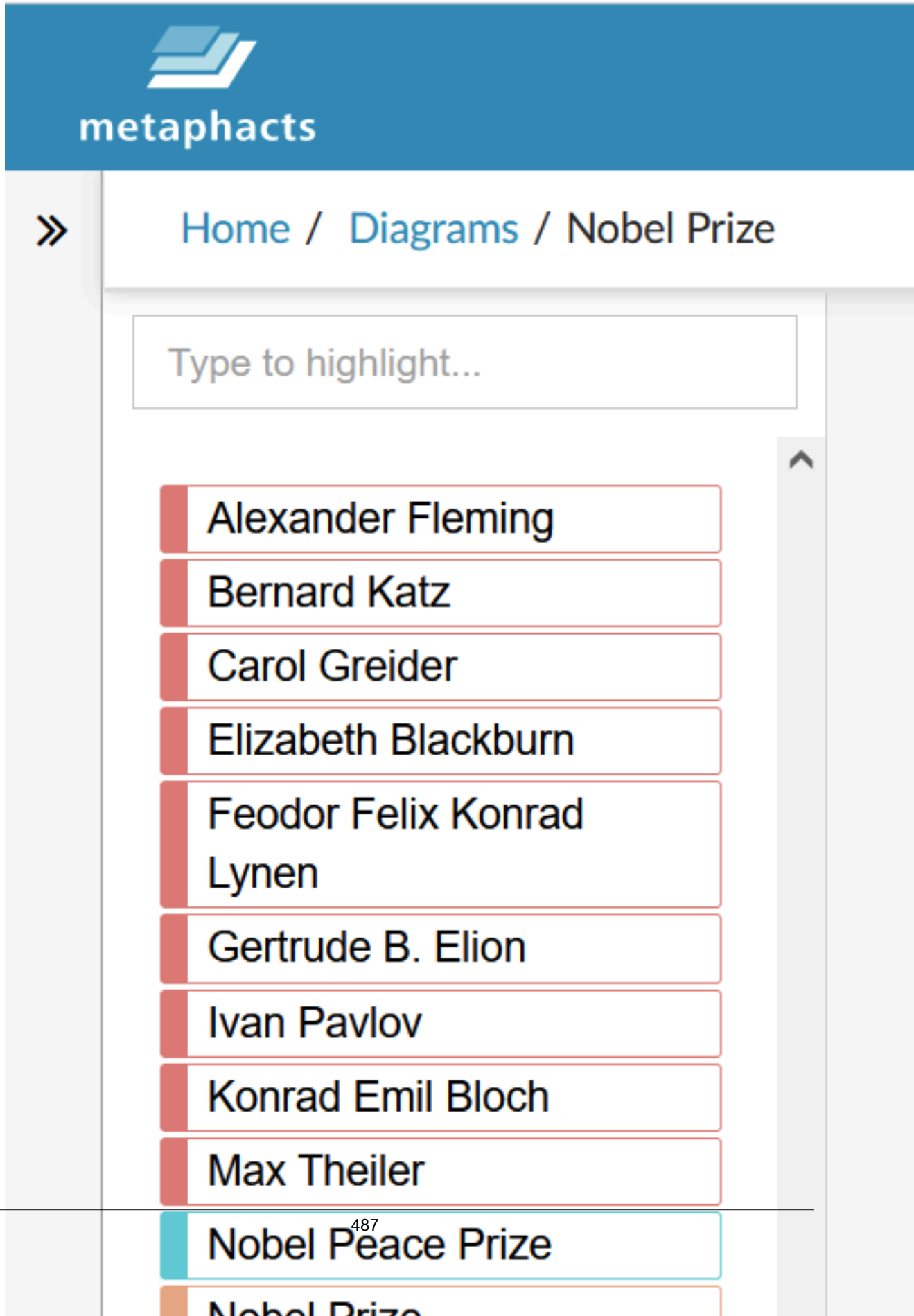
要開始使用，請訪問AWS市場中的石墨師。



隱喻

隱喻提供了一個靈活的開放平台，用於描述和查詢圖形數據，以及可視化和與知識圖交互。使用隱喻，您可以建立互動式 Web 應用程式，例如視覺化和儀表板上的知識圖形在海王星使用 RDF 資料模型。隱喻平台支援低程式碼開發體驗，其中包含用於資料載入的 UI、支援 OWL 和 SHACL 的視覺本體論建模介面、SPARQL 查詢 UI 和查詢型錄，以及用於圖形探索、視覺化、搜尋和撰寫的豐富 Web 元件集。

這是一個隱喻可視化示例：


metaphacts
Home / Diagrams / Nobel Prize
Type to highlight...
Alexander Fleming
Bernard Katz
Carol Greider
Elizabeth Blackburn
Feodor Felix Konrad Lynen
Gertrude B. Elion
Ivan Pavlov
Konrad Emil Bloch
Max Theiler
Nobel Peace Prize

該平台專為工程、製造、製藥、生命科學、金融、保險等領域而設計並有效地使用。要查看示例解決方案架構，請查看此博客文章。

要開始免費試用隱喻，請訪問市場。AWS

G.V ()

G.V () 是面向開發人員和數據分析師的 Gremlin 集成開發環境（IDE）工具。使用它，您可以在 Neptune 中以交互方式查詢，可視化和更新圖形數據。您可以使用 G.V () 內建的自動完成功能撰寫查詢，該功能會根據您的圖形資料模型提供建議和文件。您也可以使用 Gremlin 查詢除錯功能，深入撰寫、偵錯、測試和分析圖形遍歷程序。

使用 G.V () 的資料瀏覽器，您可以瀏覽和修改圖形，以快速建構新的圖形結構並維護現有的圖形結構。G.V () 為查詢結果提供多種視覺化格式，可協助您解譯查詢輸出並以互動方式瀏覽圖形。這些包括表格，圖形，JSON 和 Gemlin 控制台輸出格式。

要開始免費試用 G.V ()，請訪問 G.V () 網站，您還可以從文檔中了解更多信息。您也可以觀看此視頻，該視頻演示了 Gremlin 查詢調試功能。

以下是 G.V () 視覺效果範例：

從 Neptune 資料庫叢集匯出資料

從 Neptune DB 叢集匯出資料的好方法有幾種：

- 對於少量數據，只需使用查詢或查詢的結果即可。
- 對於 RDF 數據，圖形存儲協議（GSP）(p. 452)可以使導出變得容易。例如：

```
curl --request GET \
  'https://your-neptune-endpoint:port/sparql/gsp/?graph=http%3A//www.example.com/named/
graph'
```

- 還有一個功能強大且靈活的開源工具，用於導出 Neptune 數據，即neptune-export。以下章節描述此工具及其使用方式。

主題

- 使用 neptune-export (p. 490)
- 使用 Neptune-出口服務導出 Neptune 數據 (p. 491)
- 使用命neptune-export令列工具來從 Neptune 導出數據 (p. 496)
- 由海王星導出的文件-導出和neptune-export (p. 499)
- 用於控制 Neptune 匯出程序的參數 (p. 500)
- 疑難排解 Neptune 匯出程序 (p. 513)

使用 neptune-export

您可以使neptune-export用開放原始碼碼碼碼碼碼碼碼碼碼碼碼碼碼頭

- 作為海王星-出口服務 (p. 491)。 當您使用 Neptune-匯出服務從 Neptune 匯出資料時，您可以透過 REST API 觸發和監控匯出工作。
- 作為 **neptune-export**Java 命令行實用程序 (p. 496)。 若要使用此命令列工具匯出 Neptune 資料，您必須在可存取 Neptune DB 叢集的環境中執行它。

海王星-匯出口服務和 Amazon Simple Storage Service (Amazon S3) 對 Amazon Simple Storage Service (Amazon S3) 發佈到 Amazon Simple Storage Service (Amazon S3) 加密。neptune-export SSE-S3

> Note
>
> 最佳實務是在所有 Amazon S3 儲存貯體上啟用存取記錄，讓您稽核對這些儲存貯體的所有存取。

如果您嘗試從 Neptune DB 叢集匯出資料，而該叢集的資料在匯出發生時資料正在變更，則無法保證匯出資料的一致性。也就是說，如果您的叢集在匯出工作正在進行時為寫入流量提供服務，則匯出的資料可能會有不一致的情況。無論您是從叢集中的主要執行個體匯出，還是從一或多個僅供讀取複本匯出，都是如此。

為了確保匯出的資料一致，最好從資料庫叢集的複製 (p. 564)中匯出。這兩者都為導出工具提供了數據的靜態版本，並確保導出作業不會減慢原始數據庫集群中的查詢速度。

為了簡化此操作，您可以指示您要在觸發匯出工作時複製來源資料庫叢集。如果這樣做，匯出程序會自動建立翻製，並將其用於匯出，然後在匯出完成時將其刪除。

使用 Neptune-出口服務導出 Neptune 數據

您可以使用下列步驟，使用 Neptune-匯出服務，將資料從 Neptune 叢集匯出至 Amazon S3：

安裝海王星-出口服務

使用AWS CloudFormation範本建立堆疊：

安裝海王星-出口服務

1. 選擇下表中的其中一個「啟動堆疊」按鈕，在AWS CloudFormation主控台上啟動堆疊：AWS CloudFormation

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (雪梨)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國東部)	檢視	在設計工具中檢視	Launch Stack

2. 在 Select Template (選取範本) 頁面上，請選擇 Next (下一步)。
3. 在「指定詳細資訊」頁面上，設定範本的下列參數：

   - **VPC**— 設置 Neptune-出口服務的最簡單方法是將其安裝在與 Neptune 數據庫相同的 Amazon VPC 中。如果要將其安裝在單獨的 VPC 中，則可以使用 VPC 對等互連在 Neptune DB 叢集的 VPC 與 Neptune-匯出服務 VPC 之間建立連線。
   - **Subnet1**— 海王星-匯出服務必須安裝在 VPC 中的子網路中，該子網路允許從子網路到網際網路的輸出 IPv4 HTTPS 流量。這樣，海王星-導出服務可以調用 AWS Batch API 來創建和運行導出任務。

     如果您使用 Neptune 文件建立資料庫叢集 (p. 54)頁面上的CloudFormation範本建立 Neptune 叢集，則可以使用該堆疊的`PrivateSubnet1`和`PrivateSubnet2`輸出來填入此參數和下一個參數。
   - **Subnet2**— VPC 中允許傳出 IPv4 HTTPS 流量從子網際網際網際

6. 在 [檢閱] 頁面上，選取第一個核取方塊，以確認AWS CloudFormation將建立 IAM 資源。選取第二個核取方塊，確認新堆疊 CAPABILITY_AUTO_EXPAND。

   Note

   CAPABILITY_AUTO_EXPAND 明確確認在建立堆疊時，無需事先檢閱，即可擴充巨集。使用者通常會在處理過的範本中建立變更集，以便在實際建立堆疊之前，檢閱巨集所做的變更。如需詳細資訊，請參閱 AWS CloudFormationCreateStack API。

   然後選擇 Create (建立)。

允許從 Neptune-出口訪問 Neptune

Neptune-導出安裝完成後，更新您的 Neptune VPC 安全組以允許從 Neptune (p. 62)-導出訪問。建立「海王星-匯出」AWS CloudFormation 堆疊後，「輸出」索引標籤會包含 ID。NeptuneExportSecurityGroup更新您的 Neptune VPC 安全性群組，以允許從這個海王星-匯出安全性群組進行存取。

啟用從以 VPC 人雲端為基礎的 EC2 執行個體存取海王星匯出端點

如果您將 Neptune-匯出端點設定為僅 VPC，則只能從安裝了 Neptune-匯出服務的 VPC 中存取它。若要允許從 VPC 中進行海王星匯出 API 呼叫的 Amazon EC2 執行個體進行連線，請將AWS CloudFormation堆疊NeptuneExportSecurityGroup建立的連線到該 Amazon EC2 執行個體。

使用海王星-匯出 API 執行海王星-匯出工作

AWS CloudFormation堆疊的 [輸出] 索引標籤也包含NeptuneExportApiUri。每當您將請求發送到海王星-導出端點時，請使用此 URI。

執行匯出工作

- 請確定執行匯出的使用者或角色已獲授與execute-api:Invoke權限。
- 如果true在安裝海王星-導出時將EnableIAM參數設置為AWS CloudFormation堆棧中，則需要將所有請求Sigv4簽署到海王星-導出 API。我們建議您使用 awscurl 向 API 發出請求。這裡的所有示例都假設 IAM 身份驗證已啟用。
- 如果true在安裝海王星-匯出時將VPCOnly參數設定為AWS CloudFormation堆疊中，則必須從 VPC 內呼叫海王星-匯出 API，通常來自位於 VPC 中的 Amazon EC2 執行個體。

要開始導出數據，請使用並請求參數command和endpoint導出參數向NeptuneExportApiUri端點發送outputS3Path請求。

以下是將屬性圖資料從 Neptune 匯出屬性圖資料並將其發佈至 Amazon S3 的請求範例：

```
curl \
  (your NeptuneExportApiUri) \
  -X POST \
  -H 'Content-Type: application/json' \
  -d '{
        "command": "export-pg",
        "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
        "params": { "endpoint": "(your Neptune endpoint DNS name)" }
      }'
```

同樣地，以下是將 RDF 資料從 Neptune 匯出到 Amazon S3 的請求範例：

```
curl \
  (your NeptuneExportApiUri) \
  -X POST \
  -H 'Content-Type: application/json' \
  -d '{
        "command": "export-rdf",
        "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
        "params": { "endpoint": "(your Neptune endpoint DNS name)" }
      }'
```

如果您省略了 command request 參數，依預設，Neptune-匯出會嘗試從 Neptune 匯出屬性圖資料。

如果上一個命令運行成功，輸出將如下所示：

```
{
  "jobName": "neptune-export-abc12345-1589808577790",
  "jobId": "c86258f7-a9c9-4f8c-8f4c-bbfe76d51c8f"
}
```

監控您剛開始的匯出工作

要監視正在運行的作業，請將其 Jobid 附加到您的NeptuneExportApiUri，如下所示：

```
curl \
  (your NeptuneExportApiUri)(the job ID)
```

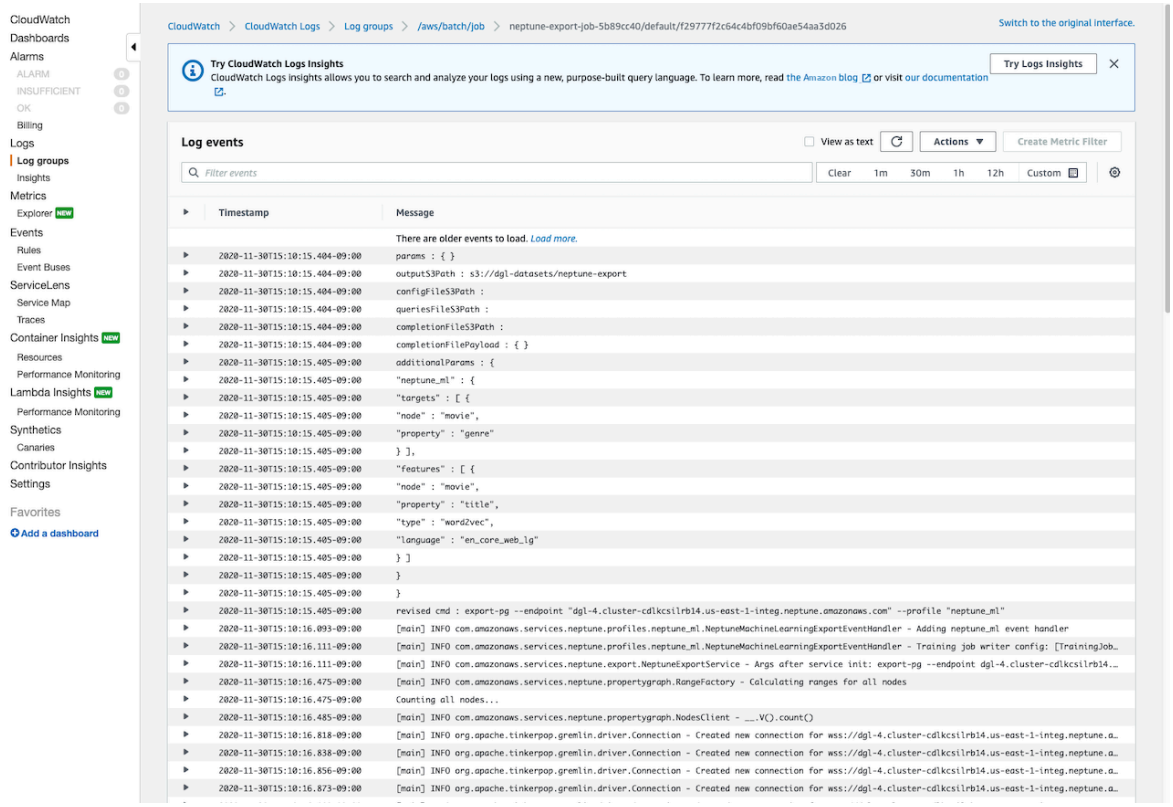
如果服務尚未啟動匯出工作，回應將如下所示：

```
{
  "jobId": "c86258f7-a9c9-4f8c-8f4c-bbfe76d51c8f",
  "status": "pending"
}
```

當您在匯出工作開始之後重複此指令時，回應會如下所示：

```
{
  "jobId": "c86258f7-a9c9-4f8c-8f4c-bbfe76d51c8f",
  "status": "running",
  "logs": "https://us-east-1.console.aws.amazon.com/cloudwatch/home?..."
}
```

如果您使用狀態呼叫所提供的 URI 開啟CloudWatch記錄檔中的記錄檔，則可以詳細監視匯出的進度：



取消執行匯出任務執行中

若要使用取消執行中的匯出工作 AWS Management Console

1. 請在以下位置開啟AWS Batch主控台。 https://console.aws.amazon.com/batch/
2. 選擇 Jobs (任務)。
3. 根據要取消的執行中工作，找出您要取消的工作`jobID`。
4. 選取 [取消工作]。

若要使用 Neptune 匯出 API 取消執行中的匯出工作：

將`HTTP DELETE`請求發送到`NeptuneExportApiUri``jobID`附加的，如下所示：

```
curl -X DELETE \
  (your NeptuneExportApiUri) (the job ID)
```

使用命neptune-export令列工具來從 Neptune 導出數據

您可以使用下列步驟，使用neptune-export命令列公用程式將資料從 Neptune 資料庫叢集匯出到 Amazon S3：

使用neptune-export命令列公用程式的先決條

開始之前

- 有 JDK 的版本 8 — 您需要安裝 Java SE 開發套件 (JDK) 的版本 8。
- 下載海王星導出實用程序-下載並安裝 neptune-export.jar 文件。
- 確定**neptune-export**可以存取 Neptune VPC — 從可存取 Neptune DB 叢集所在的 VPC 的位置執行海王星匯出。

  例如，您可以在 Neptune VPC 內的 Amazon EC2 執行個體上執行，或是在與 Neptune VPC 對等的個別 VPC 中執行，或在單獨的防禦主機上執行。
- 確定 VPC 安全群組授與存取權**neptune-export** — 檢查連接至 Neptune VPC 的 VPC 安全性群組是否允許從與neptune-export環境關聯的 IP 位址或安全性群組存取您的資料庫叢集。
- 設定必要的 IAM 許可 — 如果您的資料庫已啟用AWS Identity and Access Management (IAM) 資料庫身份驗證，請確保neptune-export執行所在的角色與允許連線 Neptune 的 IAM 政策相關聯。如需 Neptune 政策的資訊，請參閱使用 IAM 政策 (p. 176)。

  如果您想要在查詢請求中使用clusterId export 參數，neptune-export執行所在的角色需要下列 IAM 許可：
  - rds:DescribeDBClusters
  - rds:DescribeDBInstances
  - rds:ListTagsForResource

  如果要從複製的叢集匯出，neptune-export執行所在的角色需要下列 IAM 許可：
  - rds:AddTagsToResource
  - rds:DescribeDBClusters
  - rds:DescribeDBInstances
  - rds:ListTagsForResource
  - rds:DescribeDBClusterParameters
  - rds:DescribeDBParameters
  - rds:ModifyDBParameterGroup
  - rds:ModifyDBClusterParameterGroup
  - rds:RestoreDBClusterToPointInTime
  - rds:DeleteDBInstance
  - rds:DeleteDBClusterParameterGroup
  - rds:DeleteDBParameterGroup
  - rds:DeleteDBCluster
  - rds:CreateDBInstance
  - rds:CreateDBClusterParameterGroup
  - rds:CreateDBParameterGroup

  若要將匯出的資料發佈到 Amazon S3，neptune-export執行所在的角色需要 Amazon S3 位置具有下列 IAM 許可：

  - s3:PutObject
  - s3:PutObjectTagging
  - s3:GetObject
- 設定**SERVICE_REGION**環境變數 — 設定SERVICE_REGION環境變數以識別資料庫叢集所在的區域 (如需區域識別碼清單，請參閱連線至 Neptune (p. 165))。

執行neptune-export公用程式以啟動匯出作業

使用以下命令從命令行運行海王星導出並啟動導出操作：

```
java -jar neptune-export.jar nesvc \
  --root-path (path to a local directory) \
  --json (the JSON file that defines the export)
```

該命令有兩個參數：

開始出口時海王星-出口的參數

- **--root-path**— 本機目錄的路徑，其中匯出檔案在發佈到 Amazon S3 之前會寫入該目錄。
- **--json**— 定義匯出的 JSON 物件。

使用指令行公用程式的範例neptune-export命令

若要直接從來源資料庫叢集匯出屬性圖資料：

```
java -jar neptune-export.jar nesvc \
  --root-path /home/ec2-user/neptune-export \
  --json '{
            "command": "export-pg",
            "outputS3Path" : "s3://(your Amazon S3 bucket)/neptune-export",
            "params": {
              "endpoint" : "(your neptune DB cluster endpoint)"
            }
          }'
```

若要直接從來源資料庫叢集匯出 RDF 資料：

```
java -jar neptune-export.jar nesvc \
  --root-path /home/ec2-user/neptune-export \
  --json '{
            "command": "export-rdf",
            "outputS3Path" : "s3://(your Amazon S3 bucket)/neptune-export",
            "params": {
              "endpoint" : "(your neptune DB cluster endpoint)"
            }
          }'
```

如果您省略command request 參數，neptune-export公用程式預設會從 Neptune 匯出屬性圖資料。

若要從資料庫叢集的複本匯出：

```
java -jar neptune-export.jar nesvc \
  --root-path /home/ec2-user/neptune-export \
  --json '{
```

```
            "command": "export-pg",
            "outputS3Path" : "s3://(your Amazon S3 bucket)/neptune-export",
            "params": {
              "endpoint" : "(your neptune DB cluster endpoint)",
              "cloneCluster" : true
            }
          }'
```

若要使用 IAM 身份驗證從資料庫叢集匯出：

```
java -jar neptune-export.jar nesvc \
  --root-path /home/ec2-user/neptune-export \
  --json '{
            "command": "export-pg",
            "outputS3Path" : "s3://(your Amazon S3 bucket)/neptune-export",
            "params": {
              "endpoint" : "(your neptune DB cluster endpoint)"
              "useIamAuth" : true
            }
          }'
```

由海王星導出的文件-導出和neptune-export

匯出完成後，匯出檔案會發佈到您指定的 Amazon S3 位置。Amazon S3 對所有檔案配置進行加密 (SSE-S3) 進行加密。發佈到 Amazon S3 的資料夾和檔案會根據您要匯出屬性圖表還是 RDF 資料而有所不同。如果您開啟發佈檔案的 Amazon Amazon S3 位置，您會看到下列內容：

Amazon S3 中的匯出的檔案配置

- **nodes/**— 此資料夾包含逗號分隔值 (CSV) 或 JSON 格式的節點資料檔案。

  在 Neptune，節點可以有一個或多個標籤。具有不同單獨標籤的節點（或多個標籤的不同組合）被寫入不同的文件中，這意味著沒有單個文件包含具有不同標籤組合的節點的數據。如果節點有多個標籤，這些標籤會在指定給檔案之前按字母順序排序。
- **edges/**— 此資料夾包含逗號分隔值 (CSV) 或 JSON 格式的邊緣資料檔案。

  與節點檔案一樣，邊緣資料會根據其標籤的組合寫入不同的檔案。為了進行模型訓練，邊緣資料會根據邊緣的標籤以及邊緣的起始節點和結束節點標籤的組合，將邊緣資料指派給不同的檔案。
- **statements/**— 此文件夾包含龜，N-四邊形，N-三倍或 JSON 格式的 RDF 數據文件。
- **config.json**— 此檔案包含匯出程序所推斷的圖形結構描述。
- **lastEventId.json**— 此檔案包含資料庫 Neptune 串流上最後一個事件opNum的commitNum和。只有當您將匯出參數設定為true，且要匯includeLastEventId出資料的資料庫已啟用 Neptune 串流 (p. 586)時，匯出程序才會包含此檔案。

用於控制 Neptune 匯出程序的參數

無論您使用的是 Neptune-Export 服務還是命neptune-export令行公用程式，用於控制匯出的參數大部分都是相同的。它們包含傳遞到海王星-導出端點或命令行neptune-export上的 JSON 對象。

傳入匯出流\ 'b5\ '7b 的物件最多有五個頂層欄位：

```
-d '{
      "command" : "(either export-pg or export-rdf)",
      "outputS3Path" : "s3:/(your Amazon S3 bucket)/(path to the folder for exported
 data)",
      "jobsize" : "(for Neptune-Export service only)",
      "params" : { (a JSON object that contains export-process parameters) },
      "additionalParams": { (a JSON object that contains parameters for training
 configuration) }
    }'
```

內容

- command 參數 (p. 501)
- outputS3Path 參數 (p. 501)
- jobSize 參數 (p. 501)
- 該params對象 (p. 502)
- 該additionalParams對象 (p. 502)
- 匯出params頂層 JSON 物件中的參數欄位 (p. 503)
    - 匯出參數params物件中可能的欄位清單 (p. 503)
        - 所有匯出類型的共用欄位清單 (p. 503)
        - 屬性圖匯出的欄位清單 (p. 503)
        - RDF 匯出的欄位清單 (p. 503)
    - 所有匯出類型通用的欄位 (p. 503)
        - cloneCluster欄位中params (p. 503)
        - cloneClusterInstanceType欄位中params (p. 504)
        - cloneClusterReplicaCount欄位中params (p. 504)
        - clusterId欄位中params (p. 504)
        - endpoint欄位中params (p. 505)
        - endpoints欄位中params (p. 505)
        - profile欄位中params (p. 505)
        - useIamAuth欄位中params (p. 505)
        - includeLastEventId欄位中params (p. 505)
    - 性質圖表匯出的欄位 (p. 506)
        - concurrency欄位中params (p. 506)
        - edgeLabels欄位中params (p. 506)
        - filter欄位中params (p. 506)
        - filterConfigFile欄位中params (p. 507)
        - format用於屬性圖資料的欄位params (p. 507)
        - gremlinFilter欄位中params (p. 507)
        - gremlinNodeFilter欄位中params (p. 508)
        - gremlinEdgeFilter欄位中params (p. 508)
        - nodeLabels欄位中params (p. 508)
        - scope欄位中params (p. 508)

- RDF 匯出的欄位 (p. 509)
  - format用於 RDF 資料的欄位params (p. 509)
  - rdfExportScope欄位中params (p. 509)
  - sparql欄位中params (p. 509)
- 篩選匯出內容的範例 (p. 510)
  - 篩選屬性圖資料的匯出 (p. 510)
    - 使用僅匯scope出邊的範例 (p. 510)
    - 使用nodeLabels和edgeLabels僅匯出具有特定標籤的節點和邊的範例 (p. 510)
    - 使用filter僅匯出指定節點、邊緣和屬性的範例 (p. 510)
    - 使用示例gremlinFilter (p. 511)
    - 使用示例gremlinNodeFilter (p. 511)
    - 使用示例gremlinEdgeFilter (p. 511)
    - 結合filtergremlinNodeFilter、nodeLabels、edgeLabels和scope (p. 511)
  - 過濾 RDF 數據的導出 (p. 512)
    - 使用rdfExportScope和sparql匯出特定邊 (p. 512)

command 參數

command頂層參數決定要匯出屬性圖資料還是 RDF 資料。如果省略command參數，匯出程序預設為匯出性質圖資料。

- **export-pg**— 匯出屬性圖資料。
- **export-rdf**— 匯出 RDF 資料。

outputS3Path 參數

outputS3Path頂層參數為必要參數，且必須包含可將匯出檔案發佈至的 Amazon S3 位置的 URI：

```
"outputS3Path" : "s3://(your Amazon S3 bucket)/(path to output folder)"
```

該值必須以有效的值區名稱開頭s3://，以及值區內的資料夾路徑 (選擇性)。

jobSize 參數

jobSize頂層參數只能與 Neptune-匯出服務搭配使用，而不能與neptune-export指令行公用程式搭配使用，而且是選用的。它可讓您描述正在開始的匯出工作大小，這有助於判斷專用於工作的運算資源數量及其最大並行處理層級。

```
"jobsize" : "(one of four size descriptors)"
```

四個有效的大小描述符是：

- small— 最大並發性：8. 適合高達 10 GB 的映象。
- medium— 最大並發性：32. 適用於最大 100 GB 的存儲容量。
- large-最大並發性：64。適用於超過 100 GB 但小於 1 TB 的存儲容量。
- xlarge— 最大並發性：96. 適用於超過 1 TB 的存儲容量。

依預設，在 Neptune-匯出服務上啟動的匯出會以small工作的形式執行。

匯出的效能不僅取決於jobSize設定，還取決於要匯出的來源資料庫執行個體數目、每個執行個體的大小以及工作的有效並行層級。

對於屬性圖匯出，您可以使用參數設定資料庫執行個體的cloneClusterReplica伯爵 (p. 504)數目，也可以使用concurrency (p. 506)參數來設定工作的有效並行層級。

該params對象

params頂層參數是 JSON 物件，其中包含您用來控制匯出程序本身的參數，如中所述匯出params頂層 JSON 物件中的參數欄位 (p. 503)。params對象中的某些字段特定於屬性圖導出，某些字段特定於 RDF。

該additionalParams對象

additionalParams頂層參數是 JSON 物件，其中包含參數，您可以用來控制在匯出資料之後套用至資料的動作。目前，僅用additionalParams於匯出 Neptune ML (p. 671) 的訓練資料。

匯出params頂層 JSON 物件中的參數欄位

Neptune 匯出params JSON 物件可讓您控制匯出，包括匯出資料的類型和格式。

匯出參數params物件中可能的欄位清單

下面列出了可以顯示在params對象中的所有可能的頂級字段。只有這些欄位的子集才會出現在任何一個物件中。

所有匯出類型的共用欄位清單

- cloneCluster (p. 503)
- cloneClusterInstanceType (p. 504)
- cloneClusterReplicaCount (p. 504)
- clusterId (p. 504)
- endpoint (p. 505)
- endpoints (p. 505)
- profile (p. 505)
- useIamAuth (p. 505)
- includeLastEventId (p. 505)

屬性圖匯出的欄位清單

- concurrency (p. 506)
- edgeLabels (p. 506)
- filter (p. 506)
- filterConfigFile (p. 507)
- gremlinFilter (p. 507)
- gremlinNodeFilter (p. 507)
- gremlinEdgeFilter (p. 507)
- format (p. 507)
- nodeLabels (p. 508)
- scope (p. 508)

RDF 匯出的欄位清單

- format (p. 509)
- rdfExportScope (p. 509)
- sparql (p. 509)

所有匯出類型通用的欄位

cloneCluster欄位中params

(選擇性)。預設：false。

如果cloneCluster參數設定為true，則匯出程序會使用資料庫叢集的快速複製：

```
  "cloneCluster" : true
```

依預設，匯出程序會從您使用、endpoints或clusterId參數指定的資料庫叢集匯出資料。endpoint但是，如果您的資料庫叢集在匯出進行時正在使用中，且資料正在變更，則匯出程序無法保證匯出資料的一致性。

若要確保匯出的資料一致，請改用cloneCluster參數從資料庫叢集的靜態複製匯出。

複製的資料庫叢集會在與來源資料庫叢集相同的 VPC 中建立，並繼承來源的安全群組、子網路群組和 IAM 資料庫驗證設定。匯出完成時，Neptune 會刪除複製的資料庫叢集。

根據預設，複製的資料庫叢集包含與來源資料庫叢集中主要執行個體相同執行個體類型的單一執行個體。您可以使用指定不同的執行個體類型，來變更用於複製資料庫叢集的執行個體類型cloneClusterInstanceType。

> **Note**
>
> 如果您沒有使用此cloneCluster選項，而且要直接從主資料庫叢集匯出，則可能需要增加匯出資料的執行個體逾時。對於大型資料集，逾時應設定為數小時。

cloneClusterInstanceType欄位中params

(選擇性)。

如果cloneCluster參數存在且設定為true，您可以使用cloneClusterInstanceType參數來指定用於複製資料庫叢集的執行處理類型：

根據預設，複製的資料庫叢集包含與來源資料庫叢集中主要執行個體相同執行個體類型的單一執行個體。

```
  "cloneClusterInstanceType" : "(for example, r5.12xlarge)"
```

cloneClusterReplicaCount欄位中params

(選擇性)。

如果cloneCluster參數存在且設定為true，您可以使用cloneClusterReplicaCount參數來指定複製的資料庫叢集中建立的僅供讀取複本執行處理數目：

```
  "cloneClusterReplicaCount" : (for example, 3)
```

根據預設，複製的資料庫叢集由單一主要執行個體組成。此cloneClusterReplicaCount參數可讓您指定應建立多少個僅供讀取複本執行個體。

clusterId欄位中params

(選擇性)。

此clusterId參數會指定要使用的資料庫叢集識別碼：

```
  "clusterId" : "(the ID of your DB cluster)"
```

如果您使用clusterId參數，則匯出程序會使用該資料庫叢集中的所有可用執行個體來擷取資料。

> **Note**
>
> endpointendpoints、和clusterId參數是互斥的。使用其中一個。

endpoint欄位中params

(選擇性)。

用endpoint於指定資料庫叢集中 Neptune 執行個體的端點，匯出程序可以查詢以擷取資料 (請參閱端點連線 (p. 121))。這僅是 DNS 名稱，不包括通訊協定或連接埠：

```
"endpoint" : "(a DNS endpoint of your DB cluster)"
```

使用叢集或執行個體端點，但不使用主讀取器端點。

> Note
>
> endpointendpoints、和clusterId參數是互斥的。使用其中一個。

endpoints欄位中params

(選擇性)。

用endpoints於指定資料庫叢集中的 JSON 端點陣列，匯出程序可以查詢以擷取資料 (請參閱端點連線 (p. 121))。這些只是 DNS 名稱，不包括通訊協定或連接埠：

```
"endpoints": [
  "(one endpoint in your DB cluster)",
  "(another endpoint in your DB cluster)",
  "(a third endpoint in your DB cluster)"
  ]
```

如果叢集中有多個執行個體 (主要和一個或多個僅供讀取複本)，則可以使用endpoints參數將查詢分散到這些端點的清單，以改善匯出效能。

> Note
>
> endpointendpoints、和clusterId參數是互斥的。使用其中一個。

profile欄位中params

(必須匯出 Neptune ML 的訓練資料，除非*neptune_ml*欄位出現在*additionalParams*欄位中)。

此profile參數為特定工作負載提供一組預先設定的參數。目前，導出過程僅支持neptune_ml配置文件

如果您要匯出 Neptune ML 的訓練資料，請將下列參數新增至params物件：

```
"profile" : "neptune_ml"
```

useIamAuth欄位中params

(選擇性)。預設：false。

如果要匯出資料的資料庫已啟用 IAM 身份驗證 (p. 160)，則必須將useIamAuth參數設定包含為true：

```
"useIamAuth" : true
```

includeLastEventId欄位中params

如果您設定為 true，且includeLastEventId要匯出資料的資料庫已啟用 Neptune Streams (p. 586)，則匯出程序會將lastEventId.json檔案寫入您指定的匯出位置。此檔案包含串流opNum中最後一個事件的commitNum與。

```
  "includeLastEventId" : true
```

由匯出處理作業建立的複製資料庫會繼承其父項的串流設定。如果父項已啟用串流，複製也會同樣啟用串流。翻製上的串流內容將反映建立翻製時的父項內容 (包括相同的事件 ID)。

性質圖表匯出的欄位

concurrency欄位中params

(選擇性)。預設：4。

此concurrency參數會指定匯出程序應使用的 parallel 查詢數目：

```
  "concurrency" : (for example, 24)
```

一個很好的指導方針是將並行層級設定為要匯出資料的所有執行個體上 vCPUs 數量的兩倍。例如，一個 r5.xlarge 執行個體具有 4 個 vCPUs。如果您要從 3 個 r5.xlarge 執行個體的叢集匯出，您可以將並行層級設定為 24 (= 3 x 2 x 4)。

如果您使用的是海王星-匯出服務，並行等級會受到 JobSize (p. 501) 設定的限制。例如，小型工作支援並行等級為 8。如果您嘗試使用concurrency參數為小型工作指定並行等級 24，則有效層級會維持在 8。

如果您從複製的叢集匯出，匯出程序會根據複製的執行個體大小和工作大小來計算適當的並行層次。

edgeLabels欄位中params

(選擇性)。

使edgeLabels用僅匯出具有您指定標示的邊：

```
  "edgeLabels" : ["(a label)", "(another label"]
```

JSON 陣列中的每個標籤都必須是一個簡單的標籤。

scope參數的優先順序高於edgeLabels參數，因此如果scope值不包括邊，則edgeLabels參數沒有任何作用。

filter欄位中params

(選擇性)。

用filter於指定僅匯出具有特定標示的節點和/或邊，並篩選為每個節點或邊匯出的性質。

filter對象的一般結構，無論是內聯或在過濾器配置文件中，如下所示：

```
  "filter" : {
    "nodes": [ (array of node label and properties objects) ],
    "edges": [ (array of edge definition an properties objects) ]
  }
```

- **nodes**— 包含以下格式的節點和節點屬性的 JSON 陣列：

```
    "nodes : [
      {
        "label": "(node label)",
```

```
      "properties": [ "(a property name)", "(another property name)", ( ... ) ]
    }
  ]
```

- `label`— 節點的屬性圖標籤或標籤。

  需要一個值，或者，如果節點有多個標籤，則需要一個值的陣列。
- `properties`— 包含您要匯出之節點屬性名稱的陣列。

- **`edges`**— 包含下列格式的邊緣定義 JSON 陣列：

```
  "edges" : [
    {
      "label": "(edge label)",
      "properties": [ "(a property name)", "(another property name)", ( ... ) ]
    }
  ]
```

- `label`— 邊緣的屬性圖表標籤。需要一個值。
- `properties`— 包含您要匯出之邊緣屬性名稱的陣列。

`filterConfigFile`欄位中`params`

(選擇性)。

用於`filterConfigFile`以`filter`參數採用的相同格式指定包含篩選器組態的 JSON 檔案：

```
 "filterConfigFile" : "s3://(your Amazon S3 bucket)/neptune-export/(the name of the JSON
file)"
```

如篩選條件 (p. 506)需`filterConfigFile`檔案的格式，請參閱。

`format`用於屬性圖資料的欄位`params`

(選擇性)。預設值：`csv`(逗號分隔值)

此`format`參數會指定匯出的屬性圖表資料的輸出格式：

```
 "format" : (one of: csv, csvNoHeaders, json, neptuneStreamsJson)
```

- **`csv`**— 逗號分隔值 (CSV) 格式化的輸出，欄標題根據 Gremlin 載入資料格式進行格式 (p. 262)化。
- **`csvNoHeaders`**— 沒有列標題的 CSV 格式數據。
- **`json`**— JSON 格式化的數據。
- **`neptuneStreamsJson`**— 使用格雷姆林 _JSON 變更序列化格式的 JSON 格式 (p. 590)資料。

`gremlinFilter`欄位中`params`

(選擇性)。

此`gremlinFilter`參數可讓您提供 Gremlin 程式碼片段，例如用來篩選節點和邊緣的`has()`步驟：

```
 "gremlinFilter" : (a Gremlin snippet)
```

欄位名稱和字串值應以逸出雙引號括住。對於日期和時間，您可以使用日期時間 (p. 829)方法。

下列範例只會匯出具有日期建立屬性值大於 2021-10-10 的節點和邊緣：

```
  "gremlinFilter" : "has(\"created\", gt(datetime(\"2021-10-10\")))"
```

gremlinNodeFilter欄位中params

(選擇性)。

此gremlinNodeFilter參數可讓您提供用來篩選節點的 Gemlin 程式碼片段，例如has()步驟：

```
  "gremlinNodeFilter" : (a Gremlin snippet)
```

欄位名稱和字串值應以逸出雙引號括住。對於日期和時間，您可以使用日期時間 (p. 829)方法。

下列範例只會匯出具有deleted Boolean 屬性值為的節點true：

```
  "gremlinNodeFilter" : "has(\"deleted\", true)"
```

gremlinEdgeFilter欄位中params

(選擇性)。

此gremlinEdgeFilter參數可讓您提供 Gemlin 程式碼片段，例如用來篩選邊緣的has()步驟：

```
  "gremlinEdgeFilter" : (a Gremlin snippet)
```

欄位名稱和字串值應以逸出雙引號括住。對於日期和時間，您可以使用日期時間 (p. 829)方法。

下列範例僅匯出具有strength數值屬性值為 5 的邊：

```
  "gremlinEdgeFilter" : "has(\"strength\", 5)"
```

nodeLabels欄位中params

(選擇性)。

用nodeLabels於僅匯出具有您指定標籤的節點：

```
  "nodeLabels" : ["(a label)", "(another label"]
```

JSON 陣列中的每個標籤都必須是一個簡單的標籤。

scope參數的優先順序高於nodeLabels參數，因此如果scope值不包含節點，則nodeLabels參數沒有任何作用。

scope欄位中params

(選擇性)。預設：all。

該scope參數指定是否只匯出節點，還是僅匯出邊，還是同時匯出節點和邊：

```
  "scope" : (one of: nodes, edges, or all)
```

- nodes— 僅匯出節點及其屬性。

- edges— 僅匯出邊及其屬性。
- all— 匯出節點和邊及其性質 (預設)。

RDF 匯出的欄位

format用於 RDF 資料的欄位params

(選擇性)。預設：turtle

此format參數會指定匯出之 RDF 資料的輸出格式：

```
"format" : (one of: turtle, nquads, ntriples, neptuneStreamsJson)
```

- **turtle**— 烏龜格式的輸出。
- **nquads**— 沒有列標題的 N-四邊形格式的數據。
- **ntriples**— N-三倍格式化的數據。
- **neptuneStreamsJson**— 使用 SPARQL NQUADS 的 JSON 格式資料會變更序列化格式 (p. 590)。

rdfExportScope欄位中params

(選擇性)。預設：graph。

此rdfExportScope參數會指定 RDF 匯出的範圍：

```
"rdfExportScope" : (one of: graph, edges, or query)
```

- graph— 匯出所有 RDF 資料。
- edges— 僅匯出表示邊緣的三元組。
- query— 匯出使用sparql欄位發出之 SPARQL 查詢所擷取的資料。

sparql欄位中params

(選擇性)。

此sparql參數可讓您指定 SPARQL 查詢以擷取要匯出的資料：

```
"sparql" : (a SPARQL query)
```

如果您使用sparql欄位提供查詢，您也必須將rdfExportScope欄位設定為query。

篩選匯出內容的範例

以下是說明篩選匯出資料的方法的範例。

篩選屬性圖資料的匯出

使用僅匯scope出邊的範例

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "scope": "edges"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

使用nodeLabels和edgeLabels僅匯出具有特定標籤的節點和邊的範例

下列範例中的nodeLabels參數指定只應匯出具有Person標籤或Post標籤的節點。該edgeLabels參數指定僅應匯出帶有likes標籤的邊：

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "nodeLabels": ["Person", "Post"],
    "edgeLabels": ["likes"]
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

使用filter僅匯出指定節點、邊緣和屬性的範例

此範例中的filter物件會匯出country節點code及其typedesc屬性，以及route邊緣及其dist屬性。

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "filter": {
      "nodes": [
        {
          "label": "country",
          "properties": [
            "type",
            "code",
            "desc"
          ]
        }
      ],
      "edges": [
        {
          "label": "route",
          "properties": [
            "dist"
          ]
        }
      ]
```

```
    }
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

使用示例gremlinFilter

此範例使gremlinFilter用僅匯出在 2021-10-10 之後建立的節點和邊緣 (也就是說，其值大於 2021-10-10 的created屬性)：

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "gremlinFilter" : "has(\"created\", gt(datetime(\"2021-10-10\")))"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

使用示例gremlinNodeFilter

此範例使gremlinNodeFilter用僅匯出已刪除的節點 (具有 Booleandeleted 屬性值為的節點true)：

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "gremlinNodeFilter" : "has(\"deleted\", true)"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

使用示例gremlinEdgeFilter

此範例使gremlinEdgeFilter 用僅匯出具有strength數值屬性值為 5 的邊：

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "gremlinEdgeFilter" : "has(\"strength\", 5)"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

結合filtergremlinNodeFilter、nodeLabels、edgeLabels和scope

此範例中的filter物件會匯出：

- country節點code及type其desc屬性
- airport節點icao及code其runways屬性
- route邊緣與它們的dist屬性

gremlinNodeFilter參數會篩選節點，以便只匯出具有code屬性值以 A 開頭的節點。

nodeLabels和edgeLabels參數會進一步限制輸出，以便僅匯出airport節點和route邊。

最後，scope參數會從匯出中消除邊，這樣只會在輸出中保留指定的airport節點。

```
{
  "command": "export-pg",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "filter": {
      "nodes": [
        {
          "label": "airport",
          "properties": [
            "code",
            "icao",
            "runways"
          ]
        },
        {
          "label": "country",
          "properties": [
            "type",
            "code",
            "desc"
          ]
        }
      ],
      "edges": [
        {
          "label": "route",
          "properties": [
            "dist"
          ]
        }
      ]
    },
    "gremlinNodeFilter": "has(\"code\", startingWith(\"A\"))",
    "nodeLabels": [
      "airport"
    ],
    "edgeLabels": [
      "route"
    ],
    "scope": "nodes"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

過濾 RDF 數據的導出

使用rdfExportScope和sparql匯出特定邊

這個例子導出了謂詞是，< http://kelvinlawrence.net/air-routes/objectProperty/route > 其對象不是文字的三元組：

```
{
  "command": "export-rdf",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "rdfExportScope": "query",
    "sparql": "CONSTRUCT { ?s <http://kelvinlawrence.net/air-routes/objectProperty/route> ?
o } WHERE { ?s ?p ?o . FILTER(!isLiteral(?o)) }"
  },
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export"
}
```

疑難排解 Neptune 匯出程序

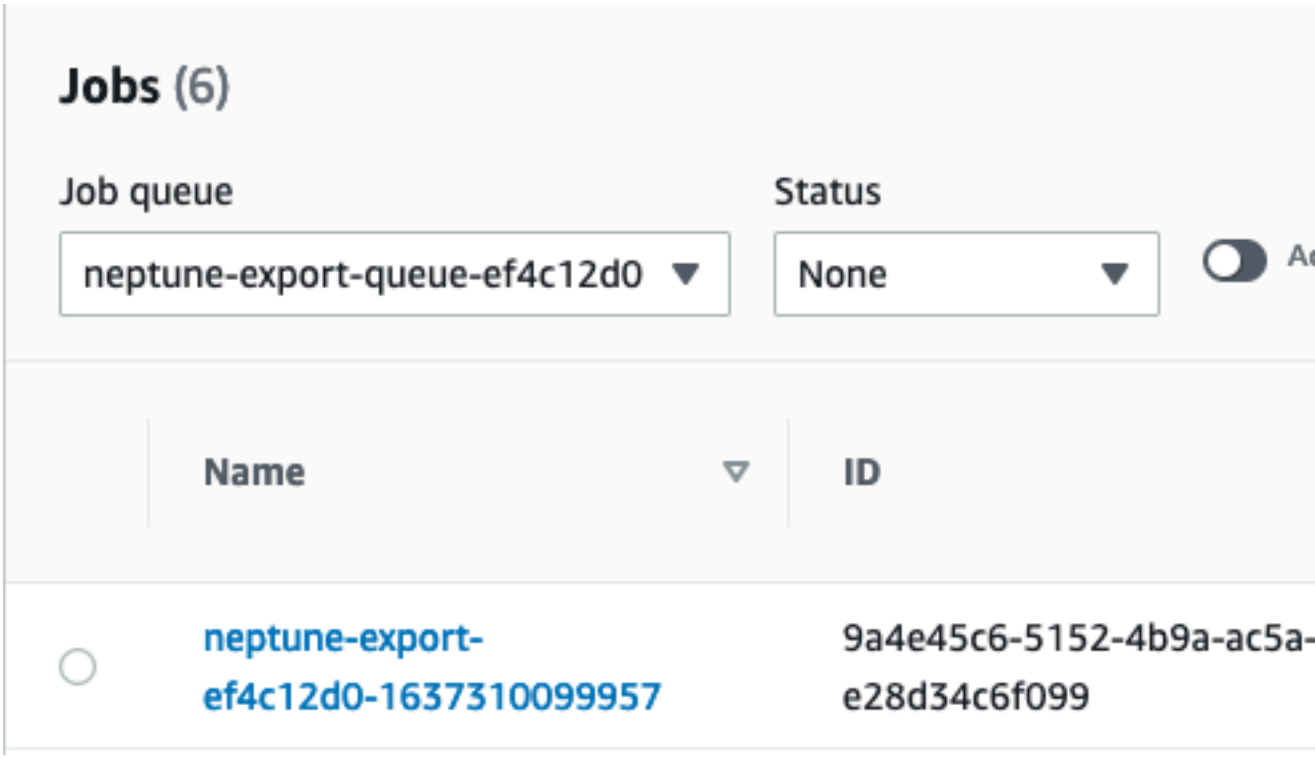
Amazon Neptune 匯出程序用AWS Batch於佈建匯出 Neptune 資料所需的運算和儲存資源。執行匯出時，您可以使用`logs`欄位中的連結存取匯出工作的 CloudWatch 記錄。

不過，執行匯出之AWS Batch工作的 CloudWatch 記錄只有在AWS Batch工作正在執行時才可用。如果 Neptune 匯出報告匯出處於擱置狀態，就不會有記錄連結可供您存取 CloudWatch 記錄檔。如果匯出工作維持在`pending`狀態超過數分鐘，佈建基礎AWS Batch資源可能發生問題。

當匯出工作離開擱置狀態時，您可以檢查其狀態，如下所示：

准准備任務的狀AWS Batch態

1. 請在以下位置開啟AWS Batch主控台。 https://console.aws.amazon.com/batch/
2. 選擇海王星-匯出工作佇列。
3. 尋找名稱與 Neptune 匯出在您開始匯出時所`jobName`傳回的名稱相符的工作。

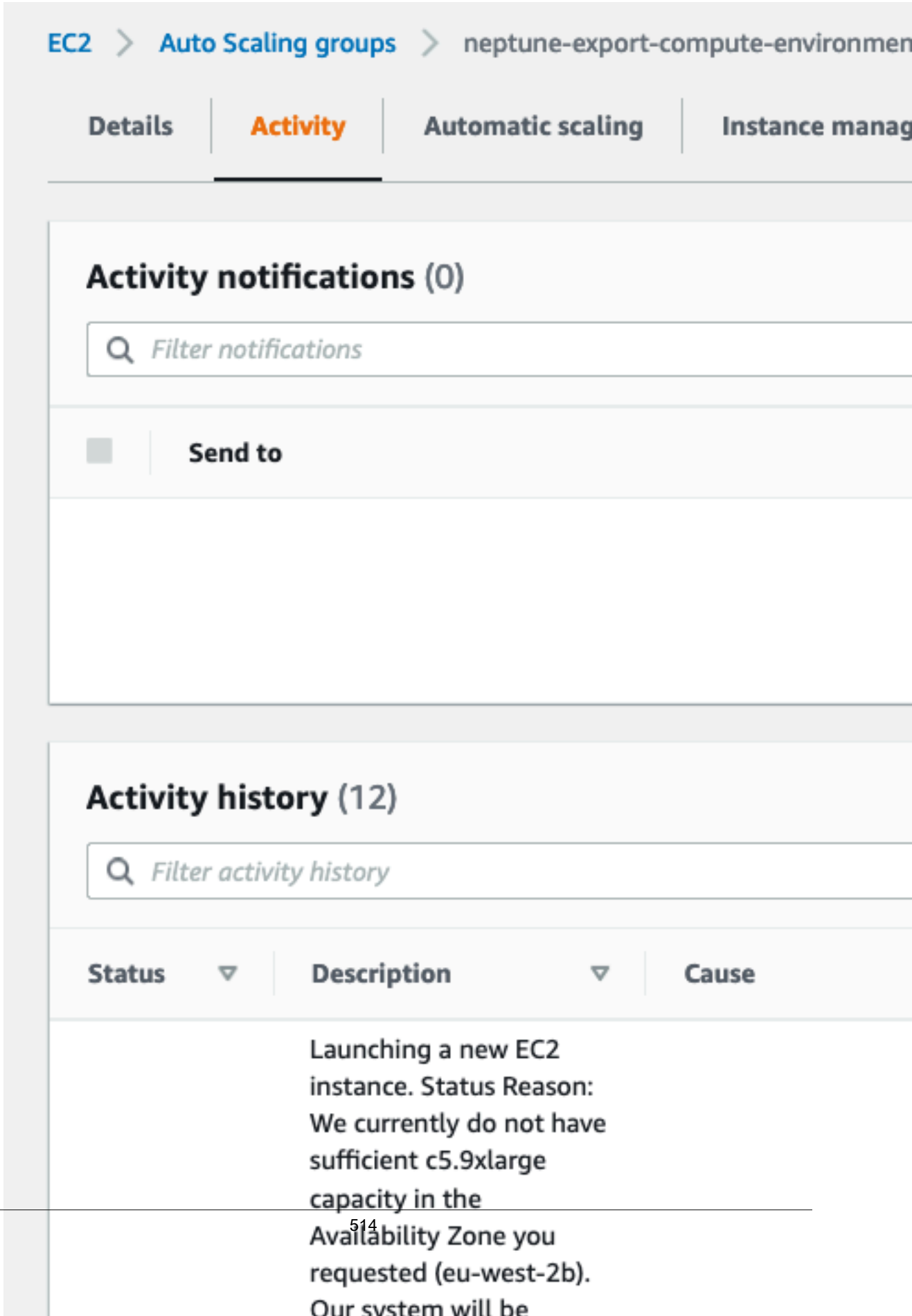


如果任務仍停留在某個`RUNNABLE`狀態，可能是因為聯網或安全問題導致容器執行個體無法加入基礎 Amazon Elastic Container Service (Amazon ECS) 叢集。請參閱本支援文章中有關驗證運算環境的網路和安全性設定章節。

您可以檢查的另一件事是 auto-scaling 問題：

若要檢查AWS Batch運算環境的 Amazon EC2 自動擴展群組

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 為海王星-匯出運算環境選取「Auto Scaling」群組。
3. 開啟「活動」標籤，並檢查活動記錄是否有未成功的事件。


EC2
Auto Scaling groups
neptune-export-compute-environmen
Details
Activity
Automatic scaling
Instance manag
Activity notifications (0)
Filter notifications
Send to
Activity history (12)
Filter activity history
Status
Description
Cause
Launching a new EC2 instance. Status Reason: We currently do not have sufficient c5.9xlarge capacity in the Availability Zone you requested (eu-west-2b). Our system will be

管理亞馬遜海王星數據庫

本節說明如何使用和來管理和維護 Neptune 資料庫叢集AWS CLI。AWS Management Console

Neptune 會在複寫拓撲中連線的資料庫伺服器叢集上運作。因此，管理 Neptune 通常涉及將變更部署到多個伺服器，並確保所有 Neptune 複本都與主要伺服器保持同步。

由於 Neptune 會隨著您的資料增加而透明地擴展基礎儲存體，因此管理 Neptune 只需要進行相對較少的磁碟儲存體管理。同樣地，因為 Neptune 會自動執行連續備份，因此 Neptune 叢集不需要為執行備份進行大量的規劃或停機時間。

主題

- 使用海王星藍/綠解決方案執行藍綠色更新 (p. 516)
- 建立具有 Neptune 許可的 IAM 使用者使用者 (p. 525)
- Amazon Neptune (p. 527)
- Amazon Neptune 參數 (p. 530)
- 使用啟動 Neptune 資料庫叢集AWS Management Console (p. 534)
- 停止和啟動 Amazon Neptune 資料庫叢集 (p. 537)
- 使用快速重設 API 來清空 Amazon Neptune 資料庫叢集 (p. 539)
- 使用主控台建立 Neptune 複本 (p. 543)
- 使用主控台修改 Neptune 資料庫叢集 (p. 544)
- 將 Neptune 複本新增至資料庫叢集 (p. 546)
- Amazon Neptune 中的效能和擴展 (p. 548)
- 自動調整 Amazon Neptune 資料庫叢集中的複本數量 (p. 549)
- 維護 Amazon Neptune 資料庫叢集 (p. 553)
- 亞馬遜海神更新 (p. 555)
- 使用AWS CloudFormation範本更新 Neptune 資料庫叢集的引擎版本 (p. 556)
- Neptune 的數據庫克隆 (p. 564)
- Amazon Neptune 執行個體 (p. 568)

使用海王星藍/綠解決方案執行藍綠色更新

Amazon Neptune 支援的一般引擎升級可能需要應用程式停機，因為資料庫在安裝和驗證更新時無法使用。無論它們是手動還是自動啟動，都是如此。

所謂的藍綠色部署策略可以大大減少停機時間：

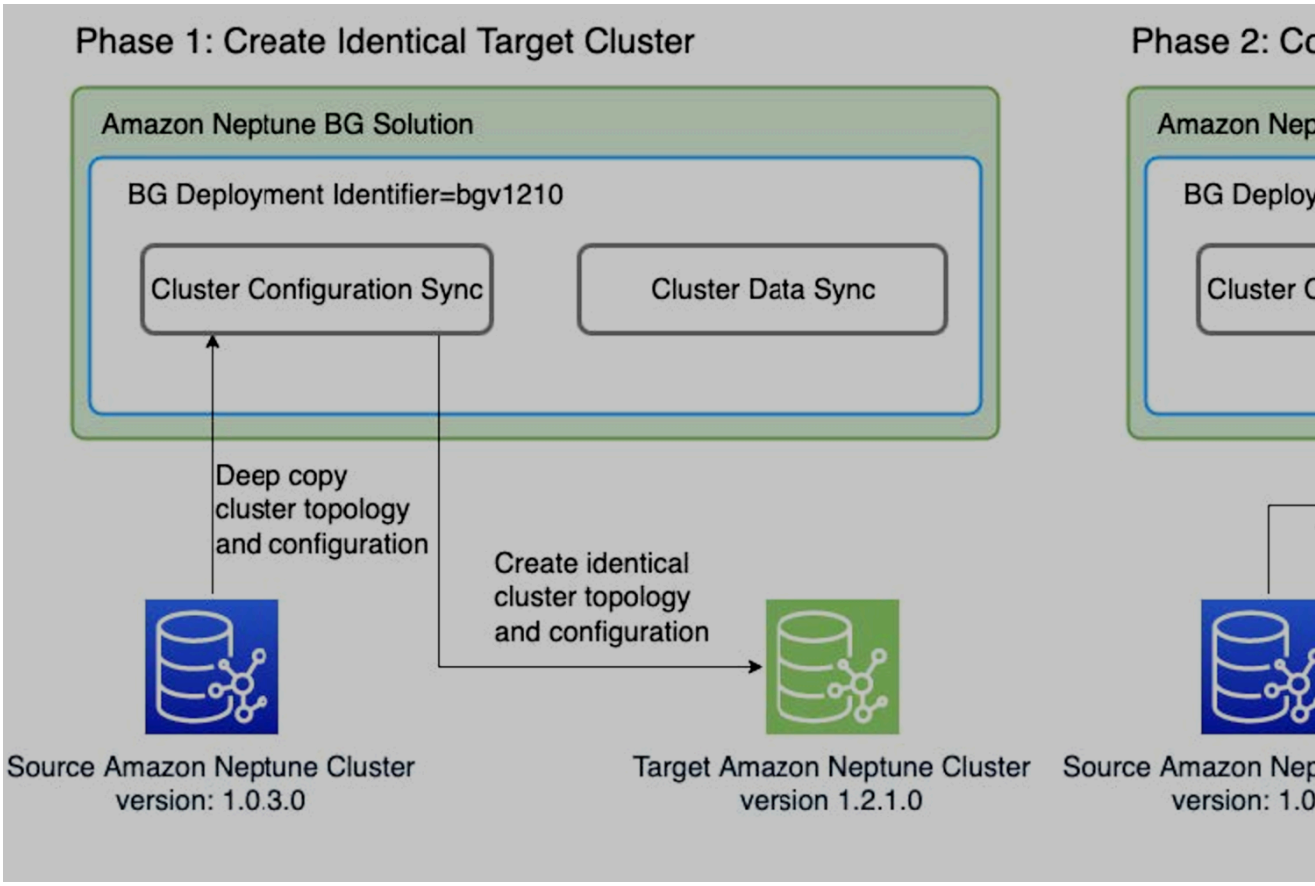
藍綠部署策略

1. 建立與您的生產 (藍色) 資料庫相同的叢集。此副本是您所謂的綠色資料庫，可做為升級的暫存環境。
2. 使用 Neptune 串流，讓綠色叢集中的資料與藍色生產叢集持續同步。
3. 將綠色資料庫升級至您選擇的新引擎版本。
4. 測試並限定更新的綠色資料庫。
5. 一旦您感到滿意，只需將生產端點從藍色叢集切換到綠色叢集，即可進行更新。

   Note

   最好在維護時段期間切換到綠色叢集。
6. 刪除藍色資料庫。

此高級流程圖說明了該過程：



儘管您可以自己手動執行藍綠色升級，但 Neptune 提供了一種名為 Neptune Blu/綠色的解決方案，可通過自動化流程來輕鬆實現。您可以使用AWS CloudFormation堆疊執行 Neptune 藍色/綠色解決方案，以建立與生

產環境同步的綠色測試環境。然後，您可以使用該預備環境來執行次要或主要引擎版本升級、圖形資料模型變更或作業系統更新，所有這些都將停機時間降至最低。

Neptune 藍/綠AWS CloudFormation範本會在與來源 Neptune 藍色資料庫相同的 VPC 中建立 EC2 執行個體，在該處安裝海王星藍色/綠色解決方案，然後執行它。您可以監視記CloudWatch錄檔中的部署進度，如下所述。

執行海王幕藍/綠堆疊的必備條件

啟動海王星藍/綠堆棧之前：

- 請務必在生產 (藍色) 叢集上啟用 Neptune 串流 (p. 586)。
- 藍色叢集中的所有執行個體都必須可用。您可以在 Neptune 主控台或使用 describe-db-instancesAPI 來檢查執行個體狀態。
- 所有執行個體也必須與資料庫叢集參數群組 (p. 527)保持同步。
- 海王星藍/綠解決方案需要藍色叢集所在的虛擬私人雲端中使用 DynamoDB 虛擬私人雲端端點。請參閱使用亞馬遜虛擬私人雲端端點存取 DynamoDB。
- 如果您要更新 Neptune 引擎，請選擇要升級至的引擎版本。請參閱 Amazon Neptune 的引擎版本 (p. 856)。
- 除非您具有管理員權限，否則您必須建立具有建立基礎AWS CloudFormation堆疊所需許可的 IAM 角色。將下列受管理的策略附加到此角色：
  - NeptuneFullAccess
  - AWSLambda_FullAccess
  - CloudWatchFullAccess,
  - AmazonDynamo資料庫 FullAccess
  - 亚马逊 FullAccess
  - 亞馬遜 ReadOnlyAccess
  - AWSStepFunctionsFullAccess
  - CloudWatchEventsFullAccess

  此外，將此自訂原則附加至角色：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:DeleteRole",
        "iam:DeleteRolePolicy",
        "iam:PutRolePolicy",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/*"
```

```
      ],
      "Effect":"Allow"
    },
    {
      "Action": [
        "cloudformation:DescribeStacks",
        "cloudformation:CreateStack"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

如需詳細資訊，請參閱使用 IAM 角色將許可授與在 Amazon EC2 執行個體上執行的應用程式。

海王幕藍/綠解決方案的限制

限制

- Neptune 藍色/綠色解決方案不適用於將叢集復原至較早的引擎版本。
- 如果您使用 SPARQL (RDF)，且BlankNodes在海王星藍/綠解決方案執行時發 (p. 422)生變更，海王星藍/綠解決方案將會造成資料不一致的問題。
- 需要變更應用程式中的硬式編碼連結，以解決新的 (綠色) 資料庫叢集。
- 在從舊的藍色端點切換到新的綠色端點時，必須先暫停藍色叢集，讓綠色叢集趕上任何擱置的寫入。在寫入工作負載繁重的情況下，這可能會造成一些可感知的停機時間。因此，排程切換以避免任何繁重的寫入 (例如大量負載)。
- 如果您想要繼續停止的 Neptune 藍/綠部署，請確定不要刪除部署所建立的任何資源，例如複製的叢集、執行個體、複寫堆疊等。如果刪除該來源的任何資源，請勿使用較早的海王星藍/綠，與. `DeploymentID` `DeployMode=resume` 相反，請使用新的`DeploymentID`.

從運行海王星藍色/綠色AWS CloudFormation堆棧 AWS Management Console

如AWS CloudFormation需有關如AWS CloudFormation何從AWS Management Console.

從以下位置執行海王星藍色/綠色解決方案 AWS Management Console

1. 請登入 AWS Management Console，開啟位於 https://console.aws.amazon.com/cloudformation 的 AWS CloudFormation 主控台。
2. 海王星藍色/綠色模板有一個亞馬遜 S3 URL。
3. 範本位於：

   ```
   https://s3.amazonaws.com/aws-neptune-customer-samples-us-east-1/neptune-bg/bg.yaml
   ```

4. 設置堆棧參數如下：

   - **`DeploymentID`**— 每個海王星藍/綠部署唯一的識別碼。

     它可做為綠色資料庫叢集識別碼，以及用來命名在部署期間建立的新資源的前置詞。
   - **`NeptuneSourceClusterId`**— 您要升級之藍色資料庫叢集的識別碼。
   - **`NeptuneTargetClusterVersion:`**— 您要將藍色資料庫叢集升級到的海王星引擎版本 (p. 856)。

這必須高於當前藍色數據庫集群的引擎版本。

- **DeploymentMode**— 指示這是新部署還是嘗試繼續先前的部署。

  有效值為：new(預設值) 和resume。

- **GraphQueryType**— 資料庫的圖形資料類型。

  有效值為：gremlin(預設值) propertygraph、pg、sparql、和rdf。

- **SubnetId**— 來自藍色資料庫叢集所在之相同 VPC 的子網路識別碼。

  如果您想要透過 EC2 Connect 透過 SSH 連線至執行個體，請提供公有子網路的 ID。

- **InstanceSecurityGroup**— EC2 執行個體的安全群組。

  安全性群組必須能夠存取藍色資料庫叢集，而且您必須能夠透過 SSH 存取執行個體。請參閱 使用 VPC 主控台建立安全群組 (p. 62)。

使用AWS CLI來執行海王星藍色/綠色堆疊 AWS CloudFormation

這是執行海王星藍/綠AWS CloudFormation堆疊的範例AWS CLI命令：

```
region=us-east-1
stack_name="test-neeptune-bg-$(date +"%d-%s-%h")-$region"

aws cloudformation create-stack \
  --stack-name $stack_name
  --template-url https://s3.amazonaws.com/aws-neptune-customer-samples-us-east-1/neptune-
bg/bg.yaml \
  --parameters '[
      {
        "ParameterKey": "DeploymentID",
        "ParameterValue": "test154"
      },
      {
        "ParameterKey": "NeptuneSourceClusterId",
        "ParameterValue": "bg-iam"
      },
      {
        "ParameterKey": "NeptuneTargetClusterVersion",
        "ParameterValue": "1.2.1.0"
      },
      {
        "ParameterKey": "DeploymentMode",
        "ParameterValue": "new"
      },
      {
        "ParameterKey": "GraphQueryType",
        "ParameterValue": "gremlin"
      },
      {
        "ParameterKey": "SubnetId",
        "ParameterValue": "subnet-9792de"
      },
      {
        "ParameterKey": "InstanceSecurityGroup"
        "ParameterValue": "sg-b66acc"
        }
    ]' \
  --capabilities CAPABILITY_IAM \
```

```
--region $region \
--profile your_aws_profile
```

手動執行海王星藍/綠解決方案，無需使用 AWS CloudFormation

手動執行海王星藍/綠解決方案

1. 使用中定義的權限建立 EC2 執行個體設定檔角色海王幕藍/綠的必備 (p. 517)。如需詳細資訊，請參閱使用 IAM 角色將許可授與在 Amazon EC2 執行個體上執行的應用程式。

   請務必同時授予neptune-db:*動作存取權，以便 Neptune 藍/綠解決方案可以從藍色和綠色叢集取得串流資料。例如，您可以使用此資源 ARN 為區域中的所有叢集提供存取權：

   ```
   arn:aws:neptune-db:(AWS region):(account ID):*/*
   ```

2. 安裝這些 Python 庫，如下所示：

   ```
   pip3 install schedule boto3 aws_requests_auth requests watchtower urllib3==1.26
   yum -y install tmux screen
   ```

3. 執行如下所示的 shell 指令碼，以啟動海王星藍/綠解決方案：

   ```
   ## Set up
   sudo -u ec2-user bash
   curl https://aws-neptune-customer-samples-us-east-1.s3.amazonaws.com/neptune-bg/bg.zip
    --output bg.zip
   unzip bg.zip
   chmod 700 bg.py

   ## Kick off the Neptune Blue/Green solution
   python3 bg.py -s (blue cluster ID) \
                 -t (target engine version) \
                 -d (Neptune Blue/Green deployment ID) \
                 -g (graph type) \
                 -m (mode)
   ```

   啟動海王星藍/綠的 Python 命令的示例是：

   ```
   python3 bg.py -s testbg-1 -t 1.2.0.2 -d test123434 -g gremlin -m new
   ```

   您可以在CloudWatch記錄中監視 Neptune 藍色/綠色解決方案的進度。根據藍色叢集拓撲、目標引擎版本和叢集配置而定，指令碼可能需要數小時才能完成。因此，最好在終端多路復用器（例如 tmux 或 screen）中運行腳本，以便它可以在不中斷的情況下運行。

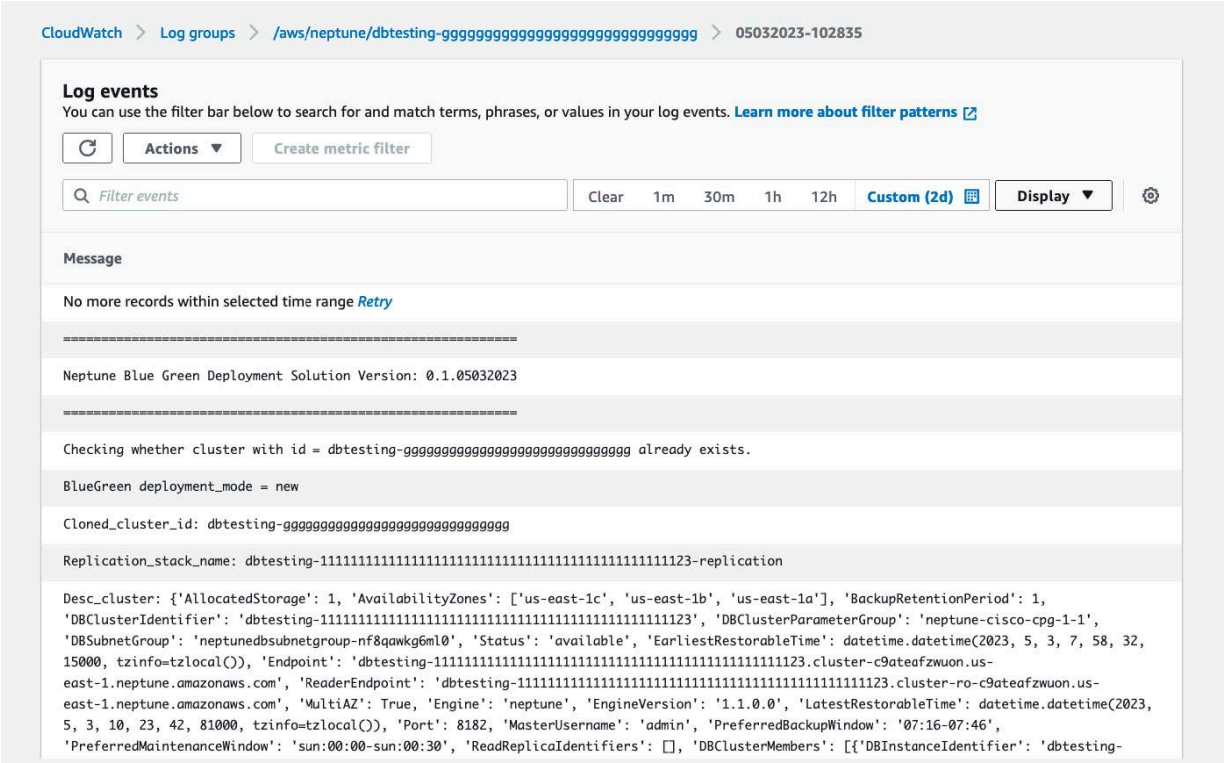
監控海王星藍/綠部署的進度

您可以移至CloudWatch主控台並監視此CloudWatch記錄群組中的記錄，以監視 Neptune 藍色/綠色解決方案：

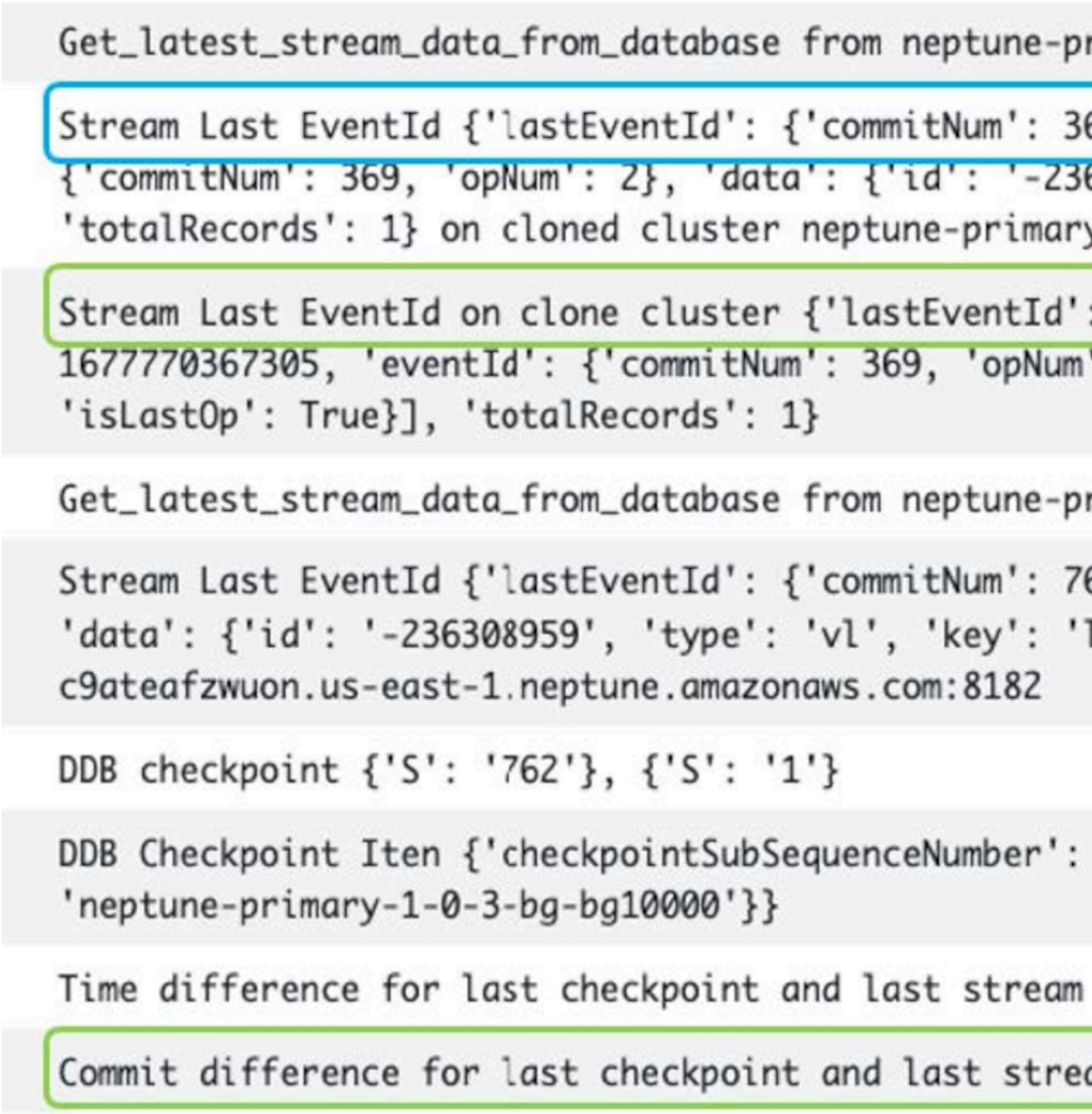
```
/aws/neptune/(Neptune Blue/Green deployment ID)
```

您也可以透過 SSH 存取作為海王星藍/綠AWS CloudFormation堆疊一部分建立的 EC2 執行個體，或由您手動執行個體，然後查看此記錄檔：/var/log/cloud-init-output.log。

記錄會顯示海王星藍色/綠色解決方案所採取的動作：



日誌消息顯示藍色和綠色數據庫群集之間的同步狀態：



從生產藍簇切割到更新的綠色集群

綠色叢集建立完成後，Neptune 藍/綠會在藍色和綠色叢集之間設定以串流為基礎的複寫 (p. 596)。然後，它會計算藍色叢集上的最新串流 EventID 與檢查點資料表中存在的複寫檢查點之間的差異，來檢查複寫延遲。檢查點資料表是由海王星到海王星複寫堆疊所建立的 DynamoDB 資料表。您可以監視海王星藍/綠 CloudWatch記錄檔中的複寫差異。當此複寫差異達到零時，您的測試環境已準備好進行測試。

如果您滿意更新是您想要的，請執行下列步驟，將流量從目前的藍色生產伺服器切換到更新的綠色暫存伺服器：

Important

最好在維護時段期間切換到綠色伺服器。

- 關閉藍色叢集上的寫入流量 — 這對於讓綠色叢集趕上藍色叢集上的所有寫入作業是必要的，以便在切斷時提交差異為零。
- 視需要更新應用程式的 IAM 政策 — 如果您已在藍色叢集上啟用 IAM 身份驗證，請務必更新應用程式的 IAM 政策以指向綠色叢集。
- 當兩個叢集同步時，提升綠色叢集 — 當綠色叢集上的提交差異為零時，表示所有對藍色叢集的寫入都已寫入綠色叢集，啟用綠色叢集的寫入流量，使其成為您的生產叢集。

海王幕藍/綠解決方案完成後的清理

將暫存 (綠色) 叢集升級至生產環境後，請清除 Neptune 藍/綠解決方案所建立的資源：

- 刪除為執行海王星藍/綠解決方案而建立的 EC2 執行個體。
- 刪除使綠色叢集與藍色叢集保持同步之 Neptune 串流型複寫 (p. 596)的AWS CloudFormation範本。主要的一個具有您之前提供的堆棧名稱，其中一個由「-複製」後面的部署 ID 組成：即。*(DeploymentID)*-replication

刪除AWS CloudFormation範本並不會刪除叢集本身。確認綠色叢集如預期般運作之後，您可以選擇性地建立快照並手動刪除藍色叢集。

海王幕藍/綠解決方案最佳實

- 在將綠色叢集切換到生產環境之前，值得徹底確認其運作正常。檢查數據的一致性和數據庫的配置。有些新的引擎版本可能也需要用戶端升級。在升級之前，請檢查引擎版本說明。在開始生產藍/綠升級之前，值得在開發，測試和生產前環境中測試所有這些問題。
- 最好在維護窗口期間執行從藍色到綠色服務器的切換。
- 為了確保在升級和同步處理後一切正常運作，值得在刪除原始叢集之前保留一段時間。如果出現不可預見的問題，它可能會很有用。
- 執行 Neptune Blu/Green 解決方案時，請避免大量寫入作業 (例如大量負載)，因為這些作業可能會造成複寫延遲，導致大量停機 理想情況下，關閉對藍色叢集的寫入和為綠色叢集啟動它們之間的時間只是幾分鐘。

海王幕藍/綠解決方案的故障

海王幕藍/綠解決方案引發的錯誤

- **`Cluster with id = (blue_green_deployment_id) already exists`**— 存在具有識別碼的現有叢集 *(########)*。

  如果叢集是在先前的 Neptune 藍/綠執行中建立，`resume`請提供新的部署識別碼或將部署模式設定為。
- **`Streams should be enabled on the source Cluster for Blue Green Deployment`**— 啟用藍色 (來源) 叢集上的海王星串流 (p. 586)。
- **`No Bulkload should be in progress on source cluster: (cluster_id)`**— 如果 Neptune 藍/綠解決方案識別正在進行的批量負載，則會終止。

  這是為了確保同步過程能夠趕上正在進行的寫入。在啟動 Neptune 藍/綠解決方案之前，請避免或取消任何正在進行的大量載入

- **`Blue Green deployment requires instances to be in sync with db cluster parameter group`**— 叢集參數群組的任何變更都應在整個資料庫叢集中保持同步。請參閱 Amazon Neptune (p. 527)。
- **`Invalid target engine version for Blue Green Deployment`**— 目標引擎版本必須在中列為作用中Amazon Neptune 的引擎版本 (p. 856)，且必須高於來源 (藍色) 叢集的目前引擎版本。

建立具有 Neptune 許可的 IAM 使用者使用者

若要存取 Neptune 主控台以建立和管理 Neptune 資料庫叢集，您需要建立具有所有必要權限的 IAM 使用者。

第一步角色角角角角角角角角角角角角角角角角角角角角角角角角角角角色

建立服務角角角角角角角角角角角角角角角

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 在「策略」頁面上，選取「建立策略」。
4. 在 [建立原則] 頁面上，選取 JSON 索引標籤，然後複製下列服務連結角色原則：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "iam:CreateServiceLinkedRole",
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
      "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
      }
    }
  ]
}
```

5. 選取下一步:標籤，然後在「新增標籤」頁面上選取「下一步：複查」。
6. 在 [檢閱策略] 頁面上，將新策略命名為NeptuneServiceLinked ""。

如需服務連結角色的詳細資訊，請參閱針對 Neptune 使用服務連結角色 (p. 215)。

建立具有所有必要許可的新 IAM 使用者使用者

接下來，建立新的 IAM 使用者並附加適當的受管政策，以授予您需要的許可，以及您建立的服務連結角色政策 (此處命名為NeptuneServiceLinked)：

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在左側的導覽窗格中，選擇 [使用者]，然後在 [使用者] 頁面上選擇 [新增使用者]。
3. 在 [新增使用者] 頁面上，輸入新 IAM 使用者的名稱，選擇 [存取金鑰]-[程式存取] 做為AWS認證類型，然後選擇 [下一步：權限]。
4. 在 [設定權限] 頁面的 [篩選器原則] 方塊中，輸入 [Neptune]。現在，從列出的策略中選擇以下內容：
   - NeptuneFullAccess
   - NeptuneConsoleFullAccess
   - NeptuneServiceLinked（假設這是您之前創建的服務鏈接角色策略命名的名稱）。
5. 接下來在「過濾器策略」框中鍵入「VPC」代替「Neptune」。從列出的策略FullAccess中選擇亞馬遜VPC。
6. 選取下一步:標籤，然後在「新增標籤」頁面中選取「下一步：複查」。
7. 在「複查」頁面中，檢查下列所有策略現在已附加至新使用者：

- NeptuneFullAccess
- NeptuneConsoleFullAccess
- NeptuneServiceLinked
- 亞馬遜 VPCFullAccess

然後，選取建立使用者。

8. 最後，下載新使用者的存取金鑰 ID 和私密存取金鑰 ID 和私密存取金鑰。

若要在其他服務 (例如 Amazon Simple Storage Service (Amazon S3) 角色，您需要新增其他許可和信任關係角色。

Amazon Neptune

您可以在 Amazon Neptune 中使用參數群組中的參數 (p. 530)來管理資料庫組態。參數群組扮演引擎組態值的容器，以套用至一或多個資料庫執行個體。

參數群組有兩種類型，即資料庫叢集參數群組和資料庫參數群組：

- 資料庫參數群組適用於執行個體層級，而且通常與 Neptune 圖形引擎的設定 (例如neptune_query_timeout參數) 相關聯。
- 「資料庫叢集參數群組」會套用到叢集中的每個執行個體，並且一般擁有較廣的設定。每個 Neptune 該叢集中的每個資料庫執行個體都會繼承資料庫叢集參數群組中包含的引擎組態值。

您在資料庫叢集參數群組中修改的任何組態值，都會覆寫資料庫參數群組中的預設值。如果您在資料庫參數群組中編輯對應的值，則這些值會覆寫資料庫叢集參數群組中的設定。

如果您建立資料庫執行個體而未指定自訂資料庫參數群組，將會使用預設的資料庫參數群組。您無法修改預設資料庫參數群組的參數設定。若要變更預設參數設定，您必須建立新的資料庫參數群組。並非所有資料庫引擎參數都可以在您建立的資料庫參數群組中變更。

參數群組是在與不同 Neptune 引擎版本相容的族群中建立的。預設參數群組族群為neptune1，與之前的所有引擎版本相容1.2.0.0。從開始已發行版本: (p. 884)，必須改用neptune1.2參數群組族群。這意味著當您升級到1.2.0.0 或更高版本時，必須先重新建立neptune1.2族群中的所有自訂參數群組，以便在升級時可以附加它們。

有些 Neptune 參數是靜態的，其他參數是動態的。差異如下所示：

靜態參數

- 靜態參數是指只有在資料庫執行個體重新啟動後才會生效的參數。換句話說，當您變更靜態參數並儲存執行個體資料庫參數群組時，您必須手動重新啟動資料庫執行個體，參數變更才會生效。目前，所有 Neptune 執行個體層級參數 (位於資料庫參數群組而非資料庫叢集參數群組中) 都是靜態的。
- 當您變更叢集層級靜態參數並儲存資料庫叢集參數群組時，參數變更會在您手動重新啟動叢集中的每個資料庫執行個體後生效。

動態參數

- 動態參數是在參數群組中更新參數之後幾乎立即生效的參數參數。換句話說，更新動態參數後，不需要重新啟動資料庫執行個體，參數變更才會生效。
- 動態叢集參數變更會套用至所有資料庫執行個體，預期會有些微的延遲。
- 更新的動態參數值不會套用至目前執行中的要求，而只會套用至變更發生後提交的要求。
- 變更動態叢集層級參數時，依預設，參數變更會立即套用至資料庫叢集，無需重新啟動。若要在重新啟動叢集中的資料庫執行個體後再執行參數變更，您可以使用ApplyMethod來將參數變更設定pending-reboot為。AWS CLI

目前，除了下列新叢集參數以外，所有參數都是靜態的：

- neptune_enable_slow_query_log(叢集層級)
- neptune_slow_query_log_threshold(叢集層級)

以下是使用資料庫參數群組中的參數時，您應該知道的一些重要提示：

- 未正確設定資料庫參數群組中的參數，可能產生各種意外影響，包括降低效能和系統不穩定。修改資料庫參數時請務必謹慎，在修改資料庫參數群組之前，請備份您的資料。在建參數群組設定變更套用到生產資料庫執行個體之前，請先在測試資料庫執行個體上嘗試這些變更。

- 變更與資料庫執行個體關聯的資料庫參數群組時，您必須手動重新啟動執行個體，資料庫執行個體才會使用新的資料庫參數群組。

  Note

  之前已發行版本: (p. 884)，資料庫叢集中的所有僅供讀取複本執行個體都會在主要 (寫入器) 執行個體重新啟動時自動重新啟動
  從向已發行版本: (p. 884)前開始，重新啟動主執行個體並不會導致任何複本執行個體重新啟動。這表示如果您要變更叢集層級參數，您必須個別重新啟動每個執行個體，才能取得參數變更。

編輯資料庫叢集參數群組或資料庫參數群組

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune
2. 在導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇您要編輯的資料庫參數群組的 Name (名稱) 連結。

   (選用) 選擇 Create parameter group (建立參數群組) 來建立新的叢集參數群組，然後建立新的群組。然後，選擇新參數群組的 Name (名稱)。

   Important

   若因為無法修改預設資料庫叢集參數群組，因此您只有預設的資料庫叢集參數群組，則此步驟是「必要」的。
4. 選擇 Edit parameters (編輯參數)。
5. 設定您要變更的參數值。
6. 選擇 Save changes (儲存變更)。
7. 如果您要變更資料庫叢集參數，請重新啟動 Neptune 叢集中的每個資料庫執行個體；如果要變更資料庫執行個體參數，請重新啟動一或多個特定執行個

建立資料庫參數群組或資料庫叢集參數群組

您可以輕鬆地使用 Neptune

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune
2. 在左側導覽窗格中，選擇 Parameter groups (參數群組)。
3. 選擇 Create DB parameter group (建立資料庫參數群組)。

   Create DB parameter group (建立資料庫參數群組) 頁面隨即出現。
4. 在參數組系列列表中，選擇海王星 1，或者，如果您的目標是 1.2.0.0 或更高版本的引擎，請選擇海王星 1.2。
5. 在 Type (類型) 清單中選擇 DB Parameter Group (資料庫參數群組) 或 DB Cluster Parameter Group (資料庫叢集參數群組)。
6. 在 Group name (群組名稱) 方塊中輸入新資料庫參數群組的名稱。
7. 在 Description (描述) 方塊中輸入新資料庫參數群組的描述。
8. 選擇 建立 。

您也可以使用下列項目來建立新的參數群組AWS CLI：

```
aws neptune create-db-parameter-group \
  --db-parameter-group-name (a name for the new DB parameter group) \
```

```
 --db-parameter-group-family (either neptune1 or neptune1.2, depending on the engine
version) \
 --description (a description for the new DB parameter group)
```

Amazon Neptune 參數

您可以在 Amazon Neptune 中使用參數群組中的參數來管理資料庫組 (p. 527)態。下列參數可供設定 Neptune 資料庫：

叢集層級參數

- 海王星啟用審計日誌 (p. 530)
- 海王星啟用 _ 慢速查詢日誌 (p. 530)
- 海王星慢查詢日誌閾值 (p. 531)
- 海王星實驗室模式 (p. 531)
- 海王星查詢超時 (p. 531)
- 海王星溪 (p. 531)
- 海王星流 _ 過期 _ 日 (p. 532)
- 海王星查找緩存 (p. 532)
- 海王星 _ 自動擴展 _ 配置 (p. 532)
- 海王星 _ 美国角色 (p. 532)
- 海王星端點 (p. 533)

執行個體層級參數

- 海王星查詢引擎 (p. 533)
- 海王星查詢超時 (p. 533)
- 海王星結果緩存 (p. 533)

已棄用參數

- 海王星強制執行 (p. 533)

neptune_enable_audit_log(叢集層級參數)

此參數可切換 Neptune 的稽核記錄。

允許的值為 0 (已停用) 和 1 (已啟用)。預設值為 0。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

您可以將稽核日誌發佈到 Amazon CloudWatch，如中所述使用 CLI 將 Neptune 稽核記錄發佈至CloudWatch 記錄 (p. 783)。

neptune_enable_slow_query_log(叢集層級參數)

使用此參數可啟用或停用 Neptune 的慢速查詢記錄 (p. 785)功能。

這是一個動態參數，表示變更其值不需要或導致資料庫叢集重新啟動。

允許的值為：

- **info**— 啟用慢速查詢記錄，並記錄可能導致效能降低的所選屬性。
- **debug**— 啟用慢速查詢記錄，並記錄查詢執行的所有可用屬性。
- **disable**— 停用慢速查詢記錄。

預設值為 disable。

您可以將慢速查詢日誌發佈到 Amazon CloudWatch，如中所述使用 CLI 將 Neptune 慢速查詢記錄檔發佈至記錄 CloudWatch (p. 783)。

neptune_slow_query_log_threshold(叢集層級參數)

此參數指定執行時間臨界值 (以毫秒為單位)，之後查詢就會被視為緩慢查詢。如果啟用了慢速查詢記錄 (p. 785)，則執行時間超過此臨界值的查詢將與其某些屬性一起記錄。

預設值為 5000 毫秒 (5 秒)。

這是一個動態參數，表示變更其值不需要或導致資料庫叢集重新啟動。

neptune_lab_mode(叢集層級參數)

設定時，此參數會啟用 Neptune 的特定實驗功能。如需目前可用的實驗性功能，請參閱海王星實驗室 (p. 127)。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

若要啟用或停用實驗性功能，請在此參數中包含 *(feature name)*=enabled 或 *(feature name)*=disabled。您可以啟用或停用多個功能，方法為使用逗號區隔它們，如下所示：

(feature #1 name)=enabled,*(feature #2 name)*=enabled

預設為停用實驗室模式功能。一個例外是DFEQueryEngine功能，該功能預設為啟用，以便在 Neptune 引擎版本 1.0.5.0 (p. 931) 中與查詢提示 (DFEQueryEngine=viaQueryHint) 搭配使用。從 Neptune 引擎版本 1.1.1.0 (p. 894) 開始，DFE 引擎不再海王星查詢引擎 (p. 533)處於實驗室模式，現在可以使用執行個體資料庫參數群組中的執行個體參數來控制。

neptune_query_timeout(叢集層級參數)

指定圖形查詢的特定逾時持續時間 (以毫秒為單位)。

允許的值範圍從 10 到 2,147,483,647 (2^{31} -1)。預設值為 120,000 (2 分鐘)。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

> Note
>
> 如果您將查詢逾時值設定得太高，特別是在無伺服器執行個體上，可能會產生非預期的費用。如果沒有合理的逾時設定，您可能會不小心發出查詢，讓執行時間遠超過預期，導致您從未預期的成本。這在無伺服器執行個體上尤其如此，該執行個體在執行查詢時可能會擴展成大型且昂貴的執行個體類型。
> 您可以使用可容納大部分查詢的查詢逾時值，並且只會導致意外長時間執行的查詢逾時，避免此類非預期的費用。

neptune_streams(叢集層級參數)

啟用或停用 Neptune (p. 585)。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

允許值為 0 (停用，此為預設值) 和 1 (啟用)。

neptune_streams_expiry_days(叢集層級參數)

指定伺服器刪除串流記錄之前經過多少天。

允許的值是從1到90 (含)。預設值為 7。

此參數為 1.2.0.0 版本中引 (p. 884)入。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

neptune_lookup_cache(叢集層級參數)

停用或重新啟用 Neptune 查閱快取。 (p. 107) 在R5d執行個體上。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

允許的值為 enabled 和 disabled。預設值為disabled，但無論何時在資料庫叢集中建立R5d執行個體，都會自動將neptune_lookup_cache參數設定為，enabled並在該執行個體上建立查閱快取。

neptune_autoscaling_config(叢集層級參數)

設定 Neptune auto-scaling (p. 549) 建立和管理的僅供讀取複本執行個體的組態參數。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

使用您設定為neptune_autoscaling_config參數值的 JSON 字串，您可以指定：

- Neptune auto-scaling 用於其建立的所有新僅供讀取複本執行個體的執行個體類型。
- 指派給這些僅供讀取複本的維護時段。
- 要與所有新僅供讀取複本產生關聯的標籤。

JSON 字符串具有這樣的結構：

```
"{
  \"tags\": [
    { \"key\" : \"reader tag-0 key\", \"value\" : \"reader tag-0 value\" },
    { \"key\" : \"reader tag-1 key\", \"value\" : \"reader tag-1 value\" },
  ],
  \"maintenanceWindow\" : \"wed:12:03-wed:12:33\",
  \"dbInstanceClass\" : \"db.r5.xlarge\"
}"
```

請注意，字串中的引號必須全部以反斜線字元 (\) 逸出。

未在neptune_autoscaling_config參數中指定的三個組態設定中的任何一個，都會從資料庫叢集的主要寫入器執行個體的組態中複製。

neptune_ml_iam_role(叢集層級參數)

指定 Neptune ML 中使用的 IAM 角色 ARN。該值可以是任何有效的 IAM 角色 ARN。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

您可以在圖形上指定用於機器學習的預設 IAM 角色 ARN。

neptune_ml_endpoint(叢集層級參數)

指定用於 Neptune ML 的端點。該值可以是任何有效的SageMaker端點名稱。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

您可以在圖形上指定機器學習的預設 SageMaker 端點。

neptune_dfe_query_engine(執行個體層級參數)

從 Neptune 引擎版本 1.1.1.0 (p. 894) 開始，此資料庫執行個體參數用於控制 DFE 查詢引擎 (p. 129)的使用方式。允許的值如下：

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

- **enabled**— 使 DFE 引擎盡可能使用，但useDFE查詢提示存在且設定為除外false。
- **viaQueryHint**(預設值) — 使 DFE 引擎僅用於明確包含設定的useDFE查詢提示的查詢true。

如果尚未明確設定此參數，則會在執行個體啟動時使用預設值。viaQueryHint

> Note
>
> 無論如何設定此參數，所有 OpenCypher 查詢都會由 DFE 引擎執行。

在 1.1.1.0 版之前，這是實驗室模式參數，而不是資料庫執行個體參數。

neptune_query_timeout(執行個體層級參數)

此資料庫執行個體參數指定一個執行個體的圖形查詢逾時持續時間 (以毫秒為單位)。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

允許的值範圍從 10 到 2,147,483,647 (2^{31} -1)。預設值為 120,000 (2 分鐘)。

> Note
>
> 如果您將查詢逾時值設定得太高，特別是在無伺服器執行個體上，可能會產生非預期的費用。如果沒有合理的逾時設定，您可能會不小心發出查詢，讓執行時間遠超過預期，導致您從未預期的成本。這在無伺服器執行個體上尤其如此，該執行個體在執行查詢時可能會擴展成大型且昂貴的執行個體類型。
> 您可以使用可容納大部分查詢的查詢逾時值，並且只會導致意外長時間執行的查詢逾時，避免此類非預期的費用。

neptune_result_cache(執行個體層級參數)

neptune_result_cache— 此資料庫執行個體參數啟用或停用快取查詢結果 (p. 330)。

這個參數是靜態的，表示在重新開機之前，對其所做的變更不會在任何執行個體上生效。

允許的值包括0 (停用，這是預設值) 和1 (已啟用)。

neptune_enforce_ssl(已停用的叢集層級參數)

(已停用) 曾經有允許 HTTP 連線至 Neptune 的區域，而且此參數在設定為 1 時強制所有連線都使用 HTTPS。不過，這個參數不再相關，因為 Neptune 現在只接受所有區域的 HTTPS 連線。

使用啟動 Neptune 資料庫叢集AWS Management Console

啟動新的 Neptune DB 叢集最簡單的方法是使用可為您建立所有必要資源的AWS CloudFormation範本，如中所述建立資料庫叢集 (p. 54)。

如果您願意，也可以使用 Neptune 主控台手動啟動新的資料庫叢集，如此處所述。

在您可以存取 Neptune 主控台以建立 Neptune 叢集之前，請先建立具有必要權限的 IAM 使用者，如中所述建立具有 Neptune 許可的 IAM 使用者使用者 (p. 525)。

然後，以該 IAM 使用者身分登入，並依照下列步驟建立新的資料庫叢集：AWS Management Console

使用主控台啟動 Neptune 資料庫叢集

1. 登入AWS管理主控台，並前往 https://console.aws.amazon.com/neptune/home 開啟 Amazon Neptune 主控台。
2. 瀏覽至「資料庫」頁面，然後選擇建立資料庫，開啟「建立資料庫」頁面。
3. 在「引擎選項」下，引擎類型為neptune，您可以選擇特定的引擎版本或接受預設值。
4. 在 [設定] 下，輸入新資料庫叢集的名稱，或接受此處提供的預設名稱。此名稱會用於執行個體的端點地址，並且必須滿足下列限制：

   - 必須包含 1 到 63 個英數字元或連字號。
   - 第一個字元必須是字母。
   - 不能以一個連字號結尾或是連續包含兩個連字號。
   - 在您AWS帳戶中的所有資料庫執行個體中，它必須是唯一的AWS。
5. 在 [範本] 下，選擇 [生產] 或 [開發與測試]。
6. 在資料庫執行個體大小下，選擇執行個體大小。這將決定新資料庫叢集的主要寫入執行個體的處理和記憶體容量。

   如果您選擇了 Pro duc tion 模板，則只能從列出的可用內存優化類中進行選擇，但是如果您選擇了「開發」和「測試」，則也可以從更經濟的可重載類別中進行選擇（有關可重載類別的討論，請參閱T3 高載執行個體 (p. 569)）。

   Note

   從 Neptune 引擎版本 1.1.0.0 (p. 915) 開始，Neptune 不再支援R4執行個體類型。
7. 在可用性和耐久性下，您可以選擇是否啟用 multi-availability-zone (異地同步備份) 部署。生產範本預設會啟用異地同步備份部署，而開發和測試範本則不會啟用。如果啟用異地同步備份部署，Neptune 會尋找您在不同可用區域 (AZ) 中建立的僅供讀取複本執行個體，以提高可用性。
8. 在 [連線] 底下，從可用選項中選取將裝載新資料庫叢集的虛擬私人雲端 (VPC)。如果您希望 Neptune 為您創建 VPC，則可以在此選擇創建新的 VPC。您必須在同一個 VPC 中建立 Amazon EC2 執行個體才能存取 Neptune 執行個體 (如需詳細資訊，請參閱每個亞馬遜海王星資料庫叢集都駐留在亞馬遜 VPC 中 (p. 154))。請注意，您無法在建立資料庫叢集後變更 VPC。

   如有需要，您可以在 [其他連線設定] 下進一步設定叢集的連線能力：

   a. 在「子網路群組」下，您可以選擇要用於新資料庫叢集的 Neptune DB 子網路群組。如果您的 VPC 還沒有任何子網路群組，Neptune 會為您建立資料庫子網路群組 (請參閱每個亞馬遜海王星資料庫叢集都駐留在亞馬遜 VPC 中 (p. 154))。
   b. 在 VPC 安全群組下，選擇一個或多個現有的 VPC 安全群組，以保護對新資料庫叢集的網路存取；如果要 Neptune 為您建立一個群組，請選擇 [建立新]，然後為新的 VPC 安全性群組提供名稱 (請參閱使用 VPC 主控台建立安全群組 (p. 62))。

c. 在 [資料庫連接埠] 下，輸入資料庫將用於應用程式連線的 TCP/IP 連接埠。Neptune 使用連接埠號碼8182做為預設值。

9. 如果您想讓 Neptune 在 Neptune 工作台中為您建立 Jupyter 筆記本，請在 [記事本組態] 下選擇 [建立筆記本] (請參閱使用 Neptune 圖形筆記本快速開始使用 (p. 30)和使用 Neptune 工作台託管海王星筆記本 (p. 31))。接著您可以選擇新筆記型電腦的設定方式：

   a. 在 Notebook 執行個體類型下，從筆記本的可用執行個體類別中選擇。

   b. 在 [記事本名稱] 下，輸入筆記本的名稱。

   c. 如果需要，您也可以在「描述 -選用」下輸入筆記本的描述。

   d. 在 IAM 角色名稱下，選擇讓 Neptune 為筆記本建立 IAM 角色，然後輸入新角色的名稱，或選擇從可用角色中選取現有的 IAM 角色。

   e. 最後，選擇您的筆記型電腦是直接連線到網際網路，還是透過 Amazon SageMaker 或透過具有 NAT 閘道的 VPC 連線至網際網路。如需詳細資訊，請參閱將筆記本執行個體 Connect 至 VPC 中的資源。

10. 在標籤下，您最多可以將 50 個標籤與您的新資料庫叢集建立關聯。

11. 在 [其他組態] 底下，您可以為新的資料庫叢集進行更多設定 (在許多情況下，您可以略過它們並立即接受預設值)：

選項	您可以做些什麼
DB instance identifier (資料庫執行個體識別符)：	您可以提供叢集的寫入器執行個體的名稱。如果不這樣做，則會使用以叢集名稱為基礎的預設識別碼。如果這樣做，請為您的AWS帳戶在目前區域中擁有的所有資料庫執行個體指定一個唯一的名稱。資料庫執行個體識別碼不區分大小寫，但儲存為全部小寫。
DB cluster parameter group (資料庫叢集參數群組)	選取資料庫叢集參數群組，以定義叢集中所有資料庫執行個體的預設組態。除非您選擇，否則 Neptune 會使用預設資料庫叢集參數群組。如需參數群組的詳細資訊，請參閱Amazon Neptune (p. 527)。
DB parameter group (資料庫參數群組)	選取資料庫參數群組，以定義叢集中主要資料庫執行個體的組態。除非您另有選擇，否則 Neptune 會使用預設參數群組。如需參數群組的詳細資訊，請參閱參數群組 (p. 527)。
IAM DB authentication (IAM 資料庫身分驗證)	如果您勾選 「啟用 IAM 資料庫身份驗證」，則會使用AWS Identity and Access Management (IAM) 驗證對資料庫的所有存取。 **Important** 這需要您使用簽AWS名版本 4 簽名來簽署所有請求。如需詳細資訊，請參閱亞馬遜海王星AWS Identity and Access Management (IAM) 概述 (p. 157)。
容錯移轉優先順序	選擇No preference或容錯移轉的優先順序層。如果您選擇階層，但其中存在爭用，則會選取與主要執行個體大小相同的複本。
Backup retention period (備份保留期間)	選擇從 1 到 35 天的時間長度，Neptune 應該保留此資料庫執行個體的自動備份的時間長度。您只

選項	您可以做些什麼
	能在備份保留期內執行 point-in-time 還原 (PITR) 至某段時間。
Copy tags to snapshots (將標籤複製到快照)	(預設為啟用) 此選項會將與資料庫叢集相關聯的所有標籤複製到其任何快照集。
Enable encryption (啟用加密)	(預設為啟用) 此選項會使資料庫叢集中的資料在靜態時加密。 如果您這樣做，請選擇用來保護用於保護用於加密此資料庫磁碟區的金鑰，請選擇用來保護用於保護此 您可以選取預設金`aws/rds`鑰，或從您帳戶中的數個主金鑰進行選擇，或輸入來自不同帳戶的金鑰。您可以在 IAM 主控台的 [加密金鑰] 索引標籤上建立新的主要加密金鑰。如需詳細資訊，請參閱在靜態時加密海王星資源 (p. 155)。
稽核記錄	如果您要將資料庫叢集中的稽核記錄發佈到記 CloudWatch 錄，請核取此選項。
Enable auto minor version upgrade (啟用自動次要版本升級)	(預設為啟用) 此選項會導致您的資料庫叢集在發行後自動升級至新的次要引擎版本。自動升級會在資料庫的維護期間進行。請參閱 升級海王星引擎的過程 (p. 858)。
Maintenance window (維護時段)	您可以選取要在資料庫叢集進行擱置修改的特定期間，例如資料庫執行個體類別的變更或自動引擎修補程式。任何此類維護作業都會在選取的期間內開始並完成。如果您未選取週期，Neptune 會隨意指定維護週期。
啟用刪除保護	預設啟用 ）刪除保護會阻止您的資料庫叢集遭意外刪除。您必須明確停用它，才能刪除資料庫叢集。

12. 選擇建立資料庫以啟動新的 Neptune 資料庫叢集及其主要執行個體。

    在 Amazon Neptune 主控台上，新的資料庫叢集會顯示在資料庫清單中。資料庫叢集的狀態為 Creating (建立中)，直到建立完成且可供使用為止。狀態變更為 Available (可用) 時，您便能連接到資料庫叢集的主執行個體。根據資料庫執行個體類別和分配的存放區，新執行個體可能要幾分鐘後才能使用。

    若要檢視新建立的叢集，請選擇 Neptune 主控台中的 [資料庫] 檢視。

    > **Note**
    >
    > 如果您使用刪除資料庫叢集中的所有 Neptune 資料庫叢集中AWS Management Console，主控台會自動刪除資料庫叢集本身。如果您使用AWS CLI或 SDK，則必須在刪除資料庫叢集的最後一個執行個體之後手動刪除資料庫叢集。

    記下叢集端點值。您需要這個才能連接至您的 Neptune 資料庫叢集。

停止和啟動 Amazon Neptune 資料庫叢集

停止和啟動 Amazon Neptune 叢集可協助您管理開發和測試環境的成本。您可以暫時停用叢集中的所有資料庫執行個體，而非每次使用叢集時，設定和卸除所有資料庫執行個體。

主題

- 停止和啟動 Neptune 資料庫叢集的概觀 (p. 537)
- 停止 Neptune 資料庫叢集 (p. 537)
- 啟動停止的 Neptune 資料庫叢集 (p. 538)

停止和啟動 Neptune 資料庫叢集的概觀

在不需要 Neptune 叢集的期間，您可以一次停止該執行個體。一旦您需要叢集，即可隨時重新啟動它。啟動和停用可簡化用於下列操作之叢集的設定和卸除程序：開發、測試或不需要連續可用性的類似活動。無論叢集中有多少執行個體，您都可以透過單一動作在AWS Management Console中完成此操作。

資料庫叢集停用時，只需支付您指定的保留時段內叢集儲存、手動快照和自動備份儲存的費用。您無須支付任何資料庫執行個體小時數的費用。

Neptune 會在七天後自動啟動您的資料庫叢集，以確保它不會落後於任何必要的維護更新。

若要將輕度載入之 Neptune 叢集的費用降至最低，您可以停止該叢集，而非刪除其所有僅供讀取複本。對於具有一個或兩個以上執行個體的叢集，經常刪除和重建資料庫執行個體實際上僅會使用AWS CLI或 Neptune API，刪除也會以正確的順序執行。例如，您必須在刪除主要執行個體之前刪除所有僅供讀取複本，以避免啟動容錯移轉機制。

如果您需要保持資料庫叢集執行中但想要減少容量，請勿使用啟動和停用。如果您的叢集太昂貴或不是非常忙碌，您可刪除一個或多個資料庫執行個體，或將您的所有資料庫執行個體變更為使用小型執行個體類別，但無法停止個別資料庫執行個體。

停止 Neptune 資料庫叢集

當您暫時不使用它時，您可以停止執行中的 Neptune DB 叢集，然後在需要時再次啟動它。叢集停用時，需支付您指定的保留時段內叢集儲存、手動快照和自動備份儲存的費用，但無須支付資料庫執行個體小時數。

停止作業會先停止所有叢集的僅供讀取複本執行個體，然後再停止主要執行個體，以避免啟動容錯移轉機制。

停止資料庫叢集AWS Management Console

使用停AWS Management Console止 Neptune 叢集

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇一個叢集。您可以從這個頁面執行停用操作，或導覽至欲停用資料庫叢集的詳細資訊頁面。
3. 針對 Actions (動作)，選擇 Stop (停止)。

停止資料庫叢集AWS CLI

若要使用停止資料庫執行個體AWS CLI，請呼叫stop-db-cluster (p. 1027)命令，並使用`--db-cluster-identifier`參數識別要停止的資料庫叢集。

Example

```
aws neptune stop-db-cluster --db-cluster-identifier mydbcluster
```

使用 Neptune 管理 API 停止資料庫叢集

若要使用 Neptune 管理 API 停止資料庫執行個體，請呼叫 StopDBCluster (p. 1027) API，並使用該DBClusterIdentifier參數來識別要停止的資料庫叢集。

停止資料庫叢集停止時會發生什麼情況

- 您可以從快照還原它 (請參閱從資料庫叢集快照還原 (p. 811))。
- 您無法修改資料庫叢集或其任何資料庫執行個體的組態。
- 您無法從叢集新增或移除資料庫執行個體。
- 如果叢集仍然有任何關聯的資料庫執行個體，則無法刪除叢集。
- 一般而言，您必須重新啟動已停止的資料庫叢集，才能執行大部分的管理動作。
- Neptune 會在您已停止的叢集重新啟動時，Neptune 會將任何排程的維護套用至這個 請記住，Neptune 會在七天後自動啟動已停止的叢集，使它不會落後於維護狀態太遠。
- Neptune 不會對已停止的資料庫叢集執行任何自動備份，因為當叢集停止時，Aeptune 不會變更。
- Neptune 不會在資料庫叢集停止時延長這個保留期。

啟動停止的 Neptune 資料庫叢集

您僅會啟動 Neptune 資料庫叢集，處於停止狀態。當您啟動叢集時，所有其資料庫執行個體會再次變為可用。叢集會保留其組態設定，例如端點、參數群組及 VPC 安全群組。

使用啟動已停止的資料庫叢集AWS Management Console

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)，然後選擇一個叢集。您可以從這個頁面執行啟動操作，或導覽至資料庫叢集的詳細資訊頁面並從此處開始。
3. 在 Actions (動作) 中，選擇 Start (啟動)。

使用啟動已停止的資料庫叢集AWS CLI

若要使用啟動已停止的資料庫叢集AWS CLI，請使用--db-cluster-identifier參數呼叫start-db-cluster (p. 1023)命令，以指定要啟動的已停止資料庫叢集。提供您在建立資料庫叢集時選擇的叢集名稱，或使用您選擇的資料庫執行個體名稱，並在其末尾-cluster附加。

Example

```
aws neptune start-db-cluster --db-cluster-identifier mydbcluster
```

使用 Neptune 管理 API 啟動已停止的資料庫叢集

若要使用 Neptune 管理 API 啟動 Neptune 資料庫叢集，請使用DBCluster參數呼叫 StartDBCluster (p. 1023) API，以指定要啟動的已停止資料庫叢集。提供您在建立資料庫叢集時選擇的叢集名稱，或使用您選擇的資料庫執行個體名稱，並在名稱結尾附加 -cluster。

使用快速重設 API 來清空 Amazon Neptune 資料庫叢集

Neptune 快速重置 REST API 可讓您快速輕鬆地重置 Neptune 圖形，從而刪除其所有數據。

您可以使用 %db_reset (p. 541) 線路魔法在 Neptune 筆記本中執行此操作。

> **Note**
>
> 此功能從 Neptune 引擎版本 1.0.4.0 (p. 950) 開始提供。

- 在大多數情況下，快速重置操作會在幾分鐘內完成。持續時間可能會因啟動作業時叢集上的負載而有所不同。
- 快速重設作業不會產生額外的 I/O。
- 快速重置後，存儲卷大小不會縮小。相反地，儲存區會在插入新資料時重複使用。這表示快速重設作業之前和之後所拍攝的快照磁碟區大小將相同。使用快速重設作業之前和之後建立的快照，還原叢集的磁碟區大小也會相同
- 在重設操作中，資料庫叢集中的所有執行個體皆會重新開機。

  > **Note**
  >
  > 在極少數情況下，這些伺服器重新啟動也可能導致叢集容錯移轉。

  > **Important**
  >
  > 使用快速重設可能會中斷 Neptune 資料庫叢集與其他服務的整合。例如：
  >
  > - 快速重設會刪除資料庫中的所有串流資料，並完全重設串流。這意味著，如果沒有新的配置，您的流消費者可能不再工作。
  > - 快速重設會移除 Neptune ML 正在使用之資 SageMaker 源的所有中繼資料，包括作業和端點。它們繼續存在於中 SageMaker，而且您可以繼續使用現有的 SageMaker 端點進行 Neptune ML 推論查詢，但 Neptune ML 管理 API 不再適用於它們。
  > - 通過快速重 ElasticSearch 置也消除了集成之類的集成，並且必須手動重新建立，然後才能再次使用它們。 full-text-search

使用 API 刪除 Neptune 資料庫叢集中的所有資料

1. 首先，您會產生一個 Token，然後可以用來執行資料庫重設。此步驟旨在協助防止任何人意外重設資料庫。

   您可以透過向資料庫叢集的寫入器執行個體上的`/system`端點傳送`HTTP POST`請求以指定動作來執行此`initiateDatabaseReset`操作。

   使用 JSON 內容類型的`curl`命令是：

   ```
   curl -X POST \
     -H 'Content-Type: application/json' \
         https://your_writer_instance_endpoint:8182/system \
     -d '{ "action" : "initiateDatabaseReset" }'
   ```

   或者，使用內`x-www-form-urlencoded`容類型：

   ```
   curl -X POST \
     -H 'Content-Type: application/x-www-form-urlencoded' \
         https://your_writer_instance_endpoint:8182/system \
   ```

```
  -d 'action=initiateDatabaseReset '
```

initiateDatabaseReset請求在其 JSON 響應中返回重置令牌，如下所示：

```
{
  "status" : "200 OK",
  "payload" : {
    "token" : "new_token_guid"
  }
}
```

令牌在發行後，有效期為一小時（60 分鐘）。

如果您將請求發送到讀取器實例或狀態端點，Neptune 將拋出ReadOnlyViolationException.

如果您發送多個initiateDatabaseReset請求，則只有生成的最新令牌對第二個步驟有效，即您實際執行重置。

如果服務器在您的initiateDatabaseReset請求後立即重新啟動，則生成的令牌將變為無效，並且您需要發送新請求以獲取新令牌。

2. 接下來，您將帶有從中initiateDatabaseReset返回的令牌的performDatabaseReset請求發送到數據庫集群的寫入器實例上的/system端點。這會從資料庫叢集中刪除所有資料。

   使用 JSON 內容類型的curl命令是：

```
curl -X POST \
  -H 'Content-Type: application/json' \
      https://your_writer_instance_endpoint:8182/system \
  -d '{
        "action" : "performDatabaseReset",
        "token" : "token_guid"
      }'
```

   或者，使用內x-www-form-urlencoded容類型：

```
curl -X POST \
  -H 'Content-Type: application/x-www-form-urlencoded' \
      https://your_writer_instance_endpoint:8182/system \
  -d 'action=performDatabaseReset&token=token_guid'
```

   該請求返回一個 JSON 響應。如果請求被接受，則響應是：

```
{
  "status" : "200 OK"
}
```

   如果您發送的令牌與發行的令牌不匹配，則響應如下所示：

```
{
  "code" : "InvalidParameterException",
  "requestId":"token_guid",
  "detailedMessage" : "System command parameter 'token' : 'token_guid' does not match
 database reset token"
}
```

   如果要求被接受且重設開始，伺服器會重新啟動並刪除資料。重置時，您無法向數據庫集群發送任何其他請求。

使用帶有 IAM 身份驗證的快速重置 API

如果您在數據庫群集上啟用了 IAM 身份驗證，則可以使用 awscurl 發送使用 IAM 身份驗證的快速重置命令：

使用 awscurl 發送帶有 IAM 身份驗證的快速重置請求

1. 正確設置AWS_ACCESS_KEY_ID和AWS_SECRET_ACCESS_KEY環境變量（AWS_SECURITY_TOKEN如果您正在使用臨時憑據）。
2. 請initiateDatabaseReset求看起來像這樣：

```
awscurl -X POST --service neptune-db "$SYSTEM_ENDPOINT" \
  -H 'Content-Type: application/json' --region us-west-2 \
  -d '{ "action" : "initiateDatabaseReset" }'
```

3. performDatabaseReset請求看起來像這樣：

```
awscurl -X POST --service neptune-db "$SYSTEM_ENDPOINT" \
  -H 'Content-Type: application/json' --region us-west-2 \
  -d '{ "action" : "performDatabaseReset" }'
```

使用 Neptune 工作台%db_reset線路魔術重置數據庫集群

Neptune 工作台支援%db_reset線路魔術，可讓您在 Neptune 筆記型電腦中執行快速資料庫重設。

如果在沒有任何參數的情況下叫用魔術，您會看到一個畫面，詢問您是否要刪除叢集中的所有資料，並顯示一個核取方塊，要求您確認在刪除叢集資料後將無法再使用叢集資料。此時，您可以選擇繼續並刪除數據，或取消操作。

更危險的選項是%db_reset使用--yes或選-y項來叫用，這會導致執行刪除，而不會再提示。

您還可以通過兩個步驟執行重置，就像使用 REST API 一樣：

```
%db_reset --generate-token
```

回應為：

```
{
  "status" : "200 OK",
  "payload" : {
    token" : "new_token_guid"
  }
}
```

則執行：

```
%db_reset --token new_token_guid
```

回應為：

```
{
  "status" : "200 OK"
}
```

快速重設操作的常見錯誤代碼

Neptune 錯誤代碼	HTTP 狀態	訊息	範例
InvalidParameterException	400	系統命令參數 '*##*' 具有不支援的值 '*XXX*'	參數無效
InvalidParameterException	400	為以下項目提供了太多的值：*action*	帶有標題「內容類型：應用程序/x-www-form-urlencoded」發送的多個操作的快速重置請求
InvalidParameterException	400	重複欄位「動作」	帶有「內容類型：應用程序/json' 發送的多個操作的快速重置請求
MethodNotAllowedException	400	錯誤的路由：/*##*	傳送至不正確端點的要求
MissingParameterException	400	缺少必需的參數：[動作]	快速重置請求不包含所需的「操作」參數
ReadOnlyViolationException	400	僅供讀取複本執行個體皆不允許寫入	快速重設要求已傳送至讀取器或狀態端點
AccessDeniedException	403	缺少身分驗證字符	在啟用 IAM 身份驗證的情況下，將快速重置請求發送到數據庫端點，但沒有正確的簽名
ServerShutdownException	500	資料庫重設正在進行。請在叢集可用之後重試查詢。	當快速重設開始時，現有和傳入的 Grimlin/SPARQL 查詢會失敗。

使用主控台建立 Neptune 複本

為 Neptune 資料庫叢集建立主要執行個體後，您可以使用 Amazon Neptune 主控台上的 [建立僅供讀取複本] 頁面來新增其他 Neptune 複本。

若要使用建立 Neptune 複本AWS Management Console

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取 Neptune 資料庫叢集之主要執行個體執行個體的核取方塊。
4. 選擇 Instance Actions (執行個體動作)，然後選擇 Create read replica (建立僅供讀取複本)。
5. 在 [建立複本資料庫執行個體] 頁面上，指定 Neptune 複本的選項。下表顯示 Neptune 僅供讀取複本的設定。

對於此選項...	執行此作業
Availability zone (可用區域)	指定可用區域。選擇與主要資料庫執行個體不同的區域。清單只包含資料庫叢集之資料庫子網路群組所對應的那些可用區域。
加密	啟用或停用加密。
DB instance class (資料庫執行個體類別)	選擇資料庫執行個體類別，為 Neptune 複本的處理和記憶體要求。如需 Neptune 在不同區域提供的資料庫執行個體類別目前清單，請參閱 Neptune 定價頁面。
Read replica source (僅供讀取複本來源)	選擇要為其建立 Neptune 複本之主要執行個體的識別符。
DB instance identifier (資料庫執行個體識別符) :	輸入執行個體的名稱，該名稱在您選取的區域中針對您的帳戶必須是唯一的。您可以選擇新增某些情報至名稱 (例如包括您選取的可用區域)，例如 `neptune-us-east-1c`。
DB parameter group (資料庫參數群組)	此執行個體的參數群組。
自動次要版本升級	如果要讓 Neptune 複本自動接收可用的次要 Neptune 資料庫複本，自動接收可用的次要 Neptune 資料庫複本，接收可用 Auto Minor Version Upgrade (自動次要版本升級) 選項僅適用於次要升級。該選項不適用於引擎維護修補程式，這些修補程式一律會自動套用，以保持系統穩定。

6. 選擇建立僅供讀取複本以建立 Neptune 複本執行個體。

使用主控台修改 Neptune 資料庫叢集

使用 AWS Management Console 修改資料庫執行個體時，您可以透過選取 Apply Immediately (立即套用)，來選擇立即套用變更。若您選擇立即套用變更，則系統會立即套用待定修改佇列中的新變更及任何變更內容。

若您未選擇立即套用變更，則系統會將變更內容放入待定修改佇列。在下一次維護時段期間，系統便會套用佇列中的任何待定變更。

> Important
>
> 如果有任何擱置中的修改需要停機才能執行，選擇立即套用變更可能會導致有問題的資料庫執行個體未預期的停機。資料庫叢集中的其他資料庫執行個體沒有任何停機時間。

> Note
>
> 當您在 Neptune 中修改資料庫叢集時，[立即套用] 設定只會影響對資料庫叢集識別碼 (IAM DB 驗證) 的變更。所有其他的修改都會立即套用，無論 Apply Immediately (立即套用) 設定的值為何。

使用主控台修改資料庫叢集

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Clusters (叢集)，然後選擇您要修改的資料庫叢集。
3. 選擇 Actions (動作)，然後選擇 Modify cluster (修改叢集)。Modify DB cluster (修改資料庫叢集) 頁面隨即出現。
4. 變更您要的任何設定。

   > Note
   >
   > 在主控台上，某些執行個體層級變更僅會套用到目前的資料庫執行個體，而其他的變更則會套用到整個資料庫叢集。若要在主控台上變更於執行個體層級修改整個資料庫叢集的設定，請遵循修改資料庫叢集中的資料庫執行個體 (p. 544)中的說明。
5. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看摘要。
6. 若要立即套用變更，請選取 Apply immediately (立即套用)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify cluster (修改叢集) 以儲存您的變更。

   若要編輯您的變更，請選擇 Back (返回)，或是若要取消您的變更，請選擇 Cancel (取消)。

修改資料庫叢集中的資料庫執行個體

使用主控台修改資料庫叢集中的資料庫執行個體

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中選擇 Instances (執行個體)，然後選擇您要修改的資料庫執行個體。
3. 選擇 Instance actions (執行個體動作)，然後選擇 Modify (修改)。Modify DB Instance (修改資料庫執行個體) 頁面隨即出現。
4. 變更您要的任何設定。

   > Note
   >
   > 有些設定會套用到整個資料庫叢集，且必須在叢集層級進行變更。若要變更這些設定，請遵循使用主控台修改 Neptune 資料庫叢集 (p. 544)中的說明。
   > 在 AWS Management Console中，某些執行個體層級變更僅會套用到目前的資料庫執行個體，而其他的變更則會套用到整個資料庫叢集。

5. 當所有變更都如您所願時，請選擇 Continue (繼續) 並查看摘要。
6. 若要立即套用變更，請選取 Apply immediately (立即套用)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify DB Instance (修改資料庫執行個體) 以儲存您的變更。

   若要編輯您的變更，請選擇 Back (返回)，或是若要取消您的變更，請選擇 Cancel (取消)。

將 Neptune 複本新增至資料庫叢集

在 Neptune 叢集中，有一個主要資料庫執行個體，以及最多 15 個 Neptune 複本。主要資料庫執行個體支援讀寫操作，並對叢集磁碟區執行所有資料修改。Neptune 執行個體連接到與主要資料庫執行個體相同的儲存磁碟區，儘支援讀取操作。Neptune 執行個體中的讀取工作負載。

建議您將資料庫叢集中的主要執行個體和 Neptune 複本分配在多個可用區域中，以改善資料庫叢集的可用性。

如需 Neptune 複本的相關詳細資訊，請參閱使用主控台建立 Neptune 複本 (p. 543)。

將 Neptune 複本新增至資料庫叢集

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中選擇資料庫，然後選擇資料庫叢集。
3. 選擇 [動作]，然後選擇 [新增閱讀器]。

   [建立僅供讀取複本資料庫執行個體] 頁面
4. 在 [建立僅供讀取複本資料庫執行個體] 頁面上，指定複本的選項。下表顯示 Neptune 複本的設定。

針對此選項	執行此作業
DB instance class (資料庫執行個體類別)	選擇資料庫執行個體類別，為您的複本選擇要使用的處理和記憶體容量。
Availability zone (可用區域)	決定您是否要指定特定的可用區域 (AZ)。清單只包含資料庫叢集的資料庫叢集的資料庫子網路群組所映射的那些可用區域。
加密	如果資料庫叢集中的執行個體尚未啟用靜態加密，請選取啟用加密以啟用。如需詳細資訊，請參閱在靜態時加密海王星資源 (p. 155)。
Read replica source (僅供讀取複本來源)	選取要為其建立僅供讀取複本之執行個體的識別符。
DB instance identifier (資料庫執行個體識別符)：	輸入新執行個體的名稱，該名稱在資料庫叢集所在的AWS區域中，該名稱在您的帳戶中必須是唯一的。您可以選擇在名稱中包含有用的資訊，例如選取的資料庫引擎 (例如**`gremlin-read-instance1`**)。
Database port (資料庫連接埠)	Neptune 叢集的連接埠與資料庫叢集的連接埠相同。預設值為 8182。
DB parameter group (資料庫參數群組)	選擇一個 parameter groups (參數群組)。Neptune 群組可供您使用，您也可以建立自己的參數群組。如需參數群組的詳細資訊，請參閱Amazon Neptune 參數 (p. 530)。
稽核記錄	如果您要透 CloudWatch 過記錄使用稽核記錄，請選取 [稽核記錄]。您也必須將資料庫叢集`neptune_enable_audit_log`參數群組中的參數設定為`enable` (1)，才能啟用稽核記錄。
Auto minor version upgrade (自動次要版本升級)	如果您要讓 Neptune 叢集自動接收可用的次要引擎版本升級。

針對此選項	執行此作業
	自動次要版本升級選項僅適用於 Amazon Neptune 資料庫叢集的次要引擎版本的升級。不適用於為維護系統穩定性而套用的一般修補程式。

5. 選擇 [建立 Neptune 複本] 以建立 Neptune 複本。

若要從資料庫叢集移除 Neptune 複本，請遵循中刪除 Amazon Neptune 中的數據庫實例 (p. 575)的指示刪除 Neptune 複本資料庫執行個體。

Amazon Neptune 中的效能和擴展

Neptune 資料庫叢集和執行個體可在三個不同層級擴充

- 儲存體擴展 (p. 548)
- 執行個體擴展； (p. 548)
- 讀取擴展 (p. 548)

Neptune 中的儲存擴充

Neptune 儲存體會隨著叢集磁碟區中的資料而自動擴展。隨著您的資料成長，您的叢集磁碟區儲存體最多可擴充功能至 128 TiB。 GovCloud

每小時會檢查一次叢集磁碟區的大小，以決定儲存成本。

Neptune 資料庫使用的儲存體以每 GB 月遞增計費，而使用的 I/O 則以每百萬個要求增量計費。您只需為 Neptune 資料庫所耗用的儲存和 I/O 付費，而且不需要事先佈建。

如需定價資訊，請參閱 Neptune 產品頁面。

Neptune 中的執行個體

您可以修改資料庫叢集中每個資料庫執行個體的資料庫執行個體類別，視需要擴展 Neptune 資料庫叢集。Neptune 支援多個最佳化資料庫執行個體

讀取 Neptune 中的縮放

您可以在資料庫叢集中建立最多 15 個 Neptune 複本，以達成 Neptune 資料庫叢集的讀取擴展。在主要執行個體寫入更新之後，每個 Neptune 複本會以最短的複本延遲從叢集磁碟區傳回相同的資料（通常遠低於 100 毫秒）。隨著讀取流量增加，您可以建立更多 Neptune 複本，並直接連接這些複本，以分散資料庫叢集的讀取負載。Neptune 複本的資料庫執行個體類別不必與主要執行個體相同。

如需新增 Neptune 複本到資料庫叢集的詳細資訊，請參閱新增 Neptune 複本 (p. 543)。

自動調整 Amazon Neptune 資料庫叢集中的複本數量

您可以使用 Neptune 自動調整規模，自動調整資料庫叢集中的 Neptune 複本數目，以符合您的連線能力和工作負載需求。自動調整規模可讓 Neptune DB 叢集處理工作負載的增加，然後，當工作負載減少時，自動調整規模會移除不必要的複本，因此您無需支付未使用的容量費用。

Neptune DB 叢集只能使用自動調度資源，該叢集已有一個主要寫入器執行個體和至少一個僅供讀取複本執行個體 (請參閱亞馬遜海王星資料庫叢集和執 (p. 117))。此外，叢集中的所有僅供讀取複本執行個體都必須處於可用狀態。如果任何僅供讀取複本處於非可用狀態，Neptune 自動調度資源將不會執行任何動作，直到叢集中的每個僅供讀取複本都可用。

查看是建立資料庫叢集 (p. 54)否需要建立新叢集。

使用AWS CLI，您可以定義擴展政策 (p. 551)並將其套用至資料庫叢集。您也可以使用編AWS CLI輯或刪除自動調度資源政策。此原則會指定下列自動調整規模參數：

- 叢集中要擁有的複本數目下限和上限。
- 複本新增之`ScaleOutCooldown`間的間隔，以及刪除複本之`ScaleInCooldown`間的間隔。
- 向上或向下擴展的度量和指標觸發值。CloudWatch

Neptune 自動調整規模動作的頻率會以下列幾種方式降低：

- 最初，為了自動縮放以添加或刪除讀取器，`CPUUtilization`高警報必須至少突破 3 分鐘，否則必須突破低警報至少 15 分鐘。
- 在第一次新增或刪除之後，後續 Neptune 自動調度資源動作的頻率受到自動調度資源政策中的`ScaleOutCooldown`和`ScaleInCooldown`設定的限制。

如果您使用的指CloudWatch標達到您在政策中指定的高臨界值，並且自上次自動調度資源調整動作以來的`ScaleOutCooldown`間隔已經過去，並且您的資料庫叢集還沒有您設定的最大複本，Neptune auto Scaling 會使用與資料庫叢集的主要執行個體相同的執行個體類型建立新複本。

同樣地，如果指標達到您指定的低臨界值，而且自上次自動調整資源調整動作後的`ScaleInCooldown`間隔已經過去，而且您的資料庫叢集的複本數目超過您指定的最小數目，Neptune 自動調整規模會刪除其中一個複本。

> **Note**
>
> Neptune 自動調整規模只會移除其建立的複本。它不會移除預先存在的複本。

您也可以使用海王星 _ 自動擴展 _ 配置 (p. 532)資料庫叢集參數，指定 Neptune 自動調整規模建立的新僅供讀取複本的執行個體類型、這些僅供讀取複本的維護時段，以及與每個新僅供讀取複本相關聯的標記。您可以在 JSON 字串中提供這些組態設定做為`neptune_autoscaling_config`參數的值，如下所示：

```
"{
  \"tags\": [
    { \"key\" : \"reader tag-0 key\", \"value\" : \"reader tag-0 value\" },
    { \"key\" : \"reader tag-1 key\", \"value\" : \"reader tag-1 value\" },
  ],
  \"maintenanceWindow\" : \"wed:12:03-wed:12:33\",
  \"dbInstanceClass\" : \"db.r5.xlarge\"
}"
```

請注意，JSON 字串中的引號必須全部以反斜線字元 (\) 逸出。像往常一樣，字符串中的所有空格都是可選的。

未在neptune_autoscaling_config參數中指定的三個組態設定中的任何一個，都會從資料庫叢集的主要寫入器執行個體的組態中複製。

自動調度資源新增僅供讀取複本執行個體時，它會在資料庫執行個體 ID 前置 autoscaled-reader (例如，)。autoscaled-reader-7r7t7z3lbd-20210828它也會在每個使用金鑰autoscaled-reader和值建立的僅供讀取複本中新增一個標籤。TRUE您可以在中資料庫執行個體詳細資訊頁面的 [標籤] 索引標籤上看到此標籤AWS Management Console。

```
"key" : "autoscaled-reader",  "value" : "TRUE"
```

透過自動調整規模建立的所有僅供讀取複本執行個體的促銷層是最低的優先順序，15依預設。這表示在容錯移轉期間，會先提升任何較高優先順序的複本 (例如手動建立的複本)。請參閱 Neptune 資料庫叢集的容錯能力 (p. 807)。

Neptune 自動調整規模是使用「應用程式自動調整」來實作，搭配使用 Neptune CPUUtilization (p. 775)CloudWatch度量做為預先定義量度的目標追蹤調整規

在 Neptune 無伺服器資料庫叢集中使用自動調整

當需求超過執行個體的容量時，Neptune Serverless 無伺服器的回應速度會比 Neptune 自動擴展快得多，而不是新增其他執行個體。自動調整規模的設計是為了配合相對穩定的工作負載增加或減少，無伺服器擅長處理需求的快速尖峰和抖動。

瞭解它們的優勢，您可以結合自動調整規模和無伺服器，以建立彈性的基礎架構，以有效處理工作負載的變更並滿足需求，同時將成本降至最低。

若要允許自動調整規模與無伺服器一起有效運作，請務必將無伺服器叢集的maxNCU (p. 579)設定設定設定足夠高，以因應尖峰和短暫的需求變更。否則，暫時性變更不會觸發無伺服器擴展，這可能會導致自動調整規模啟動許多不必要的額外執行個體。如果設maxNCU定足夠高，無伺服器擴展可以更快速且成本更低地處理這些變更。

如何為亞馬遜海王星啟用自動擴展

只能使用 Neptune 資料庫叢集啟用自動調整規模。AWS CLI您無法使用啟用自動調整比例。AWS Management Console

此外，下列 Amazon 區域不支援自動調度資源：

- 非洲（開普敦）：af-south-1
- 中東（阿聯酋）：me-central-1
- AWSGovCloud(美國東部): us-gov-east-1
- AWSGovCloud(美國西部): us-gov-west-1

為 Neptune 資料庫叢集啟用自動調整規模需要三個步驟：

1. 使用應用程式自動調整註冊資料庫叢集

啟用 Neptune 資料庫叢集自動調整規模的第一個步驟是使用AWS CLI或其中一個應用程式自動調整 SDK 向應用程式自動調整規模註冊叢集。叢集必須已有一個主要執行個體和至少一個僅供讀取複本執行個體：

例如，若要註冊叢集以自動調整規模，從一到八個額外的複本，您可以使用AWS CLIregister-scalable-target命令，如下所示：

```
aws application-autoscaling register-scalable-target \
```

```
  --service-namespace neptune \
  --resource-id cluster:(your DB cluster name) \
  --scalable-dimension neptune:cluster:ReadReplicaCount \
  --min-capacity 1 \
  --max-capacity 8
```

這等同於使用「應用RegisterScalableTarget程式自動調整比例」API 作業。

該AWS CLIregister-scalable-target命令採用以下參數：

- **--service-namespace** – 設為 neptune。

  此參數相當於應用程式自動調整比例 API 中的ServiceNamespace參數。
- **resource-id**— 將其設定為 Neptune 資料庫叢集的資源識別碼。資源類型是cluster，後跟一個冒號（':'），然後是數據庫集群的名稱。

  此參數相當於應用程式自動調整比例 API 中的ResourceID參數。
- **scalable-dimension**— 在這種情況下，可擴展維度是資料庫叢集中複本執行個體的數量，因此您將此參數設定為neptune:cluster:ReadReplicaCount。

  此參數相當於應用程式自動調整比例 API 中的ScalableDimension參數。
- **min-capacity**— 由應用程式自動擴展管理的讀取器資料庫複本執行個體數目下限。這應該在 1 到 15 的範圍內，包括在內，並且不大於max-capacity（數據庫集群中必須至少有一個讀取器才能自動擴展工作）。

  此參數相當於應用程式自動調整比例 API 中的MinCapacity參數。
- **max-capacity**— 由應用程式自動擴展管理的讀取器資料庫複本執行個體數目上限。此值不能小於min-capacity參數的值，且不得大於 15 減去預先存在的複本數目。資料庫叢集總共不能有超過 15 個複本，因此透過自動調整規模管理的最大複本數目加上預先存在的複本數目不能超過該限制。

  此max-capacityAWS CLI參數相當於「應用程式自動調整比例」API 中的MaxCapacity參數。

當您註冊資料庫叢集時，應用程式 Auto Scaling 會建立AWSServiceRoleForApplicationAutoScaling_NeptuneCluster服務連結角色。如需詳細資訊，請參閱應用程式自動調整規模使用指南中的應用程式自動調整規模的服務連結

2. 定義與資料庫叢集搭配使用的自動調度資源政策

目標追蹤擴展政策定義為 JSON 文字物件，也可以儲存在文字檔中。對於 Neptune，此原則目前只能使用 Neptune CPUUtilization (p. 775)CloudWatch度量做為名為的預先定義度量NeptuneReaderAverageCPUUtilization。

以下是 Neptune 的目標追蹤擴展設定原則範例：

```
{
  "PredefinedMetricSpecification": { "PredefinedMetricType":
 "NeptuneReaderAverageCPUUtilization" },
  "TargetValue": 60.0,
  "ScaleOutCooldown" : 600,
  "ScaleInCooldown" : 600
}
```

這裡的**TargetValue**元素包含 CPU 使用率的百分比，超過該百分比的自動擴展（也就是增加更多複本），低於其擴展（也就是刪除複本）。在此情況下，觸發縮放的目標百分比為60.0%。

ScaleInCooldown元素會指定縮放活動完成之後，在另一個縮放可以開始之前的時間量 (以秒為單位)。預設為 300 秒。此處的值 600 指定在完成一個複本刪除與另一個複本開始之間至少必須經過十分鐘。

ScaleOutCooldown元素指定向外延展活動完成之後，在另一個向外延展可以開始之前的時間量 (以秒為單位)。預設為 300 秒。此處的值 600 指定在新增一個複本完成與另一個複本開始之間至少必須經過十分鐘。

該**DisableScaleIn**元素是一個布爾值，如果存在並設置為完全`true`禁用縮放，這意味著自動縮放可能會添加複本，但永遠不會刪除任何副本。依預設，會啟用縮放，且`DisableScaleIn`為。`false`

使用應用程式 Auto Scaling 註冊 Neptune 資料庫叢集，並在文字檔中定義 JSON 資源調整原則後，接下來將擴展政策套用至已註冊的資料庫叢集。您可以使用AWS CLI`put-scaling-policy`命令來執行此操作，參數如下：

```
aws application-autoscaling put-scaling-policy \
  --policy-name (name of the scaling policy) \
  --policy-type TargetTrackingScaling \
  --resource-id cluster:(name of your Neptune DB cluster) \
  --service-namespace neptune \
  --scalable-dimension neptune:cluster:ReadReplicaCount \
  --target-tracking-scaling-policy-configuration file://(path to the JSON configuration
 file)
```

套用自動調整規模政策後，資料庫叢集會啟用自動調整規模。

您也可以使用AWS CLI`put-scaling-policy`命令來更新現有的自動資源調度資源政策。

另請參閱應用程PutScalingPolicy式自動調整規模 API 參考中的。

從 Neptune 資料庫叢集移除自動調整規模

若要從 Neptune 資料庫叢集移除自動調整規模，請使用AWS CLIdelete-scaling-policy和deregister-scalable-target命令。

維護 Amazon Neptune 資料庫叢集

Neptune 會定期對 Neptune 資源執行維護。維護作業大多數包括針對資料庫叢集基礎作業系統或資料庫引擎版本的更新。作業系統更新大多是因為安全性問題，且需要盡快完成。您可以使用執行個體狀態 API (p. 770) 來判斷您目前安裝的引擎發行版本。

維護項目時，Neptune 需要將您的資料庫叢集短暫離線。需要資源離線才能進行的維護包括調整運算操作的規模 (通常從開始到結束只需要幾分鐘)，以及必要的作業系統或資料庫修補。所需的修補程式僅會針對與安全性和執行個體可靠性相關的修補程式自動安排。這種修補不常發生 (通常幾個月才一次)，且幾乎不需要太長的維護時段。

套用 OS 更新時，資料庫執行個體不會自動進行備份。因此建議您在套用更新前備份您的資料庫執行個體。

您可以使用 Neptune 主台來檢視您的資料庫叢集是否有可以使用的維護更新。如果有可用的更新，Neptune 主控台上資料庫叢集的 [維護] 資料行中會以 [可用] 或 [必要] 一字表示更新。

若有可用的更新，您可以執行以下任何作業：

- 延遲維護項目。
- 立即套用維護項目
- 排定要在下一個維護時段中啟動的維護項目。
- 不採取動作。

> **Note**
>
> 某些作業系統更新會標記為 Required (必要)。如果您延遲必要的更新，您會收到 Neptune 的通知，指出將在何時執行更新。其他更新會標記為 Available (可用)，您可以無限期延遲這些更新。

維護時段會判斷待定的操作何時開始，但不限制這些操作的總執行時間。維護操作無法保證能在維護時段結束之前完成，並且可以在指定的結束時間後繼續進行。如需詳細資訊，請參閱Neptune 護時段 (p. 553)。

Neptune 護時段

每個資料庫叢集都有每週一次的維護時段，任何系統變更都會在此期間套用。您可以將維護時段視為依請求或要求事件執行修改及軟體修補作業的時機。若在特定某週排定維護事件，系統將在您指定的 30 分鐘維護時段內啟動該事件。此外，多數維護事件也能在 30 分鐘的維護時段內完成，但較大型的維護事件可能需要 30 分鐘以上才能完成。

30 分鐘的維護時段是從每個區域之 8 小時時段內隨機選取的。若您在資料庫叢集建立期間，沒有指定偏好的維護時段，則 Neptune 會在一週內隨機選取一天，並指派 30 分鐘的維護時段。

Neptune 會使用資料庫叢集上的部分資源。您可能會發現，該操作對效能會造成些許影響。在極少數情況下，資料庫執行個體需要執行異地同步備份容錯移轉，才能完成維護更新作業。

以下是每個AWS區維護時段的時段，系統會從此時段中指派預設維護時段

區域	時間區塊
美國西部 (奧勒岡) 區域	上午 6 時至下午 2 時 (UTC)
美國西部 (加利佛尼亞北部) 區域	上午 6 時至下午 2 時 (UTC)
美國東部 (俄亥俄) 區域	上午 3 時至上午 11 時 (UTC)
歐洲 (愛爾蘭) 區域	下午 10 時至次日上午 6 時 (UTC)

使用主控台調整慣用資料庫叢集維護時段

Neptune 資料庫叢集的維護時段應落在使用量最低的時間。因此，您可能需要不時進行變更。這段時間內，只有套用的更新需要停機時，資料庫叢集才會無法使用。停機是完成必要更新所需的最短時間。

調整偏好的資料庫叢集維護時段

1. 登入AWS管理台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主台。
2. 在導覽窗格中，選擇 Clusters (叢集)。
3. 選擇資料庫叢集，以變更其維護時段。
4. 選擇 Actions (動作)、Modify cluster (修改叢集)。
5. 在 Maintenance (維護) 區段中，更新維護時段。
6. 選擇 Continue (繼續)。

   在確認頁面上，檢閱您的變更。
7. 選取 Apply immediately (立即套用)，就能立刻將變更套用至維護時段。
8. 選擇 Modify cluster (修改叢集)，以儲存變更。

   若要編輯您的變更，請選擇 Back (返回)，或是若要取消您的變更，請選擇 Cancel (取消)。

亞馬遜海神更新

亞馬遜海王星定期發行資料庫引擎更新 客戶可以根據更新類型，選擇手動或自動套用這些更新。在套用更新的維護時段期間，叢集寫入器和讀取器執行個體無法用於處理使用者查詢。大多數維護事件最多需要 30 分鐘才能完成。若要設定叢集的維護時段，請參閱Neptune 護時段 (p. 553)。

過去的引擎版本列在 Amazon Neptune 的引擎版本 (p. 856)。

請注意，海王星號碼和處理更新的方式在 2019 年 11 月發生了變化，如中所述引擎版本號碼 (p. 859)。

您可以使用主控台手動更新引擎版本，將`AutoMinorVersionUpgrade`欄位設定為，以`true`便 Neptune 盡可能自動將引擎更新為最新版本。

您也可以使用AWS CloudFormation範本來升級資料庫叢集的引擎版本 (請參閱通過更新 CloudFormation (p. 556))。

使用主控台將引擎升級為較新的版本

您可以使用下列方式執行 Neptune 資料庫叢集的AWS Management Console次要版本升級：

使用主控台升級資料庫叢集的引擎版本

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Clusters (叢集)，然後選擇您要修改的資料庫叢集。
3. 選擇 Actions (動作)，然後選擇 Modify cluster (修改叢集)。Modify DB cluster (修改資料庫叢集) 頁面隨即出現。
4. 在 DB engine version (資料庫引擎版本) 中，選擇新版本。
5. 選擇 Continue (繼續)，並檢查修改的摘要。
6. 若要立即套用變更，請選擇 Apply immediately (立即套用)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify cluster (修改叢集) 以儲存您的變更。

   或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

設定 Neptune 以自動升級引擎版本

如果您`true`在資料庫叢集的寫入器資料庫執行個體中將`AutoMinorVersionUpgrade`欄位設定為，Neptune 會自動將叢集的引擎更新為最新版本。

您可以使用 AWS Management Console 設定此欄位：

設定資料庫叢集的 **`AutoMinorVersionUpgrade`** 欄位

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在資料庫執行個體清單中，選擇資料庫叢集的主要執行個體 (寫入器執行個體)。
3. 選擇 Instance actions (執行個體動作)，然後選擇 Modify (修改)。Modify DB Instance (修改資料庫執行個體) 頁面隨即出現。
4. 選擇 Actions (動作)，然後選擇 Modify instance (修改執行個體)。Modify DB instance (修改資料庫執行個體) 頁面隨即出現。
5. 設定自動次要版本更新欄位。

如需詳細資訊，請參閱「修改執行個體 (p. 571)」。

使用AWS CloudFormation範本更新 Neptune 資料庫叢集的引擎版本

您可以重複使用用來建立 Neptune 資料庫叢集的 Neptune AWS CloudFormation 範本，以更新其引擎版本。

Neptune 引擎版本升級可以是次要或主要升級。使用AWS CloudFormation範本可協助進行主要版本升級，這些升級通常包含重大變更。由於主要版本升級可能包含與現有應用程式回溯不相容的資料庫變更。請務必在升級前進行測試 (p. 858)，強烈建議您在升級之前一律建立資料庫叢集的手動快照。

請注意，您必須為每個主要版本進行單獨的引擎升級。您無法跳過主要版本並直接升級到以下主要版本。

在 2023 年 5 月 17 日之前，如果您使用 Neptune AWS CloudFormation 堆疊升級引擎版本，它只會建立新的空白資料庫叢集，以取代您目前的資料庫叢集。不過，從 2023 年 5 月 17 日起，Neptune AWS CloudFormation 堆疊現在支援就地引擎升級，以保留您現有的資料。

對於主要版本升級，您的範本應在中設定下列屬性`DBCluster`：

- `DBClusterParameterGroup`(自訂或預設)
- `DBInstanceParameterGroupName`
- `EngineVersion`

同樣，對於附加到數據庫集群的數據庫實例，您應該設置：

- `DBParameterGroup`(自訂/預設)

確保您的所有參數群組都已在範本中定義，無論它們是預設的還是自訂參數群組。

對於自訂參數群組，請確保現有自訂參數群組的族群與新引擎版本相容。1.2.0.0 之前的引擎版本使用了參數群組族群`neptune1`，而 1.2.0.0 (p. 884) 前版的引擎版本則需要參數群組族群。`neptune1.2`如需詳細資訊，請參閱 Amazon Neptune (p. 527)。

對於主要引擎版本升級，請在`DBCluster``DBInstanceParameterGroupName`欄位中指定具有適當族群的參數群組。

預設參數群組應升級為與新引擎版本相容的參數群組。

請注意，Neptune 在引擎升級後自動重新啟動資料庫執行個體。

主題

- 範例：次要引擎從 1.2.0.1 升級至 1.2.0.2 (p. 556)
- 範例：使用預設參數群組從 1.1.1.0 升級至 1.2.0.2 的主要版本 (p. 558)
- 範例：使用自訂參數群組從 1.1.1.0 升級至 1.2.0.2 的主要版本 (p. 559)
- 範例：主要版本從 1.1.1.0 升級至 1.2.0.2，混合使用預設和自訂參數群組 (p. 561)

範例：次要引擎從 1.2.0.1 升級至 1.2.0.2

尋找您要升級的資料庫叢集，以及您用來建立它的範本。例如：

```
Description: Base Template to create Neptune Stack with Engine Version 1.2.0.1 using custom
 Parameter Groups
Parameters:
  DbInstanceType:
```

```
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1.2
      Description: test-cfn-neptune-db-cluster-parameter-group-description
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
      Family: neptune1.2
      Description: test-cfn-neptune-db-parameter-group-description
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.2.0.1
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
    DependsOn:
      - NeptuneDBClusterParameterGroup
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

將EngineVersion屬性從更新1.2.0.1為1.2.0.2：

```
Description: Template to upgrade minor engine version to 1.2.0.2
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1.2
      Description: test-cfn-neptune-db-cluster-parameter-group-description
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
      Family: neptune1.2
      Description: test-cfn-neptune-db-parameter-group-description
```

```
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.2.0.2
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
    DependsOn:
      - NeptuneDBClusterParameterGroup
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

現在使用AWS CloudFormation來執行修訂的範本。

範例：使用預設參數群組從 1.1.1.0 升級至 1.2.0.2 的主要版本

尋找您DBCluster要升級的，以及您用來建立它的範本。例如：

```
Description: Base Template to create Neptune Stack with Engine Version 1.1.1.0 using
 default Parameter Groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.1.1.0
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
    DependsOn:
      - NeptuneDBCluster
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

- DBClusterParameterGroup將預設值更新為新引擎版本 (此處default.neptune1.2) 使用的參數群組族群中的預設值。
- 對於每個DBInstance附加到的DBCluster，請將預設值更新DBParameterGroup為新引擎版本所使用的系列中的預設值 (此處default.neptune1.2)。
- 將DBInstanceParameterGroupName性質設定為該族群中的預設參數群組 (此處default.neptune1.2)。
- 將內EngineVersion容從更新1.1.0.0為1.2.0.2。

該模板如下所示：

```
Description: Template to upgrade major engine version to 1.2.0.2 by using upgraded default
 parameter groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.2.0.2
      DBClusterParameterGroupName: default.neptune1.2
      DBInstanceParameterGroupName: default.neptune1.2
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName: default.neptune1.2
    DependsOn:
      - NeptuneDBCluster
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
```

現在使用AWS CloudFormation來執行修訂的範本。

範例：使用自訂參數群組從 1.1.1.0 升級至 1.2.0.2 的主要版本

尋找您DBCluster要升級的，以及您用來建立它的範本。例如：

```
Description: Base Template to create Neptune Stack with Engine Version 1.1.1.0 using custom
 Parameter Groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1
```

```
      Description: 'NeptuneDBClusterParameterGroup with family neptune1'
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
      Family: neptune1
      Description: 'NeptuneDBParameterGroup1 with family neptune1'
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.1.1.0
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
    DependsOn:
      - NeptuneDBClusterParameterGroup
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

- 將自訂DBClusterParameterGroup系列更新為新引擎版本所使用的系列 (此處neptune1.2)。
- 對於每個DBInstance附加到的DBCluster，請將自訂DBParameterGroup系列更新為新引擎版本所使用的自訂系列 (此處neptune1.2)。
- 將DBInstanceParameterGroupName性質設定為該族群中的參數群組 (此處neptune1.2)。
- 將內EngineVersion容從更新1.1.0.0為1.2.0.2。

該模板如下所示：

```
Description: Template to upgrade major engine version to 1.2.0.2 by modifying existing
 custom parameter groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1.2
      Description: 'NeptuneDBClusterParameterGroup with family neptune1.2'
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
```

```
      Family: neptune1.2
      Description: 'NeptuneDBParameterGroup1 with family neptune1.2'
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.2.0.2
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
      DBInstanceParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBClusterParameterGroup
  NeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

現在使用AWS CloudFormation來執行修訂的範本。

範例：主要版本從 1.1.1.0 升級至 1.2.0.2，混合使用預設和自訂參數群組

尋找您DBCluster要升級的，以及您用來建立它的範本。例如：

```
Description: Base Template to create Neptune Stack with Engine Version 1.1.1.0 using custom
 Parameter Groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1
      Description: 'NeptuneDBClusterParameterGroup with family neptune1'
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
      Family: neptune1
      Description: 'NeptuneDBParameterGroup with family neptune1'
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
```

```
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.1.1.0
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
    DependsOn:
      - NeptuneDBClusterParameterGroup
  CustomNeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
  DefaultNeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
    DependsOn:
      - NeptuneDBCluster
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

- 對於自訂叢集參數群組，請將DBClusterParameterGroup族群更新為對應於新引擎版本的族群，亦即neptune1.2。
- 對於預設叢集參數群組，請DBClusterParameterGroup將更新為對應於新引擎版本的預設值，亦即default.neptune1.2。
- 對於每個DBInstance附加到的DBCluster，請DBParameterGroup將預設值更新為新引擎版本 (此處default.neptune1.2) 所使用的系列中的預設值，並將自訂參數群組更新為使用新引擎版本所支援的族群 (此處neptune1.2)。
- 將DBInstanceParameterGroupName性質設定為新引擎版本支援的族群中的參數群組。

該模板如下所示：

```
Description: Template to update Neptune Stack to Engine Version 1.2.0.1 using custom and
 default Parameter Groups
Parameters:
  DbInstanceType:
    Description: Neptune DB instance type
    Type: String
    Default: db.r5.large
Resources:
  NeptuneDBClusterParameterGroup:
    Type: 'AWS::Neptune::DBClusterParameterGroup'
    Properties:
      Family: neptune1.2
      Description: 'NeptuneDBClusterParameterGroup with family neptune1.2'
      Parameters:
        neptune_enable_audit_log: 0
  NeptuneDBParameterGroup:
```

```
    Type: 'AWS::Neptune::DBParameterGroup'
    Properties:
      Family: neptune1.2
      Description: 'NeptuneDBParameterGroup1 with family neptune1.2'
      Parameters:
        neptune_query_timeout: 20000
  NeptuneDBCluster:
    Type: 'AWS::Neptune::DBCluster'
    Properties:
      EngineVersion: 1.2.0.2
      DBClusterParameterGroupName:
        Ref: NeptuneDBClusterParameterGroup
      DBInstanceParameterGroupName: default.neptune1.2
    DependsOn:
      - NeptuneDBClusterParameterGroup
  CustomNeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName:
        Ref: NeptuneDBParameterGroup
    DependsOn:
      - NeptuneDBCluster
      - NeptuneDBParameterGroup
  DefaultNeptuneDBInstance:
    Type: 'AWS::Neptune::DBInstance'
    Properties:
      DBClusterIdentifier:
        Ref: NeptuneDBCluster
      DBInstanceClass:
        Ref: DbInstanceType
      DBParameterGroupName: default.neptune1.2
    DependsOn:
      - NeptuneDBCluster
Outputs:
  DBClusterId:
    Description: Neptune Cluster Identifier
    Value:
      Ref: NeptuneDBCluster
```

現在使用AWS CloudFormation來執行修訂的範本。

Neptune 的數據庫克隆

使用資料庫複製，您可以快速且具成本效益地建立 Amazon Neptune 中所有資料庫的複製。複製資料庫在第一次建立時只需最少的額外空間。資料庫複製使用通copy-on-write 訊協定。無論在來源資料庫或複製資料庫上，在資料變更時會複製資料。您可以從相同的資料庫叢集建立多個複製。您也可以從其他複製建立額外的複製。如需 copy-on-write 協定如何在 Neptune 儲存體內容中運作的詳細資訊，請參閱寫入時複製通訊協定 (p. 565)。

您可以在各種使用案例中使用資料庫複製，尤其是您不想要對生產環境產生影響的使用案例，如下所示：

- 實驗並評估變更的影響，例如結構描述變更或參數群組變更。
- 執行工作負載密集的操作，例如匯出資料或執行分析查詢。
- 在非生產環境中建立生產資料庫叢集的副本，以進行開發或測試。

使用 AWS Management Console 建立資料庫叢集複本

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。選擇您要建立其複製之資料庫叢集的主要執行個體。
3. 選擇 Instance actions (執行個體動作)，然後選擇 Create clone (建立複製)。
4. 在 Create Clone (建立複製) 頁面上，輸入複製資料庫叢集主要執行個體的名稱，做為 DB instance identifier (資料庫執行個體識別符)。

   如果想要的話，請對複製資料庫叢集設定任何其他設定。如需不同資料庫叢集設定的詳細資訊，請參閱使用主控台啟動 (p. 534)。
5. 選擇 Create Clone (建立複製) 來啟動複製資料庫叢集。

使用 AWS CLI 建立資料庫叢集複本

- 呼叫 Neptune restore-db-cluster-to-point-in-time (p. 1125)AWS CLI 命令並提供以下值：
  - `--source-db-cluster-identifier`— 要建立其複本的來源資料庫叢集的名稱。
  - `--db-cluster-identifier`— 複製資料庫叢集的名稱。
  - `--restore-type copy-on-write`— 此`copy-on-write`值表示應建立複製資料庫叢集。
  - `--use-latest-restorable-time`— 這指定使用最新可還原的備份時間。

    Note

    restore-db-cluster-to- (p. 1125) 命point-in-timeAWS CLI令只會複製資料庫叢集，而不會複製該資料庫叢集的資料庫執行個體。

  下列 Linux/UNIX 範例會從 `source-db-cluster-id` 資料庫叢集建立複本，並將複本命名為 `db-clone-cluster-id`。

```
aws neptune restore-db-cluster-to-point-in-time \
  --region us-east-1 \
  --source-db-cluster-identifier source-db-cluster-id \
  --db-cluster-identifier db-clone-cluster-id \
  --restore-type copy-on-write \
  --use-latest-restorable-time
```

  如果 \ 行尾逸出字元取代為 Windows ^ 對等字元，則相同的範例適用於 Windows：

```
aws neptune restore-db-cluster-to-point-in-time ^
  --region us-east-1 ^
  --source-db-cluster-identifier source-db-cluster-id ^
  --db-cluster-identifier db-clone-cluster-id ^
  --restore-type copy-on-write ^
  --use-latest-restorable-time
```

限制

Neptune 中的資料庫複製的限制如下：

- 您無法跨AWS區域建立複製資料庫。複製資料庫必須建立在與來源資料庫相同的區域中。
- 複製的資料庫一律會使用複製來源資料庫所使用的 Neptune 引擎版本的最新修補程式。即使來源資料庫尚未升級至該修補程式版本，也是如此。但是，引擎版本本身不會改變。
- 目前，您最多可根據其他複製的複製，包括根據其他複製的複製。達到該限制之後，您必須製作另一個資料庫副本，而不是複製它。不過，如果您製作新複本，它最多也可以有 15 個複製。
- 目前不支援跨帳戶資料庫複製。
- 您可以提供不同的 virtual private cloud (VPC) 進行複製。不過，那些 VPC 中的子網路必須映射至相同的可用區域集。

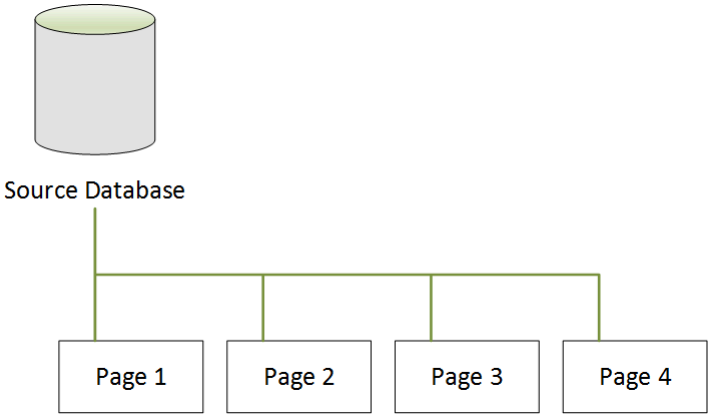
用於資料庫複製的寫入時複製通訊協定

下列範例說明 copy-on-write 協定的運作方式。

- 複製前 Neptune 資料庫 (p. 565)
- 複製後 Neptune 資料庫 (p. 565)
- 對來源資料庫進行變更時 (p. 566)
- 對複製資料庫進行變更時 (p. 566)

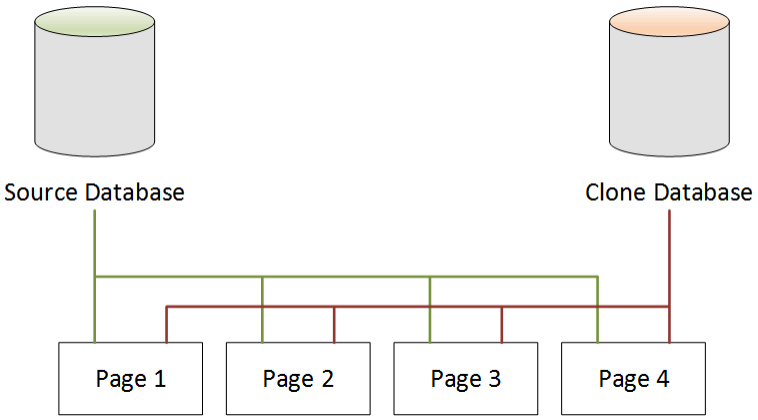
複製前 Neptune 資料庫

來源資料庫中的資料存放於頁面。在下圖中，來源資料庫有四個頁面。



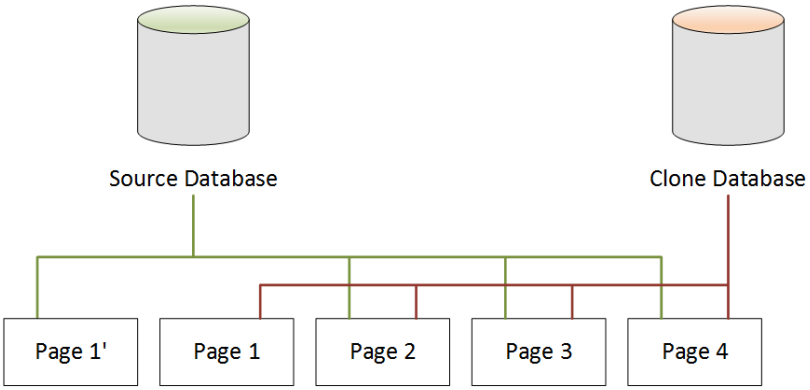
複製後 Neptune 資料庫

如下圖所示，在資料庫複製之後，來源資料庫中沒有任何變更。來源資料庫與複製資料庫都指向相同的四個頁面。不會實際複製任何頁面，所以不需要額外的儲存空間。



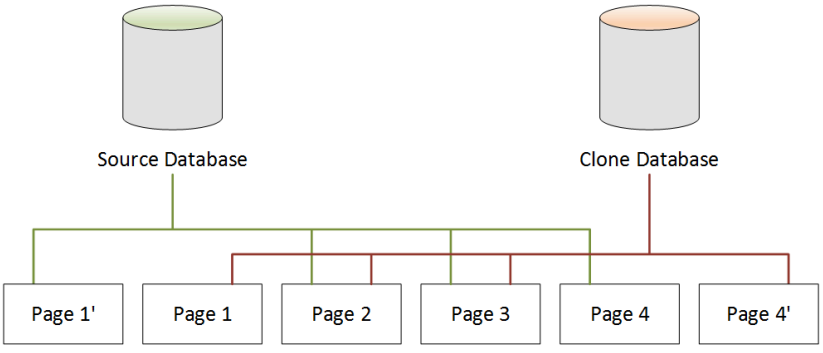
對來源資料庫進行變更時

在下列範例中，來源資料庫會對 Page 1 中的資料進行變更。系統不會寫入至原始 Page 1，而是使用額外儲存空間建立新的頁面，稱為 Page 1'。來源資料庫現在指向新的 Page 1'，同時也指向 Page 2、Page 3 和 Page 4。複製資料庫繼續指向 Page 1 到 Page 4。



對複製資料庫進行變更時

在下圖中，複製資料庫也做了變更，這次在 Page 4 中。系統不會寫入至原始 Page 4，而是使用額外儲存空間建立新的頁面，稱為 Page 4'。來源資料庫繼續指向 Page 1'，同時也指向 Page 2 到 Page 4，但複製資料庫現在會指向 Page 1 到 Page 3，同時也指向 Page 4'。



如第二個案例所示，在資料庫複製之後，在建立複製時不需要額外的儲存空間。不過，當來源資料庫與複製資料庫中發生變更時，只會建立已變更的頁面，如第三個及第四個案例所示。隨著來源資料庫與複製資料庫在一段時間後發生更多變更，您需要不斷增加更多的儲存空間來擷取並存放變更。

刪除來源資料庫

刪除來源資料庫不會影響與其相關聯的複製資料庫。複製資料庫會繼續指向來源資料庫先前所擁有的頁面。

Amazon Neptune 執行個體

以下章節提供有關執行個體層級操作的資訊。

主題

- 海王星 T3 爆發實例類 (p. 569)
- 修改 Neptune 資料庫執行個體 (並立即套用) (p. 571)
- 將 Neptune 資料庫執行個體 (p. 573)
- 在 Amazon Neptune 中重新啟動資料庫 (p. 574)
- 刪除 Amazon Neptune 中的數據庫實例 (p. 575)

海王星 T3 爆發實例類

除了固定效能執行個體類別 (例如R5和) 之外R6，Amazon Neptune 還提供了使用高效能執行個體的選項。T3在開發圖形應用程序時，您希望數據庫快速響應，但不需要始終使用它。Neptune 的db.t3.medium執行個體類別就是您在這種情況下應該使用的執行個體類別，成本遠低於最便宜的固定效能執行個體類別。

高載執行個體會在 CPU 效能的基準層級執行，直到工作負載需要更多，然後只要工作負載需要，就會在工作負載需要時高過基準。如果平均 CPU 使用率在 24 小時期間內未超過基準，則每小時價格會涵蓋高載。對於大多數開發和測試情況，這會轉換為低成本的良好性能。

如果您從T3執行個體類別開始，您可以在稍後準備好使用、或其中一個 AWS SDK 進入生產環境時，輕鬆切換到固定效能的執行個體類別。AWS Management Console AWS CLI

T3 高載由 CPU 點數管理

一個 CPU 點數代表一個虛擬 CPU 核心 (vCPU) 一分鐘的完整使用率。這也可以轉換成 vCPU 兩分鐘的 50% 使用率，或兩分鐘內的 25% 使用率，依此類推。

T3 執行個體會在閒置時產生 CPU 積分，並在作用中時使用積分，兩者均以毫秒為單位測量而得。db.t3.medium 執行個體類別有兩個 vCPU，每個 vCPU 閒置時，每小時獲得 12 個 CPU 信用點數。這表示每個 vCPU 的 20% 使用率會產生零 CPU 點數餘額。獲得的 12 個 CPU 點數支出 vCPU 的 20% 使用率 (因為 60 分鐘的 20% 也是 12)。因此，這 20% 的使用率就是 baseline (基準) 使用率，不會產生正或負的 CPU 點數餘額。

閒置時間 (CPU 使用率低於可用總數的 20%) 會導致 CPU 積分儲存在積分餘額儲存貯體中，db.t3.medium 執行個體類別的上限為 576 (24 小時內可產生的 CPU 積分上限，即 2 x 12 x 24)。超過這個限制，則會捨棄 CPU 積分。

必要時，即使在 CPU 積分餘額已低於零，CPU 使用率可以根據工作負載的需要，到爆量增加至 100%。如果執行個體連續 24 小時維持負餘額，則該期間內產生的每 -60 個 CPU 積分會收取 0.05 美元的額外費用。不過，對於大多數的開發和測試工作負載而言，高載通常是由高載前後的閒置時間所涵蓋。

> **Note**
>
> 海王星的T3執行個體類別設定為 Amazon EC2 無限制模式。

使用 AWS Management Console 建立 T3 高載執行個體

在 AWS Management Console 中，您可以建立主要資料庫叢集執行個體，或使用執行個體類別 db.t3.medium 的讀取複本執行個體，或者您可以修改現有執行個體，以使用 db.t3.medium 執行個體類別。

例如，若要在 Neptune 主控台中建立新的資料庫叢集主要執行個體：

- 選擇 Create Database (建立資料庫)。
- 選擇等於或晚於的資料庫引擎版本1.0.2.2。
- 在 Purpose (目的) 下，選擇 Development and Testing (開發和測試)。
- 做為 DB instance class (資料庫執行個體類別)，接受預設值：db.t3.medium — 2 vCPU, 4 GiB RAM。

使用 AWS CLI 建立 T3 高載執行個體

你也可以使用 AWS CLI 來做同樣的事情：

```
aws neptune create-db-cluster \
```

```
    --db-cluster-identifier (name for a new DB cluster) \
    --engine neptune \
    --engine-version "1.0.2.2"

aws neptune create-db-instance \
    --db-cluster-identifier (name of the new DB cluster) \
    --db-instance-identifier (name for the primary writer instance in the cluster) \
    --engine neptune \
    --db-instance-class db.t3.medium
```

修改 Neptune 資料庫執行個體 (並立即套用)

您可以立即套用變更至 Amazon Neptune 資料庫執行個體，或延遲到下一個維護時段才進行變更。而諸如參數群組變更等部分修改作業，則要求您手動重新啟動資料庫執行個體，才能使變更生效。

> Important
>
> 如果 Neptune 必須重新啟動資料庫執行個體，才能套用變更。因此，請務必先檢視修改作業對資料庫與應用程式所造成的影響，

常用設定與停機時間影響

下表包含您可以變更的設定、套用變更的時間，以及變更是否會造成資料庫執行個體停機的詳細資訊。

資料庫執行個體設定	停機時間備註	
DB instance class (資料庫執行個體類別)	系統停機變更，無論是否要立即套用變更，或等到下個維護時段再套用變更，或等到下一	
DB instance identifier (資料庫執行個體識別符)：	無論是立即套用還是在下一個維護時段套用，資料庫執行個體都會重新啟動，並在此變更期間發生中斷。	
Subnet group (子網路群組)	無論是立即套用還是在下一個維護時段套用，資料庫執行個體都會重新啟動，並在此變更期間發生中斷。	
安全群組	無論何時指定變更應發生，都會盡快以非同步方式套用變更，且不會造成中斷結果。	–
憑證授權單位	根據預設，當您指派新的憑證授權單位時，資料庫執行個體會重新啟動。	
Database Port (資料庫連接埠)	變更永遠會立即發生，導致資料庫執行個體重新開機，並發生中斷。	
DB parameter group (資料庫參數群組)	變更此設定不會導致中斷。參數群組名稱本身會立即變更，但在您重新啓動實例而不需容錯移轉的情況下，才會套用實際參數變更。在此情況下，資料庫執行個體不會自動重新啟動，且參數變更不會在下一個維護時段套用。但是，如果您修改新關聯的資料庫參數群組中的動態參數，則會立即套用這些變更，而不需重新開機。 如需詳細資訊，請參閱在 Amazon Neptune 中重新啟動資料庫 (p. 574)。	

資料庫執行個體設定	停機時間備註	
DB cluster parameter group (資料庫叢集參數群組)	資料庫參數群組名稱會立即變更變更。	
Backup retention period (備份保留期間)	如果您指定應立即發生變更，則此變更不會立即發生。否則，如果您將設定從非零值變更為另一個非零值，則系統會盡快以非同步的方式來套用變更。下個維護時段期間，系統才會進行變更。從零變更為非零值，或是從非零值變更為零時，將會發生中斷。	
稽核記錄	如果您要透 CloudWatch 過記錄使用稽核記錄，請選取 [稽核記錄]。您也必須將資料庫叢集`neptune_enable_audit_log`參數群組中的參數設定為`enable` (1)，才能啟用稽核記錄。	
Auto minor version upgrade (自動次要版本升級)	如果您要讓 Neptune 資料庫叢集 auto 動接收可用的次要引擎版本升級，以自動接收可用的次要引擎版本升級。 自動次要版本升級選項僅適用於 Amazon Neptune 資料庫叢集的次要引擎版本的升級。不適用於為維護系統穩定性而套用的一般修補程式。	

將 Neptune 資料庫執行個體

您可以使用重新命名 Amazon Neptune 資料庫執行個體AWS Management Console。重新命名資料庫執行個體可以發揮深遠的效果。以下是您在重新命名資料庫執行個體之前應該要先知道的事項清單。

- 重新命名資料庫執行個體時，資料庫執行個體的端點會隨之變更，因為 URL 包含您指派給資料庫執行個體的名稱。您應該將流量從舊的 URL 重新導向到新的 URL。
- 重新命名資料庫執行個體時，資料庫執行個體所用的舊 DNS 名稱會立即刪除，但其會保留在快取內幾分鐘。重新命名的資料庫執行個體的新 DNS 名稱會在大約 10 分鐘後生效。重新命名的資料庫執行個體必須等到新名稱生效後才可使用。
- 重新命名執行個體後，您將無法使用現有的資料庫執行個體名稱。
- 與資料庫執行個體相關聯的所有僅供讀取複本，在重新命名後仍會保持與該執行個體的關聯。例如，假設您有一個做為生產資料庫的資料庫執行個體，且該執行個體有多個關聯的僅供讀取複本。若您重新命名資料庫執行個體，接著在生產環境中將它替換成資料庫快照，則重新命名後的資料庫執行個體仍會擁有與其關聯的僅供讀取複本。
- 如果您重複使用資料庫執行個體名稱，與資料庫執行個體名稱關聯的指標和事件也會保留。例如，如果您提升僅供讀取複本，並將其重新命名為與先前主要執行個體相同的名稱，與主要執行個體相關聯的事件和指標將變成與重新命名後的執行個體相關聯。
- 資料庫執行個體標籤隨資料庫執行個體保留，無論是否重新命名。
- 重新命名的資料庫執行個體將保留其資料庫快照。

使用 Neptune 主控台重新命名資料庫執行個體

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選擇想要重新命名的資料庫執行個體旁邊的選項按鈕。
4. 在 Instance actions (執行個體動作) 功能表中，選擇 Modify (修改)。
5. 在 DB instance identifier (資料庫執行個體識別符) 文字方塊中，輸入新名稱。選擇 Apply immediately (立即套用)，然後選擇 Continue (繼續)。
6. 選擇 Modify DB instance (修改資料庫執行個體) 以完成變更。

在 Amazon Neptune 中重新啟動資料庫

在某些情況下，如果您修改 Amazon Neptune 資料庫執行個體、變更與執行個體相關聯的資料庫參數群組，或變更執行個體使用之參數群組中的靜態資料庫參數，則必須重新啟動執行個體以套用變更。

重新啟動資料庫執行個體，將重新啟動資料庫引擎服務。重新啟動也適用於關聯資料庫參數群組有任何變更正在等待處理的資料庫執行個體。重新啟動資料庫執行個體會暫時中斷執行個體，在此期間，資料庫執行個體狀態設定為重新啟動中。如果 Neptune 執行個體是針對多可用區而設定，重新啟動，重新啟動，重新啟動，重新啟動 重新啟動完成後將建立 Neptune 事件。

如果您的資料庫執行個體為異地同步備份部署，您可在選擇 Reboot (重新啟動) 選項時強制從某一可用區域容錯移轉到另一個區域。強制資料庫執行個體容錯移轉時，Neptune 會自動切換到另一個可用區域中的待命複本。它接著會更新資料庫執行個體的 DNS 記錄，指向待命資料庫執行個體。因此，您必須清除和重新建立資料庫執行個體任何現有的連線。

當您想要進行資料庫執行個體的故障模擬測試，或在發生故障後恢復操作到原始可用區域時，Reboot with failover (使用容錯移轉重新啟動) 是比較有利的。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的高可用性 (多可用區)。重新啟動資料庫叢集時，它會容錯移轉到待命複本。重新啟動 Neptune 複本不會起始容錯移轉。

重新啟動所需的時間為損毀復原程序的函數。若要改善重新啟動時間，建議在重新啟動程序期間盡可能減少資料庫活動，以降低傳輸中交易的轉返活動。

如果資料庫執行個體未處於 [可用] 狀態，則主控台上的 [重新開機] 選項可能會停用。這可能由多種原因導致，例如，備份進行中、客戶正請求修改，或維護時段的動作。

> Note
>
> 之前已發行版本: (p. 884)，每當主要 (寫入器) 執行個體重新啟動時，資料庫叢集中的所有僅供讀取複本都會自動重新啟動。
> 從開始已發行版本: (p. 884)，重新啟動主執行個體不會導致任何複本重新啟動。這表示如果您要變更叢集參數，則必須個別重新啟動每個執行個體，才能取得參數變更 (請參閱參數群組 (p. 527))。

使用 Neptune 主控台重新啟動資料庫執行個體

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選擇想要重新啟動的資料庫執行個體。
4. 選擇 Instance actions (執行個體動作)，然後選擇 Reboot (重新啟動)。
5. 強制容錯移轉到另一個可用區域，請選取 [重新啟動並容錯移轉？ 在重新啟動資料庫執行個體對話方塊中
6. 選擇 Reboot (重新啟動)。若要取消重新啟動，請改為選擇 Cancel (取消)。

刪除 Amazon Neptune 中的數據庫實例

只要執行個體已啟動並停用執行個體上的刪除保護，您就可以隨時刪除 Amazon Neptune 資料庫執行個體。

如果已啟用刪除保護，您就無法刪除資料庫執行個體。

您只能刪除已停用刪除保護的資料庫執行個體。不論您使用主控台、或 API 來刪除資料庫執行個體時AWS CLI，Neptune 會強制刪除保護。

當您使用AWS Management Console建立生產資料庫執行個體時，預設會啟用刪除保護。

如果您使用 AWS CLI 或 API 命令來建立資料庫執行個體，則預設會停用刪除保護。

如果要刪除未啟用刪除保護的資料庫執行個體，請先修改執行個體來將其 `DeletionProtection` 欄位設定成 `false`。

啟用或停用刪除保護並不會導致停機。

刪除資料庫執行個體之前為其建立最終快照

若要刪除資料庫執行個體，則必須指定執行個體的名稱，並決定您是否要為該執行個體擷取最終資料庫快照。如果要刪除的資料庫執行個體狀態為 Creating (正在建立)，便無法擷取最終資料庫快照。如果資料庫執行個體處於故障狀態，且狀態顯示為 failed (失敗)、incompatible-restore (不相容還原) 或 incompatible-network (不相容網路)，則僅有將 `SkipFinalSnapshot` 參數設定為 `true` 時，才能刪除該執行個體。

如果您使用刪除使用的資料庫叢集中的所有 Neptune DB 執行個體時AWS Management Console，會自動刪除整個資料庫叢集。若您使用的是 AWS CLI 或軟體開發套件，則刪除最後一個執行個體之後，您必須手動刪除資料庫叢集。

> **Important**
>
> 如果您刪除整個資料庫叢集，其所有自動備份都會同時刪除，且無法復原。這表示除非您選擇手動建立最終資料庫快照，否則之後無法將資料庫執行個體還原為最終狀態。刪除叢集時，不會刪除執行個體的手動快照。

如果您要刪除的資料庫執行個體具備僅供讀取複本，建議您提升僅供讀取複本或將其刪除。

無論有沒有包含最終資料庫快照，您都可以參考下方範例來刪除資料庫執行個體。

刪除不含最終快照的資料庫執行個體

如果您想要快速刪除資料庫執行個體，可以略過最終資料庫快照的建立步驟。刪除資料庫執行個體時，系統會一併刪除所有自動備份，且無法復原。系統並不會刪除手動快照。

使用 Neptune 主控台來刪除不含最終資料庫快照的資料庫快照的名稱

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 在 Instances (執行個體) 清單中，選擇待刪除資料庫執行個體旁邊的選項按鈕。
4. 選擇 Instance actions (執行個體動作)，然後選擇 Delete (刪除)。
5. 在 Create final snapshot? (是否建立最終快照？) 方塊中，選擇 No (否)。
6. 選擇 刪除 。

刪除包含最終快照的資料庫執行個體

如果您希望稍後可以還原刪除的資料庫執行個體，則可以建立最終資料庫快照。系統會一併刪除所有自動備份，且無法復原。系統並不會刪除手動快照。

使用 Neptune 主控台來刪除含有最終資料庫快照的資料庫快照的名稱

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 在 Instances (執行個體) 清單中，選擇待刪除資料庫執行個體旁邊的選項按鈕。
4. 選擇 Instance actions (執行個體動作)，然後選擇 Delete (刪除)。
5. 在 Create final snapshot? (是否建立最終快照？) 方塊中，選擇 Yes (是)。
6. 在 Final snapshot name (最終快照名稱) 方塊中，輸入最終資料庫快照的名稱。
7. 選擇 刪除 。

您可以檢查執行個體的運作狀態、判斷其執行個體類型、了解您目前安裝的引擎發行版本，並使用執行個體狀態 API (p. 770) 來取得有關執行個體的其他資訊。

Amazon Neptune

Amazon Neptune Serverless 是一種隨需自動調整規模組態，其架構旨在根據需要擴展資料庫叢集以滿足大幅增加的處理需求，然後在需求減少時再次縮減。它有助於自動化監視工作負載和調整 Neptune 資料庫容量的程序。由於容量會根據應用程式需求自動調整，所以只需支付應用程式需求費用

Neptune

Neptune 它適用於高度變數量的工作負載，如果您的資料庫使用量通常在短時間內很大，然後是長時間的輕度活動或完全沒有活動。Neptune Serverless 特別有用：

- 變數工作負載-CPU 活動增加 透過 Neptune Serverless，您的圖形資料庫會自動擴展容量以滿足工作負載需求，並在活動激增結束時縮小規模。您不再需要針對峰值或平均容量進行佈建。您可以指定容量上限來處理峰值工作負載，除非需要，否則不會使用該容量。

  Neptune Serverless 提供的擴展精細程度有助於讓容量緊密符合工作負載需求。Neptune 無伺服器可以根據需要以精細的增量新增或移除容量。當只需要更多容量時，它可以添加一半的 Neptune 容量單位（NCU）(p. 578)。
- 多租用戶應用程式 — 利用 Neptune Serverless，您可以為需要執行的每個應用程式建立個別的資料庫叢集，而不必個別管理這些租用戶叢集。每個租用戶叢集可能有不同的忙碌和閒置期間，具體取決於多個因素，但 Neptune Serverless 可以有效地擴展它們，而無需您的介入。
- 新的應用程式 — 當您部署新的應用程式時，您通常不確定需要多少資料庫容量。使用 Neptune Serverless，您可以設定可自動擴展的資料庫叢集，以符合新應用程式在開發時的容量需求。
- 容量規劃 — 假設您通常透過修改叢集中所有資料庫執行個體類別，藉以調整資料庫容量或驗證工作負載最佳資料庫容量。透過 Neptune，您可以消除此管理開銷。相反地，您可以將現有的資料庫執行個體從佈建到無伺服器或從無伺服器修改為佈建，而不必建立新的資料庫叢集或執行個體。
- 開發與測試 — Neptune 無伺服器也非常適合開發和測試環境。使用 Neptune Serverless，您可以建立具有足夠大容量的資料庫執行個體，以測試要求最嚴苛的應用程式，並且在測試之間系統閒置的所有其他時間使用較低的最小容量。

Neptune 無伺服器僅擴充運算容量。您的儲存磁碟區保持不變，且不受無伺服器擴充的影響。

> Note
>
> 您也可以將 Neptune auto-scaling 與 Neptune 無伺服器搭配使用 (p. 550)，以處理不同種類的工作負載變化

Amazon Neptune

- 並非所有地區都提供 — Neptune 無伺服器僅在下列地區提供：
  - US East (N. Virginia): `us-east-1`
  - US East (俄亥俄): `us-east-2`
  - US West (N. California): `us-west-1`
  - US West (Oregon): `us-west-2`
  - Canada (Central): `ca-central-1`
  - 歐洲 (斯德哥爾摩) `eu-north-1`

  - 歐洲 (愛爾蘭): eu-west-1
  - 歐洲 (倫敦) eu-west-2
  - 歐洲 (法蘭克福) eu-central-1
  - 亞太區域 (東京) ap-northeast-1
  - 亞太區域 (新加坡)：ap-southeast-1
  - 亞太區域 (雪梨)：ap-southeast-2
- 早期引擎版本不提供 — Neptune 無伺服器僅適用於引擎版本 1.2.0.1 或更新版本。
- 與 Neptune 查詢快取不相容 — 查閱快取 (p. 107)不適用於無伺服器資料庫執行個體。
- 無伺服器執行個體的最大記憶體為 256 GB — 設定MaxCapacity為 128 個 NCU (支援的最高設定) 可讓 Neptune 無伺服器執行個體擴充至 256 GB 記憶體，相當於R6g.8XL佈建執行個體類型的記憶體。

Neptune 無伺服器資料庫叢集中的容量擴充

設定 Neptune 無伺服器資料庫叢集類似於設定一般佈建的叢集，其他配置用於擴展的最小和最大單位，並將db.serverless執行個體類型設定為。擴展配置在 Neptune 容量單位 (NCU) 中定義，每個單位都包含 2 GiB (GB) 記憶體 (RAM)，以及相關的虛擬處理器容量 (vCPU) 和網路。它被設置為ServerlessV2ScalingConfiguration對象的一部分，以 JSON 表示如下：

```
"ServerlessV2ScalingConfiguration": {
  "MinCapacity": (minimum NCUs, a floating-point number such as   1.0),
  "MaxCapacity": (maximum NCUs, a floating-point number such as 128.0)
}
```

在任何時候，每個 Neptune 寫入器或讀取器執行個體都有一個由浮點數測量的容量，代表該執行個體目前正在使用的 NCU 數目。您可以使用執行個體層級的指CloudWatchServerlessDatabaseCapacity (p. 584)標，瞭解指定資料庫執行個體目前正在使用多少 NCU，以及 NCUUtilise 指標，找出執行個體使用 (p. 584)的最大容量百分比。這兩個測量結果也可在資料庫叢集層級使用，以顯示整體資料庫叢集的平均資源使用率。

建立 Neptune 無伺服器資料庫叢集時，您可以為所有無伺服器執行個體設定 N eptune 容量單位 (NCU) 的下限和最大數目。

您指定的最小 NCU 值會設定資料庫叢集中無伺服器執行個體可縮小的最小大小，同樣地，NCU 值上限會建立無伺服器執行個體可以成長的最大大小。您可以設定的最高 NCU 值上限為 128.0 NCU，而最低的最小值為 1.0 NCU。

Neptune 透過監視其對資源 (例如 CPU、記憶體和網路) 的使用率，來持續追蹤每個 Neptune Serverless 上的負載。負載是由應用程式的資料庫作業、伺服器的背景處理，以及其他系統管理工作所產生。

當無伺服器執行個體的負載達到目前容量上限時，或是 Neptune 偵測到任何其他效能問題時，執行個體就會自動擴展。當執行個體的負載下降時，容量會縮減至設定的最小容量單位，CPU 容量會在記憶體之前釋放。這種架構允許以受控的降壓方式釋放資源，並有效地處理需求波動。

您可以讓讀取器執行個體與寫入器執行個體一起擴展，或透過設定其促銷層來獨立擴展。升級層 0 和 1 中的讀取器執行個體與寫入器相同的時間擴展，這使它們保持適當的容量，以便在發生容錯移轉時從寫入器處接管工作負載。在促銷層 2 到 15 的讀者獨立於編寫器實例，以及彼此的規模。

如果您已將 Neptune 資料庫叢集建立為異地同步備份叢集以確保高可用性，Neptune 無伺服器會隨著資料庫負載而上下擴展所有 AZ 中的執行個體。您可以將次要 AZ 中讀取器執行個體的升級層設定為 0 或 1，以便隨著主要可用區中寫入器執行個體的容量進行擴展和縮減，以便隨時準備好接管目前的工作負載。

> **Note**
>
> Neptune 資料庫叢集的儲存，由分散在三個可用區域中的所有資料的六個副本組成，不論您是否將叢集建立為異地同步備份叢集。儲存複寫由儲存子系統處理，不受 Neptune 無伺服器影響。

選擇 Neptune 無伺服器資料庫叢集的最小容量值

您可以為最小容量設定的最小值為 1.0 NCU。

請確定不要將最小值設定為低於應用程式為了有效運作所需的值。將其設定得太低可能會在特定記憶體密集型工作負載中導致更高的逾時率。

將最小值設置為盡可能低可以節省成本，因為當需求不足時，您的叢集將使用最少的資源。不過，如果您的工作負載往往會大幅波動 (從非常低到非常高)，您可能會想要設定較高的最小值，因為較高的最小值可讓 Neptune 無伺服器執行個體更快地擴充。

原因是 Neptune 會根據目前的容量選擇縮放增量。如果電流容量很低，Neptune 一開始會緩慢地擴展。如果最小值較高，Neptune 會從較大的縮放增量開始，因此可以更快地擴展以因應工作負載的大幅突然增加。

選擇 Neptune 無伺服器資料庫叢集的最大容量值

您可以為最大容量設定的最大值是 128.0 NCU，您可以為最大容量設定的最小值為 2.5 NCU。無論您設定的最大容量值，必須至少與您設定的最短容量值相同。

一般而言，請將最大值設定為足以處理應用程式可能會遇到的尖峰負載。將其設定得太低可能會在特定記憶體密集型工作負載中導致更高的逾時率。

將最大值設定為盡可能高，即使是您的應用程式也可能能夠處理最未預期的工作負載。缺點是您失去了一些預測和控制資源成本的能力。非預期的需求激增最終可能會超出預算的成本。

精心定位的最大值的好處在於，它可以讓您滿足尖峰需求，同時也限制 Neptune 運算成本。

> Note
>
> 變更 Neptune Serverless 叢集的容量範圍會造成某些組態參數的預設值產生變更。Neptune 可立即套用某些新的預設值，但某些動態參數變更僅於重新啟動後生效。pending-reboot狀態表示需要重新啟動才能套用某些參數的變更。

使用您現有的組態來預估無伺服器需求

若您通常修改佈建的資料庫執行個體類別，以滿足非常高或低的工作負載，則可使用該體驗粗略估計等效的 Neptune less 容量範圍。

估計最佳的最小容量設定

您可以套用您對現有 Neptune DB 叢集的了解，以估計最適合的無伺服器最小容量設定。

例如，若您佈建的工作負載對於或等小型資料庫執行個體類別的記憶體要求過高T4g，請選擇提供與T3或資料R6g庫執行個R5體類別相當的記憶體最短 TTU 設定。

或者，假設您在叢集的工作負載較低時，使用db.r6g.xlarge資料庫執行個體類別。該資料庫執行個體類別具有 32 GiB 的記憶體，因此您可將最短 TTU 設定指定為 16，來建立可縮小至大約相同的容量 (每個 NCU 對應於 2 GiB 的記憶體)。若您的db.r6g.xlarge執行個體有時並未充分利用，則您可指定一個較低的值。

若您的資料庫執行個體可將指定數量的資料容納在記憶體或緩衝區快取中，請考慮指定一個最短 TTU 設定，以提供足夠的記憶體。否則，當無伺服器執行個體縮小時，資料可能會從緩衝區快取中移出，而且在執行個體擴充備份時，必須將資料讀回緩衝區快取。若要將資料帶回緩衝區快取的 I/O 數量相當多，則選擇較高的最短 TTU 值。

如果您發現無伺服器執行個體大部分時間都以特定容量執行，則將最小容量設定為低於該容量的效果很好。當目前容量並未明顯低於所需容量時，Neptune Serverless 可有效地估計擴充規模的數量和速度。

在混合組態 (p. 580)中，使用佈建的寫入器和 Neptune Serverless 讀取器不會與寫入器一起擴展。因為它們可獨立擴充，因此為它們設定較低的最短容量可能會導致過多的複寫延遲。當有高度寫入密集型工作負載時，他們可能沒有足夠的容量來跟上寫入器所做的變更。在此情況下，請設定與寫入器容量相當的最小容量。尤其是，若您在升級層 2—15 中的讀取器中發現複本延遲，請增加叢集的最短容量設定。

估計最佳的最大容量設定

您也可以套用您對現有 Neptune DB 叢集的了解，以估計最適合的無伺服器容量上限設定。

例如，假設您在叢集的工作負載較高時，使用`db.r6g.4xlarge`資料庫執行個體類別。該資料庫執行個體類別有 128 GiB 的記憶體，因此您可以指定 64 個 NCU 上限設定，以設定相等的 Neptune CU 無伺服器執行個體 (每個 NCU 對應約 2 GiB 的記憶體)。如果您的執行個體並未能處理工作負載，則您可指定一個較高的值，以進一步擴展資料庫`db.r6g.4xlarge`執行個體。

如果工作負載出現未預期的尖峰情況很少見，將最大容量設定為足以維持應用程式效能，即使在這些尖峰期間也是有意義的。另一方面，您可能想要設定較低的最大容量，以便在異常尖峰期間減少輸送量，但這樣可以讓 Neptune 毫無問題地處理預期的工作負載，並限制成本。

Neptune 無伺服器資料庫叢集和執行個體的其他組

除了為 Neptune 無伺服器資料庫叢集設定最小和最大容量 (p. 578)之外，還有其他幾個組態選項需要考慮。

結合資料庫叢集中的無伺服器和佈建執行個體

資料庫叢集不一定只是無伺服器 — 您可以建立無伺服器執行個體和佈建執行個體的組合 (混合組態)。

例如，假設您需要比無伺服器執行個體的可用容量。在這種情況下，您可以使用非常大的已佈建寫入器來設定叢集，並且仍然可以為讀取器來設定叢集。

或者，假設叢集上的寫入工作負載有變化，但讀取工作負載穩定。在這種情況下，您可以使用一個 Serverless 寫入器和一個或多個已佈建讀取器來設定叢集。

如使用 Amazon Neptune Serverless (p. 581)需有關如何有關如何建立混合配置資料庫叢集的資訊，請參閱。

設定 Neptune 無伺服器執行個體的促銷層級

對於包含多個 Serverless 或混合已佈建和 Serverless 和 Serverless 和 Serverless 執行個體或混合已佈建和 Serverless 執行個體或混合已佈建和 Serverless 和 Serverless 和 Serverless 此設定控制的更多行為，而非已佈建的資料庫執行個體。

在中AWS Management Console，您可以使用 [建立資料庫]、[修改執行個體] 和 [新增讀取器] 頁面的 [其他組態] 下的 [容錯移轉優先順序] 指定 您可以在 [資料庫] 頁面的選用優先順序層資料行中看到現有執行個體的這個屬性。您也可以在資料庫叢集或執行個體的詳細資訊頁面上查看此屬性。

對於已佈建的執行個體，選擇第 0—15 層僅決定 Neptune less 選擇將讀取器執行個體提升為寫入器的順序。對於 Neptune Serverless 讀取器執行個體，層數也會決定執行個體是否向上擴展以符合寫入器執行個體的容量，還是僅根據其自身的工作負載獨立擴展。

第 0 層或第 1 層的 Neptune 無伺服器讀取器執行個體保持至少與寫入器執行個體相同的最低容量，以便在容錯移轉時準備接管寫入器。如果寫入器是已佈建的執行個體，Neptune less 會估計對等的 Serverless 容量，並使用該估計值做為 Serverless 讀取器執行個體的最小容量。

第 2—15 層中的 Neptune less Serverless 2—15 層對其最小容量沒有相同的限制，並且可以獨立於寫入器的規模。當它們處於閒置狀態時，它們會縮減規模到叢集容量範圍 (p. 578)中指定的最小 NCU 值。但是，這可能會導致問題

保持讀取器容量與寫入器容量一致

要記住的一件重要事情是，您要確保您的讀取器實例可以跟上您的寫入器實例，以防止過多的複寫延遲。在兩種情況下，無伺服器讀取器執行個體不會自動與寫入器執行個體同步調整，這是特別關注的問題：

- 當您的寫入器被佈建時，您的讀者是無伺服器的。
- 當您的寫入器是無伺服器，且您的無伺服器讀者處於促銷層 2-15 時。

在這兩種情況下，請將最小無伺服器容量設定為符合預期寫入器容量的容量，以確保讀取器作業不會逾時且可能導致重新啟動。如果是佈建的寫入器執行個體，請設定最小容量以符合已佈建執行個體的容量。在無伺服器寫入器的情況下，最佳設定可能較難預測。

由於執行個體容量範圍是在叢集層級設定，因此所有無伺服器執行個體都由相同的最小和最大容量設定控制。第 0 層和第 1 層的 Reader 執行個體會與寫入器執行個體同步擴展，但促銷層 2-15 中的執行個體會根據其工作負載獨立擴展，而不同於寫入器執行個體。如果您將最小容量設定得太低，階層 2 到 15 中的閒置執行個體可能會縮減過低，以至於足以快速擴充，以便處理寫入器活動中突然爆發的情況。

避免將逾時值設定得太高

如果您在無伺服器執行個體上將查詢逾時值設定得太高，可能會產生非預期費用。

如果沒有合理的逾時設定，您可能會不小心發出需要功能強大且昂貴執行個體類型的查詢，而且會持續執行很長時間，產生您從未預期的成本。您可以通過使用查詢超時值來避免這種情況，該值可容納大多數查詢，並且只會導致意外長時間運行的查詢超時。

對於使用參數設定的一般查詢逾時值，以及使用查詢提示設定的每個查詢逾時值，都是如此。

最佳化 Neptune 無伺服器組態

如果 Neptune 無伺服器資料庫叢集未針對其執行的工作負載進行調整，您可能會注意到它並未以最佳方式執行。您可以調整最小和/或最大容量設定，以便在不遇到記憶體問題的情況下進行擴充。

- 增加叢集的最小容量設定。這可以修正閒置執行個體擴充回其記憶體比應用程式少且已啟用功能所需的容量的情況。
- 增加叢集的最大容量設定。如此可修正繁忙的資料庫無法擴充規模至具足夠記憶體的容量，以滿足工作負載和任何已啟用記憶體的功能。
- 變更有問題執行個體的工作負載。例如，您可將讀取器執行個體新增至叢集，以將讀取負載分散至更多執行個體。
- 調整應用程式的查詢，使其使用更少的資源。
- 嘗試使用佈建的執行個體大於 Neptune 無伺服器內可用 NCU 上限的執行個體，以查看它是否更適合工作負載的記憶體和 CPU 需求。

使用 Amazon Neptune Serverless

您可以將新的 Neptune DB 叢集建立為無伺服器叢集，或在某些情況下，您可以將現有的資料庫叢集轉換為無伺服器叢集。您也可以將無伺服器資料庫叢集中的資料庫執行個體與無伺服器執行個體之間進行轉換。您只能在支援 Neptune 無伺服器的其中一個AWS 區域位置使用，但有一些其他限制 (請參閱Amazon Neptune (p. 577))。

您也可以使用 Neptune AWS CloudFormation 堆疊 (p. 59)來建立 Neptune 無伺服器資料庫叢集。

建立使用無伺服器的新資料庫叢集

若要建立使用無伺服器的 Neptune DB 叢集，您可以使用 (p. 534)與建立已佈建叢集相同的方式來執行此操作。AWS Management Console不同之處在於，在資料庫執行個體大小下，您需要將資料庫執行個體類別設定為無伺服器。當您這麼做時，您需要設定叢集的無伺服器容量範圍 (p. 578)。

您也可以使用像這樣的命令來創建無服務器數據庫集群（在 Windows 上，用 '^' 替換 '\'）：AWS CLI

```
aws neptune create-db-cluster \
  --region (an AWS ## region that supports serverless) \
  --db-cluster-identifier (ID for the new serverless DB cluster) \
  --engine neptune \
  --engine-version (optional: 1.2.0.1 or above) \
  --serverless-v2-scaling-configuration "MinCapacity=1.0, MaxCapacity=128.0"
```

您也可以像這樣指定`serverless-v2-scaling-configuration`參數：

```
  --serverless-v2-scaling-configuration '{"MinCapacity":1.0, "MaxCapacity":128.0}'
```

然後，您可以執行`ServerlessV2ScalingConfiguration`屬性的`describe-db-clusters`命令，該命令應傳回您指定的容量範圍設定：

```
"ServerlessV2ScalingConfiguration": {
    "MinCapacity": (the specified minimum number of NCUs),
    "MaxCapacity": (the specified maximum number of NCUs)
}
```

將現有的資料庫叢集或執行個體轉換為無伺服器

如果您有使用引擎版本 1.2.0.1 或更新版本的 Neptune DB 叢集，您可以將其轉換為無伺服器。此過程確實會導致一些停機時間。

第一步是將容量範圍新增至現有的叢集。您可以使用AWS Management Console，或者使用這樣的AWS CLI命令來執行此操作（在 Windows 上，將 '\' 替換為 '^'）：

```
aws neptune modify-db-cluster \
  --db-cluster-identifier (your DB cluster ID) \
  --serverless-v2-scaling-configuration \
      MinCapacity=(minimum number of NCUs, such as  2.0), \
      MaxCapacity=(maximum number of NCUs, such as 24.0)
```

下一步是建立新的無伺服器資料庫執行個體，以取代叢集中現有的主要執行個體 (寫入器)。同樣地，您可以使用AWS Management Console或執行此操作以及所有後續步驟AWS CLI。在任一情況下，請將資料庫執行個體類別指定為無 Serverless。該AWS CLI命令看起來像這樣（在 Windows 上，用 '^' 替換 '\'）：

```
aws neptune create-db-instance \
  --db-instance-identifier (an instance ID for the new writer instance) \
  --db-cluster-identifier (ID of the DB cluster) \
  --db-instance-class db.serverless
  --engine neptune
```

當新的寫入器執行個體可用時，請執行容錯移轉，使其成為叢集的寫入器執行個體：

```
aws neptune failover-db-cluster \
```

```
  --db-cluster-identifier (ID of the DB cluster) \
  --target-db-instance-identifier (instance ID of the new serverless instance)
```

接下來，刪除舊的寫入器實例：

```
aws neptune delete-db-instance \
  --db-instance-identifier (instance ID of the old writer instance) \
  --skip-final-snapshot
```

最後，執行相同的動作來建立新的無伺服器執行個體，以取代您想要轉換為無伺服器執行個體的每個現有已佈建讀取器執行個體，並刪除現有的佈建執行個體 (讀取器執行個體不需要容錯移轉)。

將個別佈建資料庫執行個體變更為無伺服器

您也可以將單一已佈建的資料庫執行個體轉換為使用 Neptune Serverless。您可以通過使用如下`modify-db-instance`命令AWS CLI來執行此操作：

```
aws neptune modify-db-instance \
    --db-instance-identifier (your DB instance ID) \
    --serverless-v2-scaling-configuration \
        MinCapacity=(minimum number of NCUs, such as  3.5), \
        MaxCapacity=(maximum number of NCUs, such as 24.0)
```

請注意，您為執行個體設定的容量範圍會套用至叢集中的所有無伺服器執行個體 (如果有多個執行個體)。

您也可以使用 Neptune 主控台將單一佈建的執行個體切換至無伺服器：

使用主控台將佈建的資料庫執行個體變更為無伺服器

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 導覽至想要變更的已佈建資料庫執行個體。
3. 選取 Modify (修改) 按鈕。
4. 在 [引擎類型] 下選擇 [無伺服器]。
5. 您現在可以選擇指定容量範圍。請注意，如果您在此處設定新的容量範圍，則會將其套用至資料庫叢集中的每個無伺服器執行個體。
6. 進行任一您想要的變更，然後選取頁面底部的繼續。
7. 檢閱變更，並選擇是否應在下一個排定的維護時段或立即進行變更。
8. 選取修改資料庫例項。

修改現有無伺服器資料庫叢集的容量範圍

您可以使用AWS CLI類似的方式變更 Neptune 無伺服器資料庫叢集的容量範圍 (在 Windows 上，將 '\' 取代為 '^')：

```
aws neptune modify-db-cluster \
  --region (an AWS region that supports serverless) \
  --db-cluster-identifier (ID of the serverless DB cluster) \
  --apply-immediately \
  --serverless-v2-scaling-configuration MinCapacity=4.0, MaxCapacity=32
```

變更容量範圍會造成某些組態參數的預設值產生變更。Neptune 可以立即套用某些新的預設值，但某些動態參數變更僅於重新啟動後生效。狀態`pending-reboot`表示需要重新啟動才能套用某些參數變更。

將無伺服器資料庫執行個體變更為佈建

若要將 Neptune 無伺服器執行個體轉換為已佈建的執行個體，您只需將其執行個體類別變更為其中一個已佈建的執行個體類別即可。請參閱 修改 Neptune 資料庫執行個體 (並立即套用) (p. 571)。

使用 Amazon 監控 Serverless CloudWatch

您可以用CloudWatch來監視資料庫叢集中 Neptune 無伺服器執行個體的容量和使用率。有兩個CloudWatch指標可讓您在叢集層級和執行個體層級追蹤目前的無伺服器容量：

- **`ServerlessDatabaseCapacity`**— 作為執行個體層級指標，會在 NCU 中`ServerlessDatabaseCapacity`報告目前的執行個體容量。作為叢集層級指標，報告叢集中所有資料庫執行個體的所有`ServerlessDatabaseCapacity`值的平均值。
- **`NCUUtilization`**— 此指標報告可能使用的容量百分比。計算方式為目前 `ServerlessDatabaseCapacity` (在執行個體層級或叢集層級) 除以資料庫叢集的最大容量設定。

  如果此量度在叢集層級接近 100%，表示叢集已盡可能高地擴展，請考慮增加最大容量設定。

  如果寫入器執行個體接近 100%，而寫入器執行個體不接近最大容量，請考慮新增更多讀取器執行個體來分配讀取工作負載。

請注意，`CPUUtilization`和`FreeableMemory`指標對無伺服器執行個體的含義略有不同，與佈建執行個體的含義略有不 在無伺服器內容中，`CPUUtilization`是一個百分比，計算方法為目前使用的 CPU 量除以最大容量可用的 CPU 量。同樣地，`FreeableMemory`報告執行個體達到最大容量時可用的可用記憶體數量。

下列範例顯示如何使用 Linux AWS CLI 上的 Linux 擷取指定資料庫執行個體的最小、最大和平均容量值值，在一小時內每 10 分鐘測量一次。Linux `date` 命令指定相對於目前日期和時間的開始和結束時間。`--query`參`sort_by`數中的結果依時間順序對結果進行排序`Timestamp`：

```
aws cloudwatch get-metric-statistics \
  --metric-name "ServerlessDatabaseCapacity" \
  --start-time "$(date -d '1 hour ago')" \
  --end-time "$(date -d 'now')" \
  --period 600 \
  --namespace "AWS/Neptune"
  --statistics Minimum Maximum Average \
  --dimensions Name=DBInstanceIdentifier,Value=(instance ID) \
  --query 'sort_by(Datapoints[*].{min:Minimum,max:Maximum,avg:Average,ts:Timestamp},&ts)' \
  --output table
```

使用 Neptune 串流即時擷取圖形變更

Neptune Streams 會以完全託管的方式記錄圖形發生的所有變更，按照其進行的順序進行。啟用串流後，Neptune 會處理可用性、備份、安全性和到期。

> **Note**
>
> 此功能從開始於實驗室模式 (p. 127)中提供版本 1.0.0.0.463.0 (p. 985)，並且從 Neptune 引擎版本 1.0.2.2.R2 (p. 967) 開始可用於生產環境。

以下是您可能想要在圖形發生變更時擷取其變更的一些使用案例：

- 您可能想要應用程式在進行特定變更時自動通知人員。
- 您可能想要在另一個資料存放區中維護圖形資料的最新版本，例如亞馬遜 OpenSearch 服務 ElastiCache、亞馬遜或亞馬遜 Simple Storage Service (Amazon S3)。

Neptune 使用與圖形資料相同的原生儲存區做為變更記錄資料流。它會同步寫入變更日誌項目與進行這些變更的交易。您可以使用 HTTP REST API，從日誌串流擷取這些變更記錄。(如需相關資訊，請參閱呼叫串流 API (p. 587)。)

下圖顯示如何從 Neptune 串流擷取變更記錄資料。

Neptune

- 交易一完成時，就可以立即從寫入器和讀取器二者讀取交易所做的變更 (除了讀取器中的任何正常複寫延遲外)。
- 變更記錄會嚴格地依其發生的順序顯示 (這包括交易內所做的變更)。
- 變更串流不包含重複項目。每個變更只會記錄一次。
- 變更串流已完成。未遺失或省略任何變更。
- 變更串流包含判斷資料庫本身在任何時間點是否狀態完整所需的所有資訊，前提是起始狀態已知。
- 串流可以隨時開啟或關閉。

Neptune 串流作業屬性

- 變更日誌串流是全受管的。
- 變更日誌資料會以同步方式寫入，做為進行變更之相同交易的一部分。
- 啟用 Neptune 串流時，會產生與變更記錄資料相關的 I/O 和儲存體費用。
- 依預設，變更記錄會在建立一週後自動清除。從引擎版本 1.2.0.0 (p. 884) 開始，此保留期間可以使用海王星流 _ 流程 _ 日 (p. 532)資料庫叢集參數變更為 1 到 90 之間的任意天數。
- 串流上的讀取效能會隨執行個體擴展。
- 您可以使用僅供讀取複本實現高可用性和讀取輸送量。您可以同時建立和使用的串流讀取器數量沒有限制。
- 變更日誌資料可跨多個可用區域進行複寫，使其具有高耐用性。
- 日誌資料與您的圖形資料本身一樣安全。它可以在靜態和傳輸中加密。可以使用 IAM、亞馬遜 VPC 和 AWS Key Management Service (AWS KMS) 來控制存取。與圖形資料一樣，可以備份資料，稍後使用 point-in-time 還原 (PITR) 進行還原。

- 隨著每個交易同步寫入串流資料，會導致整體寫入效能略微降低。
- 流數據不分片，因為 Neptune 是單一分片的設計。
- 記錄資料流`GetRecords` API 使用與所有其他 Neptune 圖形作業相同的資源。這表示用戶端需要在串流請求和其他資料庫請求之間進行負載平衡。
- 串流停用時，所有日誌資料都會立即變成無法存取。這表示您必須先讀取您感興趣的所有日誌資料，然後才能停用記錄。
- 目前沒有與 AWS Lambda 的原生整合。記錄串流不會產生 Lambda 函式的事件。

主題

- 使用 Neptune 串流 (p. 586)
- Neptune 串流中的序列化格式 (p. 590)
- Neptune 範例集 (p. 592)
- 用AWS CloudFormation於使用串流使用者應用程式設定 Neptune 到海王星的複製 (p. 596)
- 使用 Neptune 串流跨區域複寫進行災難復原 (p. 600)

使用 Neptune 串流

使用 Neptune Streams 功能，您可以產生一系列完整的變更記錄項目，以記錄對圖形資料所做的每次變更。如需此功能的概觀，請參閱 使用 Neptune 串流即時擷取圖形變更 (p. 585)。

主題

- 啟用 Neptune 串流 (p. 586)
- 停用 Neptune 串流 (p. 586)
- 呼叫 Neptune 串流其他 API (p. 587)
- Neptune 串流 API 回應格式 (p. 588)
- Neptune 流 API 異常 (p. 589)

啟用 Neptune 串流

您可以隨時透過設定neptune_streams資料庫叢集參數 (p. 531)來啟用或停用 Neptune 串流。將參數設為 `1` 可啟用串流，設為 `0` 則可停用串流。

> **Note**
>
> 變更`neptune_streams`資料庫叢集參數之後，您必須重新啟動叢集，變更才會生效。

您可以設定海王星流 _expiry_days (p. 532) 資料庫叢集參數，以控制資料流記錄在刪除之前保留在伺服器上的天數 (從 1 到 90)。預設值為 7。

Neptune Streams 最初是作為一項實驗性功能引入，您可以使用資料庫叢集`neptune_lab_mode`參數在實驗室模式中啟用或停用該功能 (請參閱海王星實驗室 (p. 127))。使用實驗室模式啟用串流現在已遭取代，且未來將停用。

停用 Neptune 串流

您可以在任何時候關閉 Neptune 串流執行。

若要關閉串流，請更新資料庫叢集參數群組，將 neptune_streams 參數的值設為 0。

Important

一旦串流關閉，您再也無法存取變更日誌資料。在關閉串流之前，請務必閱讀您感興趣的內容。

呼叫 Neptune 串流其他 API

您可以使用 REST API 來存取 Neptune 串流，該 API 會將 HTTP GET 要求傳送至下列其中一個本機端點：

- 若為 SPARQL 圖形資料庫：`https://Neptune-DNS:8182/sparql/stream`。
- 對於一個小精靈或開放密碼圖 DB：`https://Neptune-DNS:8182/propertygraph/stream`或`https://Neptune-DNS:8182/pg/stream`。

Note

從引擎版本 1.1.0.0 (p. 915) 開始，Gemlin 流端點（`https://Neptune-DNS:8182/gremlin/stream`）已被棄用，以及其關聯的輸出格式（`GREMLIN_JSON`）。它仍然支持向後兼容性，但可能會在 future 的版本中刪除。

只允許 HTTP GET 操作。

Neptune 支援回應gzip壓縮，前提是 HTTP 要求包含指定gzip為可接受的壓縮格式 (也就是"Accept-Encoding: gzip") 的Accept-Encoding標頭。

參數

- limit-長，可選。範圍：1 至 10 萬。預設值：10

  指定要傳回的記錄數上限。回應也有無法修改的 10 MB 大小限制，而且會優先於 limit 參數中指定的記錄數量。如果達到 10 MB 限制，回應會包含違反閾值的記錄。
- iteratorType-字符串，可選。

  此參數可採用下列其中一個值：
  - AT_SEQUENCE_NUMBER(預設) — 指出讀取應從commitNum和opNum參數共同指定的事件序列號開始。
  - AFTER_SEQUENCE_NUMBER— 指示讀取應該在commitNum和opNum參數共同指定的事件序列號之後立即開始。
  - TRIM_HORIZON— 指示讀取應該從系統中的最後一個未修剪記錄開始，這是變更記錄資料流中最舊的未過期 (尚未刪除) 記錄。當您沒有特定的開始事件序號時，此模式在應用程式啟動期間很有用。
  - LATEST— 指出讀取應該從系統中的最新記錄開始，也就是變更記錄資料流中的最新未過期 (尚未刪除) 記錄。當需要從串流的目前頂端讀取記錄，以免處理較舊的記錄 (例如嚴重損壞修復期間或升級零停機時間) 時，這會很有用。請注意，在此模式下，最多只會傳回一筆記錄。
- commitNum-長，當 iteratorType 為AT_SEQUENCE_NUMBER或時所需AFTER_SEQUENCE_NUMBER。

  要從 change-log 串流讀取之開始記錄的遞交編號。

  如果是TRIM_HORIZON或，則會iteratorType忽略此參數LATEST。
- opNum-長，可選（默認為1）。

  所指定遞交內的操作序號，即在變更日誌串流資料中開始讀取之處。

變更 SPARQL 圖形資料的操作通常只會為每個操作產生單一變更記錄。不過，變更 Gremlin 圖形資料的操作可能會為每個操作產生多個變更記錄，如以下範例所示：

- INSERT— Girmlin 頂點可以有多個標籤，而一個 Girmlin 元素可以具有多個屬性。插入元素時，會為每個標籤和屬性產生個別的變更記錄。
- UPDATE— 變更 Gemlin 元素屬性時，會產生兩個變更記錄：第一個用於移除先前值，第二個用於插入新值。
- DELETE— 會針對每個刪除的元素內容產生個別的變更記錄。例如，刪除具有屬性的 Gremlin 邊緣時，每個屬性都會產生一個變更記錄，之後會產生一個用於刪除邊緣標籤的變更記錄。

  刪除 Gremlin 頂點時，會先刪除所有傳入和傳出邊緣屬性，接著依序刪除邊緣標籤、頂點屬性，最後再刪除頂點標籤。其中每個刪除都會產生一筆變更記錄。

Neptune 串流 API 回應格式

Neptune 串流 REST API 要求的回應具有下列欄位：

- lastEventId— 串流回應中最後變更的序列識別碼。事件 ID 由兩個欄位組成：commitNum 識別已變更圖形的交易，以及 opNum 識別該交易內的特定操作。如以下範例所示。

```
  "eventId": {
    "commitNum": 12,
    "opNum": 1
  }
```

- lastTrxTimestamp— 要求交易提交的時間，從 Unix 紀元開始，以毫秒為單位。
- format— 要傳回之變更記錄的序列化格式。可能的值是PG_JSON針對小鬼或開放密碼變更記錄，以及NQUADS SPARQL 變更記錄。
- records— 包含在回應中的序列化變更記錄串流記錄陣列。records陣列中的每個記錄都包含下列欄位：
  - commitTimestamp— 要求交易提交的時間，從 Unix 紀元開始，以毫秒為單位。
  - eventId— 串流變更記錄的序列識別碼。
  - data— 序列化的小鬼、SPARQL 或 OpenCypher 變更記錄。每個記錄的序列化格式將在下一節中進行更詳細的說明Neptune 串流中的序列化格式 (p. 590)。
  - op— 建立變更的作業。
  - isLastOp-僅當此操作是其交易中的最後一個操作時才存在。如果存在，則會將其設定為true。有助於確保消耗整個交易。
- totalRecords— 回應中的記錄總數。

例如，下列回應會針對包含多項作業的交易傳回 Gemlin 變更資料：

```
{
  "lastEventId": {
    "commitNum": 12,
    "opNum": 1
  },
  "lastTrxTimestamp": 1560011610678,
  "format": "PG_JSON",
  "records": [
    {
      "commitTimestamp": 1560011610678,
      "eventId": {
        "commitNum": 1,
        "opNum": 1
      },
      "data": {
        "id": "d2b59bf8-0d0f-218b-f68b-2aa7b0b1904a",
```

```
        "type": "vl",
        "key": "label",
        "value": {
          "value": "vertex",
          "dataType": "String"
        }
      },
      "op": "ADD"
    }
  ],
  "totalRecords": 1
}
```

下列回應會傳回交易中最後一個作業的 SPARQL 變更資料 (交易編號 97EventId(97, 1) 中所識別的作業)。

```
{
  "lastEventId": {
    "commitNum": 97,
    "opNum": 1
  },
  "lastTrxTimestamp": 1561489355102,
  "format": "NQUADS",
  "records": [
    {
      "commitTimestamp": 1561489355102,
      "eventId": {
        "commitNum": 97,
        "opNum": 1
      },
      "data": {
        "stmt": "<https://test.com/s> <https://test.com/p> <https://test.com/o> .\n"
      },
      "op": "ADD",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

Neptune 流 API 異常

下表描描描描描描描描 Neptune 描描描描描述

錯誤代碼	HTTP 代碼	OK to Retry? (確定要重試嗎？)	訊息
InvalidParameterException	400	否	提供了無效的或 out-of-range 值作為輸入參數。
ExpiredStreamException	400	否	所有請求的記錄都超過允許的最大存留期且已過期。
ThrottlingException	500	是	請求速度超出傳輸量上限。
StreamRecordsNotFoundException	404	否	找不到請求的資源。可能無法正確指定串流。

錯誤代碼	HTTP 代碼	OK to Retry? (確定要重試嗎？)	訊息
MemoryLimitExceededException	500	是	由於缺乏記憶體，請求處理未成功，但可以在伺服器較不忙碌時重試。

Neptune 串流中的序列化格式

Amazon Neptune 會根據圖形是使用 Gremlin 還是 SPARQL 建立，使用兩種不同格式來序列化圖形變更資料以記錄串流。

這兩種格式共用通用的記錄序列化格式 (如中所述)Neptune 串流 API 回應格式 (p. 588)，其中包含下列欄位：

- commitTimestamp— 要求交易提交的時間，從 Unix 紀元開始，以毫秒為單位。
- eventId— 串流變更記錄的序列識別碼。
- data— 序列化的小鬼、SPARQL 或 OpenCypher 變更記錄。將詳細說明每筆記錄的序列化格式。
- op— 建立變更的作業。

主題

- 更改序列化格式 (p. 590)
- SPARQL NQUADS 變更序列化格式 (p. 592)

更改序列化格式

> **Note**
>
> 從引擎版本 1.1.0.0 (p. 915) 開始，由 Girmlin 流端點 (GREMLIN_JSON) 輸出的 Gremlin 流輸出格式 (https://*Neptune-DNS*:8182/gremlin/stream) 被棄用。它由 PG_JSON 取代，目前與GREMLIN_JSON.

小鬼或 OpenCypher 變更記錄，包含在記錄資料流回應的data欄位中，具有下列欄位：

- id— 字串，必要。

  小精靈或開放密碼元素的 ID。
- type— 字串，必要。

  這個小精靈或開放密碼元素的類型。必須是下列其中之一：
  - vl— 小鬼的頂點標籤; 節點標籤為開放密碼.
  - vp— 小精靈的頂點屬性；開放密碼的節點屬性。
  - e— 小精靈的邊緣和邊緣標籤; OpenCypher 的關係和關係類型。
  - ep— 小精靈的邊緣屬性；開放密碼的關係屬性。
- key— 字串，必要。

  屬性名稱 對於元素標籤，這是「標籤」。
- value—value 物件，必要項。

  這是 JSON 物件，其中包含值本身的 value 欄位，以及該值之 JSON 資料類型的 datatype 欄位。

```
  "value": {
    "value": "the new value",
    "dataType": "the JSON datatype of the new value"
  }
```

- from-字符串，可選。

  如果這是一個邊緣（type = "e」），則表示相應的來自頂點或源節點的 ID。
- to-字符串，可選。

  如果這是一個邊緣（type = "e」），則對應於頂點或目標節點的 ID。

小鬼的例子

- 以下是 Gremlin 頂點標籤的範例。

```
{
  "id": "an ID string",
  "type": "vl",
  "key": "label",
  "value": {
    "value": "the new value of the vertex label",
    "dataType": "String"
  }
}
```

- 以下是 Gremlin 頂點屬性的範例。

```
{
  "id": "an ID string",
  "type": "vp",
  "key": "the property name",
  "value": {
    "value": "the new value of the vertex property",
    "dataType": "the datatype of the vertex property"
  }
}
```

- 以下是 Gremlin 邊緣的範例。

```
{
  "id": "an ID string",
  "type": "e",
  "key": "label",
  "value": {
    "value": "the new value of the edge",
    "dataType": "String"
  },
  "from": "the ID of the corresponding "from" vertex",
  "to": "the ID of the corresponding "to" vertex"
}
```

開放密碼示例

- 以下是開放密碼節點標籤範例。

```
{
  "id": "an ID string",
```

```
  "type": "vl",
  "key": "label",
  "value": {
    "value": "the new value of the node label",
    "dataType": "String"
  }
}
```

- 以下是開放密碼節點屬性的範例。

```
{
  "id": "an ID string",
  "type": "vp",
  "key": "the property name",
  "value": {
    "value": "the new value of the node property",
    "dataType": "the datatype of the node property"
  }
}
```

- 下面是一個開放密碼關係的一個例子。

```
{
  "id": "an ID string",
  "type": "e",
  "key": "label",
  "value": {
    "value": "the new value of the relationship",
    "dataType": "String"
  },
  "from": "the ID of the corresponding source node",
  "to": "the ID of the corresponding target node"
}
```

SPARQL NQUADS 變更序列化格式

Neptune 會使用 W3C RDF 1.1 N 四邊形規格中定義的資源描述架構 (RDF)N-QUADS 語言，將變更記錄到圖形中的 SPARQL 四邊形。

變更記錄中的 `data` 欄位只包含 `stmt` 欄位，其中保留 N-QUADS 陳述式，表示已變更的 quad，如下列範例所示。

```
 "stmt" : "<https://test.com/s> <https://test.com/p> <https://test.com/o> .\n"
```

Neptune 範例集

下列範例展示 Amazon Neptune

主題

- AT_SEQUENCE_NUMBER更改日誌 (p. 593)
- AFTER_SEQUENCE_NUMBER更改日誌 (p. 594)
- TRIM_HORIZON更改日誌 (p. 594)
- LATEST更改日誌 (p. 594)
- 壓縮變更日誌 (p. 595)

AT_SEQUENCE_NUMBER更改日誌

下列範例展示 G叢集。AT_SEQUENCE_NUMBER

```
curl -s "https://Neptune-DNS:8182/propertygraph/stream?
limit=1&commitNum=1&opNum=1&iteratorType=AT_SEQUENCE_NUMBER" |jq
{
  "lastEventId": {
    "commitNum": 1,
    "opNum": 1
  },
  "lastTrxTimestamp": 1560011610678,
  "format": "PG_JSON",
  "records": [
    {
      "eventId": {
        "commitNum": 1,
        "opNum": 1
      },
      "commitTimestamp": 1560011610678,
      "data": {
        "id": "d2b59bf8-0d0f-218b-f68b-2aa7b0b1904a",
        "type": "vl",
        "key": "label",
        "value": {
          "value": "vertex",
          "dataType": "String"
        }
      },
      "op": "ADD",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

這個顯示AT_SEQUENCE_NUMBER變更記錄檔的 SPARQL 範例。

```
curl -s "https://localhost:8182/sparql/stream?
limit=1&commitNum=1&opNum=1&iteratorType=AT_SEQUENCE_NUMBER" |jq
{
  "lastEventId": {
    "commitNum": 1,
    "opNum": 1
  },
  "lastTrxTimestamp": 1571252030566,
  "format": "NQUADS",
  "records": [
    {
      "eventId": {
        "commitNum": 1,
        "opNum": 1
      },
      "commitTimestamp": 1571252030566,
      "data": {
        "stmt": "<https://test.com/s> <https://test.com/p> <https://test.com/o> .\n"
      },
      "op": "ADD",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
```

```
}
```

AFTER_SEQUENCE_NUMBER更改日誌

下列範例展示 G叢集。AFTER_SEQUENCE_NUMBER

```
curl -s "https://Neptune-DNS:8182/propertygraph/stream?
limit=1&commitNum=1&opNum=1&iteratorType=AFTER_SEQUENCE_NUMBER" |jq
{
  "lastEventId": {
    "commitNum": 2,
    "opNum": 1
  },
  "lastTrxTimestamp": 1560011633768,
  "format": "PG_JSON",
  "records": [
    {
      "commitTimestamp": 1560011633768,
      "eventId": {
        "commitNum": 2,
        "opNum": 1
      },
      "data": {
        "id": "d2b59bf8-0d0f-218b-f68b-2aa7b0b1904a",
        "type": "vl",
        "key": "label",
        "value": {
          "value": "vertex",
          "dataType": "String"
        }
      },
      "op": "REMOVE",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

TRIM_HORIZON更改日誌

下列範例展示 G叢集。TRIM_HORIZON

```
curl -s "https://Neptune-DNS:8182/propertygraph/stream?limit=1&iteratorType=TRIM_HORIZON" |
jq
{
  "lastEventId": {
    "commitNum": 1,
    "opNum": 1
  },
  "lastTrxTimestamp": 1560011610678,
  "format": "PG_JSON",
  "records": [
    {
      "commitTimestamp": 1560011610678,
      "eventId": {
        "commitNum": 1,
        "opNum": 1
      },
      "data": {
        "id": "d2b59bf8-0d0f-218b-f68b-2aa7b0b1904a",
```

```
        "type": "vl",
        "key": "label",
        "value": {
          "value": "vertex",
          "dataType": "String"
        }
      },
      "op": "ADD",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

LATEST更改日誌

下列範例展示 G叢集。LATEST請注意，API 參數limitcommitNum、和opNum是完全可選的。

```
curl -s "https://Neptune-DNS:8182/propertygraph/stream?iteratorType=LATEST" | jq
{
  "lastEventId": {
    "commitNum": 21,
    "opNum": 4
  },
  "lastTrxTimestamp": 1634710497743,
  "format": "PG_JSON",
  "records": [
    {
      "commitTimestamp": 1634710497743,
      "eventId": {
        "commitNum": 21,
        "opNum": 4
      },
      "data": {
        "id": "24be4e2b-53b9-b195-56ba-3f48fa2b60ac",
        "type": "e",
        "key": "label",
        "value": {
          "value": "created",
          "dataType": "String"
        },
        "from": "4",
        "to": "5"
      },
      "op": "REMOVE",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

壓縮變更日誌

下列範例展示 G叢集。

```
curl -sH \
  "Accept-Encoding: gzip" \
  "https://Neptune-DNS:8182/propertygraph/stream?limit=1&commitNum=1" \
  -H "Accept-Encoding: gzip" \
  -v |gunzip -|jq
```

```
> GET /propertygraph/stream?limit=1 HTTP/1.1
> Host: localhost:8182
> User-Agent: curl/7.64.0
> Accept: /
> Accept-Encoding: gzip
*> Accept-Encoding: gzip*
>
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
< Connection: keep-alive
*< content-encoding: gzip*
< content-length: 191
<
{ [191 bytes data]
Connection #0 to host localhost left intact
{
  "lastEventId": "1:1",
  "lastTrxTimestamp": 1558942160603,
  "format": "PG_JSON",
  "records": [
    {
      "commitTimestamp": 1558942160603,
      "eventId": "1:1",
      "data": {
        "id": "v1",
        "type": "vl",
        "key": "label",
        "value": {
          "value": "person",
          "dataType": "String"
        }
      },
      "op": "ADD",
      "isLastOp": true
    }
  ],
  "totalRecords": 1
}
```

用AWS CloudFormation於使用串流使用者應用程式設定 Neptune 到海王星的複製

您可以使用AWS CloudFormation範本來設定 Neptune 串流取用者應用程式，以支援 Neptune 到 Neptune 的複寫。

主題

- 選擇您所在地區的AWS CloudFormation範本 (p. 596)
- 新增詳細資料關於您要建立的 Neptune 串流取用者堆疊 (p. 598)
- 執行 AWS CloudFormation 範本 (p. 600)
- 使用最新的 Lambda 成品更新串流輪詢程式 (p. 600)

選擇您所在地區的AWS CloudFormation範本

若要在AWS CloudFormation主控台上啟動適當的AWS CloudFormation堆疊，請根據您要使用的AWS區域選擇下表中的其中一個 Launch Stack 按鈕。

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (雪梨)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國東部)	檢視	在設計工具中檢視	Launch Stack

在 Create Stack (建立堆疊) 頁面中，選擇 Next (下一步)。

新增詳細資料關於您要建立的 Neptune 串流取用者堆疊

Specify Stack Details (指定堆疊詳細資訊) 頁面提供的屬性和參數，可用於控制應用程式設定。

您正在建立您正在建立之新建立之新AWS CloudFormation堆疊名稱。您通常可以使用預設值 `NeptuneStreamPoller`。

Parameters (參數) 下提供下列項目：

VPC 的網路組態串流用戶執行位置

- **`VPC`**— 提供將在其中執行輪詢 Lambda 函數的 VPC 名稱。
- **`SubnetIDs`**— 建立一個網絡接口。新增對應至 Neptune 叢集的子網路。
- **`SecurityGroupIds`**— 提供授與來源 Neptune DB 叢集寫入存取權的安全群組識別碼。
- **`RouteTableIds`**— 如果您還沒有一個 Neptune 您必須提供與子網路相關聯的路由表 ID 逗號分隔清單。
- **`CreateDDBVPCEndPoint`**— 預設為的布林值`true`，指出是否需要建立 Dynamo DB VPC 端點。只有在您的 VPC 擬私人雲端中`false`已建立 DynamoDB 端點時，才需要將其變更為。
- **`CreateMonitoringEndPoint`**— 預設為的 Boolean 值`true`，指出是否需要建立監視 VPC 端點。如果您已在 VPC 中建立監控端點，只需要將其變更為 `false` 即可。

串流輪詢器

- **`ApplicationName`**— 您通常可以將此設定保留為預設值 (`NeptuneStream`)。如果使用不同的名稱，其必須是唯一的。
- **`LambdaMemorySize`**— 用於設定 Lambda 輪詢函數可用的記憶體大小。預設值是 2,048 MB。
- **`LambdaRuntime`**— Lambda 函數中使用的語言，可從 Neptune 串流擷取項目。您可以將其設定為`python3.9`或`java8`。
- **`LambdaS3Bucket`**— 包含 Lambda 程式碼成品的 Amazon S3 儲存貯體。除非您使用的是從不同 Amazon S3 儲存貯體載入的自訂 Lambda 輪詢函數，否則請將此項保留空白。
- **`LambdaS3Key`**— 對應於您的 Lambda 程式碼成品的 Amazon S3 金鑰。除非您使用自訂的 Lambda 輪詢函數，否則請將此項保留空白。
- **`LambdaLoggingLevel`**— 一般而言，請將此設定保留為預設值，也就是`INFO`。
- **`ManagedPolicies`**— 列出用於執行 Lambda 函數的受管政策。一般而言，除非您使用自訂 Lambda 輪詢函數，否則請將此項保留空白。
- **`StreamRecordsHandler`**— 一般而言，除非您針對 Neptune 串流中的記錄使用自訂處理常式，否則請將此項保留空白。
- **`StreamRecordsBatchSize`**— 要從流中獲取的記錄數量。您可以使用此參數調整效能。預設值 (`5000`) 是很好的開始。允許的上限為 10,000。數字愈大，從串流讀取記錄所需的網路呼叫就愈少，但處理記錄所需的記憶體就愈多。較低的此參數值會導致較低的輸送量。

- **MaxPollingWaitTime**— 兩次民意調查之間的最長等待時間（以秒為單位）。決定叫用 Lambda 輪詢程式輪詢 Neptune 串流的頻率。將此值設為 0 可連續輪詢。最高值為 3,600 秒 (1 小時)。預設值 (60 秒) 是很好的開始，視圖表資料變更的速度而定。
- **MaxPollingInterval**— 最長連續輪詢期間 (以秒為單位)。使用此選項可設定 Lambda 輪詢函數的逾時時間。該值應介於 5 秒到 900 秒之間的範圍內。預設值 (600 秒) 是很好的開始。
- **StepFunctionFallbackPeriod**— 等待輪詢器的step-function-fallback-period單位數量，之後通過 Amazon E CloudWatch vents 調用步驟函數以從故障中恢復。預設值值 (5 分鐘) 是很好的開始。
- **StepFunctionFallbackPeriodUnit**— 用來測量前一個 `StepFunctionFallbackPeriodUnit` (`minuteshours`、或`days`) 的時間單位。默認（`minutes`）通常就足夠了。

Neptune

- **NeptuneStreamEndpoint**— (必要) Neptune 來源串流的端點。這採用以下兩種形式之一：
  - **https://*your DB cluster*:*port*/propertygraph/stream**（或其別名，`https://`*your DB cluster*`:`*port*`/pg/stream`）。
  - **https://*your DB cluster*:*port*/sparql/stream**.
- **Neptune Query Engine**— 選擇小精靈, 開放密碼, 或 SPARQL.
- **IAMAuthEnabledOnSourceStream**— 如果您的 Neptune 資料庫叢集使用 IAM 身份驗證，請將此參數設定為`true`。
- **StreamDBClusterResourceId**— 如果您的 Neptune 資料庫叢集使用 IAM 身份驗證，請將此參數設定為叢集資源 ID。資源 ID 和叢集 ID 不同。相反，它採用的形式是：`cluster-`後面接著 28 個字母數字字符。它可以在 Neptune 主控台的叢集詳細資料下找到。

目標 Neptune DB 叢集

- **TargetNeptuneClusterEndpoint**— 目標備份叢集的叢集端點 (僅限主機名稱)。

  請注意，如果您指定`TargetNeptuneClusterEndpoint`，您也無法指定`TargetSPARQLUpdateEndpoint`。
- **TargetNeptuneClusterPort**— 目標叢集的連接埠號碼。

  請注意，如果您指定`TargetSPARQLUpdateEndpoint`，則會忽略`TargetNeptuneClusterPort`的設定。
- **IAMAuthEnabledOnTargetCluster**— 如果要在目標叢集上啟用 IAM 身份驗證，則設定為 true。
- **TargetAWSRegion**— 目標備份叢集的AWS區域，例如`us-east-1`)。只有當目標備份叢集的AWS區域與 Neptune 來源叢集的區域不同時，您才必須提供此參數，如同跨區域複寫的情況一樣。如果來源和目標區域相同，則此參數為選擇性參數。

  請注意，如果該`TargetAWSRegion`值不是 Neptune 支援的有效AWS區域 (p. 843)，則程序會失敗。
- **TargetNeptuneDBClusterResourceId**— 可選：只有在目標資料庫叢集上啟用 IAM 身份驗證時，才需要這樣做。設定為目標叢集的資源 ID。
- **SPARQLTripleOnlyMode**— 決定是否啟用僅限三重模式的布林旗標。在僅限三重模式中，沒有命名圖形複寫。預設值為 `false`。
- **TargetSPARQLUpdateEndpoint**— SPARQL 更新之目標端點的 URL，例如`https://abc.com/xyz`。此端點可以是任何支援四元組或三元組的 SPARQL 存放區。

  請注意，如果您指定`TargetSPARQLUpdateEndpoint`，您也無法指定`TargetNeptuneClusterEndpoint`，且會忽略`TargetNeptuneClusterPort`的設定。
- **BlockSparqlReplicationOnBlankNode** — 布林旗標，如果設定為`true`，則會停止 SPARQL (RDF) 資料的複寫。BlankNode預設值為 `false`。

警示

- **`Required to create Cloud watch Alarm`**— `true` 如果要為新堆疊建立CloudWatch警示，請將此設定為。
- **`SNS Topic ARN for Cloudwatch Alarm Notifications`**— 應在其中傳送CloudWatch警示通知的 SNS 主題 ARN (僅在啟用警示時才需要)。
- **`Email for Alarm Notifications`**— 應傳送警示通知的電子郵件地址 (只有在啟用警報時才需要)。

針對警示通知的目的地，您可以僅新增 SNS、僅限電子郵件，或同時新增 SNS 和電子郵件。

執行 AWS CloudFormation 範本

現在您可以完成 Neptune 串流取用者應用程式執行個體的佈建程序，如下所示：

1. 在 AWS CloudFormation 的 Specify Stack Details (指定堆疊詳細資訊) 頁面上，選擇 Next (下一步)。
2. 在 Options (選項) 頁面上，選擇 Next (下一步)。
3. 在 [檢閱] 頁面上，選取第一個核取方塊，以確認AWS CloudFormation將建立 IAM 資源。選取第二個核取方塊，確認新堆疊 `CAPABILITY_AUTO_EXPAND`。

   Note

   `CAPABILITY_AUTO_EXPAND` 明確確認在建立堆疊時，無需事先檢閱，即可擴充巨集。使用者通常會在處理過的範本中建立變更集，以便在實際建立堆疊之前，檢閱巨集所做的變更。如需詳細資訊，請參閱 AWS CloudFormation CreateStackAPI 參考資料中的 AWS CloudFormation API。

   然後選擇 Create (建立)。

使用最新的 Lambda 成品更新串流輪詢程式

您可以使用最新的 Lambda 程式碼成品更新串流輪詢程式，如下所示：

1. 在中AWS Management Console，導覽至主父系AWS CloudFormation堆疊AWS CloudFormation並選取。
2. 選取堆疊的「更新」選項。
3. 選取「取代目前範本」。
4. 對於範本來源，請選擇 Amazon S3 URL，然後輸入以下 S3 URL：

```
https://aws-neptune-customer-samples.s3.amazonaws.com/neptune-stream/
neptune_to_neptune.json
```

5. 選取下一步而不變更任何AWS CloudFormation參數。
6. 請選擇 Update Stack (建立堆疊)。

堆疊現在會使用最新的工件來更新 Lambda 成品。

使用 Neptune 串流跨區域複寫進行災難復原

Neptune 提供兩種實作跨區域容錯移轉功能的方式：

- 跨區域快速複製與還原

- 使用 Neptune 串流在兩個不同區域的兩個叢集之間複寫資料。

跨區域快照複製與還原在不同區域中復原 Neptune 叢集的營運負荷最低。不過，在區域之間複製快照可能需要大量的資料傳輸時間，因為快照集是 Neptune 叢集的完整備份。因此，跨區域快照複製與還原可用於只需要小時的復原點目標 (RPO) 與復原時間目標 (RTO) 小時的案例。

復原點目標 (RPO) 由備份之間的時間來測量。它定義了在進行上次備份到恢復數據庫的時間之間可能會丟失多少數據。

復原時間目標 (RTO) 是以執行復原作業所需的時間來測量。這是資料庫叢集在發生失敗之後容錯移轉至復原的資料庫所花費的時間。

Neptune 串流可讓備份 Neptune 叢集隨時與主要實際執行叢集保持同步。如果發生失敗，您的資料庫就會容錯移轉至備份叢集。由於資料會不斷複製到備份叢集，因此可將 RPO 和 RTO 縮短為數分鐘，因此可隨時立即做為容錯移轉目標使用。

以這種方式使用 Neptune 串流的缺點是維護複寫元件所需的作業額外負荷，以及始終在線上使用第二個 Neptune 資料庫叢集的成本都很重要。

設定 Neptune 到海王星的複製

您的主要生產資料庫叢集位於指定來源區域中的 VPC 中。為了進行災難恢復，您需要在不同的恢復區域中複製或模擬三個主要內容：

- 存放在叢集中的資料。
- 主要叢集的配置。這包括是否使用 IAM 身份驗證、是否加密、資料庫叢集參數、執行個體參數、執行個體大小等等。
- 它使用的網路拓撲，包括目標 VPC、其安全群組等。

您可以使用 Neptune 管理 API，如下所示來收集該資訊：

- `DescribeDBClusters` (p. 1040)
- `DescribeDBInstances` (p. 1082)
- `DescribeDBClusterParameters` (p. 1099)
- `DescribeDBParameters` (p. 1097)
- `DescribeVpcs`

透過收集的資訊，您可以使用下列程序在不同的區域中設定備份叢集，在發生故障時，實際執行叢集可容錯移轉至此叢集。

1：啟用 Neptune 串流

您可以使用修改資料庫ClusterParameterGroup (p. 1094)將`neptune_streams`參數設定為 1。接著，重新啟動資料庫叢集中的所有執行個體，變更才會生效。

啟用 Neptune 串流之後，最好在來源資料庫叢集上執行至少一個新增或更新作業。這會將資料點填入變更串流，這些資料點可在稍後重新同步生產叢集與備份叢集時參考。

2: 在您要設定備份叢集的區域中建立新的 VPC

在與主要叢集不同的區域中建立新的 Neptune DB 叢集之前，您需要在目標區域中建立新的 VPC 來裝載叢集。主要叢集和備份叢集之間的連線是透過 VPC 對等建立的，這會使用不同 VPC 中私有子網路的流量。但是，若要在兩個 VPC 之間建立 VPC 對等，它們不得具有重疊的 CIDR 區塊或 IP 位址空間。您不能只在這兩個區域中使用預設 VPC，因為預設 VPC 的 CIDR 區塊永遠是相同的 (`172.31.0.0/16`)。

您可以在目標區域中使用該 VPC，只要符合以下條件即可：

- 無包含與主叢集所在 VPC CIDR 區塊重疊的 CIDR 區塊。
- 它尚未與另一個 VPC 對等，該 VPC 與主要叢集所在的 VPC 具有相同的 CIDR 區塊。

如果目標區域中沒有合適的 VPC 可用，請使用 Amazon EC2 CreateVpcAPI 建立一個虛擬私人雲端。

3：建立主要叢集的快照，並將其還原至目標備份區域

現在，您可以在目標備份區域中的適當 VPC 中建立新的 Neptune 叢集，該區域是生產叢集的副本：

在備份區域中製作生產叢集的副本

1. 在目標備份區域中，重新建立生產資料庫叢集所使用的參數和參數群組。您可以使用CreateDBClusterParameterGroup (p. 1091)、CreateDBParameterGroup (p. 1090)和來執行此操ModifyDBClusterParameterGroup (p. 1094)作ModifyDBParameterGroup (p. 1093)。

   請注意，CopyDBClusterParameterGroup (p. 1088)和 CopyDBParameterGroup (p. 1087)API 目前不支援跨區域複製。
2. 用CreateDBClusterSnapshot (p. 1110)於在生產區域的 VPC 中建立生產叢集的快照。
3. 用CopyDBClusterSnapshot (p. 1115)於將快照複製到目標備份區域中的 VPC。
4. 用於RestoreDBClusterFromSnapshot (p. 1119)使用複製的快照在目標備份區域的 VPC 中建立新的資料庫叢集。使用從主要生產叢集複製的組態設定和參數。
5. 新的 Neptune 叢集現已存在，但不包含任何執行個體。用CreateDBInstance (p. 1056)來建立新的主要/寫入器執行個體，該執行個體類型和大小與生產叢集的寫入器執行個體相同。此時不需要建立額外的僅供讀取複本，除非您的備份執行個體在容錯移轉之前用於服務目標區域的讀取 I/O。

4：在主要叢集的 VPC 與新備份叢集的 VPC 之間建立 VPC 對等

透過設定 VPC 對等，您可以讓主要叢集的 VPC 與備份叢集的 VPC 進行通訊，就像它們是單一私人網路一樣。若要執行此作業，請採用下列步驟：

1. 從生產叢集的 VPC 呼叫 CreateVpcPeeringConnectionAPI 以建立對等連線。
2. 從目標備份叢集的 VPC 呼叫 AcceptVpcPeeringConnectionAPI 以接受對等連線。
3. 從生產叢集的 VPC 中，使用 CreateRouteAPI 將路由新增至 VPC 的路由表，將所有流量重新導向至目標 VPC 的 CIDR 區塊，以便使用 VPC 對等首碼清單。
4. 同樣地，從目標備份叢集的 VPC，使用 CreateRouteAPI 將路由新增至 VPC 路由表的路由表，以將流量路由到主要叢集的 VPC。

5：設定 Neptune 串流複寫基礎結構

既然兩個叢集都已部署，並且已在兩個區域之間建立網路通訊，請使用 Neptune to NeptuneAWS CloudFormation 範本 (p. 596)，透過支援資料複寫的額外基礎架構部署 Neptune 串流消費者 Lambda 函數。在主要生產叢集的 VPC 中執行此操作。

您需要為此AWS CloudFormation堆棧提供的參數是：

- **NeptuneStreamEndpoint**— 主要叢集的串流端點，採用 URL 格式。例如：`https://(cluster name):8182/pg/stream`。
- **QueryEngine**— 必須是`gremlin`、`sparql`、或`openCypher`。
- **RouteTableIds**— 可讓您同時為 DynamoDB 虛擬私人雲端端點和監控 VPC 人雲端端點新增路由。

如果主要叢集的 VPC 上尚未存在兩個其他參數
(即`CreateMonitoringEndpoint`和)`CreateDynamoDBEndpoint`，也必須將這兩個參數設定為 true。如果它們已經存在，請確保它們設置為 false，否則AWS CloudFormation創建將失敗。

- **`SecurityGroupIds`**— 指定 Lambda 取用者用來與主要叢集的 Neptune 串流端點通訊的安全群組。

  在目標備份叢集中，附加允許源自此安全性群組之流量的安全性群組。
- **`SubnetIds`**— 主要叢集 VPC 中的子網路 ID 清單，Lambda 取用者可使用此清單與主要叢集進行通訊。
- **`TargetNeptuneClusterEndpoint`**— 目標備份叢集的叢集端點 (僅限主機名稱)。
- **`TargetAWSRegion`**— 目標備份叢集的AWS區域，例如`us-east-1`)。只有當目標備份叢集的AWS區域與 Neptune 來源叢集的區域不同時，您才必須提供此參數，如同跨區域複寫的情況一樣。如果來源和目標區域相同，則此參數為選擇性參數。

  請注意，如果該`TargetAWSRegion`值不是 Neptune 支援的有效AWS區域 (p. 843)，則程序會失敗。
- **`VPC`**— 主要叢集之 VPC 的識別碼。

所有其他參數都可以保留其預設值。

AWS CloudFormation範本部署完成後，Neptune 會開始將主要叢集的任何變更複製到備份叢集。您可以在 Lambda 取用者函數產生的 CloudWatch 記錄中監視此複寫。

其他考量

- 如果您需要在主叢集和備份叢集之間使用 IAM 身份驗證，也可以在調用AWS CloudFormation範本時進行設定。
- 如果主要叢集已啟用靜態加密，請考慮在跨目標區域複製快照並在目標區域中建立新的 KMS 金鑰關聯時，如何管理關聯的 KMS 金鑰。
- 最佳做法是在應用程式中使用的 Neptune 端點前面使用 DNS CNAME。然後，如果您需要手動容錯移轉至目標備份叢集，可以變更這些 CNames 以指向目標叢集和/或執行個體端點。

使用亞馬遜OpenSearch服務在亞馬遜海王星全文搜索

海王星與亞馬遜OpenSearch服務（服OpenSearch務）集成，以支持 Girmlin 和 SPARQL 查詢中的全文搜索。此功能從 Neptune 引擎版本 1.0.2.1 (p. 969) 開始提供，不過我們建議搭配引擎版本1.0.4.2或更高版本使用，以利用最新的修正程式。

> Note
>
> 與亞馬遜OpenSearch服務整合時，海王星需要使用彈性搜索 7.1 版或更高版本，並與 Opensearch 2.3、2.5 和未來兼容的 OpenSearch 版本一起使用。

您可以將 Neptune 與現有的OpenSearch服務叢集搭配使用，該叢集已根據數據的 Neptune 數 OpenSearch 據模型 (p. 611). 或者，您可以使用AWS CloudFormation堆疊建立與 Neptune 連結的OpenSearch服務網域。

> Important
>
> 此處描述的海王星到OpenSearch複寫程序不會複寫空白節點。這是需要注意的重要限制。
> 此外，如果您在OpenSearch叢集上啟用精細的存取控制，則還需要在 Neptune 資料庫中啟用 IAM 身份驗證 (p. 160)。

主題

- 亞馬遜海王星到複製設定 OpenSearch (p. 604)
- 從已啟用精細存取控制 (FGAC) 的OpenSearch叢集進行查詢 (p. 610)
- 在海王星全文檢索搜尋查詢中使用 Apache Lucene 查詢語法 (p. 610)
- 數據的 Neptune 數 OpenSearch 據模型 (p. 611)
- Neptune 全文檢索搜尋 (p. 614)
- Amazon Neptune 中的非字串 OpenSearch 索引 (p. 616)
- 在 Amazon Neptune Full-text-search 查詢執行 (p. 626)
- 在 Neptune 中使用全文檢索搜尋的 SPARQL 查詢範例 (p. 627)
- Gremlin 全文檢索搜尋 (p. 630)
- 排解海王星全文檢索搜尋 (p. 637)

亞馬遜海王星到複製設定 OpenSearch

亞馬遜海王星支持使用亞馬遜服務（OpenSearch服務）在 Gremlin 和 SPARQL 查詢中進行全文搜索。OpenSearch您可以使用AWS CloudFormation堆疊將OpenSearch服務網域連結至 Neptune。

在開始之前，您需要一個已啟用串流的現有 Neptune 資料庫叢集作為來源，以及一個OpenSearch服務網域做為複寫目標。

如果您已經擁有可由 Lambda 在 Neptune 資料庫叢集所在的 VPC 中存取的現有目標OpenSearch服務網域，則範本可以使用該網域。否則，您需要建立一個新的網域。

> Note
>
> 您建立的OpenSearch叢集和 Lambda 函數必須位於與 Neptune 資料庫叢集相同的 VPC 中，且叢集必須設定為 VPC 模式 (而非網際網路模式)。OpenSearch

建議您使用新建立的 Neptune 執行個體搭配OpenSearch服務使用。如果您使用已包含資料的現有執行個體，則應在進行查詢之前執行OpenSearch服務資料同步，否則可能有資料不一致的情況。此GitHub專案提供如何執行同步處理的範例：將海王星匯出至 OpenSearch (https://github.com/awslabs/ amazon-neptune-tools /樹/主export-neptune-to-elasticsearch/)。

Important

與亞馬遜OpenSearch服務整合時，海王星需要使用彈性搜索 7.1 版或更高版本，並與 Opensearch 2.3、2.5 和未來兼容的 OpenSearch 版本一起使用。

然後，下面的AWS CloudFormation範本會建立串流消費者應用程式執行個體，以提供 Neptune 到複寫功能。OpenSearch

使用您所在地區的AWS CloudFormation範本設定海王星至OpenSearch複製

若要在AWS CloudFormation主控台上啟動AWS CloudFormation堆疊，請根據您要使用的AWS區域選擇下表中的其中一個 Launch Stack 按鈕。

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國東部)	檢視	在設計工具中檢視	Launch Stack

在 Create Stack (建立堆疊) 頁面中，選擇 Next (下一步)。

新增關於您正在建立的新OpenSearch堆疊的詳細資料

「指定堆疊詳細資訊」頁面提供特性和參數，您可以用來控制全文檢索搜尋的設定：

堆疊名稱 — 您要建立的新AWS CloudFormation堆疊名稱。您通常可以使用預設值 `NeptuneStreamPoller`。

Parameters (參數) 下提供下列項目：

串流消費者執行之 VPC 的網路組態

- **VPC**— 提供將在其中執行輪詢 Lambda 函數的 VPC 名稱。
- **List of Subnet IDs**— 建立網路介面的子網路。新增對應至 Neptune 叢集的子網路。
- **List of Security Group Ids**— 提供授與來源 Neptune DB 叢集寫入存取權的安全群組識別碼。
- **List of Route Table Ids**— 如果您還沒有在海王星虛擬私人雲端中建立 Amazon DynamoDB 端點，就需要這麼做。您必須提供與子網路相關聯的路由表 ID 逗號分隔清單。
- **Require to create Dynamo DB VPC Endpoint**— 預設為的布林值`true`。只有在您的虛擬私人雲端中已建立 DynamoDB 端點時，才需要將`false`其變更為。
- **Require to create Monitoring VPC Endpoint**— 預設為的布林值`true`。如果您已在 VPC 中建立監控端點，只需要將其變更為 `false` 即可。

串流輪詢器

- **Application Name**— 您通常可以將此設定保留為預設值 (`NeptuneStream`)。如果使用不同的名稱，其必須是唯一的。
- **Memory size for Lambda Poller**— 用於設定 Lambda 輪詢函數可用的記憶體大小。預設值是 2,048 MB。

- **Lambda Runtime**— Lambda 函數中使用的語言，可從海王星串流擷取項目。您可以將其設定為`python3.9`或`java8`。
- **S3 Bucket having Lambda code artifacts**— 除非您使用的是從不同 S3 儲存貯體載入的自訂 Lambda 輪詢函數，否則請將此項保留空白。
- **S3 Key corresponding to Lambda Code artifacts**— 除非您使用自訂 Lambda 輪詢函數，否則請將此項保留空白。
- **Logging level for Lambda**— 一般而言，請將此設定保留為預設值`INFO`。
- **Managed Policies for Lambda Execution**— 一般而言，除非您使用自訂 Lambda 輪詢函數，否則請將此項保留空白。
- **Stream Records Handler**— 一般而言，除非您針對 Neptune 串流中的記錄使用自訂處理常式，否則請將此項保留空白。
- **Maximum records Fetched from Stream**— 您可以使用此參數來調整效能。預設值 (`100`) 是很好的開始。允許的上限為 10,000。數字愈大，從串流讀取記錄所需的網路呼叫就愈少，但處理記錄所需的記憶體就愈多。
- **Max wait time between two Polls (in Seconds)**— 決定叫用 Lambda 輪詢程式輪詢海王星串流的頻率。將此值設為 0 可連續輪詢。最高值為 3,600 秒 (1 小時)。預設值 (60 秒) 是很好的開始，視圖表資料變更的速度而定。
- **Maximum Continuous polling period (in Seconds)**-用於設置 Lambda 輪詢函數的超時。應介於 5 秒到 900 秒之間。預設值 (600 秒) 是很好的開始。
- **Step Function Fallback Period**— 等待輪詢器的step-function-fallback-period單位數量，之後通過 Amazon E CloudWatch vents 調用 step 函數以從故障中恢復。預設值值 (5 分鐘) 是很好的開始。
- **Step Function Fallback Period Unit**— 用於測量前一項的時間單位 `Step Function Fallback Period` (分鐘、小時、天數)。通常預設值 (分鐘) 即足夠。
- **Data replication scope**— 決定是要同時複製節點和邊緣，還是只複製節點到 OpenSearch (這僅適用於 Gemlin 引擎資料)。預設值 (All) 通常是很好的開始。
- **Ignore OpenSearch missing document error**— 用來決定中是否OpenSearch可忽略遺失文件錯誤的旗標。遺失文件錯誤很少發生，但如果不忽略，則需要手動介入。默認值 (`True`) 通常是一個很好的起點。
- **Enable Non-String Indexing**— 標記可啟用或停用沒有字串內容之欄位的索引。如果將此旗標設定為`true`，則非字串欄位會在中編製索引OpenSearch，或者 if `false` 僅對字串欄位進行索引。預設值為`true`。
- **Properties to exclude from being inserted into OpenSearch**— 要從索引中OpenSearch排除的屬性或述詞索引鍵的逗號分隔清單。如果此 CFN 參數值保留空白，則會對所有屬性索引鍵進行索引。
- **Datatypes to exclude from being inserted into OpenSearch**— 要從索引中排除的屬性或述詞資料類型的逗號分隔清單。OpenSearch如果此 CFN 參數值保留空白，則會對所有可安全地轉換為OpenSearch資料類型的屬性值進行索引。

海王星流

- **Endpoint of source Neptune Stream**— (必填) 這採用以下兩種形式之一：
  - **https://*your DB cluster*:*port*/propertygraph/stream**(或其別名`https://`*your DB cluster*`:`*port*`/pg/stream`)。
  - `https://`*your DB cluster*`:`*port*`/sparql/stream`
- **Neptune Query Engine**— 選擇小精靈或 SPARQL。
- **Is IAM Auth Enabled?**— 如果您的 Neptune 資料庫叢集使用 IAM 身份驗證，請將此參數設定為`true`。
- **Neptune Cluster Resource Id**— 如果您的 Neptune 資料庫叢集使用 IAM 身份驗證，請將此參數設定為叢集資源 ID。資源 ID 和叢集 ID 不同。相反，它採用的形式是：`cluster-`後面接著 28 個字母數字字符。它可以在 Neptune 主控台的叢集詳細資料下找到。

目標OpenSearch叢集

- **`Endpoint for OpenSearch service`**— (必要) 在 VPC 中提供OpenSearch服務的端點。
- **`Number of Shards for OpenSearch Index`**— 預設值 (5) 通常是開始的好地方。
- **`Number of Replicas for OpenSearch Index`**— 預設值 (1) 通常是開始的好地方。
- **`Geo Location Fields for Mapping`**— 如果您使用的是地理位置欄位，請在此列出屬性索引鍵。

警示

- **`Require to create Cloud watch Alarm`**— `true` 如果要為新堆疊建立CloudWatch警示，請將此設定為。
- **`SNS Topic ARN for Cloudwatch Alarm Notifications`**— 應在其中傳送CloudWatch警示通知的 SNS 主題 ARN (僅在啟用警示時才需要)。
- **`Email for Alarm Notifications`**— 應傳送警示通知的電子郵件地址 (只有在啟用警報時才需要)。

針對警示通知的目的地，您可以僅新增 SNS、僅限電子郵件，或同時新增 SNS 和電子郵件。

執行 AWS CloudFormation 範本

現在您可以完成 Neptune 串流取用者應用程式執行個體的佈建程序，如下所示：

1. 在 AWS CloudFormation 的 Specify Stack Details (指定堆疊詳細資訊) 頁面上，選擇 Next (下一步)。
2. 在 Options (選項) 頁面上，選擇 Next (下一步)。
3. 在 [檢閱] 頁面上，選取第一個核取方塊，以確認AWS CloudFormation將建立 IAM 資源。選取第二個核取方塊，確認新堆疊 `CAPABILITY_AUTO_EXPAND`。

   Note

   `CAPABILITY_AUTO_EXPAND` 明確確認在建立堆疊時，無需事先檢閱，即可擴充巨集。使用者通常會在處理過的範本中建立變更集，以便在實際建立堆疊之前，檢閱巨集所做的變更。如需詳細資訊，請參閱 AWS CloudFormation CreateStackAPI 參考資料中的 AWS CloudFormation API 作業。

   然後選擇 Create (建立)。

使用最新的 Lambda 成品更新串流輪詢器

您可以使用最新的 Lambda 程式碼成品更新串流輪詢程式，如下所示：

1. 在中AWS Management Console，導覽至主父系AWS CloudFormation堆疊AWS CloudFormation並選取。
2. 選取堆疊的「更新」選項。
3. 選取「取代目前範本」。
4. 對於範本來源，請選擇亞馬遜 S3 URL，然後輸入以下 S3 URL：

   ```
   https://aws-neptune-customer-samples.s3.amazonaws.com/neptune-stream/
   neptune_to_elastic_search.json
   ```

5. 選取下一步而不變更任何AWS CloudFormation參數。
6. 請選擇 Update Stack (建立堆疊)。

堆疊現在會使用最新的工件來更新 Lambda 成品。

在現有 Neptune 資料庫上啟用全文搜尋的不同方法

在現有 Neptune 資料庫上啟用全文搜尋的最佳方式通常如下，前提是您可以暫停寫入工作負載。它所需的停機時間受到限制，因為建立翻製的速度相對較快。

1. 停止資料庫上的所有寫入工作負載。
2. 建立資料庫的複製。
3. 啟用原始資料庫上的串流。
4. 繼續寫入工作負載。
5. 使用 github 上的這個海王星工具從複製執行一次性同步。
6. 使用此AWS CloudFormation範本可從原始資料庫同步處理，以便持續更新 (範本中不需要變更組態)。

如果您無法負擔在資料庫上暫停寫入工作負載的負擔，以下是一種比上述建議方法所需的停機時間更少的方法，但需要謹慎完成：

1. 啟用資料庫上的串流。
2. 建立資料庫的複製。
3. 記下來自克隆的流的最新`eventID`信息。
4. 使用 github 上的這個海王星工具從複製執行一次性同步。
5. 使用此AWS CloudFormation範本可從原始資料庫同步處理，以便持續更新。您確實需要設定AWS CloudFormation範本：在開始連續同步之前，請務必使用`eventID`您記錄的 DynamoDB 表中設定檢查點。

停用和重新啟用串流輪詢程序

> Warning
>
> 禁用流輪詢器過程時要小心！如果處理序暫停時間超過串流到期時間長，則可能會發生資料遺失。預設視窗為 7 天，但是從引擎版本 1.2.0.0 (p. 884) 開始，您可以將自訂串流到期時段設定為最多 90 天。

若要停用 (暫停) 串流輪詢程式處理程序

1. 登錄到AWS Management Console並打開亞馬遜EventBridge控制台 https://console.aws.amazon.com/events/.
2. 在導覽窗格中，選取 [規則]。
3. 選取規則，其名稱包含您在用來設定串流輪詢器的AWS CloudFormation範本中作為「應用程式名稱」提供的名稱。
4. 選擇 Disable (停用)。
5. 開啟步驟功能主控台，網址為 https://console.aws.amazon.com/states/。
6. 選取與串流輪詢程序相對應的執行步驟函數。同樣地，該步驟函數的名稱會包含您在用來設定串流輪詢器的AWS CloudFormation範本中，做為「應用程式名稱」提供的名稱。您可以按函數執行狀態進行篩選，以僅查看正在運行的函數。
7. 選擇 Stop (停止)。

若要重新啟用串流輪詢程式處理程序

1. 登錄到AWS Management Console並打開亞馬遜EventBridge控制台 https://console.aws.amazon.com/events/.

2. 在導覽窗格中，選取 [規則]。
3. 選取規則，其名稱包含您在用來設定串流輸詢器的AWS CloudFormation範本中作為「應用程式名稱」提供的名稱。
4. 選擇 Disable (停用)。根據指定的排程間隔的事件規則現在會觸發 Step 函數的新執行。

OpenSearch使用 Neptune 全文檢索搜尋查詢中的自訂欄位

目前的串流輸詢程式可以很容易地擴充為撰寫用於處理自訂欄位的自訂程式碼，如部落格文章中詳細說明：使用 Neptune Streams 擷取圖形變更。

Note

在中新增自訂欄位時OpenSearch，請務必將新欄位新增為述詞的內部物件 (請參閱Neptune 全文搜尋資料模型 (p. 611))。

從已啟用精細存取控制 (FGAC) 的OpenSearch叢集進行查詢

如果您已在OpenSearch叢集上啟用精細的存取控制，則還需要在 Neptune 資料庫中啟用 IAM 身份驗證 (p. 160)。

用於連線到 Neptune 資料庫的 IAM 實體 (使用者或角色) 應同時具有 Neptune 和OpenSearch叢集的權限。這意味著您的用戶或角色必須具有如下所示的OpenSearchOpenSearch服務：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-id:root"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:region:account-id:es-resource-id/*"
    }
  ]
}
```

如需詳細資訊，請參閱「亞馬遜海王星的自訂 IAM 資料存取政策聲明 (p. 203)」。

在海王星全文檢索搜尋查詢中使用 Apache Lucene 查詢語法

OpenSearch支持查詢字符串查詢使用阿帕奇 Lucene 語法。這對於在查詢中傳遞多個過濾器特別有用。

Neptune 會使用巢狀結構來儲存OpenSearch文件中的屬性 (請參閱Neptune 全文搜尋資料模型 (p. 611))。使用 Lucene 語法時，您需要使用此下一個模型中屬性的完整路徑。

這是一個小鬼的例子：

```
g.withSideEffect("Neptune#fts.endpoint", "es_endpoint")
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V()
 .has("*", "Neptune#fts predicates.name.value:\"Jane Austin\" AND entity_type:Book")
```

以下是一個例子：

```
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
  neptune-fts:config neptune-fts:endpoint 'http://localhost:9200 (http://
localhost:9200/)' .
  neptune-fts:config neptune-fts:queryType 'query_string' .
  neptune-fts:config neptune-fts:query "predicates.\\*foaf\\*name.value:Ronak AND
 predicates.\\*foaf\\*surname.value:Sh*" .
  neptune-fts:config neptune-fts:field '*' .
  neptune-fts:config neptune-fts:return ?res .
}
```

數據的 Neptune 數 OpenSearch 據模型

Amazon Neptune 使用統一的 JSON 文檔結構在 OpenSearch 服務中存儲 SPARQL 和格林數據。中的每個文件都 OpenSearch 對應一個實體，並儲存該實體的所有相關資訊。對於 Gemlin 而言，頂點和邊線被視為圖元，因此對應的 OpenSearch 文件具有頂點、標籤和屬性的相關資訊。對於 SPARQL，主題可以被視為實體，因此相應的 OpenSearch 文檔具有關於一個文檔中的所有謂詞-對象對的信息。

> **Note**
>
> 海王星-到-OpenSearch 複製實現只存儲字符串數據。但可加以修改以儲存其他資料類型。

統一的 JSON 文件結構如下所示。

```
{
  "entity_id": "Vertex Id/Edge Id/Subject URI",
  "entity_type": [List of Labels/rdf:type object value],
  "document_type": "vertex/edge/rdf-resource"
  "predicates": {
    "Property name or predicate URI": [
      {
        "value": "Property Value or Object Value",
        "graph": "(Only for Sparql) Named Graph Quad is present"
        "language": "(Only for Sparql) rdf:langString"
      },
      {
        "value": "Property Value 2/ Object Value 2",
      }
    ]
  }
}
```

- `entity_id`— 代表文件的實體唯一 ID。
  - 對於 SPARQL 而言，這是主體 URI。
  - 對於 Gremlin 而言，這是 `Vertex_ID` 或 `Edge_ID`。

- `entity_type`— 代表頂點或邊緣的一或多個標籤，或主旨的零個或多個`rdf:type`述詞值。
- `document_type`— 用於指定目前文件是否表示頂點、邊緣或 rdf 資源。
- `predicates`— 對於 Gemlin，儲存頂點或邊緣的屬性和值。若為 SPARQL，則儲存述詞物件對。

  屬性名稱採用表`properties.name.value`單 OpenSearch。若要查詢該屬性，則必須採用該形式加以命名。
- `value` —「小精靈」的屬性值或 SPARQL 的物件值。
- `graph`— SPARQL 的已命名圖形。
- `language`— SPARQL 中`rdf:langString`常值的語言標記。

OpenSearch 文件範例

資料

```
@prefix dt:   <http://example.org/datatype#> .
@prefix ex:   <http://example.org/> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:simone   rdf:type     ex:Person                    ex:g1
ex:michael  rdf:type     ex:Person                    ex:g1
ex:simone   ex:likes     "spaghetti"                  ex:g1

ex:simone   ex:knows     ex:michael                   ex:g2   # Not stored in ES
ex:simone   ex:likes     "spaghetti"                  ex:g2
ex:simone   ex:status    "La vita è un sogno"@it      ex:g2

ex:simone   ex:age       "40"^^xsd:int                DG      # Not stored in ES
ex:simone   ex:dummy     "testData"^^dt:newDataType   DG
ex:simone   ex:hates     _:bnode                              # Not stored in ES
_:bnode     ex:means     "coding"                     DG      # Not stored in ES
```

文件

```
{
  "entity_id": "http://example.org/simone",
  "entity_type": ["http://example.org/Person"],
  "document_type": "rdf-resource"
  "predicates": {
    "http://example.org/likes": [
      {
        "value": "spaghetti",
        "graph": "http://example.org/g1"
      },
      {
        "value": "spaghetti",
        "graph": "http://example.org/g2"
      }
    ]
    "http://example.org/status": [
      {
        "value": "La vita è un sogno",
        "language": "it"       // Only present for rdf:langString
      }
    ]
  }
}
```

```
{
  "entity_id" : "http://example.org/michael",
  "entity_type" : ["http://example.org/Person"],
  "document_type": "rdf-resource"
}
```

小精靈文 OpenSearch 件範例

資料

```
# Vertex 1
simone   label    Person       <== Label
simone   likes    "spaghetti"  <== Property
simone   likes    "rice"       <== Property
simone   age      40           <== Property

# Vertex 2
michael  label    Person       <== Label

# Edge 1
simone  knows     michael      <== Edge
e1      updated  "2019-07-03"  <== Edge Property
e1      through  "company"     <== Edge Property
e1      since     10          <== Edge Property
```

文件

```
{
  "entity_id": "simone",
  "entity_type": ["Person"],
  "document_type": "vertex",
  "predicates": {
    "likes": [
      {
        "value": "spaghetti"
      },
      {
        "value": "rice"
      }
    ]
  }
}
```

```
{
  "entity_id" : "michael",
  "entity_type" : ["Person"],
  "document_type": "vertex"
}
```

```
{
  "entity_id": "e1",
  "entity_type": ["knows"],
  "document_type": "edge"
  "predicates": {
    "through": [
      {
        "value": "company"
      }
```

```
    ]
  }
}
```

Neptune 全文檢索搜尋

Amazon Neptune 使用下列參數來指定 Gemlin 和 SPARQL 中的全文檢索 OpenSearch 查詢：

- **`queryType`**— (必要) 查 OpenSearch 詢的類型。（有關查詢類型的列表，請參閱OpenSearch 文檔）。Neptune 支持以下 OpenSearch 查詢類型：
  - simple_query_string — 使用具有有限但容錯 Lucene 語法的剖析器，根據提供的查詢字串傳回文件。這是預設的查詢類型。

    此查詢使用簡單的語法，將提供的查詢字串剖析並分割成以特殊運算子為基礎的字詞。然後，查詢會先個別分析每個字詞，再傳回相符的文件。

    雖然其語法比 `query_string` 查詢限制更多，但查 `simple_query_string` 詢不會傳回無效語法的錯誤。而是會忽略查詢字串所有無效的部分。
  - match —`match` 查詢是執行全文檢索搜尋的標準查詢，包括模糊比對的選項。
  - prefix — 傳回在提供的欄位中包含特定前置字元的文件。
  - 模糊 — 傳回包含與搜尋字詞類似的字詞的文件，這些字詞由 Levenshtein 編輯距離測量。

    編輯距離是指將一個字詞轉換成另一個字詞，所需要變更的單字元數量。這些變更可能包括：
    - 變更一個字元 (box 變成 fox)。
    - 移除一個字元(black 變成 lack)。
    - 插入一個字元 (sic 變成 sick)。
    - 調換兩個相鄰的字元 (act 變成 cat)。

    若要尋找類似的字詞，fuzzy 查詢會在指定的編輯距離內建立一組搜尋字詞所有可能的變異和擴展，然後傳回這些 variant 每一個的完全相符項目。
  - term — 傳回包含其中一個指定欄位中指定詞彙完全相符的文件。

    您可以使用 `term` 查詢，根據價格、產品 ID 或使用者名稱等精確值尋找文件。

    > Warning
    >
    > 文字欄位請避免使用 term 查詢。預設情況下， OpenSearch 變更文字欄位的值做為其分析的一部分，這可能會使尋找文字欄位值的完全相符項變得困難。
    > 若要搜尋文字欄位值，請改用 match 查詢。
  - query_string — 使用具有嚴格語法 (Lucene 語法) 的剖析器，根據提供的查詢字串傳回文件。

    此查詢會根據 AND 或 NOT 等運算子，使用語法剖析和分割提供的查詢字串。然後，查詢會先個別分析各分割文字，再傳回相符的文件。

    您可以使用 `query_string` 查詢建立包含萬用字元、跨多欄位搜尋及更多的複雜搜尋。查詢雖然多樣化，但卻很嚴格，且若查詢字串包含任何無效的語法即會傳回錯誤。

    > Warning
    >
    > 因為所有無效的語法都會傳回錯誤，所以我們不建議搜尋方塊使用 `query_string` 查詢。
    > 如果不需要支援查詢語法，請考慮使用 `match` 查詢。如果需要查詢語法的功能，請使用較不嚴格的 `simple_query_string` 查詢。
- **`field`**— 要在其中運 OpenSearch 行搜索的字段。只有在 `queryType` 允許時，才會忽略 (就像 `simple_query_string` 和 `query_string` 一樣)，在這種情況下，會針對所有欄位搜尋。在 Gremlin 中是隱含的。

如果查詢允許，可指定多個欄位，如同執行 `simple_query_string` 和 `query_string` 一樣。

- **`query`**— (必要) 要執行的查詢 OpenSearch。此欄位的內容可能會隨 queryType 而有所不同。不同的 queryTypes 接受不同的語法，例如，`Regexp`。在 Gremlin 中，`query` 是隱含的。
- **`maxResults`**要傳回的最層層數。預設值為`index.max_result_window` OpenSearch 設定，其本身預設值為 10,000。`maxResults` 參數可以指定任何低於該值的數字。

  Important

  如果您設定`maxResults`為大於該值的 OpenSearch `index.max_result_window`值，並嘗試擷取超過`index.max_result_window`結果，則 OpenSearch會失敗並顯示`Result window is too large`錯誤。但是，Neptune 會優雅地處理此問題，而不會傳播錯誤。如果您嘗試擷超過 `index.max_result_window` 的結果，請記住這一點。
- **`minScore`**— 搜索結果必須返回的最低分數。如需結果評分的說明，請參閱OpenSearch相關性文件。
- **`batchSize`**— Neptune 始終以批次方式擷取資料 (預設批次大小為 100)。您可以使用此參數調整效能。批次大小不能超過預`index.max_result_window` OpenSearch 設值為 10,000 的設定。
- **`sortBy`**— 選用參數，可讓您排序下列其中一項傳回的結果： OpenSearch
  - 文檔中的特定字符串字段 —

    例如，在 SPARQL 查詢中，您可以指定：

    ```
    neptune-fts:config neptune-fts:sortBy foaf:name .
    ```

    在類似的 Gremlin 查詢中，您可以指定：

    ```
    .withSideEffect('Neptune#fts.sortBy', 'name')
    ```
  - 文檔中的特定非字符串字段（*long*等）-double

    請注意，在非字符串字段上進行排序時，您需要附加`.value`到字段名稱以將其與字符串字段區分開來。

    例如，在 SPARQL 查詢中，您可以指定：

    ```
    neptune-fts:config neptune-fts:sortBy foaf:name.value .
    ```

    在類似的 Gremlin 查詢中，您可以指定：

    ```
    .withSideEffect('Neptune#fts.sortBy', 'name.value')
    ```
  - `score`— 依比賽分數排序 (預設值)。

    如果 `sortOrder` 參數存在但 `sortBy` 不存在，則會依 `sortOrder` 指定的順序由 `score` 排序結果。
  - `id`— 依 ID 排序，這表示 SPARQL 主旨 URI 或小鬼頂點或邊緣識別碼。

    例如，在 SPARQL 查詢中，您可以指定：

    ```
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.entity_id' .
    ```

    在類似的 Gremlin 查詢中，您可以指定：

    ```
    .withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.entity_id')
    ```
  - `label`— 按標籤排序。

    例如，在 SPARQL 查詢中，您可以指定：

```
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.entity_type' .
```

在類似的 Gremlin 查詢中，您可以指定：

```
    .withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.entity_type')
```

- doc_type— 依文件類型 (也就是 SPARQL 或小精靈) 排序。

  例如，在 SPARQL 查詢中，您可以指定：

```
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.document_type' .
```

  在類似的 Gremlin 查詢中，您可以指定：

```
    .withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.document_type')
```

  根據預設， OpenSearch 結果不會排序，而且其順序不具決定性，這表示相同的查詢可能會在每次執行時以不同的順序傳回項目。因此，如果結果集大於 max_result_window，則每次執行查詢時都會傳回完全不同的總結果子集。但透過排序，您可以更直接地比較不同執行的結果。

  若無 sortOrder 參數伴隨 sortBy，則會採用從最大到最小的遞減 (DESC) 順序。

- **sortOrder**— 選用參數，可讓您指定 OpenSearch 結果是從最小到最大還是從最大到最小排序 (預設值)：

  - ASC— 升序, 從最小到最大.
  - DESC— 降序, 從最大到最小.

    此為預設值，可在 sortBy 參數存在，但未指定 sortOrder 的情況時使用。

  如果都不存在sortBysortOrder也不存在，則默認情況下不會對 OpenSearch結果進行排序。

Amazon Neptune 中的非字串 OpenSearch 索引

Amazon Neptune 中的非字串 OpenSearch 索引允許將述詞的非字串值複寫為 OpenSearch 使用串流輸詢器。然後，可以安全地轉換為 OpenSearch 對應的映射或數據類型的所有謂詞值被複製到 OpenSearch。

若要在新堆疊上啟用非字串索引，必須將AWS CloudFormation範本中的Enable Non-String Indexing旗標設定為true。這是預設設定。要更新現有堆棧以支持非字符串索引，請參閱堆疊 (p. 617)下文。

Note

- 最好不要在早於之前的引擎版本上啟用非字串索引**1.0.4.2**。
- OpenSearch 使用正則表達式查詢匹配多個字段的字段名稱（其中一些包含字符串值，其中一些包含非字符串值，其中一些包含非字符串值）會失敗並顯示錯誤。如果 Neptune 中的全文檢索搜尋查詢屬於該類型，也會發生同樣的情況。
- 按非字符串字段進行排序時，將「.value」附加到字段名稱以將其與字符串字段區分開來。

內容

- 更新現有的 Neptune 全文檢索搜尋堆疊以支援非字串索引 (p. 617)
- 篩選 Neptune 全文檢索搜尋中要索引的欄位 (p. 618)
  - 按屬性或謂詞名稱過濾 (p. 618)

    - 按屬性或謂詞值類型過濾 (p. 618)
- SPARQL 和小精靈數據類型的映射到 OpenSearch (p. 619)
- 驗證數據映射 (p. 621)
- Neptune 中的非字符串 OpenSearch 查詢示例 (p. 624)
    - 1. 獲取年齡大於 30 和名稱開頭的所有頂點「Si」 (p. 624)
    - 2. 獲取年齡在 10 和 50 之間的所有節點，並與「Ronka」模糊匹配的名稱 (p. 625)
    - 3. 獲取具有最近 25 天內的時間戳的所有節點 (p. 625)
    - 4. 獲取具有給定年份和月份內的時間戳的所有節點 (p. 626)

更新現有的 Neptune 全文檢索搜尋堆疊以支援非字串索引

如果您已經在使用 Neptune 全文檢索搜尋，以下是支援非字串索引所需採取的步驟：

1. 停止流輸詢器 Lambda 函數。這可確保在匯出期間不會複製任何新更新。若要執行此操作，請停用叫用 Lambda 函數的雲端事件規則：
    - 在中AWS Management Console，導覽至 CloudWatch。
    - 選取「規則」。
    - 選擇具有 Lambda 串流輸詢程式名稱的規則。
    - 選取 [停用] 以暫時停用規則。
2. 刪除中目前的 Neptune 索引 OpenSearch。使用下列`curl`查詢從 OpenSearch 叢集刪除`amazon_neptune`索引：

    ```
    curl -X DELETE "your OpenSearch endpoint/amazon_neptune"
    ```
3. 開始一次性從 Neptune 匯出到 OpenSearch. 最好在此時設定新的 OpenSearch 堆疊，以便為執行匯出的輸詢程式挑選新的成品。

    按照此處列出的步驟 GitHub開始將 Neptune 數據的一次性導出到中 OpenSearch。
4. 更新現有串流輸詢程式的 Lambda 成品。將 Neptune 資料匯出成功完 OpenSearch 成之後，請執行下列步驟：
    - 在 AWS Management Console 中，導覽至 AWS CloudFormation。
    - 選擇主父AWS CloudFormation堆棧。
    - 選取該堆疊的「更新」選項。
    - 選取「從選項取代目前的範本」。
    - 對於範本來源，請選取 Amazon S3 URL。
    - 對於 Amazon S3 URL，請輸入：

        ```
        https://aws-neptune-customer-samples.s3.amazonaws.com/neptune-stream/
        neptune_to_elastic_search.json
        ```
    - 不變更任何AWS CloudFormation參數，請選擇「下一步」。
    - 選取 [更新堆疊]。 AWS CloudFormation將以最新的成品取代串流輸詢程式的 Lambda 程式碼成品。
5. 再次啟動串流輸詢器。透過啟用適當的 CloudWatch 規則來執行此操作：
    - 在中AWS Management Console，導覽至 CloudWatch。
    - 選取「規則」。
    - 選擇具有 Lambda 串流輸詢程式名稱的規則。

- 選取 [啟用]。

篩選 Neptune 全文檢索搜尋中要索引的欄位

AWS CloudFormation範本詳細資料中有兩個欄位可讓您指定要從索引中排除的屬性或述詞 OpenSearch 索引鍵或資料類型：

按屬性或謂詞名稱過濾

您可以使用名為的選擇性AWS CloudFormation範本參數，`Properties to exclude from being inserted into Elastic Search Index`提供逗號分隔的屬性清單或述詞索引鍵，以便從 OpenSearch 索引中排除。

例如，假設您將此參數設為`bob`：

```
"Properties to exclude from being inserted into Elastic Search Index" : bob
```

在這種情況下，下面的 Gemlin 更新查詢的流記錄將被丟棄，而不是進入索引：

```
g.V("1").property("bob", "test")
```

同樣地，您可以將參數設定為`http://my/example#bob`：

```
"Properties to exclude from being inserted into Elastic Search Index" : http://my/
example#bob
```

在這種情況下，下列 SPARQL 更新查詢的串流記錄會捨棄，而不是進入索引：

```
PREFIX ex: <http://my/example#>
INSERT DATA { ex:s1 ex:bob "test"}.
```

如果您沒有在此AWS CloudFormation模板參數中輸入任何內容，則未以其他方式排除的所有屬性鍵將被索引。

按屬性或謂詞值類型過濾

您可以使用名為的選擇性AWS CloudFormation範本參數，提供`Datatypes to exclude from being inserted into Elastic Search Index`要從 OpenSearch 索引中排除的屬性或述詞值資料類型的逗號分隔清單。

對於 SPARQL，您不需要列出完整的 XSD 類型 URI，只需列出數據類型令牌即可。您可以列出的有效數據類型令牌是：

- `string`
- `boolean`
- `float`
- `double`
- `dateTime`
- `date`
- `time`
- `byte`
- `short`
- `int`

- long
- decimal
- integer
- nonNegativeInteger
- nonPositiveInteger
- negativeInteger
- unsignedByte
- unsignedShort
- unsignedInt
- unsignedLong

對於 Gremlin，要列出的有效數據類型是：

- string
- date
- bool
- byte
- short
- int
- long
- float
- double

例如，假設您將此參數設為`string`：

```
"Datatypes to exclude from being inserted into Elastic Search Index" : string
```

在這種情況下，下面的 Gemlin 更新查詢的流記錄將被丟棄，而不是進入索引：

```
g.V("1").property("myStringval", "testvalue")
```

同樣地，您可以將參數設定為`int`：

```
"Datatypes to exclude from being inserted into Elastic Search Index" : int
```

在這種情況下，下列 SPARQL 更新查詢的串流記錄會捨棄，而不是進入索引：

```
PREFIX ex: <http://my/example#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
INSERT DATA { ex:s1 ex:bob "11"^^xsd:int }.
```

如果您沒有在此AWS CloudFormation範本參數中輸入任何內容，則其值可以安全地轉換為 OpenSearch 對等項目的所有屬性都會建立索引。系統會忽略查詢語言不支援的列出類型。

SPARQL 和小精靈數據類型的映射到 OpenSearch

中的新資料類型對應 OpenSearch 是根據屬性或物件中使用的資料類型而建立的。由於某些欄位包含不同類型的值，因此初始對應可能會排除欄位的某些值。

Neptune 數據類型映射到 OpenSearch 數據類型，如下所示：

SPARQL 類型	小精靈類型	OpenSearch 類型
XSD:int XSD:unsignedInt XSD:integer XSD:byte XSD:unsignedByte XSD:short XSD:unsignedShort XSD:long XSD:unsignedLong	byte short int long	long
XSD:float XSD:double XSD:decimal	float double	double
XSD:boolean	bool	boolean
XSD:datetime XSD:date	date	date
XSD:string XSD:time	string	text
自訂資料類型	無	text
任何其他數據類型	無	text

例如，下面的 Gemlin 更新查詢會導致「NewField」的新映射被添加到 OpenSearch，即`{ "type" : "double" }`：

```
g.V("1").property("newField" 10.5)
```

同樣地，下列 SPARQL 更新查詢會導致新增「ex: byte」的新對應 OpenSearch，亦即`{ "type" : "long" }`：

```
PREFIX ex: <http://my/example#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>

INSERT DATA { ex:test ex:byte "123"^^xsd:byte }.
```

Note

正如您所看到的，從 Neptune 映射到的項目可 OpenSearch 能會得到與 Neptune 中 OpenSearch 不同的數據類型。不過，「資料類型」中 OpenSearch有一個明確的文字欄位，可記錄項目在 Neptune 中的資料類型。

驗證數據映射

資料會使用下列程序 OpenSearch 從 Neptune 複製到：

- 如果有問題欄位的對應已存在於 OpenSearch：
  - 如果可以使用資料驗證規則將資料安全地轉換為現有對應，請將欄位儲存在中 OpenSearch。
  - 如果沒有，請刪除相應的串流更新記錄。
- 如果有問題的欄位沒有現有的對應，請尋找與 Neptune 中欄位資 OpenSearch料類型相對應的資料類型。
  - 如果可以使用資料驗證規則將欄位資 OpenSearch 料安全地轉換為資料類型，則將新對應和欄位資料儲存在中 OpenSearch。
  - 如果沒有，請刪除相應的串流更新記錄。

系統會根據對等 OpenSearch 類型或現有對 OpenSearch 應 (而非 Neptune 類型) 驗證值。例如，`"123"`中的值驗證`"123"^^xsd:int`是針對`long`類型而不是類型進行`int`的。

雖然 Neptune 嘗試將所有資料複寫到 OpenSearch，但在某些情況下，中的資料類型與 Neptune 中 OpenSearch 的資料類型完全不同，在這種情況下，會略過記錄而不是在中建立索引 OpenSearch。

例如，在 Neptune 中，一個屬性可以具有不同類型的多個值，而在 OpenSearch 欄位中的索引中必須具有相同的類型。

透過啟用偵錯記錄檔，您可以檢視從 Neptune 匯出到 OpenSearch. 除錯記錄項目的範例為：

```
Dropping Record : Data type not a valid Gremlin type
<Record>
```

數據類型被驗證如下：

- **`text`**— Neptune 中的所有值都可以安全地對應至中的文字 OpenSearch。
- **`long`**— 當 OpenSearch 對應類型很長時，Neptune 資料類型的下列規則適用 (在以下範例中，假設`"testLong"`具有`long`對應類型)：
  - `boolean`— 無效，無法轉換，並捨棄對應的串流更新記錄。

    無效的小鬼實例如下：

    ```
      "testLong" : true.
      "testLong" : false.
    ```

    無效的範例如下：

    ```
      ":testLong" : "true"^^xsd:boolean
      ":testLong" : "false"^^xsd:boolean
    ```
  - `datetime`— 無效，無法轉換，並捨棄對應的串流更新記錄。

    一個無效的小鬼的例子是：

    ```
      ":testLong" :  datetime('2018-11-04T00:00:00').
    ```

    一個無效的 SPARQL 範例為：

    ```
      ":testLong" : "2016-01-01"^^xsd:date
    ```

- floatdouble，或decimal-如果 Neptune 中的值是可以容納 64 位的整數，它是有效的，並且存儲在 OpenSearch 一個很長的時間，但如果它有一個小數部分，或者是NaN或者是一個INF或大於 9,223,372,036,372,036,854,775,807 或小於 -9,223,372,036,854,7807 或小於 -9,223,372,036,875,807 或小於 -9,223,372,036,875,807。

  有效的小鬼示例如下：

```
"testLong" :  145.0.
":testLong" :  123
":testLong" :  -9223372036854775807
```

  有效的範例如下：

```
":testLong" : "145.0"^^xsd:float
":testLong" :  145.0
":testLong" : "145.0"^^xsd:double
":testLong" : "145.0"^^xsd:decimal
":testLong" : "-9223372036854775807"
```

  無效的小鬼實例如下：

```
"testLong" :  123.45
":testLong" :  9223372036854775900
```

  無效的範例如下：

```
":testLong" :  123.45
":testLong" :  9223372036854775900
":testLong" : "123.45"^^xsd:float
":testLong" : "123.45"^^xsd:double
":testLong" : "123.45"^^xsd:decimal
```

- string— 如果 Neptune 中的值是可包含在 64 位元整數中之整數的字串表示，則該值為有效且會轉換為long in OpenSearch。對於 Elasticseearchlong 映射，任何其他字符串值都無效，並且會丟棄相應的流更新記錄。

  有效的小鬼示例如下：

```
"testLong" :  "123".
":testLong" :  "145.0"
":testLong" :  "-9223372036854775807"
```

  有效的範例如下：

```
":testLong" : "145.0"^^xsd:string
":testLong" : "-9223372036854775807"^^xsd:string
```

  無效的小鬼實例如下：

```
"testLong" :  "123.45"
":testLong" :  "9223372036854775900"
":testLong" :  "abc"
```

  無效的範例如下：

```
  ":testLong" : "123.45"^^xsd:string
  ":testLong" : "abc"
  ":testLong" : "9223372036854775900"^^xsd:string
```

- **`double`**— 如果 OpenSearch 對應類型為`double`，則套用下列規則 (此處假設「TestDouble」欄位在中具有`double`對應 OpenSearch)：
  - `boolean`— 無效，無法轉換，並捨棄對應的串流更新記錄。

    無效的小鬼實例如下：

    ```
      "testDouble" : true.
      "testDouble" : false.
    ```

    無效的範例如下：

    ```
      ":testDouble" : "true"^^xsd:boolean
      ":testDouble" : "false"^^xsd:boolean
    ```

  - `datetime`— 無效，無法轉換，並捨棄對應的串流更新記錄。

    一個無效的小鬼的例子是：

    ```
      ":testDouble" :  datetime('2018-11-04T00:00:00').
    ```

    一個無效的 SPARQL 範例為：

    ```
      ":testDouble" : "2016-01-01"^^xsd:date
    ```

  - 浮點數`NaN`或`INF` — 如果 SPARQL 中的值是浮點數`NaN`或`INF`，則無效且對應的資料流更新記錄會捨棄。

    無效的範例如下：

    ```
    "  :testDouble" : "NaN"^^xsd:float
      ":testDouble" : "NaN"^^double
      ":testDouble" : "INF"^^double
      ":testDouble" : "-INF"^^double
    ```

  - 數字或數字字串 — 如果 Neptune 中的值是任何其他數字或數值字串表示形式，可以安全地表示為`adouble`，則該值為有效且會轉換為`double` in OpenSearch。對於 OpenSearch `double`映射，任何其他字符串值都無效，並且會丟棄相應的流更新記錄。

    有效的小鬼示例如下：

    ```
      "testDouble" :  123
      ":testDouble" :  "123"
      ":testDouble" :  145.67
      ":testDouble" :  "145.67"
    ```

    有效的範例如下：

    ```
      ":testDouble" :  123.45
      ":testDouble" :  145.0
      ":testDouble" : "123.45"^^xsd:float
      ":testDouble" : "123.45"^^xsd:double
      ":testDouble" : "123.45"^^xsd:decimal
    ```

```
  ":testDouble" : "123.45"^^xsd:string
```

一個無效的小鬼的例子是：

```
  ":testDouble" :  "abc"
```

無效的 SPARQL 範例為：

```
  ":testDouble" : "abc"
```

- **`date`**— 如果 OpenSearch 對應類型為`date` Neptune`date` 和`dateTime`值是有效的，就像任何可以成功剖析為`dateTime`格式的字串值一樣。

  小鬼或 SPARQL 中的有效範例如下：

```
  Date(2016-01-01)
  "2016-01-01" "
  2003-09-25T10:49:41"
  "2003-09-25T10:49"
  "2003-09-25T10"
  "20030925T104941-0300"
  "20030925T104941"
  "2003-Sep-25" "
  Sep-25-2003"
  "2003.Sep.25"
  "2003/09/25"
  "2003 Sep 25" "
  Wed, July 10, '96"
  "Tuesday, April 12, 1952 AD 3:30:42pm PST"
  "123"
  "-123"
  "0"
  "-0"
  "123.00"
  "-123.00"
```

  無效的範例包括：

```
  123.45
  True
  "abc"
```

Neptune 中的非字符串 OpenSearch 查詢示例

Neptune 目前不直接支持 OpenSearch 範圍查詢。但是，您可以使用 Lucene 語法和查詢類型 =「query_string」來實現相同的效果，正如您在下面的示例查詢中看到的那樣。

1. 獲取年齡大於 30 和名稱開頭的所有頂點「Si」

在小精靈：

```
g.withSideEffect('Neptune#fts.endpoint', 'http://your-es-endpoint')
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V().has('*', 'Neptune#fts predicates.age.value:>30 && predicates.name.value:Si*');
```

在 SPARQL 中：

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://localhost:9200' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query "predicates.\\*foaf\\*age.value:>30 AND
 predicates.\\*foaf\\*name.value:Si*" .
    neptune-fts:config neptune-fts:field '*' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

為了簡潔起見，這裡使用而不"*foaf*age是完整的 URI。此正則表達式將檢索所有具有foaf和URIage 中的字段。

2. 獲取年齡在 10 和 50 之間的所有節點，並與「Ronka」模糊匹配的名稱

在小精靈：

```
g.withSideEffect('Neptune#fts.endpoint', 'http://your-es-endpoint')
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V().has('*', 'Neptune#fts predicates.age.value:[10 TO 50] AND
 predicates.name.value:Ronka~');
```

在 SPARQL 中：

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://localhost:9200' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query "predicates.\\*foaf\\*age.value:[10 TO 50] AND
 predicates.\\*foaf\\*name.value:Ronka~" .
    neptune-fts:config neptune-fts:field '*' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

3. 獲取具有最近 25 天內的時間戳的所有節點

在小精靈：

```
g.withSideEffect('Neptune#fts.endpoint', 'http://your-es-endpoint')
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V().has('*', 'Neptune#fts predicates.timestamp.value:>now-25d');
```

在 SPARQL 中：

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://localhost:9200' .
```

```
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query "predicates.\\*foaf\\*timestamp.value:>now-25d~" .
    neptune-fts:config neptune-fts:field '*' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

4. 獲取具有給定年份和月份內的時間戳的所有節點

在小鬼中，在 Lucene 語法中使用日期數學表達式，對於 2020 年 12 月：

```
g.withSideEffect('Neptune#fts.endpoint', 'http://your-es-endpoint')
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V().has('*', 'Neptune#fts predicates.timestamp.value:>2020-12');
```

一個小精靈的替代方案：

```
g.withSideEffect('Neptune#fts.endpoint', 'http://your-es-endpoint')
 .withSideEffect("Neptune#fts.queryType", "query_string")
 .V().has('*', 'Neptune#fts predicates.timestamp.value:[2020-12 TO 2021-01]');
```

在 Amazon Neptune Full-text-search 查詢執行

在包含的查詢中 full-text-search，Neptune 會嘗試先將 full-text-search呼叫放在查詢的其他部分之前。這樣可減少呼叫次數， OpenSearch 並且在大多數情況下會大幅改善效能。但是，這絕不是一個 hard-and-fast 規則。例如，在某些情況下，PatternNode或UnionNode可能位於全文檢索搜尋呼叫之前。

請考慮下列 Gemlin 查詢到包含 100,000 個執行個體的資料庫Person：

```
g.withSideEffect('Neptune#fts.endpoint', 'your-es-endpoint-URL')
 .hasLabel('Person')
 .has('name', 'Neptune#fts marcello~');
```

如果這個查詢是按照步驟出現的順序執行，則 100,000 個解決方案會流入 OpenSearch，導致數百個 OpenSearch 呼叫。事實上，Neptune OpenSearch 先呼叫，然後再將結果與 Neptune 結果聯結。在大多數情況下，這樣做比依原始順序執行查詢快得多。

您可以使用 noReordering 查詢提示 (p. 321)來防止重新排序查詢步驟執行：

```
g.withSideEffect('Neptune#fts.endpoint', 'your-es-endpoint-URL')
 .withSideEffect('Neptune#noReordering', true)
 .hasLabel('Person')
 .has('name', 'Neptune#fts marcello~');
```

在這第二種情況下，會先執行 .hasLabel 步驟，再執行 .has('name', 'Neptune#fts marcello~') 步驗。

對於另一個範例，考慮對相同類型的資料進行 SPARQL 查詢：

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT ?person WHERE {
  ?person rdf:type foaf:Person .
```

```
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'mike' .
    neptune-fts:config neptune-fts:return ?person .
  }
}
```

在這裡，Neptune 首先執行查詢的一SERVICE部分，然後將結果與Person數據加入。您可以使用 joinOrder 查詢提示 (p. 438)來隱藏此行為：

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
PREFIX hint: <http://aws.amazon.com/neptune/vocab/v01/QueryHints#>
SELECT ?person WHERE {
  hint:Query hint:joinOrder "Ordered" .
  ?person rdf:type foaf:Person .
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'mike' .
    neptune-fts:config neptune-fts:return ?person .
  }
}
```

再次，在第二個查詢中，這些部分會按照它們出現在查詢的順序執行。

在 Neptune 中使用全文檢索搜尋的 SPARQL 查詢範例

以下是一些在 Amazon Neptune 中使用全文檢索搜尋的 SPARQL 查詢範例。

匹配查詢實例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'match' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'michael' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

前綴查詢範例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'prefix' .
```

```
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'mich' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

模糊查詢範例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'fuzzy' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'mikael' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

術語查詢範例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'term' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:query 'Dr. Kunal' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

查詢字串查詢範例

此查詢可指定多個欄位。

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ OR rondelli' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:field foaf:surname .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

簡單查詢字串範例

下列查詢使用萬用字元 ('*') 指定欄位。

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint.com' .
    neptune-fts:config neptune-fts:queryType 'simple_query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field '*' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

SPARQL 按字符串字段排序查詢實例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:sortOrder 'asc' .
    neptune-fts:config neptune-fts:sortBy foaf:name .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

SPARQL 按非字符串字段查詢實例進行排序

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field foaf:name.value .
    neptune-fts:config neptune-fts:sortOrder 'asc' .
    neptune-fts:config neptune-fts:sortBy dc:date.value .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

按識別碼排序查詢範例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:sortOrder 'asc' .
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.entity_id' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

SPARQL 按標籤查詢範例排序

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:sortOrder 'asc' .
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.entity_type' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

按文檔類型排序查詢實例

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'mikael~ | rondelli' .
    neptune-fts:config neptune-fts:field foaf:name .
    neptune-fts:config neptune-fts:sortOrder 'asc' .
    neptune-fts:config neptune-fts:sortBy 'Neptune#fts.document_type' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

在 SPARQL 中使用 Lucene 語法的示例

Lucene 語法僅支援中的`query_string`查詢 OpenSearch。

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX neptune-fts: <http://aws.amazon.com/neptune/vocab/v01/services/fts#>
SELECT * WHERE {
  SERVICE neptune-fts:search {
    neptune-fts:config neptune-fts:endpoint 'http://your-es-endpoint' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:queryType 'query_string' .
    neptune-fts:config neptune-fts:query 'predicates.\\foaf\\name.value:micheal AND
 predicates.\\foaf\\surname.value:sh' .
    neptune-fts:config neptune-fts:field '' .
    neptune-fts:config neptune-fts:return ?res .
  }
}
```

Gremlin 全文檢索搜尋

`NeptuneSearchStep` 可以對未轉換為 Neptune 步驟的 Gremlin 周遊部分，進行全文搜尋查詢。例如，考慮使用以下查詢。

```
g.withSideEffect("Neptune#fts.endpoint", "your-es-endpoint-URL")
```

```
  .V()
      .tail(100)
      .has("name", "Neptune#fts mark*")            <== # Limit the search on name
```

這個查詢被轉換成 Neptune 下面的優化遍歷。

```
Neptune steps:
[
    NeptuneGraphQueryStep(Vertex) {
        JoinGroupNode {
            PatternNode[(?1, <~label>, ?2, <~>) . project distinct ?1 .],
 {estimatedCardinality=INFINITY}
        }, annotations={path=[Vertex(?1):GraphStep], maxVarId=4}
    },
    NeptuneTraverserConverterStep
]
+ not converted into Neptune steps: [NeptuneTailGlobalStep(100),
 NeptuneTinkerpopTraverserConverterStep, NeptuneSearchStep {
    JoinGroupNode {
        SearchNode[(idVar=?3, query=mark*, field=name) . project ask .], {endpoint=your-
OpenSearch-endpoint-URL}
    }
    JoinGroupNode {
        SearchNode[(idVar=?3, query=mark*, field=name) . project ask .], {endpoint=your-
OpenSearch-endpoint-URL}
    }
}]
```

下列範例是針對航線資料的 Gremlin 查詢：

小鬼基本不區分大小寫的match查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'match')
  .V().has("city","Neptune#fts dallas")

==>v[186]
==>v[8]
```

小鬼match查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'match')
  .V().has("city","Neptune#fts southampton")
     .local(values('code','city').fold())
     .limit(5)

==>[SOU, Southampton]
```

小鬼fuzzy查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .V().has("city","Neptune#fts allas~").values('city').limit(5)
```

```
==>Dallas
==>Dallas
==>Walla Walla
==>Velas
==>Altai
```

小鬼query_string模糊查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'query_string')
  .V().has("city","Neptune#fts allas~").values('city').limit(5)

==>Dallas
==>Dallas
```

小鬼query_string正則表達式查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'query_string')
  .V().has("city","Neptune#fts /[dp]allas/").values('city').limit(5)

==>Dallas
==>Dallas
```

小鬼混合查詢

這個查詢會在相同的查詢中使用 Neptune 內部 OpenSearch 索引和索引。

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .V().has("region","GB-ENG")
      .has('city','Neptune#fts L*')
      .values('city')
      .dedup()
      .limit(10)

==>London
==>Leeds
==>Liverpool
==>Land's End
```

簡單的小鬼全文檢索搜尋

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .V().has('desc','Neptune#fts regional municipal')
      .local(values('code','desc').fold())
      .limit(100)

==>[HYA, Barnstable Municipal Boardman Polando Field]
==>[SPS, Sheppard Air Force Base-Wichita Falls Municipal Airport]
==>[ABR, Aberdeen Regional Airport]
==>[SLK, Adirondack Regional Airport]
==>[BFD, Bradford Regional Airport]
```

```
==>[EAR, Kearney Regional Airport]
==>[ROT, Rotorua Regional Airport]
==>[YHD, Dryden Regional Airport]
==>[TEX, Telluride Regional Airport]
==>[WOL, Illawarra Regional Airport]
==>[TUP, Tupelo Regional Airport]
==>[COU, Columbia Regional Airport]
==>[MHK, Manhattan Regional Airport]
==>[BJI, Bemidji Regional Airport]
==>[HAS, Hail Regional Airport]
==>[ALO, Waterloo Regional Airport]
==>[SHV, Shreveport Regional Airport]
==>[ABI, Abilene Regional Airport]
==>[GIZ, Jizan Regional Airport]
==>[USA, Concord Regional Airport]
==>[JMS, Jamestown Regional Airport]
==>[COS, City of Colorado Springs Municipal Airport]
==>[PKB, Mid Ohio Valley Regional Airport]
```

使用query_string「+」和「-」運算符的小精靈查詢

雖然 query_string 查詢類型比預設的 simple_query_string 類型嚴格，但其確實能讓您執行較精確的查詢。下面的第一個查詢使用 query_string，第二個查詢使用預設的 simple_query_string：

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'query_string')
 . V().has('desc','Neptune#fts +London -(Stansted|Gatwick)')
      .local(values('code','desc').fold())
      .limit(10)

==>[LHR, London Heathrow]
==>[YXU, London Airport]
==>[LTN, London Luton Airport]
==>[SEN, London Southend Airport]
==>[LCY, London City Airport]
```

請注意在以下範例中，simple_query_string 如何不動聲色地忽略 '+' 和 '-' 運算子：

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .V().has('desc','Neptune#fts +London -(Stansted|Gatwick)')
      .local(values('code','desc').fold())
      .limit(10)

==>[LHR, London Heathrow]
==>[YXU, London Airport]
==>[LGW, London Gatwick]
==>[STN, London Stansted Airport]
==>[LTN, London Luton Airport]
==>[SEN, London Southend Airport]
==>[LCY, London City Airport]
==>[SKG, Thessaloniki Macedonia International Airport]
==>[ADB, Adnan Menderes International Airport]
==>[BTV, Burlington International Airport]
```

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'query_string')
  .V().has('desc','Neptune#fts +(regional|municipal) -(international|bradford)')
      .local(values('code','desc').fold())
```

```
      .limit(10)

==>[CZH, Corozal Municipal Airport]
==>[MMU, Morristown Municipal Airport]
==>[YBR, Brandon Municipal Airport]
==>[RDD, Redding Municipal Airport]
==>[VIS, Visalia Municipal Airport]
==>[AIA, Alliance Municipal Airport]
==>[CDR, Chadron Municipal Airport]
==>[CVN, Clovis Municipal Airport]
==>[SDY, Sidney Richland Municipal Airport]
==>[SGU, St George Municipal Airport]
```

與OR運營商的AND小鬼query_string查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'query_string')
  .V().has('desc','Neptune#fts (St AND George) OR (St AND Augustin)')
      .local(values('code','desc').fold())
      .limit(10)

==>[YIF, St Augustin Airport]
==>[STG, St George Airport]
==>[SGO, St George Airport]
==>[SGU, St George Municipal Airport]
```

小鬼term查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'term')
  .V().has("SKU","Neptune#fts ABC123DEF9")
      .local(values('code','city').fold())
      .limit(5)

==>[AUS, Austin]
```

小鬼prefix查詢

```
g.withSideEffect("Neptune#fts.endpoint",
                 "your-OpenSearch-endpoint-URL")
  .withSideEffect('Neptune#fts.queryType', 'prefix')
  .V().has("icao","Neptune#fts ka")
      .local(values('code','icao','city').fold())
      .limit(5)

==>[AZO, KAZO, Kalamazoo]
==>[APN, KAPN, Alpena]
==>[ACK, KACK, Nantucket]
==>[ALO, KALO, Waterloo]
==>[ABI, KABI, Abilene]
```

在 Neptune 小鬼中使用 Lucene 語法

在 Neptune Gemlin 中，您還可以使用 Lucene 查詢語法編寫非常強大的查詢。請注意，Lucene 語法僅支援中的query_string查詢 OpenSearch。

假設下列資料：

```
g.addV("person")
        .property(T.id, "p1")
        .property("name", "simone")
        .property("surname", "rondelli")

g.addV("person")
        .property(T.id, "p2")
        .property("name", "simone")
        .property("surname", "sengupta")

g.addV("developer")
        .property(T.id, "p3")
        .property("name", "simone")
        .property("surname", "rondelli")
```

使用 (queryType 為 query_string 時所呼叫的) Lucene 語法，即可按名字和姓氏搜尋此資料，如下所示：

```
g.withSideEffect("Neptune#fts.endpoint", "es_endpoint")
    .withSideEffect("Neptune#fts.queryType", "query_string")
    .V()
    .has("*", "Neptune#fts predicates.name.value:simone AND
 predicates.surname.value:rondelli")

==> v[p1], v[p3]
```

請注意，在上述 has() 步驟中，此欄位由 "*" 取代)。實際上，置於該處的任何值都由您在查詢中存取的欄位所覆寫。您可以使用 predicates.name.value, 存取名字欄位，因為這是資料模型的結構方式。

您可以按名字、姓氏和標籤進行搜尋，如下所示：

```
g.withSideEffect("Neptune#fts.endpoint", getEsEndpoint())
    .withSideEffect("Neptune#fts.queryType", "query_string")
    .V()
    .has("*", "Neptune#fts predicates.name.value:simone AND
 predicates.surname.value:rondelli AND entity_type:person")

==> v[p1]
```

標籤以 entity_type 加以存取，因為這同樣是資料模型的結構方式。

您也可以加入巢狀條件：

```
g.withSideEffect("Neptune#fts.endpoint", getEsEndpoint())
    .withSideEffect("Neptune#fts.queryType", "query_string")
    .V()
    .has("*", "Neptune#fts (predicates.name.value:simone AND
 predicates.surname.value:rondelli AND entity_type:person) OR
 predicates.surname.value:sengupta")

==> v[p1], v[p2]
```

插入一個現代的 TinkerPop 圖形

```
g.addV('person').property(T.id, '1').property('name', 'marko').property('age', 29)
 .addV('personr').property(T.id, '2').property('name', 'vadas').property('age', 27)
```

```
 .addV('software').property(T.id, '3').property('name', 'lop').property('lang', 'java')
 .addV('person').property(T.id, '4').property('name', 'josh').property('age', 32)
 .addV('software').property(T.id, '5').property('name', 'ripple').property('lang', 'java')
 .addV('person').property(T.id, '6').property('name', 'peter').property('age', 35)

g.V('1').as('a').V('2').as('b').addE('knows').from('a').to('b').property('weight',
 0.5f).property(T.id, '7')
 .V('1').as('a').V('3').as('b').addE('created').from('a').to('b').property('weight',
 0.4f).property(T.id, '9')
 .V('4').as('a').V('3').as('b').addE('created').from('a').to('b').property('weight',
 0.4f).property(T.id, '11')
 .V('4').as('a').V('5').as('b').addE('created').from('a').to('b').property('weight',
 1.0f).property(T.id, '10')
 .V('6').as('a').V('3').as('b').addE('created').from('a').to('b').property('weight',
 0.2f).property(T.id, '12')
 .V('1').as('a').V('4').as('b').addE('knows').from('a').to('b').property('weight',
 1.0f).property(T.id, '8')
```

按字符串字段值進行排序示例

```
g.withSideEffect("Neptune#fts.endpoint", "your-OpenSearch-endpoint-URL")
 .withSideEffect('Neptune#fts.queryType', 'query_string')
 .withSideEffect('Neptune#fts.sortOrder', 'asc')
 .withSideEffect('Neptune#fts.sortBy', 'name')
 .V().has('name', 'Neptune#fts marko OR vadas OR ripple')
```

按非字符串字段值進行排序示例

```
g.withSideEffect("Neptune#fts.endpoint", "your-OpenSearch-endpoint-URL")
 .withSideEffect('Neptune#fts.queryType', 'query_string')
 .withSideEffect('Neptune#fts.sortOrder', 'asc')
 .withSideEffect('Neptune#fts.sortBy', 'age.value')
 .V().has('name', 'Neptune#fts marko OR vadas OR ripple')
```

按 ID 字段值排序示例

```
g.withSideEffect("Neptune#fts.endpoint", "your-OpenSearch-endpoint-URL")
.withSideEffect('Neptune#fts.queryType', 'query_string')
.withSideEffect('Neptune#fts.sortOrder', 'asc')
.withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.entity_id')
.V().has('name', 'Neptune#fts marko OR vadas OR ripple')
```

依標籤欄位值排序範例

```
g.withSideEffect("Neptune#fts.endpoint", "your-OpenSearch-endpoint-URL")
 .withSideEffect('Neptune#fts.queryType', 'query_string')
 .withSideEffect('Neptune#fts.sortOrder', 'asc')
 .withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.entity_type')
 .V().has('name', 'Neptune#fts marko OR vadas OR ripple')
```

按document_type字段值排序示例

```
g.withSideEffect("Neptune#fts.endpoint", "your-OpenSearch-endpoint-URL")
 .withSideEffect('Neptune#fts.queryType', 'query_string')
```

```
.withSideEffect('Neptune#fts.sortOrder', 'asc')
.withSideEffect('Neptune#fts.sortBy', 'Neptune#fts.document_type')
.V().has('name', 'Neptune#fts marko OR vadas OR ripple')
```

排解海王星全文檢索搜尋

若要診斷從 Neptune 複寫到的問題OpenSearch，請參閱輪詢程式 Lambda 函數的CloudWatch記錄。這些記錄檔會提供從串流讀取的記錄數目，以及成功複寫到的記錄數目的詳細資料OpenSearch。

您也可以變更`LoggingLevel`環境變數，變更 Lambda 函數的記錄層級。

> **Note**
>
> `LoggingLevel`設為時`DEBUG`，您可以檢視其他詳細資料，例如丟棄的串流記錄以及每筆記錄被捨棄的原因，同時將資料StreamPoller從 Neptune 複寫到OpenSearch. 如果您發現缺少記錄，這可能很有用。

Neptune 串流取用者應用程式發佈兩個指標，也可協助您診斷問題：CloudWatch

- `StreamRecordsProcessed`— 應用程式每單位時間處理的記錄數。有助於追蹤應用程式執行速率。
- `StreamLagTime`— 目前時間與正在處理之串流記錄的認可時間之間的時差 (毫秒)。此指標會顯示消費者應用程式落後的程度。

此外，與複寫程序相關的所有指標都會公開在 CloudWatch 的儀表板中，與您使用 CloudWatch 範本執行個體化應用程式時所提供的 `ApplicationName` 同名。

您也可以選擇建立 CloudWatch 警示，每次輪詢失敗連續超過兩次即觸發。執行個體化應用程式時，透過將 `CreateCloudWatchAlarm` 欄位設為 `true` 以執行此作業。然後指定您希望觸發警示時收到通知的電子郵件地址。

故障診斷從串流讀取記錄時失敗的程序

如果程序在讀取串流的記錄時失敗，請確定下列事項：

- 已在叢集上啟用串流。
- 海王星串流端點格式正確：
  - 對於小精靈或開放密碼：`https://`*`your cluster endpoint`*`:`*`your cluster port`*`/propertygraph/stream`或它的別名，`https://`*`your cluster endpoint`*`:`*`your cluster port`*`/pg/stream`
  - SPARQL：`https://`*`your cluster endpoint`*`:`*`your cluster port`*`/sparql/stream`
- 已針對您的虛擬私人雲端設定 DynamoDB 端點。
- 已針對您的 VPC 子網路設定監控端點。

疑難排解寫入資料時失敗的程序 OpenSearch

如果處理程序在寫入記錄時失敗OpenSearch，請確定您具備下列項目：

- 您的彈性搜索版本為 7.1 或更高版本。
- OpenSearch可以從 VPC 中的輪詢器 Lambda 函數進行存取。
- 附加到的安全性原則OpenSearch允許輸入 HTTP/HTTPS 要求。

使用 Amazon Neptune 的AWS Lambda功能

AWS Lambda功能在 Amazon Neptune 應用中有很多用途。在這裡，我們提供將 Lambda 函數與任何常見的 Gemlin 驅動程式和語言變體搭配使用的一般指引，以及以 Java JavaScript、和 Python 撰寫的 Lambda 函數的特定範例。

Note

在 Neptune 中使用 Lambda 函數的最佳方式已隨著最近的引擎版本而有所改變。Neptune 曾經在 Lambda 執行內容回收之後很長時間保持閒置連線開啟，這可能導致伺服器上的資源洩漏。為了減輕這種情況，我們過去建議在每次 Lambda 叫用時開啟和關閉連線。不過，從引擎版本 1.0.3.0 開始，閒置連線逾時已降低，因此在回收非作用中的 Lambda 執行內容後，連線不會再洩漏，因此我們建議您在執行內容期間使用單一連線。這應該包括一些錯誤處理和 back-off-and-retry 樣板代碼，以處理意外關閉的連接。

管理AWS Lambda函數中的小鬼 WebSocket 連接

如果您使用 Gremlin 語言變體來查詢 Neptune，則驅動程式會使用連 WebSocket 線連線到資料庫。WebSockets 旨在支持長期使用的客戶端-服務器連接方案。 AWS Lambda另一方面，旨在支持相對短暫和無狀態的執行。使用 Lambda 查詢 Neptune 時，設計理念上的這種不相符可能會導致一些未預期的問題。

一個AWS Lambda函數在一個執行上下文中運行，該上下文將函數與其他函數隔離。執行內容會在第一次叫用函數時建立，並且可以在相同函數的後續叫用時重複使用。

但是，任何一個執行上下文都不會用於處理函數的多個並發調用。如果多個用戶端同時叫用您的函數，Lambda 會為函數的每個執行個體提供額外的執行內容。所有這些新的執行上下文可以依次被重複使用，用於後續的函數調用。

在某些時候，Lambda 會回收執行上下文，特別是如果它們已經處於非活動狀態一段時間。 AWS Lambda透過 Lambda 擴充功能公開執行內容生命週期，包括`Invoke`和`Shutdown`階段。`Init`您可以使用這些延伸模組撰寫程式碼，以便在回收執行內容時清除外部資源，例如資料庫連線。

常見的最佳做法是在 Lambda 處理常式函數之外開啟資料庫連線，以便在每個處理常式呼叫時重複使用。如果資料庫連線在某個時候中斷，您可以從處理常式內部重新連線。但是，使用此方法存在連接洩漏的危險。如果閒置連線在執行內容銷毀後長時間保持開啟狀態，則間歇性或突發性 Lambda 叫用案例可能會逐漸洩漏連線並耗盡資料庫資源。

Neptune 連線限制和連線逾時已隨著較新的引擎版本而變更。以前，每個執行個體最多支援 60,000 個 WebSocket連線。現在，每個 Neptune 執行個體的最大並行 WebSocket 連線數目會隨執行個體類型而有所不同。

此外，Neptune 從引擎版本 1.0.3.0 開始，將連線的閒置逾時時間從一小時縮減至約 20 分鐘。如果用戶端未關閉連線，連線會在 20 到 25 分鐘的閒置逾時後自動關閉。 AWS Lambda不會記錄執行內容的生命週期，但實驗顯示新的 Neptune 連線逾時與非作用中的 Lambda 執行內容逾時一致。回收非使用中執行內容時，其連線很有可能已經被 Neptune 關閉，或者很快就會關閉。

AWS Lambda與 Amazon Neptune 小鬼一起使用的建議

現在，我們建議在 Lambda 執行內容的整個生命週期中使用單一連線和圖形遍歷來源，而不是每個函數叫用一個（每個函數調用僅處理一個客戶端請求）。由於並發客戶端請求是由在單獨執行內容中運行的不同函數

實例處理的，因此不需要維護連接池來處理函數實例內的並發請求。如果您使用的 Gemlin 驅動程式有連線池，請將其設定為只使用一個連線。

若要處理連線失敗，請在每個查詢周圍使用重試邏輯。即使目標是在執行內容的生命週期內維護單一連線，但未預期的網路事件可能會導致該連線突然終止。這種連接失敗顯示為不同的錯誤，具體取決於您使用的驅動程序。您應該撰寫 Lambda 函數的程式碼來處理這些連線問題，並在必要時嘗試重新連線。

一些 Gimlin 驅動程序會自動處理重新連接。例如，Java 驅動程式會自動嘗試代表您的用戶端程式碼重新建立與 Neptune 的連線。使用此驅動程序，您的函數代碼只需要退回並重試查詢。相比之下， JavaScript 和 Python 驅動程序不會實現任何自動重新連接邏輯，因此使用這些驅動程序，您的函數代碼必須在備份後嘗試重新連接，並且僅在重新建立連接後重新嘗試查詢。

這裡的代碼示例確實包括重新連接邏輯，而不是假設客戶端正在處理它。

在 Lambda 中使用小精靈寫入請求的建議

如果您的 Lambda 函數修改圖形資料，請考慮採用 back-off-and-retry策略來處理下列例外狀況：

- **`ConcurrentModificationException`**-Neptune 事務語義意味著寫入請求有時會失敗，並顯示`ConcurrentModificationException`. 在這些情況下，請嘗試指數 back-off-based 重試機制。
- **`ReadOnlyViolationException`**— 由於叢集拓撲隨時可能因計劃或未計劃的事件而變更，因此寫入責任可能會從叢集中的一個執行個體遷移到另一個執行個體。如果您的函數程式碼嘗試將寫入要求傳送至不再是主要 (寫入器) 執行個體的執行個體，則要求會失敗，並顯示`ReadOnlyViolationException`. 發生這種情況時，請關閉現有連線、重新連線至叢集端點，然後重試要求。

此外，如果您使用 back-off-and-retry 策略來處理寫入請求問題，請考慮為創建和更新請求實施冪等查詢（例如，使用 fold () .coalesce () .展開/)。

在 Lambda 中使用小鬼讀取要求的建議

如果您的叢集中有一或多個僅供讀取複本，最好在這些複本之間平衡讀取要求。一種選擇是使用讀取器端點 (p. 121)。讀取器端點會平衡複本之間的連線，即使叢集拓撲在您新增或移除複本時發生變更，或將複本升級為新的主要執行個體時也一樣。

但是，在某些情況下，使用讀取器端點可能會導致叢集資源的使用不均勻。讀取器端點的運作方式是定期變更 DNS 入口所指向的主機。如果用戶端在 DNS 項目變更之前開啟了許多連線，則所有連線要求都會傳送至單一 Neptune 執行個體。在高輸送量 Lambda 案例中可能會發生這種情況，其中對 Lambda 函數的大量並行請求會導致建立多個執行內容，每個內容都有自己的連線。如果這些連線幾乎同時建立，則連線很可能全部指向叢集中的相同複本，並保持指向該複本，直到回收執行內容為止。

在執行個體之間散發請求的一種方式是，將 Lambda 函數設定為連接至執行個體端點 (從複本執行個體端點清單中隨機選擇)，而不是從讀取器端點清單中選擇的執行個體端點。這種方法的缺點是，它需要 Lambda 程式碼來處理叢集拓撲中的變更，方法是監控叢集並在叢集的成員資格變更時更新端點清單。

如果您正在編寫一個 Java Lambda 函數，需要平衡叢集中執行個體之間的讀取請求，您可以使用 Amazon Neptune 的 Gremlin 用戶端，這是一個知道叢集拓撲的 Java Gemlin 用戶端，並且可以在 Neptune 叢集中的一組執行個體公平分配連線和請求。這篇部落格文章包含一個範例 Java Lambda 函數，該函數會在 Amazon Neptune 使用小鬼用戶端。

可能會減緩 Neptune 小鬼 Lambda 函數冷啟動的因素

第一次呼叫AWS Lambda函數時稱為冷啟動。有幾個因素可能會增加冷啟動的延遲：

- 請務必為您 Lambda 函數配置到足夠的記憶體。 — 對於 Lambda 函數，冷啟動期間的編譯速度可能會比 EC2 上的速度明顯慢，因為按照指派給函數的記憶體成比例，線性AWS Lambda分配 CPU 週期。在 1,792 MB，函數等於一個完整 vCPU (每秒一個 vCPU 秒的額度)。對於以 Java 編寫的大型 Lambda 函數來說，未指派足夠記憶體來接收足夠 CPU 週期的影響尤其明顯。

- 請注意，啟用 IAM 資料庫身份驗證 (p. 160)可能會降低冷啟動 —AWS Identity and Access Management (IAM) 資料庫身份驗證的速度也會降低冷啟動速度，尤其是在 Lambda 函數必須產生新的簽署金鑰時。此延遲只會影響冷啟動，而不會影響後續請求，因為一旦 IAM DB 驗證建立連線登入資料，Neptune 只會定期驗證它們是否仍然有效。

用於建立AWS CloudFormation要在海王星中使用的 Lambda 函數

您可以使用AWS CloudFormation範本來建立可存取 Neptune 的AWS Lambda函數。

1. 若要在AWS CloudFormation主控台上啟動 Lambda 函數堆疊，請選擇下表中的其中一個「啟動堆疊」按鈕。

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (雪梨)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國東部)	檢視	在設計工具中檢視	Launch Stack

2. 在 Select Template (選取範本) 頁面上，請選擇 Next (下一步)。
3. 在 Specify Details (指定詳細資訊) 頁面上，設定下列選項：
   a. 根據您要在 Lambda 函數中使用的語言，選擇 Lambda 執行階段。這些 AWS CloudFormation 範本目前支援下列語言：
      - 蟒 3.6 (映射到`python36`亞馬遜 S3 網址)
      - 節點 18 (映射到亞馬遜 S3 網址`nodejs18x`中)
      - 紅寶石 2.5 (映射到`ruby25`亞馬遜 S3 網址)
   b. 提供適當的 Neptune 叢集端點和連接埠號碼。
   c. 提供適當的海王星安全群組。
   d. 提供適當的 Neptune 子網路參數。
4. 選擇 下一步。
5. 在 Options (選項) 頁面上，選擇 Next (下一步)。
6. 在 [檢閱] 頁面上，選取第一個核取方塊，以確認AWS CloudFormation將建立 IAM 資源。

   然後選擇 Create (建立)。

如果您需要對 Lambda 執行階段進行自己的變更，可以從區域中的 Amazon S3 位置下載一般執行階段：

```
https://s3.Amazon region.amazonaws.com/aws-neptune-customer-samples-Amazon region/
lambda/runtime-language/lambda_function.zip.
```

例如：

```
https://s3.us-west-2.amazonaws.com/aws-neptune-customer-samples-us-west-2/lambda/python36/
lambda_function.zip
```

AWS LambdaAmazon Neptune 的範例

以下使用 Java JavaScript 和 Python 編寫的示例AWS Lambda函數說明了使用該`fold().coalesce().unfold()`習語隨機生成的 ID 來提升單個頂點。

每個函數中的大部分代碼都是樣板代碼，負責管理連接並在發生錯誤時重試連接和查詢。真正的應用程序邏輯和 Gemlin 查詢分別在`doQuery()`和`query()`方法中實現。如果您使用這些範例做為自己的 Lambda 函數的基礎，則可以專注於修改`doQuery()`和`query()`。

這些函數設定為重試失敗的查詢 5 次，在重試之間等候 1 秒鐘。

這些函數需要值存在於下列 Lambda 環境變數中：

- **`NEPTUNE_ENDPOINT`**— 您的 Neptune 資料庫叢集端點。對於 Python 來說，這應該是`neptuneEndpoint`。
- **`NEPTUNE_PORT`**-Neptune 端口 對於 Python 來說，這應該是`neptunePort`。
- **`USE_IAM`** — (`true`或`false`) 如果您的資料庫已啟用AWS Identity and Access Management (IAM) 資料庫驗證，請將`USE_IAM`環境變數設定為`true`。這會導致 Lambda 函數向 Neptune 簽署 4-簽署連接請求。對於此類 IAM 資料庫身份驗證請求，請確保 Lambda 函數的執行角色具有附加適當的 IAM 政策，以允許函數連接到 Neptune 資料庫叢集 (請參閱IAM 政策的類型 (p. 177))。

Amazon Neptune 的 Java Lambda 函數示例

以下是 JavaAWS Lambda 函數記得幾件事：

- Java 驅動程序維護自己的連接池，您不需要這些連接池，因此請使用`minConnectionPoolSize(1)`和配置`Cluster`對象`maxConnectionPoolSize(1)`。
- 該`Cluster`對象的構建速度可能很慢，因為它創建了一個或多個序列化器（默認情況下是 Gyro，再加上另一個，如果您已將其配置為其他輸出格式，例如`binary`）。此範例需要一些時間。
- 連接池與第一個請求初始化。此時，如果您使用 IAM DB 驗證，驅動程序會設置`Netty`堆棧，分配字節緩衝區並創建簽名密鑰。所有這些都可以增加冷啟動延遲。
- Java 驅動程式的連線集區會監視伺服器主機的可用性，並在連線失敗時自動嘗試重新連線。它會啟動背景工作以嘗試重新建立連線。用`reconnectInterval( )`於設定重新連線嘗試之間的間隔。當驅動程式嘗試重新連線時，您的 Lambda 函數只需重試查詢即可。

  如果重試之間的間隔小於重新連線嘗試之間的間隔，則失敗的連線重試會再次失敗，因為主機被視為無法使用。這不適用於`ConcurrentModificationException`.
- 使用 Java 8 而不是 Java 11。 `Netty`在 Java 11 中，預設情況下不會啟用最佳化。
- 此範例使用重試 4j 進行重試。
- 若要在 Java Lambda 函數中使用`Sigv4`簽署驅動程式，請參閱使用具有簽名版本 4 簽名的 Java 和小鬼連接到海王星 (p. 165).

  Warning

  來`CallExecutor`自 Retry4j 的可能不是執行緒安全的。考慮讓每個線程使用自己的`CallExecutor`實例。

```
package com.amazonaws.examples.social;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestStreamHandler;
```

```
import com.evanlennick.retry4j.CallExecutor;
import com.evanlennick.retry4j.CallExecutorBuilder;
import com.evanlennick.retry4j.Status;
import com.evanlennick.retry4j.config.RetryConfig;
import com.evanlennick.retry4j.config.RetryConfigBuilder;
import org.apache.tinkerpop.gremlin.driver.Cluster;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.neptune.auth.NeptuneNettyHttpSigV4Signer;
import org.apache.tinkerpop.gremlin.driver.remote.DriverRemoteConnection;
import org.apache.tinkerpop.gremlin.driver.ser.Serializers;
import org.apache.tinkerpop.gremlin.process.traversal.AnonymousTraversalSource;
import org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.GraphTraversalSource;
import org.apache.tinkerpop.gremlin.structure.T;

import java.io.*;
import java.time.temporal.ChronoUnit;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
import java.util.concurrent.Callable;
import java.util.function.Function;

import static java.nio.charset.StandardCharsets.UTF_8;
import static org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.__.addV;
import static org.apache.tinkerpop.gremlin.process.traversal.dsl.graph.__.unfold;

public class MyHandler implements RequestStreamHandler {

  private final GraphTraversalSource g;
  private final CallExecutor<Object> executor;
  private final Random idGenerator = new Random();

  public MyHandler() {

    this.g = AnonymousTraversalSource
            .traversal()
            .withRemote(DriverRemoteConnection.using(createCluster()));


    this.executor = new CallExecutorBuilder<Object>()
            .config(createRetryConfig())
            .build();

  }

  @Override
  public void handleRequest(InputStream input,
                            OutputStream output,
                            Context context) throws IOException {

    doQuery(input, output);
  }

  private void doQuery(InputStream input, OutputStream output) throws IOException {
    try {

      Map<String, Object> args = new HashMap<>();
      args.put("id", idGenerator.nextInt());

      String result = query(args);

      try (Writer writer = new BufferedWriter(new OutputStreamWriter(output, UTF_8))) {
          writer.write(result);
      }

    } finally {
```

```
      input.close();
      output.close();
    }
  }

  private String query(Map<String, Object> args) {
    int id = (int) args.get("id");

    @SuppressWarnings("unchecked")
    Callable<Object> query = () -> g.V(id)
      .fold()
      .coalesce(
        unfold(),
        addV("Person").property(T.id, id))
      .id().next();

    Status<Object> status = executor.execute(query);

    return status.getResult().toString();
  }

  private Cluster createCluster() {
    Cluster.Builder builder = Cluster.build()
                                     .addContactPoint(System.getenv("NEPTUNE_ENDPOINT"))
                                     .port(Integer.parseInt(System.getenv("NEPTUNE_PORT")))
                                     .enableSsl(true)
                                     .minConnectionPoolSize(1)
                                     .maxConnectionPoolSize(1)
                                     .serializer(Serializers.GRAPHBINARY_V1D0)
                                     .reconnectInterval(2000);

  if (Boolean.parseBoolean(getOptionalEnv("USE_IAM", "true"))) {
    // For versions of TinkerPop 3.4.11 or higher:
    builder.handshakeInterceptor( r ->
      {
        NeptuneNettyHttpSigV4Signer sigV4Signer = new NeptuneNettyHttpSigV4Signer(region,
new DefaultAWSCredentialsProviderChain());
        sigV4Signer.signRequest(r);
        return r;
      }
    )

    // Versions of TinkerPop prior to 3.4.11 should use the following approach.
    // Be sure to adjust the  imports to include:
    //   import org.apache.tinkerpop.gremlin.driver.SigV4WebSocketChannelizer;
    //   builder = builder.channelizer(SigV4WebSocketChannelizer.class);

    return builder.create();
  }

  private RetryConfig createRetryConfig() {
    return new RetryConfigBuilder().retryOnCustomExceptionLogic(retryLogic())
                                   .withDelayBetweenTries(1000, ChronoUnit.MILLIS)
                                   .withMaxNumberOfTries(5)
                                   .withFixedBackoff()
                                   .build();
  }

  private Function<Exception, Boolean> retryLogic() {
    return e -> {
      StringWriter stringWriter = new StringWriter();
      e.printStackTrace(new PrintWriter(stringWriter));
      String message = stringWriter.toString();

      // Check for connection issues
      if ( message.contains("Timed out while waiting for an available host") ||
```

```
            message.contains("Timed-out waiting for connection on Host") ||
            message.contains("Connection to server is no longer active") ||
            message.contains("Connection reset by peer") ||
            message.contains("SSLEngine closed already") ||
            message.contains("Pool is shutdown") ||
            message.contains("ExtendedClosedChannelException") ||
            message.contains("Broken pipe")) {
          return true;
        }

        // Concurrent writes can sometimes trigger a ConcurrentModificationException.
        // In these circumstances you may want to backoff and retry.
        if (message.contains("ConcurrentModificationException")) {
            return true;
        }

        // If the primary fails over to a new instance, existing connections to the old
 primary will
        // throw a ReadOnlyViolationException. You may want to back and retry.
        if (message.contains("ReadOnlyViolationException")) {
            return true;
        }

        return false;
      };
    }

    private String getOptionalEnv(String name, String defaultValue) {
      String value = System.getenv(name);
      if (value != null && value.length() > 0) {
        return value;
      } else {
        return defaultValue;
      }
    }
}
```

如果要在函數中包含重新連接邏輯，請參閱Java 重新連接示例 (p. 311)。

JavaScript Amazon Neptune Lambda 函數示例

此範例的說明

- JavaScript 驅動程式不會維護連線集區。它總是打開一個連接。
- 範例函數會使用 gremlin-aws-sigv4 的 Sigv4 簽署公用程式，將要求簽署至啟用 IAM 驗證的資料庫。
- 它會使用開放原始碼非同步公用程式模組中的 retry () 函數來處理 backoff-and-retry 嘗試。
- Grimlin 終端步驟返回一個 JavaScript `promise`（請參閱TinkerPop 文檔）。對於`next()`，這是一個`{value, done}`元組。
- 連接錯誤在處理程序內部引發，並根據此處概述的建議使用一些 backoff-and-retry 邏輯進行處理，但有一個例外。有一種連接問題是驅動程序不會被視為異常，因此無法通過此 backoff-and-retry 邏輯容納。

  問題是，如果在驅動程序發送請求之後但在驅動程序收到響應之前關閉連接，則查詢似乎已完成，但返回 null 值。就 lambda 函數客戶端而言，該函數似乎成功完成，但響應為空。

  此問題的影響取決於您的應用程式如何處理空白回應。某些應用程式可能會將讀取要求的空白回應視為錯誤，但其他應用程式可能會將其視為空白結果。

  遇到此連線問題的寫入要求也會傳回空白回應。空響應的成功調用是否表示成功或失敗？ 如果調用 write 函數的客戶端只是將函數的成功調用視為意味著已提交對數據庫的寫入，而不是檢查響應的主體，則系統可能會丟失數據。

此問題是由驅動程式如何處理底層通訊端發出的事件所導致的。當基礎網絡套接字因ECONNRESET錯誤而關閉時，驅動程序 WebSocket 使用的關閉並發出'ws close'事件。但是，驅動程序中沒有任何內容可以用來引發異常的方式處理該事件。因此，查詢就會消失。

若要解決此問題，此處的 Lambda 函數範例會新增'ws close'事件處理常式，在建立遠端連線時將例外狀況擲回驅動程式。不過，這個例外並非沿著 Gemlin 查詢的要求-回應路徑引發，因此無法用來觸發 lambda 函數本身內的任何 backoff-and-retry 邏輯。'ws close'事件處理常式所擲回的例外狀況會導致未處理的例外狀況，導致 lambda 叫用失敗。這允許調用函數的客戶端處理錯誤，並在適當的情況下重試 lambda 調用。

我們建議您在 lambda 函數本身中實作 backoff-and-retry 邏輯，以保護您的用戶端免受間歇性連線問題的影響。但是，針對上述問題的因應措施也需要用戶端實作重試邏輯，以處理因此特定連線問題所造成的失敗。

程式碼

```
const gremlin = require('gremlin');
const async = require('async');
const {getUrlAndHeaders} = require('gremlin-aws-sigv4/lib/utils');

const traversal = gremlin.process.AnonymousTraversalSource.traversal;
const DriverRemoteConnection = gremlin.driver.DriverRemoteConnection;
const t = gremlin.process.t;
const __ = gremlin.process.statics;

let conn = null;
let g = null;

async function query(context) {

  const id = context.id;

  return g.V(id)
    .fold()
    .coalesce(
      __.unfold(),
      __.addV('User').property(t.id, id)
    )
    .id().next();
}

async function doQuery() {
  const id = Math.floor(Math.random() * 10000).toString();

  let result = await query({id: id});
  return result['value'];
}

exports.handler = async (event, context) => {

  const getConnectionDetails = () => {
    if (process.env['USE_IAM'] == 'true'){
       return getUrlAndHeaders(
         process.env['NEPTUNE_ENDPOINT'],
         process.env['NEPTUNE_PORT'],
         {},
         '/gremlin',
         'wss');
    } else {
      const database_url = 'wss://' + process.env['NEPTUNE_ENDPOINT'] + ':' +
 process.env['NEPTUNE_PORT'] + '/gremlin';
```

```
    return { url: database_url, headers: {}};
  }
};


const createRemoteConnection = () => {
  const { url, headers } = getConnectionDetails();

  const c = new DriverRemoteConnection(
    url,
    {
      mimeType: 'application/vnd.gremlin-v2.0+json',
      headers: headers
    });

   c._client._connection.on('close', (code, message) => {
       console.info(`close - ${code} ${message}`);
       if (code == 1006){
         console.error('Connection closed prematurely');
         throw new Error('Connection closed prematurely');
       }
     });

   return c;
};

const createGraphTraversalSource = (conn) => {
  return traversal().withRemote(conn);
};

if (conn == null){
  console.info("Initializing connection")
  conn = createRemoteConnection();
  g = createGraphTraversalSource(conn);
}

return async.retry(
  {
    times: 5,
    interval: 1000,
    errorFilter: function (err) {

      // Add filters here to determine whether error can be retried
      console.warn('Determining whether retriable error: ' + err.message);

      // Check for connection issues
      if (err.message.startsWith('WebSocket is not open')){
        console.warn('Reopening connection');
        conn.close();
        conn = createRemoteConnection();
        g = createGraphTraversalSource(conn);
        return true;
      }

      // Check for ConcurrentModificationException
      if (err.message.includes('ConcurrentModificationException')){
        console.warn('Retrying query because of ConcurrentModificationException');
        return true;
      }

      // Check for ReadOnlyViolationException
      if (err.message.includes('ReadOnlyViolationException')){
        console.warn('Retrying query because of ReadOnlyViolationException');
        return true;
      }
```

```
        return false;
      }

    },
    doQuery);
};
```

Amazon Neptune 的 Python Lambda 函數示例

以下是有關以下 PythonAWS Lambda 示例函數的一些注意事項：

- 它使用輪詢模塊。
- 它設置pool_size=1為防止創建不必要的連接池。
- 它設置message_serializer=serializer.GraphSONSerializersV2d0()。

```
import os, sys, backoff, math
from random import randint
from gremlin_python import statics
from gremlin_python.driver.driver_remote_connection import DriverRemoteConnection
from gremlin_python.driver.protocol import GremlinServerError
from gremlin_python.driver import serializer
from gremlin_python.process.anonymous_traversal import traversal
from gremlin_python.process.graph_traversal import __
from gremlin_python.process.strategies import *
from gremlin_python.process.traversal import T
from aiohttp.client_exceptions import ClientConnectorError
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
from botocore.credentials import ReadOnlyCredentials
from types import SimpleNamespace

import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)


reconnectable_err_msgs = [
    'ReadOnlyViolationException',
    'Server disconnected',
    'Connection refused',
    'Connection was already closed',
    'Connection was closed by server',
    'Failed to connect to server: HTTP Error code 403 - Forbidden'
]

retriable_err_msgs = ['ConcurrentModificationException'] + reconnectable_err_msgs

network_errors = [OSError, ClientConnectorError]

retriable_errors = [GremlinServerError, RuntimeError, Exception] + network_errors

def prepare_iamdb_request(database_url):

    service = 'neptune-db'
    method = 'GET'

    access_key = os.environ['AWS_ACCESS_KEY_ID']
    secret_key = os.environ['AWS_SECRET_ACCESS_KEY']
    region = os.environ['AWS_REGION']
    session_token = os.environ['AWS_SESSION_TOKEN']
```

```
    creds = SimpleNamespace(
        access_key=access_key, secret_key=secret_key, token=session_token, region=region,
    )

    request = AWSRequest(method=method, url=database_url, data=None)
    SigV4Auth(creds, service, region).add_auth(request)

    return (database_url, request.headers.items())

def is_retriable_error(e):

    is_retriable = False
    err_msg = str(e)

    if isinstance(e, tuple(network_errors)):
        is_retriable = True
    else:
        is_retriable = any(retriable_err_msg in err_msg for retriable_err_msg in
 retriable_err_msgs)

    logger.error('error: [{}] {}'.format(type(e), err_msg))
    logger.info('is_retriable: {}'.format(is_retriable))

    return is_retriable

def is_non_retriable_error(e):
    return not is_retriable_error(e)

def reset_connection_if_connection_issue(params):

    is_reconnectable = False

    e = sys.exc_info()[1]
    err_msg = str(e)

    if isinstance(e, tuple(network_errors)):
        is_reconnectable = True
    else:
        is_reconnectable = any(reconnectable_err_msg in err_msg for reconnectable_err_msg
 in reconnectable_err_msgs)

    logger.info('is_reconnectable: {}'.format(is_reconnectable))

    if is_reconnectable:
        global conn
        global g
        conn.close()
        conn = create_remote_connection()
        g = create_graph_traversal_source(conn)

@backoff.on_exception(backoff.constant,
    tuple(retriable_errors),
    max_tries=5,
    jitter=None,
    giveup=is_non_retriable_error,
    on_backoff=reset_connection_if_connection_issue,
    interval=1)
def query(**kwargs):

    id = kwargs['id']

    return (g.V(id)
        .fold()
        .coalesce(
            __.unfold(),
            __.addV('User').property(T.id, id)
```

```
        )
        .id().next())

def doQuery(event):
    return query(id=str(randint(0, 10000)))

def lambda_handler(event, context):
    result = doQuery(event)
    logger.info('result - {}'.format(result))
    return result

def create_graph_traversal_source(conn):
    return traversal().withRemote(conn)

def create_remote_connection():
    logger.info('Creating remote connection')

    (database_url, headers) = connection_info()

    return DriverRemoteConnection(
        database_url,
        'g',
        pool_size=1,
        message_serializer=serializer.GraphSONSerializersV2d0(),
        headers=headers)

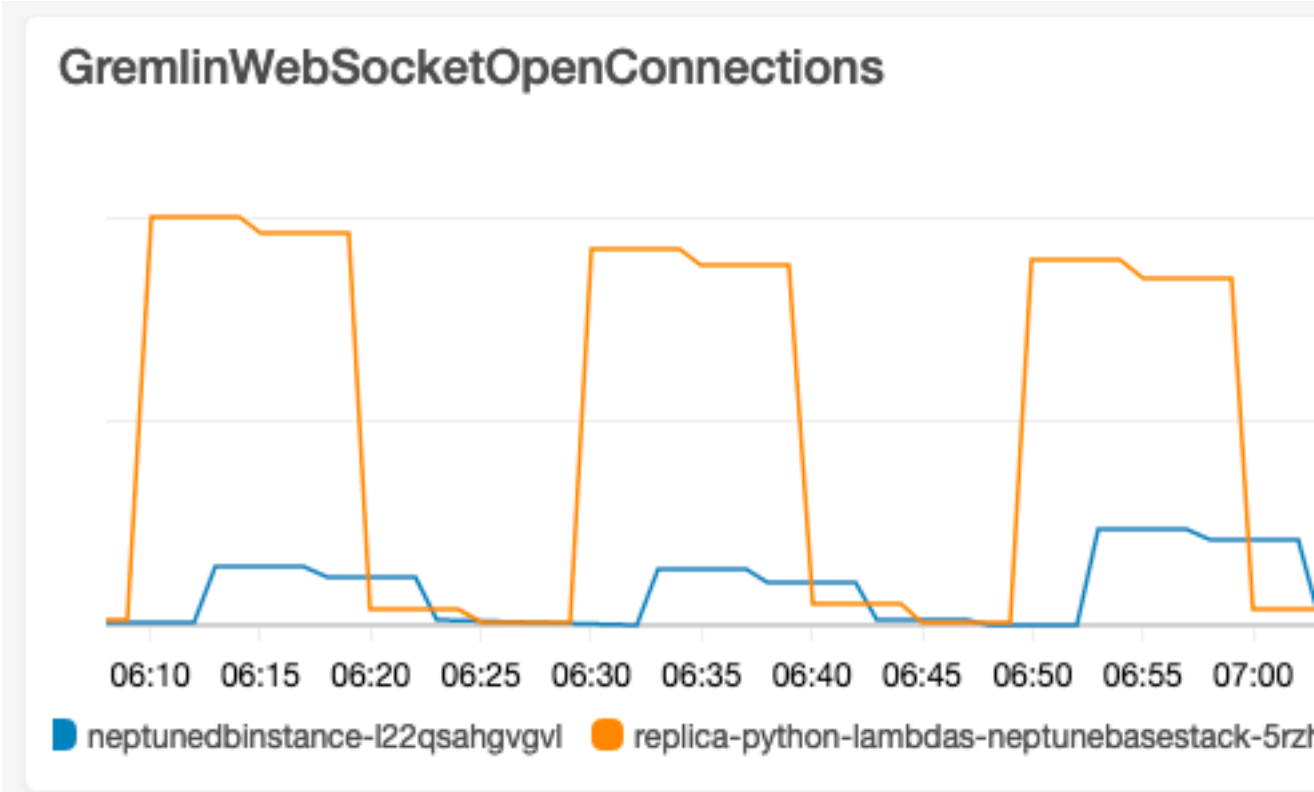
def connection_info():

    database_url = 'wss://{}:{}/gremlin'.format(os.environ['neptuneEndpoint'],
 os.environ['neptunePort'])

    if 'USE_IAM' in os.environ and os.environ['USE_IAM'] == 'true':
        return prepare_iamdb_request(database_url)
    else:
        return (database_url, {})

conn = create_remote_connection()
g = create_graph_traversal_source(conn)
```

以下是示例結果，顯示重負載和輕負載的交替週期：


GremlinWebSocketOpenConnections
06:10
06:15
06:20
06:25
06:30
06:35
06:40
06:45
06:50
06:55
07:00
neptunedbinstance-l22qsahgvgvl
replica-python-lambdas-neptunebasestack-5rzh


Lambda-ConcurrentExecutions
Count
86
43.5
1
06:10
06:15
06:20
06:25
06:30
06:35
06:40
06:45
06:50
06:55
social-b6501310_add-follows-relationship social-b6501310_add-follows-relationship
social-b6501310_create-user social-b6501310_create-user
social-b6501310_get-recent-follows social-b6501310_get-recent-follows
social-b6501310_get-user-details social-b6501310_get-user-details


Workload
Various units
2.17k

Amazon Neptune ML 用於圖表上的機器學習

在大型連接的數據集中通常存在有價值的信息，這些信息很難僅基於人類的直覺使用查詢來提取。機器學習 (ML) 技術可協助在具有數十億個關係的圖表中找出隱藏的關聯性。這些相關性可能有助於推薦產品，預測信用價值，識別欺詐和許多其他事情。

Neptune ML 功能可讓您在數小時而不需要數週的時間內，在大型圖表上建置和訓練有用的機器學習模型。為了實現這一目標，Neptune ML 使用由亞馬遜 SageMaker和深圖庫（DGL）（開源）提供支持的圖形神經網絡（GNN）技術。圖形神經網絡是人工智能的新興領域（例如，參見圖形神經網絡的綜合調查）。如需將 GNN 與 DGL 搭配使用的實作教學課程，請參閱使用深度圖庫學習圖形神經網路。

> **Note**
>
> 圖頂點在 Neptune ML 模型標識為「節點」。例如，頂點分類使用節點分類機器學習模型，而頂點迴歸則使用節點迴歸模型。

Neptune ML 可以做什麼

Neptune 同時支援傳感性推論 (根據當時的圖形資料傳回訓練時預先計算的預測) 和感應式推論 (傳回) 會根據目前的資料，即時套用資料處理和模型評估。請參閱 感應推斷和傳感推斷之間的區別 (p. 667)。

Neptune ML 可訓練機器學習模型，以支援五種不同類別的推論：

Neptune ML 目前支援的推論工作類型

- 節點分類 — 預測頂點屬性的分類特徵。

  例如, 鑑於電影肖申克的救贖, Neptune ML 可以`story`從候選集預測其`genre`屬性`[story, crime, action, fantasy, drama, family, ...]`.

  兩種兩種兩種兩種兩種兩種的節點分類工作：
  - 單一類別分類：在此類任務中，每個節點只有一個目標功能。例如，的屬`Place_of_birth`性`Alan Turing`具有值`UK`。
  - 多類別分類：在這種任務中，每個節點可以有多個目標功能。例如，電影教父*genre*的屬性具有價值觀`crime`和`story`.
- 節點迴歸 — 預測頂點的數值屬性。

  例如，鑑於電影復仇者：終局之戰，Neptune ML 可以預測其屬性`popularity`具有的值`5.0`。
- 邊緣分類 — 預測邊屬性的分類特徵。

  兩種兩種兩種兩種兩種兩種的邊分
  - 單一類別分類：在此類任務中，每個邊只有一個目標圖徵。例如，使用者和影片之間的分級邊緣可能具有屬性`liked`，其值為「是」或「否」。
  - 多類別分類：在此類任務中，每個邊緣可以有多個目標圖徵。例如，使用者和影片之間的分級可能會有多個屬性標籤值，例如「搞笑」、「心思心意」、「令人不寒」等。
- 邊迴歸 — 預測邊的數值性質。

  例如，使用者和電影之間的評分邊緣可能具有數值屬性`score`，Neptune ML 可以預測給定使用者和電影的值。

- 連結預測 — 預測特定來源節點和傳出邊緣的最可能的目標節點，或是指定目標節點和傳入邊緣的最有可能的來源節點。

  例如，用藥物疾病知識圖，給出`Aspirin`作為源節點，並`treats`作為傳出邊緣，Neptune ML 可以預測最相關的目的地節點為`heart diseasefever`，等等。

  或者，使用維基媒體知識圖，`President-of`作為邊緣或關係，並`United-States`作為目標節點，Neptune ML 可以預測最相關的頭為`George WashingtonAbraham LincolnFranklin D. Roosevelt`，，等等。

  > **Note**
  >
  > 節點分類和邊分類僅支援字串值。這意味著數字屬性值（例如`0`或）不`1`受支持，儘管字符串等效`"0"`和`"1"`是。同樣，布爾屬性值`true`和`false`不起作用，但`"true"`和`"false"`做。

使用 Neptune ML，您可以使用分為兩大類的機器學習模型：

Neptune ML 目前支援的機器學習模型類型

- 圖形神經網絡（GNN）模型-其中包括關係圖卷積網絡（R-GCNS）。GNN 模型適用於上述所有三種類型的任務。
- 知識圖嵌入 (KGE) 模型 — 這些包括`TransE`,`DistMult`, 和`RotatE`模型. 它們僅適用於鏈接預測。

使用者定義的模型 — Neptune ML 也可讓您為上述所有類型的工作提供您自己的自訂模型實作。在搭配模型使用 Neptune ML 訓練 API 之前，您可以使用 Neptune ML 工具組來開發和測試以 Python 為基礎的自訂模型實作。如Neptune ML 中的自定義模型 (p. 724)需如何建構和組織實作，以便與 Neptune ML 的訓練基礎結構相容的詳細資訊，請參閱。

設定 Neptune ML DEME

開始使用 Neptune ML 的最簡單方式是使用AWS CloudFormation快速入門範本 (p. 654)。此範本會安裝所有必要的元件，包括新的 Neptune DB 叢集、所有必要的 IAM 角色，以及新的 Neptune 圖形筆記本，讓使用 Neptune ML 變得更容易。

您也可以手動安裝 Neptune ML，如中所述在不使用快速入門AWS CloudFormation範本的情況下設定 Neptune ML (p. 657)。

使用 Neptune ML AWS CloudFormation 範本在新的資料庫叢集中快速開始使用

開始使用 Neptune BAL的最簡單方式是使用AWS CloudFormation快速啟動範本。此範本會安裝所有必要的元件，包括新的 Neptune DB 叢集、所有必要的 IAM 角色，以及新的 Neptune 圖形筆記本，讓使用 Neptune ML 變得更容易。

若要建立 Neptune ML 快速啟動堆疊

1. 若要在AWS CloudFormation主控台上啟動AWS CloudFormation堆疊，請選擇下表中的其中一個「啟動堆疊」按鈕：

區域	檢視	在設計工具中檢視	啟動
美國東部 (維吉尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國東部 (俄亥俄)	檢視	在設計工具中檢視	Launch Stack
美國西部 (加利佛尼亞北部)	檢視	在設計工具中檢視	Launch Stack
美國西部 (奧勒岡)	檢視	在設計工具中檢視	Launch Stack
加拿大 (中部)	檢視	在設計工具中檢視	Launch Stack
南美洲 (聖保羅)	檢視	在設計工具中檢視	Launch Stack
Europe (Stockholm)	檢視	在設計工具中檢視	Launch Stack
歐洲 (愛爾蘭)	檢視	在設計工具中檢視	Launch Stack
歐洲 (倫敦)	檢視	在設計工具中檢視	Launch Stack
Europe (Paris)	檢視	在設計工具中檢視	Launch Stack
歐洲 (法蘭克福)	檢視	在設計工具中檢視	Launch Stack
Middle East (Bahrain)	檢視	在設計工具中檢視	Launch Stack
中東 (阿拉伯聯合大公國)	檢視	在設計工具中檢視	Launch Stack
非洲 (開普敦)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (香港)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (東京)	檢視	在設計工具中檢視	Launch Stack

區域	檢視	在設計工具中檢視	啟動
亞太區域 (首爾)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (新加坡)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (雪梨)	檢視	在設計工具中檢視	Launch Stack
亞太區域 (孟買)	檢視	在設計工具中檢視	Launch Stack
中國 (北京)	檢視	在設計工具中檢視	Launch Stack
中國 (寧夏)	檢視	在設計工具中檢視	Launch Stack
AWSGovCloud(美國西部)	檢視	在設計工具中檢視	Launch Stack

2. 在 Select Template (選取範本) 頁面上，請選擇 Next (下一步)。
3. 在 [指定詳細資訊] 頁面上，選擇 [下一步
4. 在 Options (選項) 頁面上，選擇 Next (下一步)。
5. 在「檢閱」頁面上，您需要勾選兩個核取方塊：

   - 第一個承認AWS CloudFormation可能會使用自定義名稱創建 IAM 資源。
   - 第二個承認AWS CloudFormation可能需要新堆棧的CAPABILITY_AUTO_EXPAND功能。CAPABILITY_AUTO_EXPAND明確AWS CloudFormation允許在建立堆疊時自動展開巨集，而無需事先檢閱。

     客戶通常會從已處理的範本建立變更集，以便在實際建立堆疊之前先檢閱巨集所做的變更。如需詳細資訊，請參閱 AWS CloudFormationCreateStack API。

   然後選擇 Create (建立)。

快速入門範本會建立並設定下列項目：

- Neptune 資料庫叢集。
- 必要的 IAM 角色（並附加它們）。
- 必要的 Amazon EC2 安全群組。
- 必要的 SageMaker VPC 端點。
- Neptune ML 的資料庫叢集參數群組。
- 該參數群組中的必要參數。
- 具有預先填入 Neptune ML 筆記本樣本的筆記本電腦。SageMaker請注意，並非每個區域都提供所有執行個體大小，因此您必須確定選取的筆記型電腦執行個體大小是您所在地區支援的大小。
- 海王星-出口服務。

當快速啟動堆疊準備就緒時，請前往範本建立的SageMaker筆記本，並查看預先填入的範例。他們將協助您下載範例資料集，以用於實驗 Neptune ML 功能。

當您使用 Neptune ML 時，它們還可以為您節省大量時間。例如，請參閱% 海王星 _ 毫升 (p. 43)線條魔術，以及這些筆記本支持的% 海王星 _ 毫升 (p. 46)細胞魔術。

您也可以使用下列AWS CLI命令來執行快速入門AWS CloudFormation範本：

```
aws cloudformation create-stack \
  --stack-name neptune-ml-fullstack-$(date '+%Y-%m-%d-%H-%M') \
  --template-url https://aws-neptune-customer-samples.s3.amazonaws.com/v2/cloudformation-
templates/neptune-ml-nested-stack.json \
  --parameters ParameterKey=EnableIAMAuthOnExportAPI,ParameterValue=(true if you have IAM
 auth enabled, or false otherwise) \
               ParameterKey=Env,ParameterValue=test$(date '+%H%M')\
  --capabilities CAPABILITY_IAM \
  --region (the AWS region, like us-east-1) \
  --disable-rollback \
  --profile (optionally, a named CLI profile of yours)
```

在不使用快速入門AWS CloudFormation範本的情況下設定 Neptune ML

1. 從運作中的 Neptune 資料庫叢集開始

如果您不使用AWS CloudFormation快速啟動範本來設定 Neptune ML，則需要現有的 Neptune 資料庫叢集才能使用。如果需要，您可以使用已有的，或複製您已經使用的一個，或者您可以建立新的 (請參閱建立資料庫叢集 (p. 54))。

請確定您將使用的 Neptune DB 叢集至少執行引擎版本`1.0.5.0`。如果它運行的是較早的引擎版本，則可以按照中的說明對其進行升級海王星引擎更新 (p. 857)。

2. 安裝海王星-出口服務

若您尚未在中解釋，請先完成安裝 NepNepNepNepNepNepNepNepNeptune-出口服務使用 Neptune-出口服務導出 Neptune 數據 (p. 491)。

使用下列設定，將輸入規則新增至安裝所建立的安`NeptuneExportSecurityGroup`全性群組：

- Type (類型)：`Custom TCP`
- Protocol (通訊協定)：`TCP`
- 連接埠範圍：`80 - 443`
- 來源：*`(Neptune ##############)`*

3. 建立自訂NeptuneLoadFromS3 IAM 角色

如果您尚未在中說明，請先完成建立自訂`NeptuneLoadFromS3` IAM 角色，請先完成建立建立 IAM 角色以存取亞馬遜 S3 (p. 257)。

建立自訂NeptuneSageMakerIAMRole角色

使用 IAM 主控台建立自訂內容`NeptuneSageMakerIAMRole`，使用下列政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:CreateVpcEndpoint",
        "ec2:DeleteNetworkInterface",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
```

```
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "sagemaker.amazonaws.com"
          ]
        }
      },
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:CreateGrant",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "arn:aws:kms:*:*:key/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "logs:GetLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/sagemaker/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "sagemaker:AddTags",
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateHyperParameterTuningJob",
        "sagemaker:CreateModel",
        "sagemaker:CreateProcessingJob",
        "sagemaker:CreateTrainingJob",
        "sagemaker:CreateTransformJob",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteModel",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeHyperParameterTuningJob",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeProcessingJob",
```

```
        "sagemaker:DescribeTrainingJob",
        "sagemaker:DescribeTransformJob",
        "sagemaker:InvokeEndpoint",
        "sagemaker:ListTags",
        "sagemaker:ListTrainingJobsForHyperParameterTuningJob",
        "sagemaker:StopHyperParameterTuningJob",
        "sagemaker:StopProcessingJob",
        "sagemaker:StopTrainingJob",
        "sagemaker:StopTransformJob",
        "sagemaker:UpdateEndpoint",
        "sagemaker:UpdateEndpointWeightsAndCapacities"
      ],
      "Resource": [
        "arn:aws:sagemaker:*:*:*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "sagemaker:ListEndpointConfigs",
        "sagemaker:ListEndpoints",
        "sagemaker:ListHyperParameterTuningJobs",
        "sagemaker:ListModels",
        "sagemaker:ListProcessingJobs",
        "sagemaker:ListTrainingJobs",
        "sagemaker:ListTransformJobs"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

建立此角色時，請編輯信任關係，使其內容如下：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ec2.amazonaws.com",
          "rds.amazonaws.com",
          "sagemaker.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

最後，複製分配給這個新NeptuneSageMakerIAMRole角色的 ARN。

Important

- 請確保 Amazon S3 許可中的許可與上面的許可NeptuneSageMakerIAMRole相匹配。
- 在上述政策中arn:aws:s3:::*，通用 ARN 用於 Amazon S3 資源。如果由於某種原因無法使用通用 ARNarn:aws:s3:::graphlytics*，則必須將 Neptune EML 命令將使用的任何其他客戶 Amazon S3 資源的 ARN 添加到資源部分。

設定資料庫叢集以啟用 Neptune

若您的資料庫叢集，設定 Neptune

1. 在 Neptune 主控台中，瀏覽至參數群組，然後瀏覽至與您要使用的資料庫叢集相關聯的資料庫叢集參數群組。將neptune_ml_iam_role參數設定為指派給您剛建立之NeptuneSageMakerIAMRole角色的 ARN。
2. 瀏覽至 [資料庫]，然後選取要用於 Neptune ML 的資料庫叢集。選取動作，然後選取管理 IAM 角色。
3. 在「管理 IAM 角色」頁面上，選取新增角色，然後新增NeptuneSageMakerIAMRole. 然後新增NeptuneLoadFromS3角色。
4. 重新啟動資料庫叢集寫入器執行寫入器執行

在您的 Neptune VPC 中建立兩個 SageMaker 端點

最後，若要讓 Neptune 引擎存取必要的 SageMaker 管理 API，您需要在 Neptune VPC 中建立兩個 SageMaker 端點，如中所述 SageMaker 在您的 Neptune VPC 中建立兩個端點 (p. 663)。

為 Neptune ML 手動設定 Neptune 筆記本

Neptune SageMaker 筆記本預裝了各種用於 Neptune ML 的樣本筆記本。您可以在開放原始碼圖形筆記本 GitHub 儲存庫中預覽這些範例。

您可以使用其中一個現有的 Neptune 筆記本，或者您可以依照中的指示建立自己的筆記本使用 Neptune 工作台託管海王星筆記本 (p. 31)。

您也可以依照下列步驟設定預設 Neptune 筆記本以搭配 Neptune ML 使用：

修改 Neptune ML 的筆記本

1. 在以下位置打開亞馬遜 SageMaker 控制台 https://console.aws.amazon.com/sagemaker/。
2. 在左側的導覽窗格中，選擇記事本，然後選擇筆記本執行。尋找您要用於 Neptune ML 的 Neptune 筆記本名稱，然後選取該名稱以前往其詳細資料頁面。
3. 如果記事本執行個體正在執行，請選取筆記本詳細資料頁面右上方的 [停止] 按鈕。
4. 在筆記本執行個體設定中的生命週期組態下，選取連結以開啟筆記本生命週期的頁面。
5. 選取右上角的「編輯」，然後選取「繼續」。
6. 在 [開始筆記本] 索引標籤中，修改指令碼以包含其他匯出命令，並填入 Neptune ML IAM 角色和匯出服務 URI 的欄位，如下所示，視您的殼層而定：

```
echo "export NEPTUNE_ML_ROLE_ARN=(your Neptune ML IAM role ARN)" >> ~/.bashrc
echo "export NEPTUNE_EXPORT_API_URI=(your export service URI)" >> ~/.bashrc
```

7. 選取 Update (更新)。
8. 返回筆記本執行個體頁面。在「權限和加密」下，存在 IAM 角色 ARN 欄位。選取此欄位中的連結，即可前往此筆記本執行個體所搭配執行的 IAM 角色。
9. 創建一個新的內聯策略，如下所示：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "arn:aws:cloudwatch:[AWS_REGION]:[AWS_ACCOUNT_ID]:*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Resource": "arn:aws:logs:[AWS_REGION]:[AWS_ACCOUNT_ID]:*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:Put*",
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "arn:aws:s3:::*",
      "Effect": "Allow"
    },
    {
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:[AWS_REGION]:[AWS_ACCOUNT_ID]:*/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "sagemaker:CreateModel",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:CreateEndpoint",
        "sagemaker:DescribeModel",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DeleteModel",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DeleteEndpoint"
      ],
      "Resource": "arn:aws:sagemaker:[AWS_REGION]:[AWS_ACCOUNT_ID]:*/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "[YOUR_NEPTUNE_ML_IAM_ROLE_ARN]"",
      "Effect": "Allow"
    }
  ]
}
```

10. 儲存此新政策並將其附加至步驟 8 中的 IAM 角色。
11. 選取 SageMaker 筆記本執行個體詳細資訊頁面右上角的 [啟動] 以啟動筆記本執行個體。

使用在AWS CLI資料庫叢集上設定 Neptune ML

除了AWS CloudFormation快速啟動範本和AWS Management Console，您也可 Neptune 使用AWS CLI.

1. 針對新的 Neptune ML 叢集建立資料庫叢集參數群組

下列AWS CLI命令會建立新的資料庫叢集參數群組，並將其設定為與 Neptune ML 搭配使用：

建立和設定 Neptune ML 的資料庫叢集參數群組

1. 建立新的資料庫叢集參數群組：

```
aws neptune create-db-cluster-parameter-group \
  --db-cluster-parameter-group-name (name of the new DB cluster parameter group) \
  --db-parameter-group-family neptune1
  --description "(description of your machine learning project)" \
  --region (AWS region, such as us-east-1)
```

2. 建立設定為 ARN 的neptune_ml_iam_role資料庫叢集參數，以SageMakerExcecutionIAMRole供資料庫叢集在呼叫 SageMaker 建立作業並從託管 ML 模型取得預測時使用：

```
aws neptune modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name (name of the new DB cluster parameter group) \
  --parameters "ParameterName=neptune_ml_iam_role, \
                ParameterValue=ARN of the SageMakerExcecutionIAMRole, \
                Description=NeptuneMLRole, \
                ApplyMethod=pending-reboot" \
  --region (AWS region, such as us-east-1)
```

   設定此參數可讓 Neptune 存取， SageMaker 而不必在每次呼叫時傳遞角色。

   若要取得有關如何建立的資訊SageMakerExcecutionIAMRole，請參閱建立自訂NeptuneSageMakerIAMRole角色 (p. 657)。

3. 最後，使用describe-db-cluster-parameters來檢查新資料庫叢集參數群組中的所有參數是否按照您希望的方式進行設定：

```
aws neptune describe-db-cluster-parameters \
  --db-cluster-parameter-group-name (name of the new DB cluster parameter group) \
  --region (AWS region, such as us-east-1)
```

將新的資料庫叢集參數群組附加至您將搭配 Neptune ML 使用的資料庫叢集附加至您

現在，您可以使用下列命令，將剛建立的新資料庫叢集參數群組附加至現有的資料庫叢集：

```
aws neptune modify-db-cluster \
  --db-cluster-identifier (the name of your existing DB cluster) \
  --apply-immediately
  --db-cluster-parameter-group-name (name of your new DB cluster parameter group) \
  --region (AWS region, such as us-east-1)
```

若要使所有參數生效，您可以接著重新啟動資料庫叢集：

```
aws neptune reboot-db-instance
```

```
  --db-instance-identifier (name of the primary instance of your DB cluster) \
  --profile (name of your AWS profile to use) \
  --region (AWS region, such as us-east-1)
```

或者，如果您要建立新的資料庫叢集以搭配 Neptune ML 使用，您可以使用下列命令來建立附加新參數群組的叢集，然後建立新的主要 (寫入器) 執行個體：

```
cluster-name=(the name of the new DB cluster)
aws neptune create-db-cluster
  --db-cluster-identifier ${cluster-name}
  --engine graphdb \
  --engine-version 1.0.4.1 \
  --db-cluster-parameter-group-name (name of your new DB cluster parameter group) \
  --db-subnet-group-name (name of the subnet to use) \
  --region (AWS region, such as us-east-1)

aws neptune create-db-instance
  --db-cluster-identifier ${cluster-name}
  --db-instance-identifier ${cluster-name}-i \
  --db-instance-class (the instance class to use, such as db.r5.xlarge)
  --engine graphdb \
  --region (AWS region, such as us-east-1)
```

將連接NeptuneSageMakerIAMRole到您的資料庫叢集，以便它可以存取 SageMaker 和 Amazon S3 資源

最後，請按照中建立自訂NeptuneSageMakerIAMRole角色 (p. 657)的說明建立 IAM 角色，讓您的資料庫叢集 SageMaker 與 Amazon S3 通訊。然後，使用以下命令將您創建的NeptuneSageMakerIAMRole角色附加到數據庫叢集：

```
aws neptune add-role-to-db-cluster
  --db-cluster-identifier ${cluster-name}
  --role-arn arn:aws:iam::(the ARN number of the role's ARN):role/NeptuneMLRole \
  --region (AWS region, such as us-east-1)
```

SageMaker 在您的 Neptune VPC 中建立兩個端點

Neptune ML 在您的 Neptune 資料庫叢集的 VPC 中需要兩個 SageMaker 端點：

- com.amazonaws.*(AWS region, like us-east-1)*.sagemaker.runtime
- com.amazonaws.*(AWS region, like us-east-1)*.sagemaker.api

如果您尚未使用會自動為您建立這些快速入門AWS CloudFormation範本，您可以使用下列AWS CLI指令來建立它們：

這個創建sagemaker.runtime端點：

```
create-vpc-endpoint
  --vpc-id (the ID of your Neptune DB cluster's VPC)
  --service-name com.amazonaws.(AWS region, like us-east-1).sagemaker.runtime
  --subnet-ids (the subnet ID or IDs that you want to use)
  --security-group-ids (the security group for the endpoint network interface, or omit to
 use the default)
  --private-dns-enabled
```

這個創建sagemaker.api端點：

```
aws create-vpc-endpoint
  --vpc-id (the ID of your Neptune DB cluster's VPC)
  --service-name com.amazonaws.(AWS region, like us-east-1).sagemaker.api
  --subnet-ids (the subnet ID or IDs that you want to use)
  --security-group-ids (the security group for the endpoint network interface, or omit to
 use the default)
  --private-dns-enabled
```

您也可以使用 VPC 主控台建立這些端點。請參閱亞馬遜中的安全預測呼叫，AWS PrivateLink並 SageMaker使用保護所有 Amazon SageMaker API 呼叫的安全AWS PrivateLink。

在資料庫叢集參數群組中建立資料庫叢集參數參數 SageMaker

若要避免必須在對模型 SageMaker 進行的每個查詢中指定正在使用的模型的推論端點，請為 Neptune ML 建立資料庫叢集參數群組`neptune_ml_endpoint`中名為的資料庫叢集參數參數。將參數設定為有問題`id`的執行個體端點。

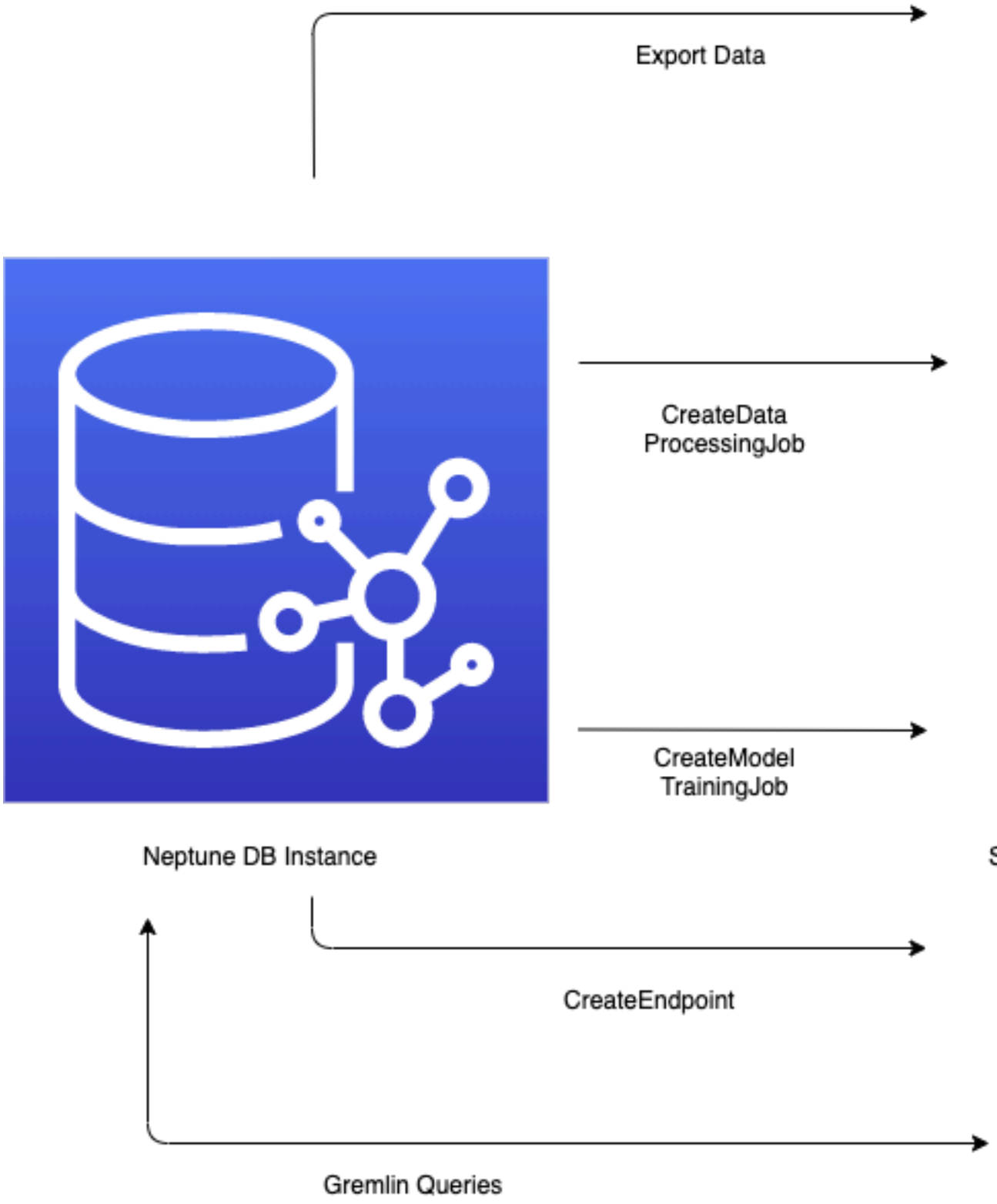
您可以使用下列AWS CLI命令執行此動作：

```
aws neptune modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name neptune-ml-demo \
  --parameters "ParameterName=neptune_ml_endpoint, \
                ParameterValue=(the name of the SageMaker inference endpoint you want to
 query), \
                Description=NeptuneMLEndpoint, \
                ApplyMethod=pending-reboot" \
  --region (AWS region, such as us-east-1)
```

如何使用 Neptune ML 功能概觀

啟動使用 Neptune ML 的工作流程

在亞馬遜 Neptune 中使用海王星 ML 功能通常涉及以下五個步驟：


Export Data
CreateData
ProcessingJob
CreateModel
TrainingJob
Neptune DB Instance
CreateEndpoint
Gremlin Queries

1. 資料匯出和組態 — 資料匯出步驟使用 Neptune-匯出服務或`neptune-export`命令列工具，以 CSV 格式將資料從 Neptune 匯出至 Simple Storage Service (Amazon S3)。名`training-data-configuration.json`為的組態檔案會同時自動產生，指定如何將匯出的資料載入至可訓練圖形中。
2. 資料預先處理 — 在此步驟中，匯出的資料集會使用標準技術預先處理，以準備進行模型訓練。可針對數值資料執行特徵標準化，文字特徵可以使用編碼`word2vec`。在此步驟結束時，會從匯出的資料集產生 DGL (深度圖表程式庫) 圖形，供模型訓練步驟使用。

   此步驟是使用帳戶中的 SageMaker 處理任務來實作，產生的資料會存放在您指定的 Amazon S3 位置。
3. 模型訓練 — 模型訓練步驟會訓練將用於預測的機器學習模型。

   模型訓練分兩個階段完成：

   - 第一階段使用 SageMaker 處理工作來產生模型訓練策略組態集，指定模型訓練將使用哪種類型的模型和模型超參數範圍。
   - 然後，第二個階段會使用 SageMaker 模型調整工作來嘗試不同的超參數組態，並選取產生效能最佳模型的訓練工作。調整工作會針對已處理的資料執行預先指定數量的模型訓練工作試驗。在此階段結束時，會使用最佳訓練工作的訓練模型參數來產生用於推論的模型加工品。
4. 在 Amazon 中建立推論端點 SageMaker — 推論端點是使用最佳訓練任務產生的模型成品啟動的 SageMaker 端點執行個體。每個模型都繫結至單一端點。端點能夠接受來自圖形資料庫的傳入要求，並傳回要求中輸入的模型預測。建立端點後，它會保持作用中狀態，直到您將其刪除為止。
5. 使用 Gremlin 查詢機器學習模型 — 您可以使用 Gremlin 查詢語言的擴充功能，從推論端點查詢預測。

   Note

   Neptune 工作台 (p. 31)包含一個線條魔術和細胞魔術，可以為您節省大量時間管理這些步驟，即：

   - % 海王星 _ 毫升 (p. 43)
   - % 海王星 _ 毫升 (p. 46)

根據不斷發展的圖形數據進行預測

隨著不斷變化的圖形，您可能希望使用新的數據定期創建新的批次預測。查詢預先計算的預測（傳感推斷）比根據最新數據（感應推斷）即時生成新預測要快得多。這兩種方法都有自己的位置，具體取決於資料變更的速度和效能需求。

感應推斷和傳感推斷之間的區別

執行傳感性推論時，Neptune 會查詢並傳回訓練時預先計算的預測。

當執行感應推斷，Neptune 構造相關的子圖並獲取其屬性。然後，DGL GNN 模型即時套用資料處理和模型評估。

因此，感應推論可以產生預測，包括節點和邊緣，這些節點和邊緣在訓練時不存在，並反映圖形的當前狀態。但是，這是以更高的延遲為代價。

如果您的圖形是動態的，您可能需要使用感應推斷來確保考慮到最新的數據，但是如果您的圖形是靜態的，則換能推斷更快，更有效率。

預設為停用。您可以在查詢中使用 GremlinNeptune #ml. 誘導推論 (p. 734) 謂詞啟用它的查詢，如下所示：

```
.with( "Neptune#ml.inductiveInference")
```

增量改進式工作流程

雖然您只需重新執行步驟一到三 (從資料匯出和設定到模型轉換) 即可更新模型成品，Neptune ML 支援使用新資料更新批次 ML 預測的更簡單方法。一種是使用增量模型工作流程 (p. 668)，另一種是使用模型重新訓練與熱啟動 (p. 668)。

增量模型工作流程

在此工作流程中，您可以更新 ML 預測，而不需要重新訓練 ML 模型。

> Note
>
> 只有當圖形資料已經使用新的節點和/或邊緣更新時，您才能執行此操作。當節點被刪除時，它目前不起作用。

1. 資料匯出和配置 — 此步驟與主工作流程中的步驟相同。
2. 增量資料預先處理 — 此步驟類似於主工作流程中的資料預處理步驟，但使用與特定訓練模型相同的處理配置。
3. 模型轉換 — 此模型轉換步驟不是模型訓練步驟，而是從主工作流程和增量資料預處理步驟的結果中取得訓練過的模型，並產生用於推論的新模型加工品。模型轉換步驟會啟動 SageMaker 處理工作，以執行產生更新模型人工因素的計算。
4. 更新 Amazon SageMaker 推論端點 — 如果您有現有的推論端點，此步驟會使用模型轉換步驟產生的新模型成品來更新端點。或者，您也可以使用新模型人工因素建立新的推論端點。

溫暖的開始模型再培訓

使用此工作流程，您可以訓練和部署新的 ML 模型，以使用增量圖形資料進行預測，但是從使用主工作流程產生的現有模型開始：

1. 資料匯出和配置 — 此步驟與主工作流程中的步驟相同。
2. 增量資料預先處理 — 此步驟與增量模型推論工作流程中的步驟相同。新的圖形資料應該使用先前用於模型訓練的相同處理方法來處理。
3. 以溫暖開始進行模型訓練 — 模型訓練類似於主要工作流程中發生的情況，但您可以利用先前模型訓練任務中的資訊來加速模型超參數搜尋。
4. 更新 Amazon SageMaker 推論端點 — 此步驟與增量模型推論工作流程中的步驟相同。

Neptune ML 中自訂模型的工作流程

Neptune ML 可讓您針對 Neptune ML 支援的任何工作實作、訓練和部署您自己的自訂模型。開發和部署自訂模型的工作流程基本上與內建模型相同，但有一些差異，如中所述自訂模型工作流 (p. 725)。

Neptune ML 階段的執行個體選擇

Neptune ML 處理的不同階段使用不同的 SageMaker 執行個體。在這裡，我們討論如何針對每個階段選擇正確執行個體類型。您可以透過 Amazon SageMaker 定價找到 SageMaker 執行個體類型和定價的相關資訊。

選取資料處理的執行個體

資 SageMaker 料處理 (p. 694)步驟需要具有足夠記憶體和磁碟儲存空間供輸入、中繼和輸出資料使用的處理執行個體。所需的特定記憶體和磁碟儲存容量取決於 Neptune ML 圖形的特性及其匯出的功能。

根據預設，Neptune ML 會選擇記憶體比磁碟上匯出圖形資料大小十倍大的最小`ml.r5`執行個體。

選取用於模型訓練和模型轉換的執行個體

為模型訓練或模型轉換選取正確的執行個體類型，取決於工作類型、圖形大小以及您的周轉需求。GPU 執行個體可提供最佳效能。我們通常建議`p3`和`g4dn`串行實例。您也可以使用`p2`或`p4d`實例。

根據預設，Neptune ML 會選擇比模型訓練和模型轉換所需記憶體更多的最小 GPU 執行個體。您可以在 Amazon S3 資料處理輸出位置找到`train_instance_recommendation.json`檔案中的選取項目。下面是一個`train_instance_recommendation.json`文件的內容的例子：

```
{
  "instance":     "(the recommended instance type for model training and transform)",
  "cpu_instance": "(the recommended instance type for base processing instance)",
  "disk_size":    "(the estimated disk space required)",
  "mem_size":     "(the estimated memory required)"
}
```

選取推論端點的執行個體

為推論端點 (p. 730)選擇正確的執行個體類型，取決於任務類型、圖形大小和預算。根據預設，Neptune ML 會選擇推論端點所需記憶體更多的最小`ml.m5d`執行個體。

> Note
>
> 如果需要超過 384 GB 的記憶體，Neptune ML 會使用`ml.r5d.24xlarge`執行個體。

您可以在用於模型訓練的 Amazon S3 位置的`infer_instance_recommendation.json`檔案中查看 Neptune ML 建議的執行個體類型。以下是該檔案的範例：

```
{
  "instance" :   "(the recommended instance type for an inference endpoint)",
  "disk_size" :  "(the estimated disk space required)",
  "mem_size" :   "(the estimated memory required)"
}
```

使用工neptune-export具或 Neptune-導出服務從 Neptune 導出數據，用於 Neptune ML

Neptune ML 需要您提供深度圖庫 (DGL) 的訓練資料，以建立和評估模型。

您可以使用 Neptune -出口服務或neptune-export實用程序 (p. 496)從 Neptune 導出 (p. 491)數據。該服務和命令列工具都以 CSV 格式發佈資料到 Amazon Simple Storage Service (Amazon S3)，並使用 Amazon S3 伺服器端加密 (SSE-S3) 加密。請參閱 由海王星導出的文件-導出和neptune-export (p. 499)。

此外，當您設定 Neptune ML 訓練資料的匯出時，匯出工作會建立並發佈加密的模型訓練組態檔案以及匯出的資料。依預設，此檔案會命名為`training-data-configuration.json`。

使用 Neptune-匯出服務匯出 Neptune ML 訓練資料的範例

此請求會匯出節點分類任務的屬性圖訓練資料：

```
curl \
  (your NeptuneExportApiUri) \
  -X POST \
  -H 'Content-Type: application/json' \
  -d '{
        "command": "export-pg",
        "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
        "params": {
          "endpoint": "(your Neptune endpoint DNS name)",
          "profile": "neptune_ml"
        },
        "additionalParams": {
          "neptune_ml": {
            "version": "v2.0",
            "targets": [
              {
                "node": "Movie",
                "property": "genre",
                "type": "classification"
              }
            ]
          }
        }
      }'
```

此請求會匯出節點分類任務的 RDF 訓練資料：

```
curl \
  (your NeptuneExportApiUri) \
  -X POST \
  -H 'Content-Type: application/json' \
  -d '{
        "command": "export-rdf",
        "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
        "params": {
          "endpoint": "(your Neptune endpoint DNS name)",
          "profile": "neptune_ml"
        },
        "additionalParams": {
          "neptune_ml": {
            "version": "v2.0",
```

```
            "targets": [
              {
                "node": "http://aws.amazon.com/neptune/csv2rdf/class/Movie",
                "predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/
genre",
                "type": "classification"
              }
            ]
          }
        }
      }'
```

匯出訓練資料時要在params物件中設定的欄位

匯出請求中的params物件可以包含各種欄位，如params文 (p. 503)件中所述。下列項目與匯出機器學習訓練資料最相關：

- **endpoint**— 用於指endpoint定資料庫叢集中 Neptune 執行個體的端點，匯出程序可以查詢以擷取資料。
- **profile**— 必須將params物件中的profile欄位設定為**neptune-ml**。

  這會導致匯出程序針對 Neptune ML 模型訓練適當地格式化匯出的資料、屬性圖資料的 CSV 格式或 RDF 資料的 N-三倍。它也會建立training-data-configuration.json檔案，並將其寫入與匯出的訓練資料相同的 Amazon S3 位置。
- **cloneCluster**— 如果設定為true，匯出程序會複製您的資料庫叢集、從複製中匯出，然後在完成時刪除複製。
- **useIamAuth**— 如果您的資料庫叢集已啟用 IAM 身份驗證 (p. 160)，則必須將此欄位設定為true。

匯出程序還提供了幾種篩選匯出資料的方法 (請參閱這些範例 (p. 510))。

使用additionalParams物件來調整模型訓練資訊的匯出

additionalParams物件包含欄位，您可以使用這些欄位來指定機器學習類別標籤和用於訓練目的的功能，以及指導建立訓練資料組態檔案。

匯出程序無法自動推斷哪些節點和邊緣屬性應該是機器學習類別標籤，以作為訓練用途的範例。它也無法自動推斷數值、分類和文字屬性的最佳功能編碼，因此您需要使用additionalParams物件中的欄位提供提示來指定這些項目，或覆寫預設編碼。

對於屬性圖資料，匯出要求additionalParams中的頂層結構可能如下所示：

```
{
  "command": "export-pg",
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "profile": "neptune_ml"
  },
  "additionalParams": {
      "neptune_ml": {
        "version": "v2.0",
        "targets": [ (an array of node and edge class label targets) ],
        "features": [ (an array of node feature hints) ]
    }
  }
```

```
}
```

對於 RDF 數據，其頂級結構可能如下所示：

```
{
  "command": "export-rdf",
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "profile": "neptune_ml"
  },
  "additionalParams": {
      "neptune_ml": {
        "version": "v2.0",
        "targets": [ (an array of node and edge class label targets) ]
    }
  }
}
```

您也可以使用下列jobs欄位來提供多種匯出組態：

```
{
  "command": "export-pg",
  "outputS3Path": "s3://(your Amazon S3 bucket)/neptune-export",
  "params": {
    "endpoint": "(your Neptune endpoint DNS name)",
    "profile": "neptune_ml"
  },
  "additionalParams" : {
    "neptune_ml" : {
      "version": "v2.0",
      "jobs": [
        {
          "name" : "(training data configuration name)",
          "targets": [ (an array of node and edge class label targets) ],
          "features": [ (an array of node feature hints) ]
        },
        {
          "name" : "(another training data configuration name)",
          "targets": [ (an array of node and edge class label targets) ],
          "features": [ (an array of node feature hints) ]
        }
      ]
    }
  }
}
```

neptune_ml欄位中的頂層元素additionalParams

中的version元素neptune_ml

指定要產生的訓練資料組態版本。

(選擇性)，類型：字串，預設值：「v2.0」。

如果確實包含version，請將其設定為v2.0。

中的jobs欄位neptune_ml

包含訓練資料配置物件的陣列，每個物件都定義資料處理工作，並包含：

- **name**— 要建立的訓練資料組態名稱。

  例如，名稱為「job-Number -1」的訓練資料組態會產生一個名為的訓練資料組態檔案`job-number-1.json`。
- **targets**— 節點和邊緣類別標籤目標的 JSON 陣列，代表用於訓練目的的機器學習類別標籤。請參閱 neptune_ml物件中的targets欄位 (p. 674)。
- **features**— 節點屬性功能的 JSON 陣列。請參閱中的features欄位neptune_ml (p. 678)。

neptune_ml物件中的targets欄位

JSON 訓練資料匯出設定中的targets欄位包含一組目標物件，這些物件會指定訓練任務，以及用於訓練此工作的機器學習類別標籤。目標物件的內容會根據您正在訓練屬性圖資料還是 RDF 資料而有所不同。

對於屬性圖節點分類和回歸工作，陣列中的目標物件可以如下所示：

```
{
  "node": "(node property-graph label)",
  "property": "(property name)",
  "type" : "(used to specify classification or regression)",
  "split_rate": [0.8,0.2,0.0],
  "separator": ","
}
```

對於屬性圖邊緣分類，回歸或鏈接預測任務，它們可以如下所示：

```
{
  "edge": "(edge property-graph label)",
  "property": "(property name)",
  "type" : "(used to specify classification, regression or link_prediction)",
  "split_rate": [0.8,0.2,0.0],
  "separator": ","
}
```

對於 RDF 分類和回歸任務，陣列中的目標對象可以如下所示：

```
{
  "node": "(node type of an RDF node)",
  "predicate": "(predicate IRI)",
  "type" : "(used to specify classification or regression)",
  "split_rate": [0.8,0.2,0.0]
}
```

對於 RDF 鏈接預測任務，數組中的目標對象可以如下所示：

```
{
  "subject": "(source node type of an edge)",
  "predicate": "(relation type of an edge)",
  "object": "(destination node type of an edge)",
  "type" : "link_prediction",
  "split_rate": [0.8,0.2,0.0]
}
```

目標物件可包含下列欄位：

內容

- 屬性圖目標物件中的欄位 (p. 675)
  - 目標對象中的node（頂點）字段 (p. 675)
  - 屬性圖目標物件中的edge欄位 (p. 675)
  - 屬性圖目標物件中的property欄位 (p. 675)
  - 屬性圖目標物件中的type欄位 (p. 675)
  - 屬性圖目標物件中的split_rate欄位 (p. 676)
  - 屬性圖目標物件中的separator欄位 (p. 676)
- RDF 目標對象中的字段 (p. 676)

- RDF 目標對象中的node字段 (p. 676)
- RDF 目標對象中的subject字段 (p. 676)
- RDF 目標對象中的predicate字段 (p. 677)
- RDF 目標對象中的object字段 (p. 677)
- RDF 目標對象中的type字段 (p. 677)
- 屬性圖目標物件中的split_rate欄位 (p. 677)

屬性圖目標物件中的欄位

目標對象中的node（頂點）字段

目標節點（頂點）的屬性圖標籤。目標物件必須包含node元素或元edge素，但不能同時包含兩者。

Anode 可以採用單個值，如下所示：

```
  "node": "Movie"
```

或者，在多標籤頂點的情況下，它可以採用一組值，如下所示：

```
  "node": ["Content", "Movie"]
```

屬性圖目標物件中的edge欄位

透過其起始節點標籤、自己的標籤及其結束節點標籤來指定目標邊緣。目標物件必須包含元edge素或元node素，但不能同時包含兩者。

字edge段的值是一個 JSON 數組，由三個字符串組成，這些字符串代表開始節點的屬性圖標籤，邊緣本身的屬性圖標籤以及終端節點的屬性圖標籤，如下所示：

```
  "edge": ["Person_A", "knows", "Person_B"]
```

如果開始節點和/或結束節點有多個標籤，請將它們包含在一個數組中，如下所示：

```
  "edge": [ ["Admin", Person_A"], "knows", ["Admin", "Person_B"] ]
```

屬性圖目標物件中的property欄位

指定目標頂點或邊緣的屬性，如下所示：

```
  "property" : "rating"
```

除了目標任務為連結預測時，此欄位為必填欄位。

屬性圖目標物件中的type欄位

指出要在或上執行的目標工作類型nodeedge，如下所示：

```
  "type" : "regression"
```

節點支援的工作類型如下：

- `classification`

- regression

支援的邊工作類型為：

- classification
- regression
- link_prediction

此欄位為必填。

屬性圖目標物件中的split_rate欄位

(選擇性) 訓練、驗證和測試階段將分別使用的節點或邊緣比例估計。這些比例由一個介於零和一之間的三個數字的 JSON 陣列表示，最多加起來一個：

```
"split_rate": [0.7, 0.1, 0.2]
```

如果您未提供選擇性split_rate欄位，則預設估計值為[0.9, 0.1, 0.0]。

屬性圖目標物件中的separator欄位

(選擇性) 搭配分類工作使用。

該separator欄位指定用於將多個類別值儲存在字串中時，用來將目標屬性值分割為多個分類值的字元。例如：

```
"separator": "|"
```

顯示separator欄位表示該工作為多目標分類工作。

RDF 目標對象中的字段

RDF 目標對象中的node字段

定義目標節點的節點類型。與節點分類工作或節點回歸工作搭配使用。在 RDF 中的節點的節點類型由定義：

```
  node_id, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, node_type
```

一個 RDF 只node能採取一個值，如下所示：

```
  "node": "http://aws.amazon.com/neptune/csv2rdf/class/Movie"
```

RDF 目標對象中的subject字段

對於連結預測工作，subject定義目標邊緣的來源節點類型。

```
  "subject": "http://aws.amazon.com/neptune/csv2rdf/class/Director"
```

> **Note**
>
> 對於連結預測工作，subject應與predicate和一起使用object。如果未提供這三個邊緣中的任何一個，則會將所有邊緣視為訓練目標。

RDF 目標對象中的predicate字段

針對節點分類和節點迴歸工作，predicate定義要使用哪些常值資料做為目標節點的目標節點功能。

```
  "predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/genre"
```

> Note
>
> 如果目標節點只有一個定義目標節點功能的述詞，則可以省略該predicate字段。

對於連結預測工作，predicate定義目標邊緣的關係類型：

```
"predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/direct"
```

> Note
>
> 對於連結預測工作，predicate應與subject和一起使用object。如果未提供這三個邊緣中的任何一個，則會將所有邊緣視為訓練目標。

RDF 目標對象中的object字段

對於連結預測工作，請object定義目標邊緣的目標節點類型：

```
  "object": "http://aws.amazon.com/neptune/csv2rdf/class/Movie"
```

> Note
>
> 對於連結預測工作，object應與subject和一起使用predicate。如果未提供這三個邊緣中的任何一個，則會將所有邊緣視為訓練目標。

RDF 目標對象中的type字段

指示要執行的目標工作類型，如下所示：

```
  "type" : "regression"
```

RDF 資料支援的工作類型為：

- link_prediction
- classification
- regression

此欄位為必填。

屬性圖目標物件中的split_rate欄位

(選擇性) 訓練、驗證和測試階段將分別使用的節點或邊緣比例估計。這些比例由一個介於零和一之間的三個數字的 JSON 陣列表示，最多加起來一個：

```
"split_rate": [0.7, 0.1, 0.2]
```

如果您未提供選擇性split_rate欄位，則預設估計值為[0.9, 0.1, 0.0]。

中的features欄位neptune_ml

屬性值和 RDF 常值有不同的格式和資料類型。為了在機器學習中獲得良好的效能，將這些值轉換為稱為功能的數值編碼非常重要。

Neptune ML 會執行特徵擷取和編碼，做為資料匯出和資料處理步驟的一部分，如中所述Neptune ML 中的功能編碼 (p. 695)。

對於性質圖資料集，匯出程序會自動推斷字串性質和包含倍數值的數值性質的圖auto徵。對於包含單一值的數值性質，它會推論numerical圖徵。對於日期屬性，它推斷datetime功能。

如果您想要覆寫自動推斷的功能規格，或為屬性新增值區數字 FastText、TF-IDF 或 SBERT 規格，您可以使用功能欄位來控制特徵編碼。

> Note
>
> 您只能使用features欄位來控制屬性圖資料的特徵規格，而不能控制 RDF 資料的特徵規格。

對於任意格式文字，Neptune ML 可以使用數個不同的模型，將字串屬性值中的記號序列轉換為固定大小的實數值向量：

- text_fasttext (p. 696)— 使用 fastText 編碼。對於使用 fastText 支援的五種語言之一的功能，建議使用此編碼方式。
- text_sbert (p. 697)— 使用句子伯特 (SBERT) 編碼模型。對於不支援的文字，text_fasttext建議使用這種編碼方式。
- text_word2vec (p. 697)— 使用谷歌最初發布的 Word2VEC 算法對文本進行編碼。只支持英語。
- text_tfidf (p. 697)— 使用詞彙頻率反向文件頻率 (TF-IDF) 向量化程式來編碼文字。TF-IDF 編碼支援其他編碼不支援的統計功能。

該features字段包含節點屬性功能的 JSON 數組。陣列可包含下列欄位：

內容

- 中的node欄位features (p. 678)
- 中的edge欄位features (p. 679)
- 中的property欄位features (p. 679)
- 圖徵type欄位的可能值 (p. 679)
- 該norm領域 (p. 680)
- 該language領域 (p. 681)
- 該max_length領域 (p. 681)
- 該separator領域 (p. 681)
- 該range領域 (p. 681)
- 該bucket_cnt領域 (p. 681)
- 該slide_window_size領域 (p. 682)
- 該imputer領域 (p. 682)
- 該max_features領域 (p. 682)
- 該min_df領域 (p. 682)
- 該ngram_range領域 (p. 683)
- 該datetime_parts領域 (p. 683)

中的node欄位features

該功能node變數指定圖徵頂點的性質圖表標示。例如：

```
"node": "Person"
```

如果一個頂點有多個標籤，使用一個數組來包含它們。例如：

```
"node": ["Admin", "Person"]
```

中的edge欄位features

該edge欄位指定特徵邊的邊類型。邊緣類型由包含源頂點的屬性圖標籤，邊緣的屬性圖標籤和目標頂點的屬性圖標籤的陣列組成。指定邊特徵時，您必須提供全部三個值。例如：

```
"edge": ["User", "reviewed", "Movie"]
```

如果邊緣類型的來源或目標頂點有多個標籤，請使用另一個陣列來包含它們。例如：

```
"edge": [["Admin", "Person"]. "edited", "Post"]
```

中的property欄位features

使用屬性參數可指定由node參數識別之頂點的性質。例如：

```
"property" : "age"
```

圖徵type欄位的可能值

該type參數指定要定義的特徵的類型。例如：

```
"type": "bucket_numerical"
```

type參數的可能值

- **"auto"**— 指定 Neptune ML 應自動偵測內容類型並套用適當的功能編碼。auto功能也可以有一個可選separator字段。

  請參閱 Neptune ML 中的自動功能編碼 (p. 698)。
- **"category"**— 此功能編碼將屬性值表示為數個類別之一。換句話說，該功能可以採用一個或多個離散值。category功能也可以有一個可選separator字段。

  請參閱 Neptune ML 中的分類特徵 (p. 695)。
- **"numerical"**— 此功能編碼將數字屬性值表示為連續間隔中的數字，其中「大於」和「小於」具有意義。

  numerical功能也可以有選擇性normimputer、和separator欄位。

  請參閱 Neptune ML 中的數值特徵 (p. 695)。
- **"bucket_numerical"**— 此功能編碼功能會將數值內容值劃分為一組值區或類別。

  例如，您可以將人們的年齡編碼為 4 個桶：孩子（0-20），年輕成人（20-40），中年（40-60）和長者（60 歲及以上）。

  bucket_numerical功能需要range和bucket_cnt欄位，也可以選擇性地包含和imputer/或slide_window_size欄位。

請參閱 Neptune ML 中的值區數值特徵 (p. 695)。

- **"datetime"**— 此功能編碼將 datetime 屬性值表示為下列分類特徵的陣列：年、月、星期和小時。

  使用datetime_parts參數可以消除這四個品類中的一個或多個品類。

  請參閱 Neptune ML 中的日期時間功能 (p. 698)。
- **"text_fasttext"**— 此功能編碼會使用 fastT ext ext 模型，將包含句子或自由格式文字的屬性值轉換為數值向量。它支持五種語言，即英語（en），中文（zh），印地語（hi），西班牙語（es）和法語（fr）。對於這五種語言中任何一種文本屬性值，text_fasttext建議使用編碼。但是，它無法處理同一句子包含多種語言單詞的情況。

  對於 fastText 支援的語言以外的其他語言，請使用text_sbert編碼。

  如果您的性質值文字字串長度超過 120 個記號，請使用該max_length欄位來限制每個編"text_fasttext"碼字串中的記號數目。

  請參閱 Neptune ML 中文本屬性值的 fastT ext 文本編碼 (p. 696)。
- **"text_sbert"**— 此編碼會使用句子 BERT (SBERT) 模型將文字屬性值轉換為數值向量。Neptune 支持兩個 SBERT 方法text_sbert128，即，這是默認的，如果你只是指定text_sbert，和text_sbert512。它們之間的區別在於被編碼的文本屬性中令牌的最大數量。text_sbert128編碼僅對前 128 個令牌進行編碼，而最多可text_sbert512編碼 512 個令牌。因此，使用text_sbert512可能需要更多的處理時間text_sbert128。這兩種方法都比text_fasttext.

  這些方text_sbert*法支援多種語言，而且可以對包含多種語言的句子進行編碼。

  請參閱 Neptune ML 中文字特徵的句子 BERT (SBERT) 編碼 (p. 697)。
- **"text_word2vec"**— 此編碼會使用 Word2VEC 演算法將文字屬性值轉換為數值向量。它僅支持英語。

  請參閱 Neptune ML 中文本功能的單詞 2VEC 編碼 (p. 697)。
- **"text_tfidf"**— 此編碼會使用術語頻率反向文件頻率 (TF-IDF) 向量器，將文字屬性值轉換為數值向量。

  您可以使用欄位、欄位和ngram_range欄位來定min_df義text_tfidf特徵編碼的參數。max_features

  請參閱 Neptune ML 中文本功能的 TF-IDF 編碼 (p. 697)。
- **"none"**— 使用none類型會導致不發生特徵編碼。系統會剖析並儲存原始屬性值。

  none僅當您計劃執行自訂特徵編碼作為自訂模型訓練的一部分時，才使用此選項。

該norm領域

數字圖徵是必要欄位。它指定了對數值使用的規範化方法：

```
"norm": "min-max"
```

支援下列正規化：

- 「min-max」 — 通過從中減去最小值，然後將其除以最大值和最小值之間的差值來標準化每個值。
- 「標準」 — 將每個值除以所有值的總和來規範化。
- 「none」 — 不要在編碼過程中對數值進行規範化。

請參閱 Neptune ML 中的數值特徵 (p. 695)。

該language領域

語言欄位會指定文字屬性值中使用的語言。它的用法取決於文本編碼方法：

- 對於text_fasttext (p. 696)編碼，此欄位為必填欄位，且必須指定下列其中一種語言：
  - en(英文)
  - zh(中文)
  - hi(印地語)
  - es(西班牙文)
  - fr(法文)
- 對於text_sbert (p. 696)編碼，不使用此欄位，因為 SBERT 編碼是多語言的。
- 對於text_word2vec (p. 697)編碼，此字段是可選的，因為text_word2vec僅支持英文。如果存在，則必須指定英語模型的名稱：

```
"language" : "en_core_web_lg"
```

- 對於text_tfidf (p. 697)編碼，不使用此字段。

該max_length領域

對於text_fasttext功能而言，此max_length欄位是可選的，其中指定要編碼的輸入文字功能中的最大記號數目。長度超過的輸入文字會max_length被截斷。例如，將 max_length 設定為 128 表示文字序列中 128 之後的任何記號都會被忽略：

```
"max_length": 128
```

該separator領域

此欄位可選擇性地與categorynumerical和auto功能搭配使用。它指定可用於將屬性值拆分為多個分類值或數值的字符：

```
"separator": ";"
```

僅當性質將多個分隔值儲存在單一字串中時，才使用此separator欄位，例如"Actor;Director"或"0.1;0.2"。

請參閱分類功能 (p. 695)、數值特徵 (p. 695)和自動編碼 (p. 698)。

該range領域

bucket_numerical功能是必要欄位。它指定要被劃分為存儲桶的數值的範圍，格式如下[*lower-bound*, *upper-bound*]：

```
"range" : [20, 100]
```

如果屬性值小於下限，則會將其指派給第一個值區，或者如果它大於上限，則會將其指派給最後一個值區。

請參閱 Neptune ML 中的值區數值特徵 (p. 695)。

該bucket_cnt領域

bucket_numerical功能是必要欄位。它指定了由range參數定義的數字範圍應分為的桶的數量：

```
"bucket_cnt": 10
```

請參閱 Neptune ML 中的值區數值特徵 (p. 695)。

該slide_window_size領域

此欄位可選擇搭配bucket_numerical功能使用，以將值指派給多個值區：

```
"slide_window_size": 5
```

幻燈片窗口的工作方式是 Neptune ML 採用窗口大小，**s**並將屬性**v**的每個數字值轉換為從 `v - s/2` 到的範圍 `v + s/2` 。然後，該值會指定給範圍重疊的每個值區。

請參閱 Neptune ML 中的值區數值特徵 (p. 695)。

該imputer領域

此欄位可選擇搭配numerical和bucket_numerical功能使用，以提供填入缺少值的填補技術：

```
"imputer": "mean"
```

支持的補充技術是：

- "mean"
- "median"
- "most-frequent"

如果您不包含輸入器參數，資料預先處理會在遇到遺漏值時停止並結束。

請參閱 Neptune ML 中的數值特徵 (p. 695) 和 Neptune ML 中的值區數值特徵 (p. 695)。

該max_features領域

text_tfidf功能可以選擇使用此欄位來指定要編碼的最大字詞數：

```
"max_features": 100
```

設定為 100 會導致 TF-IDF 向量化程式只編碼 100 個最常見的項目。如果不包含，則預設值max_features為 5,000。

請參閱 Neptune ML 中文本功能的 TF-IDF 編碼 (p. 697)。

該min_df領域

此欄位可由text_tfidf功能選擇性地使用，以指定要編碼之詞彙的最小文件頻率：

```
"min_df": 5
```

設定 5 表示字詞必須出現在至少 5 個不同的屬性值中，才能進行編碼。

如果您不包含min_df參數，則預設值為2。

請參閱 Neptune ML 中文本功能的 TF-IDF 編碼 (p. 697)。

該ngram_range領域

此欄位可由text_tfidf功能選擇性地使用，以指定應將單字或標記的大小序列視為要編碼的潛在個別術語：

```
"ngram_range": [2, 4]
```

該值[2, 4]指定 2，3 和 4 個單詞的序列應被視為潛在的個別術語。

如果您沒有明確設置，則默認值ngram_range是[1, 1]，這意味著只有單個單詞或令牌被視為要編碼的術語。

請參閱 Neptune ML 中文本功能的 TF-IDF 編碼 (p. 697)。

該datetime_parts領域

datetime功能可選地使用此欄位來指定要分類編碼 datetime 值的哪些部分：

```
"datetime_parts": ["weekday", "hour"]
```

如果不包含datetime_parts，Neptune ML 預設會編碼日期時間值的年、月、星期和小時部分。該值["weekday", "hour"]表示只有工作日和小時的日期時間值應在功能中進行分類編碼。

如果其中一個零件在訓練集中沒有多個唯一值，則不會對其進行編碼。

請參閱Neptune ML 中的日期時間功能 (p. 698)。

在中additionalParams使用參數調整模型訓練組態的範例

內容

- 屬性圖示例使用additionalParams (p. 684)
  - 指定模型訓練組態的預設分割率 (p. 684)
  - 指定模型訓練組態的節點分類工作 (p. 685)
  - 指定模型訓練組態的多類別節點分類任務 (p. 685)
  - 指定模型訓練組態的節點迴歸工作 (p. 685)
  - 指定模型訓練組態的邊緣分類工作 (p. 686)
  - 指定模型訓練組態的多類別邊緣分類工作 (p. 686)
  - 指定模型訓練組態的邊迴歸 (p. 687)
  - 指定模型訓練組態的連結預測任務 (p. 687)
  - 指定數值貯體功能 (p. 687)
  - 指定特Word2Vec徵 (p. 688)
  - 指定特FastText徵 (p. 688)
  - 指定特Sentence BERT徵 (p. 689)
  - 指定特TF-IDF徵 (p. 689)
  - 指定特datetime徵 (p. 690)
  - 指定特category徵 (p. 690)
  - 指定特numerical徵 (p. 690)
  - 指定auto特徵 (p. 691)
- RDF 範例使用additionalParams (p. 691)
  - 指定模型訓練組態的預設分割率 (p. 691)
  - 指定模型訓練組態的節點分類工作 (p. 692)
  - 指定模型訓練組態的節點迴歸工作 (p. 692)
  - 指定特定邊緣的連結預測任務 (p. 692)
  - 指定所有邊的連結預測任務 (p. 692)

屬性圖示例使用additionalParams

指定模型訓練組態的預設分割率

在下列範例中，split_rate參數會設定模型訓練的預設分割率。如果未指定預設的分割率，訓練會使用 [0.9, 0.1, 0.0] 的值。您可以針對每個目標指定一個，以每個目標split_rate為基礎覆寫預設值。

在下列範例中，default split_rate欄位指示除非針對每個目標進行覆寫，否則[0.7,0.1,0.2]應使用的分割率：」

```
"additionalParams": {
"neptune_ml": {
  "version": "v2.0",
  "split_rate": [0.7,0.1,0.2],
  "targets": [
    (...)
  ],
  "features": [
    (...)
  ]
```

```
  }
}
```

指定模型訓練組態的節點分類工作

若要指出哪個節點屬性包含標示範例以供訓練用途，請使用將節點分類元素加入至targets陣列"type" : "classification"。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

在下列範例中，nodetarget 會指出應將每個Movie節點的genre屬性視為節點類別標籤。此split_rate值會覆寫預設的分割率：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "node": "Movie",
        "property": "genre",
        "type": "classification",
        "split_rate": [0.7,0.1,0.2]
      }
    ],
    "features": [
      (...)
    ]
  }
}
```

指定模型訓練組態的多類別節點分類任務

若要指出哪個節點內容包含多個標示範例以供訓練之用，請使"type" : "classification"用和separator指定可用於將目標內容值分割為多個分類值的字元，將節點分類元素新增至目標陣列。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

在下列範例中，nodetarget 會指出應將每個Movie節點的genre屬性視為節點類別標籤。該separator字段指示每個類型屬性包含多個分號分隔的值：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "node": "Movie",
        "property": "genre",
        "type": "classification",
        "separator": ";"
      }
    ],
    "features": [
      (...)
    ]
  }
}
```

指定模型訓練組態的節點迴歸工作

若要指出哪個節點屬性包含標記回歸以供訓練用途，請使用將節點迴歸元素加入至目標陣列"type" : "regression"。如果您想要覆寫預設的分割率，您也可以這樣做，您也可以這樣做。

下列node目標指出每個Movie節點的rating屬性應視為節點迴歸標籤：

```
    "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "node": "Movie",
        "property": "rating",
        "type" : "regression",
        "split_rate": [0.7,0.1,0.2]
      }
    ],
    "features": [
      ...
    ]
  }
}
```

指定模型訓練組態的邊緣分類工作

若要指出哪個邊屬性包含標示範例以供訓練用途，請使用將邊元素加入至`targets`陣列`"type" : "regression"`。如果您想要覆寫預設的分割率，您也可以這樣做，您也可以這樣做。

下列`edge`目標指出應將每個`knows`邊緣的`metAtLocation`屬性視為邊緣類別標籤：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "edge": ["Person", "knows", "Person"],
        "property": "metAtLocation",
        "type": "classification"
      }
    ],
    "features": [
      (...)
    ]
  }
}
```

指定模型訓練組態的多類別邊緣分類工作

若要指出哪個邊屬性包含多個標示範例以進行訓練，請將邊元素新增至`targets`陣列`"type" : "classification"`、使用，以及指定用於將目標屬性值分割為多個分類值的字元的`separator`欄位。如果您想要覆寫預設的分割率，您也可以這樣`split_rate`做。

下列`edge`目標指出應將每個`repliedTo`邊緣的`sentiment`屬性視為邊類別標籤。分隔符號欄位會指出每個情緒屬性都包含多個逗號分隔值：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "edge": ["Person", "repliedTo", "Message"],
        "property": "sentiment",
        "type": "classification",
        "separator": ","
      }
    ],
    "features": [
```

```
      (...)
    ]
  }
}
```

指定模型訓練組態的邊迴歸

若要指出哪個邊性質包含標示回歸範例以供訓練用途，請使用將edge元素加入至targets陣列"type" : "regression"。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

下列edge目標指示應將每個reviewed邊的rating性質視為邊迴歸：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "edge": ["Person", "reviewed", "Movie"],
        "property": "rating",
        "type" : "regression"
      }
    ],
    "features": [
      (...)
    ]
  }
}
```

指定模型訓練組態的連結預測任務

若要指出應將哪些邊用於連結預測訓練目的，請使用將邊元素加入至目標陣列"type" : "link_prediction"。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

下列edge目標指出應該使用cites邊緣進行連結預測：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "edge": ["Article", "cites", "Article"],
        "type" : "link_prediction"
      }
    ],
    "features": [
      (...)
    ]
  }
}
```

指定數值貯體功能

您可以透過加入"type": "bucket_numerical"至features陣列來為節點性質指定數值資料圖徵。

下列node功能表示應將每個Person節點的age屬性視為數值時段特徵：

```
  "additionalParams": {
  "neptune_ml": {
    "targets": [
```

```
      ...
    ],
    "features": [
      {
        "node": "Person",
        "property": "age",
        "type": "bucket_numerical",
        "range": [1, 100],
        "bucket_cnt": 5,
        "slide_window_size": 3,
        "imputer": "median"
      }
    ]
  }
}
```

指定特Word2Vec徵

您可以透過加入"type": "text_word2vec"至features陣列來指定節點屬性的Word2Vec特徵。

下列node功能表示應將每個Movie節點的description屬性視為Word2Vec特徵：

```
"additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Movie",
        "property": "description",
        "type": "text_word2vec",
        "language": "en_core_web_lg"
      }
    ]
  }
}
```

指定特FastText徵

您可以透過加入"type": "text_fasttext"至features陣列來指定節點屬性的FastText特徵。此language欄位為必填，且您也必須指定下列其中一種語言代碼：

- en(英文)
- zh(中文)
- hi(印地語)
- es(西班牙文)
- fr(法文)

請注意，text_fasttext編碼一次無法處理一個功能中的多種語言。

下列node功能表示應將每個Movie節點的 Frenchdescription 屬性視為FastText特徵：

```
"additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
```

```
      ...
    ],
    "features": [
      {
        "node": "Movie",
        "property": "description",
        "type": "text_fasttext",
        "language": "fr",
        "max_length": 1024
      }
    ]
  }
}
```

指定特Sentence BERT徵

您可以透過加入"type": "text_sbert"至features陣列來指定節點屬性的Sentence BERT特徵。您不需要指定語言，因為該方法會使用多語言模型自動編碼文字功能。

下列node功能表示應將每個Movie節點的description屬性視為Sentence BERT特徵：

```
"additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Movie",
        "property": "description",
        "type": "text_sbert128",
      }
    ]
  }
}
```

指定特TF-IDF徵

您可以透過加入"type": "text_tfidf"至features陣列來指定節點屬性的TF-IDF特徵。

下列node功能表示應將每個Person節點的bio屬性視為TF-IDF特徵：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Movie",
        "property": "bio",
        "type": "text_tfidf",
        "ngram_range": [1, 2],
        "min_df": 5,
        "max_features": 1000
      }
    ]
  }
}
```

指定特datetime徵

匯出程序會自動推論日期性質的datetime圖徵。但是，如果要限制datetime_parts用於datetime特徵的內容，或取代特徵規格，以便將通常被視為特徵的屬性明確視為auto特徵，則可以透過在datetime特徵陣列中增加一個"type": "datetime"來執行此操作。

下列node功能表示應將每個Post節點的createdAt屬性視為datetime特徵：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Post",
        "property": "createdAt",
        "type": "datetime",
        "datetime_parts": ["month", "weekday", "hour"]
      }
    ]
  }
}
```

指定特category徵

匯出程序會自動推論字串auto性質和包含倍數值的數值性質的圖徵。對於包含單一值的數值性質，它會推論numerical圖徵。對於日期屬性，它推斷datetime功能。

如果您要取代特徵規格，以便將性質視為分類特徵，請在特徵陣列中加入一個"type": "category"。如果屬性包含多個值，請包括一個separator欄位。例如：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Post",
        "property": "tag",
        "type": "category",
        "separator": "|"
      }
    ]
  }
}
```

指定特numerical徵

匯出程序會自動推論字串auto性質和包含倍數值的數值性質的圖徵。對於包含單一值的數值性質，它會推論numerical圖徵。對於日期屬性，它推斷datetime功能。

如果您要取代特徵規格，以便將性質視為numerical特徵，請加入"type": "numerical"至特徵陣列。如果屬性包含多個值，請包括一個separator欄位。例如：

```
  "additionalParams": {
```

```
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "Recording",
        "property": "duration",
        "type": "numerical",
        "separator": ","
      }
    ]
  }
}
```

指定auto特徵

匯出程序會自動推論字串auto性質和包含倍數值的數值性質的圖徵。對於包含單一值的數值性質，它會推論numerical圖徵。對於日期屬性，它推斷datetime功能。

如果您要取代特徵規格，以便將性質視為auto特徵，請加入"type": "auto"至特徵陣列。如果屬性包含多個值，請包括一個separator欄位。例如：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      ...
    ],
    "features": [
      {
        "node": "User",
        "property": "role",
        "type": "auto",
        "separator": ","
      }
    ]
  }
}
```

RDF 範例使用additionalParams

指定模型訓練組態的預設分割率

在下列範例中，split_rate參數會設定模型訓練的預設分割率。如果未指定預設的分割率，訓練會使用 [0.9, 0.1, 0.0] 的值。您可以針對每個目標指定一個，以每個目標split_rate為基礎覆寫預設值。

在下列範例中，default split_rate欄位指示除非針對每個目標進行覆寫，否則[0.7,0.1,0.2]應使用的分割率：」

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "split_rate": [0.7,0.1,0.2],
    "targets": [
      (...)
    ]
  }
}
```

指定模型訓練組態的節點分類工作

若要指出哪個節點屬性包含標示範例以供訓練用途，請使用將節點分類元素加入至targets陣列"type" : "classification"。添加節點字段以指示目標節點的節點類型。新增predicate欄位以定義使用哪些常值資料做為目標節點的目標節點功能。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

在下列範例中，nodetarget 會指出應將每個Movie節點的genre屬性視為節點類別標籤。此split_rate值會覆寫預設的分割率：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "node": "http://aws.amazon.com/neptune/csv2rdf/class/Movie",
        "predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/genre",
        "type": "classification",
        "split_rate": [0.7,0.1,0.2]
      }
    ]
  }
}
```

指定模型訓練組態的節點迴歸工作

若要指出哪個節點屬性包含標記回歸以供訓練用途，請使用將節點迴歸元素加入至目標陣列"type" : "regression"。新增node欄位以指示目標節點的節點類型。新增predicate欄位以定義使用哪些常值資料做為目標節點的目標節點功能。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

下列node目標指出每個Movie節點的rating屬性應視為節點迴歸標籤：

```
    "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "node": "http://aws.amazon.com/neptune/csv2rdf/class/Movie",
        "predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/rating",
        "type": "regression",
        "split_rate": [0.7,0.1,0.2]
      }
    ]
  }
}
```

指定特定邊緣的連結預測任務

若要指出應將哪些邊用於連結預測訓練目的，請使用將邊元素加入至目標陣列"type" : "link_prediction"。加入subjectpredicate和object欄位以指定邊類型。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

下列edge目標表示Movies應該使用連接Directors到的directed邊緣進行連結預測：

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "subject": "http://aws.amazon.com/neptune/csv2rdf/class/Director",
        "predicate": "http://aws.amazon.com/neptune/csv2rdf/datatypeProperty/directed",
```

```
        "object": "http://aws.amazon.com/neptune/csv2rdf/class/Movie",
        "type" : "link_prediction"
      }
    ]
  }
}
```

指定所有邊的連結預測任務

若要指出所有邊緣都應用於連結預測訓練目的，請使用將edge元素加入至目標陣列"type" : "link_prediction"。請勿新增subjectpredicate、或object欄位。如果您想要覆寫預設的分割率，您也可以這樣split_rate做。

```
  "additionalParams": {
  "neptune_ml": {
    "version": "v2.0",
    "targets": [
      {
        "type" : "link_prediction"
      }
    ]
  }
}
```

處理從 Neptune 匯出的圖形資料以進行訓練

資料處理步驟會採用匯出程序所建立的 Neptune 圖形資料，並建立訓練期間深層圖庫 (DGL) 所使用的資訊。這包括執行各種資料對映和轉換：

- 解析節點和邊緣以構建 DGL 所需的圖形和 ID 映射文件。
- 將節點和邊屬性轉換為 DGL 所需的節點和邊特徵。
- 將數據拆分為培訓，驗證和測試集。

管理 Neptune ML 的資料處理步驟

從 Neptune 匯出要用於模型訓練的資料之後，您可以使用curl (或awscurl) 命令啟動資料處理工作，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/dataprocessing \
  -H 'Content-Type: application/json' \
  -d '{
        "inputDataS3Location" : "s3://(Amazon S3 bucket name)/(path to your input folder)",
        "id" : "(a job ID for the new job)",
        "processedDataS3Location" : "s3://(S3 bucket name)/(path to your output folder)",
        "configFileName" : "training-job-configuration.json"
      }'
```

有關如何使用此命令的詳細資訊資料處理命令 (p. 753)，以及有關如何取得執行中工作狀態、如何停止執行中工作以及如何列出所有執行中工作的資訊。

處理 Neptune ML 的更新圖形資料

您也可以提供 API，previousDataProcessingJobId以確保新的資料處理工作使用與先前工作相同的處理方法。當您想要在 Neptune 中取得更新圖形資料的預測時，可以透過對新資料重新訓練舊模型，或在新資料上重新計算模型人工因素，這是必要的。

您可以通過使用curl（或awscurl）命令來執行此操作，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/dataprocessing \
  -H 'Content-Type: application/json' \
  -d '{ "inputDataS3Location" : "s3://(Amazon S3 bucket name)/(path to your input folder)",
        "id" : "(a job ID for the new job)",
        "processedDataS3Location" : "s3://(Amazon S3 bucket name)/(path to your output
 folder)",
        "previousDataProcessingJobId", "(the job ID of the previous data-processing job)"}'
```

將previousDataProcessingJobId參數值設定為與訓練模型相對應的先前資料處理工作的工作 ID。

> **Note**
>
> 目前不支援更新圖形中的節點刪除。如果已在更新的圖形中移除節點，您必須啟動全新的資料處理工作，而不是使用previousDataProcessingJobId。

Neptune ML 中的功能編碼

屬性值有不同的格式和資料類型。為了在機器學習中獲得良好的效能，將這些值轉換為稱為功能的數值編碼非常重要。

Neptune ML 使用此處描述的功能編碼技術，執行功能擷取和編碼作為資料匯出和資料處理步驟的一部分。

> Note
>
> 如果您計劃在自訂模型實作中實作自己的特徵編碼，您可以選取none作為特徵編碼類型，以停用資料預處理階段中的自動特徵編碼。然後不會在該節點或 edge 屬性上進行任何特徵編碼，而是將原始屬性值剖析並儲存在字典中。資料預先處理仍會從匯出的資料集建立 DGL 圖形，但是建構的 DGL 圖形沒有預先處理的訓練功能。
> 只有當您計劃執行自訂特徵編碼作為自訂模型訓練的一部分時，才應使用此選項。如需詳細資訊，請參閱 Neptune ML 中的自定義模型 (p. 724)。

Neptune ML 中的分類特徵

可以從可能值的固定列表中獲取一個或多個不同值的屬性是分類特徵。在 Neptune ML 中，分類特徵使用一熱編碼進行編碼。以下示例顯示了不同食物的屬性名稱如何根據其類別進行一次熱編碼：

```
 Food        Veg.   Meat   Fruit    Encoding
---------    ----   ----   -----    --------
Apple          0      0      1         001
Chicken        0      1      0         010
Broccoli       1      0      0         100
```

> Note
>
> 任何類別功能中的類別數目上限為 100。如果性質具有超過 100 個品類的值，則僅將其中最常見的 99 個品類放置在不同的品類中，其餘品類將放置在名為的特殊品類中OTHER。

Neptune ML 中的數值特徵

其值為實數的任何屬性都可以編碼為 Neptune ML 中的數值特徵。數值特徵使用浮點數進行編碼。

您可以指定在編碼數值特徵時使用的資料規範化方法，如下所示："norm": "*normalization technique*"。支持以下正規化技術：

- 「none」 — 不要在編碼過程中對數值進行規範化。
- 「min-max」 — 通過從中減去最小值，然後將其除以最大值和最小值之間的差值來標準化每個值。
- 「標準」 — 將每個值除以所有值的總和來規範化。

Neptune ML 中的值區數值特徵

您可以將數值壓縮為類別，而不是使用原始數字表示數值屬性。例如，您可以將人們的年齡劃分為兒童（0-20 歲），年輕人（20-40 歲），中年人（40-60 歲）和長者（從 60 歲開始）。使用這些數值桶，您可以將數值屬性轉換為一種類別特徵。

在 Neptune ML 中，您可能會導致數值屬性編碼為值區數字特徵，您必須提供兩件事：

- 形式中的數值範圍 "range": [*a*, *b*] ，其中a和b是整數。
- 桶計數，在形式 "bucket_cnt": *c* ，其中c是桶的數量，也是一個整數。

Neptune ML 然後計算每個存儲桶的大小 `( b - a ) / c` ，並將每個數值編碼為它落入的任何存儲桶的數量。任何小於的值都會a被視為屬於第一個值區，而任何大於的值都會b被視為屬於最後一個值區。

您也可以選擇性地使數值落入多個存儲桶，方法是指定幻燈片窗口大小，如下所示：`"slide_window_size": s` ，其中s是一個數字。Neptune ML 接著會將內容v的每個數值轉換為從到之間 `v - s/2` 的範圍 `v + s/2` ，並將該值v指派給範圍涵蓋的每個值區。

最後，您還可以選擇性地提供一種填充數值特徵和值區數值特徵的缺失值的方法。您可以使用指插技術是 `"imputer": "imputation technique "`、或之一來執行`"mean"`此操作`"most-frequent"`。`"median"`如果您沒有指定輸出器，則缺少的值可能會導致處理停止。

Neptune ML 中的文字特徵編碼

對於自由格式文字，Neptune ML 可以使用數個不同的模型，將屬性值字串中的記號序列轉換為固定大小的實值向量：

- `text_fasttext` (p. 696)— 使用 fastText 編碼。對於使用 fastText 支援的五種語言之一的功能，建議使用此編碼方式。
- `text_sbert` (p. 697)— 使用句子伯特 (SBERT) 編碼模型。對於不支援的文字，`text_fasttext`建議使用這種編碼方式。
- `text_word2vec` (p. 697)— 使用谷歌最初發布的 Word2Vec 算法對文本進行編碼。只支持英語。
- `text_tfidf` (p. 697)— 使用詞彙頻率反向文件頻率 (TF-IDF) 向量化程式來編碼文字。TF-IDF 編碼支援其他編碼不支援的統計功能。

Neptune ML 中文本屬性值的 fastT ext 文本編碼

Neptune ML 可以使用 fastText 模型，將文字屬性值轉換為固定大小的實值向量。這是 FastText 支援的五種語言中任何一種文字屬性值的建議編碼方法：

- `en`(英文)
- `zh`(中文)
- `hi`(印地語)
- `es`(西班牙文)
- `fr`(法文)

請注意，fastText 無法處理包含多種語言單詞的句子。

該`text_fasttext`方法可以選擇使用字`max_length`段，該字段指定要編碼的文本屬性值中令牌的最大數量，之後該字符串被截斷。當文字屬性值包含長字串時，這可以改善效能，因`max_length`為如果未指定，fastText 會將所有記號編碼，而不論字串長度為何。

這個範例會指定法文電影標題是使用 fastText 編碼的：

```
{
    "file_name" : "nodes/movie.csv",
    "separator" : ",",
    "node" : ["~id", "movie"],
    "features" : [
      {
        "feature": ["title", "title", "text_fasttext"],
        "language": "fr",
        "max_length": 1024
      }
    ]
  }
```

Neptune ML 中文字特徵的句子 BERT (SBERT) 編碼

Neptune ML 可以使用句子 BERT (SBERT) 模型，將字串屬性值中的記號序列轉換為固定大小的實值向量。Neptune 支持兩種 SBERT 方法：`text_sbert128`，這是默認的，如果你只是指定`text_sbert`，和`text_sbert512`。兩者之間的差異是編碼的文本屬性值字符串的最大長度。`text_sbert128`編碼會在編碼 128 個標記後截斷文字字串，而在編碼 512 個標記後`text_sbert512`截斷文字字串。因此，`text_sbert512`需要的處理時間超過`text_sbert128`。這兩種方法都比`text_fasttext`.

SBERT 編碼是多語言的，因此不需要為您正在編碼的屬性值文字指定語言。SBERT 支持多種語言，並且可以對包含多種語言的句子進行編碼。如果您要編碼包含 FastText 不支援的語言文字的屬性值，則建議使用 SBERT 編碼方法。

下列範例會指定影片標題編碼為 SBERT，最多可包含 128 個標記：

```
{
    "file_name" : "nodes/movie.csv",
    "separator" : ",",
    "node" : ["~id", "movie"],
    "features" : [
      { "feature": ["title", "title", "text_sbert128"] }
    ]
  }
```

Neptune ML 中文本功能的單詞 2VEC 編碼

Neptune ML 可以將字符串屬性值編碼為 Word2Vec 功能（Word2Vec 算法最初由谷歌發布）。該`text_word2vec`方法使用其中一個 Spacy 訓練的模型將字符串中的令牌編碼為密集向量。這只支援使用 en_core_web_lg 模型的英文語言。

下列範例會指定影片標題是使用 Word2VEC 編碼：

```
{
    "file_name" : "nodes/movie.csv",
    "separator" : ",",
    "node" : ["~id", "movie"],
    "features" : [
      {
        "feature": ["title", "title", "text_word2vec"],
        "language": "en_core_web_lg"
      }
    ]
  }
```

請注意，語言欄位是選用的，因為 Neptune 只支援英文`en_core_web_lg`模型。

Neptune ML 中文本功能的 TF-IDF 編碼

Neptune ML 可以將文本屬性值編碼為`text_tfidf`功能。這種編碼使用術語頻率逆文檔頻率（TF-IDF）矢量化器，然後進行尺寸縮小操作，將文本中的單詞序列轉換為數字向量。

TF-IDF (術語頻率 — 反向文件頻率) 是用來測量單字在文件集中有多重要的數值。它的計算方法是將單字在指定性質值中出現的次數除以其中顯示的此類性質值的總數。

例如，如果在給定的電影標題中出現兩次「親吻」一詞（例如「親吻砰砰」），並且「親吻」出現在所有 4 部電影的標題中，則「親吻砰砰」標題中「親吻」的 TF-IDF 值將是 2 / 4 。

最初建立的向量具有 d 維度，其中 d 是該類型之所有屬性值中唯一項數目。維度縮減作業會使用隨機稀疏投影，將該數目減少為 100。然後通過合併其中的所有`text_tfidf`特徵來生成圖表的詞彙。

您可以通過多種方式控制 TF-IDF 矢量化器：

- **max_features**— 使用max_features參數，您可以將text_tfidf特徵中的項數限制為最常見的項數。例如，如果您設定max_features為 100，則只會包含前 100 個最常用的術語。max_features如果您沒有明確設定，則預設值為 5,000。
- **min_df**— 使用min_df參數，您可以將text_tfidf特徵中的項數限制為至少具有指定文件頻率的項數。例如，如果設定min_df為 5，則只會使用出現在至少 5 個不同屬性值中的術語。min_df如果您沒有明確設置它，則默認值為 2。
- **ngram_range**—ngram_range 參數決定要將哪些字詞組合視為術語。例如，如果您設定ngram_range為[2, 4]，則會在「吻吻爆炸」標題中找到以下 6 個詞彙：
  - 2 字術語：「吻吻」，「吻爆炸」和「砰砰」。
  - 3 字術語：「吻吻爆炸」和「吻砰砰」。
  - 4 個字的術語：「吻吻砰砰」。

  ngram_range 的預設設定為 [1, 1]。

Neptune ML 中的日期時間功能

Neptune ML 可以通過將datetime屬性值的一部分編碼為一熱數組來將它們轉換為分類特徵。使用datetime_parts參數可指定下列一或多個要編碼的部["year", "month", "weekday", "hour"]分：如果未設置datetime_parts，默認情況下，所有四個部分都被編碼。

例如，如果日期時間值的範圍跨越 2010 到 2012 年，則日期時間項目的四個部分2011-04-22 01:16:34如下：

- 年 —[0, 1, 0].

  由於跨度只有 3 年（2010 年，2011 年和 2012 年），因此一熱陣列有三個條目，每年一個。
- 月 —[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0].

  在這裡，一熱陣列在一年中的每個月都有一個條目。
- 平日 —[0, 0, 0, 0, 1, 0, 0].

  ISO 8601 標準指出，星期一是一周的第一天，並且由於 2011 年 4 月 22 日是星期五，所以相應的一個熱平日陣列在第五位置很熱。
- 小時 —[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].

  1 AM 小時是在 24 個成員的一個熱陣列中設置的。

月份中的日期，分鐘和秒不是斷然編碼的。

如果所討論的總datetime範圍僅包含一年內的日期，則不會編碼任何year數組。

您可以使用imputer參數和可用於數datetime值特徵的其中一個策略來指定插入策略來填充缺失值。

Neptune ML 中的自動功能編碼

您可以設定為特徵編碼方法，而不是手動指定用於圖形中屬性auto的特徵編碼方法。Neptune ML 接著會嘗試根據其基礎資料類型推斷每個屬性的最佳功能編碼。

以下是 Neptune ML 在選取適當功能編碼時使用的一些啟發式法：

- 如果屬性只有數值，並且可以轉換為數值資料類型，則 Neptune ML 通常將其編碼為數值。不過，如果內容的唯一值數目小於總值的 10%，且這些唯一值的基數小於 100，則 Neptune ML 會使用分類編碼。
- 如果屬性值可以轉換為datetime類型，則 Neptune ML 將它們編碼為datetime功能。

- 如果屬性值可以強制為布林值 (1/0 或 True /False)，則 Neptune ML 會使用類別編碼。
- 如果屬性是具有 10% 以上唯一值的字串，且每個值的平均記號數大於或等於 3，Neptune ML 會推斷屬性類型為文字，並自動偵測使用的語言。如果檢測到的語言是 fastText (p. 696) 支持的語言之一，即英語，中文，印地語，西班牙語和法語，則 Neptune ML 使用`text_fasttext`對文本進行編碼。另一方面，Neptune ML 使用`text_sbert` (p. 697)。
- 如果該屬性是未歸類為文本功能的字符串，則 Neptune ML 假定它是一個分類功能，並使用類別編碼。
- 如果每個節點對於推斷為類別功能的屬性都有自己的唯一值，Neptune ML 會從訓練圖表中卸除屬性，因為它可能是不會對學習提供資訊的識別碼。
- 如果已知該屬性包含有效的 Neptune 分隔符號，例如分號 (「;」)，則 Neptune ML 只能將屬性視為`MultiNumerical`或`MultiCategorical`。
  - Neptune ML 首先嘗試將值編碼為數字特徵。如果成功，Neptune ML 會使用數字編碼來建立數值向量特徵。
  - 否則，Neptune ML 將值編碼為多分類。
- 如果 Neptune ML 無法推斷屬性值的資料類型，Neptune ML 會從訓練圖表中刪除屬性。

編輯訓練資料組態檔

Neptune 匯出程序會將 Neptune ML 資料從 Neptune 資料庫叢集匯出至 S3 儲存貯體。它將節點和邊分別匯出到nodes/和edges/資料夾中。它也會建立 JSON 訓練資料組態檔案，training-data-configuration.json依預設名稱。此檔案包含有關圖形結構描述、其特徵類型、特徵轉換和標準化作業，以及分類或迴歸工作的目標圖徵的相關資訊。

在某些情況下，您可能想要直接修改組態檔案。其中一種情況是，當您想要變更特徵的處理方式或建構圖形的方式，而不需要在每次您想要修改您要解決之機器學習任務的規格時重新執行匯出。

若要編輯訓練資料組態檔

1. 將檔案下載到您的本機電腦。

   除非您在傳遞至匯出程序的additionalParams/neptune_ml參數中指定了一或多個具名工作，否則檔案將具有預設名稱，即training-data-configuration.json。您可以使用像這樣的AWS CLI 命令來下載文件：

   ```
   aws s3 cp \
     s3://(your Amazon S3 bucket)/(path to your export folder)/training-data-
   configuration.json \
     ./
   ```

2. 使用文字編輯器編輯檔案。
3. 上傳修改後的檔案。使用如下所示的AWS CLI 命令，將修改後的檔案上傳回 Amazon S3 中的相同位置：

   ```
   aws s3 cp \
     training-data-configuration.json \
     s3://(your Amazon S3 bucket)/(path to your export folder)/training-data-
   configuration.json
   ```

JSON 訓練資料設定檔範例

以下是描述節點分類工作圖形的訓練資料組態檔範例：

```
{
  "version" : "v2.0",
  "query_engine" : "gremlin",
  "graph" : [
    {
      "edges" : [
        {
          "file_name" : "edges/(movie)-included_in-(genre).csv",
          "separator" : ",",
          "source" : ["~from", "movie"],
          "relation" : ["", "included_in"],
          "dest" : [ "~to", "genre" ]
        },
        {
          "file_name" : "edges/(user)-rated-(movie).csv",
          "separator" : ",",
          "source" : ["~from", "movie"],
          "relation" : ["rating", "prefixname"], # [prefixname#value]
          "dest" : ["~to", "genre"],
          "features" : [
            {
              "feature" : ["rating", "rating", "numerical"],
              "norm" : "min-max"
```

```
            }
          ]
        }
      ],
      "nodes" : [
        {
          "file_name" : "nodes/genre.csv",
          "separator" : ",",
          "node" : ["~id", "genre"],
          "features" : [
            {
              "feature": ["name", "genre", "category"],
              "separator": ";"
            }
          ]
        },
        {
          "file_name" : "nodes/movie.csv",
          "separator" : ",",
          "node" : ["~id", "movie"],
          "features" : [
            {
              "feature": ["title", "title", "word2vec"],
              "language": ["en_core_web_lg"]
            }
          ]
        },
        {
          "file_name" : "nodes/user.csv",
          "separator" : ",",
          "node" : ["~id", "user"],
          "features" : [
            {
              "feature": ["age", "age", "numerical"],
              "norm" : "min-max",
              "imputation": "median",
            },
            {
              "feature": ["occupation", "occupation", "category"],
            }
          ],
          "labels" : [
            {
              "label": ["gender", "classification"],
              "split_rate" : [0.8, 0.2, 0.0]
            }
          ]
        }
      ]
    },
    "warnings" : [ ]
  ]
}
```

JSON 訓練資料組態檔案的結構

訓練組態檔案是指匯出程序儲存在nodes/和edges/資料夾中的 CSV 檔案。

下的每個文件nodes/存儲有關具有相同屬性圖節點標籤的節點的信息。節點檔案中的每個資料行都會儲存節點識別碼或節點屬性。檔案的第一行包含指定每個欄的~id或屬性名稱的標頭。

下的每個檔案都會edges/儲存有關具有相同屬性圖表邊緣標籤之節點的資訊。節點檔案中的每個資料行都會儲存來源節點識別碼、目標節點識別碼或 Edge 屬性。檔案的第一行包含一個標頭~from，指定每個欄的~to、或屬性名稱。

訓練資料組態檔案具有三個頂層元素：

```
{
  "version" : "v2.0",
  "query_engine" : "gremlin",
  "graph" : [ ... ]
}
```

- version— (字串) 正在使用的組態檔案版本。
- query_engine— (字串) 用於匯出圖形資料的查詢語言。目前，只有「小鬼」是有效的。
- graph— (JSON 陣列) 列出一或多個組態物件，這些物件包含每個將使用的節點和邊緣的模型參數。

  圖形陣列中的配置物件具有下一節所述的結構。

graph陣列中列出的配置物件的內容

graph陣列中的配置物件可以包含三個頂層節點：

```
   {
     "edges"    : [ ... ],
     "nodes"    : [ ... ],
     "warnings" : [ ... ],
   }
```

- edges— (JSON 物件陣列) 每個 JSON 物件都會指定一組參數，以定義在模型處理和訓練期間如何處理圖形中的邊緣。這僅與小鬼引擎一起使用。
- nodes— (JSON 物件陣列) 每個 JSON 物件都會指定一組參數，以定義在模型處理和訓練期間如何處理圖形中的節點。這僅與小鬼引擎一起使用。
- warnings— (JSON 物件陣列) 每個物件都包含在資料匯出程序期間產生的警告。

edges陣列中列出的邊緣配置物件的內容

edges陣列中列出的邊緣配置物件可以包含下列頂層欄位：

```
     {
       "file_name" : "(path to a CSV file)",
       "separator" : "(separator character)",
       "source"    : ["(column label for starting node ID)", "(starting node type)"],
       "relation"  : ["(column label for the relationship name)", "(the prefix name for
 the relationship name)"],
       "dest"      : ["(column label for ending node ID)", "(ending node type)"],
       "features"  : [(array of feature objects)],
       "labels"    : [(array of label objects)]
     }
```

- **file_name**— 字串，指定 CSV 檔案的路徑，此檔案儲存有關具有相同屬性圖標籤之邊緣的資訊。

  該檔案的第一行包含欄標籤的標題行。

  前兩欄標籤為~from和~to。第一列（~from列）存儲邊緣的起始節點的 ID，第二列（~to列）存儲邊緣的結束節點的 ID。

  標題行中的其餘欄標示會為每個剩餘欄指定已匯出至該欄的邊性質的名稱。
- **separator**— 包含分隔符號的字串，可分隔該 CSV 檔案中的欄。

- **source**— 包含指定邊緣起始節點的兩個字串的 JSON 陣列。第一個字串包含儲存起始節點 ID 之資料行的標頭名稱。第二個字符串指定節點類型。
- **relation**— 包含指定邊緣關係類型的兩個字串的 JSON 陣列。第一個字符串包含關係名稱 (`relname`) 存儲在列的頭名。第二個字串包含關係名稱的前置詞 (`prefixname`)。

  完整的關係類型由兩個字符串組合，它們之間有一個連字符，如下所示：*`prefixname`*`-`*`relname`*。

  如果第一個字符串是空的，所有邊緣具有相同的關係類型，即字`prefixname`符串。
- **dest**— 包含指定邊緣結束節點的兩個字串的 JSON 陣列。第一個字串包含儲存節點 ID 之資料行的標頭名稱。第二個字符串指定節點類型。
- **features**— 屬性值功能物件的 JSON 陣列。每個屬性值圖徵物件包含下列欄位：
  - 功能-由三個字符串組成的 JSON 數組。第一個字串包含包含屬性值之欄的標頭名稱。第二個字串包含特徵名稱。第三個字串包含特徵類型。
  - norm — (選擇性) 指定要套用至屬性值的標準化方法。
- **labels**— 物件的 JSON 陣列。每個物件都會定義邊緣的目標特徵，並指定訓練和驗證階段應採取的邊緣比例。每個物件包含下列欄位：
  - 標籤-由兩個字符串組成的 JSON 數組。第一個字串包含包含目標圖徵性質值的欄的標頭名稱。第二個字串會指定下列其中一種目標工作類型：
    - `"classification"`-邊分類任務。`label`陣列中第一個字串所識別之資料行中提供的屬性值會被視為分類值。對於邊緣分類工作，`label`陣列中的第一個字串不能為空白。
    - `"regression"`— 邊迴歸任務。由`label`陣列中第一個字串所識別的欄中提供的性質值會被視為數值。對於邊緣迴歸工作，`label`陣列中的第一個字串不能為空。
    - `"link_prediction"`-鏈接預測任務。不需要屬性值。對於連結預測工作，會忽略`label`陣列中的第一個字串。
  - **split_rate**— JSON 陣列，其中包含三個介於零和一之間的數字，加起來最多為一個，並代表訓練、驗證和測試階段將分別使用的節點比例估計。此欄位或`custom_split_filenames`可以定義，但不能同時定義兩者。請參閱拆分率 (p. 676)。
  - **custom_split_filenames**— JSON 物件，指定定義訓練、驗證和測試種群的檔案名稱。`split_rate`可以同時定義和定義。但不能同時定義。如需詳細資訊，請參閱 自訂 train-validation-test 比例 (p. 705)。

nodes陣列中列出的節點配置物件的內容

列出的節點配置物件可包含下`nodes`列欄位：

```
    {
      "file_name" : "(path to a CSV file)",
      "separator" : "(separator character)",
      "node"      : ["(column label for the node ID)", "(node type)"],
      "features"  : [(feature array)],
      "labels"    : [(label array)],
    }
```

- **file_name**— 字串，指定 CSV 檔案的路徑，此檔案儲存有關具有相同屬性圖標籤之節點的資訊。

  該檔案的第一行包含欄標籤的標題行。

  第一列標籤是`~id`，第一列 (`~id`列) 存儲節點 ID。

  標題行中的其餘欄標示會為每個剩餘欄指定節點屬性的名稱，其值已匯出至該欄。
- **separator**— 包含分隔符號的字串，可分隔該 CSV 檔案中的欄。
- **node**— 包含兩個字串的 JSON 陣列。第一個字串包含儲存節點 ID 之資料行的標頭名稱。第二個字符串指定圖形中的節點類型，該類型對應於節點的屬性圖標籤。

- **features**— 節點功能物件的 JSON 陣列。請參閱 列示在節點或邊的features陣列中的圖徵物件的內容 (p. 704)。
- **labels**— 節點標籤物件的 JSON 陣列。請參閱 節點labels陣列中列出的節點標籤物件的內容 (p. 705)。

列示在節點或邊的features陣列中的圖徵物件的內容

節點features陣列中列示的節點特徵物件可以包含下列頂層欄位：

- **feature**— 由三個字串組成的 JSON 陣列。第一個字串包含包含特徵性質值之欄的標頭名稱。第二個字串包含特徵名稱。

  第三個字串包含特徵類型。中列出了有效的特徵類型圖徵type欄位的可能值 (p. 679)。
- **norm**— 此欄位對於數值圖徵而言是必需的。它指定了對數值使用的規範化方法。有效值為"none""min-max"、和「標準」。如需詳細資訊，請參閱 該norm領域 (p. 680)。
- **language**— 語言欄位會指定文字屬性值中使用的語言。它的用法取決於文本編碼方法：
  - 對於text_fasttext (p. 696)編碼，此欄位為必填欄位，且必須指定下列其中一種語言：
    - en(英文)
    - zh(中文)
    - hi(印地語)
    - es(西班牙文)
    - fr(法文)

    但是，一次text_fasttext無法處理多種語言。
  - 對於text_sbert (p. 696)編碼，不使用此欄位，因為 SBERT 編碼是多語言的。
  - 對於text_word2vec (p. 697)編碼，此字段是可選的，因為text_word2vec僅支持英文。如果存在，則必須指定英語模型的名稱：

    ```
    "language" : "en_core_web_lg"
    ```
  - 對於tfidf (p. 697)編碼，不使用此字段。
- **max_length**— 對於text_fasttext (p. 696)功能而言，此欄位是選用的，其中指定要編碼的輸入文字功能中的最大記號數目。到達之後max_length的輸入文字會被忽略。例如，將 max_length 設定為 128 表示會忽略文字序列中 128 之後的任何記號。
- **separator**— 此欄位可選擇性地與categorynumerical和auto功能搭配使用。它指定可用於將屬性值拆分為多個分類值或數值的字符。

  請參閱 該separator領域 (p. 681)。
- **range**— 此欄位是bucket_numerical功能的必要欄位。它指定要被分成桶的數值的範圍。

  請參閱 該range領域 (p. 681)。
- **bucket_cnt**— 此欄位是bucket_numerical功能的必要欄位。它指定了由range參數定義的數字範圍應分成的桶的數量。

  請參閱 Neptune ML 中的值區數值特徵 (p. 695)。
- **slide_window_size**— 此欄位可選擇搭配bucket_numerical功能使用，以將值指派給多個值區。

  請參閱 該slide_window_size領域 (p. 682)。
- **imputer**— 此欄位可選擇與numericalbucket_numerical、和datetime功能搭配使用，以提供填入缺少值的補充技術。支援的插補技術為"mean""median"、和"most_frequent"。

  請參閱 該imputer領域 (p. 682)。
- **max_features**—text_tfidf 功能可選擇使用此欄位，以指定要編碼的最大術語數。

請參閱 該max_features領域 (p. 682)。

- **min_df**—text_tfidf 功能可選擇使用此欄位，以指定要編碼之術語的最小文件頻率

  請參閱 該min_df領域 (p. 682)。
- **ngram_range**— 此欄位可由text_tfidf功能選擇性地使用，以指定要視為要編碼的潛在個別字詞數或標記數目範圍

  請參閱 該ngram_range領域 (p. 683)。
- **datetime_parts**—datetime 功能選擇性地使用此欄位，以指定要分類編碼 datetime 值的哪些部分。

  請參閱 該datetime_parts領域 (p. 683)。

節點labels陣列中列出的節點標籤物件的內容

節點labels陣列中列出的標籤物件會定義節點目標功能，並指定訓練、驗證和測試階段將使用的節點比例。每個物件可包含下列欄位：

```
    {
      "label"      : ["(column label for the target feature property value)", "(task
type)"],
      "split_rate" : [(training proportion), (validation proportion), (test proportion)],
      "custom_split_filenames" : {"train": "(training file name)", "valid": "(validation
file name)", "test": "(test file name)"},
      "separator"  : "(separator character for node-classification category values)",
    }
```

- **label**— 包含兩個字串的 JSON 陣列。第一個字串包含儲存圖徵性質值的欄的標頭名稱。第二個字串指定目標工作類型，可以是：
  - "classification"-節點分類任務。指定欄中的性質值用於建立分類圖徵。
  - "regression"— 節點回歸任務。指定欄中的性質值用於建立數字圖徵。
- **split_rate**— JSON 陣列，其中包含三個介於零和一之間的數字，加起來最多一個，並分別表示訓練、驗證和測試階段將使用的節點比例估計。請參閱 分割率 (p. 676)。
- **custom_split_filenames**— JSON 物件，指定定義訓練、驗證和測試種群的檔案名稱。split_rate可以同時定義和定義。但不能同時定義。如需詳細資訊，請參閱 自訂 train-validation-test 比例 (p. 705)。
- **separator**— 包含分隔符號的字串，可分隔分類任務的分類特徵值。

> **Note**
>
> 如果邊緣和節點都沒有提供標籤物件，則會自動假設工作為連結預測，且邊緣會隨機分割成 90% 以進行訓練，10% 用於驗證。

自訂 train-validation-test 比例

依預設，Neptune ML 會使用此split_rate參數，以使用此參數中定義的比例，將圖形隨機分割為訓練、驗證和測試人口。若要更精確地控制在這些不同族群中使用哪些實體，可以建立明確定義它們的檔案，然後編輯訓練資料組態檔案，將 (p. 700)這些索引檔案對應至人口。此對應是由訓練組態檔案中custom_split_filesnames (p. 705)金鑰的 JSON 物件所指定。如果使用此選項，則必須為和金鑰提供檔案名稱，train而且對於validation金鑰而言是選擇性的test。

這些文件的格式應與 Gremlin 數據格式 (p. 262)匹配。特別是，對於節點層級任務，每個文件應包含一個帶有~id標題的列，該列出節點 ID，對於邊級任務，文件應分別指定~from並~to指示邊緣的源和目標節點。這些檔案必須放置在與用於資料處理的匯出資料相同的 Amazon S3 位置 (請參閱：outputS3Path (p. 501))。

對於屬性分類或回歸工作，這些檔案可以選擇性地定義機器學習工作的標籤。在這種情況下，檔案必須具有與訓練資料組態檔案中定義 (p. 705)的標頭名稱相同的屬性欄。如果在匯出的節點和邊緣檔案和自訂分割檔案中都定義了性質標籤，則自訂分割檔案的優先順序將優先考慮。

使用 Neptune ML 訓練模型

處理從 Neptune 匯出的資料以進行模型訓練之後，您可以使用curl (或awscurl) 命令啟動模型訓練工作，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
      }'
```

有關如何使用此命令的詳細資訊模型訓練指令 (p. 757)，以及有關如何取得執行中工作狀態、如何停止執行中工作以及如何列出所有執行中工作的資訊。

您也可以提供一個previousModelTrainingJobId來使用已完成 Neptune ML 模型訓練工作的資訊，以加速新訓練工作中的超參數搜尋。在對新圖形資料進行模型重新訓練，以及對相同圖形資料 (p. 668)進行增量訓練 (p. 668)時，此功能非常有用。使用像這樣的命令：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
        "previousModelTrainingJobId" : "(the model-training job-id of a completed job)"
      }'
```

您可以提供customModelTrainingParameters物件，在 Neptune ML 訓練基礎結構上訓練自己的模型實作，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
        "modelName": "custom",
        "customModelTrainingParameters" : {
          "sourceS3DirectoryPath": "s3://(your Amazon S3 bucket)/(path to your Python
 module)",
          "trainingEntryPointScript": "(your training script entry-point name in the Python
 module)",
          "transformEntryPointScript": "(your transform script entry-point name in the
 Python module)"
        }
      }'
```

如模型訓練指令 (p. 757)需有關如何取得執行中工作狀態、如何停止執行中工作以及如何列出所有執行中工作的詳細資訊，請參閱。如需有Neptune ML 中的自定義模型 (p. 724)關如何實作和使用自訂模型的資訊，請參閱。

主題

- Amazon Neptune ML 中的模型和模型培訓 (p. 709)
- 在 Neptune ML 中自訂模型超參數組態 (p. 711)
- 模型訓練最佳實務 (p. 719)

Amazon Neptune ML 中的模型和模型培訓

Neptune ML 使用圖形神經網絡（GNN）為各種機器學習任務創建模型。圖形神經網絡已被證明可以獲得圖形機器學習任務的 state-of-the-art結果，並且在從圖形結構化數據中提取信息模式方面非常出色。

Neptune ML 中的圖形神經網絡（GNN）

圖形神經網絡（GNN）屬於一系列神經網絡，它們通過考慮附近節點的結構和特徵來計算節點表示。GNN 可補充其他不適合圖形資料的傳統機器學習和神經網路方法。

GNN 用於解決機器學習任務，例如節點分類和回歸（預測節點的屬性）以及邊緣分類和回歸（預測邊緣的屬性）或鏈路預測（預測圖形中的兩個節點是否應該連接）。

一般而言，將 GNN 用於機器學習任務涉及兩個階段：

- 一種編碼階段，其中 GNN 計算圖中每個節點的 d 維向量。這些向量也稱為表示或嵌入。
- 解碼階段，這使得基於所述編碼表示的預測。

對於節點分類和迴歸，節點表現法可直接用於分類和回歸工作。對於邊分類和迴歸，邊上入射節點的節點表示會用作分類或迴歸的輸入。對於連結預測，邊緣可能性分數是使用一對節點表示法和一個邊類型表現法來計算。

深度圖庫（DGL）促進了 GNN 的有效定義和培訓這些任務。

不同的 GNN 模型消息傳遞的制定下統一。在此檢視中，圖表中節點的表示方式是使用節點鄰居的表示法 (訊息)，以及節點的初始表示法來計算。在 NeptUnem 中，節點的初始表示是從其節點屬性中提取的特徵派生的，或者是可學習的，並且取決於節點的身份。

Neptune ML 還提供了連接節點功能和可學習的節點表示，以充當原始節點表示的選項。

對於 Neptune ML 涉及與節點屬性的圖形的各種任務，我們使用關係圖卷積網絡（R-GCN））來執行編碼階段。R-GCN 是 GNN 架構，非常適合具有多個節點和邊緣類型的圖形（這些類型稱為異構圖）。

R-GCN 網絡由固定數量的層組成，一個接一個地堆疊。R-GCN 的每一層都使用其可學習的模型參數從即時聚合資訊, 一個節點的 1 跳鄰里. 由於後續層使用前一層的輸出表示作為輸入, 影響節點的最終嵌入的圖形鄰域的半徑取決於層的數目 (`num-layer`), R-GCN 網絡的.

例如，這表示雙層網路會使用兩個躍點之外的節點中的資訊。

要了解有關 GNN 的更多信息，請參閱圖形神經網絡的綜合調查。有關深圖庫（DGL）的更多信息，請訪問 DGL 網頁。如需將 DGL 與 GNN 搭配使用的實作教學課程，請參閱使用深度圖庫學習圖形神經網路。

訓練圖表神經網路

在機器學習中，獲取模型以學習如何對任務進行良好預測的過程稱為模型訓練。這通常是透過指定要最佳化的特定目標，以及用來執行此最佳化的演算法來執行。

這個過程用於培訓 GNN 來學習下游任務的良好表示為好。我們為該任務創建一個客觀函數，該任務在模型訓練期間將其最小化。例如，對於節點分類，我們使用CrossEntropyLoss作為目標，它懲罰錯誤的分類，對於節點回歸，我們最小化MeanSquareError。

目標通常是一個損失函數，它採用特定數據點的模型預測，並將它們與該數據點的地面真值進行比較。它會傳回損失值，顯示模型的預測距離有多遠。訓練程序的目標是將損失降到最低，並確保模型預測接近地面真相。

深度學習中用於訓練過程的最佳化演算法通常是梯度下降的變體。在 Neptune ML 中，我們使用 Adam，這是一種基於低階時刻的適應性估計隨機目標函數的一階梯度最佳化演算法。

雖然模型訓練程序會嘗試確保學習到的模型參數接近目標函數的最小值，但模型的整體效能也取決於模型的超參數，而這些參數是訓練演算法未學習的模型設定。例如，學習到的節點表示法的維度是會影響模型效能的超參數。num-hidden因此，在機器學習中，通常會執行超參數最佳化 (HPO) 來選擇合適的超參數。

Neptune ML 使用 SageMaker 超參數調整工作來啟動具有不同超參數組態的多個模型訓練執行個體，以嘗試為一系列超參數設定尋找最佳模型。請參閱 在 Neptune ML 中自訂模型超參數組態 (p. 711)。

Neptune ML 中的知識圖嵌入模型

知識圖 (KG) 是編碼關於不同實體 (節點) 及其關係 (邊) 資訊的圖形。在 Neptune ML 中，當圖形不包含節點屬性（僅與其他節點的關係）時，預設會套用知識圖嵌入模型以執行連結預測。雖然, R-GCN 模型與可學習嵌入也可以通過指定模型類型作為這些圖用於這些圖"rgcn", 知識圖嵌入模型更簡單，被設計為有效的學習表示為大規模知識圖.

在連結預測任務中使用知識圖嵌入模型，以預測完成三重節點的節點或關係，(**h, r, t**)其中**h**是來源節點，**r**是關係類型，並且**t**是目標節點。

在 Neptune ML 中實現的知識圖嵌入模型是distmulttransE、和rotatE。若要深入瞭解知識圖形嵌入模型，請參閱 DGL-KE。

在 Neptune ML 中訓練自訂模型

Neptune ML 可讓您針對特定案例定義和實作自己的自訂模型。如Neptune ML 中的自定義模型 (p. 724)需如何實作自訂模型以及如何使用 Neptune ML 基礎結構進行訓練的相關資訊，請參閱。

在 Neptune ML 中自訂模型超參數組態

當您啟動 Neptune ML 模型訓練工作時，Neptune ML 會自動使用從先前資料處理 (p. 694)工作推斷出來的資訊。它會使用這些資訊來產生超參數組態範圍，這些範圍可用來建立SageMaker 超參數調整工作，以訓練您的工作的多個模型。這樣，您就不必為要訓練的模型指定一長串超參數值。而是根據工作類型、圖形類型和調整工作設定來選取模型超參數範圍和預設值。

不過，您也可以修改資料處理工作所產生的 JSON 組態檔，覆寫預設的超參數組態並提供自訂超參數。

使用 Neptune ML 模型訓練 API (p. 757)，您可以控制數個高階超參數調整工作設定maxHPONumberOfTrainingJobs，例如maxHPOParallelTrainingJobs、和trainingInstanceType。若要對模型超參數進行更精細的控制，您可以自訂資料處理工作所產生的model-HPO-configuration.json檔案。檔案會儲存在您為處理任務輸出指定的 Amazon S3 位置。

您可以下載檔案、編輯檔案以覆寫預設的超參數組態，然後將檔案上傳回相同的 Amazon S3 位置。請勿變更檔案名稱，編輯時請小心遵循這些指示。

若要從 Amazon S3 下載檔案：

```
aws s3 cp \
  s3://(bucket name)/(path to output folder)/model-HPO-configuration.json \
  ./
```

完成編輯後，請將檔案上傳回原來的位置：

```
aws s3 cp \
  model-HPO-configuration.json \
  s3://(bucket name)/(path to output folder)/model-HPO-configuration.json
```

model-HPO-configuration.json檔案結構

此model-HPO-configuration.json檔案會指定要訓練的模型、機器學習以task_type及應變更或修正的各種模型訓練執行的超參數。

超參數會歸類為屬於不同的層級，這些層級表示呼叫超參數調整工作時給予超參數的優先順序：

- 第 1 層超參數具有最高優先順序。如果您設定maxHPONumberOfTrainingJobs為小於 10 的值，則只會調整第 1 層超參數，其餘的則採用其預設值。
- 第 2 層超參數的優先順序較低，因此，如果調整工作的訓練工作總數超過 10 個，但少於 50 個，則會同時調整第 1 層和第 2 層超參數。
- 只有當您的訓練工作總數超過 50 項時，第 3 層超參數才會與第 1 層和第 2 層一起調整。
- 最後，固定的超參數完全不會進行調整，並且始終採用其預設值。

model-HPO-configuration.json檔案範例

以下是範例model-HPO-configuration.json檔案：

```
{
  "models": [
    {
      "model": "rgcn",
      "task_type": "node_class",
      "eval_metric": {
        "metric": "acc"
```

```
    },
    "eval_frequency": {
        "type":  "evaluate_every_epoch",
        "value":  1
    },
    "1-tier-param": [
      {
          "param": "num-hidden",
          "range": [16, 128],
          "type": "int",
          "inc_strategy": "power2"
      },
      {
        "param": "num-epochs",
        "range": [3,30],
        "inc_strategy": "linear",
        "inc_val": 1,
        "type": "int",
        "node_strategy": "perM"
      },
      {
        "param": "lr",
        "range": [0.001,0.01],
        "type": "float",
        "inc_strategy": "log"
      }
    ],
    "2-tier-param": [
      {
        "param": "dropout",
        "range": [0.0,0.5],
        "inc_strategy": "linear",
        "type": "float",
        "default": 0.3
      },
      {
        "param": "layer-norm",
        "type": "bool",
        "default": true
      }
    ],
    "3-tier-param": [
      {
        "param": "batch-size",
        "range": [128, 4096],
        "inc_strategy": "power2",
        "type": "int",
        "default": 1024
      },
      {
        "param": "fanout",
        "type": "int",
        "options": [[10, 30],[15, 30], [15, 30]],
        "default": [10, 15, 15]
      },
      {
        "param": "num-layer",
        "range": [1, 3],
        "inc_strategy": "linear",
        "inc_val": 1,
        "type": "int",
        "default": 2
      },
      {
        "param": "num-bases",
        "range": [0, 8],
```

```
          "inc_strategy": "linear",
          "inc_val": 2,
          "type": "int",
          "default": 0
        }
      ],
      "fixed-param": [
        {
          "param": "concat-node-embed",
          "type": "bool",
          "default": true
        },
        {
          "param": "use-self-loop",
          "type": "bool",
          "default": true
        },
        {
          "param": "low-mem",
          "type": "bool",
          "default": true
        },
        {
          "param": "l2norm",
          "type": "float",
          "default": 0
        }
      ]
    }
  ]
}
```

model-HPO-configuration.json檔案的元素

該文件包含一個 JSON 對象，其中包含一個名為的單個頂級數組，models該數組包含單一模型配置對象。自訂檔案時，請確定models陣列中只有一個模型配置物件。如果您的檔案包含多個模型組態物件，調整工作將會失敗並顯示警告。

模型配置物件包含下列頂層元素：

- **model**— (字串) 要訓練的模型類型 (請勿修改)。有效值為：
  - "rgcn"— 這是節點分類和回歸工作以及異質連結預測工作的預設值。
  - "transe"— 這是 KGE 連結預測工作的預設值。
  - "distmult"— 這是 KGE 連結預測工作的替代模型類型。
  - "rotate"— 這是 KGE 連結預測工作的替代模型類型。

  通常不要直接修改model值，因為不同的模型類型通常具有實質上不同的適用超參數，這可能會在訓練工作開始後導致剖析錯誤。

  若要變更模型類型，請使用模型訓練 API (p. 757) 中的modelName參數，而不是在model-HPO-configuration.json檔案中加以變更。

  變更模型類型並進行微調超參數變更的方法是複製您要使用的模型的預設模型組態範本，然後將其貼到model-HPO-configuration.json檔案中。如果推斷的任務類型支援多個模型，則會hpo-configuration-templates在與model-HPO-configuration.json檔案相同的 Amazon S3 位置命名一個資料夾。此資料夾包含適用於工作的其他模型的所有預設超參數組態。

  例如，如果您想要將KGE連結預測任務的模型和超參數組態從預設模型變更為transedistmult模型，只需將檔案內容貼到hpo-configuration-templates/distmult.json檔案中model-HPO-configuration.json，然後視需要編輯超參數即可。

Note

如果您在modelTraining API 中設定modelName參數，同時變更model-HPO-configuration.json檔案中的model和超參數規格，且這些規格不同，則model-HPO-configuration.json檔案中的model值優先順序，且會忽略該modelName值。

- **task_type**— (字串) 由資料處理工作推斷或直接傳遞至資料處理工作的機器學習工作類型 (請勿修改)。有效值為：
  - "node_class"
  - "node_regression"
  - "link_prediction"

  資料處理工作會檢查匯出的資料集以及資料集屬性所產生的訓練工作組態檔案，以推斷工作類型。

  此值不得變更。如果您想要訓練不同的工作，則需要執行新的資料處理工作 (p. 694)。如果task_type值不是您預期的值，您應該檢查資料處理工作的輸入，以確定它們是正確的。這包括modelTraining API 的參數，以及資料匯出程序所產生的訓練工作組態檔案中的參數。
- **eval_metric**— (字串) 評估量度應用於評估模型效能，以及在 HPO 執行期間選取效能最佳的模型。有效值為：
  - "acc"— 標準分類精度。這是單一標籤分類工作的預設值，除非在資料處理期間發現不平衡的標籤，在此情況下，預設值為"F1"。
  - "acc_topk"— 正確標籤位於最高**k**預測的次數。您也可以通過topk作為額外鍵傳入**k**來設置該值。
  - "F1"— F1 得分。
  - "mse"— 回歸工作的平方誤差度量。
  - "mrr"-平均倒數排名指標。
  - "precision"— 模型精確度，計算方式為真正值與預測正值的比例：= true-positives / (true-positives + false-positives)。
  - "recall"— 模型召回，計算為真正值與實際正值的比例：= true-positives / (true-positives + false-negatives)。
  - "roc_auc"— ROC 曲線下方的區域。這是多標籤分類的預設值。

  例如，若要將量度變更為F1，請依照下列方式變更eval_metric值：

```
"  eval_metric": {
    "metric": "F1",
  },
```

  或者，若要將量度變更為topk準確度分數，您可以變更eval_metric如下：

```
  "eval_metric": {
    "metric": "acc_topk",
    "topk": 2
  },
```

- **eval_frequency**— (物件) 指定在訓練驗證集上模型效能期間應檢查的頻率。根據驗證效能，然後可以啟動提前停止，並儲存最佳模型。

  該eval_frequency對象包含兩個元素，即"type"和"value"。例如：

```
  "eval_frequency": {
    "type":  "evaluate_every_pct",
    "value":  0.1
  },
```

有效type值為：

- **evaluate_every_pct**— 指定每個評估要完成的訓練百分比。

  對於evaluate_every_pct，"value"欄位包含一個介於零和一之間的浮點數，表示該百分比。

- **evaluate_every_batch**— 指定每個評估要完成的訓練批次數。

  對於evaluate_every_batch，"value"欄位包含表示該批次計數的整數。

- **evaluate_every_epoch**— 指定每個評估的時代數目，其中新紀元從午夜開始。

  對於evaluate_every_epoch，該"value"字段包含一個表示該紀元計數的整數。

的預設設定eval_frequency為：

```
"eval_frequency": {
  "type":  "evaluate_every_epoch",
  "value":  1
},
```

- **1-tier-param**— (必要) 第 1 層超參數的陣列。

  如果您不想調整任何超參數，您可以將其設為空的陣列。這不會影響 SageMaker 超參數調整工作所啟動的訓練工作總數。這只是意味著所有的訓練作業，如果有超過 1，但少於 10，將使用相同的一組超參數運行。

  另一方面，如果你想處理所有的可調超參數具有相同的意義，那麼你可以把所有的超參數在這個數組中。

- **2-tier-param**— (必要) 第 2 層超參數的陣列。

  這些參數只maxHPONumberOfTrainingJobs有在值大於 10 時才會進行調整。否則，它們會固定為預設值。

  如果您的訓練預算最多為 10 個訓練工作，或者因為任何其他原因不想要 Tier-2 超參數，但是您想要調整所有可調整的超參數，則可以將其設定為空陣列。

- **3-tier-param**— (必要) 第 3 層超參數的陣列。

  這些參數只maxHPONumberOfTrainingJobs有在值大於 50 時才會進行調整。否則，它們會固定為預設值。

  如果您不想要 Tier-3 超參數，則可以將其設置為空數組。

- **fixed-param**— (必要) 固定超參數陣列，這些參數僅採用其預設值，且在不同的訓練工作中不會有所不同。

  如果您想要變更所有超參數，可以將其設定為空白陣列，然後將值設定為足以改變所有層的maxHPONumberOfTrainingJobs大小，或是製作所有超參數第 1 層。

代表、、中每個超參數的 JSON 物件1-tier-param2-tier-param3-tier-param，並fixed-param包含下列元素：

- **param**— (字串) 超參數的名稱 (請勿變更)。

  請參閱 Neptune ML 中有效的超參數名稱清 (p. 716)單。

- **type**— (字串) 超參數類型 (請勿變更)。

  有效類型為：boolint、和float。

- **default**— (字串) 超參數的預設值。

您可以設定新的預設值。

可調整的超參數也可以包含下列元素：

- **range**— (陣列) 連續可調式超參數的範圍。

  這應該是具有兩個值的數組，即最小值和最大值 range (`[min, max]`) 。
- **options**— (陣列) 可分類可調超參數的選項。

  這個數組應該包含所有要考慮的選項：

  ```
  "options" : [value1, value2, ... valuen]
  ```
- **inc_strategy**— (字串) 連續可調式超參數範圍的增量變更類型 (不變更)。

  有效值為 `log`、`linear` 和 `power2`。僅適用於僅適用於範圍鍵設定時。

  修改此選項可能會導致未使用完整範圍的超參數進行調整。
- **inc_val**— (浮動) 連續增量對於連續可調參數不同的量 (請勿變更)。

  僅適用於僅適用於範圍鍵設定時。

  修改此選項可能會導致未使用完整範圍的超參數進行調整。
- **node_strategy**— (字串) 表示此超參數的有效範圍應根據圖形中的節點數目而變更 (請勿變更)。

  有效值為`"perM"` (每百萬) ，`"per10M"` (每 10 百萬) 和`"per100M"` (每 1 億) 。

  而不是變更此值，`range`而是變更。
- **edge_strategy**— (字串) 表示此超參數的有效範圍應根據圖形中的邊數而變更 (請勿變更)。

  有效值為`"perM"` (每百萬) ，`"per10M"` (每 10 百萬) 和`"per100M"` (每 1 億) 。

  而不是變更此值，`range`而是變更。

Neptune ML 中所有超參數的列表

下列清單包含可針對任何模型類型和工作在 Neptune ML 中任何位置設定的所有超參數。因為它們並非全部適用於每種模型類型，因此請務必只在您使用的模型範本中顯示的`model-HPO-configuration.json`檔案中設定超參數。

- **batch-size**— 在一個正向傳遞中使用的批次目標節點的大小。類型：`int`。

  將此值設定為較大的值可能會導致 GPU 執行個體訓練時出現記憶體問題。
- **concat-node-embed**— 指示是否透過將節點處理的特徵與可學習的初始節點嵌入連接來取得節點的初始表示法，以增加模型的表示性。類型：`bool`。
- **dropout**— 套用至脫離圖層的輟學概率。類型：`float`。
- **edge-num-hidden**— 邊緣功能模塊的隱藏圖層大小或單位數量。僅在設定`use-edge-features`為時使用`True`。類型：浮動。
- **enable-early-stop**— 切換是否使用提前停止功能。類型：`bool`。預設：`true`。

  使用此布林參數可關閉提前停止功能。
- **fanout**— 鄰近取樣期間為目標節點取樣的鄰近節點數。類型：`int`。

此值與緊密結合，num-layers並且應始終位於相同的超參數層中。這是因為您可以為每個潛在的 GNN 圖層指定發散。

因為這個超參數可能會造成模型效能差異很大，所以應該固定或設定為第 2 層或 Tier-3 超參數。將其設定為較大的值可能會導致 GPU 執行個體訓練時出現記憶體問題。

- **gamma**— 分數函數中的邊距值。類型：float。

這僅適用於KGE連結預測模型。

- **l2norm**— 優化器中使用的重量衰減值，對權重施加 L2 標準化懲罰。類型：bool。
- **layer-norm**— 指示是否對rgcn模型使用圖層標準化。類型：bool。
- **low-mem**— 指示是否以犧牲速度為代價，使用關係訊息傳遞函數的低記憶體實作。類型：bool。
- **lr**— 學習速度。類型：float。

這應該設定為 Tier-1 超參數。

- **neg-share**— 在連結預測中，指出正取樣邊是否可以共用負邊取樣。類型：bool。
- **num-bases**— 模rgcn型中基礎分解的基數。使用num-bases該值小於圖形中的邊類型數量，可作為rgcn模型的規則化器。類型：int。
- **num-epochs**— 訓練運行的時代數量。類型：int。

一個時代是一個完整的訓練通過圖形。

- **num-hidden**— 隱藏的圖層大小或單位數量。類型：int。

這也會設定無特徵節點的初始嵌入大小。

將此值設定為較大的值而不減少，batch-size可能會導致 GPU 執行個體的訓練 out-of-memory 問題。

- **num-layer**— 模型中 GNN 圖層的數量。類型：int。

此值與扇出參數緊密結合，並且應該在相同的超參數層中設定扇出之後才出現。

由於這可能會導致模型效能差異很大，因此應將其固定或設定為 Tier-2 或 Tier-3 超參數。

- **num-negs**— 在鏈路預測中，每個正面樣品的負樣本數量。類型：int。
- **per-feat-name-embed**— 指示是否在組合特徵之前透過獨立轉換來嵌入每個特徵。類型：bool。

當設定為時true，將節點的所有轉換特徵連接起來並進一步轉換為尺寸之前，將每個節點的每個特徵獨立轉換為固定尺num_hidden寸大小。

當設定為時false，會連接特徵，而不會有任何特徵特定的轉換。

- **regularization-coef**— 在鏈路預測中，正規化損失係數。類型：float。
- **rel-part**— 指示是否使用關係分割區進行KGE連結預測。類型：bool。
- **sparse-lr**— 學習節點嵌入的學習速率。類型：float。

可學習的初始節點嵌入用於沒有特徵或設置時concat-node-embed的節點。稀疏可學習節點嵌入層的參數是使用單獨的優化器進行訓練，該優化器可以具有單獨的學習速率。

- **use-class-weight**— 指示是否針對不平衡的分類工作套用類別權重。如果設定為true，則會使用標籤計數來設定每個類別標籤的權重。類型：bool。
- **use-edge-features**— 指示是否在訊息傳遞期間使用邊特徵。如果設定為true，則會將自訂邊緣圖徵模組新增至具有圖徵的邊類型的 RGCN 圖層。類型：bool。
- **use-self-loop**— 指示是否在訓練rgcn模型中包含自我迴圈。類型：bool。
- **window-for-early-stop**— 控制最新驗證分數的平均數，以決定提前停止。預設為 3。型別為整數。另請參閱 早期停止 Neptune ML 中的模型訓練程序 (p. 720)。類型：int。預設：3。

請參閱 。

在 Neptune ML 中自定義超參數

當您編輯model-HPO-configuration.json檔案時，以下是最常見的變更類型：

- 編輯range超參數的最小值和/或最大值。
- 將超參數設定為固定值，方法是將超參數移至fixed-param區段，並將其預設值設定為您希望使用的固定值。
- 變更超參數的優先順序，方法是將超參數置於特定階層中、編輯其範圍，並確定已正確設定其預設值。

模型訓練最佳實務

您可以採取一些措施來改善 Neptune ML 模型的效能。

選擇正確的節點屬性

並非圖表中的所有屬性都有意義或與您的機器學習任務相關。在資料匯出期間，應排除任何不相關的性質。

以下是一些最佳實務做法：

- 使用領域專家來協助評估功能的重要性，以及將其用於預測的可行性。
- 刪除您確定是多餘的或無關的功能，以減少數據中的噪音和不重要的相關性。
- 在構建模型時迭代。隨時調整功能、功能組合和調整目標。

Amazon Machine Learning 開發人員指南中的功能處理提供與 Neptune ML 相關的功能處理的其他準則。

處理離群值資料點

離群值是與剩餘資料明顯不同的資料點。資料異常值可能會破壞或誤導訓練程序，導致訓練時間較長或不精確的模型。除非它們真的很重要，否則您應該在匯出資料之前消除異常值。

移除重複的節點和邊

存儲在 Neptune 圖可能具有重複的節點或邊緣。這些冗餘元素將為 ML 模型訓練引入噪音。在匯出資料之前消除重複的節點或邊緣。

調整圖形結構

匯出圖形時，您可以變更特徵的處理方式及建構圖形的方式，以改善模型效能。

以下是一些最佳實務做法：

- 當邊緣屬性具有邊品類的意義時，在某些情況下，值得將其轉換為邊類型。
- 用於數值屬性的預設標準化政策是`min-max`，但在某些情況下，其他規範化政策效果更好。您可以預先處理屬性並變更標準化政策，如中所述`model-HPO-configuration.json`檔案的元素 (p. 713)。
- 匯出程序會根據性質類型自動產生圖徵類型。例如，它將`String`性質視為分類特徵，`Float`並將`Int`性質視為數值特徵。如有需要，可在匯出後修改特徵類型 (請參閱`model-HPO-configuration.json`檔案的元素 (p. 713))。

調整超參數範圍和預設值

資料處理作業會從圖形推斷超參數組態範圍。如果產生的模型超參數範圍和預設值不適用於圖形資料，您可以編輯 HPO 組態檔案，以建立您自己的超參數調整策略。

以下是一些最佳實務做法：

- 當圖形變大時，預設隱藏的尺寸大小可能不夠大，無法包含所有資訊。您可以變更`num-hidden`超參數來控制隱藏的標註大小。
- 對於知識圖嵌入（KGE）模型，您可能需要根據圖形結構和預算更改正在使用的特定模型。

  `TrainsE`模型在處理 one-to-many (1-N)、 many-to-one (N-1) 和 many-to-many (N-N) 關係時有困難。 `DistMult`模型在處理對稱關係時遇到困難。 `RotatE`擅長建模各種關係，但比培訓`DistMult`期`TrainsE`間更昂貴。

- 在某些情況下，當節點識別和節點功能資訊都很重要時，您應該使用`concat-node-embed`將節點的功能與初始嵌入相連，告知 Neptune ML 模型取得節點的初始表示法。
- 當您在某些超參數上取得合理的良好效能時，可以根據這些結果調整超參數搜尋空間。

早期停止 Neptune ML 中的模型訓練程序

提前停止可大幅縮短模型訓練執行時間和相關成本，而不會降低模型效能。它還可以防止模型在訓練數據上過度擬合。

提前停止取決於對驗證集效能的常規測量值。最初，效能隨著訓練的進行而改善，但是當模型開始過度擬合時，它會再次開始下降。提前停止特徵可識別模型開始過度擬合的點，並在此時停止模型訓練。

Neptune ML 監視驗證指標呼叫，並將最新的驗證指標與上次**n**評估中的驗證度量平均值進行比較，其中**n**是使用`window-for-early-stop`參數集的數字。一旦驗證指標比平均值差，Neptune ML 就會停止模型訓練並儲存到目前為止的最佳模型。

您可以使用下列參數來控制提前停止：

- **`window-for-early-stop`**— 此參數的值為整數，可指定決定提前停止時要平均的最近驗證分數。預設值為 3。
- **`enable-early-stop`**— 使用此布林參數可關閉提前停止功能。依預設，其值為 `true`。

在 Neptune ML 中提早停止 HPO 程序

Neptune ML 中的提前停止功能也會停止使用 SageMaker HPO 預熱啟動功能，與其他訓練工作相比效能不佳的訓練工作。這也可以降低成本並提高 HPO 的質量。

如需其運作方式的說明，請參閱執行暖啟動超參數調整工作。

Warm start 提供了將從先前培訓工作中學到的信息傳遞給後續培訓工作的能力，並提供了兩個明顯的好處：

- 首先，先前訓練工作的結果會用來選取超參數的良好組合，以便在新的調整工作中進行搜尋。
- 其次，它允許提前停止以訪問更多模型運行，從而減少調整時間。

此功能會在 Neptune ML 中自動啟用，可讓您在模型訓練時間和效能之間取得平衡。如果您對目前模型的效能感到滿意，則可以使用該模型。否則，您會執行更多的 HPO，並以先前執行的結果為預熱啟動，以便發現更好的模型。

取得專業支援服務

AWS提供專業的支援服務，協助您解決 Neptune 專案中機器學習的問題。如果您遇到困難，請伸出AWS援手。

使用訓練過的模型來產生新的模型加工品

使用 Neptune ML 模型轉換命令，您可以使用預先訓練的模型參數計算模型成品，例如節點嵌入到處理過的圖形資料上。

增量推論的模型轉換

在增量模型推論工作流程 (p. 668)中，處理從 Neptune 匯出的更新圖形資料之後，您可以使用 curl (或 awscurl) 命令啟動模型轉換工作，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "mlModelTrainingJobId": "(the ML model training job-id)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform/"
      }'
```

然後，您可以將此工作的 ID 傳遞給建立端點 API 呼叫，以建立新端點或使用此工作產生的新模型成品更新現有端點。這可讓新的或更新的端點為更新的圖形資料提供模型預測。

任何訓練工作的模型轉換

您也可以提供`trainingJobName`參數，為 Neptune ML 模型訓練期間啟動的任何 SageMaker 訓練工作產生模型加工品。由於 Neptune ML 模型訓練工作可能會啟動許多 SageMaker 訓練工作，因此您可以彈性地根據任何 SageMaker 訓練工作來建立推論端點。

例如：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "trainingJobName" : "(name a completed SageMaker training job)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform/"
      }'
```

如果原始訓練工作是針對使用者提供的自訂模型，您必須在呼叫模型轉換時包含`customModelTransformParameters`物件。如需有Neptune ML 中的自定義模型 (p. 724)關如何實作和使用自訂模型的資訊，請參閱。

> **Note**
>
> 該`modeltransform`命令始終在該訓練的最佳 SageMaker 訓練工作上執行模型轉換。

模型轉換命令 (p. 761)如需模型轉換工作的詳細資訊，請參閱。

Neptune ML 模型培訓生產的工件

在模型訓練之後，Neptune ML 會使用訓練最佳的模型參數來產生啟動推論端點和提供模型預測所需的模型成品。這些成品由訓練任務封裝，並存放在最佳 SageMaker 訓練任務的 Amazon S3 輸出位置。

以下各節描述了各種任務的模型加工品中包含的內容，以及模型轉換命令如何使用預先存在的訓練模型，即使在新的圖形資料上也能產生人工因素。

針對不同工作產生的成品

訓練流程所產生的模型人工因素內容取決於目標機器學習任務：

- 節點分類與迴歸 — 對於節點屬性預測，成品包括模型參數、GNN 編碼器 (p. 709)中的節點嵌入、訓練圖表中節點的模型預測，以及推論端點的一些組態檔案。在節點分類和節點回歸工作中，會針對訓練期間存在的節點預先計算模型預測，以減少查詢延遲。
- 邊緣分類和回歸 — 對於邊緣屬性預測，加工品還包括模型參數和節點嵌入。模型解碼器的參數對於推論尤為重要，因為我們透過將模型解碼器套用至邊緣來源和目標頂點的嵌入，來計算邊緣分類或邊緣回歸預測。
- 連結預測 — 對於連結預測，除了針對邊緣屬性預測產生的成品外，DGL 圖形也會包含為成品，因為連結預測需要訓練圖表執行預測。鏈接預測的目的是預測可能與源頂點結合以形成圖形中特定類型的邊緣的目標頂點。為此，來源頂點的內嵌節點和已學習到的邊緣類型表示會與所有可能目標頂點的節點嵌入結合，以產生每個目標頂點的邊緣可能性評分。然後對分數進行排序以排名潛在的目標頂點並返回頂級候選人。

對於每個工作類型，DGL 的「圖形神經網路」模型權重會儲存在模型人工因素中。除了使用預先計算的預測和嵌入（轉換推斷）以減少延遲之外，Neptune ML 還可以在圖形更改時計算新的模型輸出（感應推論）。

建立新模型成品

Neptune ML 中的模型訓練後產生的模型成品會直接繫結至訓練程序。這意味著預先計算的嵌入和預測僅存在於原始訓練圖中的實體。雖然 Neptune ML 端點的感應推論模式可以即時計算新實體的預測，但是您可能想要在新實體上產生批次預測，而不需要查詢端點。

為了取得已新增至圖形的新實體的批次模型預測，需要針對新的圖形資料重新計算新模型人工因素。這是使用`modeltransform`命令來完成的。如果您只想要批次預測而不設定端點，或者想要產生所有預測以便將其寫回圖表時，可以使用此`modeltransform`指令。

由於模型訓練會在訓練程序結束時隱含地執行模型轉換，因此訓練工作一律會在訓練圖形資料上重新計算模型加工品。但是，該`modeltransform`命令也可以計算未用於訓練模型的圖形資料上的模型人工因素。為此，必須使用與原始圖形資料相同的特徵編碼來處理新的圖形資料，並且必須遵守相同的圖形結構描述。

您可以先建立新的資料處理工作，該工作是在原始訓練圖形資料上執行的資料處理工作的複製，然後在新的圖形資料上執行它 (請參閱處理 Neptune ML 的更新圖形資料 (p. 694))，以達成此目的。然後，使用新的`dataProcessingJobId`和舊的呼叫`modeltransform`命令，`modelTrainingJobId`以重新計算更新的圖形資料上的模型加工品。

對於節點屬性預測，節點嵌入和預測會在新的圖形資料上重新計算，即使是原始訓練圖形中存在的節點也是如此。

對於邊緣屬性預測和鏈接預測，也會重新計算節點嵌入，並且類似地覆蓋任何現有節點嵌入。為了重新計算節點嵌入，Neptune ML 會將先前訓練過的模型中學到的 GNN 編碼器套用至具有新功能的新圖形資料的節點。

對於沒有特徵的節點，會重新使用從原始模型訓練中學到的初始表現法。對於沒有功能且不存在於原始訓練圖形中的新節點，Neptune ML 會將其表示初始化為原始訓練圖形中存在之該節點類型的已學習初始節點表

示法的平均值。如果您有許多沒有功能的新節點，這可能會導致模型預測的效能下降，因為它們都會初始化為該節點類型的平均初始嵌入。

如果您的模型在`concat-node-embed`設定為 true 的情況下進行訓練，則會透過將節點特徵與可學習的初始表示連接起來建立初始節點表示。因此，對於更新的圖形，新節點的初始節點表示也使用平均初始節點嵌入，並與新的節點功能相連。

此外，目前不支援節點刪除。如果已在更新的圖形中移除節點，則必須在更新的圖形資料上重新訓練模型。

重新計算模型加工品會在新圖形上重複使用學習到的模型參數，並且只有在新圖形與舊圖形非常相似時才能完成。如果您的新圖形不夠相似，則需要重新訓練模型，以在新圖形資料上取得類似的模型效能。構成足夠相似的內容取決於圖形資料的結構，但根據經驗法則，如果您的新資料與原始訓練圖形資料不同 10-20% 以上，您應該重新訓練模型。

對於所有節點都具有特徵的圖形，臨界值的較高端 (20% 不同) 會套用，但對於許多節點沒有特徵，而新增至圖形的新節點沒有屬性的圖形，則下端 (10% 不同) 甚至可能太高。

模型轉換命令 (p. 761)如需模型轉換工作的詳細資訊，請參閱。

Neptune ML 中的自定義模型

Neptune ML 允許您使用 Python 定義自己的自定義模型實現。您可以使用 Neptune ML 基礎結構訓練和部署自訂模型，就像對內建模型一樣，並使用它們透過圖形查詢取得預測。

> **Note**
>
> 自訂模型目前不支援即時感應式推論 (p. 667)。

您可以依照 Neptune ML 工具組範例，並使用 Neptune ML 工具組中提供的模型元件，開始在 Python 中實作您自己的自訂模型。下列各節將詳細說明。

內容

- Neptune ML 中的自定義模型概述 (p. 725)
  - 何時在 Neptune ML 中使用自定義模型 (p. 725)
  - 在 Neptune ML 中開發和使用自訂模型的工作流程 (p. 725)
- Neptune ML 中的自定義模型開發 (p. 727)
  - Neptune ML 中的自訂模型訓練指令碼開發 (p. 727)
  - Neptune ML 中的自定義模型轉換腳本開發 (p. 728)
  - Neptunemodel-hpo-configuration.json ML 中的自訂檔案 (p. 728)
  - 在 Neptune ML 中對自訂模型實作進行本機測試 (p. 729)

Neptune ML 中的自定義模型概述

何時在 Neptune ML 中使用自定義模型

Neptune ML 的內建模型可處理 Neptune ML 支援的所有標準工作，但是在某些情況下，您可能想要對特定工作的模型進行更精細的控制，或者需要自訂模型訓練程序。例如，自訂模型會在下列情況自訂模型：

- 非常大型文字模型的文字功能的功能編碼必須在 GPU 上執行。
- 您想要使用在深度圖庫（DGL）中開發的自定義圖形神經網絡（GNN）模型。
- 您想要使用表格模型或整體模型進行節點分類和回歸。

在 Neptune ML 中開發和使用自訂模型的工作流程

Neptune ML 中的自訂模型支援旨在與現有的 Neptune ML 工作流程無縫整合。它的運作方式是在 Neptune ML 基礎結構上的來源模組中執行自訂程式碼，以訓練模型。就像內建模式一樣，Neptune ML 會自動啟動 SageMaker HyperParameter調整工作，並根據評估指標選取最佳模型。然後，它會使用來源模組中提供的實作來產生用於部署的模型加工品。

自訂模型的資料匯出、訓練組態和資料預先處理與內建模型相同。

在數據預處理之後，您可以反復和交互地使用 Python 開發和測試您的自定義模型實現。當您的模型已準備好生產時，您可以像這樣將生成的 Python 模塊上傳到 Amazon S3：

```
aws s3 cp --recursive (source path to module) s3://(bucket name)/(destination path for your
 module)
```

然後，您可以使用一般預設 (p. 665)工作流程或增量 (p. 668)資料工作流程，將模型部署到生產環境中，但有一些差異。

對於使用自訂模型的模型訓練，您必須向 Neptune ML 模型訓練 API 提供`customModelTrainingParameters` JSON 物件，以確保使用您的自訂程式碼。`customModelTrainingParameters`物件中的欄位如下所示：

- **`sourceS3DirectoryPath`**— (必要) 到實作模型之 Python 模組所在之 Amazon S3 位置的路徑。這必須指向有效的現有 Amazon S3 位置，其中至少包含訓練指令碼、轉換指令碼和`model-hpo-configuration.json`檔案。
- **`trainingEntryPointScript`**— (選擇性) 指令碼模組中的進入點名稱，該指令碼會執行模型訓練，並將超參數作為命令列引數 (包括固定的超參數)。

  預設：`training.py`。
- **`transformEntryPointScript`**— (選擇性) 在識別超參數搜尋的最佳模型之後，應執行指令碼模組中的進入點名稱，以計算模型部署所需的模型加工品。它應該能夠在沒有命令行參數的情況下運行。

  預設：`transform.py`。

例如：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
```

```
       "modelName": "custom",
       "customModelTrainingParameters" : {
         "sourceS3DirectoryPath": "s3://(your Amazon S3 bucket)/(path to your Python
 module)",
         "trainingEntryPointScript": "(your training script entry-point name in the Python
 module)",
         "transformEntryPointScript": "(your transform script entry-point name in the
 Python module)"
       }
     }'
```

同樣地，若要啟用自訂模型轉換，您必須將customModelTransformParameters JSON 物件提供給 Neptune ML 模型轉換 API，其欄位值與訓練工作中儲存的模型參數相容。customModelTransformParameters物件包含下列欄位：

- **sourceS3DirectoryPath**— (必要) 到實作模型之 Python 模組所在之 Amazon S3 位置的路徑。這必須指向有效的現有 Amazon S3 位置，其中至少包含訓練指令碼、轉換指令碼和model-hpo-configuration.json檔案。
- **transformEntryPointScript**— (選擇性) 在識別超參數搜尋的最佳模型之後，應執行指令碼模組中的進入點名稱，以計算模型部署所需的模型加工品。它應該能夠在沒有命令行參數的情況下運行。

  預設：transform.py。

例如：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "trainingJobName" : "(name of a completed SageMaker training job)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform/"
        "customModelTransformParameters" : {
          "sourceS3DirectoryPath": "s3://(your Amazon S3 bucket)/(path to your Python
 module)",
          "transformEntryPointScript": "(your transform script entry-point name in the
 Python module)"
        }
     }'
```

Neptune ML 中的自定義模型開發

開始自訂模型開發的好方法是遵循 Neptune ML 工具組範例來建構和撰寫訓練模組。Neptune ML 工具組也會在 modelzoo 中實作模組化圖形 ML 模型元件，您可以堆疊並使用這些元件來建立自訂模型。

此外，此工具組還提供公用程式功能，可協助您在模型訓練和模型轉換期間產生必要的加工品。您可以在自定義實現中導入此 Python 包。此工具組中提供的任何功能或模組也可在 Neptune ML 訓練環境中使用。

如果你的 Python 模塊有額外的外部依賴關係，你可以通過在模塊的目錄中創建一個`requirements.txt`文件來包含這些額外的依賴關係。然後會在執行訓練指令碼之前安裝`requirements.txt`檔案中列出的套件。

實現自定義模型的 Python 模塊至少需要包含以下內容：

- 訓練指令碼入口點
- 轉換腳本進入點
- `model-hpo-configuration.json` 檔案

Neptune ML 中的自訂模型訓練指令碼開發

您的自訂模型訓練指令碼應該是可執行的 Python 指令碼，例如 Neptune ML 工具組的`train.py`範例。它必須接受超參數名稱和值作為命令行參數。在模型訓練期間，會從`model-hpo-configuration.json`檔案取得超參數名稱。如果超參數是可調的，則超參數值會落在有效的超參數範圍內；如果無法調整，則採用預設的超參數值。

您的訓練指令碼會使用如下語法在 SageMaker 訓練執行個體上執行：

```
python3 (script entry point) --(1st parameter) (1st value) --(2nd parameter) (2nd
 value) (...)
```

對於所有工作，Neptune ML 除了指定的超參數之外，還 AutoTrainer 會傳送數個必要的參數至訓練指令碼，而且您的指令碼必須能夠處理這些額外的參數，才能正常運作。

這些額外的必要參數會因工作而有所不同：

用於節點分類或節點迴歸

- **`task`**— Neptune ML 內部使用的任務類型。對於節點分類，這是`node_class`，對於節點回歸，它是`node_regression`。
- **`model`**— Neptune ML 內部使用的型號名稱，`custom`在這種情況下。
- **`name`**— Neptune ML 內部使用的任務名稱，在此情況下用`node_class-custom`於節點分類，以及用`node_regression-custom`於節點迴歸。
- **`target_ntype`**— 用於分類或回歸的節點類型名稱。
- **`property`**— 用於分類或回歸的節點屬性名稱。

用於鏈接預測

- **`task`**— Neptune ML 內部使用的任務類型。對於鏈接預測，這是`link_predict`。
- **`model`**— Neptune ML 內部使用的型號名稱，`custom`在這種情況下。
- **`name`**— Neptune ML 內部使用的任務名稱，`link_predict-custom`在這種情況下。

適用於邊分類或邊迴歸

- **`task`**— Neptune ML 內部使用的任務類型。對於邊分類，這是`edge_class`，對於邊迴歸，它是`edge_regression`。

- **model**— Neptune ML 內部使用的型號名稱，custom在這種情況下。
- **name**— Neptune ML 內部使用的任務名稱，在此情況下用edge_class-custom於邊緣分類，以及用edge_regression-custom於邊緣迴歸。
- **target_etype**— 用於分類或回歸的邊類型名稱。
- **property**— 用於分類或回歸的邊性質名稱。

您的指令碼應該儲存模型參數，以及訓練結束時所需的任何其他成品。

您可以使用 Neptune ML 工具組公用程式函數來判斷已處理的圖形資料的位置、應儲存模型參數的位置，以及訓練執行個體上可用的 GPU 裝置。如需如何使用這些公用程式函數的範例，請參閱 train.py 訓練指令碼範例。

Neptune ML 中的自定義模型轉換腳本開發

需要轉換指令碼，才能利用 Neptune ML 增量工作流程 (p. 668)，在不重新訓練模型的情況下，在不進行模型的情況下進行模型推論。即使模型部署所需的所有成品都是由訓練指令碼所產生，如果您想要在不重新訓練模型的情況下產生更新的模型，您仍然需要提供轉換指令碼。

> Note
>
> 自訂模型目前不支援即時感應式推論 (p. 667)。

您的自訂模型轉換指令碼應該是可執行的 Python 指令碼，例如 Neptune ML 工具組的 transform.py 範例指令碼。因為這個指令碼是在沒有命令列引數的模型訓練期間叫用，所以指令碼接受的任何命令列引數都必須有預設值。

指令碼在 SageMaker 訓練執行個體上執行，語法如下：

```
python3 (your transform script entry point)
```

您的轉換腳本將需要各種信息，例如：

- 處理過的圖形資料的位置。
- 儲存模型參數的位置，以及應儲存新模型人工因素的位置。
- 執行個體上可用的裝置。
- 產生最佳模型的超參數。

這些輸入是使用指令碼可呼叫的 Neptune ML 公用程式函數來取得。有關如何執行此操作的示例，請參閱工具包的示例 transform.py 腳本。

指令碼應儲存節點內嵌、節點 ID 對應，以及每項工作模型部署所需的任何其他人工因素。如需不同 Neptune ML 工作所需之模型成品的詳細資訊，請參閱模型人工因素文 (p. 722)件。

Neptunemodel-hpo-configuration.json ML 中的自訂檔案

model-hpo-configuration.json檔案會定義自訂模型的超參數。它的格式與 Neptune ML 內建模型所使用的model-hpo-configuration.json檔案格式 (p. 711)相同，而且優先於 Neptune ML 自動產生並上傳到處理資料位置的版本。

當您在模型中新增超參數時，也必須在此檔案中新增超參數項目，以便將超參數傳遞至訓練指令碼。

如果您希望超參數可調整，則必須為超參數提供範圍，並將其設定為tier-1tier-2、或tier-3參數。如果設定的訓練工作總數允許調整其階層中的超參數，則會調整超參數。對於不可調整的參數，您必須提供預設值，並將超參數新增至檔案的fixed-param區段。有關如何執行此操作的示例model-hpo-configuration.json文件，請參閱工具包的示例文件。

您也必須提供「最 SageMaker HyperParameter 佳化」工作將用來評估受訓的候選模型的量度定義。為此，您可以像這樣將`eval_metric` JSON 對象添加到`model-hpo-configuration.json`文件中：

```
"eval_metric": {
  "tuning_objective": {
      "MetricName": "(metric_name)",
      "Type": "Maximize"
  },
  "metric_definitions": [
    {
      "Name": "(metric_name)",
      "Regex": "(metric regular expression)"
    }
  ]
},
```

物件中的`metric_definitions`陣列`eval_metric`會列出您要從訓練執行個體擷取之每個量度 SageMaker 的測量結果定義物件。每個測量結果定義物件都有一個`Name`索引鍵，可讓您提供量度名稱 (例如「準確性」、「F1」等)。`Regex`索引鍵可讓您提供一個規則運算式字串，以符合特定量度在訓練記錄中列印的方式。如需如何定義測量結果的詳細資訊，請參閱「SageMaker HyperParameter 調整」頁面。

`eval_metric`然後，中的`tuning_objective`物件可讓您指定`metric_definitions`應使用中的哪些測量結果作為評估測量結果，作為超參數最佳化的目標測量結果。的值`MetricName`必須與`Name`中其中一個定義中的 a 的值相符`metric_definitions`。的值`Type`應該是「最大化」或「最小化」，具體取決於度量應該被解譯為 greater-is-better (如「準確性」) 還是 less-is-better (如「」mean-squared-error」)。

此`model-hpo-configuration.json`檔案區段中的錯誤可能會導致 Neptune ML 模型訓練 API 工作失敗，因為 SageMaker HyperParameter 調整工作將無法選取最佳模型。

在 Neptune ML 中對自訂模型實作進行本機測試

您可以使用 Neptune ML 工具組 Conda 環境在本機執行程式碼，以便測試和驗證您的模型。如果您正在開發 Neptune 筆記本實例，則此 Conda 環境將預先安裝在 Neptune 筆記本實例上。如果您要在不同的執行個體上開發，則需要遵循 Neptune ML 工具組中的本機設定指示。

當您呼叫模型訓練 API 時，Conda 環境會準確地重現模型 (p. 757)將執行的環境。所有範例訓練指令碼和轉換指令碼都可讓您傳遞命令列`--local`旗標，在本機環境中執行指令碼，以便輕鬆偵錯。在開發自己的模型時，這是一個很好的做法，因為它可以讓您以互動方式和反覆測試模型實作。在 Neptune ML 生產訓練環境中進行模型訓練期間，會省略此參數。

建立要查詢的推論端點

推論端點可讓您查詢模型訓練程序所建構的特定模型。端點會附加至訓練程序能夠產生的指定類型的最佳效能模型。然後，端點能夠接受來自 Neptune 的 Gemlin 查詢，並返回該模型對查詢中輸入的預測。建立推論端點之後，它會保持作用中狀態，直到您將其刪除為止。

管理 Neptune ML 的推論端點

完成從 Neptune 匯出之資料的模型訓練之後，您可以使用`curl` (或`awscurl`) 命令建立推論端點，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "mlModelTrainingJobId": "(the model-training job-id of a completed job)"
      }'
```

您也可以使用完成的模型轉換工作所建立的模型建立推論端點，方法大致相同：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "mlModelTransformJobId": "(the model-transform job-id of a completed job)"
      }'
```

有關如何使用這些命令的詳細資訊端點指令 (p. 765)，以及有關如何取得端點狀態、如何刪除端點以及如何列出所有推論端點的資訊。

Neptune ML 中的推論查詢

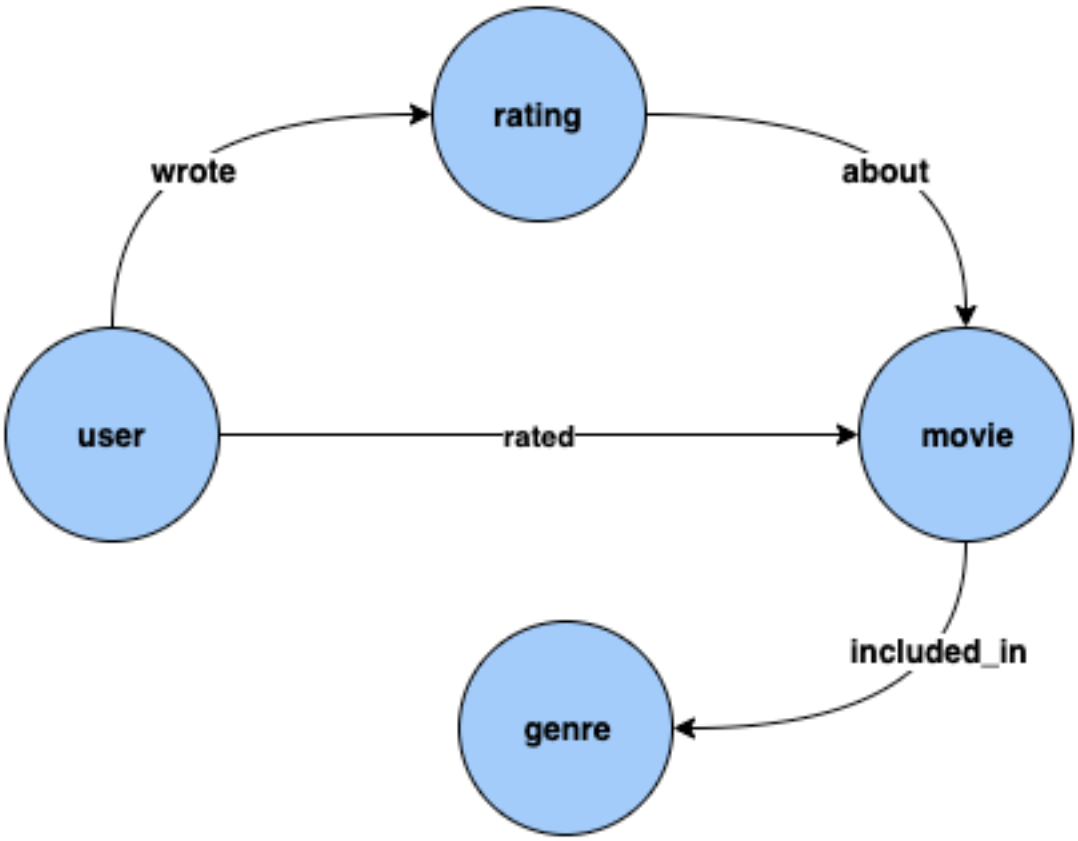
您可以使用小鬼或 SPARQL 來查詢 Neptune ML 推論端點。然而，即時感應式推論 (p. 667)目前僅支援 Gemlin 查詢。

Neptune ML 中的小鬼推論查詢

如中所述Neptune ML DEDEDE (p. 652)，Neptune ML 支援可執行下列類型推論工作的訓練模型：

- 節點分類 — 預測頂點屬性的分類特徵。
- 節點迴歸 — 預測頂點的數值性質。
- 邊分類 — 預測邊屬性的分類特徵。
- 邊迴歸 — 預測邊的數值性質。
- 連結預測 — 預測給定來源節點和傳出邊緣的目標節點，或指定目標節點和傳入邊緣的來源節點。

我們可以通過使用 GroupLens Research 提供的 MovieLens 100k 數據集的示例來說明這些不同的任務。該數據集由電影，用戶和用戶對電影的評級組成，我們從中創建了一個屬性圖，如下所示：



節點分類：在上面的數據集，Genre是Movie由邊緣連接到頂點類型的頂點類型included_in。但是，如果我們調整數據集以製Genre作頂點類型的分類特徵Movie，那麼推斷Genre添加到我們知識圖中的新電影的問題可以使用節點分類模型來解決。

節點回歸：如果我們考慮頂點類型Rating，它具有類似的屬性timestamp和score，那麼推斷一個數值Score的問題Rating可以使用節點回歸模型來解決。

邊緣分類：同樣地，對於Rated邊緣，如果我們有一Scale個屬性可以具有其中一個值,Love,,Like,Dislike,Neutral,,Hate，，，，，則推斷Scale新電影/評分的Rated邊緣的問題可以使用邊緣分類模型來解決。

邊迴歸：同樣，對於相同的Rated邊緣，如果我們有一Score個包含額定值的數值的屬性，那麼這可以從邊緣回歸模型推斷。

鏈接預測：例如，找到最有可能對給定電影評分的前十名用戶，或者找到給定用戶最有可能評分的十大電影之類的問題，屬於鏈接預測。

> Note
>
> 對於 Neptune ML 使用案例，我們有一套非常豐富的筆記本，旨在讓您實際了解每個使用案例。當您使用 Neptune MLAWS CloudFormation 範本建立 Neptune ML 叢集時，您可以與 Neptune (p. 654) 叢集一起建立這些筆記本。這些筆記本也可以在 github 上找到。

主題

- 在小靈推論查詢中使用的 Neptune ML 謂詞 (p. 733)
- Neptune ML 中的小鬼節點分類查詢 (p. 735)
- Neptune ML 中的小靈節點回歸查詢 (p. 739)
- Neptune ML 中的小精靈邊緣分類查詢 (p. 740)
- Neptune ML 中的小精靈邊緣回歸查詢 (p. 742)
- 在 Neptune ML 中使用鏈接預測模型的 Gremlin 鏈接預測查詢 (p. 744)
- Neptune ML 小鬼推論查詢的例外狀況清單 (p. 746)

在小靈推論查詢中使用的 Neptune ML 謂詞

Neptune#ml.deterministic

這個述詞是感應式推論查詢的一個選項，也就是說，包含Neptune#ml.inductiveInference (p. 734)述詞的查詢。

使用感應推斷時，Neptune 引擎會建立適當的子圖來評估訓練過的 GNN 模型，而此子圖的要求取決於最終模型的參數。具體而言，num-layer參數會決定目標節點或邊緣的遍歷躍點數目，而fanouts參數則指定要在每個躍點上取樣的鄰近項目數 (請參閱 HPO 參數 (p. 711))。

依預設，感應式推論查詢會在非決定性模式下執行，Neptune 會隨機建置鄰域。在進行預測時，這種正常的隨機鄰近抽樣有時會導致不同的預測。

當您包含Neptune#ml.deterministic在感應式推論查詢中時，Neptune 引擎會嘗試以確定性的方式對鄰近項目進行取樣，以便相同查詢的多次呼叫每次都會傳回相同的結果。但是，不能保證結果完全具決定性，因為對分佈式系統的基礎圖形和成品的更改仍然可能會引起波動。

您可以像這樣在查詢中包含Neptune#ml.deterministic述詞：

```
.with("Neptune#ml.deterministic")
```

如果Neptune#ml.deterministic謂詞包含在不同時包含的查詢中Neptune#ml.inductiveInference，則只會忽略述詞。

Neptune#ml.disableInductiveInferenceMetadataCache

這個述詞是感應式推論查詢的一個選項，也就是說，包含Neptune#ml.inductiveInference (p. 734)述詞的查詢。

對於感應式推論查詢，Neptune 會使用存放在 Amazon S3 中的中繼資料檔案來決定建立鄰里時的躍點數和扇出數。Neptune 通常會快取此模型中繼資料，以避免重複從 Amazon S3 擷取檔案。緩存可以通過在查詢中包括Neptune#ml.disableInductiveInferenceMetadataCache謂詞被禁用。雖然 Neptune 直接

從 Amazon S3 擷取中繼資料可能會比較慢，但是當 SageMaker 端點在重新訓練或轉換後更新並且快取過時時，這會很有用。

您可以像這樣在查詢中包含Neptune#ml.disableInductiveInferenceMetadataCache述詞：

```
.with("Neptune#ml.disableInductiveInferenceMetadataCache")
```

以下是 Jupyter 筆記本中查詢範例的外觀：

```
%%gremlin
g.with("Neptune#ml.endoint", "ep1")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .with("Neptune#ml.disableInductiveInferenceMetadataCache")
 .V('101').properties("rating")
 .with("Neptune#ml.regression")
 .with("Neptune#ml.inductiveInference")
```

Neptune#ml.endpoint

Neptune#ml.endpoint述詞會在with()步驟中使用來指定推論端點 (如有必要)：

```
 .with("Neptune#ml.endpoint", "the model's SageMaker inference endpoint")
```

您可以透過端點id或其 URL 來識別端點。例如：

```
 .with( "Neptune#ml.endpoint", "node-classification-movie-lens-endpoint" )
```

或者：

```
 .with( "Neptune#ml.endpoint", "https://runtime.sagemaker.us-east-1.amazonaws.com/
endpoints/node-classification-movie-lens-endpoint/invocations" )
```

> **Note**
>
> 如果您將 Neptune DB 叢集參數群組中的參數設 (p. 664)定為端點id或 URL，則不需要在每個查詢中包含Neptune#ml.endpoint述詞。neptune_ml_endpoint

Neptune#ml.iamRoleArn

Neptune#ml.iamRoleArn在with()步驟中使用以指定 SageMaker 執行 IAM 角色的 ARN，如有必要：

```
 .with("Neptune#ml.iamRoleArn", "the ARN for the SageMaker execution IAM role")
```

如需如何建立 SageMaker 執行 IAM 角色的詳細資訊，請參閱建立自訂NeptuneSageMakerIAMRole角色 (p. 657)。

> **Note**
>
> 如果您將 Neptune DB 叢集參數群組中的參數設 (p. 662)定為 SageMaker 執行 IAM 角色的 ARN，則不需要在每個查詢中包含Neptune#ml.iamRoleArn述詞。neptune_ml_iam_role

Neptune #ml. 誘導推論

轉論推論預設為啟用 Gemlin。若要進行即時的感應式推論 (p. 667)查詢，請包含如下Neptune#ml.inductiveInference述述詞：

```
.with("Neptune#ml.inductiveInference")
```

如果您的圖形是動態的，則感應推斷通常是最佳選擇，但是如果您的圖形是靜態的，則傳感推斷會更快，更有效率。

Neptune#ml.limit

述Neptune#ml.limit詞可以選擇限制每個實體返回的結果數量：

```
.with( "Neptune#ml.limit", 2 )
```

默認情況下，限制為 1，並且可以設置的最大數量為 100。

Neptune#ml.threshold

述Neptune#ml.threshold詞可以選擇性地建立結果評分的截止臨界值：

```
.with( "Neptune#ml.threshold", 0.5D )
```

這可讓您捨棄分數低於指定臨界值的所有結果。

Neptune#ml.classification

Neptune#ml.classification述詞會附加至properties()步驟，以確定需要從節點分類模型的 SageMaker 端點擷取屬性：

```
.properties( "property key of the node classification
model" ).with( "Neptune#ml.classification" )
```

Neptune#ml.regression

Neptune#ml.regression謂詞附加到properties()步驟，以確定需要從節點回歸模型的 SageMaker 端點獲取屬性：

```
.properties( "property key of the node regression model" ).with( "Neptune#ml.regression" )
```

Neptune#ml.prediction

Neptune#ml.prediction謂詞附加到in()並out()步驟確定這是一個鏈接預測查詢：

```
.in("edge label of the link prediction
model").with("Neptune#ml.prediction").hasLabel("target node label")
```

Neptune#ml.score

Neptune#ml.score述詞用於 Gemlin 節點或邊緣分類查詢中，以擷取機器學習信賴度分數。Neptune#ml.score述詞應該與properties()步驟中的查詢述詞一起傳遞，以取得節點或邊緣分類查詢的 ML 信賴度分數。

您可以在邊分類部分找到包含其他節點分類範例 (p. 737)的節點分類範例，以及邊分類 (p. 740)範例。

Neptune ML 中的小鬼節點分類查詢

對於 Neptune ML 中的小鬼節點分類：

- 該模型是在頂點的一個屬性上進行訓練。這個屬性的一組唯一值被稱為一組節點類，或者簡單地說，類。
- 一個頂點的屬性的節點類或分類屬性值可以從節點分類模型推斷。此屬性尚未貼附至頂點時非常有用。
- 為了從節點分類模型中獲取一個或多個類，您需要使用與謂詞的`with()`步驟`Neptune#ml.classification`來配置`properties()`步驟。如果輸出格式是頂點屬性，則輸出格式與您所期望的格式相似。

  Note

  節點分類僅適用於字串屬性值。這意味著數字屬性值（例如`0`或）不`1`受支持，儘管字符串等效`"0"`和`"1"`是。同樣，布爾屬性值`true`和`false`不起作用，但`"true"`和`"false"`做。

以下是範例節點分類查詢：

```
g.with( "Neptune#ml.endpoint","node-classification-movie-lens-endpoint" )
 .with( "Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role" )
 .with( "Neptune#ml.limit", 2 )
 .with( "Neptune#ml.threshold", 0.5D )
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre").with("Neptune#ml.classification")
```

此查詢的輸出應類似以下所示：

```
==>vp[genre->Action]
==>vp[genre->Crime]
==>vp[genre->Comedy]
```

在上面的查詢中，`V()`和`properties()`步驟使用如下：

該`V()`步驟包含要從節點分類模型中獲取類的頂點集：

```
 .V( "movie_1", "movie_2", "movie_3" )
```

`properties()`步驟包含模型訓練的索引鍵，而且必`.with("Neptune#ml.classification")`須指出這是節點分類 ML 推論查詢。

一個`properties().with("Neptune#ml.classification")`步驟目前不支援多個內容索引鍵。例如，下列查詢會導致例外狀況：

```
g.with("Neptune#ml.endpoint", "node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre", "other_label").with("Neptune#ml.classification")
```

如需特定錯誤訊息，請參閱 Neptune ML 例外狀況清單 (p. 746)。

`properties().with("Neptune#ml.classification")`步驟可與下列任一步驟搭配使用：

- `value()`
- `value().is()`
- `hasValue()`
- `has(value,"")`
- `key()`
- `key().is()`
- `hasKey()`

- has(key,"")
- path()

其他節點分類查詢

如果推論端點和對應的 IAM 角色都儲存在資料庫叢集參數群組中，節點分類查詢可以如下所示：

```
g.V("movie_1", "movie_2", "movie_3").properties("genre").with("Neptune#ml.classification")
```

您可以使用以下union()步驟混合查詢中的頂點屬性和類：

```
g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .union(
   properties("genre").with("Neptune#ml.classification"),
   properties("genre")
 )
```

您也可以創建一個無界的查詢，如下所示：

```
g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V()
 .properties("genre").with("Neptune#ml.classification")
```

您可以使用步驟與select()步驟一起檢索節點類與頂as()點：

```
g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" ).as("vertex")
 .properties("genre").with("Neptune#ml.classification").as("properties")
 .select("vertex","properties")
```

您也可以篩選節點類別，如下列範例所示：

```
g.with("Neptune#ml.endpoint", "node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre").with("Neptune#ml.classification")
 .has(value, "Horror")

g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre").with("Neptune#ml.classification")
 .has(value, P.eq("Action"))

g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre").with("Neptune#ml.classification")
 .has(value, P.within("Action", "Horror"))
```

您可以使用Neptune#ml.score述詞取得節點分類可信度分數：

```
 g.with("Neptune#ml.endpoint","node-classification-movie-lens-endpoint")
```

```
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V( "movie_1", "movie_2", "movie_3" )
 .properties("genre", "Neptune#ml.score").with("Neptune#ml.classification")
```

回應看起來像這樣：

```
==>vp[genre->Action]
==>vp[Neptune#ml.score->0.01234567]
==>vp[genre->Crime]
==>vp[Neptune#ml.score->0.543210]
==>vp[genre->Comedy]
==>vp[Neptune#ml.score->0.10101]
```

在節點分類查詢中使用感應推論

假設您要在 Jupyter 筆記本中將新節點添加到現有圖形中，如下所示：

```
%%gremlin
g.addV('label1').property(id,'101').as('newV')
 .V('1').as('oldV1')
 .V('2').as('oldV2')
 .addE('eLabel1').from('newV').to('oldV1')
 .addE('eLabel2').from('oldV2').to('newV')
```

然後，您可以使用感應推論查詢來獲得反映新節點的類型和可信度分數：

```
%%gremlin
g.with("Neptune#ml.endoint", "nc-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('101').properties("genre", "Neptune#ml.score")
 .with("Neptune#ml.classification")
 .with("Neptune#ml.inductiveInference")
```

但是，如果您多次運行查詢，則可能會得到一些不同的結果：

```
# First time
==>vp[genre->Action]
==>vp[Neptune#ml.score->0.12345678]

# Second time
==>vp[genre->Action]
==>vp[Neptune#ml.score->0.21365921]
```

你可以使相同的查詢確定性：

```
%%gremlin
g.with("Neptune#ml.endoint", "nc-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('101').properties("genre", "Neptune#ml.score")
 .with("Neptune#ml.classification")
 .with("Neptune#ml.inductiveInference")
 .with("Neptune#ml.deterministic")
```

在這種情況下，每次結果大致相同：

```
# First time
==>vp[genre->Action]
```

```
==>vp[Neptune#ml.score->0.12345678]
# Second time
==>vp[genre->Action]
==>vp[Neptune#ml.score->0.12345678]
```

Neptune ML 中的小靈節點回歸查詢

節點迴歸類似於節點分類，不同之處在於從每個節點的回歸模型推斷出來的值是數值。您可以使用與節點分類相同的 Gemlin 查詢進行節點回歸，但下列差異除外：

- 再次，在 Neptune ML，節點指的是頂點。
- 該`properties()`步驟採用形式，`properties().with("Neptune#ml.regression")`而不是`properties().with("Neptune#ml.classification")`。
- `"Neptune#ml.limit`「和`"Neptune#ml.threshold"`述詞不適用。
- 當您篩選值時，您必須指定一個數值。

以下是範例頂點分類查詢：

```
g.with("Neptune#ml.endpoint","node-regression-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::0123456789:role/sagemaker-role")
 .V("movie_1","movie_2","movie_3")
 .properties("revenue").with("Neptune#ml.regression")
```

您可以篩選使用迴歸模型推論的值，如下列範例所示：

```
g.with("Neptune#ml.endpoint","node-regression-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("movie_1","movie_2","movie_3")
 .properties("revenue").with("Neptune#ml.regression")
 .value().is(P.gte(1600000))

g.with("Neptune#ml.endpoint","node-regression-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("movie_1","movie_2","movie_3")
 .properties("revenue").with("Neptune#ml.regression")
 .hasValue(P.lte(1600000D))
```

在節點回歸查詢中使用感應推論

假設您要在 Jupyter 筆記本中將新節點添加到現有圖形中，如下所示：

```
%%gremlin
g.addV('label1').property(id,'101').as('newV')
 .V('1').as('oldV1')
 .V('2').as('oldV2')
 .addE('eLabel1').from('newV').to('oldV1')
 .addE('eLabel2').from('oldV2').to('newV')
```

然後，您可以使用感應推論查詢來獲得考慮到新節點的評分：

```
%%gremlin
g.with("Neptune#ml.endoint", "nr-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('101').properties("rating")
 .with("Neptune#ml.regression")
 .with("Neptune#ml.inductiveInference")
```

由於查詢不是確定性的，因此如果根據鄰近區域執行多次，查詢可能會傳回稍有不同的結果：

```
# First time
==>vp[rating->9.1]

# Second time
==>vp[rating->8.9]
```

如果您需要更一致的結果，則可以使查詢具有確定性：

```
%%gremlin
g.with("Neptune#ml.endoint", "nc-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('101').properties("rating")
 .with("Neptune#ml.regression")
 .with("Neptune#ml.inductiveInference")
 .with("Neptune#ml.deterministic")
```

現在每次結果都會大致相同：

```
# First time
==>vp[rating->9.1]

# Second time
==>vp[rating->9.1]
```

Neptune ML 中的小精靈邊緣分類查詢

對於 Neptune ML 中的小精靈邊緣分類：

- 模型會根據邊的一個屬性進行訓練。此屬性的唯一值集合稱為一組類別。
- 邊緣的類別或分類屬性值可以從邊分類模型推斷出來，如果此性質尚未貼附至邊緣，此功能非常有用。
- 為了從邊緣分類模型中獲取一個或多個類，您需要使用該`with()`步驟與述詞，`"Neptune#ml.classification"`以配置`properties()`步驟。輸出格式類似於邊緣屬性時所期望的格式。

> **Note**
>
> 邊分類僅適用於字串性質值。這意味著數字屬性值（例如`0`或）不`1`受支持，儘管字符串等效`"0"`和`"1"`是。同樣，布爾屬性值`true`和`false`不起作用，但`"true"`和`"false"`做。

以下是邊緣分類查詢的範例，該查詢使用`Neptune#ml.score`述詞要求置信度分數：

```
g.with("Neptune#ml.endpoint","edge-classification-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .E("relationship_1","relationship_2","relationship_3")
 .properties("knows_by", "Neptune#ml.score").with("Neptune#ml.classification")
```

回應看起來像這樣：

```
==>p[knows_by->"Family"]
==>p[Neptune#ml.score->0.01234567]
==>p[knows_by->"Friends"]
==>p[Neptune#ml.score->0.543210]
==>p[knows_by->"Colleagues"]
==>p[Neptune#ml.score->0.10101]
```

一個小鬼邊緣分類查詢的語法

對於一個簡單的圖形，其中User是頭部和尾部節點，並且Relationship是連接它們的邊緣，邊緣分類查詢的示例如下：

```
g.with("Neptune#ml.endpoint","edge-classification-social-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .E("relationship_1","relationship_2","relationship_3")
 .properties("knows_by").with("Neptune#ml.classification")
```

此查詢的輸出應類似以下所示：

```
==>p[knows_by->"Family"]
==>p[knows_by->"Friends"]
==>p[knows_by->"Colleagues"]
```

在上面的查詢中，E()和properties()步驟使用如下：

- 此E()步驟包含要從邊緣分類模型擷取類別的一組邊緣：

  ```
  .E("relationship_1","relationship_2","relationship_3")
  ```

- properties()步驟包含模型訓練的索引鍵，而且必.with("Neptune#ml.classification")須指出這是邊緣分類 ML 推論查詢。

一個properties().with("Neptune#ml.classification")步驟目前不支援多個內容索引鍵。例如，下列查詢會導致擲出例外狀況：

```
g.with("Neptune#ml.endpoint","edge-classification-social-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .E("relationship_1","relationship_2","relationship_3")
 .properties("knows_by", "other_label").with("Neptune#ml.classification")
```

如需特定錯誤訊息，請參閱Neptune ML 小鬼推論查詢的例外狀況清單 (p. 746)。

properties().with("Neptune#ml.classification")步驟可與下列任一步驟搭配使用：

- value()
- value().is()
- hasValue()
- has(value,"")
- key()
- key().is()
- hasKey()
- has(key,"")
- path()

在邊緣分類查詢中使用感應推論

假設你要在 Jupyter 筆記本中為現有的圖形添加一個新的邊緣，如下所示：

```
%%gremlin
g.V('1').as('fromV')
```

```
.V('2').as('toV')
.addE('eLabel1').from('fromV').to('toV').property(id, 'e101')
```

然後，您可以使用感應性推論查詢來獲得考慮到新邊緣的比例：

```
%%gremlin
g.with("Neptune#ml.endoint", "ec-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .E('e101').properties("scale", "Neptune#ml.score")
 .with("Neptune#ml.classification")
 .with("Neptune#ml.inductiveInference")
```

由於查詢不是確定性的，因此如果您多次執行查詢，結果會根據隨機鄰域而有所不同：

```
# First time
==>vp[scale->Like]
==>vp[Neptune#ml.score->0.12345678]

# Second time
==>vp[scale->Like]
==>vp[Neptune#ml.score->0.21365921]
```

如果您需要更一致的結果，則可以使查詢具有確定性：

```
%%gremlin
g.with("Neptune#ml.endoint", "ec-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .E('e101').properties("scale", "Neptune#ml.score")
 .with("Neptune#ml.classification")
 .with("Neptune#ml.inductiveInference")
 .with("Neptune#ml.deterministic")
```

現在，每次運行查詢時，結果或多或少都會相同：

```
# First time
==>vp[scale->Like]
==>vp[Neptune#ml.score->0.12345678]

# Second time
==>vp[scale->Like]
==>vp[Neptune#ml.score->0.12345678]
```

Neptune ML 中的小精靈邊緣回歸查詢

邊迴歸類似於邊緣分類，不同之處在於從 ML 模型推斷的值是數值。對於邊迴歸，Neptune ML 支援與分類相同的查詢。

需要注意的要點是：

- 您必須使用 ML 述詞`"Neptune#ml.regression"`來設定此使用案例的`properties()`步驟。
- `"Neptune#ml.limit"`和述`"Neptune#ml.threshold"`詞不適用於此使用案例。
- 要對值進行過濾，您需要將該值指定為數字。

一個小鬼邊緣回歸查詢的語法

對於一個簡單的圖表，其中`User`是頭節點，`Movie`是尾部節點，並且`Rated`是連接它們的邊緣，這裡是一個示例邊緣回歸查詢，它可以找到數值的額定值，在這裡稱為 score，用於邊緣`Rated`：

```
g.with("Neptune#ml.endpoint","edge-regression-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .E("rating_1","rating_2","rating_3")
 .properties("score").with("Neptune#ml.regression")
```

您也可以篩選從 ML 迴歸模型推斷出來的值。對於由"rating_1"、"rating_2"和所識別的現有Rated邊(從User到Movie)"rating_3"，其中邊屬性Score不存在於這些額定值中，您可以使用類似以下Score的查詢來推斷其大於或等於 9 的邊：

```
g.with("Neptune#ml.endpoint","edge-regression-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .E("rating_1","rating_2","rating_3")
 .properties("score").with("Neptune#ml.regression")
 .value().is(P.gte(9))
```

在邊緣迴歸查詢中使用感應推論

假設你要在 Jupyter 筆記本中為現有的圖形添加一個新的邊緣，如下所示：

```
%%gremlin
g.V('1').as('fromV')
.V('2').as('toV')
.addE('eLabel1').from('fromV').to('toV').property(id, 'e101')
```

然後，您可以使用感應性推論查詢來獲得考慮到新邊緣的分數：

```
%%gremlin
g.with("Neptune#ml.endoint", "er-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .E('e101').properties("score")
 .with("Neptune#ml.regression")
 .with("Neptune#ml.inductiveInference")
```

由於查詢不是確定性的，因此如果您多次執行查詢，結果會根據隨機鄰域而有所不同：

```
# First time
==>ep[score->96]

# Second time
==>ep[score->91]
```

如果您需要更一致的結果，則可以使查詢具有確定性：

```
%%gremlin
g.with("Neptune#ml.endoint", "er-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .E('e101').properties("score")
 .with("Neptune#ml.regression")
 .with("Neptune#ml.inductiveInference")
 .with("Neptune#ml.deterministic")
```

現在，每次運行查詢時，結果或多或少都會相同：

```
# First time
==>ep[score->96]

# Second time
```

```
==>ep[score->96]
```

在 Neptune ML 中使用鏈接預測模型的 Gremlin 鏈接預測查詢

鏈接預測模型可以解決以下問題：

- 頭節點預測：給定頂點和邊緣類型，該頂點可能從哪個頂點鏈接？
- 尾節點預測：給定頂點和邊緣標籤，該頂點可能鏈接到哪個頂點？

> **Note**
>
> Neptune ML 尚不支援邊緣預測。

對於下面的例子，考慮與頂點`User`和`Movie`由邊緣鏈接一個簡單的圖形`Rated`。

以下是頭節點預測查詢範例，用於預測最有可能評分影片、`"movie_1"`、`"movie_2"`、和的前五名使用者`"movie_3"`：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .with("Neptune#ml.limit", 5)
 .V("movie_1", "movie_2", "movie_3")
 .in("rated").with("Neptune#ml.prediction").hasLabel("user")
```

下面是一個類似的尾部節點預測，用於預測用戶可能評分`"user_1"`的前五大電影：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("user_1")
 .out("rated").with("Neptune#ml.prediction").hasLabel("movie")
```

邊緣標籤和預測的頂點標籤都是必需的。如果省略任一，則擲出例外狀況。例如，下列查詢沒有預測頂點標籤的下列查詢會擲出例外狀況：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("user_1")
 .out("rated").with("Neptune#ml.prediction")
```

同樣地，沒有邊緣標籤的下列查詢會擲回例外狀況：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("user_1")
 .out().with("Neptune#ml.prediction").hasLabel("movie")
```

如需這些例外狀況傳回的特定錯誤訊息，請參閱 Neptune ML 例外狀況清單 (p. 746)。

其他連結預測查詢

您可以使用的`select()`步驟`as()` 步驟與輸入頂點一起輸出預測的頂點：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("movie_1").as("source")
 .in("rated").with("Neptune#ml.prediction").hasLabel("user").as("target")
 .select("source","target")
```

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("user_1").as("source")
 .out("rated").with("Neptune#ml.prediction").hasLabel("movie").as("target")
 .select("source","target")
```

您可以進行無界查詢，如下所示：

```
g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("user_1")
 .out("rated").with("Neptune#ml.prediction").hasLabel("movie")

g.with("Neptune#ml.endpoint","node-prediction-movie-lens-endpoint")
 .with("Neptune#ml.iamRoleArn","arn:aws:iam::0123456789:role/sagemaker-role")
 .V("movie_1")
 .in("rated").with("Neptune#ml.prediction").hasLabel("user")
```

在連結預測查詢中使用感應式推論

假設您要在 Jupyter 筆記本中將新節點添加到現有圖形中，如下所示：

```
%%gremlin
g.addV('label1').property(id,'101').as('newV1')
 .addV('label2').property(id,'102').as('newV2')
 .V('1').as('oldV1')
 .V('2').as('oldV2')
 .addE('eLabel1').from('newV1').to('oldV1')
 .addE('eLabel2').from('oldV2').to('newV2')
```

然後，您可以使用感應性推論查詢來預測頭節點，同時考慮到新節點：

```
%%gremlin
g.with("Neptune#ml.endoint", "lp-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('101').out("eLabel1")
 .with("Neptune#ml.prediction")
 .with("Neptune#ml.inductiveInference")
 .hasLabel("label2")
```

結果：

```
==>V[2]
```

同樣地，您可以使用感應式推論查詢來預測尾端節點，同時考慮到新節點：

```
%%gremlin
g.with("Neptune#ml.endoint", "lp-ep")
 .with("Neptune#ml.iamRoleArn", "arn:aws:iam::123456789012:role/NeptuneMLRole")
 .V('102').in("eLabel2")
 .with("Neptune#ml.prediction")
 .with("Neptune#ml.inductiveInference")
 .hasLabel("label1")
```

結果：

```
==>V[1]
```

Neptune ML 小鬼推論查詢的例外狀況清單

- **BadRequestException**— 無法載入提供之角色的認證。

  訊息:`Unable to load credentials for role: the specified IAM Role ARN.`
- **BadRequestException**— 未授權指定的 IAM 角色叫用 SageMaker 端點。

  訊息:`User: the specified IAM Role ARN is not authorized to perform: sagemaker:InvokeEndpoint on resource: the specified endpoint.`
- **BadRequestException**— 指定的端點不存在。

  訊息:`Endpoint the specified endpoint not found.`
- **InternalFailureException**— 無法從 Amazon S3 擷取 Neptune ML 即時感應式推論中繼資料。

  訊息:`Unable to fetch Neptune ML - Real-Time Inductive Inference metadata from S3. Check the permissions of the S3 bucket or if the Neptune instance can connect to S3.`
- **InternalFailureException**— Neptune ML 無法在 Amazon S3 中找到用於即時感應推論的中繼資料檔案。

  訊息:`Neptune ML cannot find the metadata file for Real-Time Inductive Inference in S3.`
- **InvalidParameterException**— 指定的端點在語法上無效。

  訊息:`Invalid endpoint provided for external service query.`
- **InvalidParameterException**— 指定的 SageMaker 執行 IAM 角色 ARN 在語法上無效。

  訊息:`Invalid IAM role ARN provided for external service query.`
- **InvalidParameterException**— 在查詢的`properties()`步驟中指定了多個屬性索引鍵。

  訊息:`ML inference queries are currently supported for one property key.`
- **InvalidParameterException**— 在查詢中指定多個邊緣標籤。

  訊息:`ML inference are currently supported only with one edge label.`
- **InvalidParameterException**— 在查詢中指定了多個頂點標籤條件約束。

  訊息:`ML inference are currently supported only with one vertex label constraint.`
- **InvalidParameterException**—`Neptune#ml.classification` 和`Neptune#ml.regression`述詞都存在於相同的查詢中。

  訊息:`Both regression and classification ML predicates cannot be specified in the query.`
- **InvalidParameterException**— 在連結預測查詢的`in()`或`out()`步驟中指定了多個邊緣標籤。

  訊息:`ML inference are currently supported only with one edge label.`
- **InvalidParameterException**— 使用 Neptune #ml .score 指定了一個以上的內容索引鍵。

  訊息:`Neptune ML inference queries are currently supported for one property key and one Neptune#ml.score property key.`
- **MissingParameterException**— 未在查詢中指定端點，或未指定為資料庫叢集參數。

  訊息:`No endpoint provided for external service query.`
- MissingParameterException— 未在查詢中指定 SageMaker 執行 IAM 角色，也未指定為資料庫叢集參數。

訊息:No IAM role ARN provided for external service query.

- **MissingParameterException**— 查詢中的properties()步驟遺失內容金鑰。

  訊息:Property key needs to be specified using properties() step for ML inference queries.

- **MissingParameterException**— 未在連結預測查詢的in()或out()步驟中指定邊緣標籤。

  訊息:Edge label needs to be specified while using in() or out() step for ML inference queries.

- **MissingParameterException**— 未使用 Neptune #ml .score 指定任何內容索引鍵。

  訊息:Property key needs to be specified along with Neptune#ml.score property key while using the properties() step for Neptune ML inference queries.

- **UnsupportedOperationException**—both() 步驟用於連結預測查詢。

  訊息:ML inference queries are currently not supported with both() step.

- **UnsupportedOperationException**— 未在連結預測查詢中使用in()或has()步out()驟指定預測的頂點標籤。

  訊息:Predicted vertex label needs to be specified using has() step for ML inference queries.

- **UnsupportedOperationException**— 目前未最佳化的步驟不支援 Gremlin ML 感應式推論查詢。

  訊息:Neptune ML - Real-Time Inductive Inference queries are currently not supported with Gremlin steps which are not optimized for Neptune. Check the Neptune User Guide for a list of Neptune-optimized steps.

- **UnsupportedOperationException**— 目前在repeat步驟內不支援 Neptune ML 推論查詢。

  訊息:Neptune ML inference queries are currently not supported inside a repeat step.

- **UnsupportedOperationException**— 每個 Grinlin 查詢目前不支援一個以上的 Neptune ML 推論查詢。

  訊息:Neptune ML inference queries are currently supported only with one ML inference query per gremlin query.

Neptune ML 中的 SPARQL 推論查詢

Neptune ML 將 RDF 圖形映射到屬性圖中以模型 ML 任務。目前，它支援下列使用案例：

- 物件分類 — 預測物件的分類特徵。
- 物件迴歸 — 預測物件的數值性質。
- 物件預測 — 預測給定主題和關係的物件。
- 主旨預測 — 預測給定物件和關係的主體。

> **Note**
>
> Neptune ML 不支援使用 SPARQL 的主體分類和回歸使用案例。

SPARQL 推論查詢中使用的 Neptune ML 述詞

下列述詞可與 SPARQL 推論搭配使用：

neptune-ml:timeout謂詞

指定與遠端伺服器的連線逾時。不應與查詢請求超時混淆，這是服務器滿足請求所需的最大時間。

請注意，如果在neptune-ml:timeout述詞指定的服務逾時發生之前發生查詢逾時，也會取消服務連線。

neptune-ml:outputClass謂詞

neptune-ml:outputClass謂詞僅用於定義用於對象預測的預測對象的類或子預測主題的預測對象的類。

neptune-ml:outputScore謂詞

neptune-ml:outputScore述詞是正數，代表機器學習模型輸出正確的可能性。

neptune-ml:modelType謂詞

述neptune-ml:modelType詞指定了正在訓練的機器學習模型的類型：

- OBJECT_CLASSIFICATION
- OBJECT_REGRESSION
- OBJECT_PREDICTION
- SUBJECT_PREDICTION

neptune-ml:input謂詞

述neptune-ml:input詞是指用作 Neptune ML 輸入的 URI 清單。

neptune-ml:output謂詞

neptune-ml:output謂詞指的是 Neptune ML 返回結果的綁定集的列表。

neptune-ml:predicate謂詞

根據正在執行的任務，neptune-ml:predicate謂詞的使用方式不同：

- 對於對象或主體預測：定義謂詞的類型（邊緣或關係類型）。
- 對於對象分類和回歸：定義我們要預測的文字（屬性）。

neptune-ml:batchSize謂詞

neptune-ml:batchSize指定遠端服務呼叫的輸入大小。

物件分類範例

對於 Neptune ML 中的 SPARQL 物件分類，模型會根據其中一個述詞值進行訓練。當該謂詞尚未與給定主題一起存在時，這很有用。

只能使用物件分類模型推斷分類述詞值。

下面 < http://www.example.org/team > 的查詢旨在預測類型的所有輸入謂詞值foaf:Person：

```
SELECT * WHERE { ?input a foaf:Person .
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:modelType 'OBJECT_CLASSIFICATION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/team> ;
                      neptune-ml:output ?output .
  }
}
```

此查詢可自訂如下：

```
SELECT * WHERE { ?input a foaf:Person .
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:endpoint 'node-prediction-account-balance-endpoint' ;
                      neptune-ml:iamRoleArn 'arn:aws:iam::0123456789:role/sagemaker-role' ;

                      neptune-ml:batchSize "40"^^xsd:integer ;
                      neptune-ml:timeout "1000"^^xsd:integer ;

                      neptune-ml:modelType 'OBJECT_CLASSIFICATION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/team> ;
                      neptune-ml:output ?output .
  }
}
```

物件迴歸範例

物件迴歸類似於物件分類，不同之處在於從每個節點的回歸模型推斷出一個數值述詞。您可以針對物件迴歸使用與物件分類相同的 SPARQL 查詢，但the Neptune#ml.limitNeptune#ml.threshold與述詞不適用的例外。

下面 < http://www.example.org/accountbalance > 的查詢旨在預測類型的所有輸入謂詞值foaf:Person：

```
SELECT * WHERE { ?input a foaf:Person .
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:modelType 'OBJECT_REGRESSION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/accountbalance> ;
                      neptune-ml:output ?output .
  }
}
```

此查詢可自訂如下：

```
SELECT * WHERE { ?input a foaf:Person .
```

```
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:endpoint 'node-prediction-account-balance-endpoint' ;
                      neptune-ml:iamRoleArn 'arn:aws:iam::0123456789:role/sagemaker-role' ;

                      neptune-ml:batchSize "40"^^xsd:integer ;
                      neptune-ml:timeout "1000"^^xsd:integer ;

                      neptune-ml:modelType 'OBJECT_REGRESSION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/accountbalance> ;
                      neptune-ml:output ?output .
  }
}
```

物件預測範例

對象預測預測一個給定的主題和謂詞的對象值。

下面的對象預測查詢旨在預測類型的輸入`foaf:Person`會喜歡什麼電影：

```
?x a foaf:Person .
?x   <http://www.example.org/likes> ?m .
?m a <http://www.example.org/movie> .

## Query
SELECT * WHERE { ?input a foaf:Person .
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:modelType 'OBJECT_PREDICTION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/likes> ;
                      neptune-ml:output ?output ;
                      neptune-ml:outputClass <http://www.example.org/movie> .
  }
}
```

查詢本身可以自定義如下：

```
SELECT * WHERE { ?input a foaf:Person .
  SERVICE neptune-ml:inference {
    neptune-ml:config neptune-ml:endpoint 'node-prediction-user-movie-prediction-
endpoint' ;
                      neptune-ml:iamRoleArn 'arn:aws:iam::0123456789:role/sagemaker-role' ;

                      neptune-ml:limit "5"^^xsd:integer ;
                      neptune-ml:batchSize "40"^^xsd:integer ;
                      neptune-ml:threshold "0.1"^^xsd:double ;
                      neptune-ml:timeout "1000"^^xsd:integer ;
                      neptune-ml:outputScore ?score ;

                      neptune-ml:modelType 'OBJECT_PREDICTION' ;
                      neptune-ml:input ?input ;
                      neptune-ml:predicate <http://www.example.org/likes> ;
                      neptune-ml:output ?output ;
                      neptune-ml:outputClass <http://www.example.org/movie> .
  }
}
```

主體預測範例

主題預測預測給定謂詞和對象的主題。

例如，下面的查詢預測誰（類型foaf:User）將觀看給定的電影：

```
SELECT * WHERE { ?input (a foaf:Movie) .
    SERVICE neptune-ml:inference {
        neptune-ml:config neptune-ml:modelType 'SUBJECT_PREDICTION' ;
                          neptune-ml:input ?input ;
                          neptune-ml:predicate <http://aws.amazon.com/neptune/csv2rdf/
object_Property/rated> ;
                          neptune-ml:output ?output ;
                          neptune-ml:outputClass <http://aws.amazon.com/neptune/csv2rdf/
class/User> ;        }
}
```

Neptune ML SPARQL 推論查詢的例外狀況清單

- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference expects at least 1 value for the parameter *(parameter name)*, found zero.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference expects at most 1 value for the parameter *(parameter name)*, found *(a number)* values.
- **BadRequestException**— 訊息:Invalid predicate *(predicate name)* provided for external service http://aws.amazon.com/neptune/vocab/v01/services/ml#inference query.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference expects the predicate *(predicate name)* to be defined.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference expects the value of (parameter) *(parameter name)* to be a variable, found: *(type)*"
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference expects the input *(parameter name)* to be a constant, found: *(type)*.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference is expected to return only 1 value.
- **BadRequestException**— 訊息:"The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference only allows StatementPatternNodes.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference does not allow the predicate *(predicate name)*.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference predicates cannot be variables, found: *(type)*.
- **BadRequestException**— 訊息:The SERVICE http://aws.amazon.com/neptune/vocab/v01/services/ml#inference predicates are expected to be part of the namespace *(namespace name)*, found: *(namespace name)*.

Neptune ML 管理 API 參考

內容

- 使用dataprocessing指令進行資料處理 (p. 753)
  - 使用 Neptune MLdataprocessing 命令建立資料處理工作 (p. 753)
  - 使用 Neptune MLdataprocessing 命令取得資料處理工作的狀態 (p. 754)
  - 使用 Neptune MLdataprocessing 命令停止資料處理工作 (p. 755)
  - 使用 Neptune MLdataprocessing 命令列出作用中的資料處理工作 (p. 755)
- 使用modeltraining指令進行模型訓練 (p. 757)
  - 使用 Neptune MLmodeltraining 命令建立模型訓練工作 (p. 757)
  - 使用 Neptune MLmodeltraining 命令取得模型訓練工作的狀態 (p. 759)
  - 使用 Neptune MLmodeltraining 命令停止模型訓練工作 (p. 760)
  - 使用 Neptune MLmodeltraining 命令列出作用中的模型訓練工作 (p. 760)
- 使用modeltransform指令進行模型轉換 (p. 761)
  - 使用 Neptune MLmodeltransform 命令建立模型轉換工作 (p. 761)
  - 使用 Neptune MLmodeltransform 命令取得模型轉換工作的狀態 (p. 763)
  - 使用 Neptune MLmodeltransform 命令停止模型轉換工作 (p. 763)
  - 使用 Neptune MLmodeltransform 命令列出作用中的模型轉換工作 (p. 764)
- 使用endpoints指令管理推論端點 (p. 765)
  - 使用 Neptune MLendpoints 命令建立推論端點 (p. 765)
  - 使用 Neptune MLendpoints 命令取得推論端點的狀態 (p. 766)
  - 使用 Neptune MLendpoints 命令刪除執行個體端點 (p. 766)
  - 使用 Neptune MLendpoints 命令列出推論端點 (p. 767)
- Neptune ML 管理 API 錯誤和例外狀況 (p. 768)

使用dataprocessing指令進行資料處理

您可以使用 Neptune MLdataprocessing 命令來建立資料處理工作、檢查其狀態、停止工作，或列出所有作用中的資料處理工作。

使用 Neptune MLdataprocessing 命令建立資料處理工作

建立新工作的典型 Neptune MLdataprocessing 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/dataprocessing \
  -H 'Content-Type: application/json' \
  -d '{
        "inputDataS3Location" : "s3://(Amazon S3 bucket name)/(path to your input folder)",
        "id" : "(a job ID for the new job)",
        "processedDataS3Location" : "s3://(S3 bucket name)/(path to your output folder)"
      }'
```

啟動增量重新處理的命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/dataprocessing \
  -H 'Content-Type: application/json' \
  -d '{
        "inputDataS3Location" : "s3://(Amazon S3 bucket name)/(path to your input folder)",
        "id" : "(a job ID for this job)",
        "processedDataS3Location" : "s3://(S3 bucket name)/(path to your output folder)"
        "previousDataProcessingJobId" : "(the job ID of a previously completed job to
 update)"
}'
```

dataprocessing工作建立的參數

- **id**— (選擇性) 新工作的唯一識別碼。

  類型：字符串。預設值：自動產生的 UUID。
- **previousDataProcessingJobId**— (選擇性) 在舊版資料上執行的已完成資料處理工作的作業 ID。

  類型：字符串。預設值：無。

  附註：使用此項目進行增量資料處理，以便在圖形資料發生變更時更新模型 (但資料已刪除時則不適用)。
- **inputDataS3Location**— (必要) 您要 SageMaker下載執行資料處理任務所需資料之 Amazon S3 位置的 URI。

  類型：字符串。
- **processedDataS3Location**— (必要)。您想要用於儲存 SageMaker資料處理任務結果。

  類型：字符串。
- **sagemakerIamRoleArn**— (選用) 要 SageMaker 執行之 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **neptuneIamRoleArn**— (SageMaker可選)，可以擔任這個 IAM 角色來代表您執行任務。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **processingInstanceType**— (選擇性) 資料處理期間使用的 ML 執行個體類型。它的內存應足夠大，以容納處理過的數據集。

類型：字符串。預設：記憶體大於磁碟上匯出圖形資料大小十倍的最小ml.r5類型。

注意：Neptune ML 可以自動選取執行個體類型。請參閱 選取資料處理的執行個體 (p. 669)。

- **processingInstanceVolumeSizeInGB**— (選擇性) 處理執行個體的磁碟區大小。輸入數據和處理過的數據都存儲在磁盤上，因此卷大小必須足夠大以容納兩個數據集。

  類型：整數。預設：0。

  注意：如果未指定或 0，Neptune ML 會根據資料大小自動選擇磁碟區大小。
- **processingTimeOutInSeconds**— (選擇性) 資料處理工作的逾時 (以秒為單位)。

  類型：整數。預設值為 1 小時86,400 (1 天)。
- **modelType**— (選用) Neptune ML 目前支援的兩種模型類型之一：異質圖形模型 (heterogeneous) 和知識圖 (kge)。

  類型：字符串。預設值：無。

  附註：如果未指定，Neptune ML 會根據資料自動選擇模型類型。
- **configFileName**— (選用) 描述如何載入匯出的圖形資料以進行訓練的資料規格檔案。Neptune 匯出工具組會自動產生檔案。

  類型：字符串。預設：training-data-configuration.json。
- **subnets**— (選用) Neptune VPC 中子網路的識別碼。

  類型：字串清單。預設值：無。
- **securityGroupIds**— (選用) VPC 安全性群組識別碼。

  類型：字串清單。預設值：無。
- **volumeEncryptionKMSKey**— (選用AWS KMS) 金鑰， SageMaker 用於加密連接到執行處理任務的 ML 運算執行個體的儲存磁碟區資料。AWS Key Management Service

  類型：字符串。預設值：無。
- **enableInterContainerTrafficEncryption**— (選擇性) 在訓練或超參數調整工作中啟用或停用容器間流量加密。

  類型：布林值。預設值：真。

  > **Note**
  >
  > 此enableInterContainerTrafficEncryption參數僅適用於引擎版本 1.2.0.2.R3 (p. 874)。
- **s3OutputEncryptionKMSKey**— (選擇性) SageMaker 用來加密訓練工作輸出的AWS Key Management Service (AWS KMS) 金鑰。

  類型：字符串。預設值：無。

使用 Neptune MLdataprocessing 命令取得資料處理工作的狀態

工作狀態的範例 Neptune MLdataprocessing 命令如下所示：

```
curl -s \
  "https://(your Neptune endpoint)/ml/dataprocessing/(the job ID)" \
  | python -m json.tool
```

dataprocessing工作狀態的參數

- **id**— (必要) 資料處理工作的唯一識別碼。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用 Neptune MLdataprocessing 命令停止資料處理工作

用於停止工作的範例 Neptune MLdataprocessing 命令如下所示：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/dataprocessing/(the job ID)"
```

或者這個：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/dataprocessing/(the job ID)?clean=true"
```

dataprocessing停止工作的參數

- **id**— (必要) 資料處理工作的唯一識別碼。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **clean**— (選擇性) 此旗標指定在任務停止時應刪除所有 Amazon S3 成品。

  類型：布林值。預設：FALSE。

使用 Neptune MLdataprocessing 命令列出作用中的資料處理工作

列出使用中工作的範例 Neptune MLdataprocessing 命令如下所示：

```
curl -s "https://(your Neptune endpoint)/ml/dataprocessing"
```

或者這個：

```
curl -s "https://(your Neptune endpoint)/ml/dataprocessing?maxItems=3"
```

dataprocessing清單工作的參數

- **maxItems**— (選擇性) 要傳回的最大項目數。

  類型：整數。預設：10。允許的最大值：1024。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用modeltraining指令進行模型訓練

您可以使用 Neptune MLmodeltraining 命令來建立模型訓練工作、檢查其狀態、停止工作，或列出所有使用中的模型訓練工作。

使用 Neptune MLmodeltraining 命令建立模型訓練工作

用於創建全新工作的 Neptune MLmodeltraining 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
      }'
```

用於建立增量模型訓練的更新工作的 Neptune MLmodeltraining 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
        "previousModelTrainingJobId" : "(the job ID of a completed model-training job to
 update)",
      }'
```

使用使用者提供的自訂模型實作建立新工作的 Neptune MLmodeltraining 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltraining
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "dataProcessingJobId" : "(the data-processing job-id of a completed job)",
        "trainModelS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-graph-
autotrainer"
        "modelName": "custom",
        "customModelTrainingParameters" : {
          "sourceS3DirectoryPath": "s3://(your Amazon S3 bucket)/(path to your Python
 module)",
          "trainingEntryPointScript": "(your training script entry-point name in the Python
 module)",
          "transformEntryPointScript": "(your transform script entry-point name in the
 Python module)"
        }
      }'
```

modeltraining工作建立的參數

- **id**— (選擇性) 新工作的唯一識別碼。

  類型：字符串。預設值：自動產生的 UUID。

- **dataProcessingJobId**— (必要) 已完成資料處理工作的工作 ID，該工作已建立訓練將使用的資料。

  類型：字符串。
- **trainModelS3Location**— (必要) Amazon S3 中要存放模型成品的位置。

  類型：字符串。
- **previousModelTrainingJobId**— (選擇性) 您要根據更新的資料增量更新之已完成模型訓練工作的工作 ID。

  類型：字符串。預設值：無。
- **sagemakerIamRoleArn**— (選用) 要 SageMaker 執行之 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **modelName**— (選用) 用於訓練的模型類型。默認情況下，ML 模型是自動基於數據處理中`modelType`使用的，但您可以在這裡指定不同的模型類型。

  類型：字符串。預設值：`rgcn`用於異質圖表和`kge`知識圖表。有效值：對於異質圖表：`rgcn`。對於`kge`圖形：`transedistmult`、或`rotate`。對於自定義模型實現：`custom`。
- **baseProcessingInstanceType**— (選擇性) 用於準備和管理 ML 模型訓練的 ML 執行個體類型。

  類型：字符串。注意：這是根據記憶體需求選擇的 CPU 執行個體，用於處理訓練資料和模型。請參閱 選取用於模型訓練和模型轉換的執行個體 (p. 669)。
- **trainingInstanceType**— (選用) 用於模型訓練的 ML 執行個體類型。所有 Neptune 機器學習型號都支援 CPU、GPU 和多 GPU 訓練。

  類型：字符串。預設：`ml.p3.2xlarge`。

  附註：選擇適合訓練的執行個體類型取決於工作類型、圖表大小和您的預算。請參閱 選取用於模型訓練和模型轉換的執行個體 (p. 669)。
- **trainingInstanceVolumeSizeInGB**— (選擇性) 訓練執行個體的磁碟區大小。輸入數據和輸出模型都存儲在磁盤上，因此卷大小必須足夠大以容納兩個數據集。

  類型：整數。預設：`0`。

  備註：如果未指定或 0，Neptune ML 會根據資料處理步驟中產生的建議選取磁碟區大小。請參閱 選取用於模型訓練和模型轉換的執行個體 (p. 669)。
- **trainingTimeOutInSeconds**— (選擇性) 訓練工作的逾時時間 (以秒為單位)。

  類型：整數。預設值：`86,400`(1 天)。
- **maxHPONumberOfTrainingJobs**— 超參數調整工作要開始的訓練工作總數上限。

  類型：整數。預設：`2`。

  注意：Neptune ML 會自動調整機器學習模型的超參數。若要取得效能良好的模型，請至少使用 10 個工作(換句話說，設`maxHPONumberOfTrainingJobs`為 10)。在一般情況下，調整運行越多，結果越好。
- **maxHPOParallelTrainingJobs**— 超參數調整工作要啟動的 parallel 訓練工作數目上限。

  類型：整數。預設：`2`。

  注意：您可以執行的 parallel 工作數量受訓練執行個體上可用資源的限制。
- **subnets**— (選用) Neptune VPC 中子網路的識別碼。

  類型：字串清單。預設值：無。

- **securityGroupIds**— (選用) VPC 安全性群組識別碼。

  類型：字串清單。預設值：無。
- **volumeEncryptionKMSKey**— (選用) 使用的AWS Key Management Service (AWS KMS) 金鑰，SageMaker 用於加密連接到執行訓練任務的 ML 運算執行個體的儲存磁碟區資料。

  類型：字符串。預設值：無。
- **s3OutputEncryptionKMSKey**— (選擇性) SageMaker 用來加密處理工作輸出的AWS Key Management Service (AWS KMS) 金鑰。

  類型：字符串。預設值：無。
- **enableInterContainerTrafficEncryption**— (選擇性) 在訓練或超參數調整工作中啟用或停用容器間流量加密。

  類型：布林值。預設值：真。

  > Note
  >
  > 此`enableInterContainerTrafficEncryption`參數僅適用於[引擎版本 1.2.0.2.R3 (p. 874)](#)。
- **enableManagedSpotTraining**— (選用) 使用 Amazon 彈性運算雲端競價型執行個體，將訓練機器學習模型的成本最佳化。如需詳細資訊，請參閱 Amazon 的 Managagining (Amazon) SageMaker。

  類型：布林值。預設值：假。
- **customModelTrainingParameters**— (選用) 自訂模型訓練的組態。此為具有下列欄位的 JSON 物件：
  - **sourceS3DirectoryPath**— (必要) 到實作模型之 Python 模組所在之 Amazon S3 位置的路徑。這必須指向有效的現有 Amazon S3 位置，其中至少包含訓練指令碼、轉換指令碼和`model-hpo-configuration.json`檔案。
  - **trainingEntryPointScript**— (選擇性) 指令碼模組中的進入點名稱，該指令碼會執行模型訓練，並將超參數作為命令列引數 (包括固定的超參數)。

    預設：`training.py`。
  - **transformEntryPointScript**— (選擇性) 在識別超參數搜尋的最佳模型之後，應執行指令碼模組中的進入點名稱，以計算模型部署所需的模型加工品。它應該能夠在沒有命令行參數的情況下運行。

    預設：`transform.py`。
- **maxWaitTime**— (選擇性) 使用競價型執行個體執行模型訓練時，等待的時間上限 (以秒為單位)。應大於`trainingTimeOutInSeconds`。

  類型：整數。

使用 Neptune ML`modeltraining` 命令取得模型訓練工作的狀態

工作狀態的範例 Neptune ML`modeltraining` 命令如下所示：

```
curl -s \
  "https://(your Neptune endpoint)/ml/modeltraining/(the job ID)" \
  | python -m json.tool
```

`modeltraining`工作狀態的參數

- **id**— (必要) 模型訓練工作的唯一識別元。

  類型：字符串。

- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用 Neptune MLmodeltraining 命令停止模型訓練工作

用於停止工作的範例 Neptune MLmodeltraining 命令如下所示：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/modeltraining/(the job ID)"
```

或者這個：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/modeltraining/(the job ID)?clean=true"
```

modeltraining停止工作的參數

- **id**— (必要) 模型訓練工作的唯一識別元。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **clean**— (選擇性) 此旗標指定在任務停止時應刪除所有 Amazon S3 成品。

  類型：布林值。預設：FALSE。

使用 Neptune MLmodeltraining 命令列出作用中的模型訓練工作

列出使用中工作的範例 Neptune MLmodeltraining 命令如下所示：

```
curl -s "https://(your Neptune endpoint)/ml/modeltraining" | python -m json.tool
```

或者這個：

```
curl -s "https://(your Neptune endpoint)/ml/modeltraining?maxItems=3" | python -m json.tool
```

modeltraining清單工作的參數

- **maxItems**— (選擇性) 要傳回的最大項目數。

  類型：整數。預設：10。允許的最大值：1024。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用modeltransform指令進行模型轉換

您可以使用 Neptune MLmodeltransform 命令來建立模型轉換工作、檢查其狀態、停止工作或列出所有作用中的模型轉換工作。

使用 Neptune MLmodeltransform 命令建立模型轉換工作

用於建立增量轉換工作 (不重新訓練模型) 的 Neptune MLmodeltransform 命令看起來像這樣：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-transform job ID)",
        "dataProcessingJobId" : "(the job-id of a completed data-processing job)",
        "mlModelTrainingJobId" : "(the job-id of a completed model-training job)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform"
      }'
```

Neptune MLmodeltransform 命令用於從已完成的 SageMaker 訓練工作建立工作如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-transform job ID)",
        "trainingJobName" : "(name of a completed SageMaker training job)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform",
        "baseProcessingInstanceType" : ""
      }'
```

用於建立使用自訂模型實作的工作的 Neptune MLmodeltransform 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/modeltransform
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique model-training job ID)",
        "trainingJobName" : "(name of a completed SageMaker training job)",
        "modelTransformOutputS3Location" : "s3://(your Amazon S3 bucket)/neptune-model-
transform/"
        "customModelTransformParameters" : {
          "sourceS3DirectoryPath": "s3://(your Amazon S3 bucket)/(path to your Python
 module)",
          "transformEntryPointScript": "(your transform script entry-point name in the
 Python module)"
        }
      }'
```

modeltransform工作建立的參數

- **id**— (選擇性) 新工作的唯一識別碼。

  類型：字符串。預設值：自動產生的 UUID。
- **dataProcessingJobId**— 已完成資料處理工作的作業 ID。

  類型：字符串。

附註：您必須同時包含dataProcessingJobId和mlModelTrainingJobId、或trainingJobName。

- **mlModelTrainingJobId**— 已完成模型訓練工作的工作 ID。

  類型：字符串。

  附註：您必須同時包含dataProcessingJobId和mlModelTrainingJobId、或trainingJobName。

- **trainingJobName**— 已完成 SageMaker 訓練工作的名稱。

  類型：字符串。

  注意：您必須同時包括dataProcessingJobId和mlModelTrainingJobId參數或trainingJobName參數。

- **sagemakerIamRoleArn**— (選用) 要 SageMaker 執行之 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

- **customModelTransformParameters** — (選用) 使用自訂模型之模型轉換的組態資訊。customModelTransformParameters物件包含下列欄位，其值必須與訓練工作中儲存的模型參數相容：

  - **sourceS3DirectoryPath**— (必要) 到實作模型之 Python 模組所在之 Amazon S3 位置的路徑。這必須指向有效的現有 Amazon S3 位置，其中至少包含訓練指令碼、轉換指令碼和model-hpo-configuration.json檔案。
  - **transformEntryPointScript**— (選擇性) 在識別超參數搜尋的最佳模型之後，應執行指令碼模組中的進入點名稱，以計算模型部署所需的模型加工品。它應該能夠在沒有命令行參數的情況下運行。

    預設：transform.py。

- **baseProcessingInstanceType**— (選擇性) 用於準備和管理 ML 模型訓練的 ML 執行個體類型。

  類型：字符串。注意：這是根據處理轉換資料和模型的記憶體需求而選擇的 CPU 執行個體。請參閱 選取用於模型訓練和模型轉換的執行個體 (p. 669)。

- **baseProcessingInstanceVolumeSizeInGB**— (選擇性) 訓練執行個體的磁碟區大小。輸入數據和輸出模型都存儲在磁盤上，因此卷大小必須足夠大以容納兩個數據集。

  類型：整數。預設：0。

  備註：如果未指定或 0，Neptune ML 會根據資料處理步驟中產生的建議選取磁碟區大小。請參閱 選取用於模型訓練和模型轉換的執行個體 (p. 669)。

- **subnets**— (選用) Neptune VPC 中子網路的識別碼。

  類型：字串清單。預設值：無。

- **securityGroupIds**— (選用) VPC 安全性群組識別碼。

  類型：字串清單。預設值：無。

- **volumeEncryptionKMSKey**— (選用AWS KMS) 金鑰， SageMaker 用於加密連接到執行轉換任務的 ML 運算執行個體的儲存磁碟區資料。AWS Key Management Service

  類型：字符串。預設值：無。

- **enableInterContainerTrafficEncryption**— (選擇性) 在訓練或超參數調整工作中啟用或停用容器間流量加密。

  類型：布林值。預設值：真。

Note

此enableInterContainerTrafficEncryption參數僅適用於引擎版本 1.2.0.2.R3 (p. 874)。

- **s3OutputEncryptionKMSKey**— (選擇性) SageMaker 用來加密處理工作輸出的AWS Key Management Service (AWS KMS) 金鑰。

  類型：字符串。預設值：無。

使用 Neptune MLmodeltransform 命令取得模型轉換工作的狀態

工作狀態的範例 Neptune MLmodeltransform 命令如下所示：

```
curl -s \
  "https://(your Neptune endpoint)/ml/modeltransform/(the job ID)" \
  | python -m json.tool
```

modeltransform工作狀態的參數

- **id**— (必要) 模型轉換工作的唯一識別碼。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用 Neptune MLmodeltransform 命令停止模型轉換工作

用於停止工作的範例 Neptune MLmodeltransform 命令如下所示：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/modeltransform/(the job ID)"
```

或者這個：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/modeltransform/(the job ID)?clean=true"
```

modeltransform停止工作的參數

- **id**— (必要) 模型轉換工作的唯一識別碼。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。
- **clean**— (選擇性) 此旗標指定在任務停止時應刪除所有 Amazon S3 成品。

  類型：布林值。預設：FALSE。

使用 Neptune MLmodeltransform 命令列出作用中的模型轉換工作

列出使用中工作的範例 Neptune MLmodeltransform 命令如下所示：

```
curl -s "https://(your Neptune endpoint)/ml/modeltransform" | python -m json.tool
```

或者這個：

```
curl -s "https://(your Neptune endpoint)/ml/modeltransform?maxItems=3" | python -m
 json.tool
```

modeltransform清單工作的參數

- **maxItems**— (選擇性) 要傳回的最大項目數。

  類型：整數。預設：10。允許的最大值：1024。
- **neptuneIamRoleArn**— (選用) IAM 角色的 ARN，可提供 Neptune SageMaker 和 Amazon S3 資源的存取權。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會發生錯誤。

使用endpoints指令管理推論端點

您可以使用 Neptune MLendpoints 命令來建立推論端點、檢查其狀態、刪除或列出現有的推論端點。

使用 Neptune MLendpoints 命令建立推論端點

Neptune MLendpoints 命令，用於從訓練工作所建立的模型建立推論端點，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "mlModelTrainingJobId": "(the model-training job-id of a completed job)"
      }'
```

Neptune MLendpoints 命令，用於從訓練工作建立的模型更新現有推論端點，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "update" : "true",
        "mlModelTrainingJobId": "(the model-training job-id of a completed job)"
      }'
```

用於從模型轉換工作所建立的模型建立推論端點的 Neptune MLendpoints 命令如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "mlModelTransformJobId": "(the model-training job-id of a completed job)"
      }'
```

Neptune MLendpoints 命令，用於從模型轉換工作所建立的模型更新現有推論端點，如下所示：

```
curl \
  -X POST https://(your Neptune endpoint)/ml/endpoints
  -H 'Content-Type: application/json' \
  -d '{
        "id" : "(a unique ID for the new endpoint)",
        "update" : "true",
        "mlModelTransformJobId": "(the model-training job-id of a completed job)"
      }'
```

用於建立**endpoints**推論端點的參數

- **id**— (選用) 新推論端點的唯一識別碼。

  類型：字符串。預設值：自動產生的時間戳記名稱。
- **mlModelTrainingJobId**— 已完成模型訓練工作的工作 ID，該工作已建立推論端點將指向的模型。

  類型：字符串。

  備註：您必須提供mlModelTrainingJobId或mlModelTransformJobId。

- **mlModelTransformJobId**— 已完成的模型轉換工作的工作 ID。

  類型：字符串。

  備註：您必須提供mlModelTrainingJobId或mlModelTransformJobId。
- **update**— (選擇性) 如果存在，此參數表示這是更新要求。

  類型：布林值。預設：false

  備註：您必須提供mlModelTrainingJobId或mlModelTransformJobId。
- **neptuneIamRoleArn**— (選用) 提供 Neptune 存取權限 SageMaker 和 Amazon S3 資源的 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會擲回錯誤。
- **modelName**— (選用) 用於訓練的模型類型。默認情況下，ML 模型是自動基於數據處理中modelType使用的，但您可以在這裡指定不同的模型類型。

  類型：字符串。預設值：rgcn用於異質圖表和kge知識圖表。有效值：對於異質圖表：rgcn。對於知識圖：kgetranse、distmult、或rotate。
- **instanceType**— (選擇性) 用於線上服務的 ML 執行個體類型。

  類型：字符串。預設：ml.m5.xlarge。

  備註：為推論端點選擇 ML 執行個體，視工作類型、圖形大小和預算而定。請參閱 選取推論端點的執行個體 (p. 669)。
- **instanceCount**— (選用) 要部署到端點進行預測的 Amazon EC2 執行個體的最小數量。

  類型：整數。預設：1。
- **volumeEncryptionKMSKey**— (選用) 使用的AWS Key Management Service (AWS KMS) 金鑰，SageMaker 用於加密連接到執行端點的 ML 運算執行個體的儲存磁碟區資料。

  類型：字符串。預設值：無。

使用 Neptune MLendpoints 命令取得推論端點的狀態

執行個體端點狀態的範例 Neptune MLendpoints 命令如下所示：

```
curl -s \
  "https://(your Neptune endpoint)/ml/endpoints/(the inference endpoint ID)" \
  | python -m json.tool
```

endpoints執行處理-端點狀態的參數

- **id**— (必要) 推論端點的唯一識別碼。

  類型：字符串。
- **neptuneIamRoleArn**— (選用) 提供 Neptune 存取權限 SageMaker 和 Amazon S3 資源的 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會擲回錯誤。

使用 Neptune MLendpoints 命令刪除執行個體端點

用於刪除執行個體端點的範例 Neptune MLendpoints 命令如下所示：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/endpoints/(the inference endpoint ID)"
```

或者這個：

```
curl -s \
  -X DELETE "https://(your Neptune endpoint)/ml/endpoints/(the inference endpoint ID)?
clean=true"
```

`endpoints`刪除推論端點的參數

- **`id`**— (必要) 推論端點的唯一識別碼。

  類型：字符串。
- **`neptuneIamRoleArn`**— (選用) 提供 Neptune 存取權限 SageMaker 和 Amazon S3 資源的 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會擲回錯誤。
- **`clean`**— (選擇性) 表示也應刪除與此端點相關的所有人工因素。

  類型：布林值。預設：FALSE。

使用 Neptune MLendpoints 命令列出推論端點

用於列出推論端點的 Neptune MLendpoints 命令如下所示：

```
curl -s "https://(your Neptune endpoint)/ml/endpoints" \
  | python -m json.tool
```

或者這個：

```
curl -s "https://(your Neptune endpoint)/ml/endpoints?maxItems=3" \
  | python -m json.tool
```

`dataprocessing`清單推論端點的參數

- **`maxItems`**— (選擇性) 要傳回的最大項目數。

  類型：整數。預設：10。允許的最大值：1024。
- **`neptuneIamRoleArn`**— (選用) 提供 Neptune 存取權限 SageMaker 和 Amazon S3 資源的 IAM 角色的 ARN。

  類型：字符串。注意：這必須列在您的資料庫叢集參數群組中，否則會擲回錯誤。

Neptune ML 管理 API 錯誤和例外狀況

所有 Neptune ML 管理 API 例外狀況都會傳回 400 個 HTTP 程式碼。在收到這些例外狀況之後，不應重試產生例外狀況的命令。

- **MissingParameterException**— 錯誤訊息：

  ```
  Required credentials are missing. Please add IAM role to the cluster or pass as a parameter to this request.
  ```
- **InvalidParameterException**— 錯誤訊息：
  - `Invalid ML instance type.`
  - `Invalid ID provided. ID can be 1-48 alphanumeric characters.`
  - `Invalid ID provided. Must contain only letters, digits, or hyphens.`
  - `Invalid ID provided. Please check whether a resource with the given ID exists.`
  - `Another resource with same ID already exists. Please use a new ID.`
  - `Failed to stop the job because it has already completed or failed.`
- **BadRequestException**— 錯誤訊息：
  - `Invalid S3 URL or incorrect S3 permissions. Please check your S3 configuration.`
  - `Provided ModelTraining job has not completed.`
  - `Provided SageMaker Training job has not completed.`
  - `Provided MLDataProcessing job is not completed.`
  - `Provided MLModelTraining job doesn't exist.`
  - `Provided ModelTransformJob doesn't exist.`
  - `Unable to find SageMaker resource. Please check your input.`

Neptune 限制

- 目前支援的推論類型包括節點分類、節點迴歸、邊緣分類、邊迴歸和連結預測 (請參閱Neptune ML DEDEDE (p. 652))。
- Neptune ML 可支援的最大圖形大小取決於資料準備 (p. 669)、模型訓練 (p. 669)和推論 (p. 669)期間所需的記憶體和儲存空間量。
  - SageMaker 資料處理執行個體的記憶體大小上限為 768 GB。因此，如果資料處理階段需要超過 768 GB 的記憶體，就會失敗。
  - SageMaker 訓練執行個體的記憶體大小上限為 732 GB。因此，如果訓練階段需要超過 732 GB 的記憶體，就會失敗。
- SageMaker 端點的推論裝載大小上限為 6 MiB。因此，如果子圖有效負載超過此大小，則感應推斷失敗。
- Neptune ML 目前僅在 Neptune 及其所依賴的其他服務 (例如AWS Lambda Amazon API 閘道和 Amazon SageMaker) 都受到支援的區域提供。

  中國（北京）和中國（寧夏）的差異與 IAM 身份驗證的默認使用有關，如此處所述以及其他差異。
- Neptune ML 所啟動的連結預測推論端點目前只能預測訓練期間圖形中存在之節點的可能連結。

  例如，考慮與`Movie`頂點`User`和`Rated`邊的圖形。使用對應的 Neptune ML 連結預測建議模型，您可以將新使用者新增至圖表，並讓模型為他們預測影片，但模型只能建議模型訓練期間出現的影片。雖然`User`節點嵌入是使用其本機子圖和 GNN 模型即時計算的，因此可能會隨著使用者對電影進行評分而隨時間變更，但它與靜態、預先計算的影片嵌入相比較，以取得最終建議。
- Neptune ML 支援的 KGE 模型僅適用於連結預測工作，而且這些表示特定於訓練期間圖形中存在的頂點和邊緣類型。這表示在訓練期間，推論查詢中所提及的所有頂點和邊類型都必須存在於圖形中。如果不重新訓練模型，則無法對新邊類型或頂點進行預測。

SageMaker 資源限制

視您的活動和資源使用狀況而定，您可能會遇到錯誤訊息，指出您已超過配額 (ResourceLimitExceeded)。而且您需要擴充 SageMaker 資源，請依照此頁面上的資 SageMaker 源要求增加服務配額程序中的步驟，向 SupAWS port 要求增加配額。

SageMaker 資源名稱對應於 Neptune ML 階段，如下所示：

- Neptune 資料處理、模型訓練和模型轉換工作會使用。 SageMaker `ProcessingJob`
- Neptune 模型訓練工作會 SageMaker `HyperParameterTuningJob`使用。
- Neptune 模型訓練工作會 SageMaker `TrainingJob`使用。
- Neptune 推論端點會 SageMaker `Endpoint`使用。

監控亞馬遜海王星資

Amazon Neptune 支援各種監控效能和使用情況的方法：

- 執行個體狀態 — 檢查 Neptune 叢集圖形資料庫引擎的健全狀況、找出已安裝的引擎版本，以及使用執行個體狀態 API 取得其他執行 (p. 770)個體相關資訊。
- 亞馬遜 CloudWatch — 海王星自動將指標發送到警報，CloudWatch並支持CloudWatch警報。如需詳細資訊，請參閱the section called “使用 CloudWatch” (p. 772)。
- 稽核記錄檔 — 使用 Neptune 主控台檢視、下載或監視資料庫記錄檔。如需詳細資訊，請參閱the section called “使用海王星稽核記錄” (p. 781)。
- 將日誌發佈到 Amazon CloudWatch 日誌 — 您可以設定 Neptune 資料庫叢集，將稽核日誌資料發佈到 Amazon CloudWatch 日誌中的日誌群組。使用CloudWatch日誌，您可以對日誌數據進行實時分析，用CloudWatch於創建警報和查看指標，以及使用CloudWatch日誌將日誌記錄存儲在高度耐用的存儲中。請參閱 海王星CloudWatch日誌 (p. 782)。
- AWS CloudTrail— 海王星支持使用 API 日誌記錄CloudTrail. 如需詳細資訊，請參閱the section called “使用記錄海王星 API 呼叫 AWS CloudTrail” (p. 789)。
- 事件通知訂閱 — 訂閱 Neptune 事件以隨時了解正在發生的事情。如需詳細資訊，請參閱the section called “事件通知” (p. 791)。
- 標記 — 使用標籤將中繼資料新增至 Neptune 資源，並根據標籤追蹤使用情況。如需詳細資訊，請參閱the section called “標記海王星資源” (p. 799)。

主題

- 檢查 Neptune 執行個體的健全狀況狀態 (p. 770)
- 使用 Amazon 監控海王星 CloudWatch (p. 772)
- 搭配 Amazon 海王星叢集使用稽核日誌 (p. 781)
- 將海王星日誌發佈到亞馬遜CloudWatch日 (p. 782)
- 為海王星筆記本啟用亞馬遜CloudWatch日誌 (p. 784)
- 使用亞馬遜海王星慢速查詢記錄 (p. 785)
- 使用記錄亞馬遜海王星 API 呼叫 AWS CloudTrail (p. 789)
- 使用海王星事件通知 (p. 791)
- 標記亞馬遜海王星資 (p. 799)

檢查 Neptune 執行個體的健全狀況狀態

Amazon Neptune 提供了一種機制來檢查主機上圖形資料庫的狀態。這也是確認您可以連線到執行個體的好方法。

使用 `curl` 檢查執行個體的運作狀態：

```
curl -G https://your-neptune-endpoint:port/status
```

如果執行個體運作狀態良好，`status` 命令會傳回下列欄位：

- **`status`**— `"healthy"` 如果執行個體沒有發生問題，則設定為。

如果執行個體從當機中復原或重新開機，而且有從最新伺服器關閉時執行的作用中交易，status則會設定為"recovery"。

- **startTime**— 設定為目前伺服器處理序啟動的 UTC 時間。
- **dbEngineVersion**— 設定為在資料庫叢集上執行的 Neptune 引擎版本。

  若此引擎版本自發行以來已手動進行修補，則版本號碼會加上 "Patch-" 前綴。
- **role**— "reader" 如果執行個體是僅供讀取複本，則設定為，或執行個體是"writer"否為主執行個體。
- **dfeQueryEngine**— "enabled" 如果 DFE 引擎 (p. 129)已完全啟用，或者 DFE 引擎僅與useDFE查詢提示設定為 true (預設值) 的查詢搭配使用，viaQueryHint則設定viaQueryHint為。
- **gremlin**— 包含叢集上可用的 Gemlin 查詢語言的相關資訊。具體而言，它包含一個version字段，用於指定引擎正在使用的當前TinkerPop版本。
- **sparql**— 包含叢集上可用之 SPARQL 查詢語言的相關資訊。具體而言，它包含一個version字段，用於指定引擎正在使用的當前 SPARQL 版本。
- **opencypher**— 包含叢集上可用的 OpenCypher 查詢語言的相關資訊。具體而言，它包含一個version字段，用於指定引擎正在使用的當前 OperCypher 版本。
- **labMode**— 包含引擎正在使用的實驗室模式 (p. 127)設定。
- **rollingBackTrxCount**— 如果有要回復的交易，則此欄位會設定為此類交易的數目。如果沒有，則該字段根本不會出現。
- **rollingBackTrxEarliestStartTime**— 設定為要回復之最早交易的開始時間。如果沒有任何交易正在回滾，則該字段根本不會顯示。
- **features**— 包含有關在資料庫叢集上啟用的功能的狀態資訊：
  - **lookupCache**— 的目前狀態查閱快取 (p. 107)。此欄位只會出現在R5d執行個體類型上，因為這些是唯一可以存在查詢快取的執行個體。該字段是一個 JSON 對象的形式如下：

    ```
    "lookupCache": {
      "status": "current lookup cache status"
    }
    ```

    在R5d執行個體上：
    - 如果已啟用查閱快取，則狀態會列為"Available"。
    - 如果查閱快取已停用，則狀態會列為"Disabled"。
    - 如果執行個體已達到磁碟限制，則狀態會列為"Read Only Mode - Storage Limit Reached"。
  - **ResultCache**— 的目前狀態快取查詢結果 (p. 330)。此欄位的格式為 JSON 物件：

    ```
    "ResultCache": {
      "status": "current results cache status"
    }
    ```

    - 如果已啟用結果快取，則狀態會列為"Available"。
    - 如果快取已停用，則狀態會列為"Disabled"。
  - **IAMAuthentication**— 指定是否已在資料庫叢集上啟用 AWS Identity and Access Management (IAM) 身份驗證：
    - 如果已啟用 IAM 身份驗證，則狀態會列為"enabled"。
    - 如果 IAM 身份驗證已停用，則狀態會列為"disabled"。
  - **Streams**— 指定是否已在資料庫叢集上啟用 Neptune 串流：
    - 如果啟用串流，狀態會列為"enabled"。
    - 如果串流已停用，則狀態會列為"disabled"。
  - **AuditLog**— 等於是enabled否已啟用稽核記錄，或以其他方式啟用disabled。

  - **SlowQueryLogs**— 等於info或debug啟用慢速查詢記錄 (p. 785)，或以其他方式。disabled
  - **QueryTimeout**— 查詢逾時的值 (以毫秒為單位)。
- **settings**— 套用至執行個體的設定：
  - **clusterQueryTimeoutInMs**— 針對整個叢集設定的查詢逾時值 (以毫秒為單位)。
  - **SlowQueryLogsThreshold**— 針對整個叢集設定的查詢逾時值 (以毫秒為單位)。

實例 status 命令輸出的示例

以下是來自執行個體 status 命令的輸出範例，(在此情況下，在執行個R5d體上執行)：

```
{
  "status":"healthy",
  "startTime":"Wed Dec 29 02:29:24 UTC 2021",
  "dbEngineVersion":"development",
  "role":"writer",
  "dfeQueryEngine":"viaQueryHint",
  "gremlin":{"version":"tinkerpop-3.5.2"},
  "sparql":{"version":"sparql-1.1"},
  "opencypher":{"version":"Neptune-9.0.20190305-1.0"},
  "labMode":{
    "ObjectIndex":"disabled",
    "ReadWriteConflictDetection":"enabled"
  },
  "features":{
    "ResultCache":{"status":"disabled"},
    "IAMAuthentication":"disabled",
    "Streams":"disabled",
    "AuditLog":"disabled",
    "SlowQueryLogs": "disabled",
  },
  "settings":{
    "clusterQueryTimeoutInMs":"120000",
    "SlowQueryLogsThreshold": "5000",
  }
}
```

若執行個體發生問題，status 命令會傳回 HTTP 500 錯誤碼。如果主機無法連線，請求將逾時。確認您是從虛擬私有雲端 (VPC) 內部存取執行個體，並且您的安全群組允許您存取它。

使用 Amazon 監控海王星 CloudWatch

整合了 Amazon Neptune 和亞馬遜CloudWatch，因此您可以收集和分析效能指標。您可以使用 CloudWatch 主控台、AWS Command Line Interface (AWS CLI) 或 CloudWatch API 監控這些指標。

CloudWatch 也可讓您設定警示，以便在指標值超過指定閾值時收到通知。您甚至可以設置CloudWatch事件以在發生違規時採取糾正措施。如需使用 CloudWatch 和警示的詳細資訊，請參閱 CloudWatch 文件。

主題

- 檢視 CloudWatch 資料 (主控台) (p. 773)
- 檢視 CloudWatch 資料 (AWS CLI) (p. 773)
- 檢視 CloudWatch 資料 (API) (p. 773)
- 用CloudWatch來監視 Neptune 中的資料庫執行個體效 (p. 774)
- 海王星CloudWatch度量 (p. 774)
- 海王星CloudWatch尺寸 (p. 780)

檢視 CloudWatch 資料 (主控台)

若要檢視 Neptune 叢集 (主控台) 的CloudWatch資料

1. 請登入AWS Management Console並開啟CloudWatch主控台，網址為 https://console.aws.amazon.com/cloudwatch/。
2. 在導覽窗格中，選擇 Metrics (指標)。
3. 在「所有測量結果」窗格中，選擇 Neptune，然後選擇「資料庫」ClusterIdentifier。
4. 在上方窗格中，向下捲動以檢視叢集的完整指標清單。可用的 Neptune 測量結果選項會顯示在檢視清單中。

若要選取或取消選取個別指標，請在結果窗格中，選擇資源名稱和指標旁的核取方塊。圖表顯示所選取項目的指標顯示於主控台底部。若要進一步了解CloudWatch圖形，請參閱 Amazon CloudWatch 使用者指南中的圖形指標。

檢視 CloudWatch 資料 (AWS CLI)

若要檢視 Neptune 叢集的CloudWatch資料 (AWS CLI)

1. 安裝 AWS CLI。如需指示，請參閱使AWS Command Line Interface用者指南。
2. 使用 AWS CLI 擷取資訊。中列出了海王星的相關CloudWatch參數海王星CloudWatch度量 (p. 774)。

   以下範例擷取 `gremlin-cluster` 叢集的每秒 Gremlin 請求數的 CloudWatch 指標。

```
aws cloudwatch get-metric-statistics \
    --namespace AWS/Neptune  --metric-name GremlinRequestsPerSec \
    --dimensions Name=DBClusterIdentifier,Value=gremlin-cluster \
    --start-time 2018-03-03T00:00:00Z --end-time 2018-03-04T00:00:00Z \
    --period 60 --statistics=Average
```

檢視 CloudWatch 資料 (API)

CloudWatch 亦支援 `Query` 動作，讓您可以使用程式設計方式請求資訊。如需詳細資訊，請參閱CloudWatch查詢 API 文件和 Amazon CloudWatch API 參考資料。

當CloudWatch動作需要 Neptune 監視專用的參數時 (例如)`MetricName`，請使用中列出的值海王星CloudWatch度量 (p. 774)。

以下範例顯示低階 CloudWatch 請求，使用以下參數：

- `Statistics.member.1 = Average`
- `Dimensions.member.1 = DBClusterIdentifier=gremlin-cluster`
- `Namespace  = AWS/Neptune`
- `StartTime  = 2013-11-14T00:00:00Z`
- `EndTime  = 2013-11-16T00:00:00Z`
- `Period  = 60`
- `MetricName  = GremlinRequestsPerSec`

該 CloudWatch 請求看起來類似如下。不過，這只是顯示請求的形式；您必須根據您的指標與時間範圍建構自己的請求。

```
https://monitoring.amazonaws.com/
      ?SignatureVersion=2
      &Action=GremlinRequestsPerSec
      &Version=2010-08-01
      &StartTime=2018-03-03T00:00:00
      &EndTime=2018-03-04T00:00:00
      &Period=60
      &Statistics.member.1=Average
      &Dimensions.member.1=DBClusterIdentifier=gremlin-cluster
      &Namespace=AWS/Neptune
      &MetricName=GremlinRequests
      &Timestamp=2018-03-04T17%3A48%3A21.746Z
      &AWSAccessKeyId=AWS Access Key ID;
      &Signature=signature
```

用CloudWatch來監視 Neptune 中的資料庫執行個體效

您可以使用 Neptune 中的CloudWatch指標來監視資料庫執行個體上發生的情況，並追蹤資料庫觀察到的查詢佇列長度。下列量度特別有用：

- **CPUUtilization**— 顯示 CPU 使用率的百分比。
- **VolumeWriteIOPs**— 顯示每 5 分鐘報告叢集磁碟區的平均磁碟 I/O 寫入次數。
- **MainRequestQueuePendingRequests**— 顯示在輸入佇列擱置執行中等待的要求數目。

您也可以使用 Gremlin 查詢狀態端點 (p. 324)搭配參數，瞭解伺服器上擱置的要求數目。includeWaiting這將為您提供所有等待查詢的狀態。

下列指標可協助您調整 Neptune 佈建和查詢策略，以提升效率和效能：

- 一致的延遲，高CPUUtilization，高VolumeWriteIOPs和低MainRequestQueuePendingRequests結合表明服務器正在積極參與以可持續的速度處理並發寫請求，只需很少的 I/O 等待。
- 一致的延遲、低CPUUtilization、低VolumeWriteIOPs且沒有MainRequestQueuePendingRequests一起顯示您在主資料庫執行個體上有過多的容量來處理寫入請求。
- 高延遲CPUUtilization和高，VolumeWriteIOPs但可變的延遲，並MainRequestQueuePendingRequests共同顯示您傳送的工作數量超過伺服器在指定時間間隔內所能處理的工作量。請考慮建立批次要求或調整大小，以減少交易額外負荷執行相同的工作量，並/或擴展主執行個體，以增加能夠同時處理寫入要求的查詢執行緒數目。
- 「低CPUUtilization與高」VolumeWriteIOPs 表示查詢執行緒正在等待儲存層的 I/O 作業完成。如果您看到可變的延遲，而且有些增加MainRequestQueuePendingRequests，請考慮建立批次請求或調整大小，以減少交易額外負荷來執行相同的工作量。

海王星CloudWatch度量

> **Note**
>
> Amazon Neptune CloudWatch 只會在指標具有非零值時才會傳送至指標。對於所有 Neptune 度量，彙總粒度為 5 分鐘。

主題

- 海王星CloudWatch度量 (p. 775)
- CloudWatch海王星現在已棄用的量度 (p. 778)

海王星CloudWatch度量

下表列出 Neptune 支援的CloudWatch測量結果。

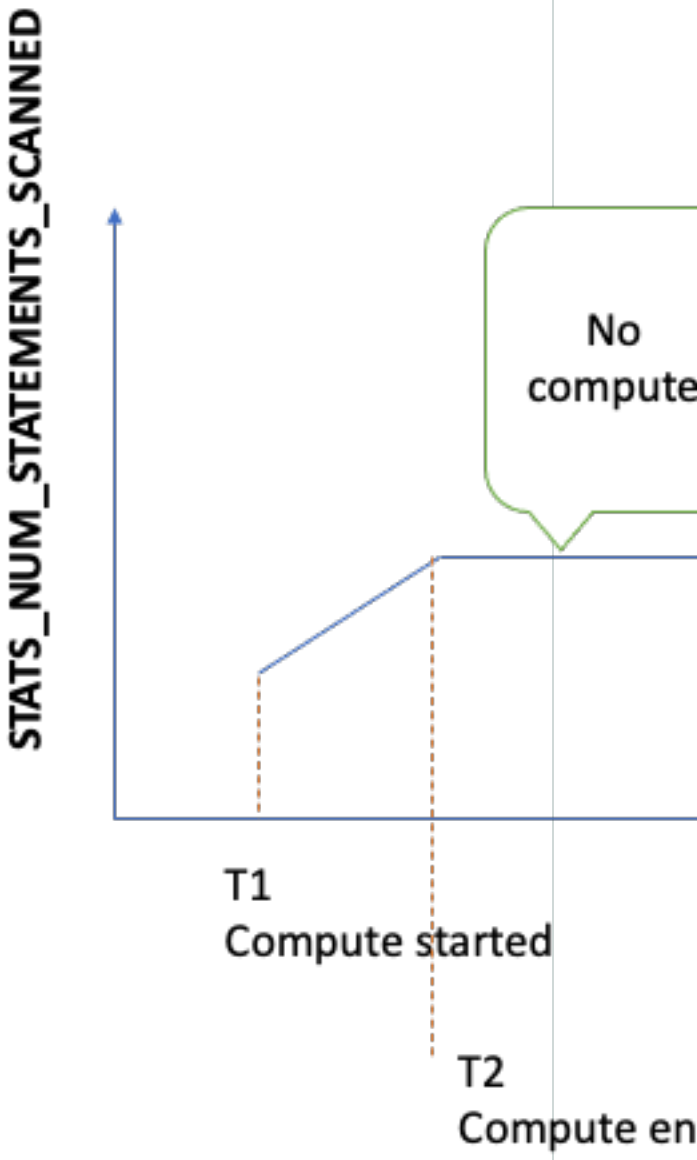
Note

每當伺服器重新啟動時，所有累積度量都會重設為零，無論是進行維護、重新開機或從當機復原。

海王星CloudWatch度量

指標	描述
BackupRetentionPeriodStorageUsed	Neptune DB 叢集的備份保留時段用來支援的備份儲存總量 (以位元組為單位)。包含在 TotalBackupStorageBilled 指標報告的總數中。
BufferCacheHitRatio	由緩衝區快取提供服務的請求的百分比。此測量結果在診斷查詢延遲時很有用，因為快取遺漏會造成顯著延遲。如果快取命中率低於 99.9，請考慮升級執行個體類型，以便在記憶體中快取更多資料。
ClusterReplicaLag	如果是僅供讀取複本，則為從主要執行個體複寫更新時的延遲量，以毫秒單位。
ClusterReplicaLagMaximum	主執行個體與資料庫叢集中每個 Neptune 資料庫執行個體之間的最大延遲量 (以毫秒為單位)。
ClusterReplicaLagMinimum	主要執行個體與資料庫叢集中每個 Neptune 資料庫執行個體之間的最小延遲量 (以毫秒為單位)。
CPUUtilization	CPU 使用率。
EngineUptime	執行個體已執行的時間量，以秒為單位。
FreeableMemory	可用的隨機存取記憶體的數量，以位元組為單位。
GlobalDbDataTransferBytes	Neptune 全域資料庫中，從主要資料庫傳輸AWS 區域到次AWS 區域要重做日誌資料的位元組數目。
GlobalDbReplicatedWriteIO	從全域資料庫中的主要AWS 區域資料庫複製到次要叢集磁碟區的寫入 I/O 作業數目AWS 區域。 Neptune 全域資料庫中每個資料庫叢集的計費計算會使用該VolumeWriteIOPS度量來說明在該叢集內執行的寫入。對於主要資料庫叢集，計費計算用於將跨區域複寫GlobalDbReplicatedWriteIO到次要資料庫叢集納入考量。
GlobalDbProgressLag	次要叢集在使用者交易和系統交易的主要叢集後面的毫秒數。
GremlinRequestsPerSec	每秒對 Gremlin 引擎的請求數量。
GremlinWebSocketOpenConnections	開啟至海王星的WebSocket連線數目。
LoaderRequestsPerSec	每秒載入器請求的數量。
MainRequestQueuePendingRequests	等待在輸入佇列中等待執行的要求數目。Neptune會在超過最大佇列容量時開始節流請求。

指標	描述
NCUUtilization	僅適用於 Neptune 無伺服器 (p. 577)資料庫執行個體或資料庫叢集。在執行個體層級，報告百分比計算為相關執行個體目前正在使用的 Neptune 容量單位 (NCU) 數目，除以叢集的最大 NCU 容量設定。一個 NCU 或海王星容量單位，由 2 GiB（千兆字節）的內存（RAM），以及相關的虛擬處理器容量（vCPU）和網絡組成。 在叢集層級，NCUUtilization報告叢集整體使用的最大容量百分比。
NetworkThroughput	Neptune DB 叢集中每個執行個體從用戶端接收和傳輸至用戶端的網路輸送量量，以每秒位元組為單位。此輸送量不包括資料庫叢集中執行個體與叢集磁碟區之間的網路流量。
NetworkTransmitThroughput	Neptune DB 叢集中每個執行個體傳輸至用戶端的外送網路輸送量量，以每秒位元組為單位。此輸送量不包括資料庫叢集中執行個體與叢集磁碟區之間的網路流量。
NumTxCommitted	每秒成功遞交的交易數量。
NumTxOpened	每秒在伺服器上開啟的交易數量。
NumTxRolledBack	針對寫入查詢，因為錯誤，每秒回復伺服器上的交易數目。如果是唯讀查詢，此測量結果等於每秒完成的唯讀交易數目。
OpenCypherRequestsPerSec	每秒 (HTTPS 和螺栓) 對開放密碼引擎的要求數目。
OpenCypherBoltOpenConnections	開啟螺栓連接到海王星的數量。
ServerlessDatabaseCapacity	作為執行個體層級指標，會在 NCU 中ServerlessDatabaseCapacity報告指定 Neptune 無伺服器 (p. 577)執行個體的目前執行個體容量。一個 NCU 或海王星容量單位，由 2 GiB（千兆字節）的內存（RAM），以及相關的虛擬處理器容量（vCPU）和網絡組成。 在叢集層級，ServerlessDatabaseCapacity報告叢集中資料庫執行個體所有ServerlessDatabaseCapacity值的平均值。
SnapshotStorageUsed	Neptune DB 叢集在其備份保留期間外的所有快照耗用的備份儲存體總量，以位元組為單位。包含在 TotalBackupStorageBilled 指標報告的總數中。
SparqlRequestsPerSec	每秒對 SPARQL 引擎的請求數量。

指標	描述
StatsNumStatementsScanned	自伺服器啟動以來針對 DFE 統計資料 (p. 131)掃描的陳述式總數。 每次觸發統計計算時，這個數字都會增加，但是當沒有計算發生時，它仍然是靜態的。因此，如果您隨著時間的推移繪製圖形，則可以判斷計算何時發生以及何時未發生：  透過查看量度增加的期間中的圖形斜率，您也可以判斷計算的速度。

指標	描述
	如果沒有此類指標，則表示您的資料庫叢集上的統計資料功能已停用，或者您正在執行的引擎版本沒有統計功能。如果測量結果值為零，表示沒有發生統計資料運算。
TotalBackupStorageBilled	針對指定的 Neptune 資料庫叢集向您收費的備份儲存體總量，以位元組為單位。包括以 BackupRetentionPeriodStorageUsed 和 SnapshotStorageUsed 指標衡量的備份儲存。
TotalRequestsPerSec	每秒來自所有來源對伺服器的請求總數。
TotalClientErrorsPerSec	每秒由於用戶端問題而傳回錯誤的請求總數。
TotalServerErrorsPerSec	伺服器上每秒由於內部失敗而傳回錯誤的請求總數。
UndoLogListSize	還原記錄清單中還原記錄的計數。 還原記錄包含已確認交易的記錄，這些記錄會在所有作用中交易比確認時間更近時到期。過期的記錄會定期清除。刪除作業的記錄可能需要比其他異動型態的記錄更長的時間。 清除僅由資料庫叢集的寫入器執行個體完成，因此清除速率取決於寫入器執行個體類型。如果資UndoLogListSize料庫叢集較高且不斷增加，請升級寫入器執行個體以提高清除率。 此外，如果您要從早於的版本升級到引擎版本1.2.0.0或更高版本1.2.0.0，請首先確保該UndoLogListSize值接近 0。由於引擎版本1.2.0.0及更新版本對還原記錄使用不同的格式，因此只有在完全清除先前的還原記錄之後才能開始升級。如需詳細資訊，請參閱 升級至 1.2.0.0 或更高版本 (p. 859)。
VolumeBytesUsed	配置給 Neptune 資料庫叢集的儲存空間總量，以位元組為單位。這是您收取費用的儲存空間量。它是分配給資料庫叢集群的最大存儲量 (在叢集存在時)，而不是您當前使用的數量 (請參閱 海王星儲存計費 (p. 119))。
VolumeReadIOPs	來自叢集磁碟區的計費讀取 I/O 操作平均數量，每隔 5 分鐘回報一次。計費的讀取作業是在叢集磁碟區層級計算，從 Neptune DB 叢集中的所有執行個體彙總，然後每隔 5 分鐘報告一次。
VolumeWriteIOPs	對叢集磁碟區執行寫入磁碟 I/O 操作的平均數量，每隔 5 分鐘回報一次。

CloudWatch海王星現在已棄用的量度

這些 Neptune 量度的使用現在已被取代。它們仍然受到支援，但在未來可能會被淘汰，因為有新的和更好的指標可用。

指標	描述
GremlinHttp1xx	每秒 Gremlin 端點的 HTTP 1xx 回應數量。 建議您改為使用新的 Http1xx 組合指標。
GremlinHttp2xx	每秒 Gremlin 端點的 HTTP 2xx 回應數量。 建議您改為使用新的 Http2xx 組合指標。
GremlinHttp4xx	每秒 Gremlin 端點的 HTTP 4xx 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
GremlinHttp5xx	每秒 Gremlin 端點的 HTTP 5xx 錯誤數量。 建議您改為使用新的 Http5xx 組合指標。
GremlinErrors	在 Gremlin 周遊中的錯誤數量。
GremlinRequests	對 Gremlin 引擎的請求數量。
GremlinWebSocketSuccess	每秒成功WebSocket連線到 Gemlin 端點的數目。
GremlinWebSocketClientErrors	每秒 G WebSocket rinlin 端點上的用戶端錯誤數目。
GremlinWebSocketServerErrors	每秒 Grinlin 端點上的WebSocket伺服器錯誤數目。
GremlinWebSocketAvailableConnections	目前可用的潛在WebSocket連線數目。
Http100	每秒端點的 HTTP 100 回應數量。 建議您改為使用新的 Http1xx 組合指標。
Http101	每秒端點的 HTTP 101 回應數量。 建議您改為使用新的 Http1xx 組合指標。
Http1xx	每秒端點的 HTTP 1xx 回應數量。
Http200	每秒端點的 HTTP 200 回應數量。 建議您改為使用新的 Http2xx 組合指標。
Http2xx	每秒端點的 HTTP 2xx 回應數量。
Http400	每秒端點的 HTTP 400 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
Http403	每秒端點的 HTTP 403 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
Http405	每秒端點的 HTTP 405 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
Http413	每秒端點的 HTTP 413 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。

指標	描述
Http429	每秒端點的 HTTP 429 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
Http4xx	每秒端點的 HTTP 4xx 錯誤數量。
Http500	每秒端點的 HTTP 500 錯誤數量。 建議您改為使用新的 Http5xx 組合指標。
Http501	每秒端點的 HTTP 501 錯誤數量。 建議您改為使用新的 Http5xx 組合指標。
Http5xx	每秒端點的 HTTP 5xx 錯誤數量。
LoaderErrors	來自載入器請求的錯誤數量。
LoaderRequests	載入器請求的數量。
SparqlHttp1xx	每秒 SPARQL 端點的 HTTP 1xx 回應數量。 建議您改為使用新的 Http1xx 組合指標。
SparqlHttp2xx	每秒 SPARQL 端點的 HTTP 2xx 回應數量。 建議您改為使用新的 Http2xx 組合指標。
SparqlHttp4xx	每秒 SPARQL 端點的 HTTP 4xx 錯誤數量。 建議您改為使用新的 Http4xx 組合指標。
SparqlHttp5xx	每秒 SPARQL 端點的 HTTP 5xx 錯誤數量。 建議您改為使用新的 Http5xx 組合指標。
SparqlErrors	SPARQL 查詢中的錯誤數量。
SparqlRequests	對 SPARQL 引擎的請求數量。
StatusErrors	來自狀態端點的錯誤數量。
StatusRequests	向狀態端點請求的數量。

海王星CloudWatch尺寸

Amazon Neptune 的指標是以帳戶、圖形名稱或作業的值來限定。您可以使用 Amazon CloudWatch 主控台擷取 Neptune 資料以及下表中的任何維度。

維度	描述
DBInstanceIdentifier	篩選您為叢集中的特定資料庫執行個體請求的資料。
DBClusterIdentifier	篩選您要求特定 Neptune 資料庫叢集的資料。

維度	描述
DBClusterIdentifier, EngineName	依照叢集篩選資料。所有 Neptune 執行個體的引擎名稱為 neptune。
DBClusterIdentifier, Role	篩選您要求特定 Neptune 資料庫叢集的資料，並依執行個體角色 (WRITER/READER) 彙總指標。例如，您可以為屬於某個叢集的所有讀取者執行個體彙總指標。
DBClusterIdentifier, SourceRegion	依全域資料庫主要區域中的主要叢集篩選資料。
DatabaseClass	篩選您為某個資料庫類別中的所有執行個體請求的資料。例如，您可以為屬於資料庫類別 db.r4.large 的所有執行個體彙總指標
EngineName	所有 Neptune 執行個體的引擎名稱為 neptune。
GlobalDbDBClusterIdentifier, SecondaryRegion	依次要區域中指定全域資料庫的次要叢集篩選資料。

搭配 Amazon 海王星叢集使用稽核日誌

若要稽核 Amazon Neptune 資料庫叢集活動，請設定資料庫叢集參數來啟用稽核日誌的收集。當稽核日誌啟用時，可用來記錄任意組合的支援事件。您可以檢視或下載稽核日誌來進行檢閱。

啟用海王星稽核記錄

使用 neptune_enable_audit_log 參數以啟用 (1) 或停用 (0) 稽核日誌。

在資料庫叢集所用的參數群組中設定此參數。您可以使用中顯示的程序編輯資料庫叢集參數群組或資料庫參數群組 (p. 528)來修改參數AWS Management Console，或使用 modify-db-cluster-parameter-group AWS CLI 指令或 ModifyDB ClusterParameterGroup API 指令以程式設計方式修改參數。

修改此參數後，您必須重新啟動資料庫執行個體，才能套用變更。

使用主控台檢視 Neptune 稽核記錄

您可以使用 AWS Management Console來檢視和下載稽核日誌。在 Instances (執行個體) 頁面上，選擇資料庫執行個體以顯示其詳細資訊，然後捲動至 Logs (日誌) 區段。

若要下載日誌檔案，請在 Logs (日誌) 區段中選取該檔案，然後選擇 Download (下載)。

海王星稽核記錄詳細

日誌檔案為 UTF-8 格式。日誌會寫入多個檔案中，數量隨著執行個體大小而不同。若要查看最新事件，您可能必須檢閱所有稽核日誌檔案。

日誌項目不會循序排列。您可以使用 timestamp 值來排序它們。

日誌檔案合計達到 100 MB 時會輪換。無法設定此限制。

稽核日誌檔案的資料列依指定順序包含以下逗號分隔的資訊：

欄位	描述
時間戳記	所記錄事件的 Unix 時間戳記 (精確度達到微秒)。
ClientHost	連線使用者來自的主機名稱或 IP。
ServerHost	記錄事件所針對之執行個體的主機名稱或 IP。
ConnectionType	連線類型。可以是`WebsocketHTTP_POST`、`HTTP_GET`、或`Bolt`。

來電者的 IAM ARN	用來簽署請求的 IAM 使用者或 IAM 角色的 ARN。如果停用 IAM 身分驗證會空白。格式為：
	`arn:`*`partition`*`:`*`service`*`:`*`region`*`:`*`account`*`:`*`resource`*
	例如：
	`arn:aws:iam::123456789012:user/Anna`
	`arn:aws:sts::123456789012:assumed-role/AWSNeptuneNotebookRole/SageMaker`

Auth Context	包含具備身分驗證資訊的序列化 JSON 物件。如果使用者已經驗證，則 `authenticationSucceeded` 欄位為 `True`。 如果停用 IAM 身分驗證會空白。
HttpHeader	HTTP 標頭資訊。可包含一個查詢。空白表示WebSocket和螺栓連接。
承載	小鬼，SPARQL，或開放密碼查詢。

將海王星日誌發佈到亞馬遜CloudWatch日

您可以設定 Neptune DB 叢集，將稽核記錄資料和/或緩慢查詢日誌資料發佈到 Amazon CloudWatch Logs 中的日誌群組。使用 CloudWatch Logs，您可以對記錄資料執行即時分析，並用CloudWatch來建立警示和檢視指標。您可以使用CloudWatch日誌將日誌記錄存儲在高度耐用的存儲中。

若要稽核發佈記錄至CloudWatch記錄，必須明確啟用稽核記錄 (請參閱啟用稽核日誌 (p. 781))。同樣地，若要將慢速查詢記錄檔發佈至記CloudWatch錄檔，必須明確啟用慢速查詢記錄檔 (請參閱)。使用亞馬遜海王星慢速查詢記錄 (p. 785)

> **Note**
>
> 請注意以下事項：
>
> - 將記錄發佈至時需支付額外費用CloudWatch。如需詳細資訊，請參閱CloudWatch定價頁面。
> - 您無法將記錄發佈至中國 (北京) 或中國 (寧夏) 區域的CloudWatch記錄檔。
> - 如果停用匯出記錄資料，Neptune 不會刪除現有的記錄群組或記錄串流。如果停用匯出記錄資料，CloudWatch記錄檔中會保留現有的記錄資料 (視記錄保留而定)，而且您仍會對儲存的稽核記錄資料產生費用。您可以使用CloudWatch記錄主控台、或記錄 API 刪除記錄串流和記CloudWatch錄群組。AWS CLI

使用主控台將 Neptune 記錄發佈至記CloudWatch錄檔

從主控台將 Neptune 記CloudWatch錄檔發佈至記錄檔

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇您要發佈記錄資料的 Neptune 資料庫叢集。
4. 在 Actions (動作) 中，選擇 Modify (修改)。
5. 在 [記錄匯出] 區段中，選擇您要開始發佈至CloudWatch記錄檔的記錄檔。
6. 選擇 Continue (繼續)，然後在摘要頁面上選擇 Modify DB Cluster (修改資料庫叢集)。

使用 CLI 將 Neptune 稽核記錄發佈至CloudWatch記錄

您可以使用具有下列參數的AWS CLI`create-db-cluster`命令，建立將稽核CloudWatch記錄發佈至記錄的新資料庫叢集：

```
aws neptune create-db-cluster \
    --region us-east-1  \
    --db-cluster-identifier my_db_cluster_id \
    --engine neptune \
    --enable-cloudwatch-logs-exports '["audit"]'
```

您可以使用具有下列參數的AWS CLI`modify-db-cluster`命令，將現有的資料庫叢集設定為將稽核CloudWatch記錄發佈至記錄：

```
aws neptune modify-db-cluster \
    --region us-east-1  \
    --db-cluster-identifier my_db_cluster_id \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

使用 CLI 將 Neptune 慢速查詢記錄檔發佈至記錄 CloudWatch

您也可以使用具有下列參數的AWS CLI`create-db-cluster`命令，建立新的資料庫叢集，將慢速查詢CloudWatch記錄檔發佈至記錄檔：

```
aws neptune create-db-cluster \
    --region us-east-1  \
    --db-cluster-identifier my_db_cluster_id \
    --engine neptune \
    --enable-cloudwatch-logs-exports '["slowquery"]'
```

同樣地，您可以使用具有下列參數的AWS CLI`modify-db-cluster`命令，將現有的資料庫叢集設定為將慢速查詢CloudWatch記錄檔發佈至記錄檔：

```
aws neptune modify-db-cluster --region us-east-1  \
    --db-cluster-identifier my_db_cluster_id \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["slowquery"]}'
```

在亞馬遜監控海王星日誌事件 CloudWatch

啟用 Neptune 日誌後，您可以在 Amazon CloudWatch 日誌中監控日誌事件。Neptune DB 叢集會在下列前置詞 (*cluster-name*代表資料庫叢集名稱) 下自動建立新的記錄群組，並*log_type*代表記錄檔類型：

```
/aws/neptune/cluster-name/log_type
```

例如，若您設定匯出功能來包含 mydbcluster 資料庫叢集的稽核日誌，則日誌資料會存放在 /aws/neptune/mydbcluster/audit 日誌群組中。

來自資料庫叢集的所有資料庫執行個體的所有事件，將會透過不同的日誌串流推送至日誌群組。

如果存在具有指定名稱的記錄群組，Neptune 會使用該記錄群組匯出 Neptune DB 叢集的記錄資料。您可以使用自動化組態 (例如 AWS CloudFormation)，以預先定義的日誌保留期間、指標篩選條件和客戶存取權來建立日誌群組。否則，系統會使用 [記錄檔] 中的預設記錄保留期間 [永不過期] 自動建立新的CloudWatch記錄群組。

您可以使用CloudWatch記錄主控台AWS CLI、或CloudWatch記錄 API 來變更記錄保留期間。如需變更記錄檔中記錄保留期間的詳細資訊，請參閱變更CloudWatch記錄檔中的CloudWatch記錄檔資料保留。

您可以使用CloudWatch記錄主控台AWS CLI、或CloudWatch記錄 API 來搜尋資料庫叢集的記錄事件中的資訊。如需搜尋和篩選日誌資料的詳細資訊，請參閱搜尋和篩選日誌資料。

為海王星筆記本啟用亞馬遜CloudWatch日誌

CloudWatchNeptune 筆記本的記錄預設為停用。請按照以下步驟啟用它們，用於調試或其他目的：

使用啟AWS Management Console用 Neptune 筆記本的記CloudWatch錄檔

1. 在以下位置打開亞馬遜SageMaker控制台 https://console.aws.amazon.com/sagemaker/。
2. 在左側的導覽窗格中，選擇記事本，然後選擇記事本執行個體。尋找您要啟用記錄檔的 Neptune 筆記本名稱。
3. 選擇上述步驟中提到的筆記本執行個體名稱，進入詳細資料頁面。
4. 如果記事本執行個體正在執行，請選取筆記本詳細資料頁面右上方的停止按鈕。
5. 在「權限和加密」下，存在 IAM 角色 ARN 欄位。選取此欄位中的連結以移至此筆記本的 IAM 角色。
6. 建立下列策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DeleteLogDelivery",
        "logs:Describe*",
        "logs:GetLogDelivery",
        "logs:GetLogEvents",
        "logs:ListLogDeliveries",
        "logs:PutLogEvents",
        "logs:PutResourcePolicy",
        "logs:UpdateLogDelivery"
      ],
      "Resource": "*"
    }
```

```
    ]
}
```

7. 儲存此新政策，並將其附加到步驟 4 中的 IAM 角色。
8. 選取SageMaker筆記本執行個體詳細資訊頁面右上角的 [啟動]。
9. 記錄檔開始流動後，您應該會在詳細資料頁面的 [筆記本執行個體設定] 區段左下方附近，在標示為 [生命週期組態] 欄位下方看到 [檢視記錄] 連結。

如果您的筆記型電腦無法啟動，SageMaker主控台的筆記型電腦詳細資料頁面會顯示訊息，指出筆記型電腦執行個體啟動時間超過 5 分鐘。您可以在以下名稱下找到與此問題相關的CloudWatch記錄：*(your-notebook-name)*/LifecycleConfigOnStart。

如有必要，請參閱使用亞馬遜記錄 Amazon SageMaker 事件以CloudWatch獲取更多詳細資訊

使用亞馬遜海王星慢速查詢記錄

識別、偵錯和最佳化執行緩慢的查詢可能很困難。啟用 Neptune 的慢速查詢記錄時，會自動記錄所有長時間執行查詢的屬性，讓這個程序更容易。

Note

在海王星引擎版本 (p. 861) 1.2.1.0 中引入了慢速查詢記錄。

您可以使用海王星啟用慢速查詢日誌記錄功能 _ 慢 _ 查詢日誌資料庫叢集 (p. 530)參數。依預設，此參數設定為`disable`。將其設定為`info`或`debug`啟用慢速查詢記錄。此`info`設定會記錄每個緩慢執行查詢的一些有用屬性，而`debug`設定會記錄所有可用的屬性。

若要設定被視為執行緩慢查詢的臨界值，請使用 neptune_slow_query_log_threshold (p. 531) DB 叢集參數來指定執行中查詢會被視為緩慢的毫秒數，並在啟用慢速查詢記錄時記錄。預設值為 5000 毫秒 (5 秒)。

您可以在 (p. 528)中設定這些資料庫叢集參數AWS Management Console，或使用 modify-db-cluster-parameter-group AWS CLI 命令或 ModifyDB ClusterParameterGroup (p. 1094) 管理功能。

Note

慢速查詢記錄參數是動態的，這表示變更其值不需要或導致資料庫叢集重新啟動。

若要檢視中的慢速查詢記錄檔 AWS Management Console

您可以在中檢視和下載慢速查詢記錄檔AWS Management Console，如下所示：

在 [執行個體] 頁面上，選擇資料庫執行個體，然後捲動至 [記錄] 區段。然後，您可以在那裡選擇一個日誌文件，然後選擇下載以下載它。

由海王星緩慢查詢日誌生成的文件

Neptune 中的慢速查詢記錄所產生的記錄檔具有下列特性：

- 這些文件被編碼為 UTF-8。
- 查詢及其屬性以 JSON 格式記錄。
- 空和空屬性不會記錄，除了`queryTime`數據。
- 記錄檔跨越多個檔案，其數目會隨執行個體大小而有所不同。
- 日誌項目不會循序排列。您可以使用它們的`timestamp`值來排序它們。
- 若要查看最新事件，您可能必須檢閱所有慢速查詢記錄檔。

- 當記錄檔總計達到 100 MiB 時，便會旋轉記錄檔。無法設定此限制。

在info模式中記錄的查詢屬性

neptune_enable_slow_query_log資料庫叢集參數設定為時，會記錄下列屬性以進行慢速查詢info：

群組	屬性	描述
requestResponseMetadata	requestId	查詢的請求 ID。
	requestType	請求類型，如 HTTP 或 WebSocket.
	responseStatusCode	查詢響應狀態碼，如 200。
	exceptionClass	查詢執行後返回錯誤的異常類。
查詢統計	query	查詢字串。
	queryFingerprint	查詢的指紋。
	queryLanguage	查詢語言，如小鬼，SPARQL 或開放密碼。
記憶體統計	allocatedPermits	分配給查詢的許可證。
	approximateUsedMemoryBytes	執行期間查詢使用的近似記憶體。
查詢時間	startTime	查詢開始時間 (UTC)。
	overallRunTimeMs	查詢總運行時間，以毫秒為單位。
	parsingTimeMs	查詢剖析時間，以毫秒為單位。
	waitingTimeMs	查詢格雷姆林/斯帕克爾/開密碼隊列等待時間，以毫秒為單位
	executionTimeMs	查詢執行時間，以毫秒為單位。
	serializationTimeMs	查詢序列化時間，以毫秒為單位。
陳述式計數器	scanned	掃描的陳述式數目。
	written	寫入的陳述數量。
	deleted	已刪除的陳述式數目。
交易計數器	committed	已完成的交易數目。
	rolledBack	倒回的交易數目。
頂點計數器	added	加入的頂點數目。
	removed	移除的頂點數目。
	propertiesAdded	加入的頂點性質數目。

群組	屬性	描述
	propertiesRemoved	移除的頂點性質數目。
教育計劃	added	加入的邊數。
	removed	移除的邊線數目。
	propertiesAdded	加入的邊屬性數目。
	propertiesRemoved	移除邊屬性的數目。
結果快取	hitCount	結果快取命中計數。
	missCount	結果快取未命中計數。
	putCount	結果快取放置計數。
並行建立	acceptedQueryCountAtStart	在開始時接受目前查詢執行的平行查詢。
	runningQueryCountAtStart	在啟動時執行目前查詢時執行的平行查詢。
	acceptedQueryCountAtEnd	在結束時接受與當前查詢執行的並行查詢。
	runningQueryCountAtEnd	並行查詢在結束時運行與當前查詢執行。
查詢批次	queryProcessingBatchSize	查詢處理期間的批次大小。
	querySerialisationBatchSize	查詢序列化期間的批次大小。

在debug模式中記錄的查詢屬性

將 neptune_enable_slow_query_log DB 叢集參數設定為時debug，除了以模式記錄的屬性之外，還會記錄下列儲存體計數器屬性：info

屬性	描述
statementsScannedInAllIndexes	在所有索引中掃描的陳述式。
statementsScannedSPOGIndex	在 SPOG 索引中掃描的陳述式。
statementsScannedPOGSIndex	在 POGS 索引中掃描的語句。
statementsScannedGPSOIndex	在 GPSO 索引中掃描的語句。
statementsScannedOSGPIndex	在 OSGP 索引中掃描的陳述式。
statementsScannedInChunk	以區塊形式一起掃描的陳述式。
postFilteredStatementScans	掃描後篩選後留下的陳述式。
distinctStatementScans	掃描不同的陳述式。
statementsReadInAllIndexes	在所有索引中掃描後篩選之後讀取的陳述式。

屬性	描述
statementsReadSPOGIndex	在 SPOG 索引中掃描後過濾後讀取的語句。
statementsReadPOGSIndex	在 POGS 索引中掃描後過濾後讀取的語句。
statementsReadGPSOIndex	在 GPSO 索引中掃描後過濾後讀取的語句。
statementsReadOSGPIndex	在 OSGP 索引中掃描後篩選後讀取的陳述式。
accessPathSearches	存取路徑搜尋的數目。
fullyBoundedAccessPathSearches	完全有界的密鑰訪問路徑搜索的數量。
accessPathSearchedByPrefix	按前綴搜索的訪問路徑數。
searchesWhereRecordsWereFound	輸出有 1 個或多個記錄的搜尋次數。
searchesWhereRecordsWereNotFound	沒有記錄作為輸出的搜尋數目。
totalRecordsFoundInSearches	從所有搜尋中找到的記錄總數。
statementsInsertedInAllIndexes	在所有索引中插入的陳述式數目。
statementsUpdatedInAllIndexes	在所有索引中更新的陳述式數目。
statementsDeletedInAllIndexes	在所有索引中刪除的陳述式數目。
predicateCount	謂詞的數目。
dictionaryReadsFromValueToIdTable	從值讀取到 ID 表的字典數量。
dictionaryReadsFromIdToValueTable	從值資料表 ID 讀取的字典數目。
dictionaryWritesToValueToIdTable	寫入值到 ID 表的字典數量。
dictionaryWritesToIdToValueTable	寫入值資料表 ID 的字典數目。
rangeCountsInAllIndexes	所有索引中的範圍計數數。
deadlockCount	查詢中的死結數目。
singleCardinalityInserts	執行的單一主教插入數目。
singleCardinalityInsertDeletions	單一基數插入期間刪除的陳述式數目。

慢速查詢的偵錯記錄範例

下列 Gemlin 查詢的執行時間可能會比為慢速查詢設定的臨界值更長的時間：

```
gremlin=g.V().has('code','AUS').repeat(out().simplePath()).until(has('code','AGR')).path().by('code').l
```

然後，如果在調試模式下啟用了慢速查詢日誌記錄，下面的屬性將被記錄在這樣的形式查詢：

```
{
  "requestResponseMetadata": {
    "requestId": "5311e493-0e98-457e-9131-d250a2ce1e12",
    "requestType": "HTTP_GET",
    "responseStatusCode": 200
  },
```

```
  "queryStats": {
    "query":
"gremlin=g.V().has('code','AUS').repeat(out().simplePath()).until(has('code','AGR')).path().by('code')
    "queryFingerprint":
"g.V().has(string0,string1).repeat(__.out().simplePath()).until(__.has(string0,string2)).path().by(str
    "queryLanguage": "Gremlin"
  },
  "memoryStats": {
    "allocatedPermits": 20,
    "approximateUsedMemoryBytes": 14838
  },
  "queryTimeStats": {
    "startTime": "23/02/2023 11:42:52.657",
    "overallRunTimeMs": 2249,
    "executionTimeMs": 2229,
    "serializationTimeMs": 13
  },
  "statementCounters": {
    "read": 69979
  },
  "transactionCounters": {
    "committed": 1
  },
  "concurrentExecutionStats": {
    "acceptedQueryCountAtStart": 1
  },
  "queryBatchStats": {
    "queryProcessingBatchSize": 1000,
    "querySerialisationBatchSize": 1000
  },
  "storageCounters": {
    "statementsScannedInAllIndexes": 69979,
    "statementsScannedSPOGIndex": 44936,
    "statementsScannedPOGSIndex": 4,
    "statementsScannedGPSOIndex": 25039,
    "statementsReadInAllIndexes": 68566,
    "statementsReadSPOGIndex": 43544,
    "statementsReadPOGSIndex": 2,
    "statementsReadGPSOIndex": 25020,
    "accessPathSearches": 27,
    "fullyBoundedAccessPathSearches": 27,
    "dictionaryReadsFromValueToIdTable": 10,
    "dictionaryReadsFromIdToValueTable": 17,
    "rangeCountsInAllIndexes": 4
  }
}
```


使用記錄亞馬遜海王星 API 呼叫 AWS CloudTrail

Amazon Neptune 整合了這項服務AWS CloudTrail，可提供 Neptune 中使用者、角色或服務所採取之動作記錄的AWS服務。CloudTrail擷取 Neptune 的 API 呼叫做為事件，包括來自 Neptune 主控台的呼叫，以及從對 Neptune API 的程式碼呼叫。

CloudTrail僅記錄 Neptune 管理 API 呼叫的事件，例如建立執行個體或叢集。若您希望稽核您圖表的變更，您可以使用稽核日誌。如需詳細資訊，請參閱搭配 Amazon 海王星叢集使用稽核日誌 (p. 781)。

> **Important**
>
> 亞馬遜海王星主控台和 API 呼叫會記錄為對亞馬遜關聯式資料庫服務 (Amazon RDS) API 的呼叫。AWS CLI

如果您建立追蹤，您可以啟用持續傳遞CloudTrail事件到 Amazon S3 儲存貯體，包括 Neptune 的事件。如果您不設定追蹤記錄，仍然可以透過 CloudTrail 主控台中的 Event history (事件歷史記錄) 檢視最新的事件。使用收集的資訊CloudTrail，您可以判斷向 Neptune 發出的要求、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 AWS CloudTrail 使用者指南。

海王星信息 CloudTrail

當您建立帳戶時，系統會在您的 AWS 帳戶中啟用 CloudTrail。當 Amazon Neptune 中發生活動時，該活動會與事件歷史記錄中的其他AWS服務CloudTrail事件一起記錄在事件中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷程記錄檢視事件。

如需AWS帳戶中持續的事件記錄 (包括 Neptune 的事件)，請建立追蹤。追蹤可CloudTrail將日誌檔交付到 Amazon S3 儲存貯體。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案

如果使用 Neptune 主控台、海王星命令列界面或海王星 SDK API 代表您的AWS帳戶採取動作，則會將該動作AWS CloudTrail記錄為對 Amazon RDS API 發出的呼叫。例如，如果您使用 Neptune 主控台修改資料庫執行個體或呼叫AWS CLImodify-db-instance命令，則AWS CloudTrail記錄會顯示對 Amazon RDS API 修改資料庫執行個體動作的呼叫。如需記錄的海王星 API 動作清單AWS CloudTrail，請參閱海王星 API 參考。

> **Note**
>
> AWS CloudTrail僅記錄 Neptune 管理 API 呼叫的事件，例如建立執行個體或叢集。若您希望稽核您圖表的變更，您可以使用稽核日誌。如需詳細資訊，請參閱搭配 Amazon 海王星叢集使用稽核日誌 (p. 781)。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail 使用者身分元素。

瞭解海王星記錄檔項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

下列範例顯示建立資料庫執行個體快照，然後使用 Neptune 主控台刪除該執行個體的使用者CloudTrail記錄。主控台由 `userAgent` 元素識別。透過主控台提出的請求 API 呼叫 (`CreateDBSnapshot`) 和 (`DeleteDBInstance`) 可在各記錄的 `eventName` 元素中找到。使用者 (`Alice`) 的相關資訊則可在 `userIdentity` 元素中找到。

```
{
```

```
  Records:[
  {
    "awsRegion":"us-west-2",
    "eventName":"CreateDBSnapshot",
    "eventSource":"",
    "eventTime":"2014-01-14T16:23:49Z",
    "eventVersion":"1.0",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS Console, aws-sdk-java\/unknown-version Linux\/2.6.18-kaos_fleet-1108-
prod.2 Java_HotSpot(TM)_64-Bit_Server_VM\/24.45-b08",
    "userIdentity":
    {
      "accessKeyId":"",
      "accountId":"123456789012",
      "arn":"arn:aws:iam::123456789012:user/Alice",
      "principalId":"AIDAI2JXM4FBZZEXAMPLE",
      "sessionContext":
      {
        "attributes":
        {
          "creationDate":"2014-01-14T15:55:59Z",
          "mfaAuthenticated":false
        }
      },
      "type":"IAMUser",
      "userName":"Alice"
    }
  },
  {
    "awsRegion":"us-west-2",
    "eventName":"DeleteDBInstance",
    "eventSource":"",
    "eventTime":"2014-01-14T16:28:27Z",
    "eventVersion":"1.0",
    "sourceIPAddress":"192.0.2.01",
    "userAgent":"AWS Console, aws-sdk-java\/unknown-version Linux\/2.6.18-kaos_fleet-1108-
prod.2 Java_HotSpot(TM)_64-Bit_Server_VM\/24.45-b08",
    "userIdentity":
    {
      "accessKeyId":"",
      "accountId":"123456789012",
      "arn":"arn:aws:iam::123456789012:user/Alice",
      "principalId":"AIDAI2JXM4FBZZEXAMPLE",
      "sessionContext":
      {
        "attributes":
        {
          "creationDate":"2014-01-14T15:55:59Z",
          "mfaAuthenticated":false
        }
      },
      "type":"IAMUser",
      "userName":"Alice"
    }
  }
  ]
}
```


使用海王星事件通知

主題

- 亞馬遜海王星事件類別和事件消息 (p. 792)

- 訂閱海王星事件通知 (p. 798)
- 管理海王星事件通知訂閱 (p. 799)

亞馬遜海王星使用亞馬遜簡單通知服務 (Amazon SNS) 在海王星事件發生時提供通知。這些通知可以是 Amazon SNS 在某個AWS區域支援的任何形式，例如電子郵件、文字訊息或呼叫 HTTP 端點。

Neptune 將這些事件分組到您可以訂閱的類別中，以便在該類別中的事件發生時收到通知。您可以訂閱資料庫執行個體、資料庫叢集、資料庫快照、資料庫叢集快照或資料庫參數群組的事件類別。例如，如果您訂閱指定資料庫執行個體的備份類別，當發生會影響資料庫執行個體的備份相關事件時，您將會收到通知。當事件通知訂閱變更時，您也會收到通知。

事件同時發生在資料庫叢集和資料庫執行個體層級，因此如果您訂閱資料庫叢集或資料庫執行個體，就可以接收事件。

事件通知會傳送至您在建立訂閱時提供的地址。您可能想要建立數個不同的訂閱，例如接收所有事件通知的訂閱，以及另一個僅包含生產資料庫執行個體重大事件的訂閱。您可以輕鬆關閉通知，而無需刪除訂閱。若要這麼做，請在 Neptune 主控台中將 [已啟用] 選項按鈕設定為 [否]。

> **Important**
>
> Amazon Neptune 不保證事件串流中傳送的事件順序。事件順序可能會改變。

海王星使用 Amazon SNS 主題的亞馬遜資源名稱 (ARN) 來識別每個訂閱。當您建立訂閱時，Neptune 主控台會為您建立 ARN。

海王星事件通知的計費方式是透過 Amazon SNS 進行。使用事件通知需要支付 Amazon SNS 費用。如需詳細資訊，請參閱 Amazon 簡易通知服務定價。

亞馬遜海王星事件類別和事件消息

Neptune 會在您可以使用 Neptune 主控台訂閱的類別中產生大量事件。每個類別都適用於來源類型，可以是資料庫執行個體、資料庫快照或資料庫參數群組。

> **Note**
>
> 海王星使用現有的 Amazon RDS 事件定義和 ID。

源自資料庫執行個體的海王星

當資料庫執行個體為來源類型時，下表依事件類別顯示事件清單。

類別	Amazon RDS 事件 ID	描述
可用性	RDS-EVENT-0006	資料庫執行個體已重新啟動。
可用性	RDS-EVENT-0004	資料庫執行個體關機。
可用性	RDS-EVENT-0022	重新啟動 Neptune 引擎時發生錯誤。
備份	RDS-EVENT-0001	備份資料庫執行個體。
備份	RDS-EVENT-0002	已完成資料庫執行個體備份。
設定更改	RDS-EVENT-0009	資料庫執行個體已新增至安全群組。

類別	Amazon RDS 事件 ID	描述
設定更改	RDS-EVENT-0024	資料庫執行個體正在轉換為多可用區域資料庫執行個體。
設定更改	RDS-EVENT-0030	資料庫執行個體正在轉換為單一可用區資料庫執行個體。
設定更改	RDS-EVENT-0012	套用修改至資料庫執行個體類別。
設定更改	RDS-EVENT-0018	此資料庫執行個體目前的儲存設定正在變更。
設定更改	RDS-EVENT-0011	此資料庫執行個體的參數群組已變更。
設定更改	RDS-EVENT-0092	此資料庫執行個體的參數群組已完成更新。
設定更改	RDS-EVENT-0028	此資料庫執行個體的自動備份已停用。
設定更改	RDS-EVENT-0032	此資料庫執行個體的自動備份已啟用。
設定更改	RDS-EVENT-0025	資料庫執行個體已轉換為多可用區域資料庫執行個體。
設定更改	RDS-EVENT-0029	資料庫執行個體已轉換為單一可用區資料庫執行個體。
設定更改	RDS-EVENT-0014	此資料庫執行個體的資料庫執行個體類別已變更。
設定更改	RDS-EVENT-0017	此資料庫執行個體的儲存設定已變更。
設定更改	RDS-EVENT-0010	資料庫執行個體已從安全群組移除。
建立	RDS-EVENT-0005	已建立資料庫執行個體。
刪除	RDS-EVENT-0003	資料庫執行個體已遭刪除。
容錯移轉	RDS-EVENT-0034	Neptune 未嘗試要求的容錯移轉，因為資料庫執行個體最近發生了容錯移轉。
容錯移轉	RDS-EVENT-0013	導致備用執行個體提升的多可用區域容錯移轉已啟動。

類別	Amazon RDS 事件 ID	描述
容錯移轉	RDS-EVENT-0015	導致備用執行個體提升的多可用區域容錯移轉已完成。DNS 傳輸到新的主資料庫執行個體可能需要幾分鐘的時間。
容錯移轉	RDS-EVENT-0065	執行個體已從部分容錯移轉中復原。
容錯移轉	RDS-EVENT-0049	多可用區域容錯移轉已完成。
容錯移轉	RDS-EVENT-0050	多可用區域啟用已在執行個體成功復原之後開始。
容錯移轉	RDS-EVENT-0051	多可用區域啟用已完成。現在應可存取您的資料庫。
失敗	RDS-EVENT-0031	因為不相容的組態或基礎儲存問題，資料庫執行個體已失敗。開始資point-in-time-restore料庫執行個體。
失敗	RDS-EVENT-0036	資料庫執行個體位於不相容的網路中。部分指定的子網路 ID 無效或不存在。
失敗	RDS-EVENT-0035	資料庫執行個體有無效的參數。例如，如果此執行個體類別與記憶體相關的參數設定過高造成資料庫叢集無法啟動，則客戶動作將為修改記憶體參數並重新啟動資料庫執行個體。
失敗	RDS-EVENT-0082	海王星無法從 Amazon S3 儲存貯體複製備份資料。海王星存取 Amazon S3 儲存貯體的許可可可可可可可可可設定不正確。
低儲存	RDS-EVENT-0089	資料庫執行個體已消耗超過其分配儲存容量的 90%。您可以使用 Free Storage Space (可用儲存空間) 指標來監控資料庫執行個體的儲存空間。

類別	Amazon RDS 事件 ID	描述
低儲存	RDS-EVENT-0007	資料庫執行個體的分配儲存空間已耗盡。若要解決此問題，您應為此資料庫執行個體分配額外儲存空間。
維護	RDS-EVENT-0026	資料庫執行個體的離線維護正在進行。資料庫執行個體目前無法使用。
維護	RDS-EVENT-0027	資料庫執行個體的離線維護已完成。資料庫執行個體現在可用。
維護	RDS-EVENT-0047	資料庫執行個體的修補作業已完成。
則通知	RDS-EVENT-0044	操作者發出的通知。如需更多詳細資訊，請參閱事件訊息。
則通知	RDS-EVENT-0048	資料庫執行個體的修補作業已延遲。
則通知	RDS-EVENT-0087	資料庫執行個體已停止。
則通知	RDS-EVENT-0088	資料庫執行個體已啟動。
則通知	RDS-EVENT-0154	資料庫執行個體由於超過允許停止的時間上限，正在啟動。
則通知	RDS-EVENT-0158	資料庫執行個體處於無法升級的狀態。
則通知	RDS-EVENT-0173	資料庫執行個體已修補。
僅供讀取複本	RDS-EVENT-0045	僅供讀取複寫程序發生錯誤。如需更多詳細資訊，請參閱事件訊息。
僅供讀取複本	RDS-EVENT-0046	僅供讀取複本已恢復複寫。此訊息會在您首次建立僅供讀取複本時出現，或做為確認複寫正常運作的監控訊息顯示。如果此訊息出現在 RDS-EVENT-0045 通知之後，則在錯誤或複寫停止之後已恢復複寫。
僅供讀取複本	RDS-EVENT-0057	僅供讀取複本上的複寫已終止。

類別	Amazon RDS 事件 ID	描述
僅供讀取複本	RDS-EVENT-0062	僅供讀取複本上的複寫已手動停止。
僅供讀取複本	RDS-EVENT-0063	僅供讀取複本上的複寫已重設。
復原	RDS-EVENT-0020	資料庫執行個體的復原已啟動。復原時間依據恢復的資料量而有不同。
復原	RDS-EVENT-0021	資料庫執行個體的復原已完成。
復原	RDS-EVENT-0023	已要求手動備份，但 Neptune 目前正在建立資料庫快照集。Neptune 完成資料庫快照集之後，再次提交要求。
復原	RDS-EVENT-0052	多可用區域執行個體的復原已啟動。復原時間依據恢復的資料量而有不同。
復原	RDS-EVENT-0053	多可用區域執行個體的復原已完成。
還原	RDS-EVENT-0008	資料庫執行個體已從資料庫快照還原。
還原	RDS-EVENT-0019	資料庫執行個體已從 point-in-time備份還原。

源自資料庫叢集的海王星事件

下表顯示了當數據庫集群是源類型的事件類別的事件列表。

類別	RDS 事件 ID	描述
容錯移轉	RDS-EVENT-0069	資料庫叢集的容錯移轉已失敗。
容錯移轉	RDS-EVENT-0070	資料庫叢集的容錯移轉已重新啟動。
容錯移轉	RDS-EVENT-0071	資料庫叢集的容錯移轉已結束。
容錯移轉	RDS-EVENT-0072	資料庫叢集的容錯移轉已在相同的可用區域中開始。
容錯移轉	RDS-EVENT-0073	資料庫叢集的容錯移轉已跨越可用區域開始。

類別	RDS 事件 ID	描述
失敗	RDS-EVENT-0083	海王星無法從 Amazon S3 儲存貯體複製備份資料。海王星存取 Amazon S3 儲存貯體的許可可可可可可可可可設定不正確。
維護	RDS-EVENT-0156	資料庫叢集有可用的資料庫引擎次要版本升級。
則通知	RDS-EVENT-0076	移轉至海王星資料庫叢集失敗。
則通知	RDS-EVENT-0077	移轉至 Neptune DB 叢集期間，嘗試將資料表從來源資料庫轉換為資料庫表單失敗。
則通知	RDS-EVENT-0150	資料庫叢集已停止。
則通知	RDS-EVENT-0151	資料庫叢集已啟動。
則通知	RDS-EVENT-0152	資料庫叢集停止失敗。
則通知	RDS-EVENT-0153	資料庫叢集由於超過允許停止的時間上限，正在啟動。

源自資料庫叢集快照的海王星事件

下表顯示當 Neptune DB 叢集快照集為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	描述
備份	RDS-EVENT-0074	已開始建立手動資料庫叢集快照。
備份	RDS-EVENT-0075	手動資料庫叢集快照已建立。
則通知	RDS-EVENT-0162	資料庫叢集快照匯出工作失敗。
則通知	RDS-EVENT-0163	已取消資料庫叢集快照匯出工作。
則通知	RDS-EVENT-0164	已完成資料庫叢集快照匯出工作。
備份	RDS-EVENT-0168	建立自動叢集快照。
備份	RDS-EVENT-0169	已建立自動叢集快照。
建立	RDS-EVENT-0170	已建立資料庫叢集。

類別	RDS 事件 ID	描述
刪除	RDS-EVENT-0171	資料庫叢集已刪除。
則通知	RDS-EVENT-0172	已將資料庫叢集從 [舊的資料庫叢集名稱] 重新命名為 [新的資料庫叢集名稱]。

源自資料庫叢集參數群組的海王星事件

下表顯示資料庫叢集參數群組為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	描述
設定更改	RDS-EVENT-0037	參數群組已修改。

源自安全性群組的 Neptune 事件

下表顯示安全性群組為來源類型時的事件類別和事件清單。

類別	RDS 事件 ID	描述
設定更改	RDS-EVENT-0038	安全群組已修改。
失敗	RDS-EVENT-0039	由 [使用者] 擁有的安全群組不存在，安全群組的授權已撤回。

訂閱海王星事件通知

您可以使用 Neptune 主控台來訂閱事件通知，如下所示：

訂閱 Neptune 事件通知

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。
3. 在 Event subscriptions (事件訂閱) 窗格中，選擇 Create event subscription (建立事件訂閱)。
4. 在 Create event subscription (建立事件訂閱) 對話方塊中，執行下列動作：
   a. 在 Name (名稱) 中，輸入事件通知訂閱的名稱。
   b. 在 Send notifications to (傳送通知至) 中，選擇 Amazon SNS 主題的現有 Amazon SNS ARN，或選擇 create topic (建立主題) 以輸入主題名稱及收件人清單。
   c. 在 Source type (來源類型) 中選擇來源類型。
   d. 選擇 Yes (是) 以啟用訂閱。如果您要建立訂閱，但還不要傳送通知，選擇 No (否)。
   e. 根據您選取的來源類型，選擇事件類別以及您要從中接收事件通知的來源。
   f. 選擇 建立 。

管理海王星事件通知訂閱

如果您在 Neptune 主控台的導覽窗格中選擇事件訂閱，則可以檢視訂閱類別和目前訂閱的清單。

您也可以修改或刪除特定訂閱。

修改海王星事件通知訂閱

修改您目前的 Neptune 事件通知訂閱

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。Event subscriptions (事件訂閱) 窗格顯示所有事件通知訂閱。
3. 在 Event subscriptions (事件訂閱) 窗格中，選擇您要修改的訂閱，然後選擇 Edit (編輯)。
4. 在 Target (目標) 或 Source (來源) 區段中，對訂閱進行變更。您可以在「來源」區段中選取或取消選取來源識別碼，來新增或移除來源識別碼。
5. 選擇 編輯 。Neptune 主控台會指出訂閱正在修改中。

刪除海王星事件通知訂閱

您可以刪除不再需要的訂閱。該主題的所有訂閱者將不會再收到該訂閱指定的事件通知。

若要刪除 Neptune 事件通知訂閱

1. 登入AWS管理主控台，然後在 https://console.aws.amazon.com/neptune/home 開啟亞馬遜海王星主控台。
2. 在導覽窗格中，選擇 Event subscriptions (事件訂閱)。
3. 在 [事件訂閱] 窗格中，選擇您要刪除的訂閱。
4. 選擇 刪除 。
5. Neptune 主控台會指出正在刪除訂閱。

標記亞馬遜海王星資

您可以使用 Neptune 標籤將中繼資料新增至您的 Neptune 資源。此外，您可以搭配 AWS Identity and Access Management (IAM) 政策使用標籤來管理 Neptune 資源的存取，並控制可套用至這些資源的動作。最後，您可以使用標籤將類似的標記資源分類為群組以追蹤成本。

所有 Neptune 管理資源都可加上標籤，包括下列項目：

- 資料庫執行個體
- 資料庫叢集
- 僅供讀取複本
- 資料庫快照
- 資料庫叢集快照
- 事件訂閱
- 資料庫參數群組
- 資料庫叢集參數群組

- 資料庫子網路群組

海王星資源標籤概述

Amazon Neptune 標籤是您定義並與 Neptune 資源建立關聯的名稱-值配對。此名稱叫做金鑰。為金鑰提供值是選用的。您可以使用標籤將任意資訊指派給 Neptune 資源。例如，您可以使用標籤金鑰來定義類別，此標籤值可以是該類別中的項目。例如，您可以定義「項目」的標籤鍵和標籤值「Salix」，表示海王星資源已分配給 Salix 項目。您也可以使用標籤，透過使用金鑰 (例如或)，將 Neptune 資源指定為用於測試`environment=test`或生產環境`environment=production`。建議您使用一組一致的標籤金鑰，以便更輕鬆地追蹤與 Neptune 資源相關聯的中繼資料。

使用標籤來整理您的 AWS 帳單，以反映您自身的成本結構。因此，請進行註冊以取得包含標籤鍵值的 AWS 帳戶 帳單。接著，若要查看合併資源的成本，請根據具有相同標籤鍵值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。如需詳細資訊，請參閱《AWS Billing》使用者指南中的使用成本分配標籤。

每個 Neptune 資源都有一個標籤集，其中包含指派給該 Neptune 資源的所有標籤。標籤集最多可以包含 10 個標籤，也可以是空的。如果您將標籤新增至與資源上現有標籤具有相同索引鍵的 Neptune 資源，則新值會覆寫舊值。

AWS 不會將任何語意套用至您的標籤；標籤將嚴格解釋為字元字串。Neptune 可以根據您在建立資源時使用的設定，在資料庫執行個體或其他 Neptune 資源上設定標籤。例如，Neptune 可能會新增標籤，指出資料庫執行個體用於生產或測試。

- 標籤金鑰是標籤必要的名稱。字串值長度可以是 1 到 128 個 Unicode 字元，不可在前面加上「`aws:`」或「`rds:`」。此字串只能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「–」(Java regex："^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$")。
- 標籤值即為選用的標籤字串值，字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上「`aws:`」。此字串只能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「–」(Java regex："^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$")。

  標籤組中的值不必是唯一的，並且可以是 null。例如，您可以在 `project/Trinity` 和 `cost-center/Trinity` 標籤集中有一個鍵值對。

  Note

  您可以將標籤新增到快照。不過，您的帳單不會反映這個分組。

您可以使用AWS Management ConsoleAWS CLI、或 Neptune API 來新增、列出和刪除 Neptune 資源上的標籤。使用AWS CLI或海王星 API 時，您必須為要使用的海王星資源提供 Amazon 資源名稱 (ARN)。如需建構 ARN 的詳細資訊，請參閱 為海王星構建 ARN (p. 803)。

標籤是快取用於授權之用。因此，Neptune 資源上標籤的新增和更新可能需要幾分鐘的時間才能使用。

複製海王星中的標籤

當您建立或還原資料庫執行個體時，您可以指定標籤從資料庫執行個體複製到資料庫執行個體的快照。複製標籤可確保資料庫快照的中繼資料符合來源資料庫執行個體的中繼資料，以及資料庫快照的任何存取政策也符合來源資料庫執行個體的存取政策。根據預設，不會複製標籤。

您可以指定標籤複製到資料庫快照以用於下列動作：

- 建立資料庫執行個體。
- 還原資料庫執行個體。

- 建立僅供讀取複本。
- 複製資料庫快照。

Note

如果您為create-db-cluster-snapshotAWS CLI命令的`--tag-key`參數包含值 (或為 創建數據庫ClusterSnapshot (p. 1110) API 動作提供至少一個標籤)，Neptune 不會將標籤從來源資料庫執行個體複製到新的資料庫快照集。即使來源資料庫執行個體啟用 `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`) 選項，也是如此。
這表示您可以從資料庫快照建立資料庫執行個體的複本，避免新增不適用於新資料庫執行個體的標籤。使用AWS CLI`create-db-cluster-snapshot`命令 (或 `CreateDBClusterSnapshot` Neptune API 動作) 建立資料庫快照集之後，您可以按照本主題稍後的說明新增標籤。

在海王星中使用標記 AWS Management Console

標記 Amazon Neptune 資源的程序與所有資源類似。下列程序顯示如何標記 Neptune 資料庫執行個體。

新增標籤至資料庫資訊個體

1. 登入AWS管理主控台，然後在 https://console.aws.amazon.com/neptune/home 開啟亞馬遜海王星主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。

   Note

   若要過濾 Instances (執行個體) 窗格中的資料庫執行個體清單，請在 Filter instances (過濾執行個體) 方塊中輸入文字字串。只有包含此字串的資料庫執行個體會顯示出來。
3. 選擇您要設定標籤的資料庫執行個體。
4. 選擇 Instance actions (執行個體動作)，然後選擇 See details (請參閱詳細資訊)。
5. 在詳細內容區段，向下捲動至 Tags (標籤) 區段。
6. 選擇 Add (新增)。此時將會顯示 Add tags (新增標籤) 視窗。
7. 在 Tag key (標籤金鑰) 和 Value (值) 中輸入值。
8. 若要新增另一個標籤，您可以選擇 Add another Tag (新增另一個標籤)，然後在它的 Tag key (標籤金鑰) 和 Value (值) 中輸入值。

   視需要重複此步驟。
9. 選擇 Add (新增)。

從資料庫資訊個體刪除標籤

1. 登入AWS管理主控台，然後在 https://console.aws.amazon.com/neptune/home 開啟亞馬遜海王星主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。

   Note

   若要過濾 Instances (執行個體) 窗格中的資料庫執行個體清單，請在 Filter instances (過濾執行個體) 方塊中輸入文字字串。只有包含此字串的資料庫執行個體會顯示出來。
3. 選擇您要設定標籤的資料庫執行個體。
4. 選擇 Instance actions (執行個體動作)，然後選擇 See details (請參閱詳細資訊)。
5. 在詳細內容區段，向下捲動至 Tags (標籤) 區段。
6. 選擇您要刪除的標籤。

7. 選擇 Remove (移除)，然後選擇 Remove tags (移除標籤) 視窗中的 Remove (移除)。

在海王星中使用標記 AWS CLI

您可以使用 Neptune 中的資料庫執行個體新增、列出或移除標籤AWS CLI。

- 若要將一或多個標籤新增至 Neptune 資源，請使用AWS CLI命令`add-tags-to-resource`。
- 若要列出 Neptune 資源上的標籤，請使用AWS CLI命令`list-tags-for-resource`。
- 若要從 Neptune 資源中移除一或多個標籤，請使用AWS CLI命令`remove-tags-from-resource`。

若要進一步了解如何建構必要的 Amazon Resource Name (ARN)，請參閱為海王星構建 ARN (p. 803)。

使用 API 在海王星中標記

您可以使用 Neptune API 新增、列出或移除資料庫執行個體的標籤。

- 若要將標籤新增至 Neptune 資源，請使用此`AddTagsToResource`作業。
- 若要列出指派給 Neptune 資源的標籤，請使用 `ListTagsForResource`.
- 若要從 Neptune 資源移除標籤，請使用此`RemoveTagsFromResource`作業。

若要進一步了解如何建構必要的 ARN，請參閱為海王星構建 ARN (p. 803)。

使用海王星 API 使用 XML 時，標籤會使用下列結構描述：

```
<Tagging>
    <TagSet>
        <Tag>
            <Key>Project</Key>
            <Value>Trinity</Value>
        </Tag>
        <Tag>
            <Key>User</Key>
            <Value>Jones</Value>
        </Tag>
    </TagSet>
</Tagging>
```

下表列出允許的 XML 標籤及其特性。`Key` 和 `Value` 的值區分大小寫。例如，`project=Trinity` 和 `PROJECT=Trinity` 是兩個不同的標籤。

標籤元素	描述
`TagSet`	標籤組是指派給 Neptune 資源之所有標籤的容器。每個資源只能有一個標籤集。您`TagSet`只能透過海王星 API 使用。
`Tag`	標籤是使用者定義的鍵值組。標籤集內可以有 1 到 50 個標籤。
`Key`	鍵是標籤的必要名稱。字串值長度可以是 1 到 128 個 Unicode 字元，不可在前面加上「`rds:`」或「`aws:`」。此字串只能包含一組 Unicode 字母、數字、空格、「`_`」、「`.`」、「`/`」、「`=`」、「`+`」、「`-`」(Java regex：`"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`)。 鍵在標籤集內必須是唯一的。例如，標籤集內的鍵組不能有相同的鍵但使用不同的值，例如 `project/Trinity` 和 `project/Xanadu`。

標籤元素	描述
值	值是標籤的選用值。字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上「rds:」或「aws:」。此字串只能包含一組 Unicode 字母、數字、空格、「_」、「.」、「/」、「=」、「+」、「-」(Java regex："^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$")。 標籤集內的值不必是唯一的，並且可以是 null。例如，您可以在 project/Trinity 和 cost-center/Trinity 標籤集中有一個鍵值對。

使用亞馬遜海王星的管理 ARN

在亞馬遜網絡服務中創建的資源，每個資源都使用亞馬遜資源名稱（ARN）進行唯一標識。對於某些 Amazon Neptune 作業，您必須透過指定海王星資源的 ARN 來唯一識別海王星資源。

> **Important**
>
> 亞馬遜海王星共用亞馬遜 RDS ARN 的格式，用於使用Neptune 管理 API (p. 996). 海王星系統管理 ARN 包含rds且不neptune-db包含。如需識別 Neptune 資料資源的資料平面 ARN，請參閱指定資料 (p. 209)資源。

主題

- 為海王星構建 ARN (p. 803)
- 在亞馬遜海王星中獲取現有的 ARN (p. 804)

為海王星構建 ARN

您可以使用下列語法為 Amazon 海王星資源建構 ARN。請注意，海王星共用亞馬遜 RDS ARN 的格式。

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

下表顯示針對特定 Neptune 系統管理資源類型建構 ARN 時應使用的格式。

資源類型	ARN 格式
資料庫執行個體	<region>ARN: aws: RDS:: *:db:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:db:my-instance-1`
資料庫叢集	<region>ARN: aws: RDS:: *:cluster:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:cluster:my-cluster-1`
事件訂閱	<region>ARN: aws: RDS:: *:es:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:es:my-subscription`

資源類型	ARN 格式
DB parameter group (資料庫參數群組)	<region>ARN: aws: RDS:: *:pg:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:pg:my-param-enable-logs`
DB cluster parameter group (資料庫叢集參數群組)	<region>ARN: aws: RDS:: *:cluster-pg:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:cluster-pg:my-cluster-param-timezone`
資料庫叢集快照	<region>ARN: AWS:: `rds:` *:cluster-snapshot:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:cluster-snapshot:my-snap-20160809`
資料庫子網路群組	<region>ARN: AWS:: `rds:` *:subgrp:*<account><name> 例如： `arn:aws:rds:us-east-2:123456789012:subgrp:my-subnet-10`

在亞馬遜海王星中獲取現有的 ARN

您可以使用AWS Management Console、AWS Command Line Interface (AWS CLI) 或海王星 API 來取得海王星資源的 ARN。

使用 AWS Management Console 取得現有的 ARN

若要使用主控台取得 ARN，請瀏覽至您需要 ARN 的資源，然後檢視該資源的詳細資訊。例如，若要取得資料庫執行個體的 ARN，請選擇導覽面板中的 Instances (執行個體)，然後從清單中選擇所要的執行個體。ARN 位在 Instance Details (執行個體詳細資訊) 部分。

使用 AWS CLI 取得現有的 ARN

若要使用取AWS CLI得特定 Neptune 資源的 ARN，請使用該資源的`describe`命令。下表顯示每個 AWS CLI 命令以及搭配命令來取得 ARN 的 ARN 屬性。

AWS CLI 命令	ARN 屬性
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	資料庫 ParameterGroupArn
describe-db-cluster-parameter-組	資料庫 ClusterParameterGroupArn

AWS CLI 命令	ARN 屬性
describe-db-instances	資料庫 InstanceArn
describe-events	SourceArn
describe-db-subnet-groups	資料庫 SubnetGroupArn
describe-db-clusters	資料庫 ClusterArn
describe-db-cluster-snapshots	資料庫 ClusterSnapshotArn

例如，下列 AWS CLI 命令可取得資料庫執行個體的 ARN。

Example

對於 Linux、OS X 或 Unix：

```
aws neptune describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2
```

針對 Windows：

```
aws neptune describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2
```

使用 API 取得現有 ARN

若要取得特定 Neptune 資源的 ARN，請呼叫下列 API 動作，並使用顯示的 ARN 屬性。

Neptune API 動作	ARN 屬性
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
描述 B ParameterGroups	資料庫 ParameterGroupArn
描述 B ClusterParameterGroups	資料庫 ClusterParameterGroupArn
DescribeDBInstances	資料庫 InstanceArn
DescribeEvents	SourceArn
描述 B SubnetGroups	資料庫 SubnetGroupArn
DescribeDBClusters	資料庫 ClusterArn
描述 B ClusterSnapshots	資料庫 ClusterSnapshotArn

備份與還原 Amazon Neptune 資料庫叢集

本節會說明如何備份與還原 Amazon Neptune 資料庫叢集。

主題

- 備份與還原 Neptune 資料庫叢集的概觀 (p. 807)
- 在 Neptune 中建立資料庫叢集快照 (p. 810)
- 從資料庫叢集快照還原 (p. 811)
- 複製資料庫叢集快照 (p. 813)
- 共享資料庫叢集快照 (p. 817)
- 刪除 Neptune 快照 (p. 821)

備份與還原 Neptune 資料庫叢集的概觀

本節提供在 Amazon Neptune 中備份與還原 Amazon Neptune 資料的頂層資訊。

主題

- Neptune 資料庫叢集的容錯能力 (p. 807)
- Neptune 備份 (p. 807)
- CloudWatch 有助於管理 Neptune 備份儲存體的指標 (p. 808)
- 從 Neptune 備份還原資料 (p. 808)
- Neptune 中的 Backup 窗口 (p. 809)

Neptune 資料庫叢集的容錯能力

Neptune 資料庫叢集特意設計為具備容錯能力。叢集磁碟區跨單一 AWS 區域中的多個可用區域，每個可用區域包含叢集磁碟區資料的副本。此功能意味著資料庫叢集可承受可用區域故障，完全不會遺失資料，服務只會短暫中斷。

如果資料庫叢集中的主要執行個體失敗，Neptune 有兩種方式自動容錯移轉至新的主要執行個體：

- 將現有的 Neptune 複本提升至新的主要執行個體
- 建立新的主要執行個體

如果資料庫叢集有一或多個 Neptune 複本，則在失敗事件期間會將 Neptune 複本提升為主要執行個體。失敗事件會導致短暫中斷，在此期間，讀取和寫入操作會失敗，並引發例外狀況。不過，服務通常會在 120 秒之內恢復，往往不超過 60 秒。若要提高資料庫叢集的可用性，建議在兩個以上不同的可用區域建立至少一或多個 Neptune 複本。

您可以指派每個 Neptune 複本的優先順序，以自訂複本在失敗之後提升為主要執行個體的順序。優先順序從 0 (代表最高優先順序) 到 15 (代表最低優先順序)。如果主要執行個體失敗，Neptune 會將優先順序最高的 Neptune 複本提升為新的主要執行個體。您隨時都可修改 Neptune 複本的優先順序。修改優先順序不會觸發容錯移轉。

您可以使用設AWS CLI定資料庫執行個體的容錯移轉優先順序，如下所示：

```
aws neptune modify-db-instance --db-instance-identifier (the instance ID) --promotion-
tier (the failover priority value)
```

多個 Neptune 複本可以共用相同的優先順序，形成提升層。如果兩個以上的 Neptune 複本擁有相同的優先順序，Neptune 會提升最大的複本。如果兩個以上的 Neptune 複本共用相同的優先順序和大小，Neptune 會提升相同提升層中的任意複本。

如果資料庫叢集未包括任何 Neptune 複本，則會在失敗事件期間重建主要執行個體。失敗事件會導致中斷，在此期間，讀取和寫入操作會失敗，並引發例外狀況。建立新的主要執行個體後，服務就會恢復，通常不超過 10 分鐘。將 Neptune 複本提升為主要執行個體比建立新的主要執行個體快得多。

Neptune 備份

Neptune 會自動備份您的叢集磁碟區，並在備份保留期保留還原資料。Neptune 備份具有連續性和增量性，因此，您可以快速還原到備份保留期內的任何時間點。寫入備份資料時不會影響資料庫服務的效能或中斷服務。當您建立或修改資料庫叢集時，您可以指定 1 到 35 天的備份保留期。

若要控制您的備份儲存用量，您可以降低備份保留間隔、移除不再需要的舊手動快照，或同時進行兩者。若要管理您的成本，您可以監控連續備份和超出保留期的手動快照所消耗的儲存量。您可以降低備份保留間隔，並移除不再需要的手動快照。

如果希望備份可以保留超過備份保留期，您也可以在叢集磁碟區中建立資料快照。儲存快照會產生 Neptune 的標準儲存費用。如需 Neptune 儲存體定價的詳細資訊，請參閱 Amazon Neptune 定價。

Neptune 在整個備份保留期內保留遞增還原資料。因此，您只需要建立希望在備份保留期後仍能保留的資料快照。您可以從快照建立新的資料庫叢集。

Important

如果刪除資料庫叢集，其所有自動備份都會同時刪除，且無法復原。這表示除非您選擇手動建立最終資料庫快照，否則之後無法將資料庫執行個體還原為最終狀態。刪除叢集時，系統並不會刪除手動快照。

Note

- 無論資料庫叢集的建立方式為何，Amazon Neptune 資料庫叢集的預設備份保留期為一天。
- 您無法停用 Neptune 上的自動備份。Neptune 的備份保留期是由資料庫叢集管理。

CloudWatch 有助於管理 Neptune 備份儲存體的指標

您可以使用 Amazon CloudWatch 指標`TotalBackupStorageBilled`、`SnapshotStorageUsed`、和`BackupRetentionPeriodStorageUsed`來檢閱和監控 Neptune 備份所使用的儲存用量，如下所示：

- `BackupRetentionPeriodStorageUsed` 表示目前用於儲存連續備份的備份儲存量 (以位元組為單位)。此值取決於叢集磁碟區的大小，以及您在保留期中所做的變更量。不過，為了方便計費，該值不會超過保留期內累積的叢集磁碟區大小。例如，如果您叢集的 `VolumeBytesUsed` 大小為 107,374,182,400 位元組 (100GiB)，而保留期間為兩天，則 `BackupRetentionPeriodStorageUsed` 的最大值為 214,748,364,800 位元組 (100 GiB + 100 GiB)。
- `SnapshotStorageUsed` 表示用於儲存在備份保留期間外的手動快照的備份儲存量 (以位元組為單位)。手動快照不會計入快照備份儲存，手動快照的建立時間戳記在保留期間內。所有自動快照也不會計入您的快照備份儲存。每個快照的大小是您取得快照時的叢集磁碟區大小。`SnapshotStorageUsed` 值取決於您保留的快照數量和每個快照的大小。例如，假設您在保留期間之外有一個手動快照，且叢集在拍攝快照時的 `VolumeBytesUsed` 大小為 100 GiB。的量 SnapshotStorageUsed 是 107,374,182,400 個位元組 (100 GiB)。
- `TotalBackupStorageBilled` 表示 `BackupRetentionPeriodStorageUsed` 和 `SnapshotStorageUsed` 的總和 (以位元組為單位)，減去相當於叢集磁碟區大小的一天免費備份儲存量。可用備份儲存等同於最新的磁碟區大小。例如，如果您叢集的 `VolumeBytesUsed` 大小是 100 GiB，您的保留期間是兩天，而您在保留期間之外有一個手動快照，則 `TotalBackupStorageBilled` 是 214,748,364,800 個位元組 (200 GiB + 100 GiB - 100 GiB)。

您可以透過CloudWatch 主控台監視 Neptune 叢集並使用 CloudWatch 指標建立報告。如需如何使用 CloudWatch 指標的詳細資訊，請參閱監控海王星 (p. 770)和中的指標表格海王星CloudWatch度量 (p. 775)。

從 Neptune 備份還原資料

您可以從 Neptune 保留的備份資料或您已儲存的資料庫叢集快照來建立新的 Neptune 資料庫叢集，以復原資料。您可以利用從備份資料所建立之資料庫叢集的新副本，快速還原至備份保留期之內的任何時間點。Neptune 備份在備份保留期之內的連續和增量性質，意味著您不需要經常建立資料的快照來改善還原時間。

若要判斷資料庫執行個體的最晚或最早可還原時間，請在 Neptune 主控控制台尋找`Latest Restorable Time`或`Earliest Restorable Time`值。資料庫叢集的最晚可還原時間是指可讓您還原資料庫叢集的最近時間點，通常在目前時間的 5 分鐘內。最早可還原時間指定您在備份保留期之內可將叢集磁碟區還原到多久以前。

您可以檢查 `Latest Restorable Time` 和 `Earliest Restorable Time` 值，以判斷資料庫叢集還原何時完成。在還原操作完成之前，`Latest Restorable Time` 和 `Earliest Restorable Time` 值會傳

回 NULL。如果 `Latest Restorable Time` 或 `Earliest Restorable Time` 傳回 NULL，則您無法要求備份或還原操作。

使用 AWS Management Console 將資料庫執行個體還原至指定的時間

1. 登入AWS管理主控控控控控控控控制台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控控控控制台
2. 在導覽窗格中，選擇 Instances (執行個體)。選擇您要還原之資料庫叢集的主要執行個體。
3. 選擇 Instance actions (執行個體動作)，然後選擇 Restore to point in time (還原至時間點)。

   在 [啟動資料庫執行個體] 視窗中，選擇 [還原時間] 下的 [
4. 在 Custom (自訂) 下指定您要還原至何時的日期和時間。
5. 在 [設定] 下輸入資料庫執行個體識別碼的新還原資料庫執行個體名稱。
6. 選擇 Launch DB Instance (啟動資料庫執行個體)，以啟動還原的資料庫執行個體。

   將會以您指定的名稱建立新的資料庫執行個體，並建立新的資料庫叢集。資料庫叢集名稱是新的資料庫執行個體名稱，後面加上 `-cluster`。例如，若新的資料庫執行個體名稱是 `myrestoreddb`，則新的資料庫叢集名稱為 `myrestoreddb-cluster`。

Neptune 中的 Backup 窗口

自動備份會每天在偏好的備份時段內執行。如果備份需要的時間超過所分配的備份時段，備份會在時段結束後繼續執行直到完成。備份時段不可與每週的資料庫執行個體維護時段重疊。

在自動備份期間，儲存輸入/輸出可能會在備份程序初始化時短暫暫停 (通常在幾秒內)。在備份異地同步備份部署時，可能會有幾分鐘的延遲增加情形。

備份時段通常會從每個區域之 8 小時時段內隨機選取 Neptune。針對每個區域的時段，預設 Backup 時段會從此時段中指派。Amazon RDS 使用者指南的備份時段會記錄在 Amazon RDS 使用者指南的備份時段落。

在 Neptune 中建立資料庫叢集快照

Neptune 會建立資料庫叢集的儲存體磁碟區快照，因此會備份整個資料庫叢集，而不只是個別的資料庫。當您建立資料庫叢集快照時，您需要識別要備份的資料庫叢集。接著，為您的資料庫叢集快照命名，以便稍後可從該快照還原。建立資料庫叢集快照所需的時間長短隨資料庫的大小而異。快照包含整個儲存磁碟區。因此，檔案大小 (例如暫存檔案) 也會影響建立快照所需的時間。

您可以使用AWS Management Console、或 Neptune API 建立資料庫叢集快照。AWS CLI

使用主控台建立資料庫叢集快照

建立資料庫叢集快照

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 在資料庫執行個體清單，選取資料庫叢集的主要執行個體。
4. 選擇 Instance actions (執行個體動作)，然後選擇 Take snapshot (擷取快照)。

   Take DB Snapshot (建立資料庫快照) 視窗隨即顯示。
5. 在 Snapshot name (快照名稱) 方塊中輸入資料庫叢集快照的名稱。
6. 選擇 Take Snapshot (擷取快照)。

從資料庫叢集快照還原

建立資料庫叢集的 Amazon Neptune 快照，Neptune 會建立叢集的儲存體磁碟區快照快照，因此會備份其所有資料，而不只是個別的執行個體。您稍後可從此資料庫叢集快照還原資料庫叢集快照還原資料庫叢集快照還原 還原資料庫叢集時，您需提供要做為還原來源的資料庫叢集快照名稱，然後提供要透過還原操作建立的新資料庫叢集名稱。

內容

- 從快照還原 Neptune 資料庫叢集叢集叢集快照還原 Neptune 資料庫叢集 (p. 811)
  - 您無法還原到現有的資料庫叢集 (p. 811)
  - 不會還原任何實例 (p. 811)
  - 不還原自訂參數群組 (p. 811)
  - 沒有自訂安全性群組 (p. 811)
  - 您無法從共用加密快照還原 (p. 812)
  - 還原的資料庫叢集會使用與以前一樣多的儲存 (p. 812)
- 如何從快照還原快照還原快照 (p. 812)
  - 使用主控台從快照還原 (p. 812)

從快照還原 Neptune 資料庫叢集叢集叢集快照還原 Neptune 資料庫叢集

您無法還原到現有的資料庫叢集

還原程序一律會建立新的資料庫叢集，因此您無法還原至已存在的資料庫叢集。

不會還原任何實例

還原建立的新資料庫叢集沒有與其相關聯的執行個體。

一旦還原完成且您的新資料庫叢集可供使用，請明確建立您需要的執行個體。您可以在 Neptune 主控台上執行此操作，或使用CreateDBInstance (p. 1056) API。

不還原自訂參數群組

由還原建立的新資料庫叢集會自動具有與其相關聯的預設資料庫參數群組。

一旦還原完成且您的新資料庫叢集可供使用，請將您還原的執行個體正在使用的任何自訂資料庫叢集進行關聯。若要這麼做，請使用 Neptune 主控台或ModifyDBInstance (p. 1069) API 上的 [修改] 命令。

> **Important**
>
> 建議您儲存正在建立快照的資料庫叢集中使用的自訂參數群組。然後，當您從該快照還原時，您可以輕鬆地將正確的參數群組與還原的資料庫叢集建立關聯。

沒有自訂安全性群組

由還原建立的新資料庫叢集會自動擁有與其相關聯的預設安全性群組。

一旦還原完成且您的新資料庫叢集可供使用，請將您還原的執行個體正在使用的任何自訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂訂關聯 若要這麼做，請使用 Neptune 主控台或ModifyDBInstance (p. 1069) API 上的 [修改] 命令。

您無法從共用加密快照還原

您無法從同時共用又加密的資料庫叢集快照還原資料庫叢集快照還原資料庫叢集快照

請改為建立快照的非共用副本，然後從副本還原。

還原的資料庫叢集會使用與以前一樣多的儲存

當您從資料庫叢集快照還原資料庫叢集時，配置給新叢集的儲存空間數量與配置給建立快照的資料庫叢集相同，無論該配置的儲存空間實際使用量有多少。

換句話說，向您收費的「高水位標記」不會改變。重設高水位需要從圖形中導出資料，然後將其重新載入新資料庫叢集 (請參閱海王星儲存計費 (p. 119))。

如何從快照還原快照還原快照

您可使用AWS Management Console、或 Neptune API 從資料庫叢集快照還原資料庫叢集快照還原資料庫叢集快照還原資料庫叢集AWS CLI

使用主控台從快照還原

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要從中還原的資料庫叢集快照。
4. 選擇 Actions (動作) 和 Restore Snapshot (還原快照)。
5. 在 Restore DB Instance (還原資料庫執行個體) 頁面上，於 DB Instance Identifier (資料庫執行個體識別符) 方塊中輸入已還原的資料庫叢集名稱。
6. 選擇 Restore DB Instance (還原資料庫執行個體)。
7. 如果您想要將資料庫叢集的功能還原為建立快照來源的資料庫叢集的功能，您必須修改資料庫叢集，才能使用安全群組。後續步驟假設您的資料庫叢集位於虛擬私有雲端 (VPC) 中。如果您的資料庫叢集不在 VPC 中，請使用 Amazon EC2 主控台來找到資料庫叢集所需的安全性群組

   a. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
   b. 在導覽窗格中，選擇 Security Groups (安全群組)。
   c. 選擇您想要用於資料庫叢集的安全群組。如果必要，請新增規則以將安全群組連結至 EC2 執行個體的安全群組。

複製資料庫叢集快照

透過 Neptune，即可自動或手動複製資料庫叢集快照。複製快照之後，副本即為手動快照。

您可以在相同AWS區域內以及跨區AWS域複製快照。

將自動快照複製到另一個AWS帳戶需要兩個步驟：首先，從自動快照建立手動快照，然後將手動快照複製到另一個帳戶。

除了複製，您也可以將手動快照共用給其他 AWS 帳戶。如需詳細資訊，請參閱共享資料庫叢集快照 (p. 817)。

主題

- 複製快照時的限制 (p. 813)
- 保留資料庫叢集快照副本 (p. 813)
- 複製快照時處理加密 (p. 814)
- 跨AWS區域複製快照 (p. 814)
- 使用主控台複製資料庫叢集快照 (p. 814)
- 使用 AWS CLI 複製資料庫叢集快照 (p. 815)

複製快照時的限制

以下是複製快照時的一些限制：

- 您可以在中國 (北京) 和中國 (寧夏) 和中國 (寧夏) 間複製快照，但無法在這些中國區域和其他AWS區域之間複製快照。
- 您可以在AWS GovCloud (US-East) 和 (US-West) 和AWS GovCloud (US-West) 之間複製快照，但無法在這些AWS GovCloud (US)區域和其他AWS區域之間複製快照。
- 如果您在目標快照變成可用之前刪除來源快照，則快照副本可能會失敗。刪除來源快照之前，請確認目標快照的狀態為 `AVAILABLE`。
- 每一帳戶的單一區域最多可有 5 個進行中的快照副本請求。
- 根據涉及的區域和要複製的資料量而定，跨區域快照複製可能需要數小時才會完成。

  如果有大量跨區域快照複製請求來自某個來源的來源AWS區域，在一些進行中的複製完成之前，Neptune 可能會將該來源AWS區域新提出的跨區域複製請求排入佇列。位於該佇列中的複製請求不會顯示進度資訊。只有在複製開始後才會顯示進度資訊。

保留資料庫叢集快照副本

Neptune 刪除自動快照，如下所示：

- 在其保留期結束時。
- 當您停用資料庫叢集的自動快照時。
- 當您刪除資料庫叢集時。

如果想要讓自動快照保留期間更長，請複製自動快照來建立手動快照，即可保留到您刪除為止。如果手動快照超出預設儲存空間，則其可能產生 Neptune 儲存成本。

如需備份儲存成本的詳細資訊，請參閱 Neptune 定價。

複製快照時處理加密

您可以複製以 AWS KMS 加密金鑰所加密的快照。如果您複製加密快照，則快照的副本也必須加密。您可以使用與原始快照相同的AWS KMS加密金鑰來加密副本，或指定不同的AWS KMS加密金鑰。

複製時，您不能加密未加密的資料庫叢集快照。

對於 Amazon Neptune 資料庫叢集快照，您也可以不加密資料庫叢集快照，而是在還原時指定AWS KMS加密金鑰。還原的資料庫叢集會以指定的金鑰加密。

跨AWS區域複製快照

> **Note**
>
> 此功能從 Neptune 引擎版本 1.0.2.1 (p. 969) 開始提供。

將快照複製到與來源快照之AWS區域不同的區AWS域時，第一個複本是完整的快照複本，即使您複製的是增量式快照也是如此。完整的快照副本內含所有還原資料庫執行個體所需的資料及中繼資料。完成第一個快照副本之後，即可將相同資料庫執行個體的增量式快照，複製至相同AWS帳戶之中的相同目的地區域。

增量式快照僅含相同資料庫執行個體最近一次拍攝快照之後所變更的資料。增量式快照複製速度較快，而且儲存成本較完整的快照複製低。加密與未加密快照均適用跨 AWS 區域的增量式快照複製。

> **Important**
>
> 對於共用快照，不支援複製增量式快照。對於共用快照，即使位於相同區域內，所有副本都是完整快照。

根據涉及的區AWS域和要複製的資料量而定，跨區域快照複製可能需要數小時才會完成。

使用主控台複製資料庫叢集快照

如果來源資料庫引擎是 Neptune，則快照即為資料庫叢集快照。對於每個AWS帳戶，您一次最多可以複製五個資料庫叢集快AWS照。支援複製加密和未加密的資料庫叢集快照。

如需資料傳輸定價的詳細資訊，請參閱 Neptune 定價。

若要取消已開始進行的複製操作，請在目標資料庫叢集快照處於 copying (複製中) 狀態時，刪除其快照。

下列程序適用於複製加密或未加密的資料庫叢集快照：

複製資料庫叢集快照

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選取您要複製之資料庫叢集快照的核取方塊。
4. 選擇 Actions (動作)，然後選擇 Copy Snapshot (複製快照)。Make Copy of DB Snapshot (複製資料庫快照) 頁面隨即顯示。
5. 在 New DB Snapshot Identifier (新的資料庫快照識別符) 中輸入資料庫叢集快照副本的名稱。
6. 若要將快照中的標籤和值複製到快照的副本，請選擇 Copy Tags (複製標籤)。
7. 在 Enable Encryption (啟用加密) 中，選擇以下其中一項：
   - 如果資料庫叢集快照未加密，且您不要將副本加密，請選擇 Disable encryption (停用加密)。
   - 如果資料庫叢集快照未加密，但您想要將副本加密，請選擇 Enable encryption (啟用加密)。在此情況下，在主金鑰中，指定用來加密資料庫叢集快照複本的AWS KMS金鑰識別符。

- 如果資料庫叢集快照已加密，請選擇 Enable encryption (啟用加密)。在此情況下，您必須加密副本，所以已選取 Yes (是)。在主金鑰中，指定用來加密資料庫叢集快照複本的AWS KMS金鑰識別符。

8. 選擇 Copy Snapshot (複製快照)。

使用 AWS CLI 複製資料庫叢集快照

您可以使用此copy-db-cluster-snapshot (p. 1115)AWS CLI命令來複製資料庫快照。

如果您要將快照複製到新的AWS區域，請在新的區域中執行此命令。

使用以下參數說明和範例來決定使用 AWS CLI 複製快照時要使用哪些參數。

- `--source-db-cluster-snapshot-identifier` – 來源資料庫快照的識別符。
  - 如果來源快照和副本位於相同的AWS區域，請指定有效的資料庫快照識別符，如`neptune:instance1-snapshot-20130805`。
  - 如果來源快照和副本位於不同的AWS區域，請指定有效的資料庫快照 ARN，請指定有效的資料庫快照 ARN`arn:aws:neptune:us-west-2:123456789012:snapshot:instance1-snapshot-20130805`。
  - 如果您要從共用的手動資料庫快照複製，此參數必須是共用資料庫快照的 Amazon 資源名稱 (ARN)。
  - 如果您要複製加密快照，此參數必須是來源AWS區域的 ARN 格式，且必須符合`PreSignedUrl`參數`SourceDBSnapshotIdentifier`中的。
- `--target-db-cluster-snapshot-identifier`— 加密資料庫快照之新副本的識別符。
- `--kms-key-id`— 加密資料庫快照的AWS KMS金鑰 ID。AWS KMS 金鑰 ID 是 AWS KMS 加密金鑰的 Amazon Resource Name (ARN)、AWS KMS 金鑰識別符或 AWS KMS 金鑰別名。
  - 如果您從AWS帳戶複製加密的資料庫快照，您可以指定此參數的值，以使用新的AWS KMS加密金鑰來加密複本。如果您不指定此參數的值，則會使用與來源資料庫快照相同的AWS KMS金鑰來加密資料庫快照的副本。
  - 您無法使用此參數來建立未加密快照的加密複本。嘗試這樣做會產生錯誤。
  - 如果您將加密的快照複製到不同的AWS區域，則必須指定目的地AWS區域的AWS KMS金鑰。 AWS KMS加密金鑰專用於其建立時所在的AWS區域，您無法將一個區域的加密金鑰用在另一個AWSAWS區域。
- `--source-region`— 來源資料庫快照所在的AWS區域 ID。如果您將加密快照複製到不同的 AWS 區域，則必須指定此選項。
- `--region`— 您要將快照複製到其中的AWS區域 ID。如果您將加密快照複製到不同的 AWS 區域，則必須指定此選項。

Example 從未加密的快照到相同區域

下列程式碼會使用新名稱`mydbsnapshotcopy`建立從`us-east-1`AWS區域到區域的快照複本。`us-west-2`

對於 Linux、OS X 或 Unix：

```
aws neptune copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier instance1-snapshot-20130805 \
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy
```

針對 Windows：

```
aws neptune copy-db-cluster-snapshot ^
  --source-db-cluster-snapshot-identifier instance1-snapshot-20130805 ^
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy
```

Example 從未加密的快照跨區域

下列程式碼會使用新名稱mydbsnapshotcopy建立從us-east-1AWS區域到區域的快照複本。us-west-2在 us-west-2 區域中執行命令。

對於 Linux、OS X 或 Unix：

```
aws neptune copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier arn:aws:neptune:us-
east-1:123456789012:snapshot:instance1-snapshot-20130805 \
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy \
  --source-region us-east-1 \
  --region us-west-2
```

針對 Windows：

```
aws neptune copy-db-cluster-snapshot ^
  --source-db-cluster-snapshot-identifier arn:aws:neptune:us-
east-1:123456789012:snapshot:instance1-snapshot-20130805 ^
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy ^
  --source-region us-east-1 ^
  --region us-west-2
```

Example 從已加密的快照跨區域

下列程式碼範例會在區域中，將加密的資料庫快照從us-east-1AWS區域複製到us-west-2區域。在 us-west-2 區域中執行命令。

對於 Linux、OS X 或 Unix：

```
aws neptune copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier arn:aws:neptune:us-
west-2:123456789012:snapshot:instance1-snapshot-20161115 \
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy \
  --source-region us-east-1 \
  --region us-west-2
  --kms-key-id my_us_west_2_key
```

針對 Windows：

```
aws neptune copy-db-cluster-snapshot ^
  --source-db-cluster-snapshot-identifier arn:aws:neptune:us-
west-2:123456789012:snapshot:instance1-snapshot-20161115 ^
  --target-db-cluster-snapshot-identifier mydbsnapshotcopy ^
  --source-region us-east-1 ^
  --region us-west-2
  --kms-key-id my-us-west-2-key
```

共享資料庫叢集快照

使用 Neptune ement Services Developer Guide 中的方式共享手動資料庫叢集快照：

- 共用手動資料庫叢集快照 (無論加密或未加密) 可讓授權的 AWS 帳戶複製快照。
- 分享手動資料庫叢集快照 (無論加密或未加密) 可讓授權的 AWS 帳戶直接從快照還原資料庫叢集，而不需要先複製再還原。

> **Note**
>
> 若要共用自動資料庫叢集快照，請複製該自動快照來建立手動資料庫叢集快照，然後共用該複本。

如需從資料庫叢集快照還原資料庫叢集的詳細資訊，請參閱如何從快照還原快照還原快照 (p. 812)。

您可以將手動快照最多共用給其他 20 個 AWS 帳戶。您也可以將未加密的手動快照以公有形式共用，讓所有 AWS 帳戶都可使用此快照。將快照以公有形式共用時，請小心不要在任何公有快照中包含您的私人資訊。

> **Note**
>
> 使用AWS Command Line Interface (AWS CLI) 或 Neptune per API，從共享快照還原資料庫叢集時，您必須指定共享快照的 Amazon Resource Name (ARN) 作為快照識別符。

主題

- 共用加密的資料庫叢集快照。 (p. 817)
- 共享資料庫叢集快照 (p. 819)

共用加密的資料庫叢集快照。

您可以共用已使用 AES-256 加密演算法「靜態」加密的資料庫叢集快照。如需詳細資訊，請參閱在靜態時加密海王星資源 (p. 155)。若要這樣做，您必須採取下列步驟：

1. 將用於對快照進行加密的 AWS Key Management Service (AWS KMS) 加密金鑰，與您希望能夠存取此快照的任何帳戶共享。

   您可以將其他AWS帳戶新增到 KMS 金AWS KMS鑰政策，以便與另一個帳戶共享加密金鑰。如需有關更新金鑰政策的詳細資訊，請參閱 https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html 開發人員指南中的AWS KMS金鑰政策。有關建立金鑰政策的範例，請參閱本主題稍後的建立 IAM 政策以允許複製加密快照 (p. 818)。
2. 使用AWS Management ConsoleAWS CLI、或 Neptune per API，與其他帳戶共享加密的快照。

共用加密快照有下列限制：

- 您無法將加密快照以公有形式共用。
- 如果快照以共用此快照共用此快照的AWS帳戶之預設AWS KMS加密金鑰。

允許存取 AWS KMS 加密金鑰

若要讓另一個AWS帳戶複製您的帳戶所共享的已加密資料庫叢集快照，您與之共享快照的帳戶必須能夠存取用於加密該快照的 KMS 金鑰。若要允許另一個AWS帳戶存取金AWS KMS鑰政策，請以您在 KMS 金鑰政策中以所共享之AWS帳戶的 ARN，更新 KMS 金鑰`Principal`的金鑰政策。然後允許 `kms:CreateGrant` 動作。如需一般指示，請參閱AWS Key Management Service開發人員指南中的允許其他帳戶中的使用者使用 KMS 金鑰。

在您授權AWS帳戶存取您的 KMS 加密金鑰之後，該AWS帳戶若要複製您加密的快照，必須建立 IAM 使用者 (如果仍無此使用者)。KMS 安全性限制不允許針對此使用根AWS帳戶身分識別。該AWS帳戶還必須將 IAM 政策附加到該 IAM 使用者，以允許 IAM 使用者以您的 KMS 金鑰來複製加密資料庫叢集快照。

在以下金鑰政策範例中，使用者 111122223333 是 KMS 加密金鑰的擁有者，使用者 444455556666 是共用金鑰的帳戶。這個更新的金鑰政策將使用者 444455556666 之 AWS 根帳戶身分的 ARN 納入為政策的 `Principal`，並允許 `kms:CreateGrant` 動作，以授予 AWS 帳戶存取 KMS 金鑰的權利。

```
{
  "Id=": "key-policy-1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid=": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/KeyUser",
        "arn:aws:iam::444455556666:root"
      ]},
      "Action": [
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid=": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/KeyUser",
        "arn:aws:iam::444455556666:root"
      ]},
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
    }
  ]
}
```

建立 IAM 政策以允許複製加密快照

在外部AWS帳戶可以存取您的 KMS 金鑰之後，該帳戶的擁有者就可以建立政策，以允許針對該帳戶所建立的 IAM 使用者複製以該 KMS 金鑰加密的加密快照。

以下列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列列範例AWS444455556666 它可以讓 IAM 使用者，能夠從AWS帳戶複製共111122223333用快照已在us-west-2區域c989c1dd-a3f2-4a5d-8d96-e793d082ab26中以 KMS 金鑰加密。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid=": "AllowUseOfTheKey",
            "Effect": "Allow",
```

```
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:111122223333:key/c989c1dd-a3f2-4a5d-8d96-
e793d082ab26"]
        },
        {
            "Sid": "AllowAttachmentOfPersistentResources",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:111122223333:key/c989c1dd-a3f2-4a5d-8d96-
e793d082ab26"],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

如需有關更新金鑰政策的詳細資訊，請參閱 https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html 開發人員指南中的AWS Key Management Service金鑰政策。

共享資料庫叢集快照

您可以使用AWS Management Console、或 Neptune per API 共享資料庫叢集快照。AWS CLI

使用主控台共享資料庫叢集快照

使用 Neptune per 主控台台，您可以將手動資料庫叢集快照分享給最多 20 個AWS帳戶。您也可以停止與一或多個帳戶共享手動快照。

共享手動資料庫叢集快照

1. 登入AWS管理主控台台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune Developer Guide。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇您要共享的手動快照。
4. 選擇 Actions (動作) 和 Share Snapshot (共享快照)。
5. 在 DB snapshot visibility (資料庫快照可見度) 中，選擇下列其中一個選項。

   - 如果來源未加密，請選擇 [公用] 以允許所有AWS帳戶從手動資料庫叢集快照還原資料庫叢集。或選擇 [專用]，僅允許您指定從手動資料庫叢集快照還原資料庫叢集的AWS帳戶。

     Warning

     如果您將 DB Snapshot Visibility (資料庫快照可見度) 設為 Public (公有)，則所有 AWS 帳戶皆可從您的手動資料庫叢集快照還原資料庫叢集，而且可以存取您的資料。請勿將任何包含私人資訊的手動資料庫叢集快照以 Public (公有) 形式共用。

- 如果來源資料庫叢集已加密，DB snapshot visibility (資料庫快照可見度) 會設為 Private (私有)，因為加密快照無法以公有形式共用。

6. 在AWS帳戶 ID 中，輸入AWS您要允許從手動快照還原資料庫叢集的帳戶識別碼。接著選擇 Add (新增)。重複上一步來包含其他 AWS 帳戶識別符，最多 20 個 AWS 帳戶。

   如果您誤將 AWS 帳戶識別符新增至允許的帳戶清單，您可以選擇錯誤 AWS 帳戶識別符右邊的 Delete (刪除)，以從清單中刪除此識別符。
7. 為您要允許還原手動快照的所有AWS帳戶新增識別碼之後，請選擇 [儲存]。

如要停止與 AWS 帳戶共享手動資料庫叢集快照

1. 在 https://console.aws.amazon.com/neptune/home 上打開 Amazon Neptune 控制台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇您希望停止共享的手動快照。
4. 選擇 Actions (動作)，然後選擇 Share Snapshot (共享快照)。
5. 若要移除 AWS 帳戶的許可，請從授權帳戶清單中，對該帳戶的 AWS 帳戶識別符選擇 Delete (刪除)。
6. 選擇 Save (儲存)。

刪除 Neptune 快照

您可以使用AWS Management Console、或 Neptune 管理 API 刪除資料庫快照：AWS CLI

使用主控台刪除

1. 登入AWS管理主控台，開啟位於 https://console.aws.amazon.com/neptune/home 的 Amazon Neptune 主控台。
2. 在導覽窗格中，選擇 Snapshots (快照)。
3. 選擇想要刪除的資料庫快照。
4. 針對 Actions (動作)，選擇 Delete Snapshot (刪除快照)。
5. 在確認頁面上，選擇 Delete (刪除)。

使用 AWS CLI 刪除

您也可以使用 AWS CLI delete_db_cluster_snapshot (p. 1113) 命令，刪除資料庫快照，使用 `--db-snapshot-identifier` 參數以識別您要刪除的快照：

對於 Linux、OS X 或 Unix：

```
aws neptune delete-db-cluster-snapshot \
    --db-snapshot-identifier <name-of-the-snapshot-to-delete>
```

針對 Windows：

```
aws neptune delete-db-cluster-snapshot ^
    --db-snapshot-identifier <name-of-the-snapshot-to-delete>
```

使用 Neptune 管理 API 刪除

您可以使用其中一個 SDK，藉由呼叫 刪除資料庫 ClusterSnapshot (p. 1113) API 來刪除資料庫快照，並使用 `DBSnapshotIdentifier` 參數以識別要刪除的資料庫快照。

最佳做法：充分利用 Neptune

下列是搭配 Amazon Neptune 運作的一般建議。使用本節作為參考，快速找到使用 Amazon Neptune 並提升效能的建議。

內容

- Amazon Neptune 基本操作準則 (p. 823)
  - Amazon Neptune 安全最佳實務 (p. 824)
  - 避免叢集中的不同執行個體類別 (p. 824)
  - 在大量載入期間避免重新啟動 (p. 824)
  - 如果您有大量述詞，請啟用 OSGP 索引 (p. 824)
  - 盡可能避免長時間執行的交易 (p. 824)
  - 使用 Neptune 指標的最佳實務 (p. 825)
  - 調校 Neptune 查詢的最佳實務 (p. 825)
  - 跨僅供讀取複本的負載平衡 (p. 825)
  - 使用臨時更大的實例加載更快 (p. 826)
  - 容錯移轉至僅供讀取複本來調整您的寫入器執行個體 (p. 826)
  - 資料預先擷取工作中斷錯誤後重試上傳 (p. 826)
- 使用 Neptune 使用 Gemlin 的一般最佳實務 (p. 827)
  - 在您將部署它的上下文中測試 Girmlin 代碼 (p. 827)
  - 結構更新查詢以利用 DFE 引擎 (p. 828)
  - 建立有效率的多執行緒 Gremlin 寫入 (p. 828)
  - 利用建立時間屬性清除記錄 (p. 828)
  - 對 Groovy 時間資料使用 datetime( ) 方法 (p. 829)
  - 在 GLV 時間資料使用原生日期和時間 (p. 829)
- 使用小鬼 Java 客戶端和 Neptune 的最佳實踐 (p. 830)
  - 請使用最新版本的 Apache TinkerPop Java 客戶端 (p. 830)
  - 跨多個線程重新使用客戶端對象 (p. 830)
  - 為讀寫端點創建單獨的 Java 客戶端對象 (p. 830)
  - 將多個僅供讀取複本端點新增至 Gemlin Java 連線集區 (p. 831)
  - 關閉用戶端以避免連線限制 (p. 831)
  - 容錯移轉後建立新連線 (p. 831)
  - 設置maxInProcessPerConnection和maxSimultaneousUsagePerConnection相同的值 (p. 831)
  - 將查詢作為字節碼發送到服務器，而不是作為字符串 (p. 831)
  - 始終完全消耗查詢返回的 ResultSet 或迭代器 (p. 832)
  - 批量添加頂點和邊 (p. 832)
  - 在 Java 虛擬機器中停用 DNS 快取 (p. 833)
  - 選擇性地在每個查詢層級設定逾時 (p. 833)
  - 解決 3.3.4 以前的客戶端中保持活動的錯誤 (p. 834)
  - 排除「小鬼 Java 客戶端」異常問題 (p. 834)
    - java.util.concurrent.TimeoutException 疑難排解 (p. 834)
- 使用開放密碼和螺栓的 Neptune 最佳實踐 (p. 835)
  - 偏好定向於查詢中的雙向邊緣 (p. 835)

- Neptune 不支持事務中的多個並發查詢 (p. 835)
- 容錯移轉後建立新連線 (p. 836)
- 長壽命應用的連接處理 (p. 836)
- 連線處理 AWS Lambda (p. 836)
- 完成後關閉驅動程序對象 (p. 836)
- 使用明確的交易模式進行讀取和寫入 (p. 836)
  - 唯讀交易 (p. 836)
  - 唯讀交易 (p. 836)
- 例外狀況的重試邏輯 (p. 838)

- 使用 SPARQL 的 Neptune 最佳做法 (p. 840)
  - 預設查詢所有具名圖形 (p. 840)
  - 指定要載入的具名圖表 (p. 841)
  - 在您的查詢中選擇 FILTER、FILTER...IN 和 VALUES (p. 841)

Amazon Neptune 基本操作準則

下列是使用 Neptune 時，應遵循的基本操作準則。

- 瞭解 Neptune 資料庫執行個體，以便根據效能和使用案例需求適當調整其大小。請參閱 亞馬遜海王星資料庫叢集和執 (p. 117)。
- 監控您的 CPU 和記憶體用量。這可協助您了解何時遷移到有更大 CPU 或記憶體容量的資料庫執行個體類別，以達到您需要的查詢效能。您可以設定 Amazon CloudWatch 在使用模式變更或接近部署容量時通知您。這樣做有助於維持系統效能和可用性。如需詳細資訊，請參閱監控執行個 (p. 118)和監控海王星 (p. 770)。

  由於 Neptune 有自己的記憶體管理員，所以即使 CPU 使用率很高，也會看到相對較低的記憶體使用率是正常的。執行查詢時遇到 out-of-memory 例外狀況是增加可用記憶體所需的最佳指標。
- 啟用自動備份並設定備份時段，在方便的時間進行備份。
- 測試資料庫執行個體的容錯移轉，以了解您的使用案例執行此程序需時多長。它也有助於確保可存取您資料庫執行個體的應用程式，可以在容錯移轉後，自動連線到新的資料庫執行個體。
- 如果可能，請在相同區域和 VPC 中執行用戶端與 Neptune 叢集，因為使用 VPC 對等互連進行跨區域連線可能會導致查詢回應時間的延遲。對於以毫秒為單位的查詢回應，必須將用戶端和 Neptune 叢集保持在相同的區域和 VPC 中。
- 當您建立僅供讀取複本執行個體時，它至少應該與主要寫入器執行個體一樣大。這有助於在檢查時保持複寫延遲，並避免複本重新啟動。請參閱 避免叢集中的不同執行個體類別 (p. 824)。
- 在升級到新的主要引擎版本之前，請務必在升級之前對其進行測試。您可以透過複製資料庫叢集，以便複製叢集執行新的引擎版本，然後在複製上測試應用程式來執行此操作。
- 為了便於容錯移轉，所有執行個體最好都應該是相同的大小。

主題

- Amazon Neptune 安全最佳實務 (p. 824)
- 避免叢集中的不同執行個體類別 (p. 824)
- 在大量載入期間避免重新啟動 (p. 824)
- 如果您有大量述詞，請啟用 OSGP 索引 (p. 824)
- 盡可能避免長時間執行的交易 (p. 824)
- 使用 Neptune 指標的最佳實務 (p. 825)
- 調校 Neptune 查詢的最佳實務 (p. 825)
- 跨僅供讀取複本的負載平衡 (p. 825)

- 使用臨時更大的實例加載更快 (p. 826)
- 容錯移轉至僅供讀取複本來調整您的寫入器執行個體 (p. 826)
- 資料預先擷取工作中斷錯誤後重試上傳 (p. 826)

Amazon Neptune 安全最佳實務

使用AWS Identity and Access Management (IAM) 帳戶控制對 Neptune API 動作的存取。控制建立、修改或刪除 Neptune 資源的動作 (例如資料庫執行個體、安全群組、選項群組或參數群組)，以及執行一般管理動作 (例如備份和還原資料庫執行個體) 的動作。

- 盡可能使用臨時認證而不是持續性憑證。
- 將個別 IAM 帳戶指派給管理 Amazon Realase Service (Amazon RDS) 資源的每個使用者。切勿使用AWS 帳號根使用者來管理 Neptune 資源。為每個人 (包含您) 建立 IAM 使用者。
- 授予每個使用者執行其職責所需的最低許可集。
- 使用 IAM 群組來有效管理多個使用者的許可。
- 定期輪替您的 IAM 登入資料。

如需使用 IAM 存取 Neptune 資源的詳細資訊，請參閱亞馬遜海王星的安全 (p. 153)。如需使用 IAM 的一般資訊，請參閱 IAM 使用者指南中的AWS Identity and Access Management和 IAM 最佳實務。

避免叢集中的不同執行個體類別

當您的資料庫叢集包含不同類別的執行個體時，可能會發生一段時間的問題 最常見的問題是，由於複寫延遲，小型讀取器執行個體可能會進入重複重新啟動的週期。如果讀取器節點的資料庫執行個體類別組態比寫入器資料庫執行個體的組態還弱，則變更量可能太大，無法 catch 讀取器。

> Important
>
> 若要避免因複寫延遲而重複重新啟動，請設定資料庫叢集，讓所有執行個體都具有相同的執行個體類別 (大小)。

您可以使用 Amazon 中的`ClusterReplicaLag`指標查看寫入器執行個體 (主要執行個體) 與資料庫叢集中讀取器之間的延遲 CloudWatch。此指`VolumeWriteIOPs`標也可讓您偵測叢集中可能會造成複寫延遲的大量寫入活動。

在大量載入期間避免重新啟動

如果您因為大量載入期間的複寫延遲而遭遇重複僅供讀取複本重新啟動的週期，則您的複本可能無法跟上資料庫叢集中的寫入器。

將讀取器縮放為大於寫入器，或在大量載入期間暫時將其移除，然後在完成後重新建立讀取器。

如果您有大量述詞，請啟用 OSGP 索引

如果您的資料模型包含大量不同述詞 (在大多數情況下超過千個)，您可能會遇到效能降低和更高的作業成本。

在這種情況下，您可以啟用 OSGP 索引 (p. 105)來改善效能。請參閱 OSGP 指數 (p. 128)。

盡可能避免長時間執行的交易

無論是唯讀或讀寫，長時間執行的作業事件都可能會造成下列類型的非預期問題：

在讀取器執行個體或具有並行寫入的寫入器執行個體上，長時間執行的交易可能會導致不同版本的資料大量累積。這可能會對讀取查詢產生較高的延遲，以篩選掉大部分結果。

在某些情況下，超過數小時的累積版本可能會導致新的寫入受到限制。

如果執行個體重新啟動，長時間執行的讀寫交易也可能會造成問題。如果執行個體從維護事件或當機重新啟動，則所有未認可的寫入都會復原。這類還原作業通常會在背景執行，而且不會封鎖執行個體回復，但是任何與要回復之作業衝突的新寫入都會失敗。

例如，如果在上次執行中斷連線之後重試相同的查詢，則在重新啟動執行個體時可能會失敗。

復原操作所需的時間與所涉及的變更大小成正比。

使用 Neptune 指標的最佳實務

若要識別由於資源不足和其他常見瓶頸造成的效能問題，您可以監控 Neptune 資料庫叢集的可用指標。

定期監控效能指標以蒐集各種時間範圍的平均值、最大值和最小值。效能降級時您便可以得知。使用這些資料，您可以針對特定指標臨界值設定 Amazon CloudWatch 警示，在達到這些臨界值便警示您。

設定新的資料庫叢集並讓它以一般工作負載執行時，嘗試擷取不同間隔數量 (例如，一小時、24 小時、一週、兩週) 所有效能指標的平均值、最大值和最小值。這可讓您了解正常運作情況為何。這有助於比較尖峰與離峰時段的操作。您接著可以使用此資訊來識別效能下降到標準層級之下的時機，並設定警示來應對。

如使用 Amazon 監控海王星 CloudWatch (p. 772)需如何檢視 Neptune 度量的相關資訊，請參閱。

以下是您開始的最重要指標：

- BufferCacheHitRatio— 由緩衝區快取提供服務的百分比。快取遺漏會為查詢執行增加顯著延遲。如果快取命中率低於 99.9%，而延遲對應用程式造成問題，請考慮升級執行個體類型，以便在記憶體中快取更多資料。
- CPU 使用率 — 已使用的電腦處理容量之之百分比。取決於您的查詢效能目標，較高的 CPU 使用量值有可能是適當的。
- 可釋放記憶體 — 資料庫執行個體上有多少 RAM 可用，以 MB 為單位。Neptune 有自己的記憶體管理員，因此此度量可能比您預期的要低。如果查詢經常拋出 out-of-memory 異常，那麼您應該考慮將實例類升級為具有更多 RAM 的實例類的一個好兆頭。

Monitoring (監控) 標籤指標的 CPU 及記憶體指標在 75% 會以紅線標記。如果執行個體記憶體的使用量經常超過該線，請檢查您的工作負載，並考慮將您的執行個體升級以改善查詢效能。

調校 Neptune 查詢的最佳實務

改善 Neptune 效能的一個最佳方式是調校最常使用與資源最密集的查詢，以讓它們的執行成本較不昂貴。

如需有關如何調整 Gemlin 查詢的資訊，請參閱小精靈查詢提示 (p. 319)和調整小精靈查詢 (p. 364)。如需如何調校 SPARQL 查詢的資訊，請參閱 查詢提示 (p. 437)。

跨僅供讀取複本的負載平衡

讀取器端點循環配置資源路由的運作方式是改變 DNS 項目指向的主機。用戶端必須建立新的連線並解析 DNS 記錄，才能取得新的僅供讀取複本的連線，因為 WebSocket 連線通常會長時間保持作用狀態。

若要針對連續請求取得不同的僅供讀取複本，請確保用戶端在每次連線時都能解析 DNS 項目。這可能需要關閉連線並重新連接到讀取者端點。

您也可以明確連接到執行個體端點，在各個僅供讀取複本之間對請求進行負載平衡。

使用臨時更大的實例加載更快

使用較大的執行個體可提高您的載入效能。如果您不是使用大型的執行個體類型，但您想要更高的載入速度，可以使用較大的執行個體來進行載入，然後將其刪除。

Note

下列程序是用於新的叢集。如果您有現有的叢集，您可以新增較大的執行個體，然後將其提升至主要資料庫執行個體。

使用較大的執行個體大小將資料載入

1. 建立具有單一 r5.12xlarge 執行個體的叢集。這個執行個體是主要資料庫執行個體。
2. 建立相同大小的一個或多個僅供讀取複本 (r5.12xlarge)。

   您可以使用較小的大小建立僅供讀取複本，但如果它們的大小不足以跟上主執行個體的寫入，則可能必須經常重新啟動。產生的停機時間會大幅降低效能。
3. 在批量加載器命令中，包括“parallelism” : “OVERSUBSCRIBE”告訴 Neptune 使用所有可用的 CPU 資源進行加載 (請參閱海王星裝載機請求參 (p. 276)) 。然後，加載操作將以 I/O 允許的速度進行，這通常需要 60-70% 的 CPU 資源。
4. 使用 Neptune 載入程式載入您的資料。載入工作會在主要資料庫執行個體上執行。
5. 資料載入完成後，請務必將叢集中的所有執行個體縮減至相同的執行個體類型，以避免額外的費用和重複的重新啟動問題 (請參閱避免不同的實例大小 (p. 824))。

容錯移轉至僅供讀取複本來調整您的寫入器執行個體

調整資料庫叢集中執行個體大小 (包括寫入器執行個體) 的最佳方法是建立或修改僅供讀取複本執行個體，使其具有您想要的大小，然後故意容錯移轉至該僅供讀取複本。您的應用程式看到的停機時間只是變更寫入器 IP 位址所需的時間，應該是 3 到 5 秒左右。

您用來容錯移轉目前寫入器執行個體至僅供讀取複本執行個體的 Neptune 管理 API 特意為FailoverDBCluster (p. 1032)。如果您正在使用 Gremlin Java 客戶端，那么您可能需要在故障轉移后創建一個新的客戶端對象，以獲取新的 IP 地址，如下 (p. 831)所述。

確保將所有實例更改為相同的大小，以避免重複重新啟動的週期，如下所述。

資料預先擷取工作中斷錯誤後重試上傳

當您使用大量載入器將資料載入 Neptune 時，有時可能會產生LOAD_FAILED狀態，PARSING_ERROR並回應詳細資訊的要求而報告和訊Data prefetch task interrupted息，如下所示：

```
"errorLogs" : [
  {
    "errorCode" : "PARSING_ERROR",
    "errorMessage" : "Data prefetch task interrupted: Data prefetch task for 11467 failed",
    "fileName" : "s3://some-source-bucket/some-source-file",
    "recordNum" : 0
  }
]
```

當此錯誤發生時，只要重試一次大量上傳請求。

當出現通常原因不在於您的請求或資料的暫時中斷時，就會出現這個錯誤，而且通常再次執行大量上傳請求就能加以解決。

如果您使用的是預設設定，即 `"mode":"AUTO"` 和 `"failOnError":"TRUE"`，這時載入器會略過其已成功載入的檔案，並恢復其在中斷發生時尚未載入的檔案載入。

使用 Neptune 使用 Gemlin 的一般最佳實務

在 Neptune 使用 Girmlin 圖遍歷語言時，請遵循這些建議。如需將 Girmlin 與 Neptune 搭配使用的資訊，請參閱the section called “Gremlin” (p. 300)。

Important

在 TinkerPop 版本 3.4.11 中進行了更改，可提高查詢處理方式的正確性，但目前有時會嚴重影響查詢性能。
例如，此類查詢的執行速度可能會明顯變慢：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  out()
```

極限步驟之後的頂點現在會以非最佳方式擷取，因為 TinkerPop 3.4.11 的變更。若要避免這種情況，您可以在之後的任何點加入 barrier () 步驟來修改查詢`order().by()`。例如：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  barrier().
  out()
```

TinkerPop 在 Neptune 引擎版本 1.0.5.0 (p. 931) 中已啟用 3.4.11。

主題

- 在您將部署它的上下文中測試 Girmlin 代碼 (p. 827)
- 結構更新查詢以利用 DFE 引擎 (p. 828)
- 建立有效率的多執行緒 Gremlin 寫入 (p. 828)
- 利用建立時間屬性清除記錄 (p. 828)
- 對 Groovy 時間資料使用 datetime( ) 方法 (p. 829)
- 在 GLV 時間資料使用原生日期和時間 (p. 829)

在您將部署它的上下文中測試 Girmlin 代碼

在 Gemlin 中，客戶端可以通過多種方式向服務器提交查詢：使用 WebSocket或字節碼 GLV，或者通過使用基於字符串的腳本通過 Gemlin 控制台。

重要的是要認識到，根據您提交查詢的方式，Gemlin 查詢執行可能會有所不同。如果在字節碼模式下提交，則返回空結果的查詢可能會被視為已成功，但如果在腳本模式下提交，則會被視為失敗。例如，如果您包含`next()`在指令碼模式查詢中，`next()`則會傳送至伺服器，但使用用 ByteCode 戶端通常會處理`next()`本身。在第一種情況下，如果找不到任何結果，查詢就會失敗，但在第二種情況下，無論結果集是否為空，查詢都會成功。

如果您在一個上下文中開發和測試您的代碼（例如，通常以文本形式提交查詢的 Gemlin 控制台），但是然後在不同的上下文中部署代碼（例如通過使用字節碼的 Java 驅動程序），則可能會遇到代碼在生產環境中的行為與開發環境中的行為不同的問題。

Important

請務必在將要部署的 GLV 內容中測試 Gremlin 程式碼，以避免發生未預期的結果。

結構更新查詢以利用 DFE 引擎

您可以盡可能充分利用 Neptune DFE 引擎，大幅改善 upsert 查詢的效能。

用小鬼mergeV()和步驟製作高效的升頻者 mergeE() (p. 334)說明如何建構 upsert 查詢，以盡可能有效地使用 DFE 引擎。

建立有效率的多執行緒 Gremlin 寫入

使用 Gremlin 將數據多線程加載到 Neptune 有一些準則。

如有可能，請提供每個執行緒一組頂點或邊緣，以插入或修改不衝突的項目。例如，執行緒 1 的地址 ID 範圍為 1–50,000，執行緒 2 地址 ID 範圍為 50,001–100,000，以此類推。這可減少出現 `ConcurrentModificationException` 的機率。為安全起見，所有寫入請圍以 `try/catch` 區塊。若任何一個項目失敗，您便可以在短暫的延遲後重試它們。

以介於 50 和 100 (頂點或邊緣) 間的批次大小批次寫入，通常效果很好。若您要為每個頂點新增許多屬性，接近 50 的數字可能比接近 100 的數字更好。這很值得進行一些實驗。因此針對批次寫入，您可以使用如下的內容：

```
g.addV('test').property(id,'1').as('a').
  addV('test').property(id,'2').
  addE('friend').to('a').
```

接著會在每一個批次操作中重複這段過程。

使用批次的效率，遠比在每一次 Gremlin 與伺服器的來回行程中新增一個頂點或邊緣高。

如果您使用的是 Gremlin 語言變體 (GLV) 用戶端，您可以先建立一個周遊，再以程式設計方式建立一個批次。然後，為它新增內容，最後再對它反覆運算，例如：

```
  t.addV('test').property(id,'1').as('a')
  t.addV('test').property(id,'2')
  t.addE('friend').to('a')
  t.iterate()
```

如果可能，最好使用 Gremlin 語言變體用戶端。但是，您可以串連字串來建立批次，用將查詢提交為文字字串的用戶端執行類似的作業。

若您使用其中一個 Gremlin 用戶端程式庫，而非基本的 HTTP 來進行查詢，則執行緒應該全部都會共享相同的用戶端、叢集或連線集區。您可能需要調整設定以取得最佳的傳輸量，例如連線集區的大小以及 Gemlin 用戶端使用的工作者執行緒數目等設定。

利用建立時間屬性清除記錄

您可以把建立時間儲存為頂點上的屬性，然後定期丟棄它們，藉此清除過時記錄。

如果需要讓資料存放一段特定時間，然後從圖形中移除 (頂點存留時間)，您可以在建立頂點時儲存時間戳記屬性。然後，您可以定期對所有在特定時間前建立的頂點發出 `drop()` 查詢，例如：

```
g.V().has("timestamp", lt(datetime('2018-10-11')))
```

對 Groovy 時間資料使用 datetime( ) 方法

Neptune 提供了datetime用於指定在 Gemlin Groovy 變種發送查詢的日期和時間的方法。這包括 Gremlin 主控台、使用 HTTP REST API 的文字字串，以及任何其他使用 Groovy 的序列化。

> **Important**
>
> 這「只」適用於以「文字字串」傳送 Gremlin 查詢的方法。如果使用的是 Gremlin 語言變體，您必須使用該語言的原生日期類別與函數。如需詳細資訊，請參閱下一節「the section called “原生日期與時間” (p. 829)」。
> 從 TinkerPop 3.5.2（在 Neptune 發動機版本 1.1.1.0 (p. 894) 中引入）開始，datetime是 TinkerPop.

您可以使用 datetime 方法來存放和比較日期：

```
g.V('3').property('date',datetime('2001-02-08'))
```

```
g.V().has('date',gt(datetime('2000-01-01')))
```

在 GLV 時間資料使用原生日期和時間

如果您是使用 Gremlin 語言變體 (GLV)，必須使用該程式語言為 Gremlin 時間資料提供的原生日期和時間類別與函數。

官方的 TinkerPop Java，Node.js（JavaScript），蟒蛇或 .NET 庫都是小鬼語言變體庫。

> **Important**
>
> 這「只」適用於「Gremlin 語言變體 (GLV) 程式庫」。如果您使用的方法將 Gemlin 查詢作為文本字符串發送，則必須使用 Neptune 提供的datetime()方法。這包括 Gremlin 主控台、使用 HTTP REST API 的文字字串，以及任何其他使用 Groovy 的序列化。如需詳細資訊，請參閱上一節the section called “datetime( )” (p. 829)。

Python

以下是 Python 範例的一部分，它會為頂點建立一個名為 'date' 且 ID 為 '3' 的屬性。它將值設為使用 Python datetime.now() 方法產生的日期。

```
import datetime

g.V('3').property('date',datetime.datetime.now()).next()
```

如需使用 Python 連線至 Neptune 的完整範例，請參閱使用 Python 連接到海王星資料庫執行個體 (p. 313)

Node.js (JavaScript)

以下是部分範例，其中 JavaScript 會針對 ID 為 'date' 的頂點建立名為 '3' 的單一屬性。它將值設為使用 Node.js Date() 建構函數產生的日期。

```
g.V('3').property('date', new Date()).next()
```

如需使用 Node.js 連線至 Neptune 的完整範例，請參閱使用 Node.js 連線至海王星資料庫執行個體 (p. 316)

Java

以下是 Java 範例的一部分，它會為頂點建立一個名為 'date' 且 ID 為 '3' 的屬性。它將值設為使用 Java Date() 建構函數產生的日期。

```
import java.util.date

g.V('3').property('date', new Date()).next();
```

如需使用 Java 連線至 Neptune 的完整範例，請參閱使用 Java 客戶端連接到海王星數據庫實例 (p. 306)

.NET (C#)

以下是 C# 範例的一部分，它會為頂點建立一個名為 'date' 且 ID 為 '3' 的屬性。它將值設為使用 .NET *DateTime.UtcNow* 屬性產生的日期。

```
Using System;

g.V('3').property('date', DateTime.UtcNow).next()
```

如需使用 C# 連線至 Neptune 的完整範例，請參閱使用 .NET 連線至 Neptune 資料庫執行個體 (p. 314)

使用小鬼 Java 客戶端和 Neptune 的最佳實踐

請使用最新版本的 Apache TinkerPop Java 客戶端

如果可以的話，請始終使用您正在使用的引擎版本所支持的最新版本的 Apache TinkerPop Gremlin Java 客戶端。較新的版本包含許多錯誤修復，可以提高客戶端的穩定性，性能和可用性。

如阿帕奇 TinkerPop Java 小鬼客戶端 (p. 305)需與各種 Neptune 引擎版本相容的用戶端版本清單，請參閱。

跨多個線程重新使用客戶端對象

跨多個執行緒重複使用相同的用戶端 (或 GraphTraversalSource) 物件。也就是說，在您的應用程式中建立 org.apache.tinkerpop.gremlin.driver.Client 類別的共用執行個體，而不是在每個執行緒中建立。Client 物件是安全執行緒，初始化執行緒的額外負荷很大。

這也適用於 GraphTraversalSource，這會在內部建立 Client 物件。例如，以下程式碼會執行個體化新的 Client 物件：

```
import static
 org.apache.tinkerpop.gremlin.process.traversal.AnonymousTraversalSource.traversal;

  /////

GraphTraversalSource traversal = traversal()
                                    .withRemote(DriverRemoteConnection.using(cluster));
```

為讀寫端點創建單獨的 Java 客戶端對象

只在寫入器端點執行寫入，以及只從一個或多個唯讀端點讀取，可以提高效能。

```
Client readerClient = Cluster.build("https://reader-endpoint")
          ...
          .connect()

Client writerClient = Cluster.build("https://writer-endpoint")
          ...
          .connect()
```

將多個僅供讀取複本端點新增至 Gemlin Java 連線集區

建立 Gremlin Java `Cluster` 物件時，可以使用 `.addContactPoint()` 方法將多個僅供讀取複本執行個體新增到連線集區的接觸點。

```
Cluster.Builder readerBuilder = Cluster.build()
          .port(8182)
          .minConnectionPoolSize(…)
          .maxConnectionPoolSize(…)
          ……
          .addContactPoint("reader-endpoint-1")
          .addContactPoint("reader-endpoint-2")
```

關閉用戶端以避免連線限制

完成用戶端後，請務必關閉用戶端，以確保伺服器關閉 WebSocket 連線，並釋放與連線相關聯的所有資源。如果您使用 `Cluster.close( )` 關閉叢集，此程序會自動發生，因為會隨後在內部呼叫 `client.close( )`。

如果用戶端未正確關閉，Neptune 會在 20 到 25 分鐘後終止所有閒置 WebSocket連線。不過，如果您在完成 WebSocket連線時並未明確關閉連線，且即時連線數目達到WebSocket 並行連線限制 (p. 846)，則會拒絕其他連線，並顯示 HTTP429 錯誤碼。此時，您必須重新啟動 Neptune 執行個體以關閉連線。

調用的建議`cluster.close()`不適用於 JavaAWS Lambda 函數。如需詳細資訊，請參閱 管理AWS Lambda函數中的小鬼 WebSocket 連接 (p. 638)。

容錯移轉後建立新連線

如果發生容錯移轉，因為叢集 DNS 名稱已解析為 IP 地址，所以 Gremlin 驅動程式可能會繼續連線到舊的寫入器。如果發生這種情況，您可以在容錯移轉後建立新的 `Client` 物件。

設置maxInProcessPerConnection和maxSimultaneousUsa同的值

`maxInProcessPerConnection`和`maxSimultaneousUsagePerConnection`參數都與您可以在單一連線上提交的最大同時查詢數量有 WebSocket關。在內部，這些參數是相互關聯的，修改一個參數而不修改另一個，可能導致用戶端在嘗試從用戶端連線集區擷取連線時收到逾時。

我們建議保持預設的最小處理中和同時使用量值，並將 `maxInProcessPerConnection` 和 `maxSimultaneousUsagePerConnection` 設定為相同的值。

設定這些參數的值是查詢複雜性和資料模型的函數。在傳回許多資料的查詢使用案例中，每個查詢需要更多連線頻寬，因此，參數應該具有更低的值，而 `maxConnectionPoolSize` 應該具有更高的值。

相反地，如果查詢傳回較少的資料量，則 `maxInProcessPerConnection` 和 `maxSimultaneousUsagePerConnection` 應設定為大於 `maxConnectionPoolSize` 的值。

將查詢作為字節碼發送到服務器，而不是作為字符串

在以下情況提交查詢時，使用位元碼來取代字串具有優勢：

- 早期擷取無效的查詢語法：使用位元碼變體可讓您在編譯階段偵測無效的查詢語法。如果您使用字串變體，直到查詢送至伺服器之前都無法發現無效的語法，此時會傳回錯誤。

- 避免基於字符串的性能損失：任何基於字符串的查詢提交（無論您使用 WebSockets 還是 HTTP）都會導致分離的頂點，這意味著 Vertex 對象包含 ID，Label 以及與頂點關聯的所有屬性（請參閱元素的屬性）。

  如果不需要屬性，這可能會導致伺服器上不必要的計算。例如，如果客戶想要使用查詢 `g.V("hakuna#1")` 取得 ID 為「hakuna#1」的頂點。如果查詢以字串提交傳送出去，伺服器需花時間擷取 ID、標籤以及與此頂點關聯的所有屬性。如果查詢以位元碼提交傳送出去，伺服器只需花費擷取頂點 ID 和標籤的時間。

換言之，與其提交這種查詢：

```
 final Cluster cluster = Cluster.build("localhost")
                                .port(8182)
                                .maxInProcessPerConnection(32)
                                .maxSimultaneousUsagePerConnection(32)
                                .serializer(Serializers.GRAPHBINARY_V1D0)
                                .create();

 try {
     final Client client = cluster.connect();
     List<Result> results =
client.submit("g.V().has('name','pumba').out('friendOf').id()").all().get();
     System.out.println(verticesWithNamePumba);
 } finally {
     cluster.close();
 }
```

不如使用位元碼提交查詢，如下所示：

```
 final Cluster cluster = Cluster.build("localhost")
                                .port(8182)
                                .maxInProcessPerConnection(32)
                                .maxSimultaneousUsagePerConnection(32)
                                .serializer(Serializers.GRAPHBINARY_V1D0)
                                .create();

 try {
     final GraphTraversalSource g =
traversal().withRemote(DriverRemoteConnection.using(cluster));
     List<Object> verticesWithNamePumba = g.V().has("name",
"pumba").out("friendOf").id().toList();
     System.out.println(verticesWithNamePumba);
 } finally {
     cluster.close();
 }
```

始終完全消耗查詢返回的 ResultSet 或迭代器

用戶端物件應一律完全耗用 `ResultSet` (若是字串提交) 或 `GraphTraversal` 傳回的反覆運算器。如果查詢結果未完全耗用，伺服器保留它們，直到用戶端完成耗用它們為止。

如果您的應用程式只需要一部分結果，您可以使用 `limit(X)` 步驟加上您的查詢，來限制伺服器產生的結果數量。

批量添加頂點和邊

除非您使用工作階段，否則對 Neptune DB 的每個查詢都會在單一交易的範圍內執行。這表示如果您需要使用 gremlin 查詢插入許多資料，使用 50-100 的批次大小來同時批次處理，可透過降低建立的負載交易量來改善效能。

舉例來說，新增 5 個頂點到資料庫會如下所示：

```
// Create a GraphTraversalSource for the remote connection
final GraphTraversalSource g =
 traversal().withRemote(DriverRemoteConnection.using(cluster));
// Add 5 vertices in a single query
g.addV("Person").property(T.id, "P1")
 .addV("Person").property(T.id, "P2")
 .addV("Person").property(T.id, "P3")
 .addV("Person").property(T.id, "P4")
 .addV("Person").property(T.id, "P5").iterate();
```

在 Java 虛擬機器中停用 DNS 快取

在想要負載平衡多個僅供讀取複本請求的環境中，您需要停用 Java 虛擬機器 (JVM) 中的 DNS 快取並在建立叢集時提供 Neptune 的讀取器端點。停用 JVM DNS 快取可確保每次新連線都會再次解析 DNS，如此一來，就會將請求分配到所有僅供讀取複本。您可以使用以下行在應用程序的初始化代碼中執行此操作：

```
java.security.Security.setProperty("networkaddress.cache.ttl", "0");
```

但是，亞馬遜 Gemlin Java 客戶端代碼提供了一個更完整和強大的負載平衡解決方案 GitHub。Amazon Java Gemlin 用戶端知道您的叢集拓撲，並公平地將連線和請求分配到 Neptune 叢集中的一組執行個體。如需使用該用戶端的 Java Lambda 函數範例，請參閱此部落格文章。

選擇性地在每個查詢層級設定逾時

Neptune 可讓您使用參數群組選項設定查詢逾時neptune_query_timeout (請參閱參數 (p. 530))。不過從 Java 用戶端版本 3.3.7 開始，您也可以使用類似下列的程式碼，覆寫全域逾時：

```
 final Cluster cluster = Cluster.build("localhost")
                                .port(8182)
                                .maxInProcessPerConnection(32)
                                .maxSimultaneousUsagePerConnection(32)
                                .serializer(Serializers.GRAPHBINARY_V1D0)
                                .create();

 try {
     final GraphTraversalSource g =
traversal().withRemote(DriverRemoteConnection.using(cluster));
     List<Object> verticesWithNamePumba = g.with(ARGS_EVAL_TIMEOUT, 500L).V().has("name",
"pumba").out("friendOf").id().toList();
     System.out.println(verticesWithNamePumba);
 } finally {
     cluster.close();
 }
```

或者，對於字串查詢提交，程式碼將如下所示：

```
 RequestOptions options = RequestOptions.build().timeout(500).create();
 List<Result> result = client.submit("g.V()", options).all().get();
```

> **Note**
>
> 如果您將查詢逾時值設定得太高，特別是在無伺服器執行個體上，可能會產生非預期的費用。如果沒有合理的逾時設定，您的查詢執行時間可能會比預期的長得多，產生您從未預期的成本。這在無伺服器執行個體上尤其如此，該執行個體在執行查詢時可能會擴展成大型且昂貴的執行個體類型。

您可以使用符合您期望的執行階段的查詢逾時值，避免此類非預期的費用，而且只會導致異常的長時間執行逾時。

解決 3.3.4 以前的客戶端中保持活動的錯誤

僅適用於 3.3.3 及以下的用戶端版本：

舊版 Gremlin 用戶端中有一個錯誤，會在每次查詢傳送新的 KeepAlive 請求到伺服器，而不是每個 websocket 連線一次。這會耗用寶貴的伺服器資源，因為伺服器處理不必要的持續作用請求。如需詳細資訊，請參閱 TINKERPOP-2030。

您可以透過升級至 Gremlin Java 用戶端版本 3.3.4 或更新版本來避免這個問題。

從用戶端停用持續作用的解決方法是將其間隔設定為 0：

```
Cluster.Builder readerBuilder = Cluster.build()
          .port(8182)
          ...
          .keepAliveInterval(0)
```

排除「小鬼 Java 客戶端」異常問題

Gremlin Java 客戶端拋出特定於其操作的異常。本節提供額外資訊以協助您進行故障排除。

java.util.concurrent.TimeoutException 疑難排解

Gemlin Java 客戶端在等待其中一個 WebSocket 連接中的插槽變為可用`java.util.concurrent.TimeoutException`時，Gimlin Java 客戶端會拋出一個 Gimlin 請求超時。此逾時持續時間由`maxWaitForConnection`用戶端可設定參數控制。

> Note
>
> 由於用戶端逾時的要求永遠不會傳送至伺服器，因此它們不會反映在伺服器擷取的任何量度中，例如`GremlinRequestsPerSec`。

此類型的逾時通常透過下列兩種方式造成：

- 伺服器實際上已達到最大容量。如果是這種情況，伺服器上的佇列就會填滿，您可以透過監視指標來偵MainRequestQueuePendingRequests (p. 775) CloudWatch 測該佇列。伺服器可以處理的 parallel 查詢數目取決於其執行個體大小。

  如果`MainRequestQueuePendingRequests`指標未在伺服器上顯示擱置要求的累積，則伺服器可以處理更多要求，而且逾時是由用戶端節流造成的。
- 用戶端節流的要求。通常可以透過變更用戶端組態設定來修正此問題。

  用戶端可以傳送 parallel 請求數量上限：

  ```
  maxParallelQueries = maxConnectionPoolSize * Max( maxSimultaneousUsagePerConnection,
   maxInProcessPerConnection )
  ```

  傳送超過`maxParallelQueries`用戶端會導致`java.util.concurrent.TimeoutException`例外狀況。您通常可以透過下列幾種方式解決此問題：
  - 增加連線逾時持續時間。如果延遲對您的應用程式而言並不重要，請增加用戶端的`maxWaitForConnection`設定。然後，用戶端會等待更長時間才逾時，進而增加延遲。
  - 增加每個連線的要求上限。這允許使用相同的 WebSocket 連接發送更多請求。透過增加用戶端`maxSimultaneousUsagePerConnection`和`maxInProcessPerConnection`設定來執行此操作。這些設定通常應該有相同的值。

- 增加連線集區中的連線數目。透過增加用戶端的`maxConnectionPoolSize`設定來執行此操作。成本會增加資源消耗，因為每個連線都使用記憶體和作業系統檔案描述元，並且在初始化期間需要 SSL 和 WebSocket交握。

使用開放密碼和螺栓的 Neptune 最佳實踐

使用 OpenCypher Cypher Cypher 查詢語言和 Bolt 如需有關在 Neptune 中使用 OpenCypher 的資訊，請參閱。使用開放密碼訪問海王星圖 (p. 382)

偏好定向於查詢中的雙向邊緣

Neptune 執行查詢最佳化時，雙向邊緣會使建立最佳化查詢計劃變得困難。次優計劃要求引擎執行不必要的工作並導致性能較差。

因此，盡可能使用有向邊而不是雙向邊緣。例如，使用：

```
MATCH p=(:airport {code: 'ANC'})-[:route]->(d) RETURN p)
```

而不是：

```
MATCH p=(:airport {code: 'ANC'})-[:route]-(d) RETURN p)
```

大多數數據模型實際上不需要遍歷兩個方向的邊緣，因此查詢可以通過切換到使用定向邊來實現顯著的性能改進。

如果您的數據模型確實需要遍歷雙向邊緣，則使模`MATCH`式中的第一個節點（左側）成為具有最嚴格篩選的節點。

以「找到我所有的`routes`往返ANC機場」的例子。編寫此查詢以在ANC機場開始，如下所示：

```
MATCH p=(src:airport {code: 'ANC'})-[:route]-(d) RETURN p
```

引擎可以執行最少的工作量來滿足查詢，因為受限制最多的節點是放置在模式中的第一個節點（左側）。然後，引擎可以最佳化查詢。

這比在模式結束時過濾ANC機場更可取，如下所示：

```
MATCH p=(d)-[:route]-(src:airport {code: 'ANC'}) RETURN p
```

當最受限制的節點未放在模式中的第一位時，引擎必須執行額外的工作，因為它無法最佳化查詢，而且必須執行其他查詢才能到達結果。

Neptune 不支持事務中的多個並發查詢

雖然 Bolt 驅動程式本身允許在交易中同時進行查詢，但 Neptune 不支援同時執行的交易中的多個查詢。Neptune 會改為要求在交易中依序執行多個查詢，並且在啟動下一個查詢之前，完全使用每個查詢的結果。

以下範例說明如何使用 Bolt 在交易中依序執行多個查詢，以便在下一個查詢開始之前完全消耗每個查詢的結果：

```
final String query = "MATCH (n) RETURN n";
```

```
try (Driver driver = getDriver(HOST_BOLT, getDefaultConfig())) {
  try (Session session = driver.session(readSessionConfig)) {
    try (Transaction trx = session.beginTransaction()) {
      final Result res_1 = trx.run(query);
      Assert.assertEquals(10000, res_1.list().size());
      final Result res_2 = trx.run(query);
      Assert.assertEquals(10000, res_2.list().size());
    }
  }
}
```

容錯移轉後建立新連線

如果發生容錯移轉，Bolt 驅動程式可以繼續連線至舊的寫入器執行個體，而不是新的使用中執行個體，因為 DNS 名稱已解析為特定 IP 位址。

若要避免這種情況發生，請在任何容錯移轉後關閉`Driver`物件，然後再重

長壽命應用的連接處理

建置長期使用壽命的應用程式 (例如在容器或 Amazon EC2 執行個體上執行的應用程式) 時，請將`Driver`物件實例化一次，然後在應用程式的生命週期內重複使用該物件。`Driver` 物件是安全執行緒，初始化執行緒的額外負荷很大。

連線處理 AWS Lambda

由於螺栓驅動器的連接開銷和管理需求，因此不建議在AWS Lambda功能中使用螺栓驅動器。請改用 HTTPS 端點 (p. 386)。

完成後關閉驅動程序對象

請務必在完成用戶端後關閉用戶端，以便伺服器關閉 Bolt 連線，並釋放與連線相關聯的所有資源。如果您使用關閉驅動程序，則會自動發生此`driver.close()`情況

如果驅動程式未正確關閉，Neptune 會在 20 分鐘後終止所有閒置的 Bolt 連線，如果您使用 IAM 驗證，則會在 10 天後終止。

Neptune 支持不超過 1000 個並發螺栓連接。如果您在完成連線時未明確關閉連線，且即時連線數目達到 1000 個限制，則任何新的連線嘗試都會失敗。

使用明確的交易模式進行讀取和寫入

使用 Neptune 和 Bolt 驅動程式的交易時，最好將讀取和寫入交易的存取模式明確設定為正確的設定。

唯讀交易

對於唯讀交易，如果您在建置工作階段時未傳遞適當的存取模式設定，則會使用預設的隔離層級，也就是突變查詢隔離。因此，對於唯讀交易來說，將存取模式設定為`read`明確是很重要的。

自動提交讀取事務示例：

```
SessionConfig sessionConfig = SessionConfig
  .builder()
  .withFetchSize(1000)
```

```
  .withDefaultAccessMode(AccessMode.READ)
  .build();
Session session = driver.session(sessionConfig);
try {
  (Add your application code here)
} catch (final Exception e) {
  throw e;
} finally {
  driver.close()
}
```

讀取交易範例：

```
Driver driver = GraphDatabase.driver(url, auth, config);
SessionConfig sessionConfig = SessionConfig
  .builder()
  .withDefaultAccessMode(AccessMode.READ)
  .build();
driver.session(sessionConfig).readTransaction(
  new TransactionWork<List<String>>() {
    @Override
    public List<String> execute(org.neo4j.driver.Transaction tx) {
      (Add your application code here)
    }
  }
);
```

在這兩種情況下，SNAPSHOT隔離 (p. 109)都是使用 Neptune 唯讀事務語義 (p. 110)來實現的。

由於僅供讀取複本只接受唯讀查詢，所以提交至僅供讀取複本的任何查詢都會在SNAPSHOT隔離語意下執行。

只讀事務沒有臟讀或不可重複讀取。

唯讀交易

對於突變查詢，有三種不同的機制來創建一個寫入事務，每個方法如下所示：

隱式寫入事務示例：

```
Driver driver = GraphDatabase.driver(url, auth, config);
SessionConfig sessionConfig = SessionConfig
  .builder()
  .withDefaultAccessMode(AccessMode.WRITE)
  .build();
driver.session(sessionConfig).writeTransaction(
  new TransactionWork<List<String>>() {
    @Override
    public List<String> execute(org.neo4j.driver.Transaction tx) {
      (Add your application code here)
    }
  }
);
```

自動提交寫入事務示例：

```
SessionConfig sessionConfig = SessionConfig
  .builder()
  .withFetchSize(1000)
  .withDefaultAccessMode(AccessMode.Write)
```

```
  .build();
Session session = driver.session(sessionConfig);
try {
  (Add your application code here)
} catch (final Exception e) {
    throw e;
} finally {
    driver.close()
}
```

明確寫入交易範例：

```
Driver driver = GraphDatabase.driver(url, auth, config);
SessionConfig sessionConfig = SessionConfig
  .builder()
  .withFetchSize(1000)
  .withDefaultAccessMode(AccessMode.WRITE)
  .build();
Transaction beginWriteTransaction = driver.session(sessionConfig).beginTransaction();
  (Add your application code here)
beginWriteTransaction.commit();
driver.close();
```

寫入交易的隔離層級

- 作為突變查詢的一部分進行的讀取會在READ COMMITTED事務隔離下運行。
- 作為突變查詢的一部分進行的讀取沒有骯髒的讀取。
- 在突變查詢中讀取時，記錄和記錄範圍被鎖定。
- 當一個變異交易讀取了索引範圍時，有一個強有力的保證，這個範圍不會被任何並發事務修改，直到讀取結束。

突變查詢不是安全執行緒安全。

如需衝突，請參閱使用鎖定等待逾時的衝突解決機制 (p. 111)。

突變查詢不會在失敗的情況下自動重試。

例外狀況的重試邏輯

對於允許重試的所有例外狀況，通常最好使用指數輪詢和重試策略，該策略在重試之間提供逐漸更長的等待時間，以便更好地處理暫時性問題，例如錯誤。ConcurrentModificationException下方顯示指數輪詢和 Retry 模式的範例：

```
public static void main() {
  try (Driver driver = getDriver(HOST_BOLT, getDefaultConfig())) {
    retriableOperation(driver, "CREATE (n {prop:'1'})")
        .withRetries(5)
        .withExponentialBackoff(true)
        .maxWaitTimeInMilliSec(500)
        .call();
  }
}

protected RetryableWrapper retriableOperation(final Driver driver, final String query){
  return new RetryableWrapper<Void>() {
    @Override
    public Void submit() {
      log.info("Performing graph Operation in a retry manner......");
      try (Session session = driver.session(writeSessionConfig)) {
```

```
        try (Transaction trx =  session.beginTransaction()) {
            trx.run(query).consume();
            trx.commit();
        }
      }
      return null;
    }

    @Override
    public boolean isRetryable(Exception e) {
      if (isCME(e)) {
        log.debug("Retrying on exception.... {}", e);
        return true;
      }
      return false;
    }

    private boolean isCME(Exception ex) {
      return ex.getMessage().contains("Operation failed due to conflicting concurrent
 operations");
    }
  };
}



/**
 * Wrapper which can retry on certain condition. Client can retry operation using this
 class.
 */
@Log4j2
@Getter
public abstract class RetryableWrapper<T> {

  private long retries = 5;
  private long maxWaitTimeInSec = 1;
  private boolean exponentialBackoff = true;

  /**
   * Override the method with custom implementation, which will be called in retryable
 block.
   */
  public abstract T submit() throws Exception;

  /**
   * Override with custom logic, on which exception to retry with.
   */
  public abstract boolean isRetryable(final Exception e);

  /**
   * Define the number of retries.
   *
   * @param retries -no of retries.
   */
  public RetryableWrapper<T> withRetries(final long retries) {
    this.retries = retries;
    return this;
  }

  /**
   * Max wait time before making the next call.
   *
   * @param time - max polling interval.
   */
  public RetryableWrapper<T> maxWaitTimeInMilliSec(final long time) {
    this.maxWaitTimeInSec = time;
```

```
    return this;
  }

  /**
   * ExponentialBackoff coefficient.
   */
  public RetryableWrapper<T> withExponentialBackoff(final boolean expo) {
    this.exponentialBackoff = expo;
    return this;
  }

  /**
   * Call client method which is wrapped in submit method.
   */
  public T call() throws Exception {
    int count = 0;
    Exception exceptionForMitigationPurpose = null;
    do {
      final long waitTime = exponentialBackoff ? Math.min(getWaitTimeExp(retries),
 maxWaitTimeInSec) : 0;
      try {
          return submit();
      } catch (Exception e) {
        exceptionForMitigationPurpose = e;
        if (isRetryable(e) && count < retries) {
          Thread.sleep(waitTime);
          log.debug("Retrying on exception attempt - {} on exception cause - {}", count,
 e.getMessage());
        } else if (!isRetryable(e)) {
          log.error(e.getMessage());
          throw new RuntimeException(e);
        }
      }
    } while (++count < retries);

    throw new IOException(String.format(
          "Retry was unsuccessful.... attempts %d. Hence throwing exception " + "back to
 the caller...", count),
          exceptionForMitigationPurpose);
  }

  /*
   * Returns the next wait interval, in milliseconds, using an exponential backoff
   * algorithm.
   */
  private long getWaitTimeExp(final long retryCount) {
    if (0 == retryCount) {
      return 0;
    }
    return ((long) Math.pow(2, retryCount) * 100L);
  }
}
```

使用 SPARQL 的 Neptune 最佳做法

將 SPARQL 查詢問語言與 Neptune 一起使用 SPARQL 查詢問語言，請遵循 如需有關在 Neptune 中使用 SPARQL 的資訊，請參閱使用 SPARQL 存取海王星圖 (p. 422)。

預設查詢所有具名圖形

Amazon Neptune 將每三個與一個命名的圖形相關聯 預設圖形是定義成所有具名圖形的聯集。

若您提交 SPARQL 查詢而未透過 GRAPH 關鍵字或 FROM NAMED 之類的建構式明確指定圖形，Neptune 一律會考慮資料庫執行個體中的所有三元組。例如，下列查詢會傳回 Neptune SPARQL 端點的所有三元組：

```
SELECT * WHERE { ?s ?p ?o }
```

顯現於多個圖形中的三元組將僅傳回一次。

如需預設圖形規格的相關資訊，請參閱 SPARQL 1.1 查詢語言規格的 RDF 資料集一節。

指定要載入的具名圖表

Amazon Neptune 將每三個與一個命名的圖形相關聯 如果您在載入、插入或更新三元組時未指定具名圖形，Neptune 會使用 URI 定義的後援命名圖形`http://aws.amazon.com/neptune/vocab/v01/DefaultNamedGraph`。

如果您使用 Neptune 批量載入器，則可以使用`parserConfiguration: namedGraphUri`參數指定要用於所有三元組（或第四個位置為空白的四邊形）的具名圖形。如需有關 Neptune 載入程式`Load`命令語法的資訊，請參閱the section called “載入器命令” (p. 275)。

在您的查詢中選擇 FILTER、FILTER...IN 和 VALUES

有三種基本方式可在 SPARQL 查詢中注入值：`FILTER`、`FILTER...IN` 和 `VALUES`。

例如，假設您想要在單次查詢中，查詢多名人員的好友。使用 `FILTER`，您可以建構如下查詢：

```
PREFIX ex: <https://www.example.com/>
PREFIX foaf : <http://xmlns.com/foaf/0.1/>

SELECT ?s ?o
WHERE {?s foaf:knows ?o. FILTER (?s = ex:person1 || ?s = ex:person2)}
```

這會傳回將 `?s` 繫結至 `ex:person1` 或 `ex:person2`，且外寄節點標示為 `foaf:knows` 之圖形中的所有三元組。

您也可以使用傳回同等結果的 `FILTER...IN` 建立查詢：

```
PREFIX ex: <https://www.example.com/>
PREFIX foaf : <http://xmlns.com/foaf/0.1/>

SELECT ?s ?o
WHERE {?s foaf:knows ?o. FILTER (?s IN (ex:person1, ex:person2))}
```

您也可以使用在這種情況下也會傳回同等結果的 `VALUES` 建立查詢：

```
PREFIX ex: <https://www.example.com/>
PREFIX foaf : <http://xmlns.com/foaf/0.1/>

SELECT ?s ?o
WHERE {?s foaf:knows ?o. VALUES ?s {ex:person1 ex:person2}}
```

雖然在許多情況下，這些查詢在語意上是相等的，但在某些情況下，兩個 `FILTER` 變體和 `VALUES` 變體不同：

- 第一種情況是當您注入重複值時，例如注入相同的人員兩次。在這種情況下，`VALUES` 查詢會在結果中包含重複項目。您可以將 `DISTINCT` 新增至 `SELECT` 子句，明確消除這類重複。但有時候，您可能真的需要在查詢結果中出現重複項目，以注入多餘值。

不過，當相同的值出現多次時，FILTER 和 FILTER...IN 版本只擷取一次值。

- 第二種情況與 VALUES 一律執行完全符合，而 FILTER 可能會套用類型提升，並在某些情況下執行模糊映射有關。

  例如，當您在值子句中包含 "2.0"^^xsd:float 等常值時，VALUES 查詢會尋找與此常值完全符合的項目，包括常值的值和資料類型。

  相反地，FILTER 則會產生與這些數值常值模糊符合的項目。符合項目可能包括值相同、但數值資料類型不同的常值，例如 xsd:double。

  Note

  列舉字串常值或 URI 時，FILTER 和 VALUES 的行為間無差異。

FILTER 和 VALUES 之間的差異會影響最佳化和產生的查詢評估策略。除非您的使用案例需要模糊相符，否則建議您使用 VALUES，因為它會避免查看有關類型轉換的特殊案例。因此，VALUES 通常會產生更有效率的查詢，執行速度更快但價格低廉。

Amazon Neptune 限制

區域

亞馬遜海王星在以下AWS區域提供：

- 美國東部 (維吉尼亞北部): us-east-1
- 美國東部 (俄亥俄州): us-east-2
- 美國西部 (加利佛尼亞北部): us-west-1
- 美國西部 (奧勒岡): us-west-2
- 加拿大 (中部): ca-central-1
- 南美洲（聖保羅）：sa-east-1
- 歐洲 (斯德哥爾摩): eu-north-1
- 歐洲 (愛爾蘭): eu-west-1
- 歐洲 (倫敦): eu-west-2
- 歐洲（巴黎）：eu-west-3
- 歐洲 (法蘭克福): eu-central-1
- 中東（巴林）：me-south-1
- 中東（阿聯酋）：me-central-1
- 非洲（開普敦）：af-south-1
- 亞太區域 (香港): ap-east-1
- 亞太區域 (東京): ap-northeast-1
- 亞太區域 (首爾): ap-northeast-2
- 亞太區域 (新加坡)：ap-southeast-1
- 亞太區域 (雪梨)：ap-southeast-2
- 亞太區域 (孟買)：ap-south-1
- 中國 (北京): cn-north-1
- 中國 (寧夏): cn-northwest-1
- AWSGovCloud(美國西部): us-gov-west-1
- AWSGovCloud(美國東部): us-gov-east-1

儲存叢集磁碟區的大小上限

除了中國以外的所有支援區域，Neptune 叢集磁碟區的大小上限為 128 TB (TiB)GovCloud，限制為 64 TiB。開頭的所有引擎版本都是如此發行版本： (p. 960)。請參閱 Amazon 海王星儲存、可靠性和可用性 (p. 119)。

中國地區差異

與許多AWS服務一樣，Amazon Neptune 在中國（北京）和中國（寧夏）的運營方式與其他AWS地區略有不同。

例如，當 Neptune ML 使用 Amazon API 閘道建立其匯出服務時，依預設會啟用 IAM 身份驗證。在中國地區，改變該選項的過程與其他地區略有不同。

這些和其他差異在這裡解釋。

資料庫執行個體大小

Neptune 支援不同AWS區域的不同資料庫執行個體類別 若要了解特定區域中支援的類別，請參閱 Amazon Neptune 定價，然後選擇您有興趣的區域。

每個AWS帳戶的限制

對於某些管理功能，Amazon Neptune 會使用與 Amazon 關聯式資料庫服務 (Amazon RDS) 共用的操作技術。

每個AWS帳戶對於您可以建立的 Amazon 海王星和 Amazon RDS 資源數量，每個區域都有限制。這些資源包括資料庫執行個體和資料庫叢集。

在您到達某項資源的上限時，所發出用來建立該資源的額外呼叫都會伴隨著異常而失敗。

如需亞馬遜海王星和亞馬遜 RDS 之間共用的限制清單，請參閱亞馬遜 RDS 使用者指南中的限制。

連接到海王星需要 VPC

亞馬遜海王星是僅限虛擬私有雲 (VPC) 的服務。

此外，該服務不允許從 VPC 外部來存取執行個體。

海王星需要 SSL

從引擎版本開始`1.0.4.0`，Amazon Neptune 只允許透過 HTTPS 連線到任何執行個體或叢集端點的安全通訊端層 (SSL)。

海王星需要使用以下強式密碼套件的 TLS 1.2 版：

- `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
- `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`
- `TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA`
- `TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA`

可用區域和資料庫子網路群組

對於在至少兩個受支援的可用區域 (AZ) 中具有子網路的每個叢集，Amazon Neptune 需要一個資料庫子網路群組。

建議您在不同的可用區域中使用三個或多個子網路。

HTTP 要求承載最大值 (150 MB)

Gremlin 與 SPARQL HTTP 請求的總大小必須小於 150 MB。如果要求超過此大小，Neptune 會傳回HTTP `400: BadRequestException`。

此限制不適用於小鬼連線WebSockets。

小精靈實現差異

Amazon 海王星 Gemlin 實作具有特定的實作詳細資料，可能與其他 Gemlin 實作不同。

如需詳細資訊，請參閱亞馬遜海王星的小精靈標準合規 (p. 83)。

海王星不支持字符串數據中的空字符

海王星不支持字符串中的空字符。這在屬性圖資料中適用於小精靈和開放密碼，以及 RDF/SPARQL 資料。

從 URI 進行 SPARQL UPDATE LOAD

來自 URI 的 `SPARQL UPDATE LOAD` 僅適用於相同 VPC 中的資源。

這包括位於與已建立 Amazon S3 VPC 端點的叢集位於相同區域的 Amazon S3 URL。

亞馬遜 S3 網址必須是 HTTPS，且任何身份驗證都必須包含在該網址中。如需詳細資訊，請參閱 Amazon 簡單儲存服務 API 參考中的驗證請求：使用查詢參數。

如需有關建立 VPC 端點的資訊，請參閱 建立亞馬遜 S3 VPC 端點 (p. 270)。

如果您需要從檔案載入資料，建議您使用 Amazon 海王星載入器 API。如需詳細資訊，請參閱使用 Amazon 海王星大容量載入器擷取資料 (p. 255)。

> Note
>
> 亞馬遜海王星加載器 API 是非酸性的。

IAM 身份驗證和存取控制

在 1.2.0.0 (p. 884) 版之前的 Neptune 引擎版本中，僅在資料庫叢集層級支援 IAM 身份驗證和存取控制。不過，從版本`1.2.0.0`開始，您可以使用 IAM 政策中的條件金鑰，在更精細的層級控制以查詢為基礎的存取。如需詳細資訊，請參閱「使用 Neptune 資料存取原則陳述式的查詢動作 (p. 203)」與「亞馬遜海王星 AWS Identity and Access Management（IAM）概述 (p. 157)」

亞馬遜海王星主控台需要NeptuneReadOnlyAccess許可。您可以透過撤銷此存取權限來限制 IAM 使用者的存取權。如需詳細資訊，請參閱 AWS亞馬遜海王星的受管 (預先定義) 政策 (p. 179)

Amazon Neptune 不支援以使用者名稱/密碼為基礎的存取控制。

WebSocket同時連接和最長連接時間

每個 Neptune 資料庫執行個體的同時WebSocket連線數目有限制。達到該限制時，Neptune 會限制任何請求以打開新WebSocket連接，以防止耗盡所有分配的堆積內存。

對於 Neptune 和所有無伺服器執行個體支援的所有較大型執行個體類型，WebSocket連線的並行數目上限為 32K (32,768)。

下表列出較小執行個體類型的最大同時WebSocket連線數目：

執行個體類型	最大同時WebSocket連線
db.t3.medium	512
db.t4g.medium	512
db.r5.large	2,048
db.r5d.large	2,048
db.r5.xlarge	4,096
db.r5.2xlarge	8,192
db.r5d.2xlarge	8,192
db.r5.4xlarge	16,384
db.r5d.4xlarge	16,384
db.r6g.large	2,048
db.r6gd.large	2,048
db.r6g.xlarge	4,096
db.r6gd.xlarge	4,096
db.r6g.2xlarge	8,192
db.r6gd.2xlarge	8,192
db.r6g.4xlarge	16,384
db.r6gd.4xlarge	16,384
db.x2g.large	2,048
大數據庫	2,048
db.x2g.xlarge	4,096
資料庫 .x2 公分大	4,096
db.x2iedn.xlarge	4,096
db.x2g.2xlarge	8,192
資料庫 .x2 公分大	8,192
db.x2g.4xlarge	16,384

執行個體類型	最大同時WebSocket連線
資料庫 .x2 公分大	16,384
db.x2iedn.2xlarge	16,384
db.x2iezn.2xlarge	16,384
serverless	32,768
(其他大型執行個體類型)	32,768

Note

從海王星引擎版本 1.1.0.0 (p. 915) 開始，Neptune 不再支援R4執行個體類型。

當用戶端以正常方式關閉連線時，該關閉會立即反映在開啟的連線數量上。

如果用戶端未關閉連線，連線可能會在 20 到 25 分鐘的閒置逾時後自動關閉 (閒置逾時是自從用戶端收到上一則訊息後所經過的時間)。但是，只要未達到閒置逾時，Neptune 就會無限期地保持連線開啟。

啟用 IAM 身份驗證後，如果WebSocket連線尚未關閉，則連線一律會在建立 10 天後的幾分鐘內中斷連線。

性質和標示的限制

您在一個圖形中可擁有無限個頂點和邊緣或 RDF quad。

任何一個頂點或邊緣也可以擁有無限個屬性或標籤。

個別屬性或標籤的大小上限為 55 MB。在 RDF 術語中，這意味著 RDF quad 的任何資料欄 (S、P、O 或 G) 中的值都不能超過 55 MB。

如果需要將較大的物件 (例如映像) 與圖形中的頂點或節點建立關聯，則可以將其作為檔案存放在 Amazon S3 中，並使用 Amazon S3 路徑作為屬性或標籤。

影響海王星散裝載機的限制

您一次無法將超過 64 個 Neptune 大量載入工作排入佇列。

海王星僅跟踪最近的 1,024 個批量加載作業。

海王星僅存儲每個作業的最後 10,000 個錯誤詳細信息。

使用其他 AWS 服務

您可以將 Amazon Neptune 與許多其他AWS服務搭配使用：

將 Neptune 其服務務務務務務務

- AWS Glue—AWS Glue 是一項無伺服務，可在資料上執行擷取、轉換和載入 (ETL) 任務。

  Neptune 提供了一個開源庫 neptune-python-utilities，可以簡化在 Glue 作業中使用 Python 和小靈的操作。該 Neo4j 的星火連接器也支持運行斯卡拉和開放密碼 Glue 作業。
- Amazon SageMaker — Amazon SageMaker 是功能齊全的機器學習平台，可用於建置、訓練和部署高品質的機器學習模型。

  Neptune 以兩 SageMaker 種主要方式整合：
  - Neptune 為 Jupyter 筆記本提供了一個開源 Python 包，可以在 Neptune 圖形筆記本項目上找到 GitHub。此軟件包包含一組 Jupyter 魔術，教程筆記本和代碼示例，這些示例在交互式編碼環境中提供，您可以在其中了解圖形技術和 Neptune。Neptune 為主控的 Jupyter 筆記本提供完全受控的環境 SageMaker，並自動連結至開放原始碼 Neptune 圖形筆記本專案中的筆記本。
  - Neptune ML 功能可讓您在數小時而不需要數週的時間內，在大型圖表上建置和訓練有用的機器學習模型。為了實現這一目標，Neptune ML 使用由亞馬遜 SageMaker 和深圖庫（DGL）提供支持的圖形神經網絡（GNN）技術。
- AWS Lambda—AWS Lambda 功能在 Neptune 應用中有很多用途。

  如需有關如何將 Lambda 函數與任何常用的 Gremlin 驅動程式和語言變體搭配使用，以及以 Java、和 Python 撰寫之 Lambda 函數的特定範例的詳細資訊，請參閱使用 Amazon Neptune 的AWS Lambda功能 (p. 638)。 JavaScript
- Amazon Athena — Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon Simple Storage Service 中的資料。

  Neptune 提供連接至 Athena 的連接器，可讓 Athena 與儲存在 Neptune 中的資料進行通訊。
- AWS Database Migration Service（AWS DMS）-AWS Database Migration Service 是一種AWS Web 服務，您可以使用將數據從一個數據庫遷移到另一個數據庫。

  AWS DMS可以快速安全地將資料從支援的來源資料庫載入 Neptune (p. 292)。來源資料庫在遷移期間仍然能夠維持完全正常運作，讓倚賴它的應用程式可以將停機時間縮到最短。
- AWS Backup—AWS Backup 是一項全受管的備份服務，可讓您在雲端及內部部部

Neptune 錯誤

Amazon Neptune 有兩組不同的錯誤：

- 僅適用於 Neptune 資料庫叢集端點的圖形引擎錯誤。
- 與使用AWS SDK 和AWS Command Line Interface (AWS CLI) 建立和修改 Neptune 資源的 API 相關聯的錯誤。

主題

- 圖形引擎錯誤訊息和代碼 (p. 849)
- 資料庫叢集管理 API 錯誤訊息和代碼 (p. 853)
- Neptune 載入器的錯誤和饋送訊息 (p. 854)

圖形引擎錯誤訊息和代碼

遇到時，Amazon Neptune 端點會傳回小鬼和 SPARQL 的標準錯誤。

Neptune 特有的錯誤也可以從相同的端點傳回。本節記錄 Neptune 錯誤訊息、代碼和建議的動作。

> Note
>
> 這些錯誤僅適用於 Neptune 資料庫叢集端點。使用AWS SDK 建立和修改 Neptune 資源的 API，並具AWS CLI有不同的常見錯誤集。如需這些錯誤的資訊，請參閱the section called “API 錯誤” (p. 853)。

圖形引擎錯誤格式

Neptune 錯誤消息返回相關的 HTTP 錯誤代碼和 JSON 格式的響應。

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: LDM6CJP8RMQ1FHKSC1RBVJFPNVV4KQNSO5AEMF66Q9ASUAAJG
Content-Type: application/x-amz-json-1.0
Content-Length: 465
Date: Thu, 15 Mar 2017 23:56:23 GMT

{
    "requestId": "0dbcded3-a9a1-4a25-b419-828c46342e47",
    "code": "ReadOnlyViolationException",
    "detailedMessage": "The request is rejected because it violates some read-only
 restriction, such as a designation of a replica as read-only."
}
```

圖形引擎查詢錯誤

下表包含錯誤代碼、訊息和 HTTP 狀態。

也會指出是否可以重試請求。一般而言，如果新嘗試可能成功，就可以重試請求。

Neptune Service Error Code	HTTP status	Ok to Retry?	Message
AccessDeniedException	403	No	Authentication or authorization failure.
BadRequestException	400	No	The request could not be completed.
BadRequestException	400	No	Request size exceeds max allowed value of 157286400 bytes.
CancelledByUserException	500	Yes	The request processing was cancelled by an authorized client.
ConcurrentModificationException	500	Yes	The request processing did not succeed due to a modification conflict. The client should retry the request.
ConstraintViolationException	400	Yes	The query engine discovered, during the execution of the request, that the completion of some operation is impossible without violating some data integrity constraints, such as persistence of in- and out-vertices while adding an edge. Such conditions are typically observed if there are concurrent modifications to the graph, and are transient. The client should retry the request.
InternalFailureException	500	Yes	The request processing has failed.
InvalidNumericDataException	400	No	Invalid use of numeric data which cannot be represented in 64-bit storage size.
InvalidParameterException	400	No	An invalid or out-of-range value was supplied for some input parameter or invalid syntax in a supplied RDF file.

Neptune Service Error Code	HTTP status	Ok to Retry?	Message
MalformedQueryException	400	No	The request is rejected because it contains a query that is syntactically incorrect or does not pass additional validation.
MemoryLimitExceededException	500	Yes	The request processing did not succeed due to lack of memory, but can be retried when the server is less busy.
MethodNotAllowedException	405	No	The request is rejected because the chosen HTTP method is not supported by the used endpoint.
MissingParameterException	400	No	A required parameter for the specified action is not supplied.
QueryLimitExceededException	500	Yes	The request processing did not succeed due to the lack of a limited resource, but can be retried when the server is less busy.
QueryLimitException	400	No	Size of query exceeds system limit.
QueryTooLargeException	400	No	The request was rejected because its body is too large.
ReadOnlyViolationException	400	No	The request is rejected because it violates some read-only restriction, such as a designation of a replica as read-only.
ThrottlingException	500	Yes	Rate of requests exceeds the maximum throughput. OK to retry.
TimeLimitExceededException	500	Yes	The request processing timed out.
TooManyRequestsException	429	Yes	The rate of requests exceeds the maximum throughput. OK to retry. 只有在啟用 IAM 身份驗證時才會發生此例外狀況。

Neptune Service Error Code	HTTP status	Ok to Retry?	Message
`UnsupportedOperationException`	400	No	The request uses a currently unsupported feature or construct.

IAM 身分驗證錯誤

這些錯誤只發生在有啟用 IAM 身分驗證的叢集。

下表包含錯誤代碼、訊息和 HTTP 狀態。

Neptune Service Error Code	HTTP status	Message
Incorrect IAM User/Policy	403	You do not have sufficient access to perform this action.
Incorrect or Missing Region	403	Credential should be scoped to a valid Region, not '*region*'.
Incorrect or Missing Service Name	403	Credential should be scoped to correct service: '`###-DB`'.
Incorrect or Missing Host Header / Invalid Signature	403	The request signature we calculated does not match the signature you provided. Check your AWS Secret Access Key and signing method. Consult the service documentation for details. Host header is missing or hostname is incorrect.
Missing `X-Amz-Security-Token`	403	'`x-amz-security-token`' is named as a `SignedHeader`, but it does not exist in the HTTP request
Missing Authorization Header	403	The request did not include the required authorization header, or it was malformed.
Missing Authentication Token	403	Missing Authentication Token.
Old Date	403	Signature expired: *20181011T213907Z* is now earlier than *20181011T213915Z* (*20181011T214415Z* - *5*.)
Future Date	403	Signature not yet current: *20500224T213559Z* is still later than *20181108T225Z* (*20188T225425Z* + *5*.)
Incorrect Date Format	403	Date must be in ISO-8601 'basic format'. Got '*##*'. See https://en.wikipedia.org/wiki/ISO_8601.

Neptune Service Error Code	HTTP status	Message
Unknown/Missing Access Key or Session Token	403	The security token included in the request is invalid.
Unknown/Missing Secret Key	403	The request signature we calculated does not match the signature you provided. Check your AWS Secret Access Key and signing method. Consult the service documentation for details. Host header is missing or hostname is incorrect.
`TooManyRequestsException`	429	The rate of requests exceeds the maximum throughput. OK to retry.

資料庫叢集管理 API 錯誤訊息和代碼

這些 Amazon Neptune 錯誤與使用AWS SDK 和建立和修改 Neptune 資源的 API 相關聯AWS CLI。

下表包含錯誤代碼、訊息和 HTTP 狀態。

Neptune Service Error Code	HTTP status	Message
`AccessDeniedException`	400	您沒有足夠存取權可執行此動作。
`IncompleteSignature`	400	請求簽署不符合 AWS 標準。
`InternalFailure`	500	因為不明的錯誤、例外或故障，處理請求失敗。
`InvalidAction`	400	請求的動作或操作無效。確認已正確輸入動作。
`InvalidClientTokenId`	403	提供的 X.509 憑證或 AWS 存取金鑰 ID 不存在於我們的記錄中。
`InvalidParameterCombination`	400	同時使用了不應搭配使用的參數。
`InvalidParameterValue`	400	為輸入參 out-of-range 數提供了無效的或值。
`InvalidQueryParameter`	400	為輸入參 out-of-range 數提供了無效的或值。
`MalformedQueryString`	404	查詢字串包含語法錯誤。
`MissingAction`	400	請求中遺失動作或必要參數。
`MissingAuthenticationToken`	403	請求必須包含有效 (已註冊) 的 AWS 存取金鑰 ID 或 X.509 憑證。

Neptune Service Error Code	HTTP status	Message
`MissingParameter`	400	未提供適用於指定動作的必要參數。
`OptInRequired`	403	AWS 存取金鑰 ID 需要訂閱服務。
`RequestExpired`	400	請求之送達時間為超過請求之日期戳記的 15 分鐘後或超過請求過期日期的 15 分鐘後 (例如預先簽章的 URL)，或者請求的日期戳記為未來 15 分鐘後。
`ServiceUnavailable`	503	由於伺服器暫時故障，請求失敗。
`ThrottlingException`	400	由於請求調節，因此請求遭到拒絕。
`ValidationError`	400	輸入不符合 AWS 服務規定的限制。

Neptune 載入器的錯誤和饋送訊息

以下訊息由 Neptune 載入器的 `status` 端點傳回。如需詳細資訊，請參閱Get-Status API (p. 282)。

下表是載入器的饋送代碼和描述。

Error or Feed Code	Description
LOAD_NOT_STARTED	已記錄載入，但未開始。
LOAD_IN_PROGRESS	指示載入正在進行中，並指定目前載入的檔案數目。 當載入程式剖析檔案時，會建立一或多個要 parallel 載入的區塊。由於單一檔案可以產生多個區塊，因此此訊息所包含的檔案計數通常小於大量載入程序所使用的執行緒數目。
LOAD_COMPLETED	載入已完成，無任何錯誤或錯誤在可接受的閾值內。
LOAD_CANCELLED_BY_USER	使用者已取消載入。
LOAD_CANCELLED_DUE_TO_ERRORS	系統因為錯誤已取消載入。
LOAD_UNEXPECTED_ERROR	載入失敗並發生未預期的錯誤。
LOAD_FAILED	由於一或多個錯誤，載入失敗。
LOAD_S3_READ_ERROR	由於間歇性或暫時性的 Amazon S3 連線問題導致饋送失敗。如有任何饋送收到此錯誤，整體載入狀態將設為 LOAD_FAILED。
LOAD_S3_ACCESS_DENIED_ERROR	存取 S3 儲存貯體遭拒。如有任何饋送收到此錯誤，整體載入狀態將設為 LOAD_FAILED。

Error or Feed Code	Description
LOAD_COMMITTED_W_WRITE_CONFLICTS	載入資料遞交時發生未解決的寫入衝突。 將嘗試解決寫入衝突載入在單獨的交易和更新饋送的負載狀態的進度。如果最終饋送狀態為 LOAD_COMMITTED_W_WRITE_CONFLICTS，請嘗試重新繼續載入，它可能會成功而不會發生寫入衝突。寫入衝突通常與錯誤輸入資料無關，但資料中的重複會提高發生寫入衝突的可能性。
LOAD_DATA_DEADLOCK	Load was automatically rolled back due to deadlock.
LOAD_DATA_FAILED_DUE_TO_FEED_MODIFIED_OR_DELETED	載入失敗，因為檔案已在載入開始後刪除或更新。
LOAD_FAILED_BECAUSE_DEPENDENCY_NOT_SATISFIED	未執行載入請求，因為其相依性檢查失敗。
LOAD_IN_QUEUE	載入請求已排入佇列，正在等待執行。
LOAD_FAILED_INVALID_REQUEST	載入失敗，因為請求無效 (例如，指定來源/儲存貯體可能不存在，或檔案格式無效)。

Amazon Neptune 的引擎版本

Amazon Neptune 定期發行引擎更新。

您可以使用執行個體狀態 API (p. 770) 或 Neptune 主控台判斷目前安裝的引擎版本。版本號碼會告訴您執行的是原始發行版本 (隱含.R1)，還是較新的修補程式 (.R2.R3、等等)。如需版本編號的更多資訊，請參閱引擎版本號碼 (p. 859)。

> **Note**
>
> 對於從以後的次要引擎發行版本，隱含的1.0.1.0.R1起始版本為.R2，後續修補程式為.R3、等等。

如需如何更新 Neptune DB 引擎的相關資訊，以及更新的其他相關資訊，請參閱海王星引擎更新 (p. 857)。

下表列出自 1.0.1.0 以來的所有引擎版本，以及版本 e 的相關資訊。nd-of-life您可以使用此表格中的日期來規劃測試和升級週期。

版本	主要版本	次要版本	狀態	Released	生命週期結束	升級至：
1.2.1.0 (p. 861)	1.2	1.0	活躍	2023-03-08	2025-10-30	N/A
1.2.0.2 (p. 870)	1.2	0.2	活躍	2022-11-16	2024-10-31	1.2.1.0
1.2.0.1 (p. 879)	1.2	0.1	活躍	2022-10-26	2024-10-31	1.2.1.0
1.2.0.0 (p. 884)	1.2	0.0	活躍	2022-07-21	2024-10-31	1.2.1.0
1.1.1.0 (p. 894)	1.1	1.0	活躍	2022-04-19	2024-10-31	1.2.1.0
1.1.0.0 (p. 915)	1.1	0.0	活躍	2021-11-19	2024-01-30	1.1.1.0
1.0.5.1 (p. 924)	1.0	5.1	棄用	2021-10-01	2023-01-30	1.1.0.0
1.0.5.0 (p. 931)	1.0	5.0	棄用	2021-07-27	2023-01-30	1.1.0.0
1.0.4.2 (p. 938)	1.0	4.2	棄用	2021-06-01	2023-01-30	1.1.0.0
1.0.4.1 (p. 943)	1.0	4.1	棄用	2020-12-08	2023-01-30	1.1.0.0
1.0.4.0 (p. 950)	1.0	4.0	棄用	2020-10-12	2023-01-30	1.1.0.0
1.0.3.0 (p. 954)	1.0	3.0	棄用	2020-08-03	2023-01-30	1.1.0.0
1.0.2.2 (p. 960)	1.0	2.2	棄用	2020-03-09	2022-07-29	1.0.3.0
1.0.2.1 (p. 969)	1.0	2.1	棄用	2019-11-22	2022-07-29	1.0.3.0
1.0.2.0 (p. 978)	1.0	2.0	棄用	2019-11-08	2020-05-19	1.0.3.0
1.0.1.2 (p. 983)	1.0	1.2	棄用	2019-10-15	—	—
1.0.1.1 (p. 984)	1.0	1.1	棄用	2019-08-13	—	—
1.0.1.0. (p. 984)*	1.0	1.0 版*	棄用	07-02 和更新 2019-07-02	—	—

規劃 Amazon Neptune 引擎版本壽命

Neptune 引擎版本幾乎總是在日曆季度結束時達到其生命週期結束。發生重要安全性或可用性問題時，才會發生例外狀況

當引擎版本到達其生命週期結束時，您將需要將 Neptune 資料庫升級至較新的版本。

一般而言，Neptune 引擎版本繼續提供如下：

- 次要引擎版本：次要引擎版本在發行後至少 6 個月內仍然可用。
- 主要引擎版本：主要引擎版本在發行後至少可用 12 個月。

在引擎版本到達生命週期結束前至少 3 個月，AWS會向與您AWS帳戶關聯的電子郵件地址發送自動電子郵件通知，並將相同的消息發佈到您的 AWSHealth 儀表板。這會讓您有時間規劃和準備升級。

當引擎版本到達其生命週期結束時，您將無法再使用該版本建立新的叢集或執行個體，也無法使用該版本自動調度資源建立執行個體。

實際到達生命週期結束的引擎版本將在維護時段期間自動升級。在引擎版本生命週期結束前 3 個月傳送給您的訊息將包含此自動更新所涉及的詳細資料，包括您將自動升級的版本、對資料庫叢集的影響，以及我們建議的動作。

> **Important**
>
> 您必須負責將資料庫引擎版本保持在最新狀態。 AWS敦促所有客戶將其資料庫升級到最新的引擎版本，以便受益於最新的安全性、隱私權和可用性保護措施。如果您在不受支援的引擎或軟體上操作資料庫超過棄用日期 (以下稱「Legacy Engine」)，您將面臨更大的安全性、隱私權和營運風險 (包括停機事件) 的可能性。
> 在任何引擎上操作您的數據庫均受制於管理您使用AWS服務的協議。舊版引擎無法正常使用。AWS如果認為傳統引擎對服務、其關係企業或任何第三方構成安全性或責任風險，或有傷害風險，則AWS不再對 Legacy Engine 提供支援，且AWS可能隨時限制對任何 Legacy Engine 的存取或使用任何傳統引擎。AWS您決定繼續在舊版引擎中執行您的內容，可能會導致您的內容無法使用、損毀或無法復原。在傳統引擎上執行的資料庫受服務等級協定 (SLA) 例外規範。
> 在舊版引擎上執行的資料庫和相關軟體包含錯誤、錯誤、瑕疵和/或有害元件。因此，儘管協議或服務條款有任何相反的規定，仍「按原樣」AWS 提供傳統引擎。

亞馬遜海神引擎更新

亞馬遜海王星定期發行引擎更新。您可以使用執行個體狀態 API (p. 770) 來判斷您目前安裝的引擎發行版本。

引擎版本列在 Amazon Neptune 的引擎版本 (p. 856)，而修補程式列在最新更新 (p. 3)。

您可以在本節中找到有關如何發行更新以及如何升級 Neptune 引擎的資訊。例如，引擎版本編號將在中說明海王星引擎版本編號 (p. 859)。

> **Note**
>
> 在 2019 年 11 月之前，Neptune 一次只支援一個引擎版本，引擎版本號碼全部採用表單`1.0.1.0.200`*`<xxx>`*，修補程式編號xxx在哪裡。新的引擎版本均以舊版修補程式發布。
> 從 2019 年 11 月開始，Neptune 支援多種版本，讓客戶更有效地控制其升級路徑。因此，引擎版本編號已變更。

主題

- 升級前一律測試 (p. 858)

- 升級前一律建立手動快照 (p. 858)
- 升級海王星引擎的過程 (p. 858)
- 從 1.2.0.0 之前的版本升級至引擎 1.2.0.0 或更新版本 (p. 859)
- 海王星引擎版本編號 (p. 859)
- 使用控制台將引擎升級到更新的版本 (p. 860)
- 使用 CLI 手動執行 Neptune 資料庫引擎的次要升級 (p. 860)
- 使用 CLI 手動執行海王星資料庫引擎的主要升級 (p. 860)

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

升級海王星引擎的過程

> **Note**
>
> 更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您會遇到 20 或 30 秒到數分鐘的停機時間，之後您就可以使用資料庫叢集繼續執行。

修補程式版本 (`.R2`.`R3`、等) 一律會在下一個維護時段自動安裝。

- 您可以使用 AWS Management Console 手動升級引擎，如 使用控制台將引擎升級到更新的版本 (p. 860) 所述。
- 您也可以使用 CLI 手動升級引擎 (p. 860)。

如果您使用 AWS CLI 手動升級，請務必指定要升級為哪個引擎版本。若未指定，引擎可能會升級為並非最新的版本，或是並非您預期的版本。

您可以隨時使用 執行個體狀態 (p. 770) API 判斷資料庫叢集目前執行的引擎版本。如果您是使用 CreateDBCluster (p. 1005) 建立叢集，API 會在其回應中傳回當時用於建立叢集的引擎版本。

自動更新的時間取決於資料庫叢集的AWS區域和維護時段設定。您可以在 Neptune 主控台上檢視或變更維護時段設定。如需詳細資訊，請參閱Neptune 護時段 (p. 553)。

主要引擎版本必須手動安裝。

Note

您可以使用用來建立資料庫叢集的AWS CloudFormation範本，在其上執行主要的引擎更新。請參閱使用AWS CloudFormation範本更新 Neptune 資料庫叢集的引擎版本 (p. 556)。

從 1.2.0.0 之前的版本升級至引擎 1.2.0.0 或更新版本

引擎版本 1.2.0.0 (p. 884) 引入了幾項重大變更，這些變更可能會使從舊版升級變得更加複雜：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多的還原記錄 (200,000 或更多)，則在等待清除還原記錄完成時，升級嘗試可能會逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

海王星引擎版本編號

海王星版本號有 3 個部分。以版本編號 `1.0.2.0.R2` 為例：

1. 第一個部分 (前兩個數字，亦即 `1.0.2.0.R2` 中的 `1.0`) 為資料庫主要版本編號。

   只有在發生主要不相容變更時，此部分才會改變，例如變更資料的儲存方式，而且此儲存方式需要在升級時進行資料遷移。升級為新的主要版本通常需要與將要升級之叢集大小成比例的停機時間，而且可能比其他升級需要更長的時間。
2. 第二個部分 (第三、四個數字，亦即 `1.0.2.0.R2` 中的 `2.0`) 是引擎版本的次要版本編號。

   您可以選擇自動將引擎升級為新的次要版本，如下所述，或選擇在每個次要版本變更時再做決定。
3. 第三個部分 (最後一個數字，亦即 `1.0.2.0.R2` 中的 `R2`) 是引擎次要版本的修補程式編號。

   修補程式包括在次要引擎版本之間所部署的緊急變更，例如修正錯誤或變更安全性。引擎版本的第一版是 `R0`，而其修補程式將編號為 `R1`、`R2` 等，以此類推。

   修補程式一律會在系統維護時段自動套用，如下所述。

您可以使用 AWS CLI 列出可用的引擎版本，如下所述：

對於 Linux、OS X 或 Unix：

```
aws neptune describe-db-engine-versions \
    --region <your-region> \
    --engine neptune
```

針對 Windows：

```
aws neptune describe-db-engine-versions ^
    --region <your-region> ^
    --engine neptune
```

可用的引擎版本僅包含具有高於現有版本編號的發行版本，並且已定義其升級路徑者。

使用控制台將引擎升級到更新的版本

您可以使用下列方式執行 Neptune 資料庫叢集的AWS Management Console次要版本升級：

使用主控台升級資料庫叢集的引擎版本

1. 登入AWS管理主控台，然後開啟亞馬遜海王星主控台，網址為 https://console.aws.amazon.com/neptune/home。
2. 在導覽窗格中，選擇 Clusters (叢集)，然後選擇您要修改的資料庫叢集。
3. 選擇 Actions (動作)，然後選擇 Modify cluster (修改叢集)。Modify DB cluster (修改資料庫叢集) 頁面隨即出現。
4. 在 DB engine version (資料庫引擎版本) 中，選擇新版本。
5. 選擇 Continue (繼續)，並檢查修改的摘要。
6. 若要立即套用變更，請選擇 Apply immediately (立即套用)。
7. 在確認頁面上，檢閱您的變更。如果都正確，請選擇 Modify cluster (修改叢集) 以儲存您的變更。

   或者，選擇 Back (上一步) 以編輯變更，或是選擇 Cancel (取消) 以取消變更。

使用 CLI 手動執行 Neptune 資料庫引擎的次要升級

您可以使用AWS CLI和 ModifyDBCluster (p. 1017) API 手動執行資料庫叢集引擎版本的次要升級，如下所示：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
  --db-cluster-identifier (your-neptune-cluster) \
  --engine-version (new-engine-version) \
  --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
  --db-cluster-identifier (your-neptune-cluster) ^
  --engine-version (new-engine-version) ^
  --apply-immediately
```

請務必在其中包含您要升級為哪個引擎版本。若未包含，引擎可能會升級為並非最新的版本。

使用 CLI 手動執行海王星資料庫引擎的主要升級

若要使用AWS CLI和 ModifyDBCluster (p. 1017) API 手動執行資料庫叢集的引擎版本的主要升級，您需要提供比次要升級更多的參數：

包括您要升級到的引擎版本是非常重要的。若未包含，引擎可能會升級為並非最新的版本。

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
  --db-cluster-identifier (your-neptune-cluster) \
  --engine neptune \
  --engine-version (new-engine-version) \
  --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
  --db-cluster-identifier (your-neptune-cluster) ^
  --engine neptune ^
  --engine-version <new-engine-version> ^
  --apply-immediately
```

而不是`--apply-immediately`，您可以指定`--no-apply-immediately`。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
  --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
  ---db-instance-parameter-group-name (name of the custom instance parameter group)
```

亞馬遜海王星引擎

截至 2023 年 3 月 08 日，引擎版本 1.2.1.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此版本的後續修補程式版本

- 發行版本: (p. 868)
- 發行版本: (p. 865)

這個引擎版本的新功能

- 增加了對 TinkerPop3.6.2 的支持，其中添加了許多新的 Gemlin 功能，例如新的`mergeV()`，`mergeE()element()`，和步驟。`fail()mergeV()`和`mergeE()`步驟特別值得注意，因為它們提供了一個期待已久的聲明式選項來執行類似 Upsert 的操作，這應該大大簡化現有的代碼模式並使 Gemlin 更容易閱讀。3.6.x 版本還添加了 regex 謂詞，一個新的重載到需要一個`property()`步驟`Map`，以及`by()`調製行為的主要修訂，在使用它的所有步驟中都更加一致。

  請參閱TinkerPop變更記錄檔和升級頁面，以取得 3.6 版變更的相關資訊，以及升級時應考量的事項。

  如果您使`fold().coalesce(unfold(), <mutate>)`用條件式插入，建議您移轉至此處和此處所述的新`mergeV/E()`語法。Neptune 使用`Merge`比 for 更窄的鎖定模式`Coalesce`，這可以減少並發修改異常（CME）。

  如需有關此TinkerPop版本中可用新功能的詳細資訊，請參閱史蒂芬·馬萊特的部落格，探索亞馬遜海王星中 Apache TinkerPop 3.6.x 的新功能。
- 增加了對 R6i 實例類型的支持，由第三代英特爾至強可擴展處理器提供支持。與同類 R5 執行個體類型相比，每個 vCPU 最多可提供 15% 的運算/價格效能，最多可提供高達 15% 的記憶體頻寬，最多可提高 20% 的記憶體頻寬。
- 新增屬性圖和 RDF 圖形的圖形摘要 API (p. 138) 端點，可讓您取得有關圖形的快速摘要報告。

  對於屬性 (PG) 圖形，圖形摘要 API 會提供節點、邊緣標籤以及屬性金鑰的唯讀清單，以及節點、邊緣和屬性計數。對於 RDF 圖，它提供了類和謂詞鍵的列表，以及四邊形，主題和謂詞的計數。

  以下更改與新的圖形摘要 API 一起進行：
  - 添加了一個新的GetGraphSummary (p. 206)數據通道操作。
  - 添加了一個新的`rdf/statistics`端點來替換`sparql/statistics`端點，該端點現已被棄用。
  - 將統計狀態回應中的`summary`欄位名稱變更為`signatureInfo`，以免將其與圖形摘要資訊混淆。先前的引擎版本會繼續`summary`在 JSON 回應中使用。
  - 將統計狀態回應中`date`欄位的精確度從分鐘變更為毫秒。先前的格式是`2020-05-07T23:13Z`（分鐘精度），而新格式為`2023-01-24T00:47:43.319Z`（毫秒精度）。兩者都符合 ISO 8601 標準，但此更改可能會破壞現有代碼，具體取決於日期的解析方式。
  - 在 Workbench 中新增新的%statistics (p. 41)生產線魔法，可讓您擷取 DFE 引擎統計資料。
  - 在 Workbench 中新增%summary (p. 41)線條魔術，可讓您擷取圖形摘要資訊。
- 將慢速查詢記錄新增至記錄 (p. 785)執行時間超過指定臨界值的查詢。您可以使用兩個新的動態參數來啟用和控制慢速查詢記錄，即海王星_啟用_慢速查詢日誌和海王星 (p. 530)慢_查詢_LOG_閾值。 (p. 531)
- 增加了對兩個動態參數的支持，即新的集群參數 (p. 527)，海王星啟用_慢速查詢日誌和海王星慢_查詢_ (p. 530) 日誌閾值。 (p. 531)當您變更動態參數時，它會立即生效，而不需要重新啟動執行個體。
- 添加了一個海王星專用 OpenCypher removeKeyFromMap () (p. 102) 函數，該函數從地圖中刪除指定的鍵並返回生成的新映射。

這個引擎版本的改善項目

- 擴展的 Grimlin DFE 支持與本地範圍的`limit`步驟。
- 新增 DFE 引擎中的小精靈`by()DedupGlobalStep`調變支援。

- 新增 DFE 支援小精靈SelectStep和. SelectOneStep
- 各種 Gemlin 運算子的效能改善和正確性修正，包括repeat、、和. coalesce store aggregate
- 改進了涉及MERGE和的開放密碼查詢的性能。OPTIONAL MATCH
- 改進了涉及文字值映射列表UNWIND的 OpenCypher 查詢的性能。
- 改善具有IN篩選條件的 OpenCypher 查詢的效能。id例如：

```
MATCH (n) WHERE id(n) IN ['1', '2', '3'] RETURN n
```

- 新增了使用 BASE 陳述式為 SPARQL 查詢指定基礎 IRI 的功能 (請參閱查詢和更新的預設基本 IRI (p. 93))。
- 縮短了小鬼和 OpenCypher 僅邊緣散裝負載的負載處理等待時間。
- 當 Neptune 重新啟動時，大量載入會非同步恢復，以避免因 Amazon S3 連線問題所造成的漫長等待時間，再嘗試繼續失敗。
- 已改善 SPARQL 描述查詢的處理方式，這些查詢的 Descri beMode (p. 445) 查詢提示設定為 "CBD" (簡潔的界限描述)，而且涉及大量空白節點。

此引擎版本中修正的缺陷

- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串"null"，而不是博爾特和 SPARQL-JSON 中的空值。
- 修復了列表理解中的 OpenCypher 錯誤，該錯誤會產生空值，而不是為列表元素提供的值。
- 修復了一個 OpenCypher 錯誤，其中字節值未正確序列化。
- 修復了當輸入是邊緣遍歷到子遍歷內的頂點時發生的小鬼錯誤。UnionStep
- 修正了導致與相關的步驟標籤UnionStep無法正確傳播到每個子項的最後一個步驟的 Grimlin 錯誤。
- 修復了步驟後帶有標籤的dedup步驟的 Gremlin 錯誤，其中附加到repeat步dedup驟的標籤無法在進一步查詢中使用。
- 修復了一個小鬼錯誤，這錯誤令翻譯repeat步驟內部的步union驟失敗並出現內部錯誤。
- 修復了 DFE 查詢的 Grimlin 正確性問題，通過回到 T limit inkerpop 進行非聯集步驟的子遍歷。像這樣的表單中的查詢會受到影響：

```
g.withSideEffect('Neptune#useDFE', true).V().as("a").select("a").by(out().limit(1))
```

- 修復了SPARQL GRAPH模式不會考慮FROM NAMED子句提供的數據集的 SPARQL 錯誤。
- 修正 SPARQL 錯誤，其中DESCRIBE包含某些和FROM/或FROM NAMED子句的 SPARQL 並不總是正確使用預設圖表中的資料，有時會擲回例外狀況。請參閱 SPARQL 描述相對於默認圖形的行為 (p. 447)。
- 修正了 SPARQL 錯誤，以便在拒絕空字元時傳回正確的例外狀況訊息。
- 修復了影響包含PipelinedHashIndexJoin (p. 471)運算符的計劃的 SPARQL 說明 (p. 453)錯誤。
- 修正當提交傳回常數值的查詢時，會擲回內部錯誤的錯誤。(
- 修正鎖死偵測器邏輯偶爾會導致引擎沒有回應的問題。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.2.1.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.6.2
- 小精靈最新版本支持：3.6.2
- 開放密碼版本：Neptune-9.0.20190305-1.1
- 版本：1.1

引擎版本 1.2.1.0 的升級路徑

您可以從任何大於或等於 1.1. (p. 915) 0.0 的舊版 Neptune 引擎發行版本手動升級至此發行版本。

Note

從引擎版本 1.2.0.0 (p. 884) 開始，所有自訂參數群組和自訂叢集參數群組，您使用的引擎版本早於之前的版本，現在`1.2.0.0`必須使用參數群組系列重新建立。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組將無法從以`1.2.0.0`後使用的版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。

您將不會自動升級至此主要版本。

升級至此版本

亞馬遜海王星 1.2.1.0 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二三年六月十三日，引擎版本 1.2.1.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的新功能

- 已新增使用 IAM 角色鏈結 (p. 259)跨帳戶大量載入的支援。

這個引擎版本的改善項目

- 改進了 Gremlin 的`fail()`步驟，以區分它從泛型產生的異常，`InternalFailureException`並確保提供給它的任何用戶提供的消息都被傳回給調用者。
- 改善了`store`、`aggregate`、、和的 Gemlin 查詢引擎最佳化。`cap` `limit` `hasLabel`

- 增加了對開放式三角函數的支持：
  - acos()
  - asin()
  - atan()
  - atan2()
  - cos()
  - cot()
  - degrees()
  - pi()
  - radians()
  - sin()
  - tan()
- 增加了對幾個開放密碼聚合函數的支持：
  - percentileDisc()
  - stDev()
- 增加了對開放密碼epochmillis()函數的支持，該函數將一個轉換為datetime. epochmillis 例如：

```
MATCH (n) RETURN epochMillis(n.someDateTime)
1698972364782
```

- 增加了對開放密碼模 ( ) %運算符的支持。
- 已新增對 OpenCypher 靜態除錯說明工具的支援。
- 增加了對開放密碼randomUUID()函數的支持。
- 改善開放密碼效能：
  - 改進了解析器和查詢規劃器。
  - 改善 DFE 引擎的 CPU 使用率。
  - 改善包含多個重複使用相同變數之更新子句的查詢效能。例子是：

```
MERGE (n {name: 'John'})
  or
MERGE (m {name: 'Jim'})
  or
MERGE (n)-[:knows {since: 2023}]#(m)
```

  - 針對多跳查詢模式的最佳化查詢計劃，例如：

```
MATCH (n)-->()-->()-->(m)
RETURN n m
```

  - 通過參數化查詢改進了列表和映射注入的性能。例如：

```
UNWIND $idList as id MATCH (n {`~id`: id})
RETURN n.name
```

  - 改善包含WITH的查詢執行，使其成為適當的障礙。
  - 經過優化，以避免在聚合函數中Unfold對值進行冗餘實現化。
- 改善子VALUES句中包含大量靜態輸入的 SPARQL 查詢效能，例如：

```
SELECT ?n WHERE { VALUES (?name) { ("John") ("Jim") ... many values ... } ?n a ?
n_type . ?n ?name . }
```

- 已改善 CBD 查詢效能。

此引擎版本中修正的缺陷

- 修正了在查詢規劃階段期間，具有深層巢狀的長時間查詢造成 CPU 使用率和查詢逾時的 Gemlin 錯誤。
- 修復了使mergeV用或時NullPointerException可能會拋出無效的小鬼錯誤。mergeE

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.1.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.6.2
- 小精靈最新版本支持：3.6.2
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.2.1.0.R3 的路徑

升級至此版本

亞馬遜海王星 1.2.1.0.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二三年五月二日，引擎版本 1.2.1.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 已新增`enableInterContainerTrafficEncryption`參數至所有 Neptune ML API (p. 752)，您可以在訓練或超參數調整工作中啟用和停用容器間流量加密。
- 增加了對小鬼`mergeV()`和多標籤的支持。`mergeE()`

此引擎版本中修正的缺陷

- 修復了 OpenCypher 錯誤，其中更新和返回查詢沒有處理 `orderBylimit`，或`skip`正確。
- 修復了 OpenCypher 錯誤，該錯誤允許一個請求中包含的參數被另一個同時請求中包含的參數覆蓋。
- 修復了一個 OpenCypher 錯誤，其中緩慢的查詢日誌不包含正確的查詢時間。
- 修復`GroupCountStep`了一個 Gremlin 錯誤，其中包含的查詢作為字符串提交時可能發生事務洩漏。
- 修復了啟用緩慢查詢日誌時查WebSocket詢失敗的 Grimlin 錯誤。
- 修復了在請求的慢速查詢日誌中缺少存儲計數器調試日誌的 Gimlin 錯誤。WebSocket
- 修正了一些涉及`mergeV()`和的小鬼錯誤。`mergeE()`
- 修正錯誤估計命名圖形查詢成本的 SPARQL 錯誤，導致查詢計劃和錯誤不佳。out-of-memory
- 修正了影響啟用 IAM 的叢集上的小鬼和 OpenCypher 查詢授權的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.1.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.6.2`
- 小精靈最新版本支持：`3.6.2`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.2.1.0.R2 的路徑

升級至此版本

亞馬遜海王星 1.2.1.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 11 月 20 日，引擎版本 1.2.0.2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。

- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此版本的後續修補程式版本

- 發行版本: (p. 877)
- 發行版本: (p. 874)
- 發行版本: (p. 872)

這個引擎版本的新功能

- 在海王星 ML 中引入了 Gemlin 的實時感應推論 (p. 667)。
- 引入 OpenCypher 擴充功能，支援指定實體的自訂識別碼值 (p. 102)，而不是海王星以其他方式產生的 UUID。分配自定義 ID 的能力使得從 Neo4j 遷移到海王星變得更加容易。

  Warning

  這個 OpenCypher 規格的延伸不向後相容，因為現在被視~id為保留的屬性名稱。如果您已在資料和查詢中~id作為性質使用，則必須在升級至此版本之前將性質移轉至新的屬性金鑰 (p. 103)。~id
- 新增了幾個新的 SPARQL DESCRIBE 模式 (p. 444)以及查詢提示來設定它們。

這個引擎版本的改善項目

- 改善了開放密碼效能，特別是針對 VLP 查詢。
- 改善具有非終端限制的 Gemlin 查詢的 DFE 效能，例如：

```
g.withSideEffect('Neptune#useDFE',true).V().hasLabel('Student').limit(5).out('takesCourse')
```

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

引擎版本 1.2.0.2 的升級路徑

升級至此版本

亞馬遜海王星 1.2.0.2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至二零二三年五月零八日，引擎版本 1.2.0.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此引擎版本中修正的缺陷

- 已修正 SPARQL 錯誤，其中透過`VALUES`子句插入大量值可能導致效能降低。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.2.2.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.5`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.2.0.2.R4 的路徑

如果您執行的是引擎版本，Neptune DB 叢集將在下一個維護時段期間自動升級至此維護修補程式版本`1.2.0.2`。

升級至此版本

亞馬遜海王星 1.2.0.2.R4 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
```

```
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二三年三月二十七日，引擎版本 1.2.0.2.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

> **Note**
>
> 如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。neptune1.2舊版使用了參數群組族群neptune1，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且UndoLogsListSize (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果UndoLogsListSizeCloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：request.setResourcePath("/openCypher"));。在其他語言中，/openCypher可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 對於無伺服器資料庫叢集，將最小容量設定變更為 1.0 NCU，將最低有效最大設定變更為 2.5 NCU。請參閱 Neptune 無伺服器資料庫叢集中的容量擴充 (p. 578)
- 已新增enableInterContainerTrafficEncryption參數至所有 Neptune ML API (p. 752)，您可以在訓練或超參數調整工作中啟用和停用容器間流量加密。

此引擎版本中修正的缺陷

- 修復option(Predicate)了一個沒有被識別為有效的小鬼語法的小精靈錯誤。
- 修正了導致查詢失敗，因為包含太多步驟而導致查詢無法正確清理的 Grimlin 錯誤。
- 修正了一個小鬼正確性問題，該問題會limit因為回到 Tinkerpop 而影響非聯集步驟的子項遍歷 DFE 查詢。下面是這樣一個查詢的一個例子：

```
g.withSideEffect('Neptune#useDFE', true).V().as("a").select("a").by(out().limit(1))
```

- 修復了當以字符串形式提交的查詢包含時潛在的 Gemlin 事務洩漏。GroupCountStep
- 修復了 OpenCypher 錯誤，其中未在地圖列表或列表中正確推斷出參數值的類型。
- 修復了 OpenCypher 錯誤，其中更新和返回查詢沒有處理 orderBylimit，或skip正確。
- 修復了 OpenCypher 錯誤，該錯誤允許一個請求中包含的參數被另一個同時請求中包含的參數覆蓋。
- 已修正 SPARQL 錯誤，其中在VALUES子句中插入大量值可能導致效能降低。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.2.2.2.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.5
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

引擎版本 1.2.0.2.R3 的升級路徑

如果您執行的是引擎版本，Neptune DB 叢集將在下一個維護時段期間自動升級至此維護修補程式版本1.2.0.2。

升級至此版本

亞馬遜海王星 1.2.0.2.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開preupgrade頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> ```

```
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至 2022 年 12 月 15 日，引擎版本 1.2.0.2.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 改進了涉及`MERGE`和的開放密碼查詢的性能。`OPTIONAL MATCH`
- 改進了涉及文字值映射列表`UNWIND`的 OpenCypher 查詢的性能。
- 改善具有`IN`篩選條件的 OpenCypher 查詢的效能。`id`例如：

```
MATCH (n) WHERE id(n) IN ['1', '2', '3'] RETURN n
```

- 各種 Gemlin 運算子的效能改善和正確性修正，包括`repeat`、、和. `coalesce` `store` `aggregate`

此引擎版本中修正的缺陷

- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串`"null"`，而不是博爾特和 SPARQL-JSON 中的空值。
- 修復了導致附加的步驟標籤`UnionStep`不會傳播到其子遍歷的最後一個路徑元素的小鬼錯誤。

- 修正了在 DFE 引擎的by()遍歷下valueMap()無法最佳化的小精靈錯誤。
- 修復了作為較長 Gemlin 事務的一部分執行的讀取查詢不會鎖定行的 Grimlin 錯誤。
- 修正稽核記錄錯誤，導致記錄不必要的資訊，以及記錄中遺失某些欄位。
- 已修正稽核記錄錯誤，其中未記錄啟用 IAM 之資料庫叢集的 HTTP 要求的 IAM ARN。
- 修正查詢快取錯誤，以便限制用於寫入快取記憶體的增量記憶體。
- 修正查閱快取錯誤，此錯誤涉及在寫入失敗時為查閱快取設定唯讀模式。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.2.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

引擎版本 1.2.0.2.R2 的升級路徑

如果您執行的是引擎版本，Neptune DB 叢集將在下一個維護時段期間自動升級至此維護修補程式版本1.2.0.2。

升級至此版本

亞馬遜海王星 1.2.0.2.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 10 月 26 日，正在普遍部署引擎版本 1.2.0.1。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此版本的後續修補程式版本

- 維護版本: (p. 882)

這個引擎版本的新功能

- 推出 Amazon Neptune 無伺服器 (p. 577)，這是一種隨需自動調整規模組態，可擴展您的資料庫叢集以滿足處理需求的增加，然後在需求減少時再次縮減。

這個引擎版本的改善項目

- 改進了小鬼查詢`order-by`的性能。在 a 結尾的 Gemlin 查詢`NeptuneGraphQueryStep`現在使用較大的塊大小以獲得更好的性能。`order-by`這不適用`order-by`於查詢計劃的內部（非根）節點。
- 改進了小鬼更新查詢的性能。在加入邊或屬性時，必須鎖住頂點和邊線以防刪除。此變更可消除交易中的重複鎖定，進而改善效能。
- 透過`dedup`向下推送至原生執行層，改善在`repeat()`子查詢`dedup()`內部使用的 Gemlin 查詢的效能。
- 新增 IAM 身份驗證錯誤的使用者友好錯誤訊息 這些訊息現在會顯示您的 IAM 使用者或角色 ARN、資源 ARN，以及要求的未經授權動作清單。未經授權的動作清單可協助您查看所使用的 IAM 政策中可能遺漏或明確拒絕的項目。

此引擎版本中修正的缺陷

- 修復了在升級到 TinkerPop 3.5 `PartitionStrategy` 後使用錯誤導致錯誤消息「PartitionStrategy不適用於匿名遍歷」的錯誤，導致無法執行遍歷。
- 修復了涉及`WherePredicateStep`翻譯的 Gemlin 正確性錯誤，其中 Neptune 的查詢引擎對使用`where(P.neq('x'))`和變化的查詢產生不正確的結果。
- 修復了在某些情況下導致重複節點和邊緣創建的`MERGE`子句中的 OpenCypher 錯誤。
- 修正處理`OPTIONAL`子句中包含`(NOT) EXISTS`的查詢時的 SPARQL 錯誤，此錯誤在某些情況下會遺失查詢結果。
- 已修正大量載入程式錯誤，此錯誤會在重度插入負載下造成效能回歸。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

引擎版本 1.2.0.1 的升級路徑

升級至此版本

亞馬遜海王星 1.2.0.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

截至 2022 年 12 月 13 日，引擎版本 1.2.0.1.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 已改善涉及常值`UNWIND`對應清單的 OpenCypher 查詢效能。
- 各種 Gemlin 運算子的效能改善和正確性修正，包括`repeat`、、和. `coalesce` `store` `aggregate`

此引擎版本中修正的缺陷

- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串`"null"`，而不是博爾特和 SPARQL-JSON 中的空值。
- 修正稽核記錄錯誤，導致記錄不必要的資訊，以及記錄中遺失某些欄位。
- 修正查詢快取錯誤，以便限制用於寫入快取記憶體的增量記憶體。
- 修正查閱快取錯誤，此錯誤涉及在寫入失敗時為查閱快取設定唯讀模式。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.1.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.2.0.1.R2 的路徑

如果您執行的是引擎版本 1.2.0.1，Neptune DB 叢集將在下一個維護時段期間自動升級至此修補程式版本 (p. 879)

升級至此版本

亞馬遜海王星 1.2.0.1.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.2.0.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.2.0.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> ```

```
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一年七月二十一日，引擎版本 1.2.0.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果您要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此版本的後續修補程式版本

- 發行版本: (p. 891)
- 版本發行: (p. 888)

這個引擎版本的新功能

- 增加了對全局數據庫 (p. 73)的支持。Neptune 全域資料庫跨越多個AWS 區域，且由一個區域中的主要資料庫叢集和其他區域中最多五個次要資料庫叢集所組成。
- 根據資料平面動作，新增了對 Neptune IAM 政策中更精細的存取控制支援，比之前提供的更精細的存取控制。這是一項重大變更，必須調整以取代`connect`動作為基礎的現有 IAM 政策，才能使用更精細的資料平面動作。請參閱 IAM 政策的類型 (p. 177)。

- 改善讀取器執行個體可用 之前，當寫入器執行個體重新啟動時，Neptune 叢集中的所有讀取器執行個體也會自動重新 從引擎版本 1.2.0.0 開始，讀取器執行個體會在寫入器重新啟動後保持作用中，進而改善讀取器可用性。您可以個別重新啟動讀取器執行個體，以取得參數群組變更。請參閱 在 Amazon Neptune 中重新啟動資料庫 (p. 574)。
- 添加了一個新的海王星流 _expiry_days 數據庫群集參數，該參數允許您設置在刪除之前將流記錄 (p. 532)保留在服務器上的天數。範圍為 1 到 90，預設值為 7。

這個引擎版本的改善項目

- 改進了查詢的 Grimlin 序列化性能。ByteCode
- Neptune 現在會使用 DFE 引擎處理文字述詞，以提升效能。
- 海王星現在使用 DFE 引擎處理 Grimlin `limit()` 步驟，包括非終端和子周遊限制。
- 變更了 Gemlin `union()` 步驟的 DFE 處理以處理其他新功能，這表示參考節點會如預期般顯示在查詢設定檔中。
- 透過並行化 DFE 中某些昂貴的聯結作業，效能提升最多 5 倍。
- 新增了對`OrderGlobalStep order(global)`小靈 DFE 引擎的`by()`調變支援。
- 在 DFE 的詳細說明中增加了注入靜態值的顯示。
- 改善修剪重複模式時的效能。
- 在格林 DFE 引擎中增加了訂單保存支持。
- 改進了具有空過濾器的 Gremlin 查詢的性能，例如：

```
g.V().hasId(P.within([]))
```

```
g.V().hasId([])
```

- 改善 SPARQL 查詢使用的數值太大而無法在內部呈現時的錯誤訊息。
- 在停用串流時減少索引搜尋，改善刪除具有關聯邊緣之頂點的效能。
- 將 DFE 支援延伸至`has()`步驟的更多變體，特別是至`hasKey()`、`hasLabel()`、以及範圍內字串 /URI 的述詞。`has()`這會影響下列查詢：

```
// hasKey() on properties
g.V().properties().hasKey("name")
g.V().properties().has(T.key, TextP.startingWith("a"))
g.E().properties().hasKey("weight")
g.E().properties().hasKey(TextP.containing("t"))

// hasLabel() on vertex properties
g.V().properties().hasLabel("name")

// range predicates on ID and Label fields
g.V().has(T.label, gt("person"))
g.E().has(T.id, lte("(an ID value)"))
```

- 添加了一個海王星特定 OpenCypher join() (p. 101)函數，該函數將列表中的字符串連接為單個字符串。
- 更新 Neptune 管理的原則 (p. 179)，以納入新全域資料庫 API 的資料存取權限和權限。

此引擎版本中修正的缺陷

- 修復了未指定內容類型的 HTTP 請求會自動失敗的錯誤。
- 修正查詢最佳化工具中無法在查詢內使用服務呼叫的 SPARQL 錯誤。
- 修復了海龜 RDF 解析器中的 SPARQL 錯誤，該錯誤令 Unicode 數據的特定組合導致失敗。

- 修正 SPARQL 錯誤，其中特定GRAPH和SELECT子句組合產生不正確的查詢結果。
- 修正了在聯集步驟中使用任何篩選步驟的查詢造成正確性問題的 Grimlin 錯誤，如下所示：

```
g.V("1").union(hasLabel("person"), out())
```

- 修復了一個小鬼錯誤，其中count()的both().simplePath()會導致返回的結果實際數量增加一倍，沒有。count()
- 修正 OpenCypher 錯誤，其中伺服器針對啟用 IAM 身份驗證的叢集的 Bolt 請求產生錯誤的簽章不匹配例外狀況。
- 修復了 OpenCypher 錯誤，其中使用 HTTP 保持活動的查詢如果在失敗的請求後提交，則可能會錯誤地關閉該錯誤。
- 修正提交傳回常數值的查詢時，可能會擲回內部錯誤的 OpenCypher 錯誤。
- 已修正說明詳細資料中的錯誤，以便 DFE 子查詢Time(ms)現在可以正確地將 DFE 子查詢中運算子的 CPU 時間加總。請考慮下面的解釋輸出摘錄作為一個例子：

```
subQuery1
#######################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments                          # Mode
 # Units In # Units Out # Ratio # Time (ms) #
#######################################################################################################################
  ...
#######################################################################################################################
# 1  # 2      # -      # DFEChunkLocalSubQuery # subQuery=...graph#336e.../graph_1 # -
 # 1        # 1         # 1.00  # 0.38      #
#    #        #        #                      # coordinationTime(ms)=0.026        #
 #          #           #       #           #
#######################################################################################################################
  ...
subQuery=...graph#336e.../graph_1
#######################################################################################################################
# ID # Out #1 # Out #2 # Name                  # Arguments                          # Mode
 # Units In # Units Out # Ratio # Time (ms) #
#######################################################################################################################
# 0  # 1      # -      # DFESolutionInjection  # solutions=[?100 -> [-10^^<LONG>]] # -
 # 0        # 1         # 0.00  # 0.04      #
#    #        #        #                      # outSchema=[?100]                   #
 #          #           #       #           #
#######################################################################################################################
# 1  # 3      # -      # DFERelationalJoin     # joinVars=[]                        # -
 # 2        # 1         # 0.50  # 0.29      #
#######################################################################################################################
# 2  # 1      # -      # DFESolutionInjection  # outSchema=[]                      # -
 # 0        # 1         # 0.00  # 0.01      #
#######################################################################################################################
# 3  # -      # -      # DFEDrain              # -                                  # -
 # 1        # 0         # 0.00  # 0.02      #
#######################################################################################################################
```

下表最後一列的子查詢時間加起來為 0.36 ms () .04 + .29 + .01 + .02 = .36。當您添加到該子查詢 (.36 + .026 = .386) 的協調時間時，您會得到一個接近上表的最後一列 (即 0.38 ms) 中記錄的子查詢的時間的結果。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4

- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

引擎版本 1.2.0.0 的升級路徑

因為這是一個主要的引擎版本，沒有自動升級到它。

您只能從引擎版本 1.1.1.0 的最新修補程式版本1.2.0.0手動升級至發行版本 (p. 894)。先前的引擎發行版本必須先升級至的最新版本，1.1.1.0才能升級至1.2.0.0。

因此，在您嘗試升級至此版本之前，請確認您目前執行的是最新的修補程式版本1.1.1.0。如果您不是，請先升級至的最新修補程式版本1.1.1.0。

在升級之前，您還必須使用參數群組系列重新建立任何已與舊版一起使用的自訂資料庫叢集參數群neptune1.2組。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。

如果您先升級為發行版，1.1.1.0然後立即升級至1.2.0.0，您可能會遇到如下錯誤：

```
   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
   Cannot modify engine version because instance (instance identifier) is
   running on an old configuration. Apply any pending maintenance actions on the instance
 before
   proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置動作完成，或立即觸發維護時段，讓先前的升級完成 (請參閱維護 Amazon Neptune 資料庫叢集 (p. 553))。

升級至此版本

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine neptune \
    --engine-version 1.2.0.0 \
    --allow-major-version-upgrade \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine neptune ^
    --engine-version 1.2.0.0 ^
    --allow-major-version-upgrade ^
    --apply-immediately
```

而不是--apply-immediately，您可以指定--no-apply-immediately。若要執行主要版本升級，需要此allow-major-version-upgrade參數。此外，請務必包含引擎版本，否則您的引擎可能已升級到其他版本。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
    --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
--db-instance-parameter-group-name (name of the custom instance parameter group)
```

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至 2022 年 12 月 15 日，引擎版本 1.2.0.0.R3 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進

行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

如果UndoLogsListSizeCloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。

- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 改進了涉及`MERGE`和的開放密碼查詢的性能。`OPTIONAL MATCH`
- 已改善涉及常值`UNWIND`對應清單的 OpenCypher 查詢效能。
- 改善具有`IN`篩選條件的 OpenCypher 查詢的效能。`id`例如：

```
MATCH (n) WHERE id(n) IN ['1', '2', '3'] RETURN n
```

- 各種 Gemlin 運算子的效能改善和正確性修正，包括`repeat`、、和. `coalesce` `store` `aggregate`

此引擎版本中修正的缺陷

- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串`"null"`，而不是博爾特和 SPARQL-JSON 中的空值。
- 修復了 OpenCypher 錯誤，以便在值為列表或映射列表時能夠正確解釋參數類型。
- 修正稽核記錄錯誤，導致記錄不必要的資訊，以及記錄中遺失某些欄位。
- 已修正稽核記錄錯誤，其中未記錄啟用 IAM 之資料庫叢集的 HTTP 要求的 IAM ARN。
- 修正查詢快取錯誤，以便限制用於寫入快取記憶體的增量記憶體。
- 修正查閱快取錯誤，此錯誤涉及在寫入失敗時為查閱快取設定唯讀模式。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.2.0.R3 的路徑

如果您執行的是引擎版本 `1.2.0.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您只能從引擎版本 `1.1.1.0` 的最新修補程式版本`1.2.0.0`手動升級至發行版本 (p. 894)。先前的引擎發行版本必須先升級至的最新版本，`1.1.1.0`才能升級至`1.2.0.0`。

如果您先升級為發行版，`1.1.1.0`然後立即升級至`1.2.0.0`，您可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the instance
before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

升級至此版本

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine neptune \
    --engine-version 1.2.0.0 \
    --allow-major-version-upgrade \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine neptune ^
    --engine-version 1.2.0.0 ^
    --allow-major-version-upgrade ^
    --apply-immediately
```

而不是`--apply-immediately`，您可以指定`--no-apply-immediately`。若要執行主要版本升級，需要此allow-major-version-upgrade參數。此外，請務必包含引擎版本，否則您的引擎可能已升級到其他版本。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
    --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
    --db-instance-parameter-group-name (name of the custom instance parameter group)
```

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 10 月 14 日，引擎版本 1.2.0.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Note

如果從 1.2.0.0 之前的引擎版本升級：

- 引擎版本 1.2.0.0 (p. 884) 為自訂參數群組和自訂叢集參數群組引入了新格式。因此，如果要從 1.2.0.0 之前的引擎版本升級至 1.2.0.0 或更新版本的引擎，則必須使用參數群組系列重新建立所有現有的自訂參數群組和自訂叢集參數群組。`neptune1.2`舊版使用了參數群組族群`neptune1`，而這些參數群組不適用於 1.2.0.0 及更高版本。如需詳細資訊，請參閱 Amazon Neptune (p. 527)。
- 引擎版本 1.2.0.0 也引入了復原記錄的新格式。因此，必須清除舊版引擎建立的任何還原日誌，且`UndoLogsListSize` (p. 778)CloudWatch指標必須降至零，才能開始從 1.2.0.0 之前的版本進行任何升級。如果在嘗試啟動更新時有太多還原記錄 (200,000 或更多)，升級嘗試可能會在等待清除還原記錄檔完成時逾時。

  您可以升級叢集的寫入器執行個體 (此為清除發生的位置)，以加快清除率。在嘗試升級之前執行此操作可能會在開始之前減少還原記錄的數量。將寫入器的大小增加為 24XL 執行個體類型，可以將清除率提高到每小時超過一百萬筆記錄。

  如果`UndoLogsListSize`CloudWatch量度非常大，開啟支援案例可協助您探索其他策略來降低指標。
- 最後，1.2.0.0 版中發生了重大變更，影響使用 Bolt 通訊協定搭配 IAM 身份驗證的早期程式碼。從 1.2.0.0 版開始，Bolt 需要一個用於 IAM 簽署的資源路徑。在 Java 中，設置資源路徑可能如下所示：`request.setResourcePath("/openCypher"));`。在其他語言中，`/openCypher`可以附加到端點 URI。如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 改進了小鬼查詢`order-by`的性能。在 a 結尾的 Gemlin 查詢`NeptuneGraphQueryStep`現在使用較大的塊大小以獲得更好的性能。`order-by`這不適用`order-by`於查詢計劃的內部（非根）節點。

- 改進了小鬼更新查詢的性能。在加入邊或屬性時，必須鎖住頂點和邊線以防刪除。此變更可消除交易中的重複鎖定，進而改善效能。
- 透過dedup向下推送至原生執行層，改善在repeat()子查詢dedup()內部使用的 Gemlin 查詢的效能。
- 新增了小鬼Neptune#cardinalityEstimates查詢提示。設定為時false，此選項會停用基數估計值。
- 新增 IAM 身份驗證錯誤的使用者友好錯誤訊息 這些訊息現在會顯示您的 IAM 使用者或角色 ARN、資源 ARN，以及要求的未經授權動作清單。未經授權的動作清單可協助您查看所使用的 IAM 政策中可能遺漏或明確拒絕的項目。

此引擎版本中修正的缺陷

- 修復了涉及WherePredicateStep翻譯的 Gemlin 正確性錯誤，其中 Neptune 的查詢引擎對使用where(P.neq('x'))和變化的查詢產生不正確的結果。
- 修復了在升級到 TinkerPop 3.5 PartitionStrategy 後使用錯誤導致錯誤消息「PartitionStrategy不適用於匿名遍歷」的錯誤，導致無法執行遍歷。
- 修正了與最終聯結和子群joinTime組內部統計相關的各種 Gemlin 錯誤。Project.ASK
- 修復了在某些情況下導致重複節點和邊緣創建的MERGE子句中的 OpenCypher 錯誤。
- 修復了一個事務錯誤，即使對應的並發字典插入被回滾，會話也可以插入圖形數據並提交。
- 已修正大量載入程式錯誤，此錯誤會在重度插入負載下造成效能回歸。
- 修正處理OPTIONAL子句中包含(NOT) EXISTS的查詢時的 SPARQL 錯誤，其中在某些情況下會遺失查詢結果。(
- 修正驅動程式在評估開始前逾時而取消要求時，可能會出現當機的錯誤。如果要求佇列中的項目逾時，伺服器上的所有查詢處理執行緒都已耗用，則可能會進入此狀態。由於請求隊列中的超時並未立即發送消息，因此響應似乎向客戶端保持待處理狀態。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.2.0.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.2.0.0.R2 的路徑

如果您執行的是引擎版本 1.2.0.0，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您只能從引擎版本 1.1.1.0 的最新修補程式版本1.2.0.0手動升級至發行版本 (p. 894)。先前的引擎發行版本必須先升級至的最新版本，1.1.1.0才能升級至1.2.0.0。

如果您先升級為發行版，1.1.1.0然後立即升級至1.2.0.0，您可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the instance
before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

升級至此版本

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine neptune \
    --engine-version 1.2.0.0 \
    --allow-major-version-upgrade \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine neptune ^
    --engine-version 1.2.0.0 ^
    --allow-major-version-upgrade ^
    --apply-immediately
```

而不是`--apply-immediately`，您可以指定`--no-apply-immediately`。若要執行主要版本升級，需要此allow-major-version-upgrade參數。此外，請務必包含引擎版本，否則您的引擎可能已升級到其他版本。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
    --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
    --db-instance-parameter-group-name (name of the custom instance parameter group)
```

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 4 月 19 日，引擎版本 1.1.1.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

此版本的後續修補程式版本

- 維護版本:1.1.1.2 版本 (p. 912)
- 發行版本: (p. 910)
- 發行版本: (p. 907)
- 版本發行日期: (p. 904)

- 發行版本: (p. 901)
- 發行版本: (p. 898)

這個引擎版本的新功能

- OpenCypher 查詢語言 (p. 382)現在已通常可供生產使用。

  Warning

  對於使用 OpenCypher 搭配 IAM 身分驗證的程式碼，此版本中有一項重大變更。在 OpenCypher 的海王星預覽中，IAM 簽名中的主機字串包含通訊協定`bolt://`，如下所示：

  ```
  "Host":"bolt://(host URL):(port)"
  ```

  從此引擎發行版本開始，必須省略通訊協定：

  ```
  "Host":"(host URL):(port)"
  ```

  如需範例，請參閱 使用博爾特通訊協定 (p. 389)。
- 增加了對 TinkerPop`3.5.2`. 在這個版本中的變化是支持遠程事務和字節碼對會話（使用 `g.tx`）的支持，以及該`datetime()`函數的增加到 Grimlin 語言。

  Warning

  TinkerPop3.5.0、3.5.1 和 3.5.2 中引入了一些突破性變更，這些變更可能會影響您的小鬼程式碼。例如，使用由這樣的孩子產生的遍歷將不再起GraphTraversalSource作用：。`g.V().union(identity(), g.V())`
  現在，請使用像這樣的匿名遍歷：。`g.V().union(identity(), __.V())`
- 已新增對AWS全域條件金鑰 (p. 210)的支援，您可以在 IAM 資料存取政策中使用這些 (p. 187)金鑰，以控制對 Neptune 資料庫叢集中儲存之資料的存取。
- 海王星 DFE 查詢引擎 (p. 129)現在通常可用於與 OpenCypher 查詢語言一起生產使用，但尚未用於格林和 SPARQL 查詢。現在，您可以使用它自己的海王星查詢引擎 (p. 533)例證參數而不是實驗室模式參數來啟用它。

這個引擎版本的改善項目

- 新增 OpenCypher (p. 382) 的新功能，例如參數化查詢支援、參數化查詢的抽象語法樹 (AST) 快取、可變長度路徑 (VLP) 改進，以及新的運算子和子句。如需目前亞馬遜海王星中的開放密碼標準合規性 (p. 95)的語言支援層級，請參閱。
- 對 OpenCypher 進行了顯著的性能改進，用於簡單的讀取和寫入工作負載，與版本 1.1.0.0 相比，產生了更高的吞吐量。
- 移除 OpenCypher 處理可變長度路徑的雙向和深度限制。
- 完成對 Gremlin `within` 和`without`述詞的 DFE 引擎支援，包括它們與其他謂詞運算子結合的案例。例如：

  ```
  g.V().has('age', within(12, 15, 18).or(gt(30)))
  ```
- 當範圍為全域 (也就是不是`order(local)`)，以及未使用`by()`調變器時，針對 Gemlin `order` 步驟的 DFE 引擎擴充支援。例如，此查詢現在可支援 DFE：

  ```
  g.V().values("age").order()
  ```

- 在 Neptune 串流變更記錄回應格式中新增isLastOp欄位，以指出記錄是其交易 (p. 588)中的最後一個作業。
- 大幅改善稽核記錄的效能，並在啟用稽核記錄時減少延遲。
- 在審計日誌中將 Gremlin WebSocket 字節碼和 HTTP 查詢轉換為用戶可讀的格式。查詢現在可以直接從要在 Neptune 筆記本和其他地方執行的稽核記錄複製。請注意，這項對目前稽核記錄格式的變更會構成重大變更。

此引擎版本中修正的缺陷

- 修正了一個罕見的 Gemlin 錯誤，該錯誤在使用嵌套filter()和count()步驟組合時沒有返回結果，例如在以下查詢中：

```
g.V("1").filter(out("knows")
        .filter(in("knows")
        .hasId("notExists"))
        .count())
```

- 修復了 Gremlin 錯誤，該錯誤在使用聚合步驟中存儲的頂點to()或與步驟一起from()遍歷時返回錯誤。addE這種查詢的一個例子是：

```
g.V("id").aggregate("v").out().addE().to(select("v").unfold()))
```

- 修復了使用 DFE 引擎時，not步驟在邊緣情況下失敗的錯誤。例如：

```
g.V().not(V())
```

- 修復了一個小鬼錯誤，其中sideEffect值在內部to()和from()遍歷不可用。
- 修正偶爾導致快速重設觸發執行個體容錯移轉的錯誤。
- 修正大量載入程式錯誤，其中失敗的交易在開始下一個載入工作之前不會關閉。
- 修正大量載入程式錯誤，其中記憶體不足的情況可能會導致系統當機。
- 已新增重試，以修正載入程式沒有等待足夠長的時間，讓 IAM 登入資料在容錯移轉後可用的大量載入程式錯誤。
- 修正未針對非查詢端點 (例如端點) 正確清除內部認證快取的錯誤。status
- 修正了一個串流錯誤，以確保串流提交序列號是否正確排序。
- 修正在啟用 IAM 的叢集上，長時間執行連線會在十天內終止的錯誤。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.1.1.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

引擎版本 1.1.1.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。請注意，主要版本引擎 (1.1.0.0) 之前的版本將需要更長的時間才能升級到此發行版本。

您將不會自動升級至此版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*(autogenerated snapshot ID)*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine neptune \
    --engine-version 1.1.1.0 \
    --allow-major-version-upgrade \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine neptune ^
    --engine-version 1.1.1.0 ^
    --allow-major-version-upgrade ^
    --apply-immediately
```

而不是`--apply-immediately`，您可以指定`--no-apply-immediately`。若要執行主要版本升級，需要此allow-major-version-upgrade參數。此外，請務必包含引擎版本，否則您的引擎可能已升級到其他版本。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
    --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
--db-instance-parameter-group-name (name of the custom instance parameter group)
```

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> Note
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二三年一月二十三日，引擎版本 1.1.1.0.R7 現正普遍部署。請注意，新版本需要數天才能在每個區域推出。

> Important
>
> 從舊版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
> 為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。

在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

這個引擎版本的改善項目

- 改進了涉及MERGE和的開放密碼查詢的性能。OPTIONAL MATCH
- 改進了涉及文字值映射列表UNWIND的 OpenCypher 查詢的性能。
- 改善具有IN篩選條件的 OpenCypher 查詢的效能。id例如：

```
MATCH (n) WHERE id(n) IN ['1', '2', '3'] RETURN n
```

- 各種 Gemlin 運算子的效能改善和正確性修正，包括repeat、、和. coalesce store aggregate

此引擎版本中修正的缺陷

- 修復了 OpenCypher 錯誤，其中使用 HTTP 保持活動的請求如果在失敗的請求後提交，則可能會錯誤地關閉它。
- 修復了 OpenCypher 錯誤，其中參數類型並不總是正確解釋為列表或地圖列表。
- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串"null"，而不是博爾特和 SPARQL-JSON 中的空值。
- 修復了 OpenCypher 錯誤代碼和查詢超時失敗和out-of-memory錯誤的錯誤消息。
- 修正了在 DFE 引擎的by()遍歷下valueMap()無法最佳化的小精靈錯誤。
- 修正鎖死偵測器邏輯偶爾會導致引擎沒有回應的問題。
- 修正稽核記錄錯誤，導致記錄不必要的資訊，以及記錄中遺失某些欄位。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.1.0.R7 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.3
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.1.0.R7 的路徑

如果您執行的是引擎版本 1.1.1.0，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 9 月 23 日，引擎版本 1.1.1.0.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

> **Important**
>
> 從舊版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
> 為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
> 在升級開始時，Neptune 會產生一個快照，其名稱由`preupgrade`後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
> 當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
> 此程序會產生下列事件：
>
> - 每個叢集事件訊息：
>   - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`

    - `Database cluster major version has been upgraded`
  - 每個執行個體事件訊息：
    - `Applying off-line patches to DB instance`
    - `DB instance shutdown`
    - `Finished applying off-line patches to DB instance`
    - `DB instance restarted`

    Note

    對於使用 OpenCypher 和 IAM 身份驗證的代碼，此版本中有一個突破性的變化。到目前為止，IAM 簽名中的主機字符串包含協議，例如`bolt://`，如下所示：

    ```
    "Host":"bolt://(host URL):(port)"
    ```

    從引擎釋放開始`1.1.1.0`，必須省略協議：

    ```
    "Host":"(host URL):(port)"
    ```

    如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 改進了小鬼查詢`order-by`的性能。在 a 結尾的 Gemlin 查詢`NeptuneGraphQueryStep`現在使用較大的塊大小以獲得更好的性能。`order-by`這不適用`order-by`於查詢計劃的內部（非根）節點。
- 改進了小鬼更新查詢的性能。在加入邊或屬性時，必須鎖住頂點和邊線以防刪除。此變更可消除交易中的重複鎖定，進而改善效能。

此引擎版本中修正的缺陷

- 修復了在某些情況下導致重複節點和邊緣創建的`MERGE`子句中的 OpenCypher 錯誤。
- 修正處理`OPTIONAL`子句中包含`(NOT) EXISTS`的 SPARQL 查詢時，在某些情況下會遺失查詢結果的錯誤。
- 修正在進行大量載入時延遲伺服器重新啟動的錯誤。
- 修復了 OpenCypher 可變長度模式雙向遍歷與關係屬性上的濾波器會導致錯誤的錯誤。這種可變長度陣列的範例為。`(n)-[r*1..2]->(m)`
- 修正與快取資料如何傳回用戶端相關的錯誤，在某些情況下會導致非預期的長延遲。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.1.1.0.R6 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.1.0.R6 的路徑

如果您執行的是引擎版本 `1.1.1.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一年七月二十一日，引擎版本 1.1.1.0.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
在升級開始時，Neptune 會產生一個快照，其名稱由`preupgrade`後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`

- `DB instance restarted`

Note

對於使用 OpenCypher 搭配 IAM 身分驗證的程式碼，此版本中有一項重大變更。到目前為止，IAM 簽名中的主機字符串包含協議，例如`bolt://`，如下所示：

```
"Host":"bolt://(host URL):(port)"
```

從引擎釋放開始`1.1.1.0`，必須省略協議：

```
"Host":"(host URL):(port)"
```

如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 進行了改進，以支持死鎖檢測。

此引擎版本中修正的缺陷

- 修正在特定情況下無法正常關閉資料庫叢集的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.1.0.R5 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.1.0.R5 的路徑

如果您執行的是引擎版本 `1.1.1.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由`preupgrade`後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：

    - Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-*(autogenerated snapshot ID)*]
    - Database cluster major version has been upgraded
  - 每個執行個體事件訊息：
    - Applying off-line patches to DB instance
    - DB instance shutdown
    - Finished applying off-line patches to DB instance
    - DB instance restarted

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開preupgrade頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> ```

```
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 6 月 23 日，引擎版本 1.1.1.0.R4 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

對於使用 OpenCypher 和 IAM 身份驗證的代碼，此版本中有一個突破性的變化。到目前為止，IAM 簽名中的主機字符串包含協議，例如`bolt://`，如下所示：

```
"Host":"bolt://(host URL):(port)"
```

從引擎釋放開始`1.1.1.0`，必須省略協議：

```
"Host":"(host URL):(port)"
```

如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 更新執行個體類型的x2g執行個體組態
- 改善頂點掉落的效能。

此引擎版本中修正的缺陷

- 修正了 Gemlin 錯誤，其中解決方案無法維持多次查詢的穩定順序，或是針對特定種類的 ASK 聯結在多個讀取器之間維持穩定順序。
- 此外，縮小了之前版本中導致 Gremlin 中某些類型的 ASK 連接的性能回歸的更改範圍。
- 修復了在子遍歷中出現邊緣輸入和遍歷到頂點時發生的union()步驟中的 Grimlin 錯誤。
- 修正了一個小鬼配置文件錯誤，其中一些步驟被報告為實際上沒有優化。
- 修復了 SPARQL 錯誤，其中嵌套到UNION子句中的FILTER運算式中使用的變量被分配無效範圍信息。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.1.0.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.1.0.R4 的路徑

如果您執行的是引擎版本 1.1.1.0，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-*(autogenerated snapshot ID)*]
  - Database cluster major version has been upgraded
- 每個執行個體事件訊息：
  - Applying off-line patches to DB instance
  - DB instance shutdown
  - Finished applying off-line patches to DB instance

- DB instance restarted

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開preupgrade頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 6 月 07 日，引擎版本 1.1.1.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由`preupgrade`後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

對於使用 OpenCypher 和 IAM 身份驗證的代碼，此版本中有一個突破性的變化。到目前為止，IAM 簽名中的主機字符串包含協議，例如`bolt://`，如下所示：

```
"Host":"bolt://(host URL):(port)"
```

從引擎釋放開始`1.1.1.0`，必須省略協議：

```
"Host":"(host URL):(port)"
```

如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

這個引擎版本的改善項目

- 新增對採用 Graviton2 執行個體類型的支援，並針對記憶x2g體密集型工作負載進行最佳化。這些最初只有四種可用AWS 區域：
  - 美國東部 (維吉尼亞北部) (`us-east-1`)
  - 美國東部 (俄亥俄) (`us-east-2`)
  - 美國西部 (奧勒岡) (`us-west-2`)
  - 歐洲 (愛爾蘭) (`eu-west-1`)

  如需詳細資訊，請參閱 Neptune 定價頁面。
- 改進了涉及多個邊緣或頂點遍歷，屬性查找或標籤查找的 Gemlin 步驟的性能。

此引擎版本中修正的缺陷

- 修正了一個小鬼在子遊otherV()走步驟的處理中的錯誤。
- 修復了僅union具有過濾器步驟的查詢中的一個小鬼錯誤。例如：

```
g.V().union(has("name"), out("knows")).out()
```

- 修復了 SPARQL 錯誤，其中嵌套到UNION子句中的FILTER運算式中使用的變量被分配無效範圍信息。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.1.1.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：3.5.2
- 小精靈最新版本支持：3.5.4
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.1.0.R3 的路徑

如果您執行的是引擎版本 1.1.1.0，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

> **Important**
>
> 從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
> 在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
> 當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
> 此程序會產生下列事件：
>
> - 每個叢集事件訊息：
>   - Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-*(autogenerated snapshot ID)*]
>   - Database cluster major version has been upgraded
> - 每個執行個體事件訊息：
>   - Applying off-line patches to DB instance
>   - DB instance shutdown
>   - Finished applying off-line patches to DB instance
>   - DB instance restarted

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星維護版本，版本 1.1.1.0.R2

截至 2022 年 5 月 16 日，引擎版本 1.1.1.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

對於使用 OpenCypher 搭配 IAM 身分驗證的程式碼，此版本中有一項重大變更。到目前為止，IAM 簽名中的主機字符串包含協議，例如`bolt://`，如下所示：

```
"Host":"bolt://(host URL):(port)"
```

從引擎釋放開始`1.1.1.0`，必須省略協議：

```
"Host":"(host URL):(port)"
```

如需範例，請參閱 使用博爾特通訊協定 (p. 389)。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.1.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈最早的版本支持：`3.5.2`
- 小精靈最新版本支持：`3.5.4`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.1.0.R2 的路徑

如果您執行的是引擎版本，您的叢集將在下一個維護時段期間自動升級至此維護修補程式版本`1.1.1.0`。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.1.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.1.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一九年一月十一日，引擎版本 1.1.0.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

> **Important**
>
> 從舊版本升級至此引擎版本，**`1.1.0.0`**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
> 為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
> 在升級開始時，Neptune 會產生一個快照，其名稱由`preupgrade`後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
> 當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
> 此程序會產生下列事件：
>
> - 每個叢集事件訊息：
>   - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
>   - `Database cluster major version has been upgraded`

- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

從此引擎版本開始，Neptune 不再支援R4執行個體類型 (p. 53)。如果您在資料庫叢集中使用R4執行個體，則在升級至此版本之前，必須手動將其取代為不同的執行個體類型。如果您的寫入器執行個體是R4，請依照下列指示 (p. 826)移動它。

此版本的後續修補程式版本

- 維護版本: (p. 921)
- 維護版本: (p. 919)

這個引擎版本的新功能

- 推出由 Graviton2 處理器提供支援的一般用途T4g和記憶體最佳化R6g資料庫執行個體。AWS與目前同級 x86 型執行個體相比，針對各種工作負載，以 Graviton2 為基礎的執行個體提供明顯更優異的價格/效能。應用程式在這些新的執行個體類型上正常運作，而且升級至這些執行個體類型時不需要移植應用程式程式碼。

  如需有關定價和區域可用性的詳細資訊，請參閱 Amazon Neptune 定價頁面。
- 在海王星 ML 中引入了自定義模型 (p. 724)。
- 已新增對海王星 ML 中 SPARQL 推論查詢 (p. 748)的支援。
- 為屬性圖數據添加了新的流端點 (p. 587)，即：

```
https://Neptune-DNS:8182/propertygraph/stream
```

  此端點的輸出格式PG_JSON，命名為，與舊的GREMLIN_JSON格式輸出完全相同gremlin/stream。

  新propertygraph/stream端點將海王星串流支援延伸至 OpenCypher，並以其相關聯的GREMLIN_JSON輸出格式取代gremlin/stream端點。

這個引擎版本的改善項目

- 對海王星流進行了改進：
  - 在 Neptune 串流變更記錄回應格式的records物件中新增commitTimestamp欄位，以便為變更記錄 (p. 588)資料流中的每個記錄提供時間戳記。
  - 已新增LATEST值至iteratorType參數，可讓您從串流擷取最後一個有效的 EventID。請參閱 呼叫串流 API (p. 587)。
- 已新增在 Gremlin 節點分類和回歸查詢中取得推論置信度 (p. 735)分數的支援。
- 增加了對開放密OPTIONAL MATCH碼子句的支持。
- 增加了對開放密MERGE碼子句的支持。
- 增加了對ORDER BY在WITH子句中使用的支持。
- 增加了 OpenCypher 中模式理解的支持，以及對超出存在檢查的模式表達式的擴展支持。

- 對 OpenCypher 中的DELETE和DELETE DETACH子句的擴展支持，以便它們現在可以與其他更新子句一起使用。
- 擴展的支持CREATE和UPDATE子句RETURN在開放密碼器中使用。
- 在 DFE 引擎中新增了對小精靈和limit步驟的支援。range skip
- 改進了 DFE 引擎中的查詢執行，當兩者都explain不profile被請求時。
- 已改善 DFE 引擎中value運算式的查詢執行。
- 改進了一些鏈接的 Gemlin 條件插入模式，以避免並發修改異常，並允許鏈接查詢模式，如下所示：
  - 依 ID 插入條件式頂點，例如：

```
g.V(ID).fold().coalesce(unfold(), g.addV("L1").property(id,ID))
```

  - 帶有多個標籤的條件頂點插入，例如：

```
g.V(ID).fold().coalesce(unfold(), g.addV("L1::L2").property(id,ID))
```

  - 依 ID 的條件式邊插入，例如：

```
g.E(ID).fold().coalesce(unfold(), V(from).addE(label).to(V(to)).property(id, ID))
```

  - 帶有多個標籤的條件式邊緣插入，例如：

```
g.E(ID).fold().coalesce(unfold(),
 g.addE(label).from(V(from)).to(V(to)).property(id, ID))
```

  - 條件式插入後跟一個查詢，例如：

```
g.V(ID).fold().coalesce(unfold(),
 g.addV("L1").property(id,ID)).project("myvalues").by(valueMap())
```

  - 帶有添加屬性的條件插入，例如：

```
g.V(ID).fold().coalesce(unfold(),
 g.addV("L1").property(id,ID).property("name","pumba"))
```

此引擎版本中修正的缺陷

- 停用無法支援T3.medium執行個體類型的統計資料 (p. 131)功能。
- 已修正explain具有非常數值的IN函數中的 SPARQL 錯誤。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.1.0.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.11
- 版本：1.1

引擎版本 1.1.0.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您將不會自動升級至此版本。

升級至此版本

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine neptune \
    --engine-version 1.1.0.0 \
    --allow-major-version-upgrade \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine neptune ^
    --engine-version 1.1.0.0 ^
    --allow-major-version-upgrade ^
    --apply-immediately
```

而不是--apply-immediately，您可以指定--no-apply-immediately。若要執行主要版本升級，需要此allow-major-version-upgrade參數。此外，請務必包含引擎版本，否則您的引擎可能已升級到其他版本。

如果您的叢集使用自訂叢集參數群組，請務必包含此參數以指定它：

```
    --db-cluster-parameter-group-name (name of the custom DB cluster parameter group)
```

同樣地，如果叢集中有任何執行個體使用自訂資料庫參數群組，請務必包含此參數以指定它：

```
    --db-instance-parameter-group-name (name of the custom instance parameter group)
```

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開preupgrade頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星維護版本，版本 1.1.0.0.R3

截至 2022 年 12 月 23 日，引擎版本 1.1.0.0.R3 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
為了順利完成升級，每個可用區域 (AZ) 中的每個子網路必須至少有一個可用的 IP 位址，每個 Neptune 執行個體。例如，如果子網路 1 中有一個寫入器執行個體和兩個讀取器執行個體，子網路 2 中有兩個讀取器執行個體，則子網路 1 必須至少有 3 個可用的 IP 位址，而子網路 2 必須至少有 2 個可用的 IP 位址，才能開始升級。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-`*`(autogenerated snapshot ID)`*`]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

這個引擎版本的改善項目

- 各種 Gemlin 運算子的效能改善和正確性修正，包括`repeat`、、和. `coalesce` `store` `aggregate`

此引擎版本中修正的缺陷

- 修正 CPU 尖峰問題。

- 修復了一個 OpenCypher 錯誤，其中查詢返回字符串"null"，而不是在博爾特和 SPARQL-JSON 中的空值。
- 修正稽核記錄錯誤，導致記錄不必要的資訊，以及記錄中遺失某些欄位。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.0.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.11
- 開放密碼版本：Neptune-9.0.20190305-1.0
- 版本：1.1

升級至引擎版本 1.1.0.0.R3 的路徑

如果您執行的是引擎版本，您的叢集將在下一個維護時段期間自動升級至此維護修補程式版本1.1.0.0。

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

從此引擎版本開始，Neptune 不再支援R4執行個體類型 (p. 53)。如果您在資料庫叢集中使用R4執行個體，則在升級至此版本之前，必須手動將其取代為不同的執行個體類型。如果您的寫入器執行個體是R4，請依照下列指示 (p. 826)移動它。

升級至此版本

亞馬遜海王星 1.1.0.0.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.0.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.0.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星維護版本，版本 1.1.0.0.R2

截至 2022 年 5 月 16 日，引擎版本 1.1.0.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

Important

從舊版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用數分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

此引擎版本中修正的缺陷

- 修正未針對非查詢端點 (例如狀態端點) 正確清除內部認證快取的錯誤。
- 修正引擎升級後造成複寫延遲增加的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.1.0.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 開放密碼版本：`Neptune-9.0.20190305-1.0`
- 版本：`1.1`

升級至引擎版本 1.1.0.0.R2 的路徑

如果您執行的是引擎版本，您的叢集將在下一個維護時段期間自動升級至此維護修補程式版本`1.1.0.0`。

Important

從之前的任何版本升級至此引擎版本，**1.1.0.0**也會在資料庫叢集中的所有執行個體上觸發作業系統升級。由於不會處理作業系統升級期間發生的使用中寫入要求，因此在開始升級之前，您必須暫停對要升級之叢集的所有寫入工作負載 (包括大量資料載入)。
在升級開始時，Neptune 會產生一個快照，其名稱由preupgrade後面接著根據您的資料庫叢集資訊自動產生的識別碼組成。您不需要為此快照付費，如果在升級過程中出現任何問題，您可以使用它來還原資料庫叢集。
當引擎升級完成後，新的引擎版本將在舊的作業系統上短暫提供，但不到 5 分鐘的時間內，叢集中的所有執行個體都會同時開始作業系統升級。您的資料庫叢集目前將無法使用大約 6 分鐘。升級完成後，您可以繼續寫入工作負載。
此程序會產生下列事件：

- 每個叢集事件訊息：
  - `Upgrade in progress: Creating pre-upgrade snapshot [preupgrade-(autogenerated snapshot ID)]`
  - `Database cluster major version has been upgraded`
- 每個執行個體事件訊息：
  - `Applying off-line patches to DB instance`
  - `DB instance shutdown`
  - `Finished applying off-line patches to DB instance`
  - `DB instance restarted`

Note

從此引擎版本開始，Neptune 不再支援R4執行個體類型 (p. 53)。如果您在資料庫叢集中使用R4執行個體，則在升級至此版本之前，必須手動將其取代為不同的執行個體類型。如果您的寫入器執行個體是R4，請依照下列指示 (p. 826)移動它。

升級至此版本

亞馬遜海王星 1.1.0.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.1.0.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.1.0.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一年十月一日，引擎版本 1.0.5.1 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本: (p. 929)
- 發行版本: (p. 928)
- 維護版本:4 月 5 日 (p. 926)

這個引擎版本的新功能

- 新增了用於快取 (p. 330)指定查詢結果的結果快取。
- 在海王星開放密碼中添加日期/時間支持。
- 增加了對海王星開放密碼器中元素的支持`List`和`Map`訪問。

這個引擎版本的改善項目

- 使海王星 OpenCypher 端點名稱不區分大小寫。
- 改進的開放密碼解釋。
- 改進了小鬼單一 upsert 查詢模式終止與步驟。`iterate()` `profile()`
- 改進了小鬼`keys()`和`property()`功能的性能。
- 當它與全局範圍使用的小鬼`dedup()`步驟在 DFE 中運行。
- 當 DFE 引擎啟用時，HAS會在 DFE 引擎中執行下列格林述詞：
  - `EQ`
  - `NEQ`

  - LT
  - LTE
  - GT
  - GTE
  - BETWEEN
  - INSIDE
  - OUTSIDE
  - WITHIN
  - AND (connectives)
  - OR (connectives)
- 已改善 LIMIT 查詢效能。
- 改善了一般彙總查詢的效能。

此引擎版本中修正的缺陷

- 修正了一個允許邊緣連接到另一個邊緣的小鬼錯誤。
- 修正了導致選擇次最佳加入策略的小鬼錯誤。
- 修正了當存在超過 100 個屬性時，導致節點序列化和關係停止的 Gemlin 錯誤。
- 修正拖慢具有大圖形模式查詢的查詢執行規劃速度的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.5.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.11
- 版本：1.1

引擎版本 1.0.5.1 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星維護版本，版本 1.0.5.1.R4

截至 2022 年 5 月 16 日，引擎版本 1.0.5.1.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.5.1.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

升級至引擎版本 1.0.5.1.R4 的路徑

如果您執行的是引擎版本，您的叢集將在下一個維護時段期間自動升級至此維護修補程式版本1.0.5.1。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.1.R4 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開preupgrade頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> ```

```
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2022 年 1 月 13 日，引擎版本 1.0.5.1.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正查詢無法取得所有需要的資源時，可能導致資源洩漏的錯誤。
- 修正由未宣告的記憶體配置造成的查詢執行期間的小記憶體洩漏問題。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.5.1.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

升級至引擎版本 1.0.5.1.R3 的路徑

如果您執行的是引擎版本 `1.0.5.1`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.1.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

截至二零二一年十月二十六日，引擎版本 1.0.5.1.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正在可重複讀取隔離下建立舊版圖形元素時發生暫時性錯誤時，導致伺服器重新啟動的錯誤。Neptune 現在會傳回錯誤，以便用戶端可以重試。
- 修正在單一基數更新期間發生暫時性錯誤時，導致伺服器重新啟動的錯誤。Neptune 現在會傳回錯誤，以便用戶端可以重試。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.5.1.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

升級至引擎版本 1.0.5.1.R2 的路徑

如果您執行的是引擎版本 `1.0.5.1`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.1.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> ```

```
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一年七月二十七日，引擎版本 1.0.5.0 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本: (p. 937)
- 發行版本: (p. 935)
- 維護版本:5 月 5 日 (p. 933)

這個引擎版本的新功能

- Neptune ML (p. 652) 已發佈以供生產使用，具有許多新功能，而且不再處於實驗室模式。
- 已在實驗室模式中新增對开放密码 (p. 382)查詢語言的初始支援。 開放密碼是用於密碼查詢語言的開源標準。它的語法是在密碼查詢語言參考（版本 9）中指定，並由 OpenCy pher 項目進行維護。

  如需有使用開放密碼訪問海王星圖 (p. 382)關 Neptune 實作語言的資訊，請參閱。

  也支援海王星用戶端用於 OpenCypher 查詢的螺栓通訊協定的支援。請參閱 使用博爾特協議向海王星進行開放密碼查詢 (p. 389)。

  OpenCypher 的支援現在會自動啟用，但這取決於海王星 DFE 发动机 (p. 129)，目前僅在實驗室 (p. 127)模式中可用。neptune_lab_modeDB 叢集參數中的預DFEQueryEngine設設定現在為DFEQueryEngine=viaQueryHint，這表示引擎已啟用，但僅用於存在useDFE查詢提示並設定為的查詢true。如果您透過設定來停用 DFE 引擎DFEQueryEngine=disabled，您將無法使用開放式密碼器。
- 已新增對 SPARQL 1.1 圖形儲存區 HTTP 通訊協定的支援。請參閱 在亞馬遜海王星中使用 SPARQL 1.1 圖形存放區 HTTP 通訊協定 (GSP) (p. 452)。
- 將預設的實驗室模式設定變更海王星 DFE 发动机 (p. 129)為viaQueryHint，這表示 DFE 引擎現在預設為啟用，但僅用於存在useDFE查詢提示並設定為的查詢。true
- 添加了一個新的亞馬遜CloudWatch指標StatsNumStatementsScanned，用於監控海王星 DFE 引擎的統計數據的計算。請參閱 使用StatsNumStatementsScannedCloudWatch測量結果監督統計資料運算 (p. 134)。

這個引擎版本的改善項目

- 增加了對阿帕奇 TinkerPop 3.4.11 的支持。

  Important

  在TinkerPop版本 3.4.11 中進行了更改，可提高查詢處理方式的正確性，但目前有時會嚴重影響查詢性能。

例如，此類查詢的執行速度可能會明顯變慢：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  out()
```

極限步驟之後的頂點現在會以 3.4.11 變更的非最佳方式擷取。TinkerPop若要避免這種情況，您可以在之後的任何點加入 barrier () 步驟來修改查詢`order().by()`。例如：

```
g.V().hasLabel('airport').
  order().
    by(out().count(),desc).
  limit(10).
  barrier().
  out()
```

- 現在，海王星 DFE 替代`joinOrder`查詢引擎支援 SPARQL 查詢提示 (p. 438)。
- Neptune 狀態 API (p. 770) 的輸出已擴充並重新組織，以提供有關資料庫叢集的設定和功能的更清晰。

  新輸出具有頂層`features`物件，其中包含有關資料庫叢集功能的狀態資訊，以及包含設定資訊的頂層`settings`物件。若要檢閱新格式，請參閱實例 status 命令輸出的示例 (p. 772)。
- 當使用伺服器上的最後一個事件 ID 請求`AFTER_SEQUENCE_NUMBER`串流時，事件識別碼已過期時，已改善串流變更記錄的處理。如果要求的事件 ID 是伺服器上最近清除的事件 ID，則伺服器不會再擲回到期的事件識別碼錯誤。

此引擎版本中修正的缺陷

- 修復了與數值排序相關的小鬼錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.5.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

引擎版本 1.0.5.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.0 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星維護版本，版本 1.0.5.0.R5

截至 2022 年 5 月 16 日，正在普遍部署引擎版本。請注意，新版本需要數天才能在每個區域推出。

此發行版本支援的查詢語言版本

將資料庫叢集升級到 1.0.5.0.R5 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.11
- 版本：1.1

升級至引擎版本 1.0.0.R5 的路徑

如果您執行的是引擎版本 1.0.5.0，您的叢集將在下一個維護時段期間自動升級至此維護修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.0.R5 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎版

截至二零二一九年九月十五日，引擎版本 1.0.5.0.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正在下列任一情況下，引擎無回應的錯誤：
  - 批量負載發生在自動統計計算正在進行的同時。
  - 統計計算是在已經發生的同一時間手動請求。
- 修正鎖死偵測和鎖定擷取中可能導致引擎當機的錯誤。
- 修正 Gemlin 錯誤，其中引擎在 Gremlin 推論查詢中遇到來自遠端 ML 端點的未知資料時擲回錯誤。
- 修正 ML 模型管理 API 中與模型轉換工作和執行個體建議相關的數個錯誤。(
- 修正產生節點和邊緣 ID 時可能導致引擎當機的錯誤。
- 修正使用大型圖形模式查詢產生查詢計畫的速度變慢的錯誤。
- 修復了 OpenCypher 錯誤，該錯誤在檢索具有 100 個以上屬性的節點時可能導致查詢停止。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.0.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

升級至引擎版本 1.0.0.R3 的路徑

如果您執行的是引擎版本 1.0.5.0，您的叢集將在下一個維護時段期間自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.0.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

截至二零二一年八月十六日，引擎版本 1.0.5.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 停用在引擎版 (p. 931)本中進行的最佳化，`1.0.5.0`使 Neptune 查閱快取 (p. 107)在複本上重新啟動引擎仍然存在。複本重新啟動現在會清除查閱快取。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.0.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.11`
- 版本：`1.1`

升級至引擎版本 1.0.0.R2 的路徑

如果您執行的是引擎版本 `1.0.5.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.5.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.5.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.5.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

Amazon Neptune 引擎版本

> **Note**
>
> 發動機發布版本 1.0.4.2.R2 是 1.0.4.2 的第一個版本實際上被釋放。

主題

- 亚马逊海王星发动机 (p. 938)
- 亞馬遜海王星引擎 (p. 939)
- 亚马逊海王星发动机 (p. 939)
- 亚马逊海王星发动机 (p. 940)
- Amazon Neptune 引擎版本 (p. 943)

亚马逊海王星发动机

截至二零二一年八月十六日，引擎版本 1.0.4.2.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 停用在引擎版 (p. 939)本中進行的最佳化，`1.0.4.2.R4`使 Neptune 查閱快取 (p. 107)在複本上重新啟動引擎仍然存在。複本重新啟動現在會清除查閱快取。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.2.R5 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.10`
- 版本：`1.1`

升級至引擎版本 1.0.4.2.R5 的路徑

如果您執行的是引擎版本 `1.0.4.2`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

亞馬遜海王星引擎

截至二零二一年七月二十三日，引擎版本 1.0.4.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

這個引擎版本的改善項目

- 改善查閱快取的行為，以避免在複本上執行快速重設之後清除冗餘快取。
- 改進了當使用服務器上的最後一個事件 ID 請求`AFTER_SEQUENCE_NUMBER`流時，該事件 ID 已過期時對流更改日誌的處理。如果要求的事件 ID 是伺服器上最近清除的事件 ID，則伺服器不會再擲回到期的事件識別碼錯誤。

此引擎版本中修正的缺陷

- 修正 1.0.4.0.R1 中引入的錯誤，其中查詢不會傳回整個大於 760 個字元的字串值。受此錯誤影響的術語是 RDF 常值和 URI，或 Grimlin ID，鍵和字符串值。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.2.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.10`
- 版本：`1.1`

升級至引擎版本 1.0.4.2.R4 的路徑

如果您執行的是引擎版本 `1.0.4.2`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

亚马逊海王星发动机

截至二零二一年六月二十八日，引擎版本 1.0.4.2.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中的已知問題

問題：

如果存在空格，則無法在Accept標頭中遵循媒體類型的 SPARQL 錯誤。

例如，具有 `-H "Accept: text/csv; q=1.0, */*; q=0.1"` 傳回 JSON 輸出而非 CSV 輸出的查詢。

因應措施：

如果您移除標題中Accept子句中的空格，引擎會以正確的要求格式傳回輸出。換句話說，而不是使用： `-H "Accept: text/csv; q=1.0, */*; q=0.1"`

```
-H "Accept: text/csv;q=1.0,*/*;q=0.1"
```

此引擎版本中修正的缺陷

- 修正快速重設後清除複本上查閱快取的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.2.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.10`
- 版本：`1.1`

升級至引擎版本 1.0.4.2.R3 的路徑

除非您的資料庫叢集使用一或多個R5d執行個體，否則此修補程式版本為選用。如果您的叢集有R5d執行個體，系統會在下一個維護時段中自動升級該執行個體。否則，它將不會自動升級到此修補程式版本。

您可以使用AWS CLIapply-pending-maintenance-action指令(ApplyPendingMaintenanceAction (p. 1152)API) 手動`1.0.4.2.R2`將`1.0.4.2.R3`發行版本升級至此版本。

亚马逊海王星发动机

截至二零二一年六月一日起，引擎版本 1.0.4.2.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本: (p. 939)

此引擎版本中的已知問題

問題：

如果存在空格，則無法在Accept標頭中遵循媒體類型的 SPARQL 錯誤。

例如，具有 `-H "Accept: text/csv; q=1.0, */*; q=0.1"` 傳回 JSON 輸出而非 CSV 輸出的查詢。

因應措施：

如果您移除標題中Accept子句中的空格，引擎會以正確的要求格式傳回輸出。換句話說，而不是使用： `-H "Accept: text/csv; q=1.0, */*; q=0.1"`

```
-H "Accept: text/csv;q=1.0,*/*;q=0.1"
```

這個引擎版本的新功能

- 新增 R5d 執行個體類型，其中包含查閱快取，可在涉及大量屬性值或 RDF 常值查詢的使用案例中加速讀取速度。請參閱 海王星查閱快取可加速讀取查詢 (p. 107)。
- 新增了一個新的實驗室模式參數，該參數僅允許使用查詢提示在每個查詢的基礎上調用實驗 DFE 引擎。`useDFE`

這個引擎版本的改善項目

- 增加了對 TinkerPop 3.4.10 的支持。
- 增加了在發送 Gremlin 腳本請求時使用`withStrategies( )`配置步驟的支持。具體而言`SubgraphStrategy`，`PartitionStrategy`，`ReadOnlyStrategy`、`EdgeLabelVerificationStrategy`、和`ReservedKeysVerificationStrategy`都受支援。
- 新增查詢中間`V()`遍歷的最佳化功能。以前，這樣的遍歷在海王星中沒有優化。
- 增加了對 RFC 2141 URN 的支援，以用作批量負載的`baseUri`和`namedGraphUri`參數。

此引擎版本中修正的缺陷

- 修正了解析器中不正確的查詢被視為有效的小鬼錯誤。
- 修復了一個將`aggregate()`副作用展開`cap().unfold()`到 a `valueMap()` 會引發異常的小鬼錯誤。
- 修復了一個小鬼錯誤，其中一個步驟在一個`addV()`步`property()`驟後的某些步驟失敗並出現「無法轉換為字符串」錯誤。
- 修正了一個 Gremlin 錯誤，以防止某些條件式插入模式引發並行修改異常。
- 修復了 Gimlin 錯誤，以便查詢請求超時現在不能超過會話超過會話超過時間。
- 修正 SPARQL 錯誤，即當遠端伺服器無法使用時，使用「載入」或「卸載」的更新可能會失敗，並使用 HTTP 代碼 500 而非 HTTP 代碼 400。
- 修正了當使用`commitNum`或大於 32 位元帶正負號整數限制 (2,147,483,647) 的`opNum`值時，串流 API 呼叫失敗的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.2.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.10`
- 版本：`1.1`

升級至引擎版本 1.0.4.2.R2 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.4.2.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

Amazon Neptune 引擎版本

引擎版本 1.0.4.2.R1 從未部署過。

亞馬遜海王星引擎

截至 2020 年 12 月 08 日，引擎版本 1.0.4.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本: (p. 945)
- 發行版本: (p. 946)

  Important

  發行版本： (p. 950)對於與亞馬遜海王星的所有連接，都必須強制使用 TLS 1.2 和 HTTPS。但是，該版本中的一個錯誤允許 HTTP 連線和/或過期的TLS 連線繼續運作，對於先前設定資料庫叢集參數以防止強制執行 HTTPS 連線的客戶而言。
  此錯誤已在修補程式版本 1.0.4.0.R2 和 1.0.4. 1. (p. 946) R2 (p. 952) 中修正，但修正程式在自動安裝修補程式時造成未預期的連線失敗。基於這個原因，這兩個修補程式都已回復，而且只能手動安裝，讓您有機會更新 TLS 1.2 的設定。
  與海王星的所有連接都必須使用 SSL/TLS 會影響您與 Girmlin 控制台，小鬼驅動程序，小鬼蟒，.NET，NodeJS，REST API 的連接，以及負載平衡器連接。如果您到目前為止一直在使用 HTTP 或較舊的 TLS 版本，則在將系統更新為最新的修補程式之前，您必須更新相關的用戶端和驅動程式，並將程式碼變更為僅使用 HTTPS。

這個引擎版本的新功能

- 推出海王星 ML 功能，為亞馬遜海王星帶來強大的機器學習功能。請參閱 Amazon Neptune ML 用於圖表上的機器學習 (p. 652)。
- 新增自訂 SPARQL UNLOAD 作業，用於移除從遠端來源擷取的資料。請參閱 排列更新卸載 (p. 435)。

這個引擎版本的改善項目

- 優化了一些 Gemlin 條件插入模式，以避免並發修改異常。

此引擎版本中修正的缺陷

- 修正了一個 Gemlin 錯誤，該錯誤可能會導致使用該步驟的特定查詢模式丟失結果。`as()`
- 修復了在另一個`project()`步驟中使用嵌套步驟時可能會導致錯誤的 Gemlin 錯誤，例如。`union()`
- 修正了步驟中的一個小精靈錯誤。`project()`
- 修復了基於字符串的遍歷中的小鬼錯誤，這錯誤令步驟不起作用。`none()`
- 修復了基於字符串的遍歷中的 Gremlin 錯誤，這錯誤令空地圖不被支持作為步驟的爭論者。`inject()`
- 修正 DFE 引擎中以字串為基礎的遍歷執行中的 Gremlin 錯誤，其中終端機方法 (例如) 無法正常運作。`toList()`
- 修復了無法關閉使用字符串查詢`iterate()`步驟的事務的 Gemlin 錯誤。
- 修復了在某些情況下可能導致使用該`is(P.gte(0))`模式的查詢拋出異常的 Gemlin 錯誤。

- 修復了在某些情況下可能導致使用該`order().by(T.id)`模式的查詢拋出異常的 Gemlin 錯誤。
- 修復了在某些情況下可能導致使用該`addV().aggregate()`模式的查詢產生不正確的結果的 Gemlin 錯誤。
- 修正了一個 Gemlin 錯誤，該錯誤可能導致使用`path()`步驟後跟步`project()`驟模式在某些情況下拋出異常的查詢。
- 修復了 SPARQL 錯誤，其中SUBSTR函數發出錯誤信號而不是返回空字符串。
- 修正 DFE 引擎中可能導致非封鎖查詢計畫中的聯結作業在未繫結變數存在時產生不正確結果的錯誤。(

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.8`
- 版本：`1.1`

引擎版本 1.0.4.1 的升級路徑

如果您執行的是引擎版本 `1.0.4.1`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.4.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

截至二零二一年三月二十二日，引擎版本 1.0.4.1.R1.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 禁用了可以添加或附加到現有標籤和屬性的 Gemlin 條件插入模式的優化。

此發行版本支援的查詢語言版本

在將資料庫叢集升級至 1.0.4.1.R1.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.8`
- 版本：`1.1`

升級至引擎版本 1.0.4.1.R1.1 的路徑

如果您執行的是引擎版本 `1.0.4.1`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

亞馬遜海王星 1.0.4.1.R1.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎版本

截至二零二一年二月二十一日，引擎版本 1.0.4.1.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本:1.0.4.1.2.1 (p. 948)

這個引擎版本的新功能

- Neptune 現在支援以批量載入bzip2格式壓縮單一檔案。請參閱 載入資料格式 (p. 261)。

此引擎版本中修正的缺陷

- 修正了允許使用HTTP或更早版本的 TLS，而不是 TLS 1.2 連接到海王星HTTPS的錯誤。發行版本： (p. 950)

  Important

  必須將 SSL/TLS 用於與海王星的所有連接可能是一個突破性的變化。它會影響您與 Girmlin 控制台，小鬼驅動程序，小鬼蟒蛇，.NET，節點，REST API 以及負載平衡器連接的連接。如果您到目前為止一直在使用 HTTP 或較舊的 TLS 版本，則必須在安裝此 Patch 之前更新相關的用戶端和驅動程式，並將您的程式碼變更為僅使用 HTTPS。
- 修復了在某些情況下，發生時InternalFailureException被設置為響應代碼的小鬼錯誤。ConcurrentModificationException
- 修復了在某些情況下更新邊緣或頂點可能導致瞬態的 Gemlin 錯誤。InternalFailureException

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.1.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.8
- 版本：1.1

升級至引擎版本 1.0.4.1.R2 的路徑

如果您執行的是引擎版本 1.0.4.1，您的叢集將在下一個維護時段自動升級至此修補程式版本。

升級至此版本

亞馬遜海王星 1.0.4.1.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
```

```
    --engine-version 1.0.4.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎版本

截至二零二一年三月十一日，引擎版本 1.0.4.1.R2.1 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 禁用了可以添加或附加到現有標籤和屬性的 Gemlin 條件插入模式的優化。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.1.R2.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.8`
- 版本：`1.1`

升級至引擎版本 1.0.4.1.R2.1 的路徑

如果您執行的是引擎版本 `1.0.4.1.R2`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.4.1.R2.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.1.R2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.1.R2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> ```

```
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2020 年 10 月 12 日，引擎版本 1.0.4.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 發行版本: (p. 952)

這個引擎版本的新功能

- 為小精靈添加幀級壓縮。

這個引擎版本的改善項目

- Amazon Neptune 現在需要使用安全通訊端層 (SSL) 搭配 TLSv1.2 通訊協定，以便在所有區域中使用這些強式加密套件與 Neptune 的所有連線：
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

  REST 和與海王星的WebSocket連接都是如此，這意味著在所有區域中連接到海王星時，必須使用 HTTPS 而不是 HTTP。

  因為在任何地方都不再支援使用 HTTP 或 TLS 1.1 的用戶端連線，因此在升級至此引擎版本之前，請確定您的用戶端和程式碼已更新為使用 TLS 1.2 和 HTTPS。

  > **Important**
  >
  > 必須將 SSL/TLS 用於與海王星的所有連接可能是一個突破性的變化。它會影響您與 Girmlin 控制台，小鬼驅動程序，小鬼蟒蛇，.NET，節點，REST API 以及負載平衡器連接的連接。如果您一直在使用 HTTP 進行任何或所有這些操作，則現在必須更新相關的客戶端和驅動程序，並更改代碼以使用 HTTPS，否則連接將失敗。
  > 此版本中的一個錯誤已允許 HTTP 連線和/或過期的TLS 連線繼續適用於先前設定資料庫叢集參數以防止強制執行 HTTPS 連線的客戶。此錯誤已在修補程式版本 1.0.4.0.R2 和 1.0.4. 1. (p. 946) R2 (p. 952) 中修正，但修正程式在自動安裝修補程式時造成未預期的連線失敗。
  > 基於這個原因，這兩個修補程式都已回復，而且只能手動安裝，讓您有機會更新 TLS 1.2 的設定。

- 已升級TinkerPop至 3.4.8 版。這是向後兼容的升級。有關新功能，請參閱TinkerPop更改日誌。
- 改善了「小精靈」步驟properties()的效能。
- 已新增有關說明BindOp和個人檔案報告MultiplexerOp的詳細資訊。
- 已新增資料預取，以在有快取遺漏時改善效能。
- 在批次載入器的parserConfiguration參數中新增了新allowEmptyStrings設定，允許在 CSV 載入中將空字串視為有效屬性值 (請參閱海王星裝載機請求參 (p. 276))。
- 載入程式現在允許在多值 CSV 資料行中使用逸出分號。

此引擎版本中修正的缺陷

- 修復了與步驟相關的both()潛在小鬼記憶體洩漏問題。
- 修正因為未正確處理以 '/' 結尾的端點而遺失要求量度的錯誤。
- 修正在實驗室模式下啟用 DFE 引擎時，導致複本落後並在重負載下重新啟動的錯誤。
- 修正當大量載入因out-of-memory條件而失敗時，無法回報正確錯誤訊息的錯誤。
- 修復了 SPARQL 錯誤，其中字符編碼被放置在 SPARQL 查詢響應中的內容編碼標頭中。現在charset被放置在內容類型標頭中，使 HTTP 客戶端能夠識別正在自動使用的字符集。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.8
- 版本：1.1

引擎版本 1.0.4.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.4.0 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至二零二一年二月二十一日，引擎版本 1.0.4.0.0.R2 正在普遍部署中。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正了允許使用HTTP或更早版本的 TLS，而不是 TLS 1.2 連接到海王星HTTPS的錯誤。發行版本： (p. 950)

  Important

  必須將 SSL/TLS 用於與海王星的所有連接可能是一個突破性的變化。它會影響您與 Girmlin 控制台，小鬼驅動程序，小鬼蟒蛇，.NET，節點，REST API 以及負載平衡器連接的連接。如果您到目前為止一直在使用 HTTP 或較舊的 TLS 版本，則必須在安裝此 Patch 之前更新相關的用戶端和驅動程式，並將您的程式碼變更為僅使用 HTTPS。
- 修正 CSV 大量載入中涉及結尾標籤的錯誤#。

- 修復了在某些情況下，發生時InternalFailureException被設置為響應代碼的小鬼錯誤。ConcurrentModificationException
- 修復了在某些條件下更新邊緣或頂點可能導致瞬態的 Gemlin 錯誤。InternalFailureException

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.4.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.8
- 版本：1.1

升級至引擎版本 1.0.4.0.R2 的路徑

如果您執行的是引擎版本 1.0.4.0，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.4.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.4.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.4.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2020 年 8 月 3 日，引擎版本 1.0.3.0 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 版本:1.0.0.0.0.0.2 (p. 958)
- 發行版本: (p. 956)

這個引擎版本的新功能

- Neptune 推出了一種新的替代查詢引擎（DFE），可以顯著加快查詢執行速度。請參閱 亞馬遜海王星替代查詢引擎（DFE） (p. 129)。
- DFE 依賴預先產生的有關 Neptune 圖形資料的統計資料，這些資料是透過新的統計資料端點管理的。請參閱 DFE 統計資料 (p. 131)。
- 您現在可以將新`includeQueuedLoads`參數設定為 FALSE，從載入程式 Get-Status API 傳回的載入識別碼清單中排除佇列的載入工作。請參閱 海王星裝載機獲取狀態請求參數 (p. 283)。
- Neptune 現在支援 SPARQL 查詢回應的結尾標頭，如果要求在開始傳回回應區塊後失敗，可能會包含錯誤碼和訊息。請參閱 用於多部分 SPARQL 響應的可選 HTTP 尾隨標題 (p. 430)。
- Neptune 現在也可以讓您為 Gremlin 查詢啟用區塊回應編碼。與 SPARQL 案例一樣，回應區塊的尾端標頭可能會包含錯誤碼和訊息，如果查詢開始傳回回應區塊之後發生失敗。請參閱 使用可選的 HTTP 尾隨標題來啟用多部分小鬼響應 (p. 305)。

這個引擎版本的改善項目

- 您現在可以提供批次要求的大小，以便在 Gemlin 中ElasticSearch進行全文搜尋。
- 已改善 SPARQL 群組依據查詢的記憶體使用量。
- 添加了一個新的 Gemlin 查詢優化器來修剪某些未綁定的過濾器。

- 將使用 IAM 驗證的WebSocket連線保持開啟的時間上限從 36 小時增加到 10 天。

此引擎版本中修正的缺陷

- 修正如果您在 POST 要求中傳送未編碼的 URL 參數，海王星會傳回 500 和一個 HTTP 狀態碼的錯誤`InternalServerErrorException`。現在，海王星會傳回 400 和 a 的 HTTP 狀態碼`BadRequestException`，並附上訊息：`Failure to process the POST request parameters`。
- 修復了一個沒有正確報告WebSocket連接失敗的小鬼錯誤。
- 修正一個關於副作用消失的小鬼錯誤。
- 修復了未正確支持全文檢索搜索`batchsize`參數的 Girmlin 錯誤。
- 修正了一個小精靈錯誤，以處理`toV`和`fromV`單獨為每個方向上。`bothE`
- 修復了步驟`Edge pathType`中涉及的`hasLabel`小鬼錯誤。
- 修復了使用靜態綁定重新排序聯接排序無法正常工作的 SPARQL 錯誤。
- 修正未正確報告無法使用的 Amazon S3 儲存貯體的 SPARQL 更新載入錯誤。
- 修正未正確報告子查詢中服務節點問題的 SPARQL 錯誤。
- 已修正 SPARQL 錯誤，其中包含巢狀篩選器「存在」或「篩選器不存在」條件的查詢未正確評估。
- 修正透過產生查詢呼叫 SPARQL 服務端點時，正確處理重複產生之繫結的 SPARQL 錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.3.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.3`
- 版本：`1.1`

引擎版本 1.0.3.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

如果您的叢集已將其 `AutoMinorVersionUpgrade` 參數設為 `True`，則在維護時段期間，您的叢集會在此發行日期後兩至三週自動升級至此引擎版本。

升級至此版本

亞馬遜海王星 1.0.3.0 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.3.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
```

```
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.3.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> Note
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至二零二一九年二月二十一日，引擎版本 1.0.3.0.0.R3 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正 CSV 大量載入中涉及結尾標籤的錯誤#。
- 修正了一個 Gemlin 錯誤，該錯誤可能會導致使用該步驟的特定查詢模式丟失結果。`as()`
- 修復了在另一個`project()`步驟中使用嵌套步驟時可能會導致錯誤的 Gemlin 錯誤，例如。`union()`
- 修復了當使用類似的終端方法時，實驗 DFE 引擎中字符串遍歷執行中的 Gremlin 錯誤。`toList()`

- 修復了在字符串查詢中使用`iterate()`步驟時無法關閉事務的 Gemlin 錯誤。
- 修復了在特定情況下可能導致使用該`is(P.gte(0))`模式的查詢拋出異常的 Gemlin 錯誤。
- 修復了在某些情況下，發生時`InternalFailureException`被設置為響應代碼的 Girmlin 錯誤。`ConcurrentModificationException`
- 修復了在某些條件下，更新邊緣或頂點可能導致瞬態的 Gemlin 錯誤。`InternalFailureException`

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.0.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.8`
- 版本：`1.1`

升級至引擎版本 1.0.0.R3 的路徑

如果您執行的是引擎版本 `1.0.3.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.0.0.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.3.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.3.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

截至 2020 年 10 月 12 日，引擎版本 1.0.3.0.0.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

這個引擎版本的改善項目

- 改善了「小精靈」步驟`properties()`的效能。
- 已新增有關說明`BindOp`和個人檔案報告`MultiplexerOp`的詳細資訊。
- 對於 SPARQL 查詢回應，新增`charset`至內容類型標頭，讓 HTTP 用戶端能夠辨識自動使用的字元集。

此引擎版本中修正的缺陷

- 修復`CancellationException`了一個未處理的 SPARQL 錯誤。
- 修正包含巢狀選項的查詢無法正常運作的 SPARQL 錯誤。
- 修復了 LOAD 中`ConcurrentModificationException`可能導致查詢掛起的 SPARQL 錯誤。
- 修正 SPARQL 錯誤，此錯誤會導致查詢回應無法被 gzip 壓縮。
- 修正了步驟中的一個小精靈錯誤。`groupBy()`
- 修復了一個與在步驟中使用`aggregate()`步驟相關的小鬼錯誤。`local()`
- 修復了與使用`bothE()`後跟使用彙總值的謂詞相關的小鬼錯誤。
- 修復了與步驟一起使用`bothE()`步驟相關的小精靈錯誤。`repeat()`
- 修復了與步驟相關的`both()`潛在小鬼記憶體洩漏問題。
- 修正因為未正確處理以 '/' 結尾的端點而遺失要求量度的錯誤。
- 修復了即使請求隊列未滿`ThrottlingException`也可能引發的錯誤。
- 修正了當負載失敗的原因，如獲取負載狀態的錯誤。`LOAD_DATA_FAILED_DUE_TO_FEED_MODIFIED_OR_DELETE`

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.0.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.3
- 版本：1.1

升級至引擎版本 1.0.0.R2 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

如果您的叢集已將其 AutoMinorVersionUpgrade 參數設為 True，則在維護時段期間，您的叢集會在此發行日期後兩至三週自動升級至此引擎版本。

升級至此版本

亞馬遜海王星 1.0.0.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.3.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.3.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2020 年 3 月 09 日，引擎版本 1.0.2.2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此版本的後續修補程式版本

- 版本:1.0.2.2.R2 (p. 967)
- 發行版本:1.0.2.2.R3 (p. 967)
- 發行版本:1.0.2.2.R4 (p. 965)
- 版本:1.0.2.2.R5 (p. 964)
- 發行版本: (p. 962)

這個引擎版本的改善項目

- 已將正要復原之交易的相關資訊新增至狀態 API。請參閱 執行個體狀態 (p. 770)。
- TinkerPop將阿帕奇的版本升級到 3.4.3。

  版本 3.4.3 與海王星（3.4.1）支持的先前版本向後兼容。它確實在行為中引入次要變更：當您嘗試關閉不存在的工作階段時，Gremlin 不再傳回錯誤 (請參閱防止在關閉不存在的工作階段時發生錯誤)。
- 已移除執行 Gremlin 全文搜尋步驟時的效能瓶頸。

此引擎版本中修正的缺陷

- 已修正在處理查詢中的空白圖形模式時發生的 SPARQL 錯誤。
- 已修正在處理 URL 編碼查詢中的未編碼分號時發生的 SPARQL 錯誤。
- 已修正在處理 `Union` 步驟中的重複頂點時發生的 Gremlin 錯誤。
- 已修正一個 Gremlin 錯誤，此錯誤會導致在 `.repeat()` 內具有 `.simplePath()` 或 `.cyclicPath()` 的查詢傳回不正確的結果。
- 已修正一個 Gremlin 錯誤，此錯誤會導致 `.project()` 在其子周遊未傳回任何解決方案時傳回不正確的結果。

- 已修正一個 Gremlin 錯誤，其中來自讀寫衝突的錯誤引發了 `InternalFailureException` 而不是 `ConcurrentModificationException`。
- 已修正導致 `.group().by(...).by(values("property"))` 失敗的 Gremlin 錯誤。
- 修復了步驟配置文件輸出中的小精靈錯誤。full-text-search
- 已修正 Gremlin 工作階段中的資源洩漏。
- 已修正在某些情況下狀態 API 無法報告正確的可訂購版本的錯誤。
- 修正大量載入器錯誤，此錯誤會允許在大量載入請求中使用到 Amazon S3 以外的位置做為來源。
- 已修正詳細載入狀態下的大量載入器錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.3`
- 版本：`1.1`

引擎版本 1.0.2.2 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

如果您的叢集已將其 `AutoMinorVersionUpgrade` 參數設為 `True`，則在維護時段期間，您的叢集會在此發行日期後兩至三週自動升級至此引擎版本。

升級至此版本

亞馬遜海王星 1.0.2.2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至二零二一九年二月二十九日，引擎版本 1.0.2.2.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修復了在某些情況下，發生時`InternalFailureException`被設置為響應代碼的 Girmlin 錯誤。`ConcurrentModificationException`
- 修復了在某些條件下，更新邊緣或頂點可能導致瞬態的 Gemlin 錯誤。`InternalFailureException`

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.2.R6 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.8`
- 版本：`1.1`

升級至引擎版本 1.0.2.2.R6 的路徑

如果您執行的是引擎版本 `1.0.2.2`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.2.R6 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製正在執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
>   We're sorry, your request to modify DB cluster (cluster identifier) has failed.
>   Cannot modify engine version because instance (instance identifier) is
>   running on an old configuration. Apply any pending maintenance actions on the
> instance before
>   proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至 2020 年 10 月 12 日，引擎版本 1.0.2.2.R5 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

這個引擎版本的改善項目

- 改善了「小精靈」步驟properties()的效能。
- 已新增有關說明BindOp和個人檔案報告MultiplexerOp的詳細資訊。
- 對於 SPARQL 查詢回應，新增charset至內容類型標頭，可讓 HTTP 用戶端辨識自動使用的字元集。

此引擎版本中修正的缺陷

- 修復CancellationException了一個未處理的 SPARQL 錯誤。
- 修正包含巢狀選項的查詢無法正常運作的 SPARQL 錯誤。
- 修復了 LOAD 中ConcurrentModificationException可能導致查詢掛起的 SPARQL 錯誤。
- 修正 SPARQL 錯誤，此錯誤會導致查詢回應無法被 gzip 壓縮。
- 修正了步驟中的一個小精靈錯誤。groupBy()
- 修復了一個與在步驟中使用aggregate()步驟相關的小鬼錯誤。local()
- 修復了與使用bothE()後跟使用彙總值的謂詞相關的小鬼錯誤。
- 修復了與步驟一起使用bothE()步驟相關的小精靈錯誤。repeat()
- 修復了與步驟相關的both()潛在小鬼記憶體洩漏問題。
- 修正因為未正確處理以 '/' 結尾的端點而遺失要求量度的錯誤。
- 修復了即使請求隊列未滿ThrottlingException也可能引發的錯誤。
- 修正了當負載失敗的原因，如獲取負載狀態的錯誤。LOAD_DATA_FAILED_DUE_TO_FEED_MODIFIED_OR_DELETE

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.2.R5 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.3
- 版本：1.1

升級至引擎版本 1.0.2.2.R5 的路徑

如果您執行的是引擎版本 1.0.2.2，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.2.R5 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

截至 2020 年 7 月 23 日，引擎版本 1.0.2.2.R4 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

這個引擎版本的改善項目

- 更頻繁地將未使用的記憶體釋放回作業系統，藉此改善記憶體使用量。
- 也改善了 SPARQL 群組依據查詢的記憶體使用量。
- 將使用 IAM 驗證的WebSocket連線保持開啟的時間上限從 36 小時增加到 10 天。
- 已新增指`BufferCacheHitRatio`CloudWatch標，可用於診斷查詢延遲和調整執行個體類型。請參閱 海王星度量 (p. 775)。

此引擎版本中修正的缺陷

- 修正關閉閒置或過期 IAM WebSocket 連線的錯誤。海王星現在在關閉連接之前發送一個關閉框架。
- 修正評估包含巢狀篩選器存在和/或篩選器不存在條件的查詢時的 SPARQL 錯誤。
- 修正 SPARQL 查詢終止錯誤，該錯誤會在特定極端情況下導致伺服器上的執行緒阻塞。
- 修復了步驟中涉及邊緣路徑類型的小鬼錯誤。`hasLabel`
- 修正了一個小精靈錯誤，以處理`toV`和`fromV`單獨為每個方向上。`bothE`
- 修正一個關於副作用消失的小鬼錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.2.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.3`
- 版本：`1.1`

升級至引擎版本 1.0.2.2.R4 的路徑

如果您執行的是引擎版本 `1.0.2.2`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.2.R4 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

Amazon Neptune 发动机

引擎版本 1.0.2.2.R3 已納入引擎版本 1.0.2.R4 (p. 965)。

亚马逊海王星引擎

截至 2020 年 4 月 2 日，引擎版本 1.0.2.2.R2 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

這個引擎版本的改善項目

- 您現在可以將多達 64 個大量載入的任務排入佇列，而不必等待其中一個完成，再啟動下一個。您也可以在使用 `load` 命令的 `dependencies` 參數順利完成一或多個先前佇列的載入任務時，執行佇列的載入請求。請參閱 海王星裝載程式 (p. 275)。
- F ull-text-search 輸出現在可以進行排序（請參閱全文檢索搜尋 (p. 614)）。
- 現在有用於叫用 Neptune 串流的資料庫叢集參數，而且該功能已移出實驗室模式。請參閱 啟用 Neptune 串流 (p. 586)。

此引擎版本中修正的缺陷

- 已修正伺服器啟動時的隨機故障，此故障會延遲執行個體的建立。
- 已修正最佳化工具問題，其中，查詢中的 BIND 陳述式讓最佳化工具在連結順序規劃中使用非選擇性模式展開。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.2.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.3
- 版本：1.1

升級至引擎版本 1.0.2.2.R2 的路徑

如果您執行的是引擎版本 1.0.2.2，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.2.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.2 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.2 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

此版本的後續修補程式版本

- 版本:1.0.2.1.R6 (p. 971)
- 發行版本: (p. 973) 未部署此修補程式版本。
- 發行版本:1.0.2.R (p. 973)
- 版本: (p. 975)
- 版本: (p. 976)

這個引擎版本的新功能

- 通過與亞馬遜OpenSearch服務集成添加了全文搜索功能。請參閱 海王星全文搜索 (p. 604)
- 已新增使用實驗室模式的選項，為大量述詞建立第四個索引 (OSGP 索引)。請參閱 OSGP 指數 (p. 128)。
- 將詳細資訊模式新增至 SPARQL Explain。如需詳細資訊，請參閱使用 SPARQL `explain` (p. 454)和詳細模式輸出 (p. 457)。
- 將實驗室模式資訊新增至引擎狀態報告。如需詳細資訊，請參閱 執行個體狀態 (p. 770)。
- 資料庫叢集快照現在可以跨AWS區域複製。請參閱 複製快照 (p. 813)。

這個引擎版本的改善項目

- 改進處理大量述詞時的效能。
- 增強查詢最佳化。儘管這對客戶而言，應該是完全透明的，但建議您在升級前，先測試應用程式，以確保應用程式如預期運作。

- 錯誤報告的小規模增強功能。
- 已新增 Gremlin `.project()` 和 `.identity()` 步驟的最佳化。
- 已新增非終結 Gremlin `.union()` 案例的最佳化。
- 已新增 Gremlin `.path().by()` 周遊的原生支援。
- 已新增 Gremlin `.coalesce()` 的原生支援。
- 進一步最佳化大量寫入。
- 我們現在要求 HTTPS 連線至少使用 TLS 版本 1.2 或更高版本，以防止使用過時/不安全的密碼。

此引擎版本中修正的缺陷

- 已修正 Gremlin `addE()` 內部周遊處理錯誤。
- 已修正由 AST 註釋從子系周遊洩漏到父系周遊造成的 Gremlin 錯誤。
- 已修正在 `select()` 之後呼叫 `.otherV()` 時，在 Gremlin 中發生的錯誤。
- 已修正某些 `.hasLabel()` 步驟出現在 `bothE()` 步驟之後會失敗的 Gremlin 錯誤。
- 對 Gremlin .sum() 和 .project() 進行小規模修正。
- 已修正處理 SPARQL 查詢缺少右括號的錯誤。
- 已修正 SPARQL Explain 中的一些小錯誤。
- 已修正處理並行取得載入狀態要求時的錯誤。
- 減少利用 `.project()` 步驟執行一些 Gremlin 周遊所使用的記憶體。
- 已修正 SPARQL 中特殊值的數值比較。請參閱 標準合規 (p. 83)。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

引擎版本 1.0.2.1 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.1 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

截至 2020 年 4 月 22 日，引擎版本 1.0.2.1.R6 正在普遍部署。請注意，新版本需要數天才能在每個區域推出。

此引擎版本中修正的缺陷

- 修正以下錯誤：`ConcurrentModificationConflictException` 和 `TransactionException` 未轉換成 `NeptuneGremlinException`，導致傳回 `InternalFailureException` 給客戶。
- 修正了 Neptune 在伺服器完全準備就緒之前報告其狀態為正常的錯誤。

- 修正以下錯誤：同時插入兩個 `value->id` 對應時，未按照順序遞交字典和使用者交易。
- 修正載入狀態序列化的錯誤。
- 修正 Gremlin 工作階段錯誤。
- 修復了當服務器啟動失敗時，Neptune 無法拋出異常的錯誤。
- 修復了 Neptune 在關閉通道之前無法發送 Web 套接字關閉框架的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.1.R6 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

升級至引擎版本 1.0.2.1.R6 的路徑

如果您執行的是引擎版本 `1.0.2.1`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.1.R6 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間範圍從 20-30 秒到數分鐘，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

Amazon Neptune 引擎

發動機版本 1.0.2.1.R5 從未部署過。

亚马逊海王星发动机

這個引擎版本的改善項目

- Neptune 現在嘗試始終在執行管道中首先放置任何full-text-search呼叫。這會減少呼叫量OpenSearch，進而大幅改善效能。請參閱 Full-text-search 查詢執行 (p. 626)。
- `IllegalArgumentException`如果您嘗試訪問不存在的屬性，頂點或邊緣，海王星現在會引發一個。以前，海王星`UnsupportedOperationException`在這種情況下提出了一個。

  例如，如果您嘗試新增一個參考不存在頂點的邊緣，則您現在將引發一個`IllegalArgumentException`。

此引擎版本中修正的缺陷

- 已修正 Gremlin 錯誤，其中 `project-by` 內的 `union` 周遊不會傳回結果或傳回不正確的結果。
- 已修正導致巢狀 `.project().by()` 步驟傳回不正確結果的 Gremlin 錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.1.R4 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

升級至引擎版本 1.0.2.1.R4 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

不過，不支援自動更新到此版本。

升級至此版本

亞馬遜海王星 1.0.2.1.R4 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> ```

```
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

此引擎版本中修正的缺陷

- 已修正下列錯誤：即使已在 neptune_lab_mode 參數中使用 ObjectIndex 值，正確地啟用在 實驗室模式 (p. 127) 使用的功能，仍會停用 OSGP 索引。
- 已修正因 .project().by() 步驟內部的 .fold() 而影響 Gremlin 查詢的錯誤。例如，此錯誤會造成下列查詢傳回不完整的結果：

```
g.V().project("a").by(valueMap().fold())
```

- 已修正大量載入 RDF 資料的效能瓶頸。
- 修正啟用串流且複本在主複本之前重新啟動時，造成複本當機的錯誤。
- 已修正在未重新啟動書執行個體的情況下，未挑選在執行個體上輪換的 SSL 憑證的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.1.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.1
- 版本：1.1

升級至引擎版本 1.0.2.1.R3 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

不過，不支援自動更新到此版本。

升級至此版本

亞馬遜海王星 1.0.2.1.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> Note
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

此引擎版本中修正的缺陷

- 已修正透過非循環配置資源依序周遊和非 `path()` 依序周遊影響所有 `project().by()` 查詢的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.1.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：3.4.1
- 版本：1.1

升級至引擎版本 1.0.2.1.R2 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

不過，不支援自動更新到此版本。

升級至此版本

亞馬遜海王星 1.0.2.1.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.1 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.1 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

重要提示：此引擎版本現已棄用

從 2020-05-19 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本現已由 1.0.2.1 版取代，其中包含此版本 (p. 969)中的所有錯誤修正，以及其他功能，例如全文檢索搜尋整合、OSGP 索引支援，以及跨區域的資料庫快照集叢集複製。AWS

自 2020 年 6 月 1 日起，Neptune 將在下一個維護時段期間，自動將執行此引擎版本的任何叢集升級為 1.0.2.1 版的最新修補程式 (p. 971)。您可以在此之前手動升級，如此處 (p. 969)所述。

如果您對升級有任何問題，請透過AWS支援或AWS開發人員論壇與我們聯絡。

此版本的後續修補程式版本

- 發行版本: (p. 980)
- 版本: (p. 981)

這個引擎版本的新功能

除了維護更新以外，此版本還增加了新功能，一次支援多個引擎版本 (請參閱 海王星引擎更新 (p. 857))。

因此，引擎版本的編號已變更 (請參閱 海王星引擎版本編號 (p. 859))。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.2.0 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

引擎版本 1.0.2.0 的升級路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

您不會自動升級至此版本。

升級至此版本

亞馬遜海王星 1.0.2.0 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> ```

```
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星引擎

重要提示：此引擎版本現已棄用

從 2020-05-19 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本現已由 1.0.2.1 版取代，其中包含此版本 (p. 969)中的所有錯誤修正，以及其他功能，例如全文檢索搜尋整合、OSGP 索引支援，以及跨區域的資料庫快照集叢集複製。AWS

自 2020 年 6 月 1 日起，Neptune 將在下一個維護時段期間，自動將執行此引擎版本的任何叢集升級為 1.0.2.1 版的最新修補程式 (p. 971)。您可以在此之前手動升級，如此處 (p. 969)所述。

如果您對升級有任何問題，請透過AWS支援或AWS開發人員論壇與我們聯絡。

此引擎版本中修正的缺陷

- 修正以下錯誤：`ConcurrentModificationConflictException` 和 `TransactionException` 被回報為一般 `InternalFailureException`。
- 修正運作狀態檢查中導致伺服器在啟動期間頻繁重新啟動的錯誤。
- 修正以下錯誤：資料因為在某些情況下未按照順序遞交而無法顯示在複本上。
- 修正負載狀態序列化中缺少 Amazon S3 存取許可而導致載入失敗的錯誤。
- 已修正 Gremlin 工作階段中的資源洩漏。
- 修正運作狀態檢查中的錯誤，此錯誤會在啟動管理 IAM 身份驗證的元件時隱藏不良狀態。
- 修復了海王星在關閉頻道之前無法發送關WebSocket閉幀的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.0.0.R3 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

升級至引擎版本 1.0.0.R3 的路徑

如果您執行的是引擎版本 `1.0.2.0`，您的叢集將在下一個維護時段自動升級至此修補程式版本。

您可以手動將任何較早的 Neptune 引擎發行版本升級至此版本。

升級至此版本

亞馬遜海王星 1.0.0.0.R3 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

> **Note**
>
> 如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：
>
> ```
> We're sorry, your request to modify DB cluster (cluster identifier) has failed.
> Cannot modify engine version because instance (instance identifier) is
> running on an old configuration. Apply any pending maintenance actions on the
> instance before
> proceeding with the upgrade.
> ```
>
> 如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亚马逊海王星发动机

重要提示：此引擎版本現已棄用

從 2020-05-19 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本現已由 1.0.2.1 版取代，其中包含此版本 (p. 969)中的所有錯誤修正，以及其他功能，例如全文檢索搜尋整合、OSGP 索引支援，以及跨區域的資料庫快照集叢集複製。AWS

自 2020 年 6 月 1 日起，Neptune 將在下一個維護時段期間，自動將執行此引擎版本的任何叢集升級為 1.0.2.1 版的最新修補程式 (p. 971)。您可以在此之前手動升級，如此處 (p. 969)所述。

如果您對升級有任何問題，請透過AWS支援或AWS開發人員論壇與我們聯絡。

此引擎版本中修正的缺陷

- 為伺服器上的中途分頁改善快取策略，以便在伺服器進入記憶體不足的狀態時，`FreeableMemory` 能夠更快復原。
- 已修正伺服器上處理許多並行載入狀態及/或啟動載入要求時，可能導致競爭條件和當機的錯誤。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.0.0.R2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

升級至引擎版本 1.0.0.0.R2 的路徑

您可以手動將任何先前的 Neptune 引擎發行版本升級至此版本。

不過，不支援自動更新到此版本。

升級至此版本

亞馬遜海王星 1.0.0.0.R2 現已正式推出。

如果資料庫叢集執行的引擎版本具有此版本的升級路徑，則可立即升級。您可以使用主控台上的資料庫叢集作業或使用 SDK 來升級任何合格的叢集。以下 CLI 命令將立即升級合格的叢集：

對於 Linux、OS X 或 Unix：

```
aws neptune modify-db-cluster \
    --db-cluster-identifier (your-neptune-cluster) \
    --engine-version 1.0.2.0 \
    --apply-immediately
```

針對 Windows：

```
aws neptune modify-db-cluster ^
    --db-cluster-identifier (your-neptune-cluster) ^
    --engine-version 1.0.2.0 ^
    --apply-immediately
```

更新會同時套用到資料庫叢集中的所有執行個體。更新需要在這些執行個體上重新啟動資料庫，因此您將會遇到停機時間介於 20-30 秒到數分鐘之間，之後您就可以使用資料庫叢集繼續執行。

升級前一律測試

發布新的主要或次要 Neptune 引擎版本時，請務必先在其上測試 Neptune 應用程式，然後再升級到它。即使是小幅升級，也可能會引入會影響程式碼的新功能或行為。

首先，請將目前版本中的版本說明頁面與目標版本的版本進行比較，以查看查詢語言版本或其他重大變更是否會有變更。

在升級生產資料庫叢集之前測試新版本的最佳方法是複製您的生產叢集，以便複製執行新的引擎版本。然後，您可以在複製上執行查詢，而不會影響生產資料庫叢集。

升級前一律建立手動快照

在執行升級之前，強烈建議您一律建立資料庫叢集的手動快照。擁有自動快照僅提供短期保護，而手動快照仍然可用，直到您明確刪除它為止。

在某些情況下，Neptune 會為您建立手動快照，做為升級程序的一部分，但您不應該依賴此快照，而且在任何情況下都應該建立自己的手動快照。

當您確定不需要將資料庫叢集還原為升級前狀態時，可以明確刪除您自己建立的手動快照，以及 Neptune 可能建立的手動快照集。如果 Neptune 建立手動快照集，其名稱的開`preupgrade`頭會是資料庫叢集的名稱、來源引擎版本、目標引擎版本和日期。

Note

如果您嘗試在處理擱置動作時進行升級，可能會遇到如下錯誤：

```
  We're sorry, your request to modify DB cluster (cluster identifier) has failed.
  Cannot modify engine version because instance (instance identifier) is
  running on an old configuration. Apply any pending maintenance actions on the
instance before
  proceeding with the upgrade.
```

如果遇到此錯誤，請等待擱置的動作完成，或立即觸發維護時段，讓先前的升級完成。

如需升級引擎版本的詳細資訊，請參閱海王星引擎更新 (p. 857)。如果您有任何問題或疑慮，AWS支援團隊可以透過社群論壇和進AWS階支援取得。

亞馬遜海王星引擎

重要：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

這個引擎版本的改善項目

- `IllegalArgumentException`如果您嘗試訪問不存在的屬性，頂點或邊緣，海王星現在會引發一個。以前，海王星`UnsupportedOperationException`在這種情況下提出了一個。

  例如，如果您嘗試新增一個參考不存在頂點的邊緣，則您現在將引發一個 `IllegalArgumentException`。

此引擎版本中修正的缺陷

- 修正以下錯誤：同時插入兩個 `value->id` 對應時，未按照順序遞交字典和使用者交易。
- 修正載入狀態序列化的錯誤。
- 已修正伺服器啟動時的隨機故障，此故障會延遲執行個體的建立。
- 已修正游標流失的問題。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.1.2 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.4.1`
- 版本：`1.1`

亞馬遜海王星引擎

重要：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本中修正的缺陷

- 修正了同時插入時提交的失序錯誤。
- 修正載入狀態序列化的錯誤。
- 已修正伺服器啟動時的隨機故障，此故障會延遲執行個體的建立。
- 已修正記憶體流失。

此發行版本支援的查詢語言版本

將資料庫叢集升級至 1.0.1.1 版之前，請確定您的專案與下列查詢語言版本相容：

- 小精靈版本：`3.3.2`
- 版本：`1.1`

Amazon Neptune 引擎

重要：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

Amazon Neptune 10-31

版本：

重要提示：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本中修正的缺陷

- 修正了當用戶端使用 `traversal().withRemote(...)` (亦即使用 GLV 位元碼) 連接到 Neptune 時，`tree()` 步驟回應序列化中的 Gremlin 錯誤。

  此版本解決了使用 `traversal().withRemote(...)` 連線至 Neptune 的用戶端接收到對包含 `tree()` 步驟之 Gremlin 查詢的無效回應的問題。

- 修正了 DELETE WHERE LIMIT 查詢中的 SPARQL 錯誤，其中查詢終止程序會因競爭條件而停止回應，造成查詢逾時。

2019-10-2019-10-2019-10-2019-10-2019-12-2019-12-23

版本：

重要提示：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

這個引擎版本的新功能

- 已新增 Gremlin Explain/描述檔功能 (請參閱 使用小靈分析海王星查詢執行 explain (p. 347))。
- 已新增 支持基於小鬼腳本的會話 (p. 326) 以在單一交易中啟用多個 Gremlin 周遊的執行。
- 已新增對 Neptune 中 SPARQL 聯合查詢延伸模組的支援 (請參閱 SPARQL 1.1 聯合查詢和在海王星中使用擴充功能的 SPARQL 聯合查詢 SERVICE (p. 473))。
- 已新增一項功能，讓您可透過 HTTP URL 參數或透過 SPARQL queryId 查詢提示 (請參閱 將自定義 ID 注入海王星小鬼或 SPARQL 查詢 (p. 126))，將自己的 queryId 注入至 Gremlin 或 SPARQL 查詢。
- 為 Neptune 新增了一實驗室模式 (p. 127)項功能，可讓您嘗試尚未準備好在生產環境中使用的即將推出的功能。
- 已新增即將推出的 Neptune (p. 585) 功能，可將對資料庫進行的每項變更可靠地記錄到持續一週的串流中。此功能只能在實驗室模式中使用。
- 更新並行交易的正式語意 (請參閱海王星中的事務語義 (p. 109))。此功能針對並行提供產業標準保證。

  根據預設，會啟用這些交易語意。在某些情況下，此功能可能會變更目前的載入行為，並降低載入效能。您可以在參數值中包含 ReadWriteConflictDetection=disabled，以使用資料庫叢集 neptune_lab_mode 參數來還原到先前語意。

這個引擎版本的改善項目

- 透過報告引擎使用的 SPARQL 版本 TinkerPop 和版本來改善執行個體狀態 (p. 770) API。
- 已提升 Gremlin 子圖形運算子的效能。
- 已提升 Gremlin 回應序列化的效能。
- 已提升 Gremlin Union 步驟的效能。
- 已縮短簡單 SPARQL 查詢的延遲。

此引擎版本中修正的缺陷

- 已修正 Gremlin 錯誤，其中未正確地以內部失敗形式傳回逾時。
- 已修正 SPARQL 錯誤，其中局部變數集的 ORDER BY 已導致內部伺服器錯誤。

Amazon Neptune 引擎更新 2019-12-

重要提示：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

版本：

Amazon Neptune 1.0.0.0.457.0 正式推出。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200457.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用以下 CLI 命令來升級 DB 叢集：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune upport 檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support 聯繫 Support。

此引擎版本中修正的缺陷

- 已修正先前引擎版本 (1.0.1.0.200369.0) 中引進的 Gremlin 正確性問題，方法為移除造成此問題之接續述詞處理的效能改善。
- 已修正 SPARQL 錯誤，此錯誤會導致搭配 `DISTINCT` 的查詢，以及包裝為 `OPTIONAL` 的單一模式產生 `InternalServerError`。

Amazon Neptune 08-13

重要：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

這個引擎版本的新功能

- 為的`parallelism`參數新增了一個`OVERSUBSCRIBE`選項海王星裝載程式 (p. 275)，這會導致 Neptune 大量載入器使用所有可用的執行緒和資源。

這個引擎版本的改善項目

- 改善包含簡單邏輯 OR 運算式的 SPARQL 篩選條件的效能。
- 改善處理接續述詞方面的 Gremlin 效能。

此引擎版本中修正的缺陷

- 修復了阻止`xsd:duration`從中減去的 SPARQL 錯誤`xsd:date`。
- 修正導致存在 `UNION` 時靜態內嵌的結果不完整的 SPARQL 錯誤。
- 修正查詢取消中的 SPARQL 錯誤。
- 修正類型提升期間導致溢位的 Gremlin 錯誤。
- 修正 `addE().from().to()` 步驟中處理頂點元素時的 Gremlin 錯誤。
- 修正涉及單一基數插入中 NaN 雙精位數和浮點數處理的 Gremlin 錯誤 (2019-07-26 於引擎版本 1.0.1.0.200366.0 (p. 987) 中發行)。
- 修正涉及以屬性為基礎的搜尋中產生查詢計畫的錯誤。

Amazon Neptune 引擎更新

版本：

重要：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

這個引擎版本的新功能

- 升級到 TinkerPop 3.4.1（請參閱TinkerPop 升級信息和 TinkerPop3.4.1 更改日誌）。

  對於 Neptune 客戶，這些變更提供了新的功能和改進，例如：
  - `GraphBinary` 現在以序列化格式提供。
  - 一個導致 TinkerPop Java 驅動程式記憶體洩漏的持續活動錯誤已修正，因此不再需要解決問題。

  但是，在某些情況下，它們可能會影響 Neptune 中現有的 Gemlin 代碼。例如：
  - `valueMap()` 現在會傳回 `Map<Object,Object>` 而非 `Map<String,Object>`。
  - 已修正 `within()` 步驟的不一致行為，使其能與其他步驟一致運作。先前，類型必須相符，比較作業才能進行。現在，可以準確比較不同類型的數字。例如，`33` 現在會被比較為等於 `33L`，而之前並未這麼做。
  - 已修正 `ReducingBarrierStep` 中的錯誤，使得如果沒有可用於輸出的元素，現在不會傳回任何值。
  - `select()` 範圍的順序已變更 (順序現在為 `maps`、`side-effects`、`paths`)。這會變更會將 `side-effects` 和 `select` 與具有相同金鑰名稱的 `side-effects` 的 `select` 結合的罕見查詢結果。
  - `bulkSet()` 現在是 GraphSON 通訊協定的一部分。以 `toBulkSet()` 結尾的查詢不適用於較舊的用戶端。
  - `Submit()` 步驟的一個參數化已從 3.4 用戶端移除。

  TinkerPop 3.4 中引入的許多其他變更不會影響目前的 Neptune 行為。例如，`Grimlinio()` 作為一個步驟加入`Traversal`，現在已在中棄用`Graph`，但從未在 Neptune 中啟用。
- 將單一基數頂點屬性支援新增到 Gremlin 的大量載入器 (p. 263)，用於載入屬性圖資料。
- 新增可在大量載入器中覆寫單一基數屬性之現有值的選向。
- 新增擷取 Gremlin 查詢的狀態 (p. 324)，以及取消 Gremlin 查詢 (p. 326)的功能。
- 新增 SPARQL 查詢逾時的查詢提示 (p. 442)。
- 新增在狀態 API 查看執行個體角色的能力 (請參閱 執行個體狀態 (p. 770))。
- 新增對資料庫複製的支援 (請參閱 Neptune 的數據庫克隆 (p. 564))。

這個引擎版本的改善項目

- 改進 SPARQL 查詢說明，從 FROM 子句顯示圖形變數。
- 改進篩選、等於篩選、VALUES 子句和範圍計數的 SPARQL 效能。
- 改進 Gremlin 步驟排序的效能。
- 改進 Gremlin `.repeat.dedup` 周遊的效能。
- 改善 Gremlin `valueMap()` 和 `path().by()` 周遊的效能。

此引擎版本中修正的缺陷

- 修正 SPARQL 屬性路徑的多個問題，包括具名圖形的操作。
- 修正造成記憶體問題的 SPARQL CONSTRUCT 查詢問題。

- 修正 RDF Turtle 剖析器與本機名稱的問題。
- 修正更正顯示給使用者之錯誤訊息的問題。
- 修正 Gremlin `repeat()...drop()` 周遊的問題。
- 修正 Gremlin `drop()` 步驟的問題。
- 修正 Gremlin 標籤篩選條件的問題。
- 修正 Gremlin 查詢逾時的問題。

2019-Amazon Neptune bono 引擎更新 2019-07-02

重要提示：此引擎版本現已棄用

從 2021-04-27 開始，將不會建立使用此引擎版本的新執行個體。

此引擎版本中修正的缺陷

- 修正會導致具有屬性名稱和值的特定模式不會最佳化的錯誤。

早期 Neptune 引擎版本

主題

- Amazon Neptune 引擎更新 2019-06-06-12 (p. 988)
- Amazon Neptune 引擎更新 2019-05-01 (p. 989)
- Amazon Neptune 引擎更新 2019-01-21 (p. 990)
- Amazon Neptune 引擎更新 2019 (p. 990)
- Amazon Neptune 發動台更 2018-08 發動台 (p. 991)
- Amazon Neptune 引擎更新 2018-29 發佈 (p. 992)
- Amazon Neptune 引擎更新 2018-09-06 (p. 992)
- Amazon Neptune 引擎更新 2018-07-19 (p. 993)
- Amazon Neptune 引擎擎擎擎擎擎擎擎擎擎擎 (p. 994)

Amazon Neptune 引擎更新 2019-06-06-12

版本：

Amazon Neptune 1.0.0.0.0.0.0.0.0 正式上市。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200310.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200310.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到幾分鐘的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 主控台上檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support。

改善項目

- 修正同時插入和捨棄邊緣可能導致多個邊緣具有產生相同 ID 的錯誤。
- 其他次要修正和改進。

Amazon Neptune 引擎更新 2019-05-01

版本：

Amazon Neptune 1.0.0.0.296.0 正式上市。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200296.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集將自動升級至引擎版本 1.0.1.0.200296.0。套用更新的時間取決於資料庫叢集的區域和維護時段，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到幾分鐘的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 主控台上檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support。

改善項目

- 新增 Neptune SPARQL 查詢的新`explain`功能，可協助您視覺化查詢計畫，並在必要時採取步驟將其最佳化。如需相關資訊，請參閱 SPARQL explain (p. 453)。
- 以各種方式改善 SPARQL 效能和報告。
- 以各種方式改善 Gremlin 效能和行為。
- 改善長時間執行的 `drop( )` 查詢逾時。
- 改善 `otherV( )` 查詢的效能。
- 在您查詢資料庫叢集或執行個體的 Neptune 健全狀況狀態 (亦即引擎版本號碼和叢集或執行個體的啟動時間) 時，所傳回的資訊新增了兩個欄位。請參閱 執行個體狀態 (p. 770)。
- Neptune 載入器`Get-Status` API 現在會傳回記錄載入工作開始時的`startTime`欄位。
- 載入器命令現在接受選用的 `parallelism` 參數，讓您可以限制載入器使用的執行緒數量。

Amazon Neptune 引擎更新 2019-01-21

版本：1.0.0.267.0

Amazon Neptune 1.0.0.0.267.0 正式推出。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200267.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200267.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

> **Note**
>
> 執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 主控台上檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support 聯繫 Support 聯繫 Support 聯繫 Support

改善項目

- Neptune 會等待更長的時間 (在指定的查詢逾時內)，讓任何衝突得以解決。這可減少用戶端需要處理之並行修改例外狀況的數目 (請參閱查詢錯誤 (p. 849))。
- 已修改 Gremlin 基數強制執行有時會造成引擎重新啟動的問題。
- 已提升 `emit.times` 重複查詢的 Gremlin 效能。
- 已修正 Gremlin 問題，其中 `repeat.until` 允許應該透過其進行篩選的 `.emit` 解決方案。
- 已改善 Gremlin 中的錯誤處理能力。

Amazon Neptune 引擎更新 2019

版本：

Amazon Neptune 1.0.0.0.264.0 正式上市。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200264.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200264.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 主控台上檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support。

改善項目

- 新增了對 the section called “查詢提示” (p. 319) 的支援。
- 改進 IAM 身分驗證的錯誤訊息。如需詳細資訊，請參閱the section called “IAM 錯誤” (p. 852)。
- 提升包含大量述詞的 SPARQL 查詢效能。
- 提升 SPARQL 屬性路徑的效能。
- 提升於 `addV()`、`addE()`、`property()` 步驟所用的 Gremlin 條件式變動 (例如 `fold().coalesce(unfold(), …)` 樣式) 效能。
- 提升 Gremlin 的 `by()` 和 `sack()` 調變效能。
- 提升 Gremlin 的 `group()` 和 `groupCount()` 步驟的效能。
- 提升 Gremlin 的 `store()`、`sideEffect()`、`cap().unfold()` 步驟的效能。
- 改進對 Gremlin 單一基數屬性限制的支援。
  - 改進標示為單一基數屬性的邊緣屬性和頂點屬性的單一基數強制性。
  - 引進錯誤：如果在 Neptune 載入工作中為現有邊緣屬性指定額外的屬性值，會發生錯誤。

Amazon Neptune 發動台更 2018-08 發動台

版本:

Amazon Neptune 1.0.0.0.258.0 正式上市。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200258.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200258.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到幾分鐘的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 台上檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support。

改善項目

- 新增了對 查詢提示 (p. 437) 的支援。

- 提升 SPARQL FILTER (NOT) Exists 查詢的效能。
- 提升 SPARQL DESCRIBE 查詢效能。
- 提升 Gremlin 中 repeat until 模式的效能。
- 提升在 Gremlin 中新增邊緣的效能。
- 修正 SPARQL Update DELETE 查詢在某些情況下可能會失敗的問題。
- 修正了使用 Gemlin WebSocket 伺服器處理逾時的問題。

Amazon Neptune 引擎更新 2018-29 發佈

版本：

Amazon Neptune 1.0.0.0.255.0 正式推出。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200255.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200255.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

> Note
>
> 執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以從 Neptune 29 主控台檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯絡 Support。

改善項目

- 在稽核日誌中新增 IAM 身分驗證資訊。
- 新增對臨時登入資料的支援 (使用 IAM 角色和執行個體描述檔)。
- 新增權限撤銷或 IAM 使用者或角色遭到刪除時 IAM 身份驗證的 WebSocket 連線終止。
- 將每個執行個體的最大 WebSocket 連線數限制為 60,000 個。
- 提升較小執行個體類型的大量載入效能。
- 提升 Gremlin 中包含 `and()`、`or()`、`not()`、`drop()` 運算子的查詢效能。
- NTriples 解析器現在會拒絕無效的 URI，例如包含空格的 URI。

Amazon Neptune 引擎更新 2018-09-06

版本：1.0.0.237.0

Amazon Neptune 1.0.0.0.200237.0 正式推出。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200237.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集會自動升級至引擎版本 1.0.1.0.200237.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以從檢視或變更您的維護時 Neptune 設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯絡 Support。

改善項目

- 已修正部分 SPARQL `COUNT(DISTINCT)` 查詢失敗的問題。
- 已修正具有 `DISTINCT` 子句的 `COUNT`、`SUM` 和 `MIN` 查詢可能會用盡記憶體的問題。
- 已修正 `BLOB` 類型的資料可能導致 Neptune 載入器任務失敗的問題。
- 已修正重複插入可能導致交易失敗的問題。
- 已修正無法將 `DROP ALL` 查詢取消的問題。
- 已修正 Gremlin 用戶端可能間歇性停止回應的問題。
- 已針對承載大於 150M 的情況將所有錯誤碼更新為 `HTTP 400`。
- 提高了 single-triple-patternCOUNT( )查詢的性能和準確性。
- 已改進具有 `BIND` 子句的 SPARQL `UNION` 查詢的效能。

Amazon Neptune 引擎更新 2018-07-19

版本：

Amazon Neptune 1.0.0.0.236.0 正式上市。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200236.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集將自動升級至引擎版本 1.0.1.0.200236.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

Note

執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以從 Neptune upport 檢視您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯絡 Support。

改善項目

- 已更新 `xsd:string` 資料類型的 SPARQL 序列化。xsd:string 不再隨附於 JSON 序列化，目前已與其他輸出格式一致。
- 已修正 `xsd:double`/`xsd:float` 無限的處理問題。`-INF`、`NaN` 和 `INF` 的值現已可在所有 SPARQL 資料載入器格式、SPARQL 1.1 更新及 SPARQL 1.1 查詢中正確加以辨識和處理。
- 包含空字串值的 Gremlin 查詢非預期失敗問題已獲修正。
- 在空圖表上的 Gremlin `aggregate()`和 `cap()` 非預期失敗問題已獲解決。
- 基數規格無效時 (例如 `.property(set,id,'10')`和 `.property(single,id,'10')`)，Gremlin 回傳不正確錯誤回應的問題已獲解決。
- 修正無效的 Gemlin 語法傳回為 InternalFailureException.
- 將錯誤訊息中 `TimeLimitExceeededException`的拼法修正為 `TimeLimitExceededException`。
- 變更 SPARQL 和 GREMLIN 終端節點，在無指令碼時以一致方式回應。
- 釐清並行請求過多的錯誤訊息。

Amazon Neptune 引擎擎擎擎擎擎擎擎擎擎擎

版本：

Amazon Neptune 1.0.0.0.200233.0 通常可用。所有新的 Neptune 資料庫叢集 (包括從快照還原的叢集) 都會在該區域的引擎更新完成後，在 Neptune 1.0.0.200233.0 中建立。

您可以在主控台上使用資料庫叢集操作或使用軟體開發套件，立即將現有叢集升級至此版本。您可以使用下列 CLI 命令，立即將資料庫叢集升級至此版本：

```
aws neptune apply-pending-maintenance-action \
    --apply-action system-update \
    --opt-in-type immediate \
    --resource-identifier arn:aws:rds:<region>:<account number>:<resourcetype>:<name>
```

在系統維護期間，Neptune 資料庫叢集將自動升級至引擎版本 1.0.1.0.200233.0。套用更新的時間取決於資料庫叢集的區域和維護時段設定，以及更新類型。

> Note
>
> 執行個體維護時段無法進行引擎更新。

更新會同時套用到資料庫叢集中的所有執行個體。更新時，資料庫叢集的所有執行個體上需要重新啟動資料庫，因此您會經歷 20 到 30 秒的停機時間，之後就可以繼續使用資料庫叢集。您可以在 Neptune 主控台檢視或變更您的維護時段設定。

如果您有任何問題或疑慮，請透過社群論壇及 AWSPremium SupAWS port 聯繫 Support。

改善項目

- 修正快速連續發出大量大批載入請求會導致錯誤的問題。
- 修正資料相依的問題，即查詢可能會失敗，並顯示 InternalServerError. 以下範例顯示受到影響的查詢類型。

```
g.V("my-id123").as("start").outE("knows").has("edgePropertyKey1",
 P.gt(0)).as("myedge").inV()
               .as("end").select("start", "end", "myedge").by("vertexPropertyKey1")
               .by("vertexPropertyKey1").by("edgePropertyKey1")
```

- 修復了在長時間運行的查詢超時後，Gremlin Java 客戶端無法使用相同的 WebSocket 連接連接到服務器的問題。
- 修正透過 HTTP 進行的 Gemlin 查詢或透過 WebSocket 連線的字串式查詢所包含的逸出序列無法正確處理的問題。

Amazon Neptune 叢集、執行個體及資源管理 API 參考

本章記載您可以用來管理和維護您 Neptune 資料庫叢集的 Neptune API。

Neptune 會在複寫拓撲中連線的資料庫伺服器叢集上運作。因此，管理 Neptune 通常會涉及將變更部署到多個伺服器，並確保所有 Neptune 複本都和主伺服器保持同步。

由於 Neptune 會隨著您的資料增加而透明地擴展基礎儲存體，因此管理 Neptune 只需要進行相對較少的磁碟儲存體管理。同樣地，因為 Neptune 會自動執行連續備份，因此 Neptune 叢集不需要為執行備份進行大量的規劃或停機時間。

內容

- Neptune 資料庫叢集 API (p. 1000)
  - DBCluster (結構) (p. 1001)
  - 資料庫 ClusterMember (結構) (p. 1004)
  - 資料庫 ClusterRole (結構) (p. 1004)
  - ClusterPendingModifiedValues （結構） (p. 1004)
  - CreateDBCluster (動作) (p. 1005)
  - DeleteDBCluster (動作) (p. 1012)
  - ModifyDBCluster (動作) (p. 1017)
  - StartDBCluster (動作) (p. 1023)
  - StopDBCluster (動作) (p. 1027)
  - AddRoleTo資料庫叢集 (動作) (p. 1031)
  - RemoveRoleFrom資料庫叢集 (動作) (p. 1031)
  - FailoverDBCluster (動作) (p. 1032)
  - PromoteReadReplica資料庫叢集 (動作) (p. 1036)
  - DescribeDBClusters (動作) (p. 1040)
- Neptune 全球數據庫 API (p. 1041)
  - GlobalCluster （結構） (p. 1041)
  - CreateGlobalCluster （動作） (p. 1042)
  - DeleteGlobalCluster （動作） (p. 1044)
  - ModifyGlobalCluster （動作） (p. 1045)
  - DescribeGlobalClusters （動作） (p. 1047)
  - FailoverGlobalCluster （動作） (p. 1048)
  - RemoveFromGlobalCluster （動作） (p. 1050)
- Neptune 執行個體 API (p. 1051)
  - DBInstance (結構) (p. 1051)
  - 資料庫InstanceStatusInfo (結構) (p. 1054)
  - 可訂購數據庫InstanceOption （結構） (p. 1055)
  - 驗證碼InstanceModificationsMessage (結構) (p. 1056)
  - CreateDBInstance (動作) (p. 1056)
  - DeleteDBInstance (動作) (p. 1065)
  - ModifyDBInstance (動作) (p. 1069)

        - RebootDBInstance (動作) (p. 1078)
        - DescribeDBInstances (動作) (p. 1082)
        - DescribeOrderable資料庫InstanceOptions (動作) (p. 1083)
        - DescribeValid資料庫InstanceModifications (動作) (p. 1084)
    - Neptune 參數 API (p. 1084)
        - Parameter (結構) (p. 1085)
        - 資ParameterGroup 料庫結構 (p. 1086)
        - 資ClusterParameterGroup 料庫結構 (p. 1086)
        - 資ParameterGroupStatus 料庫結構 (p. 1087)
        - 複製資料庫ParameterGroup (動作) (p. 1087)
        - 複製資料庫ClusterParameterGroup (動作) (p. 1088)
        - 建立資料庫ParameterGroup (動作) (p. 1090)
        - 建立資料庫ClusterParameterGroup (動作) (p. 1091)
        - 刪除資料庫ParameterGroup (動作) (p. 1092)
        - 刪除資料庫ClusterParameterGroup (動作) (p. 1093)
        - 修改資料庫ParameterGroup (動作) (p. 1093)
        - 修改資料庫ClusterParameterGroup (動作) (p. 1094)
        - 重設資料庫ParameterGroup (動作) (p. 1095)
        - 重設資料庫ClusterParameterGroup (動作) (p. 1096)
        - DescribeDBParameters (動作) (p. 1097)
        - 描述 BParameterGroups （ 動作 ） (p. 1098)
        - 描述 BClusterParameters （ 動作 ） (p. 1099)
        - 描述 BClusterParameterGroups （ 動作 ） (p. 1100)
        - DescribeEngineDefaultParameters （ 動作 ） (p. 1101)
        - DescribeEngineDefaultClusterParameters （ 動作 ） (p. 1102)
    - Neptune 子網路 API (p. 1102)
        - Subnet (結構) (p. 1103)
        - 資料庫SubnetGroup (結構) (p. 1103)
        - 建立資料庫SubnetGroup (動作) (p. 1104)
        - 刪除資料庫SubnetGroup (動作) (p. 1105)
        - 修改資料庫SubnetGroup (動作) (p. 1106)
        - 描述 BSubnetGroups （ 動作 ） (p. 1107)
    - Neptune 快照 API (p. 1108)
        - 資料庫 ClusterSnapshot (結構) (p. 1108)
        - 資料庫 ClusterSnapshotAttribute (結構) (p. 1110)
        - 資料庫 ClusterSnapshotAttributesResult (結構) (p. 1110)
        - 建立資料庫 ClusterSnapshot (動作) (p. 1110)
        - 刪除資料庫 ClusterSnapshot (動作) (p. 1113)
        - 複製資料庫 ClusterSnapshot (動作) (p. 1115)
        - 修改資料庫 ClusterSnapshotAttribute (動作) (p. 1117)
        - 復原 B (動ClusterFromSnapshot作) (p. 1119)
        - 復原 B (動ClusterToPointInTime作) (p. 1125)
        - 描述 B （ 動作ClusterSnapshots ） (p. 1131)
        - 描述 B （ 動作ClusterSnapshotAttributes ） (p. 1133)
    - Neptune 事件 API (p. 1134)

        - Event (結構) (p. 1134)
        - EventCategoriesMap （結構） (p. 1134)
        - EventSubscription （結構） (p. 1135)
        - CreateEventSubscription （動作） (p. 1136)
        - DeleteEventSubscription （動作） (p. 1138)
        - ModifyEventSubscription （動作） (p. 1139)
        - DescribeEventSubscriptions （動作） (p. 1141)
        - AddSourceIdentifierToSubscription （動作） (p. 1142)
        - RemoveSourceIdentifierFromSubscription （動作） (p. 1143)
        - DescribeEvents （動作） (p. 1144)
        - DescribeEventCategories （動作） (p. 1146)
    - 其他 Neptune API (p. 1146)
        - CloudwatchLogsExportConfiguration （結構） (p. 1147)
        - 資料庫EngineVersion (結構) (p. 1147)
        - EngineDefaults （結構） (p. 1148)
        - PendingMaintenanceAction （結構） (p. 1148)
        - PendingModifiedValues （結構） (p. 1149)
        - PendingCloudwatchLogsExports （結構） (p. 1149)
        - ResourcePendingMaintenanceActions （結構） (p. 1150)
        - UpgradeTarget （結構） (p. 1150)
        - Tag (結構) (p. 1150)
        - AddTagsToResource （動作） (p. 1151)
        - ListTagsForResource （動作） (p. 1151)
        - RemoveTagsFromResource （動作） (p. 1152)
        - ApplyPendingMaintenanceAction （動作） (p. 1152)
        - DescribePendingMaintenanceActions （動作） (p. 1153)
        - 描述 BEngineVersions （動作） (p. 1154)
    - 常見的 Neptune 資料類型 (p. 1155)
        - AvailabilityZone （結構） (p. 1156)
        - 資料庫 SecurityGroupMembership (結構) (p. 1156)
        - DomainMembership （結構） (p. 1156)
        - DoubleRange （結構） (p. 1156)
        - Endpoint (結構) (p. 1157)
        - Filter (結構) (p. 1157)
        - Range (結構) (p. 1157)
        - 無伺服器 V2 ScalingConfiguration (結構) (p. 1158)
        - 無伺服器 V2 ScalingConfigurationInfo (結構) (p. 1158)
        - Timezone (結構) (p. 1158)
        - VpcSecurityGroupMembership （結構） (p. 1158)
    - 針對個別 API 的 Neptune 例外狀況 (p. 1159)
        - AuthorizationAlreadyExistsFault （結構） (p. 1160)
        - AuthorizationNotFoundFault （結構） (p. 1161)
        - AuthorizationQuotaExceededFault （結構） (p. 1161)
        - CertificateNotFoundFault （結構） (p. 1161)
        - 資料庫ClusterAlreadyExistsFault (結構) (p. 1161)

- 資料庫ClusterNotFoundFault (結構) (p. 1162)
- 資料庫ClusterParameterGroupNotFoundFault (結構) (p. 1162)
- 資料庫ClusterQuotaExceededFault (結構) (p. 1162)
- 資料庫ClusterRoleAlreadyExistsFault (結構) (p. 1162)
- 資料庫ClusterRoleNotFoundFault (結構) (p. 1162)
- 資料庫ClusterRoleQuotaExceededFault (結構) (p. 1163)
- 資料庫ClusterSnapshotAlreadyExistsFault (結構) (p. 1163)
- 資料庫ClusterSnapshotNotFoundFault (結構) (p. 1163)
- 資料庫InstanceAlreadyExistsFault (結構) (p. 1163)
- 資料庫InstanceNotFoundFault (結構) (p. 1164)
- 資料庫LogFileNotFoundFault (結構) (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (結構) (p. 1164)
- 資料庫ParameterGroupNotFoundFault (結構) (p. 1164)
- 資料庫ParameterGroupQuotaExceededFault (結構) (p. 1164)
- 資料庫SecurityGroupAlreadyExistsFault (結構) (p. 1165)
- 資料庫SecurityGroupNotFoundFault (結構) (p. 1165)
- 資料庫SecurityGroupNotSupportedFault (結構) (p. 1165)
- 資料庫SecurityGroupQuotaExceededFault (結構) (p. 1165)
- 資料庫SnapshotAlreadyExistsFault (結構) (p. 1166)
- 資料庫SnapshotNotFoundFault (結構) (p. 1166)
- 資料庫SubnetGroupAlreadyExistsFault (結構) (p. 1166)
- 資料庫SubnetGroupDoesNotCoverEnough可用空間 (結構) (p. 1166)
- 資料庫SubnetGroupNotAllowedFault (結構) (p. 1166)
- 資料庫SubnetGroupNotFoundFault (結構) (p. 1167)
- 資料庫SubnetGroupQuotaExceededFault (結構) (p. 1167)
- 資料庫SubnetQuotaExceededFault (結構) (p. 1167)
- 資料庫UpgradeDependencyFailureFault (結構) (p. 1167)
- DomainNotFoundFault （結構） (p. 1168)
- EventSubscriptionQuotaExceededFault （結構） (p. 1168)
- GlobalClusterAlreadyExistsFault （結構） (p. 1168)
- GlobalClusterNotFoundFault （結構） (p. 1168)
- GlobalClusterQuotaExceededFault （結構） (p. 1168)
- InstanceQuotaExceededFault （結構） (p. 1169)
- 資料庫不足ClusterCapacityFault (結構) (p. 1169)
- 資料庫不足InstanceCapacityFault (結構) (p. 1169)
- InsufficientStorageClusterCapacityFault （結構） (p. 1169)
- 無效ClusterEndpointStateFault (結構) (p. 1170)
- 無效ClusterSnapshotStateFault (結構) (p. 1170)
- 無效ClusterStateFault (結構) (p. 1170)
- 無效InstanceStateFault (結構) (p. 1170)
- 無效ParameterGroupStateFault (結構) (p. 1170)
- 無效SecurityGroupStateFault (結構) (p. 1171)
- 無效SnapshotStateFault (結構) (p. 1171)
- 無效SubnetGroupFault (結構) (p. 1171)
- 無效SubnetGroupStateFault (結構) (p. 1171)

- 無效SubnetStateFault (結構) (p. 1172)
- InvalidEventSubscriptionStateFault （結構） (p. 1172)
- InvalidGlobalClusterStateFault （結構） (p. 1172)
- InvalidOptionGroupStateFault （結構） (p. 1172)
- InvalidRestoreFault （結構） (p. 1172)
- InvalidSubnet （結構） (p. 1173)
- 無效電腦NetworkStateFault （結構） (p. 1173)
- KMSKeyNotAccessibleFault 結構 (p. 1173)
- OptionGroupNotFoundFault （結構） (p. 1173)
- PointInTimeRestoreNotEnabledFault （結構） (p. 1174)
- ProvisionedIopsNotAvailableInAZ 錯誤 （結構） (p. 1174)
- ResourceNotFoundFault （結構） (p. 1174)
- 社群網站InvalidTopicFault (結構) (p. 1174)
- 社群網站NoAuthorizationFault (結構) (p. 1174)
- 社群網站TopicArnNotFoundFault (結構) (p. 1175)
- SharedSnapshotQuotaExceededFault （結構） (p. 1175)
- SnapshotQuotaExceededFault （結構） (p. 1175)
- SourceNotFoundFault （結構） (p. 1175)
- StorageQuotaExceededFault （結構） (p. 1176)
- StorageTypeNotSupportedFault （結構） (p. 1176)
- SubnetAlreadyInUse （結構） (p. 1176)
- SubscriptionAlreadyExistFault （結構） (p. 1176)
- SubscriptionCategoryNotFoundFault （結構） (p. 1176)
- SubscriptionNotFoundFault （結構） (p. 1177)

Neptune 資料庫叢集 API

結構：

- DBCluster (結構) (p. 1001)
- 資料庫 ClusterMember (結構) (p. 1004)
- 資料庫 ClusterRole (結構) (p. 1004)
- ClusterPendingModifiedValues （結構） (p. 1004)

動作:

- CreateDBCluster (動作) (p. 1005)
- DeleteDBCluster (動作) (p. 1012)
- ModifyDBCluster (動作) (p. 1017)
- StartDBCluster (動作) (p. 1023)
- StopDBCluster (動作) (p. 1027)
- AddRoleTo資料庫叢集 (動作) (p. 1031)
- RemoveRoleFrom資料庫叢集 (動作) (p. 1031)
- FailoverDBCluster (動作) (p. 1032)
- PromoteReadReplica資料庫叢集 (動作) (p. 1036)
- DescribeDBClusters (動作) (p. 1040)

DBCluster (結構)

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

欄位

- AllocatedStorage – 數字 (整數)。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles— 資料庫 ClusterRole (p. 1004) DB ClusterRole 物件的陣列。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime – 時間戳記。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZonesUTF-8 字串陣列。AvailabilityZone

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords – 數字 (long)。

  Neptune 不提供支援。
- BacktrackWindow – 數字 (long)。

  Neptune 不提供支援。
- BackupRetentionPeriod – 數字 (整數)。

  指定保留自動資料庫快照的天數。
- Capacity – 數字 (整數)。

  Neptune 不提供支援。
- CloneGroupId – UTF-8 字串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime – 時間戳記。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot – 布林值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone – 布林值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName – UTF-8 字串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn – UTF-8 字串。

  資料庫叢集的 Amazon Resource Name (ARN)。
- DBClusterIdentifier – UTF-8 字串。

包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。

- DBClusterMembers— 資料庫 ClusterMember (p. 1004) DB ClusterMember 物件的陣列。

  提供組成資料庫叢集的執行個體清單。
- DBClusterParameterGroup – UTF-8 字串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId – UTF-8 字串。

  資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。
- DBSubnetGroup – UTF-8 字串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。
- DeletionProtection – 布林值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime – 時間戳記。

  Neptune 不提供支援。
- EarliestRestorableTime – 時間戳記。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports – UTF-8 字串陣列。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint – UTF-8 字串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine – UTF-8 字串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion – UTF-8 字串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier— 符合規則運算式的 UTF-8 字串，且長度不可小於 1 個位元組，也不可超過 255 個位元組：'[A-Za-z] [0-9A-Za-z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。
- HostedZoneId – UTF-8 字串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- IAMDatabaseAuthenticationEnabled – 布林值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId – UTF-8 字串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。
- LatestRestorableTime – 時間戳記。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ – 布林值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues – ClusterPendingModifiedValues (p. 1004) 物件。

  此資料類型在動ModifyDBCluster作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。
- PercentProgress – UTF-8 字串。

  指定操作的進度 (百分比)。
- Port – 數字 (整數)。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow – UTF-8 字串。

  指定每日的時間範圍，若有啟用自動化備份 (由 BackupRetentionPeriod 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow – UTF-8 字串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint – UTF-8 字串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiersUTF-8 字串陣列。ReadReplicaIdentifier

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier – UTF-8 字串。

  Neptune 不提供支援。
- ReplicationType – UTF-8 字串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration – 無伺服器 V2 ScalingConfigurationInfo (p. 1158) 物件。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status – UTF-8 字串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted – 布林值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups— 物VpcSecurityGroupMembership (p. 1158)VpcSecurityGroupMembership件的陣列。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

DBCluster 會用來做為以下項目的回應元素：

- CreateDBCluster (p. 1005)
- DeleteDBCluster (p. 1012)
- FailoverDBCluster (p. 1032)

- ModifyDBCluster (p. 1017)
- PromoteReadReplica数据库集群 (p. 1036)
- 恢復 ClusterFromSnapshot (p. 1119)
- 恢復 ClusterToPointInTime (p. 1125)
- StartDBCluster (p. 1023)
- StopDBCluster (p. 1027)

資料庫 ClusterMember (結構)

包含做為資料庫叢集一部分的執行個體相關資訊。

欄位

- DBClusterParameterGroupStatus – UTF-8 字串。

  指定此資料庫叢集成員的資料庫叢集參數群組狀態。
- DBInstanceIdentifier – UTF-8 字串。

  指定此資料庫叢集成員的執行個體識別符。
- IsClusterWriter – 布林值。

  若叢集成員是資料庫叢集的主要執行個體，則該值為 `true`，否則為 `false`。
- PromotionTier – 數字 (整數)。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。

資料庫 ClusterRole (結構)

描述與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色。

欄位

- FeatureName – UTF-8 字串。

  與 Amazon Identity and Access Management (IAM) 角色關聯的功能名稱。如需支援的功能名稱清單，請參閱 the section called “描述 BEngineVersions” (p. 1154)。
- RoleArn – UTF-8 字串。

  與資料庫叢集相關聯的 IAM 角色 Amazon Resource Name (ARN)。
- Status – UTF-8 字串。

  描述 IAM 角色和資料庫叢集之間關聯的狀態。Status 屬性會傳回下列其中一個值：
  - `ACTIVE`-IAM 角色 ARN 與資料庫叢集相關聯，您可以用來代表您存取其他 Amazon 服務。
  - `PENDING` - IAM 角色 ARN 正在和資料庫叢集建立關聯。
  - `INVALID`-IAM 角色 ARN 與資料庫叢集相關聯，但是資料庫叢集無法擔任 IAM 角色，以代表您存取其他 Amazon 服務。

ClusterPendingModifiedValues (結構)

此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。

欄位

- AllocatedStorage – 數字 (整數)。

  資料庫引擎的分配儲存體大小，以 GiB 為單位。對於 Neptune，`AllocatedStorage`始終返回 1，因為 Neptune DB 群集存儲大小不固定，而是根據需要自動調整。
- BackupRetentionPeriod – 數字 (整數)。

  自動資料庫快照保留自動資料庫快照的天數。
- DBClusterIdentifier – UTF-8 字串。

  資料庫叢集的資料庫叢集的資料庫ClusterIdentifier值。
- EngineVersion – UTF-8 字串。

  資料庫引擎版本。
- IAMDatabaseAuthenticationEnabled – 布林值。

  指示是否啟用將 AWS Identity and Access Management (IAM) 帳戶映射到資料庫帳戶的值。
- Iops – 數字 (整數)。

  已佈建 IOPS (每秒 I/O 操作數) 值。此設定僅適用於非 Aurora 多可用區域資料庫叢集。
- PendingCloudwatchLogsExports – PendingCloudwatchLogsExports (p. 1149) 物件。

  此`PendingCloudwatchLogsExports`結構會指定已啟用CloudWatch日誌的待處理變更，哪些日誌已停用哪些變更。

CreateDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`create-db-cluster`，Python 的名稱是：`create_db_cluster`。

建立新的 Amazon Neptune 資料庫叢集。

您可以使用 `ReplicationSourceIdentifier` 參數來建立資料庫叢集，做為另一個資料庫叢集或 Amazon Neptune 資料庫執行個體的僅供讀取複本。

請注意，當您直接使用 `CreateDBCluster` 建立新叢集時，預設會停用刪除保護 (當您在主控台中建立新的生產叢集時，預設會啟用刪除保護)。如果資料庫叢集的 `DeletionProtection` 欄位設定為 `false`，您只能刪除資料庫叢集。

請求

- AvailabilityZones (在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`) - 一個數組 AvailabilityZone-UTF-8 字符串。

  EC2 可用區域的清單，在這些可用區域中可以建立資料庫叢集中的執行個體。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  自動備份保留的天數。您必須指定一個值 (最小值為 1)。

  預設：1

  約束:
  - 該值必須介於 1 到 35 之間
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

如果設定為*true*，則會將標籤複製到所建立之資料庫叢集的任何快照。

- DatabaseName（在 CLI 中：`--database-name`或者在 Python 中：`database_name`）-UTF-8 字符串。

  您資料庫的名稱最長可達 64 個英數字元。若您沒有提供名稱，Amazon Neptune 將不會在您建立的資料庫叢集中建立資料庫。
- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-必需：UTF-8 字符串。

  資料庫叢集識別碼 此參數是以小寫字母字串的形式儲存。

  約束:

  - 必須包含 1 到 63 個字母、數字或連字號。
  - 第一個字元必須是字母。
  - 不能以連字號結尾或連續包含兩個連字號。

  範例：`my-cluster1`
- DBClusterParameterGroupName（在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`）-UTF-8 字符串。

  要與此資料庫叢集建立關聯的資料庫叢集參數群組名稱。若省略此引數，則會使用預設。

  約束:

  - 如果提供，必須符合現有資料庫的名稱ClusterParameterGroup。
- DBSubnetGroupName（在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`）-UTF-8 字符串。

  要與此資料庫叢集建立關聯的資料庫子網路群組。

  限制條件：必須符合現有資料庫的名稱SubnetGroup。絕不能為預設值。

  範例：`mySubnetgroup`
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是啟用的。
- EnableCloudwatchLogsExports（在 CLI 中：`--enable-cloudwatch-logs-exports`或在 Python 中：`enable_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  需要啟用以匯出至 Logs 的CloudWatch日誌類型清單。
- EnableIAMDatabaseAuthentication（在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`）-布爾值。

  如果設為`true`，則為整個資料庫叢集啟用 Amazon Identity and Access Management (IAM) 身份驗證 (無法在執行個體層級設定)。

  預設：`false`。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-必需：UTF-8 字符串。

  要用於此資料庫叢集的資料庫引擎名稱。

  有效值: `neptune`
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

要用於新資料庫叢集的資料庫引擎版本號碼。

範例：1.0.2.1

- GlobalClusterIdentifier（在 CLI 中：--global-cluster-identifier或在 Python 中：global_cluster_identifier）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  應新增此新資料庫叢集之 Neptune 全域資料庫的識別碼。

- KmsKeyId（在 CLI 中：--kms-key-id或者在 Python 中：kms_key_id）-UTF-8 字符串。

  加密資料庫叢集的 Amazon KMS 金鑰識別符。

  KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。若您使用擁有用來加密新資料庫叢集 KMS 加密金鑰的相同 Amazon 帳戶建立資料庫叢集，您可以改為使用 KMS 金鑰別名，而非 KMS 加密金鑰的 ARN。

  若加密金鑰並未在 KmsKeyId 中指定：

  - 若 ReplicationSourceIdentifier 可識別加密來源，則 Amazon Neptune 會使用用於加密來源的加密金鑰。否則，Amazon Neptune 將使用您的預設加密金鑰。
  - 若 StorageEncrypted 參數為 true 且並未指定 ReplicationSourceIdentifier，則 Amazon Neptune 會使用您的預設加密金鑰。

  Amazon KMS 會為您的亞馬遜帳戶建立預設加密金鑰。您的 Amazon 帳戶在每個 Amazon 區域各有不同的預設加密金鑰。

  如果您在另一個 Amazon 區域建立加密資料庫叢集的僅供讀取複本，則必KmsKeyId須設定為在目的地 Amazon 區域有效的 KMS 金鑰 ID。此金鑰用於加密該 Amazon 區域中的僅供讀取複本。

- Port（在 CLI 中：--port，或在 Python 中：port）-數字（整數）。

  資料庫叢集中執行個體接受連線的連接埠號碼。

  預設：8182

- PreferredBackupWindow（在 CLI 中：--preferred-backup-window或者在 Python 中：preferred_backup_window）-UTF-8 字符串。

  每日的時間範圍，若使用 BackupRetentionPeriod 參數啟用自動化備份，則會在此期間建立自動化備份。

  預設是從每個 Amazon 區域 8 小時時段內隨機選取的 30 分鐘時段。若要查看可用的時段，請參閱《Amazon Neptune 使用者指南》中的海王星維護時段。

  約束:

  - 格式必須為 hh24:mi-hh24:mi。
  - 必須以國際標準時間 (UTC) 表示。
  - 不得和慣用的維護時段衝突。
  - 必須至少 30 分鐘。

- PreferredMaintenanceWindow（在 CLI 中：--preferred-maintenance-window或者在 Python 中：preferred_maintenance_window）-UTF-8 字符串。

  每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

  格式：ddd:hh24:mi-ddd:hh24:mi

  預設是從各 Amazon 亞馬遜區域 8 小時段中隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。若要查看可用的時段，請參閱《Amazon Neptune 使用者指南》中的海王星維護時段。

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日。

限制條件：必須至少是 30 分鐘的時段。

- PreSignedUrl（在 CLI 中：`--pre-signed-url`或者在 Python 中：`pre_signed_url`）-UTF-8 字符串。

  目前不支援此參數。
- ReplicationSourceIdentifier（在 CLI 中：`--replication-source-identifier`或者在 Python 中：`replication_source_identifier`）-UTF-8 字符串。

  如果此資料庫叢集已做為僅供讀取複本建立，則為來源資料庫執行個體或資料庫叢集的 Amazon Resource Name (ARN)。
- ServerlessV2ScalingConfiguration（在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`）-一個無伺服器 V2 ScalingConfiguration (p. 1158)對象。

  包含 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- StorageEncrypted（在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`）-布爾值。

  指定是否要加密資料庫叢集。
- Tags（在 CLI 中：`--tags`或在 Python 中：`tags`）-Tag (p. 1150) 標籤對象的數組。

  要指派給新資料庫叢集的標籤。
- VpcSecurityGroupIds（在 CLI 中：`--vpc-security-group-ids`或者在 Python 中：`vpc_security_group_ids`）-一個數組 VpcSecurityGroupId-UTF-8 字符串。

  與此資料庫叢集關聯的 EC2 VPC 安全群組清單。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。

- BacktrackConsumedChangeRecords (在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`) -數字 (長) 。

  Neptune 不提供支援。
- BacktrackWindow (在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`) -數字 (長) 。

  Neptune 不提供支援。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- Capacity (在 CLI 中：`--capacity`，或在 Python 中：`capacity`) -數字 (整數) 。

  Neptune 不提供支援。
- CloneGroupId (在 CLI 中：`--clone-group-id`或者在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName (在 CLI 中：`--database-name`或者在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn (在 CLI 中：`--db-cluster-arn`或者在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。
- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。
- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或者在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或者在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或者在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。

- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。

- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。

- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。

- Endpoint（在 CLI 中：`--endpoint`或者在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。

- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。

- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。

- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。

- HostedZoneId（在 CLI 中：`--hosted-zone-id`或者在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。

- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId（在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。

- LatestRestorableTime（在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`）-時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。

- MultiAZ（在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`）-布爾值。

指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。
- PercentProgress (在 CLI 中：`--percent-progress`或者在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或者在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或者在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

指定是否要加密資料庫叢集。

- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterAlreadyExistsFault (p. 1161)
- InsufficientStorageClusterCapacityFault (p. 1169)
- 資料庫ClusterQuotaExceededFault (p. 1162)
- StorageQuotaExceededFault (p. 1176)
- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 无效NetworkStateFault (p. 1173)
- 无效ClusterStateFault (p. 1170)
- 无效SubnetGroupStateFault (p. 1171)
- InvalidSubnet (p. 1173)
- 无效InstanceStateFault (p. 1170)
- 資料庫ClusterParameterGroupNotFoundFault (p. 1162)
- 公里KeyNotAccessibleFault (p. 1173)
- 資料庫ClusterNotFoundFault (p. 1162)
- 資料庫InstanceNotFoundFault (p. 1164)
- 資料庫SubnetGroupDoesNotCoverEnough可用區 (p. 1166)
- GlobalClusterNotFoundFault (p. 1168)
- InvalidGlobalClusterStateFault (p. 1172)

DeleteDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`delete-db-cluster`，Python 的名稱是：`delete_db_cluster`。

DeleteDBCluster 動作會刪除先前佈建的資料庫叢集。當您刪除資料庫叢集時，所有該資料庫叢集的自動化備份都會遭到刪除，並且無法復原。指定資料庫叢集的手動資料庫叢集快照則不會刪除。

請注意，如果刪除保護已經啟用，資料庫叢集將無法刪除。如果要刪除，您必須先將其 `DeletionProtection` 欄位設定為 `False`。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  欲刪除資料庫叢集的資料庫叢集識別符。此參數沒有大小寫之分。

  約束:

  - 必須符合現有的資料庫ClusterIdentifier。
- FinalDBSnapshotIdentifier (在 CLI 中：`--final-db-snapshot-identifier`或在 Python 中：`final_db_snapshot_identifier`) -UTF-8 字符串。

當 SkipFinalSnapshot 設為 false 時，所建立新資料庫叢集快照的資料庫叢集快照識別符。

Note

指定此參數，同時又將 SkipFinalShapshot 參數設為 true 時，便會導致錯誤。

約束:

- 必須為 1 到 255 個字母、數字或連字號。
- 第一個字元必須是字母
- 不能以連字號結尾或連續包含兩個連字號

- SkipFinalSnapshot（在 CLI 中：--skip-final-snapshot，或者在 Python 中：skip_final_snapshot）-布爾值。

  判斷最終資料庫叢集快照是否建立於資料庫叢集刪除之前。若指定 true，則不會建立資料庫叢集快照。若指定 false，則資料庫叢集快照會在刪除資料庫叢集前建立。

  Note

  若 SkipFinalSnapshot 是 false，則您必須指定 FinalDBSnapshotIdentifier 參數。

  預設：false

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：--allocated-storage，或在 Python 中：allocated_storage）-數字（整數）。

  AllocatedStorage 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。

- AssociatedRoles（在 CLI 中：--associated-roles或在 Python 中：associated_roles）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。

- AutomaticRestartTime（在 CLI 中：--automatic-restart-time，或在 Python 中：automatic_restart_time）-時間戳。

  資料庫叢集自動重新啟動的時間。

- AvailabilityZones（在 CLI 中：--availability-zones或者在 Python 中：availability_zones）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。

- BacktrackConsumedChangeRecords（在 CLI 中：--backtrack-consumed-change-records或在 Python 中：backtrack_consumed_change_records）-數字（長）。

  Neptune 不提供支援。

- BacktrackWindow（在 CLI 中：--backtrack-window或在 Python 中：backtrack_window）-數字（長）。

  Neptune 不提供支援。

- BackupRetentionPeriod（在 CLI 中：--backup-retention-period，或在 Python 中：backup_retention_period）-數字（整數）。

指定保留自動資料庫快照的天數。

- Capacity (在 CLI 中：`--capacity`，或在 Python 中：`capacity`) -數字 (整數) 。

  Neptune 不提供支援。

- CloneGroupId (在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。

- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。

- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。

- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。

- DatabaseName (在 CLI 中：`--database-name`或在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。

- DBClusterArn (在 CLI 中：`--db-cluster-arn`或在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。

- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。

- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。

- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup (在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`) -UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。

- DeletionProtection (在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`) -布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。

- EarliestBacktrackTime (在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`) -時間戳。

Neptune 不提供支援。

- EarliestRestorableTime (在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`) -時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。

- EnabledCloudwatchLogsExports (在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`) -UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。

- Endpoint (在 CLI 中：`--endpoint`或在 Python 中：`endpoint`) -UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。

- Engine (在 CLI 中：`--engine`或在 Python 中：`engine`) -UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。

- EngineVersion (在 CLI 中：`--engine-version`或在 Python 中：`engine_version`) -UTF-8 字符串。

  指示資料庫引擎版本。

- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) -UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。

- HostedZoneId (在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`) -UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。

- IAMDatabaseAuthenticationEnabled (在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`) -布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId (在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`) -UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。

- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。

- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。

- PercentProgress (在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。

- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。

- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- 資料庫ClusterSnapshotAlreadyExistsFault (p. 1163)

- SnapshotQuotaExceededFault (p. 1175)
- 无效ClusterSnapshotStateFault (p. 1170)

ModifyDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-cluster`，Python 的名稱是：`modify_db_cluster`。

修改資料庫叢集的設定。您可以透過在請求中指定這些參數及新的值，來變更一或多個資料庫組態參數。

請求

- AllowMajorVersionUpgrade（在 CLI 中：`--allow-major-version-upgrade`，或者在 Python 中：`allow_major_version_upgrade`）-布爾值。

  指示是否允許升級不同主要版本的值。

  限制：您在提供使用不同於資料庫叢集群組目前主要版本的主要版本時，您必須設定allow-major-version-upgrade標誌。`EngineVersion`
- ApplyImmediately（在 CLI 中：`--apply-immediately`，或者在 Python 中：`apply_immediately`）-布爾值。

  指定此請求中的修改及任何擱置中的請求是否要以非同步方式盡快套用的值，無論資料庫叢集的 `PreferredMaintenanceWindow` 設定為何。若將此參數設為 `false`，對資料庫叢集的變更會在下次維護時段期間才會套用。

  `ApplyImmediately`參數只會影響`NewDBClusterIdentifier`值。如果將`ApplyImmediately`參數值設定為 false，則會在下一個維護時段套用對`NewDBClusterIdentifier`值的變更。所有其他的變更都會立即套用，無論 `ApplyImmediately` 參數的值為何。

  預設：`false`
- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`）-數字（整數）。

  自動備份保留的天數。您必須指定一個值 (最小值為 1)。

  預設：1

  約束:
  - 該值必須介於 1 到 35 之間
- CloudwatchLogsExportConfiguration（在 CLI 中：`--cloudwatch-logs-export-configuration`或在 Python 中：`cloudwatch_logs_export_configuration`）-一個CloudwatchLogsExportConfiguration (p. 1147)對象。

  要啟用的日誌類型的組態設定，適用於匯出到特定資料庫叢集的 CloudWatch Logs。
- CopyTagsToSnapshot（在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`）-布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-必需：UTF-8 字符串。

  欲修改叢集的資料庫叢集識別符。此參數不區分大小寫。

  約束:
  - 必須符合現有 DBCluster 的識別碼。

- DBClusterParameterGroupName（在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`）-UTF-8 字符串。

  資料庫叢集欲使用的資料庫叢集參數群組名稱。
- DBInstanceParameterGroupName（在 CLI 中：`--db-instance-parameter-group-name`或在 Python 中：`db_instance_parameter_group_name`）-UTF-8 字符串。

  要套用至所有資料庫叢集執行個體的資料庫參數群組名稱。

  Note

  當您使用套用參數群組時`DBInstanceParameterGroupName`，參數變更不會在下一個維護時段套用，而是會立即套用。

  預設：現有名稱設定

  約束:
  - 資料庫參數群組必須位於與目標資料庫叢集版本相同的資料庫參數群組系列中。
  - `DBInstanceParameterGroupName`參數僅與`AllowMajorVersionUpgrade`參數結合使用時才有效。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是停用的。
- EnableIAMDatabaseAuthentication（在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`）-布爾值。

  True，若有啟用從 Amazon Identity of Access Management (IAM) 帳戶映射到資料庫帳戶，否則為 True。

  預設：`false`
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  您要升級的資料庫引擎版本號碼。變更此參數會造成中斷。變更會在下一次維護時段期間套用，除非 `ApplyImmediately` 參數已設為 true。

  如需有效引擎版本的清單，請參閱 Amazon Neptune 的引擎版本，或撥打the section called “描述 BEngineVersions” (p. 1154)。
- NewDBClusterIdentifier（在 CLI 中：`--new-db-cluster-identifier`或在 Python 中：`new_db_cluster_identifier`）-UTF-8 字符串。

  重新命名資料庫叢集時的新資料庫叢集識別符。此值會以小寫字母字串的形式儲存。

  約束:
  - 必須包含 1 到 63 個字母、數字或連字號
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

  範例：`my-cluster2`
- Port（在 CLI 中：`--port`，或在 Python 中：`port`）-數字（整數）。

  資料庫叢集接受連線連接埠號碼。

  限制條件：值必須為 `1150-65535`

  預設值：與原始資料庫叢集相同的連接埠。

- PreferredBackupWindow（在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`）-UTF-8 字符串。

  每日的時間範圍，若使用 `BackupRetentionPeriod` 參數啟用自動化備份，則會在此期間建立自動化備份。

  預設是從每個 Amazon 區域 8 小時時段內隨機選取的 30 分鐘時段。

  約束:
  - 格式必須為 `hh24:mi-hh24:mi`。
  - 必須以國際標準時間 (UTC) 表示。
  - 不得和慣用的維護時段衝突。
  - 必須至少 30 分鐘。
- PreferredMaintenanceWindow（在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`）-UTF-8 字符串。

  每週可能進行系統維護的時段，以國際標準時間 (UTC) 表示。

  格式：`ddd:hh24:mi-ddd:hh24:mi`

  預設是從各 Amazon 亞馬遜區域 8 小時段中隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。

  有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日。

  限制條件：必須至少是 30 分鐘的時段。
- ServerlessV2ScalingConfiguration（在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`）-一個無伺服器 V2 ScalingConfiguration (p. 1158)對象。

  包含 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- VpcSecurityGroupIds（在 CLI 中：`--vpc-security-group-ids`或者在 Python 中：`vpc_security_group_ids`）-一個數組 VpcSecurityGroupId-UTF-8 字符串。

  資料庫叢集將歸屬的 VPC 安全群組清單。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。

- AvailabilityZones (在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`) - 一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords (在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`) -數字 (長) 。

  Neptune 不提供支援。
- BacktrackWindow (在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`) -數字 (長) 。

  Neptune 不提供支援。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- Capacity (在 CLI 中：`--capacity`，或在 Python 中：`capacity`) -數字 (整數) 。

  Neptune 不提供支援。
- CloneGroupId (在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName (在 CLI 中：`--database-name`或在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn (在 CLI 中：`--db-cluster-arn`或在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。
- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。
- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。
- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。
- HostedZoneId（在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId（在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。
- LatestRestorableTime（在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`）-時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ（在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`）-布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。
- PercentProgress (在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。

- VpcSecurityGroups（在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`）-VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- StorageQuotaExceededFault (p. 1176)
- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 无效NetworkStateFault (p. 1173)
- 无效SubnetGroupStateFault (p. 1171)
- InvalidSubnet (p. 1173)
- 資料庫ClusterParameterGroupNotFoundFault (p. 1162)
- 无效SecurityGroupStateFault (p. 1171)
- 无效InstanceStateFault (p. 1170)
- 資料庫ClusterAlreadyExistsFault (p. 1161)

StartDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`start-db-cluster`，Python 的名稱是：`start_db_cluster`。

啟動使用 Amazon Neptune 主控台、Amazon CLI stop-db-cluster 命令或 StopDBCluster API 停止的 Amazon 海王星資料庫叢集叢集叢集。

請求

- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-必需：UTF-8 字符串。

  要啟動的 Neptune 資料庫叢集的資料庫叢集識別符。此參數是以小寫字母字串的形式儲存。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。

- AvailabilityZones (在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`) - 一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords (在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`) -數字 (長) 。

  Neptune 不提供支援。
- BacktrackWindow (在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`) -數字 (長) 。

  Neptune 不提供支援。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- Capacity (在 CLI 中：`--capacity`，或在 Python 中：`capacity`) -數字 (整數) 。

  Neptune 不提供支援。
- CloneGroupId (在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName (在 CLI 中：`--database-name`或在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn (在 CLI 中：`--db-cluster-arn`或在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。
- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。
- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。
- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。
- HostedZoneId（在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId（在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。
- LatestRestorableTime（在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`）-時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ（在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`）-布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。
- PercentProgress (在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。

- VpcSecurityGroups（在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`）-VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- 无效InstanceStateFault (p. 1170)

StopDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`stop-db-cluster`，Python 的名稱是：`stop_db_cluster`。

停止 Amazon Neptune 資料庫叢集。當您停止資料庫叢集時，Neptune 會留資料庫叢集的中繼資料，包括端點和資料庫參數群組。

Neptune 也會保留交易日誌，以便您在需要時執行point-in-time還原。

請求

- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-必需：UTF-8 字符串。

  要停止的 Neptune 資料庫叢集的資料庫叢集識別符。此參數是以小寫字母字串的形式儲存。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords（在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`）-數字（長）。

Neptune 不提供支援。

- BacktrackWindow (在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`) -數字 (長) 。

  Neptune 不提供支援。

- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。

- Capacity (在 CLI 中：`--capacity`，或在 Python 中：`capacity`) -數字 (整數) 。

  Neptune 不提供支援。

- CloneGroupId (在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。

- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。

- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。

- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。

- DatabaseName (在 CLI 中：`--database-name`或在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。

- DBClusterArn (在 CLI 中：`--db-cluster-arn`或在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。

- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。

- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。

- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup (在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`) -UTF-8 字符串。

指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。
- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。
- HostedZoneId（在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId（在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。
- LatestRestorableTime（在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`）-時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ（在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`）-布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。
- PendingModifiedValues（在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`）-一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。

- PercentProgress (在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- 无效InstanceStateFault (p. 1170)

AddRoleTo資料庫叢集 (動作)

這個 API 的亞馬遜 CLI 名稱是：`add-role-to-db-cluster`，Python 的名稱是：`add_role_to_db_cluster`。

建立 Identity and Access Management (IAM) 角色與 Neptune 資料庫叢集的關聯。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  要與 IAM 角色建立關聯的資料庫叢集名稱。
- FeatureName (在 CLI 中：`--feature-name`或在 Python 中：`feature_name`) -UTF-8 字符串。

  IAM 角色要與之相關聯的 Neptune 資料庫叢集之功能名稱。如需支援的功能名稱清單，請參閱 the section called “資料庫EngineVersion” (p. 1147)。
- RoleArn (在 CLI 中：`--role-arn`或在 Python 中：`role_arn`) -必需：UTF-8 字符串。

  要與 Neptune 資料庫叢集建立關聯的 IAM 角色 Amazon Resource Name (ARN)，例如 `arn:aws:iam::123456789012:role/NeptuneAccessRole`。

回應

- 無回應參數。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 資料庫ClusterRoleAlreadyExistsFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- 資料庫ClusterRoleQuotaExceededFault (p. 1163)

RemoveRoleFrom資料庫叢集 (動作)

這個 API 的亞馬遜 CLI 名稱是：`remove-role-from-db-cluster`，Python 的名稱是：`remove_role_from_db_cluster`。

從資料庫叢集解除關聯一個 Identity and Access Management (IAM) 角色。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  要從 IAM 角色解除關聯的資料庫叢集名稱。

- FeatureName (在 CLI 中：`--feature-name`或在 Python 中：`feature_name`) -UTF-8 字符串。

  要取消關聯 IAM 角色之資料庫叢集的功能名稱。如需支援的功能名稱清單，請參閱 the section called “描述 BEngineVersions” (p. 1154)。
- RoleArn (在 CLI 中：`--role-arn`或在 Python 中：`role_arn`) -必需：UTF-8 字符串。

  要從資料庫叢集解除關聯的 IAM 角色 Amazon Resource Name (ARN)，例如 `arn:aws:iam::123456789012:role/NeptuneAccessRole`。

回應

- 無回應參數。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 資料庫ClusterRoleNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)

FailoverDBCluster (動作)

這個 API 的亞馬遜 CLI 名稱是：`failover-db-cluster`，Python 的名稱是：`failover_db_cluster`。

強制資料庫叢集進行容錯移轉。

資料庫叢集的容錯移轉會將資料庫叢集中的其中一個僅供讀取複本 (唯讀執行個體) 提升為主要執行個體 (叢集寫入器)。

Amazon Neptune 會在主要執行個體失敗時，自動容錯移轉至僅供讀取複本 (若有的話)。當您想要模擬主要執行個體故障以進行測試時，可以強制容錯移轉。由於資料庫叢集中的每個執行個體都有自己的端點地址，當容錯移轉完成時，您將需要清除和重新建立任何使用這些端點地址的現有連線。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  要強制其進行容錯移轉的資料庫叢集識別符。此參數不區分大小寫。

  約束:
  - 必須符合現有 DBCluster 的識別碼。
- TargetDBInstanceIdentifier (在 CLI 中：`--target-db-instance-identifier`或在 Python 中：`target_db_instance_identifier`) -UTF-8 字符串。

  要提升為主要執行個體的執行個體名稱。

  您必須為資料庫叢集中的僅供讀取複本指定執行個體識別符。例如：`mydbcluster-replica1`。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數[資料庫 ClusterRole (p. 1004)](#)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords（在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`）-數字（長）。

  Neptune 不提供支援。
- BacktrackWindow（在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`）-數字（長）。

  Neptune 不提供支援。
- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`）-數字（整數）。

  指定保留自動資料庫快照的天數。
- Capacity（在 CLI 中：`--capacity`，或在 Python 中：`capacity`）-數字（整數）。

  Neptune 不提供支援。
- CloneGroupId（在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`）-UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime（在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`）-時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot（在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`）-布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone（在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`）-布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName（在 CLI 中：`--database-name`或在 Python 中：`database_name`）-UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn（在 CLI 中：`--db-cluster-arn`或在 Python 中：`db_cluster_arn`）-UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。

- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。
- DBClusterMembers（在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`）-數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。
- DBClusterParameterGroup（在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`）-UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId（在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`）-UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在每次存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。
- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。
- HostedZoneId（在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。

- IAMDatabaseAuthenticationEnabled (在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`) -布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId (在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`) -UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。
- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動`ModifyDBCluster`作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。
- PercentProgress (在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`，或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或在 Python 中：`replication_type`) -UTF-8 字符串。

Neptune 不提供支援。

- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)
- 无效InstanceStateFault (p. 1170)

PromoteReadReplica資料庫叢集 (動作)

這個 API 的亞馬遜 CLI 名稱是：`promote-read-replica-db-cluster`，Python 的名稱是：`promote_read_replica_db_cluster`。

不支援.

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  不支援.

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage (在 CLI 中：`--allocated-storage`，或在 Python 中：`allocated_storage`) -數字 (整數) 。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles (在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`) -數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。

- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。

- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。

- BacktrackConsumedChangeRecords（在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`）-數字（長）。

  Neptune 不提供支援。

- BacktrackWindow（在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`）-數字（長）。

  Neptune 不提供支援。

- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`）-數字（整數）。

  指定保留自動資料庫快照的天數。

- Capacity（在 CLI 中：`--capacity`，或在 Python 中：`capacity`）-數字（整數）。

  Neptune 不提供支援。

- CloneGroupId（在 CLI 中：`--clone-group-id`或在 Python 中：`clone_group_id`）-UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。

- ClusterCreateTime（在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`）-時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。

- CopyTagsToSnapshot（在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`）-布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。

- CrossAccountClone（在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`）-布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。

- DatabaseName（在 CLI 中：`--database-name`或在 Python 中：`database_name`）-UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。

- DBClusterArn（在 CLI 中：`--db-cluster-arn`或在 Python 中：db_cluster_arn）-UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。

- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：db_cluster_identifier）-UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。

- DBClusterMembers（在 CLI 中：`--db-cluster-members`或在 Python 中：db_cluster_members）-數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

提供組成資料庫叢集的執行個體清單。

- DBClusterParameterGroup（在 CLI 中：`--db-cluster-parameter-group`或在 Python 中：`db_cluster_parameter_group`）-UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。

- DbClusterResourceId（在 CLI 中：`--db-cluster-resource-id`或在 Python 中：`db_cluster_resource_id`）-UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫叢集 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。

- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。

- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。

- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。

- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。

- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。

- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。

- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼是可識別全域資料庫資料庫資料庫的唯一金鑰。

- HostedZoneId（在 CLI 中：`--hosted-zone-id`或在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。

- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId（在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`）-UTF-8 字符串。

若 StorageEncrypted 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。

- LatestRestorableTime（在 CLI 中：--latest-restorable-time，或在 Python 中：latest_restorable_time）-時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。

- MultiAZ（在 CLI 中：--multi-az，或者在 Python 中：multi_az）-布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues（在 CLI 中：--pending-modified-values或在 Python 中：pending_modified_values）-一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在動ModifyDBCluster作中會用來做為回應元素，並且包含將在下一個維護時段中會用來做為回應元素。

- PercentProgress（在 CLI 中：--percent-progress或在 Python 中：percent_progress）-UTF-8 字符串。

  指定操作的進度 (百分比)。

- Port（在 CLI 中：--port，或在 Python 中：port）-數字（整數）。

  指定資料庫引擎接聽的連接埠。

- PreferredBackupWindow（在 CLI 中：--preferred-backup-window或在 Python 中：preferred_backup_window）-UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 BackupRetentionPeriod 決定)，則自動化備份會在此期間建立。

- PreferredMaintenanceWindow（在 CLI 中：--preferred-maintenance-window或在 Python 中：preferred_maintenance_window）-UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。

- ReaderEndpoint（在 CLI 中：--reader-endpoint或在 Python 中：reader_endpoint）-UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。

- ReadReplicaIdentifiers（在 CLI 中：--read-replica-identifiers或者在 Python 中：read_replica_identifiers）-一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。

- ReplicationSourceIdentifier（在 CLI 中：--replication-source-identifier或在 Python 中：replication_source_identifier）-UTF-8 字符串。

  Neptune 不提供支援。

- ReplicationType（在 CLI 中：--replication-type或在 Python 中：replication_type）-UTF-8 字符串。

  Neptune 不提供支援。

- ServerlessV2ScalingConfiguration（在 CLI 中：--serverless-v2-scaling-configuration或在 Python 中：serverless_v2_scaling_configuration）-一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。

- Status (在 CLI 中：`--status`或在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)
- 无效ClusterStateFault (p. 1170)

DescribeDBClusters (動作)

這個 API 的亞馬遜 CLI 名稱是：`describe-db-clusters`，Python 的名稱是：`describe_db_clusters`。

傳回所佈建之資料庫叢集的相關資訊，並支援分頁。

> **Note**
>
> 此操作也可以傳回 Amazon RDS 叢集和 Amazon DocDB 叢集的資訊。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  使用者提供的資料庫叢集識別符。若指定此參數，則只會傳回特定資料庫叢集的資訊。此參數沒有大小寫之分。

  約束:
  - 如果提供，則必須與現有的數據庫匹配ClusterIdentifier。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  篩選條件，指定要描述的一或多個資料庫叢集。

  支援的篩選條件：
  - `db-cluster-id` - 接受資料庫叢集識別符及資料庫叢集 Amazon Resource Name (ARN)。結果清單只會包含由這些 ARN 識別的資料庫叢集相關資訊。
  - `engine` -接受引擎名稱 (例如 `neptune`)，並將結果清單限制為由該引擎建立的資料庫叢集。

  例如，若要從 Amazon CLI 呼叫此 API 和篩選器，以便僅傳回 Neptune 資料庫叢集，您可以使用下列命令：

  Example

```
aws neptune describe-db-clusters \
```

```
            --filters  Name=engine,Values=neptune
```

- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個 the section called “DescribeDBClusters” (p. 1040) 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`，或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

- DBClusters (在 CLI 中：`--db-clusters`，或在 Python 中：`db_clusters`) -數DBCluster (p. 1001)據庫集群對象的數組。

  包含使用者的資料庫叢集清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  可用於後續 DescribeDBClusters 請求的分頁字符。

錯誤

- 資料庫ClusterNotFoundFault (p. 1162)

Neptune 全球數據庫 API

結構：

- GlobalCluster （結構） (p. 1041)

動作:

- CreateGlobalCluster （動作） (p. 1042)
- DeleteGlobalCluster （動作） (p. 1044)
- ModifyGlobalCluster （動作） (p. 1045)
- DescribeGlobalClusters （動作） (p. 1047)
- FailoverGlobalCluster （動作） (p. 1048)
- RemoveFromGlobalCluster （動作） (p. 1050)

GlobalCluster （結構）

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

欄位

- DeletionProtection – 布林值。

  全域資料庫的刪除保護設定。
- Engine – UTF-8 字串。

  全域資料庫所使用的 Neptune 資料庫引擎 (`"neptune"`)。
- EngineVersion – UTF-8 字串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn – UTF-8 字串。

  全域資料庫的 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier— 符合規則運算式的 UTF-8 字串，長度不可小於 1 個位元組，也不可超過 255 個位元組，且需符合規則運算式：'[0-9A-Za-z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers— — 一個結構的 GlobalClusterMember陣列.

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId – UTF-8 字串。

  全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。
- Status – UTF-8 字串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted – 布林值。

  全域資料庫的儲存加密設定。

`GlobalCluster` 會用來做為以下項目的回應元素：

- CreateGlobalCluster (p. 1042)
- ModifyGlobalCluster (p. 1045)
- DeleteGlobalCluster (p. 1044)
- RemoveFromGlobalCluster (p. 1050)
- FailoverGlobalCluster (p. 1048)

CreateGlobalCluster （動作）

這個 API 的亞馬遜 CLI 名稱是：`create-global-cluster`，Python 的名稱是：`create_global_cluster`。

建立跨多個 Amazon 區域的 Neptune 全域資料庫。全域資料庫包含具有讀寫功能的單一主要叢集，可透過 Neptune e儲存子系統執行的高速複寫接收來自主要叢集的資料。

您可以建立一開始為空的全域資料庫，然後將主要叢集和次要叢集新增至該資料庫，然後將主要叢集和次要叢集新增至該資料庫，或將現有的 Neptune 叢集指定為全域資料庫的主要叢集。

請求

- DatabaseName (在 CLI 中：`--database-name`或者在 Python 中：`database_name`) -UTF-8 字符串。

  新全域資料庫的名稱 (最多 64 個英數字元。
- DeletionProtection (在 CLI 中：`--deletion-protection`或在 Python 中：`deletion_protection`) -布爾值。

  新的全域資料庫的刪除保護設定。啟用刪除保護時無法刪除全域資料庫。
- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  要在全域資料庫中使用的資料庫引擎名稱。

  有效值：`neptune`
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  全域資料庫所使用的 Neptune 引擎版本。

  有效值：`1.2.0.0`或以上。
- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) — 必需：UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  新全域資料庫叢集的叢集識別碼。
- SourceDBClusterIdentifier (在 CLI 中：`--source-db-cluster-identifier`或者在 Python 中：`source_db_cluster_identifier`) -UTF-8 字符串。

  (選用作新全域資料庫主要叢集使用作新全域資料庫主要叢集的 Amazon Resource Name (ARN)。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`或在 Python 中：`storage_encrypted`) -布爾值。

  新全域資料庫叢集的儲存加密設定。

回應

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

- DeletionProtection (在 CLI 中：`--deletion-protection`或在 Python 中：`deletion_protection`) -布爾值。

  全域資料庫的刪除保護設定。
- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  全域資料庫所使用的 Neptune 資料庫引擎 ("neptune")。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn (在 CLI 中：`--global-cluster-arn`或者在 Python 中：`global_cluster_arn`) -UTF-8 字符串。

  全域資料庫的 Amazon Resource Name (ARN)。

- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers（在 CLI 中：`--global-cluster-members`或者在 Python 中：`global_cluster_members`）-一個數組 GlobalClusterMember-一個結構。

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId（在 CLI 中：`--global-cluster-resource-id`或者在 Python 中：`global_cluster_resource_id`）-UTF-8 字符串。

  全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。
- Status（在 CLI 中：`--status`或者在 Python 中：`status`）-UTF-8 字符串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted（在 CLI 中：`--storage-encrypted`或在 Python 中：`storage_encrypted`）-布爾值。

  全域資料庫的儲存加密設定。

錯誤

- GlobalClusterAlreadyExistsFault (p. 1168)
- GlobalClusterQuotaExceededFault (p. 1168)
- 无效ClusterStateFault (p. 1170)
- 資料庫ClusterNotFoundFault (p. 1162)

DeleteGlobalCluster（動作）

這個 API 的亞馬遜 CLI 名稱是：`delete-global-cluster`，Python 的名稱是：`delete_global_cluster`。

刪除全域資料庫。必須先卸離或刪除主要叢集和所有次要叢集。

請求

- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）— 必需：UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  要刪除的全域資料庫叢集的叢集識別碼。

回應

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

- DeletionProtection（在 CLI 中：`--deletion-protection`或在 Python 中：`deletion_protection`）-布爾值。

全域資料庫的刪除保護設定。

- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  全域資料庫所使用的 Neptune 資料庫引擎 ("`neptune`")。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn (在 CLI 中：`--global-cluster-arn`或者在 Python 中：`global_cluster_arn`) -UTF-8 字符串。

  全域資料庫的 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) -UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers (在 CLI 中：`--global-cluster-members`或者在 Python 中：`global_cluster_members`) -一個數組 GlobalClusterMember-一個結構。

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId (在 CLI 中：`--global-cluster-resource-id`或者在 Python 中：`global_cluster_resource_id`) -UTF-8 字符串。

  全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`或在 Python 中：`storage_encrypted`) -布爾值。

  全域資料庫的儲存加密設定。

錯誤

- GlobalClusterNotFoundFault (p. 1168)
- InvalidGlobalClusterStateFault (p. 1172)

ModifyGlobalCluster （動作）

這個 API 的亞馬遜 CLI 名稱是：`modify-global-cluster`，Python 的名稱是：`modify_global_cluster`。

修改 Amazon Neptune e全域叢集的設定。您可以在要求中指定這些參數及其新值，以變更一或多個資料庫組態參數。

請求

- AllowMajorVersionUpgrade (在 CLI 中：`--allow-major-version-upgrade`或在 Python 中：`allow_major_version_upgrade`) -布爾值。

  指示是否允許升級主要版本的值。

限制：將EngineVersion參數指定為與資料庫叢集群組目前版本不同的主要版本時，必須允許主要版本升級。

如果您升級全域資料庫的主要版本，叢集和資料庫執行個體參數群組會設定為新版本的預設參數群組，因此您必須在完成升級後套用任何自訂參數群組。

- DeletionProtection（在 CLI 中：--deletion-protection或在 Python 中：deletion_protection）-布爾值。

  指示全域資料庫是否已啟用刪除保護。啟用刪除保護時無法刪除全域資料庫。

- EngineVersion（在 CLI 中：--engine-version或者在 Python 中：engine_version）-UTF-8 字符串。

  您要升級的資料庫引擎版本號碼。變更此參數將導致中斷。除非啟用，否則會在下一次維護時段期間由系統套ApplyImmediately用變更。

  若要列出所有可用的 Neptune e引擎版本清單，請使用下列命令：

  Example

```
aws neptune describe-db-engine-versions \
          --engine neptune \
          --query '*[]|[?SupportsGlobalDatabases == 'true'].[EngineVersion]'
```

- GlobalClusterIdentifier（在 CLI 中：--global-cluster-identifier或在 Python 中：global_cluster_identifier）— 必需：UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  要修改之全域叢集的資料庫叢集識別碼。此參數不區分大小寫。

  限制：必須符合現有全域資料庫叢集的識別碼。

- NewGlobalClusterIdentifier（在 CLI 中：--new-global-cluster-identifier或在 Python 中：new_global_cluster_identifier）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  要指派給全域資料庫的新叢集識別碼。此值會以小寫字母字串的形式儲存。

  約束:

  - 必須包含 1 到 63 個字母、數字或連字號。
  - 第一個字元必須是字母。
  - 不能以一個連字號結尾或是連續包含兩個連字號

  範例：my-cluster2

回應

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

- DeletionProtection（在 CLI 中：--deletion-protection或在 Python 中：deletion_protection）-布爾值。

  全域資料庫的刪除保護設定。

- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  全域資料庫所使用的 Neptune 資料庫引擎 ("`neptune`")。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn (在 CLI 中：`--global-cluster-arn`或者在 Python 中：`global_cluster_arn`) -UTF-8 字符串。

  全域資料庫的 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) -UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers (在 CLI 中：`--global-cluster-members`或者在 Python 中：`global_cluster_members`) -一個數組 GlobalClusterMember-一個結構。

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId (在 CLI 中：`--global-cluster-resource-id`或者在 Python 中：`global_cluster_resource_id`) -UTF-8 字符串。

  全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`或在 Python 中：`storage_encrypted`) -布爾值。

  全域資料庫的儲存加密設定。

錯誤

- GlobalClusterNotFoundFault (p. 1168)
- InvalidGlobalClusterStateFault (p. 1172)

DescribeGlobalClusters （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-global-clusters`，Python 的名稱是：`describe_global_clusters`。

傳回有關 Neptune 全域資料庫叢集的資訊 此 API 支援分頁。

請求

- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) -UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  使用者提供的資料庫叢集識別符。如果指定此參數，則只會傳回有關指定資料庫叢集的資訊。此參數不區分大小寫。

  限制：如果提供，則必須與現有的資料庫叢集識別碼相符。

- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  (選用) 上一個呼叫傳回的分頁字符`DescribeGlobalClusters`。如果指定此參數，則回應只會包含標記之外的記錄，最多可達由指定的數目為止`MaxRecords`。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。如果存在的記錄多於指定`MaxRecords`值，則回應中會包含分頁標記 Token，您可以用來擷取剩餘結果。

  預設：`100`

  限制條件：最小 20，最大 100。

回應

- GlobalClusters (在 CLI 中：`--global-clusters`，或在 Python 中：`global_clusters`) -GlobalCluster (p. 1041) GlobalClusterMember 對象的數組。

  此要求傳回的全域叢集和執行個體清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  一個分頁令牌。如果在響應中返回此參數，則可以使用更多記錄，這些記錄可以通過一個或多個額外的調用來檢索`DescribeGlobalClusters`。

錯誤

- GlobalClusterNotFoundFault (p. 1168)

FailoverGlobalCluster （動作）

這個 API 的亞馬遜 CLI 名稱是：`failover-global-cluster`，Python 的名稱是：`failover_global_cluster`。

啟動 Neptune e全域資料庫的容錯移轉程序。

Neptune 全域資料庫的容錯移轉會將其中一個次要唯讀資料庫叢集提升為主要資料庫叢集，並將主要資料庫叢集降級為次要 (唯讀) 資料庫叢集。換句話說，會切換目前主要資料庫叢集和所選目標次要資料庫叢集的角色。選取的次要資料庫叢集會採用 Neptune 全域資料庫的完整讀取/寫入功能

> **Note**
>
> 此動作僅適用於 Neptune 全域資料庫。此動作僅適用於具有健全 Neptune DB 叢集且無區域中斷的健全 Neptune 全域資料庫，以測試災難復原案例或重新設定全域資料庫拓撲。

請求

- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) — 必需：UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  應容錯移轉的 Neptune 全域資料庫識別碼。識別碼是使用者在建立 Neptune 全域資料庫時指派的唯一金鑰。換句話說，它是您要容錯移轉的全域資料庫名稱。

  限制：必須符合現有 Neptune 全域資料庫的識別碼。
- TargetDbClusterIdentifier (在 CLI 中：`--target-db-cluster-identifier`或在 Python 中：`target_db_cluster_identifier`) -必需：UTF-8 字符串。

您要提升為全域資料庫之主要資料庫叢集的 Amazon Resource Name (ARN)。

回應

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  全域資料庫的刪除保護設定。
- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  全域資料庫所使用的 Neptune 資料庫引擎（"neptune"）。
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn（在 CLI 中：`--global-cluster-arn`或者在 Python 中：`global_cluster_arn`）-UTF-8 字符串。

  全域資料庫的 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers（在 CLI 中：`--global-cluster-members`或者在 Python 中：`global_cluster_members`）-一個數組 GlobalClusterMember-一個結構。

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId（在 CLI 中：`--global-cluster-resource-id`或者在 Python 中：`global_cluster_resource_id`）-UTF-8 字符串。

  全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。
- Status（在 CLI 中：`--status`或者在 Python 中：`status`）-UTF-8 字符串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted（在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`）-布爾值。

  全域資料庫的儲存加密設定。

錯誤

- GlobalClusterNotFoundFault (p. 1168)
- InvalidGlobalClusterStateFault (p. 1172)
- 无效ClusterStateFault (p. 1170)
- 資料庫ClusterNotFoundFault (p. 1162)

RemoveFromGlobalCluster （動作）

這個 API 的亞馬遜 CLI 名稱是：`remove-from-global-cluster`，Python 的名稱是：`remove_from_global_cluster`。

從 Neptune 全域資料庫卸離 Neptune 資料庫叢集。次要叢集會變成具有讀寫功能而非唯讀功能的一般獨立叢集，且不再從主要叢集接收資料。

請求

- DbClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`）-必需：UTF-8 字符串。

  可識別要從 Neptune epe全域資料庫叢集中分離的叢集 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）— 必需：UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  要從中斷連結指定之 Neptune 資料庫叢集之 Neptune 全域資料庫的識別碼。

回應

包含 Amazon Neptune 全球資料庫的詳細資料。

此資料類型用作the section called “CreateGlobalCluster” (p. 1042)、、、the section called “DescribeGlobalClusters” (p. 1047)the section called “ModifyGlobalCluster” (p. 1045)the section called “DeleteGlobalCluster” (p. 1044)、the section called “FailoverGlobalCluster” (p. 1048)和the section called “RemoveFromGlobalCluster” (p. 1050)動作的回應元素。

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  全域資料庫的刪除保護設定。
- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  全域資料庫所使用的 Neptune 資料庫引擎（"neptune"）。
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  全域資料庫使用的 Neptune 引擎版本。
- GlobalClusterArn（在 CLI 中：`--global-cluster-arn`或者在 Python 中：`global_cluster_arn`）-UTF-8 字符串。

  全域資料庫的 Amazon Resource Name (ARN)。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別資料庫的唯一金鑰。
- GlobalClusterMembers（在 CLI 中：`--global-cluster-members`或者在 Python 中：`global_cluster_members`）-一個數組 GlobalClusterMember-一個結構。

  屬於全域資料庫一部分之所有資料庫叢集的叢集 ARN 和執行個體 ARN 清單。
- GlobalClusterResourceId（在 CLI 中：`--global-cluster-resource-id`或者在 Python 中：`global_cluster_resource_id`）-UTF-8 字符串。

全域資料庫的不可變識別碼，在所有區域中都是唯一的。此識別碼可在每次存取資料庫叢集的 KMS 金鑰時，於 CloudTrail 日誌項目中找到。

- Status（在 CLI 中：`--status`或者在 Python 中：`status`）-UTF-8 字符串。

  指定此全域資料庫的目前狀態。
- StorageEncrypted（在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`）-布爾值。

  全域資料庫的儲存加密設定。

錯誤

- GlobalClusterNotFoundFault (p. 1168)
- InvalidGlobalClusterStateFault (p. 1172)
- 資料庫ClusterNotFoundFault (p. 1162)

Neptune 執行個體 API

結構：

- DBInstance (結構) (p. 1051)
- 資料庫InstanceStatusInfo (結構) (p. 1054)
- 可訂購數據庫InstanceOption （結構） (p. 1055)
- 驗證碼InstanceModificationsMessage (結構) (p. 1056)

動作:

- CreateDBInstance (動作) (p. 1056)
- DeleteDBInstance (動作) (p. 1065)
- ModifyDBInstance (動作) (p. 1069)
- RebootDBInstance (動作) (p. 1078)
- DescribeDBInstances (動作) (p. 1082)
- DescribeOrderable資料庫InstanceOptions (動作) (p. 1083)
- DescribeValid資料庫InstanceModifications (動作) (p. 1084)

DBInstance (結構)

包含 Amazon Neptune 資料庫執行個體的詳細資訊。

此資料類型在 the section called “DescribeDBInstances” (p. 1082) 動作中會用來作為回應元素。

欄位

- AutoMinorVersionUpgrade – 布林值。

  指示會自動套用次要版本的修補程式。
- AvailabilityZone – UTF-8 字串。

  指定資料庫執行個體所在的可用區域名稱。

- BackupRetentionPeriod – 數字 (整數)。

  指定保留自動資料庫快照的天數。
- CACertificateIdentifier – UTF-8 字串。

  此資料庫執行個體的憑證授權機構憑證識別碼。
- CopyTagsToSnapshot – 布林值。

  指定是否要從資料庫執行個體將標籤複製到資料庫執行個體的快照。
- DBClusterIdentifier – UTF-8 字串。

  若資料庫執行個體是資料庫叢集的成員，則包含資料庫執行個體所屬資料庫叢集的名稱。
- DBInstanceArn – UTF-8 字串。

  資料庫執行個體的 Amazon Resource Name (ARN)。
- DBInstanceClass – UTF-8 字串。

  包含資料庫執行個體之運算和記憶體容量類別的名稱。
- DBInstanceIdentifier – UTF-8 字串。

  包含使用者提供的資料庫識別碼。此識別碼是可識別資料庫執行個體的唯一金鑰。
- DbInstancePort – 數字 (整數)。

  指定資料庫執行個體接聽的連接埠。若資料庫執行個體是資料庫叢集的一部分，這可以是與資料庫叢集連接埠不同的連接埠。
- DBInstanceStatus – UTF-8 字串。

  指定此資料庫的目前狀態。
- DbiResourceId – UTF-8 字串。

  資料庫執行個體的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫執行個體的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。
- DBName – UTF-8 字串。

  資料庫名稱。
- DBParameterGroups—資料庫ParameterGroupStatus (p. 1087) DBParameterGroup 物件的陣列。

  提供要套用到此資料庫執行個體的資料庫參數群組清單。
- DBSecurityGroups—資料庫 SecurityGroupMembership (p. 1156) DBSecurityGroup 物件的陣列。

  提供資料庫安全群組元素的清單，元素中只包含 `DBSecurityGroup.Name` 和 `DBSecurityGroup.Status` 子元素。
- DBSubnetGroup – 資料庫SubnetGroup (p. 1103) 物件。

  指定與資料庫執行個體相關聯子網路群組上的資訊，包含名稱、描述，以及子網路群組中的子網路。
- DeletionProtection – 布林值。

  指出資料庫執行個體是否已啟用刪除保護。如果刪除保護已經啟用，執行個體將無法刪除。請參閱刪除資料庫執行個體。
- DomainMemberships – 一個 DomainMembership (p. 1156) DomainMembership 物件陣列。

  不支援
- EnabledCloudwatchLogsExports – UTF-8 字串陣列。

  此資料庫執行個體設定為匯出至記錄的 CloudWatch記錄類型清單。

- Endpoint – 端點 (p. 1157) 物件。

  指定連線端點。
- Engine – UTF-8 字串。

  提供用於此資料庫執行個體的資料庫引擎名稱。
- EngineVersion – UTF-8 字串。

  指示資料庫引擎版本。
- EnhancedMonitoringResourceArn – UTF-8 字串。

  接收資料庫執行個體的 Amazon CloudWatch Resource Name (ARN)。
- IAMDatabaseAuthenticationEnabled – 布林值。

  若有啟用 Amazon Identity and Access Management (IAM) 驗證，則為 True，否則為 false。
- InstanceCreateTime – 時間戳記。

  提供建立資料庫執行個體的日期和時間。
- Iops – 數字 (整數)。

  指定佈建 IOPS (每秒 I/O 操作) 值。
- KmsKeyId – UTF-8 字串。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- LatestRestorableTime – 時間戳記。

  指定可以透過還原來 point-in-time還原資料庫的最近時間。
- LicenseModel – UTF-8 字串。

  此資料庫執行個體的授權模型資訊。
- MonitoringInterval – 數字 (整數)。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。
- MonitoringRoleArn – UTF-8 字串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。
- MultiAZ – 布林值。

  指定資料庫執行個體是否為異地同步備份部署。
- PendingModifiedValues – PendingModifiedValues (p. 1149) 物件。

  指定對資料庫執行個體所進行的變更仍在擱置中。此元素只有在變更擱置中時才會包含在其中。特定的變更會由子元素識別。
- PreferredBackupWindow – UTF-8 字串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow – UTF-8 字串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- PromotionTier – 數字 (整數)。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。
- PubliclyAccessible – 布林值。

此標記不應繼續使用。

- ReadReplicaDBClusterIdentifiers-數 ReadReplica據庫的數組ClusterIdentifier-UTF-8 字符串。

  包含一或多個資料庫叢集的識別符，這些叢集是此資料庫執行個體的僅供讀取複本。
- ReadReplicaDBInstanceIdentifiers-數 ReadReplica據庫的數組InstanceIdentifier-UTF-8 字符串。

  包含一或多個與此資料庫執行個體相關聯僅供讀取複本的識別符。
- ReadReplicaSourceDBInstanceIdentifier – UTF-8 字串。

  包含來源資料庫執行個體的識別符 (若此資料庫執行個體是僅供讀取複本)。
- SecondaryAvailabilityZone – UTF-8 字串。

  若存在，指定具備多個可用區支援之資料庫執行個體的次要可用區名稱。
- StatusInfos—資料庫InstanceStatusInfo (p. 1054) DBInstanceStatusInfo 物件的陣列。

  僅供讀取複本的狀態。若執行個體並非僅供讀取複本，此處即為空白。
- StorageEncrypted – 布林值。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- StorageType – UTF-8 字串。

  指定與此資料庫執行個體相關聯的儲存體類型。
- TdeCredentialArn – UTF-8 字串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。
- Timezone – UTF-8 字串。

  不支援.
- VpcSecurityGroups – 一個 VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 物件陣列。

  提供資料庫執行個體所屬 VPC 安全群組元素的清單。

`DBInstance` 會用來做為以下項目的回應元素：

- CreateDBInstance (p. 1056)
- DeleteDBInstance (p. 1065)
- ModifyDBInstance (p. 1069)
- RebootDBInstance (p. 1078)

資料庫InstanceStatusInfo (結構)

提供資料庫執行個體的狀態資訊清單。

欄位

- Message – UTF-8 字串。

  若執行個體發生錯誤，則為錯誤的詳細資訊。若執行個體並未處於錯誤狀態，則此值為空白。
- Normal – 布林值。

  布林值，若執行個體正常運作，則為 True，若執行個體處於錯誤狀態，則為 False。
- Status – UTF-8 字串。

資料庫執行個體的狀態。對於 StatusType 僅供讀取複本，這些值可以是複寫、錯誤、停止或終止。

- StatusType – UTF-8 字串。

  此值目前是「讀取複寫」。

可訂購數據庫InstanceOption （結構）

包含資料庫執行個體的可用選項清單。

此資料類型在 the section called “DescribeOrderable資料庫InstanceOptions” (p. 1083) 動作中會用來作為回應元素。

欄位

- AvailabilityZones – 一個 AvailabilityZone (p. 1156) AvailabilityZone 物件陣列。

  資料庫執行個體的可用區域清單。
- DBInstanceClass – UTF-8 字串。

  資料庫執行個體的資料庫執行個體類別。
- Engine – UTF-8 字串。

  資料庫執行個體的引擎類型。
- EngineVersion – UTF-8 字串。

  資料庫執行個體的引擎版本。
- LicenseModel – UTF-8 字串。

  資料庫執行個體的授權模型。
- MaxIopsPerDbInstance – 數字 (整數)。

  資料庫執行個體的總佈建 IOPS 上限。
- MaxIopsPerGib – 數字 (雙位數)。

  資料庫執行個體的每 GiB 佈建 IOPS 上限。
- MaxStorageSize – 數字 (整數)。

  資料庫執行個體的儲存體大小上限。
- MinIopsPerDbInstance – 數字 (整數)。

  資料庫執行個體的總佈建 IOPS 下限。
- MinIopsPerGib – 數字 (雙位數)。

  資料庫執行個體的每 GiB 佈建 IOPS 下限。
- MinStorageSize – 數字 (整數)。

  資料庫執行個體的儲存體大小下限。
- MultiAZCapable – 布林值。

  指出資料庫執行個體是否具備多個可用區功能。
- ReadReplicaCapable – 布林值。

  指出資料庫執行個體是否可以擁有僅供讀取複本。
- StorageType – UTF-8 字串。

指出資料庫執行個體的儲存體類型。

- SupportsEnhancedMonitoring – 布林值。

  指出資料庫執行個體是否支援增強型監控，其間隔介於 1 到 60 秒。
- SupportsGlobalDatabases – 布林值。

  指出您是否可以將 Neptune 全域資料庫與其他資料庫引擎屬性的特定組合搭配使用的值。
- SupportsIAMDatabaseAuthentication – 布林值。

  指出資料庫執行個體是否支援 IAM 資料庫身分驗證。
- SupportsIops – 布林值。

  指出資料庫執行個體是否支援佈建 IOPS。
- SupportsStorageEncryption – 布林值。

  指出資料庫執行個體是否支援加密儲存體。
- Vpc – 布林值。

  指出資料庫執行個體是否位於 VPC 中。

驗證碼InstanceModificationsMessage (結構)

您可以對資料庫執行個體進行的有效修改相關資訊。包含成功呼叫 the section called “DescribeValid資料庫InstanceModifications” (p. 1084) 動作的結果。您可以在呼叫 the section called “ModifyDBInstance” (p. 1069) 時使用此資訊。

欄位

- Storage— — 一個結構的 ValidStorageOptions陣列.

  您資料庫執行個體的有效儲存體選項。

ValidDBInstanceModificationsMessage 會用來做為以下項目的回應元素：

- DescribeValid資料庫InstanceModifications (p. 1084)

CreateDBInstance (動作)

這個 API 的亞馬遜 CLI 名稱是：create-db-instance，Python 的名稱是：create_db_instance。

建立新的資料庫執行個體。

請求

- AutoMinorVersionUpgrade（在 CLI 中：--auto-minor-version-upgrade，或者在 Python 中：auto_minor_version_upgrade）-布爾值。

  指出會在維護時段期間將次要引擎升級自動套用至資料庫執行個體。

  預設：true
- AvailabilityZone（在 CLI 中：--availability-zone或者在 Python 中：availability_zone）- UTF-8 字符串。

在其中建立資料庫執行個體的 EC2 可用區域。

預設：端點 Amazon 區域中系統隨機選擇的可用區域。

範例：`us-east-1d`

限制條件：若 MultiAZ AvailabilityZone 參數設定為，則無法指定參數`true`。指定的可用區域和目前端點必須位於同一 Amazon 區域內。

- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  自動備份保留的天數。

  不適用。自動化備份的保留期會由資料庫叢集管理。如需詳細資訊，請參閱the section called “CreateDBCluster” (p. 1005)。

  預設：1

  約束:

  - 該值必須介於 0 到 35 之間
  - 若資料庫執行個體是僅供讀取複本的來源，則無法設為 0
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  若為 True，則會從資料庫執行個體將所有標籤複製到資料庫執行個體的快照，否則為 False。預設值為 false。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  執行個體所屬資料庫叢集的識別碼。

  如需建立資料庫叢集的詳細資訊，請參閱the section called “CreateDBCluster” (p. 1005)。

  類型：字串
- DBInstanceClass (在 CLI 中：`--db-instance-class`或在 Python 中：`db_instance_class`) -必需：UTF-8 字符串。

  資料庫執行個體的運算與記憶體容量，例如 `db.m4.large`。並非所有資料庫執行個體類別碼都適用於所有 Amazon 區域內。
- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`) -必需：UTF-8 字符串。

  資料庫執行個體識別符。此參數是以小寫字母字串的形式儲存。

  約束:

  - 必須包含 1 到 63 個字母、數字或連字號。
  - 第一個字元必須是字母。
  - 不能以連字號結尾或連續包含兩個連字號。

  範例：`mydbinstance`
- DBName (在 CLI 中：`--db-name`或者在 Python 中：`db_name`) -UTF-8 字符串。

  不支援.
- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或者在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

要與此資料庫執行個體建立關聯的資料庫參數群組名稱。若忽略此引數，則會使用指定引擎的預設資料庫 ParameterGroup 。

約束:

- 必須為 1 到 255 個字母、數字或連字號。
- 第一個字元必須是字母
- 不能以連字號結尾或連續包含兩個連字號

- DBSecurityGroups (在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`) - 數據庫數組SecurityGroupName-UTF-8 字符串。

  與此資料庫執行個體關聯的資料庫安全群組清單。

  預設值：資料庫引擎的預設資料庫安全群組。
- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`) -UTF-8 字符串。

  要與此資料庫執行個體建立關聯的資料庫子網路群組。

  如果沒有資料庫子網路群組，則它是非VPC DB執行個體。
- DeletionProtection (在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`) -布爾值。

  指示資料庫執行個體是否啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是停用的。請參閱刪除資料庫執行個體。

  即使刪除保護已在上層資料庫叢集中啟用，資料庫叢集中的資料庫執行個體仍可刪除。
- Domain (在 CLI 中：`--domain`或者在 Python 中：`domain`) -UTF-8 字符串。

  指定要在其中建立執行個體的 Active Directory 網域。
- DomainIAMRoleName (在 CLI 中：`--domain-iam-role-name`或者在 Python 中：`domain_iam_role_name`) -UTF-8 字符串。

  指定對 Directory Service 發出 API 呼叫時要使用的 IAM 角色名稱。
- EnableCloudwatchLogsExports (在 CLI 中：`--enable-cloudwatch-logs-exports`或在 Python 中：`enable_cloudwatch_logs_exports`) -UTF-8 字符串的數組。

  需要啟用以匯出至 Logs 的 CloudWatch日誌類型清單。
- EnableIAMDatabaseAuthentication (在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`) -布爾值。

  Neptune 不支援 (忽略)。
- Engine (在 CLI 中：`--engine`或在 Python 中：`engine`) -必需：UTF-8 字符串。

  要為此執行個體使用的資料庫引擎名稱。

  有效值: `neptune`
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  要使用的資料庫引擎版本號碼。目前，設定此參數沒有任何作用。
- Iops (在 CLI 中：`--iops`，或在 Python 中：`iops`) -數字 (整數) 。

  初始配置給資料庫執行個體的佈建IOPS數量 (每秒輸入/輸出作業) 。
- KmsKeyId (在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`) -UTF-8 字符串。

加密資料庫執行個體的 Amazon KMS 金鑰識別符。

KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。若您使用擁有用來加密金鑰的相同 Amazon 帳戶建立資料庫執行個體，您可以改為使用 KMS 金鑰別名，而非 KM 加密金鑰的 ARN。

不適用。KMS 金鑰識別碼會由資料庫叢集管理。如需詳細資訊，請參閱the section called "CreateDBCluster" (p. 1005)。

若 `StorageEncrypted` 參數為 True，並且您沒有為 `KmsKeyId` 參數指定值，則 Amazon Neptune 會使用您的預設加密金鑰。Amazon KMS 會為您的 Amazon 帳戶建立預設加密金鑰。您的 Amazon 帳戶在每個 Amazon 區域各有不同的預設加密金鑰。

- LicenseModel（在 CLI 中：`--license-model`或者在 Python 中：`license_model`）-UTF-8 字符串。

  此資料庫執行個體的授權模型資訊。

  有效值：`license-included` | `bring-your-own-license` | `general-public-license`
- MonitoringInterval（在 CLI 中：`--monitoring-interval`，或在 Python 中：`monitoring_interval`）-數字（整數）。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。若要停用收集增強型監控指標，請指定 0。預設值為 0。

  若指定 `MonitoringRoleArn`，您必須也將 `MonitoringInterval` 設為 0 之外的值。

  有效值: `0, 1, 5, 10, 15, 30, 60`
- MonitoringRoleArn（在 CLI 中：`--monitoring-role-arn`或者在 Python 中：`monitoring_role_arn`）-UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。例如：`arn:aws:iam:123456789012:role/emaccess`。

  若 `MonitoringInterval` 設為 0 之外的值，則您必須提供 `MonitoringRoleArn` 值。
- MultiAZ（在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`）-布爾值。

  指定資料庫執行個體是否為異地同步備份部署。若 MultiAZ AvailabilityZone 參數設定為 true，則您無法設定參數。
- Port（在 CLI 中：`--port`，或在 Python 中：`port`）-數字（整數）。

  資料庫接受連線的連接埠號碼。

  不適用。連接埠會由資料庫叢集管理。如需詳細資訊，請參閱the section called "CreateDBCluster" (p. 1005)。

  預設：8182

  類型：整數
- PreferredBackupWindow（在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`）-UTF-8 字符串。

  每日時間範圍，自動化備份會在這段期間建立。

  不適用。建立自動化備份的每日時間範圍會由資料庫叢集管理。如需詳細資訊，請參閱the section called "CreateDBCluster" (p. 1005)。
- PreferredMaintenanceWindow（在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`）-UTF-8 字符串。

  每週可能進行系統維護的時間範圍，以國際標準時間 (UTC) 表示。

格式：ddd:hh24:mi-ddd:hh24:mi

預設是從各 Amazon 區域 8 小時時段中隨機選取的 30 分鐘時段，並隨機發生在一週內的某一天。

有效日：星期一、星期二、星期三、星期四、星期五、星期六、星期日。

限制條件：必須至少是 30 分鐘的時段。

- PromotionTier（在 CLI 中：--promotion-tier，或在 Python 中：promotion_tier）-數字（整數）。

  指定在現有主要執行個體失敗後，將僅供讀取複本提升為主要執行個體順序的值。

  預設：1

  有效值：0 到 15

- PubliclyAccessible（在 CLI 中：--publicly-accessible，或者在 Python 中：publicly_accessible）-布爾值。

  此標記不應繼續使用。

- StorageEncrypted（在 CLI 中：--storage-encrypted，或者在 Python 中：storage_encrypted）-布爾值。

  指定是否加密資料庫執行個體。

  不適用。資料庫執行個體的加密會由資料庫叢集管理。如需詳細資訊，請參閱the section called “CreateDBCluster” (p. 1005)。

  預設：false

- StorageType（在 CLI 中：--storage-type或者在 Python 中：storage_type）-UTF-8 字符串。

  指定要和資料庫執行個體建立關聯的儲存體類型。

  不適用。儲存體會由資料庫叢集管理。

- Tags（在 CLI 中：--tags或在 Python 中：tags）-Tag (p. 1150) 標籤對象的數組。

  要指派給新執行個體的標籤。

- TdeCredentialArn（在 CLI 中：--tde-credential-arn或者在 Python 中：tde_credential_arn）-UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。

- TdeCredentialPassword（在 CLI 中：--tde-credential-password或者在 Python 中：tde_credential_password）-UTF-8 字符串。

  在金鑰存放區指定 ARN 的密碼，用於存取裝置。

- Timezone（在 CLI 中：--timezone或者在 Python 中：timezone）-UTF-8 字符串。

  資料庫執行個體的時區。

- VpcSecurityGroupIds（在 CLI 中：--vpc-security-group-ids或在 Python 中：vpc_security_group_ids）-一個數組 VpcSecurityGroupId-UTF-8 字符串。

  與此資料庫執行個體關聯的 EC2 VPC 安全群組清單。

  不適用。資料庫叢集會管理 EC2 VPC 安全群組的關聯清單。如需詳細資訊，請參閱the section called “CreateDBCluster” (p. 1005)。

  預設值：資料庫子網路群組 VPC 的預設 EC2 VPC 安全群組。

回應

包含 Amazon Neptune 資料庫執行個體的詳細資訊。

此資料類型在 the section called “DescribeDBInstances” (p. 1082) 動作中會用來作為回應元素。

- AutoMinorVersionUpgrade (在 CLI 中：`--auto-minor-version-upgrade`，或者在 Python 中：`auto_minor_version_upgrade`) -布爾值。

  指示會自動套用次要版本的修補程式。
- AvailabilityZone (在 CLI 中：`--availability-zone`或者在 Python 中：`availability_zone`) -UTF-8 字符串。

  指定資料庫執行個體所在的可用區域名稱。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- CACertificateIdentifier (在 CLI 中：`--ca-certificate-identifier`或者在 Python 中：`ca_certificate_identifier`) -UTF-8 字符串。

  此資料庫執行個體的憑證授權機構憑證識別碼。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  指定是否要從資料庫執行個體將標籤複製到資料庫執行個體的快照。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  若資料庫執行個體是資料庫叢集的成員，則包含資料庫執行個體所屬資料庫叢集的名稱。
- DBInstanceArn (在 CLI 中：`--db-instance-arn`或者在 Python 中：`db_instance_arn`) -UTF-8 字符串。

  資料庫執行個體的 Amazon Resource Name (ARN)。
- DBInstanceClass (在 CLI 中：`--db-instance-class`或者在 Python 中：`db_instance_class`) -UTF-8 字符串。

  包含資料庫執行個體之運算和記憶體容量類別的名稱。
- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或者在 Python 中：`db_instance_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫識別碼。此識別碼是可識別資料庫執行個體的唯一金鑰。
- DbInstancePort (在 CLI 中：`--db-instance-port`，或在 Python 中：`db_instance_port`) -數字 (整數) 。

  指定資料庫執行個體接聽的連接埠。若資料庫執行個體是資料庫叢集的一部分，這可以是與資料庫叢集連接埠不同的連接埠。
- DBInstanceStatus (在 CLI 中：`--db-instance-status`或者在 Python 中：`db_instance_status`) -UTF-8 字符串。

  指定此資料庫的目前狀態。
- DbiResourceId (在 CLI 中：`--dbi-resource-id`或者在 Python 中：`dbi_resource_id`) -UTF-8 字符串。

  資料庫執行個體的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫執行個體的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBName（在 CLI 中：`--db-name`或者在 Python 中：`db_name`）-UTF-8 字符串。

  資料庫名稱。
- DBParameterGroups（在 CLI 中：`--db-parameter-groups`或在 Python 中：`db_parameter_groups`）-數資料庫ParameterGroupStatus (p. 1087)據庫ParameterGroup 對象的數組。

  提供要套用到此資料庫執行個體的資料庫參數群組清單。
- DBSecurityGroups（在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`）-數資料庫 SecurityGroupMembership (p. 1156)據庫SecurityGroup 對象的數組。

  提供資料庫安全群組元素的清單，元素中只包含 `DBSecurityGroup.Name` 和 `DBSecurityGroup.Status` 子元素。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-一個資料庫SubnetGroup (p. 1103)對象。

  指定與資料庫執行個體相關聯子網路群組上的資訊，包含名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指出資料庫執行個體是否已啟用刪除保護。如果刪除保護已經啟用，執行個體將無法刪除。請參閱刪除資料庫執行個體。
- DomainMemberships（在 CLI 中：`--domain-memberships`，或在 Python 中：`domain_memberships`）-DomainMembership (p. 1156) DomainMembership 對象的數組。

  不支援
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫執行個體設定為匯出至記錄的 CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-一個端點 (p. 1157)對象。

  指定連線端點。
- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  提供用於此資料庫執行個體的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- EnhancedMonitoringResourceArn（在 CLI 中：`--enhanced-monitoring-resource-arn`或者在 Python 中：`enhanced_monitoring_resource_arn`）-UTF-8 字符串。

  接收資料庫執行個體的 Amazon CloudWatch Resource Name (ARN)。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 驗證，則為 True，否則為 false。
- InstanceCreateTime（在 CLI 中：`--instance-create-time`，或在 Python 中：`instance_create_time`）-時間戳。

  提供建立資料庫執行個體的日期和時間。
- Iops（在 CLI 中：`--iops`，或在 Python 中：`iops`）-數字（整數）。

  指定佈建 IOPS (每秒 I/O 操作) 值。

- KmsKeyId (在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`) -UTF-8 字符串。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來 point-in-time還原資料庫的最近時間。
- LicenseModel (在 CLI 中：`--license-model`或者在 Python 中：`license_model`) -UTF-8 字符串。

  此資料庫執行個體的授權模型資訊。
- MonitoringInterval (在 CLI 中：`--monitoring-interval`，或在 Python 中：`monitoring_interval`) -數字 (整數) 。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。
- MonitoringRoleArn (在 CLI 中：`--monitoring-role-arn`或者在 Python 中：`monitoring_role_arn`) -UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫執行個體是否為異地同步備份部署。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個PendingModifiedValues (p. 1149)對象。

  指定對資料庫執行個體所進行的變更仍在擱置中。此元素只有在變更擱置中時才會包含在其中。特定的變更會由子元素識別。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- PromotionTier (在 CLI 中：`--promotion-tier`，或在 Python 中：`promotion_tier`) -數字 (整數) 。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。
- PubliclyAccessible (在 CLI 中：`--publicly-accessible`，或者在 Python 中：`publicly_accessible`) -布爾值。

  此標記不應繼續使用。
- ReadReplicaDBClusterIdentifiers (在 CLI 中：`--read-replica-db-cluster-identifiers`或在 Python 中：`read_replica_db_cluster_identifiers`) -數 ReadReplica據庫數組ClusterIdentifier-UTF-8 字符串。

  包含一或多個資料庫叢集的識別符，這些叢集是此資料庫執行個體的僅供讀取複本。
- ReadReplicaDBInstanceIdentifiers (在 CLI 中：`--read-replica-db-instance-identifiers`或在 Python 中：`read_replica_db_instance_identifiers`) -數 ReadReplica據庫數組InstanceIdentifier-UTF-8 字符串。

  包含一或多個與此資料庫執行個體相關聯僅供讀取複本的識別符。
- ReadReplicaSourceDBInstanceIdentifier (在 CLI 中：`--read-replica-source-db-instance-identifier`或者在 Python 中：`read_replica_source_db_instance_identifier`) -UTF-8 字符串。

包含來源資料庫執行個體的識別符 (若此資料庫執行個體是僅供讀取複本)。

- SecondaryAvailabilityZone (在 CLI 中：`--secondary-availability-zone`或在 Python 中：`secondary_availability_zone`) -UTF-8 字符串。

  若存在，指定具備多個可用區支援之資料庫執行個體的次要可用區名稱。

- StatusInfos (在 CLI 中：`--status-infos`或在 Python 中：`status_infos`) -數資料庫 InstanceStatusInfo (p. 1054)據庫InstanceStatusInfo 對象的數組。

  僅供讀取複本的狀態。若執行個體並非僅供讀取複本，此處即為空白。

- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。

- StorageType (在 CLI 中：`--storage-type`或在 Python 中：`storage_type`) -UTF-8 字符串。

  指定與此資料庫執行個體相關聯的儲存體類型。

- TdeCredentialArn (在 CLI 中：`--tde-credential-arn`或在 Python 中：`tde_credential_arn`) -UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。

- Timezone (在 CLI 中：`--timezone`或在 Python 中：`timezone`) -UTF-8 字符串。

  不支援.

- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫執行個體所屬 VPC 安全群組元素的清單。

錯誤

- 資料庫InstanceAlreadyExistsFault (p. 1163)
- 資料庫不足InstanceCapacityFault (p. 1169)
- 資料庫ParameterGroupNotFoundFault (p. 1164)
- 資料庫SecurityGroupNotFoundFault (p. 1165)
- InstanceQuotaExceededFault (p. 1169)
- StorageQuotaExceededFault (p. 1176)
- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 資料庫SubnetGroupDoesNotCoverEnough可用區 (p. 1166)
- 无效ClusterStateFault (p. 1170)
- InvalidSubnet (p. 1173)
- 无效NetworkStateFault (p. 1173)
- ProvisionedIopsNotAvailableInAZ 錯誤 (p. 1174)
- OptionGroupNotFoundFault (p. 1173)
- 資料庫ClusterNotFoundFault (p. 1162)
- StorageTypeNotSupportedFault (p. 1176)
- AuthorizationNotFoundFault (p. 1161)
- 公里KeyNotAccessibleFault (p. 1173)
- DomainNotFoundFault (p. 1168)

DeleteDBInstance (動作)

這個 API 的亞馬遜 CLI 名稱是：delete-db-instance，Python 的名稱是：delete_db_instance。

DeleteDBInstance 動作會刪除先前佈建的資料庫執行個體。刪除資料庫執行個體時，系統會一併刪除該執行個體所有的自動化備份，且無法復原。要由 DeleteDBInstance 刪除的資料庫執行個體手動資料庫快照則不會遭到刪除。

若您請求最終資料庫快照，則 Amazon Neptune 資料庫執行個體的狀態將為 deleting，並會持續到建立資料庫快照為止。API 動作 DescribeDBInstance 會用於監控此操作的狀態。動作一旦提交，便無法取消或回復。

請注意，當資料庫執行個體處於故障狀態，且其狀態為 failed、incompatible-restore 或 incompatible-network 時，您只能在 SkipFinalSnapshot 參數設為 true 時刪除它。

若資料庫執行個體是資料庫叢集中的唯一執行個體，或其已啟用刪除保護，則您將無法刪除該執行個體。

請求

- DBInstanceIdentifier (在 CLI 中：--db-instance-identifier或在 Python 中：db_instance_identifier) -必需：UTF-8 字符串。

  要刪除之資料庫執行個體的識別符。此參數沒有大小寫之分。

  約束:

  - 必須符合現有資料庫執行個體的名稱。
- FinalDBSnapshotIdentifier (在 CLI 中：--final-db-snapshot-identifier或在 Python 中：final_db_snapshot_identifier) -UTF-8 字符串。

  設定為時建立SnapshotIdentifier 的新資料庫快照 SkipFinalSnapshot的資料庫false。

  > Note
  >
  > 指定此參數並同時將 SkipFinalShapshot 參數設定為 true 會導致錯誤。

  約束:

  - 必須為 1 到 255 個字母或數字。
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號
  - 無法在刪除僅供讀取複本時指定。
- SkipFinalSnapshot (在 CLI 中：--skip-final-snapshot，或者在 Python 中：skip_final_snapshot) -布爾值。

  判斷最終資料庫快照是否建立於資料庫執行個體刪除之前。若指定 true，則不會建立任何 DBSnapshot。若指定 false，則會在刪除資料庫執行個體前建立資料庫快照。

  請注意，當資料庫執行個體處於故障狀態且狀態為「失敗」、「不相容-還原」或「不相容的網路」時，只有在 SkipFinalSnapshot 參數設定為「true」時才能刪除該執行個體。

  在刪除僅供讀取複本時指定 true。

  > Note
  >
  > 如果是，則必須指定 FinalDBSnapshotIdentifier 參 SkipFinalSnapshot數false。

  預設：false

回應

包含 Amazon Neptune 資料庫執行個體的詳細資訊。

此資料類型在 the section called “DescribeDBInstances” (p. 1082) 動作中會用來作為回應元素。

- AutoMinorVersionUpgrade (在 CLI 中：`--auto-minor-version-upgrade`，或者在 Python 中：`auto_minor_version_upgrade`) -布爾值。

  指示會自動套用次要版本的修補程式。
- AvailabilityZone (在 CLI 中：`--availability-zone`或在 Python 中：`availability_zone`) -UTF-8 字符串。

  指定資料庫執行個體所在的可用區域名稱。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- CACertificateIdentifier (在 CLI 中：`--ca-certificate-identifier`或在 Python 中：`ca_certificate_identifier`) -UTF-8 字符串。

  此資料庫執行個體的憑證授權機構憑證識別碼。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  指定是否要從資料庫執行個體將標籤複製到資料庫執行個體的快照。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  若資料庫執行個體是資料庫叢集的成員，則包含資料庫執行個體所屬資料庫叢集的名稱。
- DBInstanceArn (在 CLI 中：`--db-instance-arn`或在 Python 中：`db_instance_arn`) -UTF-8 字符串。

  資料庫執行個體的 Amazon Resource Name (ARN)。
- DBInstanceClass (在 CLI 中：`--db-instance-class`或在 Python 中：`db_instance_class`) -UTF-8 字符串。

  包含資料庫執行個體之運算和記憶體容量類別的名稱。
- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫識別碼。此識別碼是可識別資料庫執行個體的唯一金鑰。
- DbInstancePort (在 CLI 中：`--db-instance-port`，或在 Python 中：`db_instance_port`) -數字 (整數) 。

  指定資料庫執行個體接聽的連接埠。若資料庫執行個體是資料庫叢集的一部分，這可以是與資料庫叢集連接埠不同的連接埠。
- DBInstanceStatus (在 CLI 中：`--db-instance-status`或在 Python 中：`db_instance_status`) -UTF-8 字符串。

  指定此資料庫的目前狀態。
- DbiResourceId (在 CLI 中：`--dbi-resource-id`或在 Python 中：`dbi_resource_id`) -UTF-8 字符串。

  資料庫執行個體的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫執行個體的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBName（在 CLI 中：`--db-name`或在 Python 中：`db_name`）-UTF-8 字符串。

  資料庫名稱。
- DBParameterGroups（在 CLI 中：`--db-parameter-groups`或在 Python 中：`db_parameter_groups`）-數資料庫ParameterGroupStatus (p. 1087)據庫ParameterGroup 對象的數組。

  提供要套用到此資料庫執行個體的資料庫參數群組清單。
- DBSecurityGroups（在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`）-數資料庫 SecurityGroupMembership (p. 1156)據庫SecurityGroup 對象的數組。

  提供資料庫安全群組元素的清單，元素中只包含 `DBSecurityGroup.Name` 和 `DBSecurityGroup.Status` 子元素。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-一個資料庫SubnetGroup (p. 1103)對象。

  指定與資料庫執行個體相關聯子網路群組上的資訊，包含名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指出資料庫執行個體是否已啟用刪除保護。如果刪除保護已經啟用，執行個體將無法刪除。請參閱刪除資料庫執行個體。
- DomainMemberships（在 CLI 中：`--domain-memberships`，或在 Python 中：`domain_memberships`）-DomainMembership (p. 1156) DomainMembership 對象的數組。

  不支援
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫執行個體設定為匯出至記錄的 CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-一個端點 (p. 1157)對象。

  指定連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供用於此資料庫執行個體的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- EnhancedMonitoringResourceArn（在 CLI 中：`--enhanced-monitoring-resource-arn`或在 Python 中：`enhanced_monitoring_resource_arn`）-UTF-8 字符串。

  接收資料庫執行個體的 Amazon CloudWatch Resource Name (ARN)。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 驗證，則為 True，否則為 false。
- InstanceCreateTime（在 CLI 中：`--instance-create-time`，或在 Python 中：`instance_create_time`）-時間戳。

  提供建立資料庫執行個體的日期和時間。
- Iops（在 CLI 中：`--iops`，或在 Python 中：`iops`）-數字（整數）。

  指定佈建 IOPS (每秒 I/O 操作) 值。

- KmsKeyId (在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`) -UTF-8 字符串。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來 point-in-time還原資料庫的最近時間。
- LicenseModel (在 CLI 中：`--license-model`或在 Python 中：`license_model`) -UTF-8 字符串。

  此資料庫執行個體的授權模型資訊。
- MonitoringInterval (在 CLI 中：`--monitoring-interval`，或在 Python 中：`monitoring_interval`) -數字 (整數) 。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。
- MonitoringRoleArn (在 CLI 中：`--monitoring-role-arn`或在 Python 中：`monitoring_role_arn`) -UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫執行個體是否為異地同步備份部署。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個PendingModifiedValues (p. 1149)對象。

  指定對資料庫執行個體所進行的變更仍在擱置中。此元素只有在變更擱置中時才會包含在其中。特定的變更會由子元素識別。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- PromotionTier (在 CLI 中：`--promotion-tier`，或在 Python 中：`promotion_tier`) -數字 (整數) 。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。
- PubliclyAccessible (在 CLI 中：`--publicly-accessible`，或者在 Python 中：`publicly_accessible`) -布爾值。

  此標記不應繼續使用。
- ReadReplicaDBClusterIdentifiers (在 CLI 中：`--read-replica-db-cluster-identifiers`或在 Python 中：`read_replica_db_cluster_identifiers`) -數 ReadReplica據庫數組ClusterIdentifier-UTF-8 字符串。

  包含一或多個資料庫叢集的識別符，這些叢集是此資料庫執行個體的僅供讀取複本。
- ReadReplicaDBInstanceIdentifiers (在 CLI 中：`--read-replica-db-instance-identifiers`或在 Python 中：`read_replica_db_instance_identifiers`) -數 ReadReplica據庫數組InstanceIdentifier-UTF-8 字符串。

  包含一或多個與此資料庫執行個體相關聯僅供讀取複本的識別符。
- ReadReplicaSourceDBInstanceIdentifier (在 CLI 中：`--read-replica-source-db-instance-identifier`或在 Python 中：`read_replica_source_db_instance_identifier`) -UTF-8 字符串。

包含來源資料庫執行個體的識別符 (若此資料庫執行個體是僅供讀取複本)。

- SecondaryAvailabilityZone (在 CLI 中：`--secondary-availability-zone`或在 Python 中：`secondary_availability_zone`) -UTF-8 字符串。

  若存在，指定具備多個可用區支援之資料庫執行個體的次要可用區名稱。

- StatusInfos (在 CLI 中：`--status-infos`或在 Python 中：`status_infos`) -數資料庫 InstanceStatusInfo (p. 1054)據庫InstanceStatusInfo 對象的數組。

  僅供讀取複本的狀態。若執行個體並非僅供讀取複本，此處即為空白。

- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。

- StorageType (在 CLI 中：`--storage-type`或在 Python 中：`storage_type`) -UTF-8 字符串。

  指定與此資料庫執行個體相關聯的儲存體類型。

- TdeCredentialArn (在 CLI 中：`--tde-credential-arn`或在 Python 中：`tde_credential_arn`) -UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。

- Timezone (在 CLI 中：`--timezone`或在 Python 中：`timezone`) -UTF-8 字符串。

  不支援.

- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫執行個體所屬 VPC 安全群組元素的清單。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)
- 无效InstanceStateFault (p. 1170)
- 資料庫SnapshotAlreadyExistsFault (p. 1166)
- SnapshotQuotaExceededFault (p. 1175)
- 无效ClusterStateFault (p. 1170)

ModifyDBInstance (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-instance`，Python 的名稱是：`modify_db_instance`。

修改資料庫執行個體的設定。您可以透過在請求中指定這些參數及新的值，來變更一或多個資料庫組態參數。若要了解您可以對資料庫執行個體進行的修改，請在呼叫 the section called “ModifyDBInstance” (p. 1069) 前呼叫 the section called “DescribeValid資料庫 InstanceModifications” (p. 1084)。

請求

- AllowMajorVersionUpgrade (在 CLI 中：`--allow-major-version-upgrade`，或者在 Python 中：`allow_major_version_upgrade`) -布爾值。

指示是否允許主要版本升級。變更此參數不會導致中斷，且變更會以非同步方式盡快套用。

- ApplyImmediately（在 CLI 中：`--apply-immediately`，或者在 Python 中：`apply_immediately`）- 布爾值。

  指定此請求中的修改及任何擱置中的修改是否要以非同步方式盡快套用，無論資料庫執行個體的 `PreferredMaintenanceWindow` 設定為何。

  若將此參數設為 `false`，則對資料庫執行個體的變更會在下一個維護時段期間套用。某些參數變更會造成中斷，且會在下一次呼叫 the section called “RebootDBInstance” (p. 1078)，或是下一次故障重新開機時套用。

  預設：`false`
- AutoMinorVersionUpgrade（在 CLI 中：`--auto-minor-version-upgrade`，或者在 Python 中：`auto_minor_version_upgrade`）-布爾值。

  指出會在維護時段期間將次要版本升級自動套用至資料庫執行個體。除以下狀況外，變更此參數不會導致中斷，且變更會以非同步方式盡快套用。若在維護時段期間將此參數設為 `true`，且有可用的新次要版本，並且 Neptune 已為該引擎版本啟用自動修補時，便會導致中斷。
- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`）-數字（整數）。

  不適用。自動化備份的保留期會由資料庫叢集管理。如需詳細資訊，請參閱the section called “ModifyDBCluster” (p. 1017)。

  預設值：使用現有設定
- CACertificateIdentifier（在 CLI 中：`--ca-certificate-identifier`或在 Python 中：`ca_certificate_identifier`）-UTF-8 字符串。

  指出需要與執行個體建立關聯的憑證。
- CloudwatchLogsExportConfiguration（在 CLI 中：`--cloudwatch-logs-export-configuration`或在 Python 中：`cloudwatch_logs_export_configuration`）-一個CloudwatchLogsExportConfiguration (p. 1147)對象。

  要匯出至特定資料庫執行個體或資料庫叢集的 CloudWatch 日誌類型要匯出至 Logs 的組態設定。
- CopyTagsToSnapshot（在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`）-布爾值。

  若為 True，則會從資料庫執行個體將所有標籤複製到資料庫執行個體的快照，否則為 False。預設值為 false。
- DBInstanceClass（在 CLI 中：`--db-instance-class`或在 Python 中：`db_instance_class`）-UTF-8 字符串。

  新的資料庫執行個體運算與記憶體容量，例如 `db.m4.large`。並非所有資料庫執行個體類別碼都適用於所有 Amazon 區域內。

  若您修改資料庫執行個體類別，則會在變更時發生中斷。變更會在下一個維護時段期間套用，除非針對此請求將 `ApplyImmediately` 指定為 `true`。

  預設值：使用現有設定
- DBInstanceIdentifier（在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`）-必需：UTF-8 字符串。

  資料庫執行個體識別符。此值會以小寫字母字串的形式儲存。

  約束:
  - 必須符合現有 DBInstance 的識別符。

- DBParameterGroupName（在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`）-UTF-8 字符串。

  要套用到資料庫執行個體的資料庫參數群組名稱。變更此設定不會導致中斷。參數群組名稱本身會立即變更，但在您重新啓動實例而不需容錯移轉的情況下，才會套用實際參數變更。資料庫執行個體「不會」自動重新開機，且參數變更也「不會」在下一個維護時段期間套用。

  預設值：使用現有設定

  限制條件：資料庫參數群組必須位於和此資料庫執行個體相同的資料庫參數群組系列中。
- DBPortNumber（在 CLI 中：`--db-port-number`，或在 Python 中：`db_port_number`）-數字（整數）。

  資料庫接受連線的連接埠號碼。

  `DBPortNumber` 參數的值不得和任何在資料庫執行個體選項群組中為選項指定的連接埠值相符。

  當您變更 `DBPortNumber` 的值時，您的資料庫會重新啟動，無論 `ApplyImmediately` 參數的值為何。

  預設：8182
- DBSecurityGroups（在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`）-數據庫數組SecurityGroupName-UTF-8 字符串。

  要在此資料庫執行個體上授權的資料庫安全群組清單。變更此設定不會導致中斷，且變更會以非同步方式盡快套用。

  約束:
  - 如果提供，則必須與現有數據庫匹配SecurityGroups。
- DBSubnetGroupName（在 CLI 中：`--db-subnet-group-name`或在 Python 中：`db_subnet_group_name`）-UTF-8 字符串。

  資料庫執行個體的新資料庫子網路群組。您可以使用此參數將資料庫執行個體移動至不同的 VPC。

  變更子網路群組會在變更期間造成中斷。變更會在下一個維護時段期間套用，除非您針對 `ApplyImmediately` 參數指定 `true`。

  限制條件：若有提供，必須符合現有資料庫的名稱SubnetGroup。

  範例：`mySubnetGroup`
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫執行個體是否啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是停用的。請參閱刪除資料庫執行個體。
- Domain（在 CLI 中：`--domain`或在 Python 中：`domain`）-UTF-8 字符串。

  不支援.
- DomainIAMRoleName（在 CLI 中：`--domain-iam-role-name`或在 Python 中：`domain_iam_role_name`）-UTF-8 字符串。

  不支援
- EnableIAMDatabaseAuthentication（在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`）-布爾值。

  True，否則為使 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，否則為 false。

  您可以為以下資料庫引擎啟用 IAM 資料庫身分驗證

不適用。將 Amazon IAM 帳戶映射至資料庫帳戶由資料庫叢集管理。如需詳細資訊，請參閱the section called “ModifyDBCluster” (p. 1017)。

預設：false

- EngineVersion（在 CLI 中：--engine-version或在 Python 中：engine_version）-UTF-8 字符串。

  要升級到的目標資料庫引擎版本號碼。目前，設定此參數沒有任何作用。若要將資料庫引擎升級至最新版本，請使用 the section called “ApplyPendingMaintenanceAction” (p. 1152) API。

- Iops（在 CLI 中：--iops，或在 Python 中：iops）-數字（整數）。

  執行個體的新佈建 IOPS (每秒 I/O 操作數) 值。

  變更此設定不會導致中斷，且變更會在下一個維護時段期間套用，除非針對此請求將 ApplyImmediately 參數設為 true。

  預設值：使用現有設定

- MonitoringInterval（在 CLI 中：--monitoring-interval，或在 Python 中：monitoring_interval）-數字（整數）。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。若要停用收集增強型監控指標，請指定 0。預設值為 0。

  若指定 MonitoringRoleArn，您必須也將 MonitoringInterval 設為 0 之外的值。

  有效值: 0, 1, 5, 10, 15, 30, 60

- MonitoringRoleArn（在 CLI 中：--monitoring-role-arn或在 Python 中：monitoring_role_arn）-UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。例如：arn:aws:iam:123456789012:role/emaccess。

  若 MonitoringInterval 設為 0 之外的值，則您必須提供 MonitoringRoleArn 值。

- MultiAZ（在 CLI 中：--multi-az，或者在 Python 中：multi_az）-布爾值。

  指定資料庫執行個體是否為異地同步備份部署。變更此參數不會導致中斷，且變更會在下一個維護時段期間套用，除非針對此請求將 ApplyImmediately 參數設為 true。

- NewDBInstanceIdentifier（在 CLI 中：--new-db-instance-identifier或在 Python 中：new_db_instance_identifier）-UTF-8 字符串。

  重新命名資料庫執行個體時，資料庫執行個體的新資料庫執行個體識別符。當您變更資料庫執行個體識別符時，若您將 Apply Immediately 設為 True，則執行個體會立即重新開機。若您將 Apply Immediately 設為 False，則會在下一個維護時段期間重新開機。此值會以小寫字母字串的形式儲存。

  約束:

  - 必須包含 1 到 63 個字母、數字或連字號。
  - 第一個字元必須是字母。
  - 不能以連字號結尾或連續包含兩個連字號。

  範例：mydbinstance

- PreferredBackupWindow（在 CLI 中：--preferred-backup-window或在 Python 中：preferred_backup_window）-UTF-8 字符串。

  若已啟用自動化備份，則為每日時間範圍，自動化備份會在此期間建立。

不適用。建立自動化備份的每日時間範圍會由資料庫叢集管理。如需詳細資訊，請參閱the section called “ModifyDBCluster” (p. 1017)。

約束:

- 格式必須為 hh24:mi-hh24:mi
- 必須以國際標準時間 (UTC) 表示
- 不得和慣用的維護時段衝突
- 必須至少 30 分鐘

- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  每週可能進行系統維護的時間範圍 (UTC)，在此期間可能會發生中斷。除以下狀況外，變更此參數不會導致中斷，且變更會以非同步方式盡快套用。若有任何擱置中會造成重新開機的動作，且維護時段已變更成包含目前的時間，則變更此參數便會使資料庫執行個體重新開機。若將此時段移動到目前時間，則在目前時間和時段結束的時間之間必須至少間隔 30 分鐘，以確保套用擱置中的變更。

  預設值：使用現有設定

  格式：ddd:hh24:mi-ddd:hh24:mi

  有效日：星期一 | 星期二 | 星期三 | 星期四 | 星期五 | 星期六 | 星期日

  限制條件：必須至少 30 分鐘
- PromotionTier (在 CLI 中：`--promotion-tier`，或在 Python 中：`promotion_tier`) -數字 (整數) 。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。

  預設：1

  有效值：0 到 15
- PubliclyAccessible (在 CLI 中：`--publicly-accessible`，或者在 Python 中：`publicly_accessible`) -布爾值。

  此標記不應繼續使用。
- StorageType (在 CLI 中：`--storage-type`或在 Python 中：`storage_type`) -UTF-8 字符串。

  不支援.
- TdeCredentialArn (在 CLI 中：`--tde-credential-arn`或在 Python 中：`tde_credential_arn`) -UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。
- TdeCredentialPassword (在 CLI 中：`--tde-credential-password`或在 Python 中：`tde_credential_password`) -UTF-8 字符串。

  在金鑰存放區指定 ARN 的密碼，用於存取裝置。
- VpcSecurityGroupIds (在 CLI 中：`--vpc-security-group-ids`或在 Python 中：`vpc_security_group_ids`) -一個數組 VpcSecurityGroupId-UTF-8 字符串。

  要在此資料庫執行個體上授權的 EC2 VPC 安全群組清單。此變更會以非同步的方式盡快套用。

  不適用。資料庫叢集會管理 EC2 VPC 安全群組的關聯清單。如需詳細資訊，請參閱the section called “ModifyDBCluster” (p. 1017)。

  約束:

  - 如果提供，則必須符合現有的 VpcSecurityGroupIds。

回應

包含 Amazon Neptune 資料庫執行個體的詳細資訊。

此資料類型在 the section called “DescribeDBInstances” (p. 1082) 動作中會用來作為回應元素。

- AutoMinorVersionUpgrade (在 CLI 中：`--auto-minor-version-upgrade`，或者在 Python 中：`auto_minor_version_upgrade`) -布爾值。

  指示會自動套用次要版本的修補程式。
- AvailabilityZone (在 CLI 中：`--availability-zone`或在 Python 中：`availability_zone`) -UTF-8 字符串。

  指定資料庫執行個體所在的可用區域名稱。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- CACertificateIdentifier (在 CLI 中：`--ca-certificate-identifier`或在 Python 中：`ca_certificate_identifier`) -UTF-8 字符串。

  此資料庫執行個體的憑證授權機構憑證識別碼。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  指定是否要從資料庫執行個體將標籤複製到資料庫執行個體的快照。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  若資料庫執行個體是資料庫叢集的成員，則包含資料庫執行個體所屬資料庫叢集的名稱。
- DBInstanceArn (在 CLI 中：`--db-instance-arn`或在 Python 中：`db_instance_arn`) -UTF-8 字符串。

  資料庫執行個體的 Amazon Resource Name (ARN)。
- DBInstanceClass (在 CLI 中：`--db-instance-class`或在 Python 中：`db_instance_class`) -UTF-8 字符串。

  包含資料庫執行個體之運算和記憶體容量類別的名稱。
- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫識別碼。此識別碼是可識別資料庫執行個體的唯一金鑰。
- DbInstancePort (在 CLI 中：`--db-instance-port`，或在 Python 中：`db_instance_port`) -數字 (整數) 。

  指定資料庫執行個體接聽的連接埠。若資料庫執行個體是資料庫叢集的一部分，這可以是與資料庫叢集連接埠不同的連接埠。
- DBInstanceStatus (在 CLI 中：`--db-instance-status`或在 Python 中：`db_instance_status`) -UTF-8 字符串。

  指定此資料庫的目前狀態。
- DbiResourceId (在 CLI 中：`--dbi-resource-id`或在 Python 中：`dbi_resource_id`) -UTF-8 字符串。

  資料庫執行個體的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫執行個體的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBName（在 CLI 中：`--db-name`或在 Python 中：`db_name`）-UTF-8 字符串。

  資料庫名稱。
- DBParameterGroups（在 CLI 中：`--db-parameter-groups`或在 Python 中：`db_parameter_groups`）-數資料庫ParameterGroupStatus (p. 1087)據庫ParameterGroup 對象的數組。

  提供要套用到此資料庫執行個體的資料庫參數群組清單。
- DBSecurityGroups（在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`）-數資料庫 SecurityGroupMembership (p. 1156)據庫SecurityGroup 對象的數組。

  提供資料庫安全群組元素的清單，元素中只包含 `DBSecurityGroup.Name` 和 `DBSecurityGroup.Status` 子元素。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-一個資料庫SubnetGroup (p. 1103)對象。

  指定與資料庫執行個體相關聯子網路群組上的資訊，包含名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指出資料庫執行個體是否已啟用刪除保護。如果刪除保護已經啟用，執行個體將無法刪除。請參閱刪除資料庫執行個體。
- DomainMemberships（在 CLI 中：`--domain-memberships`，或在 Python 中：`domain_memberships`）-DomainMembership (p. 1156) DomainMembership 對象的數組。

  不支援
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫執行個體設定為匯出至記錄的 CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或在 Python 中：`endpoint`）-一個端點 (p. 1157)對象。

  指定連線端點。
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-UTF-8 字符串。

  提供用於此資料庫執行個體的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- EnhancedMonitoringResourceArn（在 CLI 中：`--enhanced-monitoring-resource-arn`或在 Python 中：`enhanced_monitoring_resource_arn`）-UTF-8 字符串。

  接收資料庫執行個體的 Amazon CloudWatch Resource Name (ARN)。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 驗證，則為 True，否則為 false。
- InstanceCreateTime（在 CLI 中：`--instance-create-time`，或在 Python 中：`instance_create_time`）-時間戳。

  提供建立資料庫執行個體的日期和時間。
- Iops（在 CLI 中：`--iops`，或在 Python 中：`iops`）-數字（整數）。

  指定佈建 IOPS (每秒 I/O 操作) 值。

- KmsKeyId (在 CLI 中：`--kms-key-id`或在 Python 中：`kms_key_id`) -UTF-8 字符串。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來 point-in-time還原資料庫的最近時間。
- LicenseModel (在 CLI 中：`--license-model`或在 Python 中：`license_model`) -UTF-8 字符串。

  此資料庫執行個體的授權模型資訊。
- MonitoringInterval (在 CLI 中：`--monitoring-interval`，或在 Python 中：`monitoring_interval`) -數字 (整數) 。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。
- MonitoringRoleArn (在 CLI 中：`--monitoring-role-arn`或在 Python 中：`monitoring_role_arn`) -UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫執行個體是否為異地同步備份部署。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個PendingModifiedValues (p. 1149)對象。

  指定對資料庫執行個體所進行的變更仍在擱置中。此元素只有在變更擱置中時才會包含在其中。特定的變更會由子元素識別。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- PromotionTier (在 CLI 中：`--promotion-tier`，或在 Python 中：`promotion_tier`) -數字 (整數) 。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。
- PubliclyAccessible (在 CLI 中：`--publicly-accessible`，或者在 Python 中：`publicly_accessible`) -布爾值。

  此標記不應繼續使用。
- ReadReplicaDBClusterIdentifiers (在 CLI 中：`--read-replica-db-cluster-identifiers`或在 Python 中：`read_replica_db_cluster_identifiers`) -數 ReadReplica據庫數組ClusterIdentifier-UTF-8 字符串。

  包含一或多個資料庫叢集的識別符，這些叢集是此資料庫執行個體的僅供讀取複本。
- ReadReplicaDBInstanceIdentifiers (在 CLI 中：`--read-replica-db-instance-identifiers`或在 Python 中：`read_replica_db_instance_identifiers`) -數 ReadReplica據庫數組InstanceIdentifier-UTF-8 字符串。

  包含一或多個與此資料庫執行個體相關聯僅供讀取複本的識別符。
- ReadReplicaSourceDBInstanceIdentifier (在 CLI 中：`--read-replica-source-db-instance-identifier`或在 Python 中：`read_replica_source_db_instance_identifier`) -UTF-8 字符串。

包含來源資料庫執行個體的識別符 (若此資料庫執行個體是僅供讀取複本)。

- SecondaryAvailabilityZone (在 CLI 中：`--secondary-availability-zone`或在 Python 中：`secondary_availability_zone`) -UTF-8 字符串。

  若存在，指定具備多個可用區支援之資料庫執行個體的次要可用區名稱。

- StatusInfos (在 CLI 中：`--status-infos`或在 Python 中：`status_infos`) -數資料庫 InstanceStatusInfo (p. 1054)據庫InstanceStatusInfo 對象的數組。

  僅供讀取複本的狀態。若執行個體並非僅供讀取複本，此處即為空白。

- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。

- StorageType (在 CLI 中：`--storage-type`或在 Python 中：`storage_type`) -UTF-8 字符串。

  指定與此資料庫執行個體相關聯的儲存體類型。

- TdeCredentialArn (在 CLI 中：`--tde-credential-arn`或在 Python 中：`tde_credential_arn`) -UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。

- Timezone (在 CLI 中：`--timezone`或在 Python 中：`timezone`) -UTF-8 字符串。

  不支援.

- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫執行個體所屬 VPC 安全群組元素的清單。

錯誤

- 无效InstanceStateFault (p. 1170)
- 无效SecurityGroupStateFault (p. 1171)
- 資料庫InstanceAlreadyExistsFault (p. 1163)
- 資料庫InstanceNotFoundFault (p. 1164)
- 資料庫SecurityGroupNotFoundFault (p. 1165)
- 資料庫ParameterGroupNotFoundFault (p. 1164)
- 資料庫不足InstanceCapacityFault (p. 1169)
- StorageQuotaExceededFault (p. 1176)
- 无效NetworkStateFault (p. 1173)
- ProvisionedIopsNotAvailableInAZ 錯誤 (p. 1174)
- OptionGroupNotFoundFault (p. 1173)
- 資料庫UpgradeDependencyFailureFault (p. 1167)
- StorageTypeNotSupportedFault (p. 1176)
- AuthorizationNotFoundFault (p. 1161)
- CertificateNotFoundFault (p. 1161)
- DomainNotFoundFault (p. 1168)

RebootDBInstance (動作)

這個 API 的亞馬遜 CLI 名稱是：`reboot-db-instance`，Python 的名稱是：`reboot_db_instance`。

您可能需要重新啟動資料庫執行個體，這通常是為了維護的原因。例如，如果您進行某些修改，或變更與資料庫執行個體相關聯的資料庫參數群組，則必須重新啓動執行個體，變更才會生效。

重新啟動資料庫執行個體，將重新啟動資料庫引擎服務。重新啟動資料庫執行個體會暫時中斷，在此期間，資料庫執行個體狀態設定為 rebooting (重新啟動中)。

請求

- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`) -必需：UTF-8 字符串。

  資料庫執行個體識別符。此參數是以小寫字母字串的形式儲存。

  約束:

  - 必須符合現有 DBInstance 的識別符。
- ForceFailover (在 CLI 中：`--force-failover`，或者在 Python 中：`force_failover`) -布爾值。

  若為 `true`，則會透過多個可用區容錯移轉進行重新開機。

  限制條件：若執行個體並未針對多個可用區進行設定，您便無法指定 `true`。

回應

包含 Amazon Neptune 資料庫執行個體的詳細資訊。

此資料類型在 the section called “DescribeDBInstances” (p. 1082) 動作中會用來作為回應元素。

- AutoMinorVersionUpgrade (在 CLI 中：`--auto-minor-version-upgrade`，或者在 Python 中：`auto_minor_version_upgrade`) -布爾值。

  指示會自動套用次要版本的修補程式。
- AvailabilityZone (在 CLI 中：`--availability-zone`或在 Python 中：`availability_zone`) -UTF-8 字符串。

  指定資料庫執行個體所在的可用區域名稱。
- BackupRetentionPeriod (在 CLI 中：`--backup-retention-period`，或在 Python 中：`backup_retention_period`) -數字 (整數) 。

  指定保留自動資料庫快照的天數。
- CACertificateIdentifier (在 CLI 中：`--ca-certificate-identifier`或者在 Python 中：`ca_certificate_identifier`) -UTF-8 字符串。

  此資料庫執行個體的憑證授權機構憑證識別碼。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  指定是否要從資料庫執行個體將標籤複製到資料庫執行個體的快照。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  若資料庫執行個體是資料庫叢集的成員，則包含資料庫執行個體所屬資料庫叢集的名稱。

- DBInstanceArn（在 CLI 中：`--db-instance-arn`或者在 Python 中：`db_instance_arn`）-UTF-8 字符串。

  資料庫執行個體的 Amazon Resource Name (ARN)。
- DBInstanceClass（在 CLI 中：`--db-instance-class`或者在 Python 中：`db_instance_class`）-UTF-8 字符串。

  包含資料庫執行個體之運算和記憶體容量類別的名稱。
- DBInstanceIdentifier（在 CLI 中：`--db-instance-identifier`或者在 Python 中：`db_instance_identifier`）-UTF-8 字符串。

  包含使用者提供的資料庫識別碼。此識別碼是可識別資料庫執行個體的唯一金鑰。
- DbInstancePort（在 CLI 中：`--db-instance-port`，或在 Python 中：`db_instance_port`）-數字（整數）。

  指定資料庫執行個體接聽的連接埠。若資料庫執行個體是資料庫叢集的一部分，這可以是與資料庫叢集連接埠不同的連接埠。
- DBInstanceStatus（在 CLI 中：`--db-instance-status`或者在 Python 中：`db_instance_status`）-UTF-8 字符串。

  指定此資料庫的目前狀態。
- DbiResourceId（在 CLI 中：`--dbi-resource-id`或者在 Python 中：`dbi_resource_id`）-UTF-8 字符串。

  資料庫執行個體的 Amazon 區域專屬不可變識別碼。此識別碼可在存取資料庫執行個體的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。
- DBName（在 CLI 中：`--db-name`或者在 Python 中：`db_name`）-UTF-8 字符串。

  資料庫名稱。
- DBParameterGroups（在 CLI 中：`--db-parameter-groups`或在 Python 中：`db_parameter_groups`）-數資料庫ParameterGroupStatus (p. 1087)據庫ParameterGroup 對象的數組。

  提供要套用到此資料庫執行個體的資料庫參數群組清單。
- DBSecurityGroups（在 CLI 中：`--db-security-groups`或在 Python 中：`db_security_groups`）-數資料庫 SecurityGroupMembership (p. 1156)據庫SecurityGroup 對象的數組。

  提供資料庫安全群組元素的清單，元素中只包含 `DBSecurityGroup.Name` 和 `DBSecurityGroup.Status` 子元素。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或在 Python 中：`db_subnet_group`）-一個資料庫SubnetGroup (p. 1103)對象。

  指定與資料庫執行個體相關聯子網路群組上的資訊，包含名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指出資料庫執行個體是否已啟用刪除保護。如果刪除保護已經啟用，執行個體將無法刪除。請參閱刪除資料庫執行個體。
- DomainMemberships（在 CLI 中：`--domain-memberships`，或在 Python 中：`domain_memberships`）-DomainMembership (p. 1156) DomainMembership 對象的數組。

  不支援
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫執行個體設定為匯出至記錄的 CloudWatch記錄類型清單。

- Endpoint (在 CLI 中：`--endpoint`或在 Python 中：`endpoint`) -一個端點 (p. 1157)對象。

  指定連線端點。
- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  提供用於此資料庫執行個體的資料庫引擎名稱。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  指示資料庫引擎版本。
- EnhancedMonitoringResourceArn (在 CLI 中：`--enhanced-monitoring-resource-arn`或者在 Python 中：`enhanced_monitoring_resource_arn`) -UTF-8 字符串。

  接收資料庫執行個體的 Amazon CloudWatch Resource Name (ARN)。
- IAMDatabaseAuthenticationEnabled (在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`) -布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 驗證，則為 True，否則為 false。
- InstanceCreateTime (在 CLI 中：`--instance-create-time`，或在 Python 中：`instance_create_time`) -時間戳。

  提供建立資料庫執行個體的日期和時間。
- Iops (在 CLI 中：`--iops`，或在 Python 中：`iops`) -數字 (整數) 。

  指定佈建 IOPS (每秒 I/O 操作) 值。
- KmsKeyId (在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`) -UTF-8 字符串。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來 point-in-time還原資料庫的最近時間。
- LicenseModel (在 CLI 中：`--license-model`或者在 Python 中：`license_model`) -UTF-8 字符串。

  此資料庫執行個體的授權模型資訊。
- MonitoringInterval (在 CLI 中：`--monitoring-interval`，或在 Python 中：`monitoring_interval`) -數字 (整數) 。

  為資料庫執行個體收集增強型監控指標點之間的間隔 (秒)。
- MonitoringRoleArn (在 CLI 中：`--monitoring-role-arn`或者在 Python 中：`monitoring_role_arn`) -UTF-8 字符串。

  允許 Neptune 將增強型監控指標傳送到 Amazon CloudWatch Logs 之 IAM 角色的 ARN。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫執行個體是否為異地同步備份部署。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個PendingModifiedValues (p. 1149)對象。

  指定對資料庫執行個體所進行的變更仍在擱置中。此元素只有在變更擱置中時才會包含在其中。特定的變更會由子元素識別。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。

- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- PromotionTier (在 CLI 中：`--promotion-tier`，或在 Python 中：`promotion_tier`) -數字 (整數) 。

  此值指定在現有主要執行個體故障後，僅供讀取複本提升為主要執行個體的順序。
- PubliclyAccessible (在 CLI 中：`--publicly-accessible`，或者在 Python 中：`publicly_accessible`) -布爾值。

  此標記不應繼續使用。
- ReadReplicaDBClusterIdentifiers (在 CLI 中：`--read-replica-db-cluster-identifiers`或在 Python 中：`read_replica_db_cluster_identifiers`) -數 ReadReplica據庫數組ClusterIdentifier-UTF-8 字符串。

  包含一或多個資料庫叢集的識別符，這些叢集是此資料庫執行個體的僅供讀取複本。
- ReadReplicaDBInstanceIdentifiers (在 CLI 中：`--read-replica-db-instance-identifiers`或在 Python 中：`read_replica_db_instance_identifiers`) -數 ReadReplica據庫數組InstanceIdentifier-UTF-8 字符串。

  包含一或多個與此資料庫執行個體相關聯僅供讀取複本的識別符。
- ReadReplicaSourceDBInstanceIdentifier (在 CLI 中：`--read-replica-source-db-instance-identifier`或者在 Python 中：`read_replica_source_db_instance_identifier`) -UTF-8 字符串。

  包含來源資料庫執行個體的識別符 (若此資料庫執行個體是僅供讀取複本)。
- SecondaryAvailabilityZone (在 CLI 中：`--secondary-availability-zone`或者在 Python 中：`secondary_availability_zone`) -UTF-8 字符串。

  若存在，指定具備多個可用區支援之資料庫執行個體的次要可用區名稱。
- StatusInfos (在 CLI 中：`--status-infos`或在 Python 中：`status_infos`) -數資料庫InstanceStatusInfo (p. 1054)據庫InstanceStatusInfo 對象的數組。

  僅供讀取複本的狀態。若執行個體並非僅供讀取複本，此處即為空白。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  不支援：資料庫執行個體的加密會由資料庫叢集管理。
- StorageType (在 CLI 中：`--storage-type`或者在 Python 中：`storage_type`) -UTF-8 字符串。

  指定與此資料庫執行個體相關聯的儲存體類型。
- TdeCredentialArn (在 CLI 中：`--tde-credential-arn`或者在 Python 中：`tde_credential_arn`) -UTF-8 字符串。

  來自金鑰存放區的 ARN，執行個體會與此 ARN 建立關聯以進行 TDE 加密。
- Timezone (在 CLI 中：`--timezone`或者在 Python 中：`timezone`) -UTF-8 字符串。

  不支援.
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫執行個體所屬 VPC 安全群組元素的清單。

錯誤

- 无效InstanceStateFault (p. 1170)
- 資料庫InstanceNotFoundFault (p. 1164)

DescribeDBInstances (動作)

這個 API 的亞馬遜 CLI 名稱是：`describe-db-instances`，Python 的名稱是：`describe_db_instances`。

傳回所佈建之執行個體的相關資訊，並支援分頁。

Note

此操作也可以傳回 Amazon RDS 執行個體和 Amazon DocDB 執行個體的資訊。

請求

- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或者在 Python 中：`db_instance_identifier`) -UTF-8 字符串。

  使用者提供的執行個體識別符。若指定此參數，則只會傳回特定資料庫執行個體的資訊。此參數沒有大小寫之分。

  約束:

  - 若有提供，則必須符合現有 DBInstance 的識別符。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  篩選條件，指定要描述的一或多個資料庫執行個體。

  支援的篩選條件：

  - `db-cluster-id` - 接受資料庫叢集識別符及資料庫叢集 Amazon Resource Name (ARN)。結果清單只會包含與透過這些 ARN 識別資料庫叢集相關聯的資料庫執行個體資訊。
  - `engine` -接受引擎名稱 (例如 `neptune`)，並將結果清單限制為由該引擎建立的資料庫執行個體。

  例如，若要從 Amazon CLI 呼叫此 API 和篩選器，以便僅傳回 Neptune 資料庫執行個體，您可以使用下列命令：

  Example

```
aws neptune describe-db-instances \
            --filters  Name=engine,Values=neptune
```

- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個 `DescribeDBInstances` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`，或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

- DBInstances (在 CLI 中：`--db-instances`或在 Python 中：`db_instances`) - 數DBInstance (p. 1051)據庫實例對象的數組。

  the section called “DBInstance” (p. 1051) 執行個體的清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)

DescribeOrderable資料庫InstanceOptions (動作)

這個 API 的亞馬遜 CLI 名稱是：`describe-orderable-db-instance-options`，Python 的名稱是：`describe_orderable_db_instance_options`。

傳回指定引擎的可排序資料庫執行個體選項清單。

請求

- DBInstanceClass (在 CLI 中：`--db-instance-class`或者在 Python 中：`db_instance_class`) - UTF-8 字符串。

  資料庫執行個體類別篩選條件值。指定此參數，以僅顯示與指定資料庫執行個體類別相符的可用供應項目。
- Engine (在 CLI 中：`--engine`或在 Python 中：`engine`) -必需：UTF-8 字符串。

  要為其擷取資料庫執行個體選項的引擎名稱。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  引擎版本篩選條件值。指定此參數，以僅顯示與指定引擎版本相符的可用供應項目。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- LicenseModel (在 CLI 中：`--license-model`或者在 Python 中：`license_model`) -UTF-8 字符串。

  授權模型篩選條件值。指定此參數，以僅顯示與指定授權模型相符的可用供應項目。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  由先前的 DescribeOrderable數據庫InstanceOptions請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`，或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- Vpc (在 CLI 中：`--vpc`，或者在 Python 中：`vpc`) -布爾值。

  VPC 篩選條件值。指定此參數，以僅顯示可用的 VPC 或非 VPC 供應項目。

回應

- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  由先前的 OrderableDBInstanceOptions 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- OrderableDBInstanceOptions (在 CLI 中：`--orderable-db-instance-options`或在 Python 中：`orderable_db_instance_options`) -可排可訂購數據庫InstanceOption (p. 1055)序數據庫 InstanceOption 對象的數組。

  the section called “可訂購數據庫InstanceOption” (p. 1055) 結構，包含資料庫執行個體可排序選項的資訊。

DescribeValid資料庫InstanceModifications (動作)

這個 API 的亞馬遜 CLI 名稱是：`describe-valid-db-instance-modifications`，Python 的名稱是：`describe_valid_db_instance_modifications`。

您可以呼叫 the section called “DescribeValid資料庫InstanceModifications” (p. 1084) 來了解您可以對資料庫執行個體進行的修改。您可以在呼叫 the section called “ModifyDBInstance” (p. 1069) 時使用此資訊。

請求

- DBInstanceIdentifier (在 CLI 中：`--db-instance-identifier`或在 Python 中：`db_instance_identifier`) -必需：UTF-8 字符串。

  客戶識別符或您資料庫執行個體的 ARN。

回應

您可以對資料庫執行個體進行的有效修改相關資訊。包含成功呼叫 the section called “DescribeValid資料庫InstanceModifications” (p. 1084) 動作的結果。您可以在呼叫 the section called “ModifyDBInstance” (p. 1069) 時使用此資訊。

- Storage (在 CLI 中：`--storage`或在 Python 中：`storage`) -一個數組 ValidStorageOptions-一個結構。

  您資料庫執行個體的有效儲存體選項。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)
- 无效InstanceStateFault (p. 1170)

Neptune 參數 API

結構：

- Parameter (結構) (p. 1085)
- 資ParameterGroup 料庫結構 (p. 1086)
- 資ClusterParameterGroup 料庫結構 (p. 1086)
- 資ParameterGroupStatus 料庫結構 (p. 1087)

動作:

- 複製資料庫ParameterGroup (動作) (p. 1087)
- 複製資料庫ClusterParameterGroup (動作) (p. 1088)
- 建立資料庫ParameterGroup (動作) (p. 1090)
- 建立資料庫ClusterParameterGroup (動作) (p. 1091)
- 刪除資料庫ParameterGroup (動作) (p. 1092)
- 刪除資料庫ClusterParameterGroup (動作) (p. 1093)
- 修改資料庫ParameterGroup (動作) (p. 1093)
- 修改資料庫ClusterParameterGroup (動作) (p. 1094)
- 重設資料庫ParameterGroup (動作) (p. 1095)
- 重設資料庫ClusterParameterGroup (動作) (p. 1096)
- DescribeDBParameters (動作) (p. 1097)
- 描述 BParameterGroups （ 動作 ） (p. 1098)
- 描述 BClusterParameters （ 動作 ） (p. 1099)
- 描述 BClusterParameterGroups （ 動作 ） (p. 1100)
- DescribeEngineDefaultParameters （ 動作 ） (p. 1101)
- DescribeEngineDefaultClusterParameters （ 動作 ） (p. 1102)

Parameter (結構)

指定參數。

欄位

- AllowedValues – UTF-8 字串。

  指定參數值的有效範圍。
- ApplyMethod – UTF-8 字串 (有效值：`immediate=""` | `pending-reboot=""`)。

  指出何時套用參數更新。
- ApplyType – UTF-8 字串。

  指定引擎限定的參數類型。
- DataType – UTF-8 字串。

  指定參數的有效資料類型。
- Description – UTF-8 字串。

  提供參數的描述。
- IsModifiable – 布林值。

  指出是 (`true`) 否 (`false`) 可以修改參數。有些參數具有安全或操作上的隱含式，防止他們遭到變更。
- MinimumEngineVersion – UTF-8 字串。

  可套用參數的最早引擎版本。
- ParameterName – UTF-8 字串。

  指定參數的名稱。

- ParameterValue – UTF-8 字串。

  指定參數的值。
- Source – UTF-8 字串。

  指出參數值的來源。

資ParameterGroup 料庫結構

包含 Amazon Neptune 資料庫參數群組的詳細資訊。

此資料類型在 the section called “描述 BParameterGroups” (p. 1098) 動作中會用來作為回應元素。

欄位

- DBParameterGroupArn – UTF-8 字串。

  資料庫參數群組的 Amazon Resource Name (ARN)。
- DBParameterGroupFamily – UTF-8 字串。

  提供此資料庫參數群組相容的資料庫參數群組系列名稱。
- DBParameterGroupName – UTF-8 字串。

  提供資料庫參數群組的名稱。
- Description – UTF-8 字串。

  提供此資料庫參數群組的客戶指定描述。

`DBParameterGroup` 會用來做為以下項目的回應元素：

- 複製資料庫ParameterGroup (p. 1087)
- 創建數據庫ParameterGroup (p. 1090)

資ClusterParameterGroup 料庫結構

包含 Amazon Neptune 資料庫叢集參數群組的詳細資訊。

此資料類型在 the section called “描述 BClusterParameterGroups” (p. 1100) 動作中會用來作為回應元素。

欄位

- DBClusterParameterGroupArn – UTF-8 字串。

  資料庫叢集參數群組的 Amazon Resource Name (ARN)。
- DBClusterParameterGroupName – UTF-8 字串。

  提供資料庫叢集參數群組的名稱。
- DBParameterGroupFamily – UTF-8 字串。

  提供此資料庫叢集參數群組相容的資料庫參數群組系列名稱。
- Description – UTF-8 字串。

  提供該客戶指定之此資料庫叢集參數群組的描述。

DBClusterParameterGroup 會用來做為以下項目的回應元素：

- 複製資料庫ClusterParameterGroup (p. 1088)
- 創建數據庫ClusterParameterGroup (p. 1091)

資ParameterGroupStatus 料庫結構

資料庫參數群組的狀態。

此資料類型在下列動作中會用來做為回應元素：

- the section called “CreateDBInstance” (p. 1056)
- the section called “DeleteDBInstance” (p. 1065)
- the section called “ModifyDBInstance” (p. 1069)
- the section called “RebootDBInstance” (p. 1078)

欄位

- DBParameterGroupName – UTF-8 字串。

  資料庫參數群組的名稱。
- ParameterApplyStatus – UTF-8 字串。

  參數更新的狀態。

複製資料庫ParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：copy-db-parameter-group，Python 的名稱是：copy_db_parameter_group。

複製指定的資料庫參數群組。

請求

- SourceDBParameterGroupIdentifier（在 CLI 中：--source-db-parameter-group-identifier或在 Python 中：source_db_parameter_group_identifier）-必需：UTF-8 字符串。

  來源資料庫參數群組的識別符或 ARN。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。

  約束:
  - 必須指定有效的資料庫參數群組。
  - 必須指定有效的資料庫參數群組識別符 (例如 my-db-param-group)，或有效的 ARN。
- Tags（在 CLI 中：--tags或在 Python 中：tags）-Tag (p. 1150) 標籤對象的數組。

  要指派給所複製資料庫參數群組的標籤。
- TargetDBParameterGroupDescription（在 CLI 中：--target-db-parameter-group-description或在 Python 中：target_db_parameter_group_description）-必需：UTF-8 字符串。

  所複製資料庫參數群組的描述。
- TargetDBParameterGroupIdentifier（在 CLI 中：--target-db-parameter-group-identifier或在 Python 中：target_db_parameter_group_identifier）-必需：UTF-8 字符串。

所複製資料庫參數群組的識別符。

約束:

- 不得為無效、空白或留空。
- 必須包含 1 到 255 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：`my-db-parameter-group`

回應

包含 Amazon Neptune 資料庫參數群組的詳細資訊。

此資料類型在 the section called “描述 BParameterGroups” (p. 1098) 動作中會用來作為回應元素。

- DBParameterGroupArn (在 CLI 中：`--db-parameter-group-arn`或在 Python 中：`db_parameter_group_arn`) -UTF-8 字符串。

  資料庫參數群組的 Amazon Resource Name (ARN)。
- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`) -UTF-8 字符串。

  提供此資料庫參數群組相容的資料庫參數群組系列名稱。
- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

  提供資料庫參數群組的名稱。
- Description (在 CLI 中：`--description`或在 Python 中：`description`) -UTF-8 字符串。

  提供此資料庫參數群組的客戶指定描述。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (p. 1164)
- 資料庫ParameterGroupQuotaExceededFault (p. 1164)

複製資料庫ClusterParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`copy-db-cluster-parameter-group`，Python 的名稱是：`copy_db_cluster_parameter_group`。

複製指定的資料庫叢集參數群組。

請求

- SourceDBClusterParameterGroupIdentifier (在 CLI 中：`--source-db-cluster-parameter-group-identifier`或在 Python 中：`source_db_cluster_parameter_group_identifier`) -必需：UTF-8 字符串。

  來源資料庫叢集參數群組的識別符或 Amazon Resource Name (ARN)。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。

約束:

- 必須指定有效的資料庫叢集參數群組。
- 如果來源資料庫叢集和副本位於相同的 Amazon 區域，請指定有效的資料庫參數群組識別符。`my-db-cluster-param-group`
- 如果來源資料庫參數群組和副本位於不同的 Amazon 區域，請指定有效的資料庫叢集參數群組的名稱`arn:aws:rds:us-east-1:123456789012:cluster-pg:custom-cluster-group1`。

- Tags (在 CLI 中：`--tags`或在 Python 中：`tags`) -Tag (p. 1150) 標籤對象的數組。

  要指派給所複製資料庫叢集參數群組的標籤。
- TargetDBClusterParameterGroupDescription (在 CLI 中：`--target-db-cluster-parameter-group-description`或在 Python 中：`target_db_cluster_parameter_group_description`) -必需：UTF-8 字符串。

  所複製資料庫叢集參數群組的描述。
- TargetDBClusterParameterGroupIdentifier (在 CLI 中：`--target-db-cluster-parameter-group-identifier`或在 Python 中：`target_db_cluster_parameter_group_identifier`) -必需：UTF-8 字符串。

  所複製資料庫叢集參數群組的識別符。

  約束:

  - 不可為 null、空白或留空
  - 必須包含 1 到 255 個字母、數字或連字號
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

  範例：`my-cluster-param-group1`

回應

包含 Amazon Neptune 資料庫叢集參數群組的詳細資訊。

此資料類型在 the section called “描述 BClusterParameterGroups” (p. 1100) 動作中會用來作為回應元素。

- DBClusterParameterGroupArn (在 CLI 中：`--db-cluster-parameter-group-arn`或在 Python 中：`db_cluster_parameter_group_arn`) -UTF-8 字符串。

  資料庫叢集參數群組的 Amazon Resource Name (ARN)。
- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  提供資料庫叢集參數群組的名稱。
- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`) -UTF-8 字符串。

  提供此資料庫叢集參數群組相容的資料庫參數群組系列名稱。
- Description (在 CLI 中：`--description`或者在 Python 中：`description`) -UTF-8 字符串。

  提供該客戶指定之此資料庫叢集參數群組的描述。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)

- 資料庫ParameterGroupQuotaExceededFault (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (p. 1164)

建立資料庫ParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`create-db-parameter-group`，Python 的名稱是：`create_db_parameter_group`。

建立新的資料庫參數群組。

最初會為資料庫執行個體所使用的資料庫引擎，使用預設參數建立資料庫參數群組。若要為任何參數提供自訂值，您必須在使用 ModifyDB 建立群組之後修改群組ParameterGroup。在您建立資料庫參數群組後，您需要使用 ModifyDBInstance 將它與您的資料庫執行個體建立關聯。當您將新的資料庫參數群組與執行中的資料庫執行個體建立關聯時，您需要在不使用容錯移轉的情況下重新開機資料庫執行個體，才能讓新的資料庫參數群組和相關聯的設定產生效果。

> **Important**
>
> 建立資料庫參數群組之後，您至少應等待 5 分鐘，再使用該資料庫參數群組當作預設參數群組，建立第一個資料庫執行個體。這可讓 Amazon Neptune 在將參數群組用來做為新資料庫執行個體的預設前，完成建立動作。這對建立資料庫執行個體預設資料庫時的關鍵參數尤其重要，像是 `character_set_database` 參數所定義之預設資料庫的字元集。您可以使用 Amazon Neptune 主控台的「參數群組」選項或 DescribeDBParameters 命令來驗證您的資料庫參數群組已建立或修改完成。

請求

- DBParameterGroupFamily（在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`）-必需：UTF-8 字符串。

  資料庫參數群組系列名稱。資料庫參數群組可和一個，並且只能和一個資料庫參數群組系列建立關聯，並且只能套用到執行與該資料庫參數群組系列相容資料庫引擎和引擎版本的資料庫執行個體。
- DBParameterGroupName（在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`）-必需：UTF-8 字符串。

  資料庫參數群組的名稱。

  約束:
  - 必須為 1 到 255 個字母、數字或連字號。
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

  > **Note**
  >
  > 此值會以小寫字母字串的形式儲存。
- Description（在 CLI 中：`--description`或在 Python 中：`description`）-必需：UTF-8 字符串。

  資料庫參數群組的描述。
- Tags（在 CLI 中：`--tags`或在 Python 中：`tags`）-Tag (p. 1150) 標籤對象的數組。

  要指派給新資料庫參數群組的標籤。

回應

包含 Amazon Neptune 資料庫參數群組的詳細資訊。

此資料類型在 the section called “描述 BParameterGroups” (p. 1098) 動作中會用來作為回應元素。

- DBParameterGroupArn (在 CLI 中：`--db-parameter-group-arn`或者在 Python 中：`db_parameter_group_arn`) -UTF-8 字符串。

  資料庫參數群組的 Amazon Resource Name (ARN)。
- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或者在 Python 中：`db_parameter_group_family`) -UTF-8 字符串。

  提供此資料庫參數群組相容的資料庫參數群組系列名稱。
- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或者在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

  提供資料庫參數群組的名稱。
- Description (在 CLI 中：`--description`或者在 Python 中：`description`) -UTF-8 字符串。

  提供此資料庫參數群組的客戶指定描述。

錯誤

- 資料庫ParameterGroupQuotaExceededFault (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (p. 1164)

建立資料庫ClusterParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`create-db-cluster-parameter-group`，Python 的名稱是：`create_db_cluster_parameter_group`。

建立新的資料庫叢集參數群組。

資料庫叢集參數群組中的參數會套用到資料庫叢集中的所有執行個體。

最初會為資料庫叢集中執行個體所使用的資料庫引擎，使用預設參數建立資料庫叢集參數群組。若要為任何參數提供自訂值，您必須在使用 the section called “修改資料庫ClusterParameterGroup” (p. 1094) 建立它後修改群組。在您建立資料庫叢集參數群組後，您需要使用 the section called “ModifyDBCluster” (p. 1017) 將它與您的資料庫叢集建立關聯。當您將新的資料庫叢集參數群組與執行中的資料庫叢集建立關聯時，您需要在不使用容錯移轉的情況下重新開機資料庫叢集中的資料庫執行個體，才能讓新的資料庫叢集參數群組和相關聯的設定產生效果。

> **Important**
>
> 建立資料庫叢集參數群組後，您應等待至少 5 分鐘，然後再建立第一個使用該資料庫叢集參數群組做為預設參數群組的資料庫叢集。這可讓 Amazon Neptune 在將資料庫叢集參數群組用來做為新資料庫叢集的預設前，完成建立動作。這對建立資料庫叢集預設資料庫時的關鍵參數尤其重要，像是 `character_set_database` 參數所定義的預設資料庫字元集。您可以使用 Amazon Neptune 主控台的參數群組選項或命the section called “描述 BClusterParameters” (p. 1099)令來確認資料庫叢集參數群組是否已建立或修改。

請求

- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`) -必需：UTF-8 字符串。

  資料庫叢集參數群組的名稱。

  約束:
  - 限與現有名稱的名稱相符ClusterParameterGroup。

Note

此值會以小寫字母字串的形式儲存。

- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`) -必需：UTF-8 字符串。

  資料庫叢集參數群組的系列名稱。資料庫叢集參數群組可和一個，並且只能和一個資料庫叢集參數群組系列建立關聯，並且只能套用到執行與該資料庫叢集參數群組系列相容資料庫引擎和引擎版本的資料庫叢集。
- Description (在 CLI 中：`--description`或在 Python 中：`description`) -必需：UTF-8 字符串。

  資料庫叢集參數群組的描述。
- Tags (在 CLI 中：`--tags`或在 Python 中：`tags`) -Tag (p. 1150) 標籤對象的數組。

  要指派給新資料庫叢集參數群組的標籤。

回應

包含 Amazon Neptune 資料庫叢集參數群組的詳細資訊。

此資料類型在 the section called “描述 BClusterParameterGroups” (p. 1100) 動作中會用來作為回應元素。

- DBClusterParameterGroupArn (在 CLI 中：`--db-cluster-parameter-group-arn`或者在 Python 中：`db_cluster_parameter_group_arn`) -UTF-8 字符串。

  資料庫叢集參數群組的 Amazon Resource Name (ARN)。
- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  提供資料庫叢集參數群組的名稱。
- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或者在 Python 中：`db_parameter_group_family`) -UTF-8 字符串。

  提供此資料庫叢集參數群組相容的資料庫參數群組系列名稱。
- Description (在 CLI 中：`--description`或者在 Python 中：`description`) -UTF-8 字符串。

  提供該客戶指定之此資料庫叢集參數群組的描述。

錯誤

- 資料庫ParameterGroupQuotaExceededFault (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (p. 1164)

刪除資料庫ParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`delete-db-parameter-group`，Python 的名稱是：`delete_db_parameter_group`。

刪除指定的數據庫ParameterGroup。ParameterGroup 要刪除的資料庫不能與任何資料庫執行個體相關聯。

請求

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`) -必需：UTF-8 字符串。

資料庫參數群組的名稱。

約束:
- 必須是現有資料庫參數群組的名稱
- 您無法刪除預設資料庫參數群組
- 不可和任何資料庫執行個體建立關聯

回應

- 無回應參數。

錯誤

- 无效ParameterGroupStateFault (p. 1170)
- 資料庫ParameterGroupNotFoundFault (p. 1164)

刪除資料庫ClusterParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`delete-db-cluster-parameter-group`，Python 的名稱是：`delete_db_cluster_parameter_group`。

刪除指定的資料庫叢集參數群組。要刪除的資料庫叢集參數群組不可和任何資料庫叢集建立關聯。

請求

- DBClusterParameterGroupName（在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`）-必需：UTF-8 字符串。

  資料庫叢集參數群組的名稱。

  約束:
  - 必須是現有資料庫叢集參數群組的名稱。
  - 您無法刪除預設資料庫叢集參數群組。
  - 不可和任何資料庫叢集建立關聯

回應

- 無回應參數。

錯誤

- 无效ParameterGroupStateFault (p. 1170)
- 資料庫ParameterGroupNotFoundFault (p. 1164)

修改資料庫ParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-parameter-group`，Python 的名稱是：`modify_db_parameter_group`。

修改資料庫參數群組的參數。若要修改超過一個參數，請提交下列項目的清單：`ParameterName`、`ParameterValue` 和 `ApplyMethod`。單一請求中最多可修改 20 個參數。

Note

動態參數的變更會立即套用。對靜態參數進行的變更，需要在不使用容錯移轉至與參數群組關聯資料庫執行個體的情況下重新開機，才能生效。

Important

修改資料庫參數群組之後，您應該等待至少5分鐘，然後再建立第一個使用該資料庫參數群組做為預設參數群組的資料庫執行個體。這可讓 Amazon Neptune 在將參數群組用來做為新資料庫執行個體的預設前，完成修改動作。這對建立資料庫執行個體預設資料庫時的關鍵參數尤其重要，像是 `character_set_database` 參數所定義之預設資料庫的字元集。您可以使用 Amazon Neptune 主控台的「參數群組」選項或 DescribeDBParameters 命令來驗證您的資料庫參數群組已建立或修改完成。

請求

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`) -必需：UTF-8 字符串。

  資料庫參數群組的名稱。

  約束:

  - 如果提供的名稱，必須符現有名稱的名稱ParameterGroup。
- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -必需：參數 (p. 1085)參數對象的數組。

  參數名稱、值，以及參數更新套用方法的陣列。必須至少提供一個參數名稱、值及套用方法；後續引數都是選用的。單一請求中最多可修改 20 個參數。

  有效值 (針對套用方法)：`immediate | pending-reboot`

  Note

  您只能在搭配動態參數時使用 immediate 值。您可以針對動態和靜態參數使用 pending-reboot 值，並且變更會在您不使用容錯移轉重新開機資料庫執行個體時套用。

回應

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或者在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

  提供資料庫參數群組的名稱。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)
- 无效ParameterGroupStateFault (p. 1170)

修改資料庫ClusterParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-cluster-parameter-group`，Python 的名稱是：`modify_db_cluster_parameter_group`。

修改資料庫叢集參數群組的參數。若要修改超過一個參數，請提交下列項目的清單：`ParameterName`、`ParameterValue` 和 `ApplyMethod`。單一請求中最多可修改 20 個參數。

Note

動態參數的變更會立即套用。對靜態參數進行的變更，需要在不使用容錯移轉至與參數群組關聯資料庫叢集的情況下重新開機，才能生效。

Important

建立資料庫叢集參數群組後，您應等待至少 5 分鐘，然後再建立第一個使用該資料庫叢集參數群組做為預設參數群組的資料庫叢集。這可讓 Amazon Neptune 在將參數群組用來做為新資料庫叢集的預設前，完成建立動作。這對建立資料庫叢集預設資料庫時的關鍵參數尤其重要，像是 `character_set_database` 參數所定義的預設資料庫字元集。您可以使用 Amazon Neptune 主控台的「參數群組」選項或 the section called “描述 BClusterParameters” (p. 1099) 命令來驗證您的資料庫叢集參數群組已建立或修改完成。

請求

- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`) -必需：UTF-8 字符串。

  欲修改資料庫叢集參數群組的名稱。
- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -必需：參數 (p. 1085)參數對象的數組。

  欲修改資料庫叢集參數群組中參數的清單。

回應

- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  資料庫叢集參數群組的名稱。

  約束:
  - 必須為 1 到 255 個字母或數字。
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

    Note

    此值會以小寫字母字串的形式儲存。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)
- 无效ParameterGroupStateFault (p. 1170)

重設資料庫ParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`reset-db-parameter-group`，Python 的名稱是：`reset_db_parameter_group`。

將資料庫參數群組的參數修改為引擎/系統的預設值。若要重設特定參數，請提供下列項目的清單：`ParameterName` 和 `ApplyMethod`。若要重設整個資料庫參數群組，請指定 `DBParameterGroup` 名稱和 `ResetAllParameters` 參數。重設整個群組時，動態參數會立即更新，靜態參數則會設為 `pending-reboot`，並在下一次資料庫執行個體重新啟動或 `RebootDBInstance` 請求時生效。

請求

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`) -必需：UTF-8 字符串。

  資料庫參數群組的名稱。

  約束:

  - 限與現有名稱的名稱相符ParameterGroup。
- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -參數 (p. 1085) 參數對象的數組。

  若要重設整個資料庫參數群組，請指定 `DBParameterGroup` 名稱和 `ResetAllParameters` 參數。若要重設特定參數，請提供下列項目的清單：`ParameterName` 和 `ApplyMethod`。單一請求中最多可修改 20 個參數。

  有效值 (針對套用方法)：`pending-reboot`
- ResetAllParameters (在 CLI 中：`--reset-all-parameters`或在 Python 中：`reset_all_parameters`) -布爾值。

  指定是 (`true`) 否 (`false`) 要將資料庫參數群組中的所有參數重設為預設值。

  預設：`true`

回應

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或者在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

  提供資料庫參數群組的名稱。

錯誤

- 无效ParameterGroupStateFault (p. 1170)
- 資料庫ParameterGroupNotFoundFault (p. 1164)

重設資料庫ClusterParameterGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`reset-db-cluster-parameter-group`，Python 的名稱是：`reset_db_cluster_parameter_group`。

將資料庫叢集參數群組的參數修改為預設值。若要重設特定參數，請提交下列項目的清單：`ParameterName` 和 `ApplyMethod`。若要重設整個資料庫叢集參數群組，請指定 `DBClusterParameterGroupName` 名稱和 `ResetAllParameters` 參數。

重設整個群組時，動態參數會立即更新，靜態參數則會設為 `pending-reboot`，並在下一次資料庫執行個體重新啟動或 the section called “RebootDBInstance” (p. 1078) 請求時生效。您必須為您欲套用更新後靜態參數資料庫叢集中的每個資料庫執行個體呼叫 the section called “RebootDBInstance” (p. 1078)。

請求

- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或在 Python 中：`db_cluster_parameter_group_name`) -必需：UTF-8 字符串。

  要重設的資料庫叢集參數群組名稱。

- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -參數 (p. 1085) 參數對象的數組。

  資料庫叢集參數群組中的參數名稱清單，會將此清單上的參數重設為其預設值。若將 `ResetAllParameters` 參數設為 `true`，您便無法使用此參數。
- ResetAllParameters (在 CLI 中：`--reset-all-parameters`或在 Python 中：`reset_all_parameters`) -布爾值。

  會在將資料庫叢集參數群組中所有參數重設為其預設值時設為 `true` 的值，否則為 `false`。若有為 `Parameters` 參數指定參數名稱清單，您便無法使用此參數。

回應

- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  資料庫叢集參數群組的名稱。

  約束:
  - 必須為 1 到 255 個字母或數字。
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

  Note

  此值會以小寫字母字串的形式儲存。

錯誤

- 无效ParameterGroupStateFault (p. 1170)
- 資料庫ParameterGroupNotFoundFault (p. 1164)

DescribeDBParameters (動作)

這個 API 的亞馬遜 CLI 名稱是：`describe-db-parameters`，Python 的名稱是：`describe_db_parameters`。

傳回特定資料庫參數群組的詳細參數清單。

請求

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或在 Python 中：`db_parameter_group_name`) -必需：UTF-8 字符串。

  要傳回其詳細資訊的特定資料庫參數群組名稱。

  約束:
  - 如果提供的名稱，必須符現有名稱的名稱ParameterGroup。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個 `DescribeDBParameters` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- Source (在 CLI 中：`--source`或者在 Python 中：`source`) -UTF-8 字符串。

  要傳回的參數類型。

  預設：傳回所有參數類型。

  有效值: `user | system | engine-default`

回應

- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -參數 (p. 1085) 參數對象的數組。

  the section called “參數” (p. 1085) 值的清單。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)

描述 BParameterGroups （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-db-parameter-groups`，Python 的名稱是：`describe_db_parameter_groups`。

傳回 `DBParameterGroup` 描述的清單。若有指定 `DBParameterGroupName`，則清單只會包含指定資料庫參數群組的描述。

請求

- DBParameterGroupName (在 CLI 中：`--db-parameter-group-name`或者在 Python 中：`db_parameter_group_name`) -UTF-8 字符串。

  要傳回其詳細資訊的特定資料庫參數群組名稱。

  約束:
  - 如果提供的名稱，必須符現有名稱的名稱ClusterParameterGroup。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個 `DescribeDBParameterGroups` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

要在回應中包含的記錄數量上限。若存在的記錄比 MaxRecords 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

預設：100

限制條件：最小 20，最大 100。

回應

- DBParameterGroups（在 CLI 中：--db-parameter-groups或在 Python 中：db_parameter_groups）-數資料庫ParameterGroup (p. 1086)據庫ParameterGroup 對象的數組。

  the section called “資料庫ParameterGroup” (p. 1086) 執行個體的清單。
- Marker（在 CLI 中：--marker或者在 Python 中：marker）-UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)

描述 BClusterParameters （動作）

這個 API 的亞馬遜 CLI 名稱是：describe-db-cluster-parameters，Python 的名稱是：describe_db_cluster_parameters。

傳回特定資料庫叢集參數群組的詳細參數清單。

請求

- DBClusterParameterGroupName（在 CLI 中：--db-cluster-parameter-group-name或在 Python 中：db_cluster_parameter_group_name）-必需：UTF-8 字符串。

  要傳回其參數詳細資訊的特定資料庫叢集參數群組名稱。

  約束:
  - 如果提供的名稱，必須符現有名稱的名稱ClusterParameterGroup。
- Filters（在 CLI 中：--filters，或在 Python 中：filters）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker（在 CLI 中：--marker或者在 Python 中：marker）-UTF-8 字符串。

  前一個 DescribeDBClusterParameters 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。
- MaxRecords（在 CLI 中：--max-records或在 Python 中：max_records）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 MaxRecords 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- Source（在 CLI 中：--source或者在 Python 中：source）-UTF-8 字符串。

指出只傳回特定來源參數的值。參數來源可以是 engine、service 或 customer。

回應

- Marker（在 CLI 中：--marker或者在 Python 中：marker）-UTF-8 字符串。

  由先前的 descriBedBClusterParameters 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。
- Parameters（在 CLI 中：--parameters或在 Python 中：parameters）-參數 (p. 1085) 參數對象的數組。

  提供資料庫叢集參數群組的參數清單。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)

描述 BClusterParameterGroups （動作）

這個 API 的亞馬遜 CLI 名稱是：describe-db-cluster-parameter-groups，Python 的名稱是：describe_db_cluster_parameter_groups。

傳回 DBClusterParameterGroup 描述的清單。若有指定 DBClusterParameterGroupName 參數，則清單只會包含指定資料庫叢集參數群組的描述。

請求

- DBClusterParameterGroupName（在 CLI 中：--db-cluster-parameter-group-name或者在 Python 中：db_cluster_parameter_group_name）-UTF-8 字符串。

  要傳回其詳細資訊的特定資料庫叢集參數群組名稱。

  約束:
  - 如果提供的名稱，必須符現有名稱的名稱ClusterParameterGroup。
- Filters（在 CLI 中：--filters，或在 Python 中：filters）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker（在 CLI 中：--marker或者在 Python 中：marker）-UTF-8 字符串。

  前一個 DescribeDBClusterParameterGroups 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。
- MaxRecords（在 CLI 中：--max-records或在 Python 中：max_records）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 MaxRecords 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

- DBClusterParameterGroups（在 CLI 中：`--db-cluster-parameter-groups`或在 Python 中：`db_cluster_parameter_groups`）-數資料庫ClusterParameterGroup (p. 1086)據庫 ClusterParameterGroup 對象的數組。

  資料庫叢集參數群組的清單。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 `DescribeDBClusterParameterGroups` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

錯誤

- 資料庫ParameterGroupNotFoundFault (p. 1164)

DescribeEngineDefaultParameters（動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-engine-default-parameters`，Python 的名稱是：`describe_engine_default_parameters`。

傳回指定資料庫引擎的預設引擎和系統參數資訊。

請求

- DBParameterGroupFamily（在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`）-必需：UTF-8 字符串。

  資料庫參數群組系列的名稱。
- Filters（在 CLI 中：`--filters`，或在 Python 中：`filters`）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 `DescribeEngineDefaultParameters` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords（在 CLI 中：`--max-records`或在 Python 中：`max_records`）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

包含成功呼叫 the section called “DescribeEngineDefaultParameters” (p. 1101) 動作的結果。

- DBParameterGroupFamily（在 CLI 中：`--db-parameter-group-family`或者在 Python 中：`db_parameter_group_family`）-UTF-8 字符串。

  指定要套用引擎預設參數的資料庫參數群組系列名稱。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  先前 EngineDefaults 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -參數 (p. 1085) 參數對象的數組。

  包含引擎預設參數的清單。

DescribeEngineDefaultClusterParameters （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-engine-default-cluster-parameters`，Python 的名稱是：`describe_engine_default_cluster_parameters`。

傳回叢集資料庫引擎的預設引擎和系統參數資訊。

請求

- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或在 Python 中：`db_parameter_group_family`) -必需：UTF-8 字符串。

  要傳回其引擎參數資訊的資料庫叢集參數群組系列名稱。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個 `DescribeEngineDefaultClusterParameters` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

包含成功呼叫 the section called “DescribeEngineDefaultParameters” (p. 1101) 動作的結果。

- DBParameterGroupFamily (在 CLI 中：`--db-parameter-group-family`或者在 Python 中：`db_parameter_group_family`) -UTF-8 字符串。

  指定要套用引擎預設參數的資料庫參數群組系列名稱。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  先前 EngineDefaults 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- Parameters (在 CLI 中：`--parameters`或在 Python 中：`parameters`) -參數 (p. 1085) 參數對象的數組。

  包含引擎預設參數的清單。

Neptune 子網路 API

結構：

- Subnet (結構) (p. 1103)
- 資料庫SubnetGroup (結構) (p. 1103)

動作:

- 建立資料庫SubnetGroup (動作) (p. 1104)
- 刪除資料庫SubnetGroup (動作) (p. 1105)
- 修改資料庫SubnetGroup (動作) (p. 1106)
- 描述 BSubnetGroups （ 動作 ） (p. 1107)

Subnet (結構)

指定子網路。

此資料類型在 the section called “描述 BSubnetGroups” (p. 1107) 動作中會用來作為回應元素。

欄位

- SubnetAvailabilityZone – AvailabilityZone (p. 1156) 物件。

  指定子網路所在的 EC2 可用區域。
- SubnetIdentifier – UTF-8 字串。

  指定子網路的識別符。
- SubnetStatus – UTF-8 字串。

  指定子網路的狀態。

資料庫SubnetGroup (結構)

包含 Amazon Neptune 子網路群組的詳細資訊。

此資料類型在 the section called “描述 BSubnetGroups” (p. 1107) 動作中會用來作為回應元素。

欄位

- DBSubnetGroupArn – UTF-8 字串。

  資料庫子網路群組的 Amazon Resource Name (ARN)。
- DBSubnetGroupDescription – UTF-8 字串。

  提供資料庫子網路群組的描述。
- DBSubnetGroupName – UTF-8 字串。

  資料庫子網路群組的名稱。
- SubnetGroupStatus – UTF-8 字串。

  提供資料庫子網路群組的狀態。
- Subnets— 子子網 (p. 1103)網路物件的陣列。

  包含 the section called “子網” (p. 1103) 元素的清單。
- VpcId – UTF-8 字串。

提供資 VpcId 料庫子網路群組。

DBSubnetGroup 會用來做為以下項目的回應元素：

- 創建數據庫SubnetGroup (p. 1104)
- 修改資料庫SubnetGroup (p. 1106)

建立資料庫SubnetGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：create-db-subnet-group，Python 的名稱是：create_db_subnet_group。

建立新的資料庫子網路群組。資料庫子網路群組在 Amazon 區域中，須至少於兩個可用區域中設有至少一個子網路。

請求

- DBSubnetGroupDescription（在 CLI 中：--db-subnet-group-description或在 Python 中：db_subnet_group_description）-必需：UTF-8 字符串。

  資料庫子網路群組的描述。
- DBSubnetGroupName（在 CLI 中：--db-subnet-group-name或在 Python 中：db_subnet_group_name）-必需：UTF-8 字符串。

  資料庫子網路群組的名稱。此值會以小寫字母字串的形式儲存。

  限制條件：包含的內容絕不能超過 255 個字母、數字、句號、底線、空格或連字號。絕不能為預設值。

  範例：mySubnetgroup
- SubnetIds（在 CLI 中：--subnet-ids或在 Python 中：subnet_ids）-需要：一個數組 SubnetIdentifier-UTF-8 字符串。

  資料庫子網路群組的 EC2 子網路 ID。
- Tags（在 CLI 中：--tags或在 Python 中：tags）-Tag (p. 1150) 標籤對象的數組。

  要指派給新資料庫子網路群組的標籤。

回應

包含 Amazon Neptune 子網路群組的詳細資訊。

此資料類型在 the section called “描述 BSubnetGroups” (p. 1107) 動作中會用來作為回應元素。

- DBSubnetGroupArn（在 CLI 中：--db-subnet-group-arn或者在 Python 中：db_subnet_group_arn）-UTF-8 字符串。

  資料庫子網路群組的 Amazon Resource Name (ARN)。
- DBSubnetGroupDescription（在 CLI 中：--db-subnet-group-description或者在 Python 中：db_subnet_group_description）-UTF-8 字符串。

  提供資料庫子網路群組的描述。
- DBSubnetGroupName（在 CLI 中：--db-subnet-group-name或者在 Python 中：db_subnet_group_name）-UTF-8 字符串。

  資料庫子網路群組的名稱。

- SubnetGroupStatus (在 CLI 中：`--subnet-group-status`或者在 Python 中：`subnet_group_status`) -UTF-8 字符串。

  提供資料庫子網路群組的狀態。
- Subnets (在 CLI 中：`--subnets`，或在 Python 中：`subnets`) -子子網 (p. 1103)網對象的數組。

  包含 the section called “子網” (p. 1103) 元素的清單。
- VpcId (在 CLI 中：`--vpc-id`或者在 Python 中：`vpc_id`) -UTF-8 字符串。

  提供資 VpcId 料庫子網路群組。

錯誤

- 資料庫SubnetGroupAlreadyExistsFault (p. 1166)
- 資料庫SubnetGroupQuotaExceededFault (p. 1167)
- 資料庫SubnetQuotaExceededFault (p. 1167)
- 資料庫SubnetGroupDoesNotCoverEnough可用區 (p. 1166)
- InvalidSubnet (p. 1173)

刪除資料庫SubnetGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`delete-db-subnet-group`，Python 的名稱是：`delete_db_subnet_group`。

刪除資料庫子網路群組。

> Note
>
> 指定的資料庫子網路群組不能和任何資料庫執行個體建立關聯。

請求

- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或在 Python 中：`db_subnet_group_name`) -必需：UTF-8 字符串。

  要刪除的資料庫子網路群組名稱。

  > Note
  >
  > 您無法刪除預設子網路群組。

  約束:

  限制條件：必須符合現有資料庫的名稱SubnetGroup。絕不能為預設值。

  範例：`mySubnetgroup`

回應

- 無回應參數。

錯誤

- 无效SubnetGroupStateFault (p. 1171)

- 无效SubnetStateFault (p. 1172)
- 資料庫SubnetGroupNotFoundFault (p. 1167)

修改資料庫SubnetGroup (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-subnet-group`，Python 的名稱是：`modify_db_subnet_group`。

修改現有的資料庫子網路群組。資料庫子網路群組在 Amazon 區域中，須至少於兩個可用區域中設有至少一個子網路。

請求

- DBSubnetGroupDescription (在 CLI 中：`--db-subnet-group-description`或者在 Python 中：`db_subnet_group_description`) -UTF-8 字符串。

  資料庫子網路群組的描述。
- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或在 Python 中：`db_subnet_group_name`) -必需：UTF-8 字符串。

  資料庫子網路群組的名稱。此值會以小寫字母字串的形式儲存。您無法修改預設子網路群組。

  限制條件：必須符合現有資料庫的名稱SubnetGroup。絕不能為預設值。

  範例：`mySubnetgroup`
- SubnetIds (在 CLI 中：`--subnet-ids`或在 Python 中：`subnet_ids`) -需要：一個數組SubnetIdentifier-UTF-8 字符串。

  資料庫子網路群組的 EC2 子網路 ID。

回應

包含 Amazon Neptune 子網路群組的詳細資訊。

此資料類型在 the section called “描述 BSubnetGroups” (p. 1107) 動作中會用來作為回應元素。

- DBSubnetGroupArn (在 CLI 中：`--db-subnet-group-arn`或者在 Python 中：`db_subnet_group_arn`) -UTF-8 字符串。

  資料庫子網路群組的 Amazon Resource Name (ARN)。
- DBSubnetGroupDescription (在 CLI 中：`--db-subnet-group-description`或者在 Python 中：`db_subnet_group_description`) -UTF-8 字符串。

  提供資料庫子網路群組的描述。
- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`) -UTF-8 字符串。

  資料庫子網路群組的名稱。
- SubnetGroupStatus (在 CLI 中：`--subnet-group-status`或者在 Python 中：`subnet_group_status`) -UTF-8 字符串。

  提供資料庫子網路群組的狀態。
- Subnets (在 CLI 中：`--subnets`，或在 Python 中：`subnets`) -子子網 (p. 1103)網對象的數組。

  包含 the section called “子網” (p. 1103) 元素的清單。

- VpcId (在 CLI 中：`--vpc-id`或者在 Python 中：`vpc_id`) -UTF-8 字符串。

  提供資 VpcId 料庫子網路群組。

錯誤

- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 資料庫SubnetQuotaExceededFault (p. 1167)
- SubnetAlreadyInUse (p. 1176)
- 資料庫SubnetGroupDoesNotCoverEnough可用區 (p. 1166)
- InvalidSubnet (p. 1173)

描述 BSubnetGroups （ 動作 ）

這個 API 的亞馬遜 CLI 名稱是：`describe-db-subnet-groups`，Python 的名稱是：`describe_db_subnet_groups`。

返回數據庫SubnetGroup 描述的列表。如果指定了一個數據庫SubnetGroupName，列表將只包含指定數據庫的描述SubnetGroup。

如需 CIDR 範圍的概觀，請前往 Wikipedia Tutorial。

請求

- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`) -UTF-8 字符串。

  要傳回其詳細資訊的資料庫子網路群組名稱。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  由先前的 descriBedBSubnetGroups 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

- DBSubnetGroups (在 CLI 中：`--db-subnet-groups`或在 Python 中：`db_subnet_groups`) -數資料庫SubnetGroup (p. 1103)據庫SubnetGroup 對象的數組。

  the section called “資料庫SubnetGroup” (p. 1103) 執行個體的清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

錯誤

- 資料庫SubnetGroupNotFoundFault (p. 1167)

Neptune 快照 API

結構：

- 資料庫 ClusterSnapshot (結構) (p. 1108)
- 資料庫 ClusterSnapshotAttribute (結構) (p. 1110)
- 資料庫 ClusterSnapshotAttributesResult (結構) (p. 1110)

動作:

- 建立資料庫 ClusterSnapshot (動作) (p. 1110)
- 刪除資料庫 ClusterSnapshot (動作) (p. 1113)
- 複製資料庫 ClusterSnapshot (動作) (p. 1115)
- 修改資料庫 ClusterSnapshotAttribute (動作) (p. 1117)
- 復原 B (動ClusterFromSnapshot作) (p. 1119)
- 復原 B (動ClusterToPointInTime作) (p. 1125)
- 描述 B (動作ClusterSnapshots) (p. 1131)
- 描述 B (動作ClusterSnapshotAttributes) (p. 1133)

資料庫 ClusterSnapshot (結構)

包含 Amazon Neptune 資料庫叢集快照的詳細資訊

此資料類型在 the section called “描述 B ClusterSnapshots” (p. 1131) 動作中會用來作為回應元素。

欄位

- AllocatedStorage – 數字 (整數)。

  以 GiB 為單位指定分配的儲存體大小。
- AvailabilityZonesUTF-8 字串陣陣列陣列AvailabilityZone陣列陣列陣列陣列陣列

  提供 EC2 可用區域清單，您可以在此還原資料庫叢集快照中的執行個體。
- ClusterCreateTime – 時間戳記。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- DBClusterIdentifier – UTF-8 字串。

  指定資料庫叢集的資料庫叢集識別符，此即建立此資料庫叢集快照的資料庫叢集。
- DBClusterSnapshotArn – UTF-8 字串。

  資料庫叢集快照的 Amazon Resource Name (ARN)。
- DBClusterSnapshotIdentifier – UTF-8 字串。

  指定資料庫叢集快照的識別碼。必須符合現有快照的識別碼。

使用 DBClusterSnapshotIdentifier 還原資料庫叢集之後，您必須為資料庫叢集未來的任何更新指定相同的 DBClusterSnapshotIdentifier。當您為更新指定此屬性時，資料庫叢集不會再次從快照還原，且資料庫中的資料不會變更。

不過，如果您未指定 DBClusterSnapshotIdentifier，則會建立空的資料庫叢集，並刪除原始資料庫叢集。如果您指定的屬性與先前的快照還原屬性不同，則會從 DBClusterSnapshotIdentifier 指定的快照還原資料庫叢集，並刪除原始資料庫叢集。

- Engine – UTF-8 字串。

  指定資料庫引擎的名稱。
- EngineVersion – UTF-8 字串。

  提供此資料庫叢集快照的資料庫引擎版本。
- IAMDatabaseAuthenticationEnabled – 布林值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId – UTF-8 字串。

  若StorageEncrypted為 true，則為 true，則為加密資料庫叢集快照的 Amazon KMS 金鑰識別符。
- LicenseModel – UTF-8 字串。

  提供此資料庫叢集快照的授權模式資訊。
- PercentProgress – 數字 (整數)。

  指定已傳輸的估計資料百分比。
- Port – 數字 (整數)。

  指定資料庫叢集在快照時接聽的連接埠。
- SnapshotCreateTime – 時間戳記。

  提供快照拍攝的時間，使用供國際標準時間 (UTC)。
- SnapshotType – UTF-8 字串。

  提供資料庫叢集快照的類型。
- SourceDBClusterSnapshotArn – UTF-8 字串。

  如果資料庫叢集快照複製自來源資料庫叢集快照，則為來源資料庫叢集快照的 Amazon Resource Name (ARN)，否則為空值。
- Status – UTF-8 字串。

  指定此資料庫叢集快照的狀態。
- StorageEncrypted – 布林值。

  指定資料庫叢集快照是否加密。
- VpcId – UTF-8 字串。

  提供與資料庫叢集快照關聯的 VPC ID。

DBClusterSnapshot 會用來做為以下項目的回應元素：

- 創建數據庫 ClusterSnapshot (p. 1110)
- 複製資料庫 ClusterSnapshot (p. 1115)
- 刪除資料庫 ClusterSnapshot (p. 1113)

資料庫 ClusterSnapshotAttribute (結構)

包含手動資料庫叢集快照屬性的名稱和值。

手動資料庫叢集快照屬性可用來授權其他 Amazon 帳戶還原手動資料庫叢集快照。如需詳細資訊，請參閱 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。

欄位

- AttributeName – UTF-8 字串。

  手動資料庫叢集快照屬性的名稱。

  名為的屬性`restore`是指具有複製或還原手動資料庫叢集快照之權限的 Amazon 帳戶清單。如需詳細資訊，請參閱 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。
- AttributeValuesUTF-8 字串陣陣列陣列AttributeValue陣列陣列陣列陣列陣

  手動資料庫叢集快照屬性的值。

  如果`AttributeName`欄位設定為`restore`，則此元素會傳回授權複製或還原手動資料庫叢集快照的 Amazon 帳戶 ID 清單。如果清單中`all`有的值，則手動資料庫叢集快照為公用，並可供任何 Amazon 帳戶複製或還原。

資料庫 ClusterSnapshotAttributesResult (結構)

包含成功呼叫 the section called “描述 B ClusterSnapshotAttributes” (p. 1133) API 動作的結果。

手動資料庫叢集快照屬性用於授權其他 Amazon 帳戶複製或還原手動資料庫叢集快照。如需詳細資訊，請參閱 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。

欄位

- DBClusterSnapshotAttributes— 資料庫 ClusterSnapshotAttribute (p. 1110) DB ClusterSnapshotAttribute 物件的陣列。

  手動資料庫叢集快照的屬性和值清單。
- DBClusterSnapshotIdentifier – UTF-8 字串。

  屬性套用目標之手動資料庫叢集快照的識別符。

`DBClusterSnapshotAttributesResult` 會用來做為以下項目的回應元素：

- 描述 B ClusterSnapshotAttributes (p. 1133)
- 修改資料庫 ClusterSnapshotAttribute (p. 1117)

建立資料庫 ClusterSnapshot (動作)

這個 API 的亞馬遜 CLI 名稱是：`create-db-cluster-snapshot`，Python 的名稱是：`create_db_cluster_snapshot`。

建立資料庫叢集的快照。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

要建立快照之資料庫叢集的識別符。此參數不區分大小寫。

約束:

- 必須符合現有 DBCluster 的識別碼。

範例：`my-cluster1`

- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或在 Python 中：`db_cluster_snapshot_identifier`）-必需：UTF-8 字符串。

  資料庫叢集快照的識別符。此參數是以小寫字母字串的形式儲存。

  約束:

  - 必須包含 1 到 63 個字母、數字或連字號。
  - 第一個字元必須是字母。
  - 不能以連字號結尾或連續包含兩個連字號。

  範例：`my-cluster1-snapshot1`
- Tags（在 CLI 中：`--tags`或在 Python 中：`tags`）-Tag (p. 1150) 標籤對象的數組。

  要指派給資料庫叢集快照的標籤。

回應

包含 Amazon Neptune 資料庫叢集快照的詳細資訊

此資料類型在 the section called “描述 B ClusterSnapshots” (p. 1131) 動作中會用來作為回應元素。

- AllocatedStorage（在 CLI 中：`--allocated-storage`或在 Python 中：`allocated_storage`）-數字（整數）。

  以 GiB 為單位指定分配的儲存體大小。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供 EC2 可用區域清單，您可以在此還原資料庫叢集快照中的執行個體。
- ClusterCreateTime（在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`）-時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`）-UTF-8 字符串。

  指定資料庫叢集的資料庫叢集識別符，此即建立此資料庫叢集快照的資料庫叢集。
- DBClusterSnapshotArn（在 CLI 中：`--db-cluster-snapshot-arn`或者在 Python 中：`db_cluster_snapshot_arn`）-UTF-8 字符串。

  資料庫叢集快照的 Amazon Resource Name (ARN)。
- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`）-UTF-8 字符串。

  指定資料庫叢集快照的識別碼。必須符合現有快照的識別碼。

  使用 `DBClusterSnapshotIdentifier` 還原資料庫叢集之後，您必須為資料庫叢集未來的任何更新指定相同的 `DBClusterSnapshotIdentifier`。當您為更新指定此屬性時，資料庫叢集不會再次從快照還原，且資料庫中的資料不會變更。

不過，如果您未指定 DBClusterSnapshotIdentifier，則會建立空的資料庫叢集，並刪除原始資料庫叢集。如果您指定的屬性與先前的快照還原屬性不同，則會從 DBClusterSnapshotIdentifier 指定的快照還原資料庫叢集，並刪除原始資料庫叢集。

- Engine（在 CLI 中：--engine或者在 Python 中：engine）-UTF-8 字符串。

  指定資料庫引擎的名稱。
- EngineVersion（在 CLI 中：--engine-version或者在 Python 中：engine_version）-UTF-8 字符串。

  提供此資料庫叢集快照的資料庫引擎版本。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：--iam-database-authentication-enabled，或者在 Python 中：iam_database_authentication_enabled）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId（在 CLI 中：--kms-key-id或者在 Python 中：kms_key_id）-UTF-8 字符串。

  若StorageEncrypted為 true，則為 true，則為加密資料庫叢集快照的 Amazon KMS 金鑰識別符。
- LicenseModel（在 CLI 中：--license-model或者在 Python 中：license_model）-UTF-8 字符串。

  提供此資料庫叢集快照的授權模式資訊。
- PercentProgress（在 CLI 中：--percent-progress或在 Python 中：percent_progress）-數字（整數）。

  指定已傳輸的估計資料百分比。
- Port（在 CLI 中：--port或在 Python 中：port）-數字（整數）。

  指定資料庫叢集在快照時接聽的連接埠。
- SnapshotCreateTime（在 CLI 中：--snapshot-create-time，或在 Python 中：snapshot_create_time）-時間戳。

  提供快照拍攝的時間，使用供國際標準時間 (UTC)。
- SnapshotType（在 CLI 中：--snapshot-type或者在 Python 中：snapshot_type）-UTF-8 字符串。

  提供資料庫叢集快照的類型。
- SourceDBClusterSnapshotArn（在 CLI 中：--source-db-cluster-snapshot-arn或者在 Python 中：source_db_cluster_snapshot_arn）-UTF-8 字符串。

  如果資料庫叢集快照複製自來源資料庫叢集快照，則為來源資料庫叢集快照的 Amazon Resource Name (ARN)，否則為空值。
- Status（在 CLI 中：--status或者在 Python 中：status）-UTF-8 字符串。

  指定此資料庫叢集快照的狀態。
- StorageEncrypted（在 CLI 中：--storage-encrypted，或者在 Python 中：storage_encrypted）-布爾值。

  指定資料庫叢集快照是否加密。
- VpcId（在 CLI 中：--vpc-id或者在 Python 中：vpc_id）-UTF-8 字符串。

  提供與資料庫叢集快照關聯的 VPC ID。

錯誤

- 資料庫ClusterSnapshotAlreadyExistsFault (p. 1163)

- 无效ClusterStateFault (p. 1170)
- 資料庫ClusterNotFoundFault (p. 1162)
- SnapshotQuotaExceededFault (p. 1175)
- 无效ClusterSnapshotStateFault (p. 1170)

刪除資料庫 ClusterSnapshot (動作)

這個 API 的亞馬遜 CLI 名稱是：`delete-db-cluster-snapshot`，Python 的名稱是：`delete_db_cluster_snapshot`。

刪除資料庫叢集快照。如果正在複製快照，即會終止複製操作。

Note

資料庫叢集快照必須為 `available` 狀態才能刪除。

請求

- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或在 Python 中：`db_cluster_snapshot_identifier`）-必需：UTF-8 字符串。

  要刪除之資料庫叢集快照的識別符。

  限制條件：必須是 `available` 狀態的現有資料庫叢集快照名稱。

回應

包含 Amazon Neptune 資料庫叢集快照的詳細資訊

此資料類型在 the section called “描述 B ClusterSnapshots” (p. 1131) 動作中會用來作為回應元素。

- AllocatedStorage（在 CLI 中：`--allocated-storage`或在 Python 中：`allocated_storage`）-數字（整數）。

  以 GiB 為單位指定分配的儲存體大小。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供 EC2 可用區域清單，您可以在此還原資料庫叢集快照中的執行個體。
- ClusterCreateTime（在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`）-時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`）-UTF-8 字符串。

  指定資料庫叢集的資料庫叢集識別符，此即建立此資料庫叢集快照的資料庫叢集。
- DBClusterSnapshotArn（在 CLI 中：`--db-cluster-snapshot-arn`或者在 Python 中：`db_cluster_snapshot_arn`）-UTF-8 字符串。

  資料庫叢集快照的 Amazon Resource Name (ARN)。
- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`）-UTF-8 字符串。

  指定資料庫叢集快照的識別碼。必須符合現有快照的識別碼。

使用 DBClusterSnapshotIdentifier 還原資料庫叢集之後，您必須為資料庫叢集未來的任何更新指定相同的 DBClusterSnapshotIdentifier。當您為更新指定此屬性時，資料庫叢集不會再次從快照還原，且資料庫中的資料不會變更。

不過，如果您未指定 DBClusterSnapshotIdentifier，則會建立空的資料庫叢集，並刪除原始資料庫叢集。如果您指定的屬性與先前的快照還原屬性不同，則會從 DBClusterSnapshotIdentifier 指定的快照還原資料庫叢集，並刪除原始資料庫叢集。

- Engine（在 CLI 中：--engine或者在 Python 中：engine）-UTF-8 字符串。

  指定資料庫引擎的名稱。

- EngineVersion（在 CLI 中：--engine-version或者在 Python 中：engine_version）-UTF-8 字符串。

  提供此資料庫叢集快照的資料庫引擎版本。

- IAMDatabaseAuthenticationEnabled（在 CLI 中：--iam-database-authentication-enabled，或者在 Python 中：iam_database_authentication_enabled）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId（在 CLI 中：--kms-key-id或者在 Python 中：kms_key_id）-UTF-8 字符串。

  若StorageEncrypted為 true，則為 true，則為加密資料庫叢集快照的 Amazon KMS 金鑰識別符。

- LicenseModel（在 CLI 中：--license-model或者在 Python 中：license_model）-UTF-8 字符串。

  提供此資料庫叢集快照的授權模式資訊。

- PercentProgress（在 CLI 中：--percent-progress或在 Python 中：percent_progress）-數字（整數）。

  指定已傳輸的估計資料百分比。

- Port（在 CLI 中：--port或在 Python 中：port）-數字（整數）。

  指定資料庫叢集在快照時接聽的連接埠。

- SnapshotCreateTime（在 CLI 中：--snapshot-create-time，或在 Python 中：snapshot_create_time）-時間戳。

  提供快照拍攝的時間，使用供國際標準時間 (UTC)。

- SnapshotType（在 CLI 中：--snapshot-type或者在 Python 中：snapshot_type）-UTF-8 字符串。

  提供資料庫叢集快照的類型。

- SourceDBClusterSnapshotArn（在 CLI 中：--source-db-cluster-snapshot-arn或者在 Python 中：source_db_cluster_snapshot_arn）-UTF-8 字符串。

  如果資料庫叢集快照複製自來源資料庫叢集快照，則為來源資料庫叢集快照的 Amazon Resource Name (ARN)，否則為空值。

- Status（在 CLI 中：--status或者在 Python 中：status）-UTF-8 字符串。

  指定此資料庫叢集快照的狀態。

- StorageEncrypted（在 CLI 中：--storage-encrypted，或者在 Python 中：storage_encrypted）-布爾值。

  指定資料庫叢集快照是否加密。

- VpcId（在 CLI 中：--vpc-id或者在 Python 中：vpc_id）-UTF-8 字符串。

  提供與資料庫叢集快照關聯的 VPC ID。

錯誤

- 无效ClusterSnapshotStateFault (p. 1170)
- 資料庫ClusterSnapshotNotFoundFault (p. 1163)

複製資料庫 ClusterSnapshot (動作)

這個 API 的亞馬遜 CLI 名稱是：`copy-db-cluster-snapshot`，Python 的名稱是：`copy_db_cluster_snapshot`。

複製資料庫叢集的快照。

若要從共享的手動資料庫叢集快照複製資料庫叢集快照，`SourceDBClusterSnapshotIdentifier` 必須是共享資料庫叢集快照的 Amazon Resource Name (ARN)。

請求

- CopyTags (在 CLI 中：`--copy-tags`，或者在 Python 中：`copy_tags`) -布爾值。

  True，則將來源資料庫叢集快照的所有標籤複製到目標資料庫叢集快照，否則為 false。預設值為 false。
- KmsKeyId (在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`) -UTF-8 字符串。

  加密資料庫叢集快照的亞馬遜 KMS 金鑰識別碼。KMS 金鑰 ID 是 KMS 加密金鑰的 Amazon Resource Name (ARN)、KMS 金鑰識別符或 KMS 金鑰別名。

  如果您從 Amazon 帳戶複製加密資料庫叢集快照，您可以指定值`KmsKeyId`以使用新的 KMS 加密金鑰來加密副本。如果不指定 `KmsKeyId` 的值，則會使用與來源資料庫叢集快照相同的 KMS 金鑰，加密資料庫叢集快照的副本。

  若您複製另一個 Amazon 帳戶所共用的加密資料庫叢集快照，則必須指定的值`KmsKeyId`。

  KMS 加密金鑰專用於其建立時所在的 Amazon 區域，您無法將某個 Amazon 區域的加密金鑰用在另一個 Amazon 區域。

  複製時，您不能加密未加密的資料庫叢集快照。如果您嘗試複製未加密的資料庫叢集快照並指定KmsKeyId參數值，則會傳回錯誤。
- PreSignedUrl (在 CLI 中：`--pre-signed-url`或者在 Python 中：`pre_signed_url`) -UTF-8 字符串。

  目前不支援。
- SourceDBClusterSnapshotIdentifier (在 CLI 中：`--source-db-cluster-snapshot-identifier`或在 Python 中：`source_db_cluster_snapshot_identifier`) -必需：UTF-8 字符串。

  要複製之資料庫叢集快照的識別符。此參數不區分大小寫。

  約束:
  - 必須指定「可用」狀態下的有效系統快照。
  - 指定有效的資料庫快照識別符。

  範例：`my-cluster-snapshot1`
- Tags (在 CLI 中：`--tags`或在 Python 中：`tags`) -Tag (p. 1150) 標籤對象的數組。

  要指派給新資料庫叢集快照副本的標籤。
- TargetDBClusterSnapshotIdentifier (在 CLI 中：`--target-db-cluster-snapshot-identifier`或在 Python 中：`target_db_cluster_snapshot_identifier`) -必需：UTF-8 字符串。

  要從來源資料庫叢集快照建立之新資料庫叢集快照的識別符。此參數不區分大小寫。

約束:

- 必須包含 1 到 63 個字母、數字或連字號。
- 第一個字元必須是字母。
- 不能以連字號結尾或連續包含兩個連字號。

範例：`my-cluster-snapshot2`

回應

包含 Amazon Neptune 資料庫叢集快照的詳細資訊

此資料類型在 the section called “描述 B ClusterSnapshots” (p. 1131) 動作中會用來作為回應元素。

- AllocatedStorage (在 CLI 中：`--allocated-storage`或在 Python 中：`allocated_storage`) -數字 (整數) 。

  以 GiB 為單位指定分配的儲存體大小。
- AvailabilityZones (在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`) -一個數組 AvailabilityZone-UTF-8 字符串。

  提供 EC2 可用區域清單，您可以在此還原資料庫叢集快照中的執行個體。
- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集識別符，此即建立此資料庫叢集快照的資料庫叢集。
- DBClusterSnapshotArn (在 CLI 中：`--db-cluster-snapshot-arn`或者在 Python 中：`db_cluster_snapshot_arn`) -UTF-8 字符串。

  資料庫叢集快照的 Amazon Resource Name (ARN)。
- DBClusterSnapshotIdentifier (在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`) -UTF-8 字符串。

  指定資料庫叢集快照的識別碼。必須符合現有快照的識別碼。

  使用 `DBClusterSnapshotIdentifier` 還原資料庫叢集之後，您必須為資料庫叢集未來的任何更新指定相同的 `DBClusterSnapshotIdentifier`。當您為更新指定此屬性時，資料庫叢集不會再次從快照還原，且資料庫中的資料不會變更。

  不過，如果您未指定 `DBClusterSnapshotIdentifier`，則會建立空的資料庫叢集，並刪除原始資料庫叢集。如果您指定的屬性與先前的快照還原屬性不同，則會從 `DBClusterSnapshotIdentifier` 指定的快照還原資料庫叢集，並刪除原始資料庫叢集。
- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  指定資料庫引擎的名稱。
- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  提供此資料庫叢集快照的資料庫引擎版本。
- IAMDatabaseAuthenticationEnabled (在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`) -布爾值。

若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId（在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若`StorageEncrypted`為 true，則為 true，則為加密資料庫叢集快照的 Amazon KMS 金鑰識別符。
- LicenseModel（在 CLI 中：`--license-model`或者在 Python 中：`license_model`）-UTF-8 字符串。

  提供此資料庫叢集快照的授權模式資訊。
- PercentProgress（在 CLI 中：`--percent-progress`或在 Python 中：`percent_progress`）-數字（整數）。

  指定已傳輸的估計資料百分比。
- Port（在 CLI 中：`--port`或在 Python 中：`port`）-數字（整數）。

  指定資料庫叢集在快照時接聽的連接埠。
- SnapshotCreateTime（在 CLI 中：`--snapshot-create-time`，或在 Python 中：`snapshot_create_time`）-時間戳。

  提供快照拍攝的時間，使用供國際標準時間 (UTC)。
- SnapshotType（在 CLI 中：`--snapshot-type`或者在 Python 中：`snapshot_type`）-UTF-8 字符串。

  提供資料庫叢集快照的類型。
- SourceDBClusterSnapshotArn（在 CLI 中：`--source-db-cluster-snapshot-arn`或者在 Python 中：`source_db_cluster_snapshot_arn`）-UTF-8 字符串。

  如果資料庫叢集快照複製自來源資料庫叢集快照，則為來源資料庫叢集快照的 Amazon Resource Name (ARN)，否則為空值。
- Status（在 CLI 中：`--status`或者在 Python 中：`status`）-UTF-8 字符串。

  指定此資料庫叢集快照的狀態。
- StorageEncrypted（在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`）-布爾值。

  指定資料庫叢集快照是否加密。
- VpcId（在 CLI 中：`--vpc-id`或者在 Python 中：`vpc_id`）-UTF-8 字符串。

  提供與資料庫叢集快照關聯的 VPC ID。

錯誤

- 資料庫ClusterSnapshotAlreadyExistsFault (p. 1163)
- 資料庫ClusterSnapshotNotFoundFault (p. 1163)
- 无效ClusterStateFault (p. 1170)
- 无效ClusterSnapshotStateFault (p. 1170)
- SnapshotQuotaExceededFault (p. 1175)
- 公里KeyNotAccessibleFault (p. 1173)

修改資料庫 ClusterSnapshotAttribute (動作)

這個 API 的亞馬遜 CLI 名稱是：`modify-db-cluster-snapshot-attribute`，Python 的名稱是：`modify_db_cluster_snapshot_attribute`。

在手動資料庫叢集快照中新增或移除屬性和值。

若要與其他 Amazon 帳戶共用手動資料庫叢集快照，請指定`restore`為`AttributeName`並使用`ValuesToAdd`參數來新增授權可還原手動資料庫叢集快照的 Amazon 帳戶的 ID。使用此值`all`將手動資料庫叢集快照設為公開，這表示所有 Amazon 帳戶都可以複製或還原該快照。對於任何手動資料庫叢集快照，如果含有您不想讓所有 Amazon 帳戶取得的私人資訊，請不要新增`all`值。如果手動資料庫叢集快照已加密，則可以共用該快照，但只能為`ValuesToAdd`參數指定授權的 Amazon 帳戶 ID 清單。本例中無法使用 `all` 做為該參數的值。

若要檢視哪些 Amazon 帳戶有權複製或還原手動資料庫叢集快照，或是公開或私有的手動資料庫叢集快照，請使用 the section called “描述 B ClusterSnapshotAttributes” (p. 1133) API 動作。

請求

- AttributeName（在 CLI 中：`--attribute-name`或在 Python 中：`attribute_name`）-必需：UTF-8 字符串。

  要修改之資料庫叢集快照屬性的名稱。

  若要管理其他 Amazon 帳戶複製或還原手動資料庫叢集快照的授權，請將此值設為`restore`。
- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或在 Python 中：`db_cluster_snapshot_identifier`）-必需：UTF-8 字符串。

  要修改屬性之資料庫叢集快照的識別符。
- ValuesToAdd（在 CLI 中：`--values-to-add`或者在 Python 中：`values_to_add`）-一個數組 AttributeValue-UTF-8 字符串。

  資料庫叢集快照屬性清單，新增 `AttributeName` 指定的屬性。

  若要授權其他 Amazon 帳戶複製或還原手動資料庫叢集快照，請將此清單設定為包含一或多個 Amazon 帳戶 ID，或`all`使手動資料庫叢集快照可由任何 Amazon 帳戶還原。對於任何手動資料庫叢集快照，如果含有您不想讓所有 Amazon 帳戶取得的私人資訊，請不要新增`all`值。
- ValuesToRemove（在 CLI 中：`--values-to-remove`或者在 Python 中：`values_to_remove`）-一個數組 AttributeValue-UTF-8 字符串。

  資料庫叢集快照屬性清單，移除 `AttributeName` 指定的屬性。

  若要移除其他 Amazon 帳戶複製或還原手動資料庫叢集快照的授權，請將此清單設定為包含一或多個 Amazon 帳戶識別碼，或`all`移除任何 Amazon 帳戶的授權以複製或還原資料庫叢集快照。如果您指定`all`，其帳戶 ID 已明確新增至`restore`屬性的 Amazon 帳戶仍可複製或還原手動資料庫叢集快照。

回應

包含成功呼叫 the section called “描述 B ClusterSnapshotAttributes” (p. 1133) API 動作的結果。

手動資料庫叢集快照屬性用於授權其他 Amazon 帳戶複製或還原手動資料庫叢集快照。如需詳細資訊，請參閱 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。

- DBClusterSnapshotAttributes（在 CLI 中：`--db-cluster-snapshot-attributes`或在 Python 中：`db_cluster_snapshot_attributes`）-數資料庫 ClusterSnapshotAttribute (p. 1110)據庫 ClusterSnapshotAttribute對象的數組。

  手動資料庫叢集快照的屬性和值清單。
- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`）-UTF-8 字符串。

  屬性套用目標之手動資料庫叢集快照的識別符。

錯誤

- 資料庫ClusterSnapshotNotFoundFault (p. 1163)
- 无效ClusterSnapshotStateFault (p. 1170)
- SharedSnapshotQuotaExceededFault (p. 1175)

復原 B (動ClusterFromSnapshot作)

這個 API 的亞馬遜 CLI 名稱是：`restore-db-cluster-from-snapshot`，Python 的名稱是：`restore_db_cluster_from_snapshot`。

從資料庫快照或資料庫叢集快照建立新的資料庫叢集。

如已指定資料庫快照，則會從具有預設組態和預設安全群組的來源資料庫快照建立目標資料庫叢集。

如已指定資料庫叢集快照，則會從與原始來源資料庫叢集有相同組態之來源資料庫叢集還原點建立目標資料庫叢集，但新的資料庫叢集使用預設安全群組建立時除外。

請求

- AvailabilityZones (在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`) - 一個數組 AvailabilityZone-UTF-8 字符串。

  提供 EC2 可用區域清單，已還原資料庫叢集中的執行個體可在這些可用區域中建立。
- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到已建立之還原資料庫叢集的任何快照。
- DatabaseName (在 CLI 中：`--database-name`或者在 Python 中：`database_name`) -UTF-8 字符串。

  不支援.
- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  要從資料庫快照或資料庫叢集快照建立之資料庫叢集的名稱。此參數沒有大小寫之分。

  約束:
  - 必須包含 1 到 63 個字母、數字或連字號
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號

  範例：`my-snapshot-id`
- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  要與新資料庫叢集建立關聯的資料庫叢集參數群組名稱。

  約束:
  - 若有提供，則必須符合現有資料庫的名稱相符ClusterParameterGroup。
- DBSubnetGroupName (在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`) -UTF-8 字符串。

  要用於新資料庫叢集的資料庫子網路群組名稱。

限制條件：若有提供，則必須符合現有資料庫的名稱相符SubnetGroup。

範例：`mySubnetgroup`

- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是停用的。
- EnableCloudwatchLogsExports（在 CLI 中：`--enable-cloudwatch-logs-exports`或在 Python 中：`enable_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  還原的資料庫叢集要匯出至 Amazon CloudWatch 日誌的日誌清單。
- EnableIAMDatabaseAuthentication（在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`）-布爾值。

  True 則為 true，以啟用將 Amazon Identity and Access Management (IAM) 帳戶映射到資料庫帳戶，否則為 false。

  預設：`false`
- Engine（在 CLI 中：`--engine`或在 Python 中：`engine`）-必需：UTF-8 字符串。

  要用於新資料庫叢集的資料庫引擎。

  預設值：與來源相同

  限制：必須與來源引擎相容
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  要用於新資料庫叢集的資料庫引擎版本。
- KmsKeyId（在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`）-UTF-8 字符串。

  從資料庫快照或資料庫叢集快照還原加密資料庫叢集時使用的 Amazon KMS 金鑰識別碼。

  KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。若您使用擁有用來加密新資料庫叢集 KMS 加密金鑰的相同 Amazon 帳戶還原資料庫叢集，您可以改為使用 KMS 金鑰別名，而非 KMS 加密金鑰的 ARN。

  如不指定 `KmsKeyId` 參數值，會發生以下情況：
  - 如已加密 `SnapshotIdentifier` 中的資料庫快照或資料庫叢集快照，則會使用用來加密資料庫快照或資料庫叢集快照的 KMS 金鑰，加密還原的資料庫叢集。
  - 如未加密 `SnapshotIdentifier` 中的資料庫快照或資料庫叢集快照，則不加密還原的資料庫叢集。
- Port（在 CLI 中：`--port`或在 Python 中：`port`）-數字（整數）。

  新資料庫叢集接受連線的連接埠號碼。

  限制條件：值必須為 `1150-65535`

  預設值：與原始資料庫叢集相同的連接埠。
- ServerlessV2ScalingConfiguration（在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`）-一個無伺服器 V2 ScalingConfiguration (p. 1158)對象。

  包含 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。

- SnapshotIdentifier（在 CLI 中：`--snapshot-identifier`或在 Python 中：`snapshot_identifier`）-必需：UTF-8 字符串。

  還原來源的資料庫快照或資料庫叢集快照識別碼。

  您可以使用名稱或 Amazon Resource Name (ARN) 指定資料庫叢集快照。不過，您只能使用 ARN 指定資料庫快照。

  約束:

  - 必須符合現有快照的識別碼。
- Tags（在 CLI 中：`--tags`或在 Python 中：`tags`）-Tag (p. 1150) 標籤對象的數組。

  要指派給已還原資料庫叢集的標籤。
- VpcSecurityGroupIds（在 CLI 中：`--vpc-security-group-ids`或者在 Python 中：`vpc_security_group_ids`）-一個數組 VpcSecurityGroupId-UTF-8 字符串。

  新資料庫叢集將歸屬之 VPC 安全群組清單。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：`--allocated-storage`或在 Python 中：`allocated_storage`）-數字（整數）。

  `AllocatedStorage` 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：`--associated-roles`或在 Python 中：`associated_roles`）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。
- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords（在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`）-數字（長）。

  Neptune 不提供支援。
- BacktrackWindow（在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`）-數字（長）。

  Neptune 不提供支援。
- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`或在 Python 中：`backup_retention_period`）-數字（整數）。

  指定保留自動資料庫快照的天數。
- Capacity（在 CLI 中：`--capacity`或在 Python 中：`capacity`）-數字（整數）。

Neptune 不提供支援。

- CloneGroupId (在 CLI 中：`--clone-group-id`或者在 Python 中：`clone_group_id`) -UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。

- ClusterCreateTime (在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`) -時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。

- CopyTagsToSnapshot (在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`) -布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。

- CrossAccountClone (在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`) -布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。

- DatabaseName (在 CLI 中：`--database-name`或者在 Python 中：`database_name`) -UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。

- DBClusterArn (在 CLI 中：`--db-cluster-arn`或者在 Python 中：`db_cluster_arn`) -UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`) -UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。

- DBClusterMembers (在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`) -數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。

- DBClusterParameterGroup (在 CLI 中：`--db-cluster-parameter-group`或者在 Python 中：`db_cluster_parameter_group`) -UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。

- DbClusterResourceId (在 CLI 中：`--db-cluster-resource-id`或者在 Python 中：`db_cluster_resource_id`) -UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別符。此識別符可在存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。

- DBSubnetGroup (在 CLI 中：`--db-subnet-group`或者在 Python 中：`db_subnet_group`) -UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。

- DeletionProtection (在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`) -布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。

- EarliestBacktrackTime (在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`) -時間戳。

Neptune 不提供支援。

- EarliestRestorableTime (在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`) -時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。

- EnabledCloudwatchLogsExports (在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`) -UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。

- Endpoint (在 CLI 中：`--endpoint`或者在 Python 中：`endpoint`) -UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。

- Engine (在 CLI 中：`--engine`或者在 Python 中：`engine`) -UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。

- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  指示資料庫引擎版本。

- GlobalClusterIdentifier (在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`) -UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別全域資料庫的唯一金鑰。

- HostedZoneId (在 CLI 中：`--hosted-zone-id`或者在 Python 中：`hosted_zone_id`) -UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。

- IAMDatabaseAuthenticationEnabled (在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`) -布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。

- KmsKeyId (在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`) -UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。

- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。

- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。

- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在`ModifyDBCluster`作業中會用來做為回應元素，並包含將在下次維護時段套用的變更。

- PercentProgress (在 CLI 中：`--percent-progress`或者在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。

- Port (在 CLI 中：`--port`或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。

- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或者在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或者在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或者在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  指定此資料庫叢集的目前狀態。
- StorageEncrypted (在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`) -布爾值。

  指定是否要加密資料庫叢集。
- VpcSecurityGroups (在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`) -VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterAlreadyExistsFault (p. 1161)
- 資料庫ClusterQuotaExceededFault (p. 1162)
- StorageQuotaExceededFault (p. 1176)

- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 資料庫SnapshotNotFoundFault (p. 1166)
- 資料庫ClusterSnapshotNotFoundFault (p. 1163)
- 資料庫不足ClusterCapacityFault (p. 1169)
- InsufficientStorageClusterCapacityFault (p. 1169)
- 无效SnapshotStateFault (p. 1171)
- 无效ClusterSnapshotStateFault (p. 1170)
- StorageQuotaExceededFault (p. 1176)
- 无效NetworkStateFault (p. 1173)
- InvalidRestoreFault (p. 1172)
- 資料庫SubnetGroupNotFoundFault (p. 1167)
- InvalidSubnet (p. 1173)
- OptionGroupNotFoundFault (p. 1173)
- 公里KeyNotAccessibleFault (p. 1173)
- 資料庫ClusterParameterGroupNotFoundFault (p. 1162)

復原 B (動ClusterToPointInTime作)

這個 API 的亞馬遜 CLI 名稱是：`restore-db-cluster-to-point-in-time`，Python 的名稱是：`restore_db_cluster_to_point_in_time`。

將資料庫叢集還原到任意時間點。使用者可以還原到 `LatestRestorableTime` 之前的任意時間點，最多 `BackupRetentionPeriod` 天。目標資料庫叢集使用與原始資料庫叢集有相同組態之來源資料庫叢集建立，但新的資料庫叢集使用預設資料庫安全群組建立時除外。

> **Note**
>
> 這個動作只會還原資料庫叢集，不會還原該資料庫叢集的資料庫執行個體。您必須呼叫 the section called “CreateDBInstance” (p. 1056) 動作，為還原的資料庫叢集建立資料庫執行個體，並且在 `DBClusterIdentifier` 中指定所還原資料庫叢集的識別符。只有在 `RestoreDBClusterToPointInTime` 動作完成後，且資料庫叢集為可用時，您才能建立資料庫執行個體。

請求

- DBClusterIdentifier (在 CLI 中：`--db-cluster-identifier`或在 Python 中：`db_cluster_identifier`) -必需：UTF-8 字符串。

  要建立之新資料庫叢集的名稱。

  約束:
  - 必須包含 1 到 63 個字母、數字或連字號
  - 第一個字元必須是字母
  - 不能以連字號結尾或連續包含兩個連字號
- DBClusterParameterGroupName (在 CLI 中：`--db-cluster-parameter-group-name`或者在 Python 中：`db_cluster_parameter_group_name`) -UTF-8 字符串。

  要與新資料庫叢集建立關聯的資料庫叢集參數群組名稱。

  約束:
  - 若有提供，則必須符合現有資料庫的名稱相符ClusterParameterGroup。

- DBSubnetGroupName（在 CLI 中：`--db-subnet-group-name`或者在 Python 中：`db_subnet_group_name`）-UTF-8 字符串。

  要用於新資料庫叢集的資料庫子網路群組名稱。

  限制條件：若有提供，則必須符合現有資料庫的名稱相符SubnetGroup。

  範例：`mySubnetgroup`
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護的值。啟用刪除保護時無法刪除資料庫。根據預設，刪除保護是停用的。
- EnableCloudwatchLogsExports（在 CLI 中：`--enable-cloudwatch-logs-exports`或在 Python 中：`enable_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  還原的資料庫叢集要匯出至CloudWatch記錄的記錄清單。
- EnableIAMDatabaseAuthentication（在 CLI 中：`--enable-iam-database-authentication`，或者在 Python 中：`enable_iam_database_authentication`）-布爾值。

  True 則為 true，以啟用將 Amazon Identity and Access Management (IAM) 帳戶映射到資料庫帳戶，否則為 false。

  預設：`false`
- KmsKeyId（在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`）-UTF-8 字符串。

  從加密的資料庫叢集還原加密資料庫叢集時使用的 Amazon KMS 金鑰識別碼。

  KMS 金鑰識別符是 KMS 加密金鑰的 Amazon Resource Name (ARN)。若您使用擁有用來加密新資料庫叢集 KMS 加密金鑰的相同 Amazon 帳戶還原資料庫叢集，您可以改為使用 KMS 金鑰別名，而非 KMS 加密金鑰的 ARN。

  您可還原至新的資料庫叢集，並使用和加密來源資料庫叢集不同的 KMS 金鑰，加密新的資料庫叢集。新資料庫叢集使用 `KmsKeyId` 參數識別的 KMS 金鑰加密。

  如不指定 `KmsKeyId` 參數值，會發生以下情況：
  - 如已加密資料庫叢集，則已還原的資料庫叢集會使用加密來源資料庫叢集的 KMS 金鑰加密。
  - 如未加密資料庫叢集，則不會加密已還原的資料庫叢集。

  如果 `DBClusterIdentifier` 是指未加密的資料庫叢集，則拒絕還原請求。
- Port（在 CLI 中：`--port`或在 Python 中：`port`）-數字（整數）。

  新資料庫叢集接受連線的連接埠號碼。

  限制條件：值必須為 `1150-65535`

  預設值：與原始資料庫叢集相同的連接埠。
- RestoreToTime（在 CLI 中：`--restore-to-time`，或在 Python 中：`restore_to_time`）-時間戳。

  將資料庫叢集還原至此日期和時間。

  有效值：值必須為國際標準時間 (UTC) 格式的時間

  約束:
  - 必須在資料庫執行個體最近一次可還原時間之前。
  - 如未提供 `UseLatestRestorableTime` 參數，則必須指定。
  - 如果 `UseLatestRestorableTime` 參數為 true，則無法指定

  - 如果 RestoreType 參數為 copy-on-write，則無法指定

  範例：2015-03-07T23:45:00Z
- RestoreType（在 CLI 中：--restore-type或者在 Python 中：restore_type）-UTF-8 字符串。

  要執行的還原類型。您可以指定下列其中一個值：
  - full-copy - 新資料庫叢集將還原為來源資料庫叢集的完整複本。
  - copy-on-write - 新資料庫叢集將還原為來源資料庫叢集的複製品。

  如果您不指定 RestoreType 值，則新資料庫叢集會還原為來源資料庫叢集的完整複本。
- ServerlessV2ScalingConfiguration（在 CLI 中：--serverless-v2-scaling-configuration或在 Python 中：serverless_v2_scaling_configuration）-一個無伺服器 V2 ScalingConfiguration (p. 1158)對象。

  包含 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- SourceDBClusterIdentifier（在 CLI 中：--source-db-cluster-identifier或在 Python 中：source_db_cluster_identifier）-必需：UTF-8 字符串。

  還原來源之來源資料庫叢集的識別符。

  約束:
  - 必須符合現有 DBCluster 的識別碼。
- Tags（在 CLI 中：--tags或在 Python 中：tags）-Tag (p. 1150) 標籤對象的數組。

  要套用到已還原資料庫叢集的標籤。
- UseLatestRestorableTime（在 CLI 中：--use-latest-restorable-time，或者在 Python 中：use_latest_restorable_time）-布爾值。

  值設定為 true 可將資料庫叢集還原到最近可還原的備份時間，否則為 false。

  預設：false

  限制條件：如果提供 RestoreToTime 參數，則無法指定。
- VpcSecurityGroupIds（在 CLI 中：--vpc-security-group-ids或者在 Python 中：vpc_security_group_ids）-一個數組 VpcSecurityGroupId-UTF-8 字符串。

  新資料庫叢集所屬之 VPC 安全群組清單。

回應

包含 Amazon Neptune 資料庫叢集的詳細資訊。

此資料類型在中會用來做為回應元素the section called “DescribeDBClusters” (p. 1040)。

- AllocatedStorage（在 CLI 中：--allocated-storage或在 Python 中：allocated_storage）-數字（整數）。

  AllocatedStorage 一律會傳回 1，因為 Neptune 資料庫叢集儲存體大小並非固定，而是會視需要自動調整。
- AssociatedRoles（在 CLI 中：--associated-roles或在 Python 中：associated_roles）-數資料庫 ClusterRole (p. 1004)據庫ClusterRole對象的數組。

  提供與資料庫叢集相關聯的 Amazon Identity and Access Management (IAM) 角色清單。與資料庫叢集相關聯的 IAM 角色，會授予資料庫叢集代您存取其他 Amazon 服務的許可。

- AutomaticRestartTime（在 CLI 中：`--automatic-restart-time`，或在 Python 中：`automatic_restart_time`）-時間戳。

  資料庫叢集自動重新啟動的時間。
- AvailabilityZones（在 CLI 中：`--availability-zones`或者在 Python 中：`availability_zones`）-一個數組 AvailabilityZone-UTF-8 字符串。

  提供可在其中建立資料庫叢集內執行個體的 EC2 可用區域清單。
- BacktrackConsumedChangeRecords（在 CLI 中：`--backtrack-consumed-change-records`或在 Python 中：`backtrack_consumed_change_records`）-數字（長）。

  Neptune 不提供支援。
- BacktrackWindow（在 CLI 中：`--backtrack-window`或在 Python 中：`backtrack_window`）-數字（長）。

  Neptune 不提供支援。
- BackupRetentionPeriod（在 CLI 中：`--backup-retention-period`或在 Python 中：`backup_retention_period`）-數字（整數）。

  指定保留自動資料庫快照的天數。
- Capacity（在 CLI 中：`--capacity`或在 Python 中：`capacity`）-數字（整數）。

  Neptune 不提供支援。
- CloneGroupId（在 CLI 中：`--clone-group-id`或者在 Python 中：`clone_group_id`）-UTF-8 字符串。

  識別與資料庫叢集建立關聯的複製群組。
- ClusterCreateTime（在 CLI 中：`--cluster-create-time`，或在 Python 中：`cluster_create_time`）-時間戳。

  指定建立資料庫叢集的時間，以國際標準時間 (UTC) 表示。
- CopyTagsToSnapshot（在 CLI 中：`--copy-tags-to-snapshot`，或者在 Python 中：`copy_tags_to_snapshot`）-布爾值。

  如果設定為*`true`*，則會將標籤複製到所建立之資料庫叢集的任何快照。
- CrossAccountClone（在 CLI 中：`--cross-account-clone`，或者在 Python 中：`cross_account_clone`）-布爾值。

  如果設定為`true`，則可跨帳戶複製資料庫叢集。
- DatabaseName（在 CLI 中：`--database-name`或者在 Python 中：`database_name`）-UTF-8 字符串。

  包含此資料庫叢集初始資料庫的名稱，該資料庫是在建立時提供的 (若建立資料庫叢集時有指定的話)。在資料庫叢集使用期間，會傳回此名稱。
- DBClusterArn（在 CLI 中：`--db-cluster-arn`或者在 Python 中：`db_cluster_arn`）-UTF-8 字符串。

  資料庫叢集的 Amazon Resource Name (ARN)。
- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`）-UTF-8 字符串。

  包含使用者提供的資料庫叢集識別碼。此識別碼為可識別資料庫叢集的唯一金鑰。
- DBClusterMembers（在 CLI 中：`--db-cluster-members`或在 Python 中：`db_cluster_members`）-數資料庫 ClusterMember (p. 1004)據庫ClusterMember對象的數組。

  提供組成資料庫叢集的執行個體清單。

- DBClusterParameterGroup（在 CLI 中：`--db-cluster-parameter-group`或者在 Python 中：`db_cluster_parameter_group`）-UTF-8 字符串。

  指定資料庫叢集的資料庫叢集參數群組名稱。
- DbClusterResourceId（在 CLI 中：`--db-cluster-resource-id`或者在 Python 中：`db_cluster_resource_id`）-UTF-8 字符串。

  資料庫叢集的 Amazon 區域專屬不可變識別符。此識別符可在存取資料庫叢集的 Amazon KMS 金鑰時，於 Amazon CloudTrail 日誌項目中找到。
- DBSubnetGroup（在 CLI 中：`--db-subnet-group`或者在 Python 中：`db_subnet_group`）-UTF-8 字符串。

  指定與資料庫叢集建立關聯子網路群組的資訊，包括名稱、描述，以及子網路群組中的子網路。
- DeletionProtection（在 CLI 中：`--deletion-protection`，或者在 Python 中：`deletion_protection`）-布爾值。

  指示資料庫叢集是否已啟用刪除保護。啟用刪除保護時無法刪除資料庫。
- EarliestBacktrackTime（在 CLI 中：`--earliest-backtrack-time`，或在 Python 中：`earliest_backtrack_time`）-時間戳。

  Neptune 不提供支援。
- EarliestRestorableTime（在 CLI 中：`--earliest-restorable-time`，或在 Python 中：`earliest_restorable_time`）-時間戳。

  指定可透過還原來point-in-time還原資料庫的最早時間。
- EnabledCloudwatchLogsExports（在 CLI 中：`--enabled-cloudwatch-logs-exports`或在 Python 中：`enabled_cloudwatch_logs_exports`）-UTF-8 字符串的數組。

  此資料庫叢集設定為匯出至記錄的CloudWatch記錄類型清單。
- Endpoint（在 CLI 中：`--endpoint`或者在 Python 中：`endpoint`）-UTF-8 字符串。

  指定資料庫叢集主要執行個體的連線端點。
- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  提供要用於此資料庫叢集的資料庫引擎名稱。
- EngineVersion（在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`）-UTF-8 字符串。

  指示資料庫引擎版本。
- GlobalClusterIdentifier（在 CLI 中：`--global-cluster-identifier`或在 Python 中：`global_cluster_identifier`）-UTF-8 字符串，長度不小於 1 個或超過 255 個字節，與正則表達式匹配：'[A-ZA-Z] [0-9A-Z-: ._] *'。

  包含使用者提供的全域資料庫叢集識別碼。此識別碼為可識別全域資料庫的唯一金鑰。
- HostedZoneId（在 CLI 中：`--hosted-zone-id`或者在 Python 中：`hosted_zone_id`）-UTF-8 字符串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- IAMDatabaseAuthenticationEnabled（在 CLI 中：`--iam-database-authentication-enabled`，或者在 Python 中：`iam_database_authentication_enabled`）-布爾值。

  若有啟用 Amazon Identity and Access Management (IAM) 帳戶對資料庫帳戶的映射，則為 true，否則為 false。
- KmsKeyId（在 CLI 中：`--kms-key-id`或者在 Python 中：`kms_key_id`）-UTF-8 字符串。

  若 `StorageEncrypted` 為 true，則為加密資料庫叢集的 Amazon KMS 金鑰識別碼。

- LatestRestorableTime (在 CLI 中：`--latest-restorable-time`，或在 Python 中：`latest_restorable_time`) -時間戳。

  指定可以透過還原來point-in-time還原資料庫的最近時間。
- MultiAZ (在 CLI 中：`--multi-az`，或者在 Python 中：`multi_az`) -布爾值。

  指定資料庫叢集是否在多個可用區域中擁有執行個體。
- PendingModifiedValues (在 CLI 中：`--pending-modified-values`或在 Python 中：`pending_modified_values`) -一個ClusterPendingModifiedValues (p. 1004)對象。

  此資料類型在`ModifyDBCluster`作業中會用來做為回應元素，並包含將在下次維護時段套用的變更。
- PercentProgress (在 CLI 中：`--percent-progress`或者在 Python 中：`percent_progress`) -UTF-8 字符串。

  指定操作的進度 (百分比)。
- Port (在 CLI 中：`--port`或在 Python 中：`port`) -數字 (整數) 。

  指定資料庫引擎接聽的連接埠。
- PreferredBackupWindow (在 CLI 中：`--preferred-backup-window`或者在 Python 中：`preferred_backup_window`) -UTF-8 字符串。

  指定每日的時間範圍，若有啟用自動化備份 (由 `BackupRetentionPeriod` 決定)，則自動化備份會在此期間建立。
- PreferredMaintenanceWindow (在 CLI 中：`--preferred-maintenance-window`或者在 Python 中：`preferred_maintenance_window`) -UTF-8 字符串。

  指定每週可能進行系統維護的時段，以世界協調時間 (UTC) 表示。
- ReaderEndpoint (在 CLI 中：`--reader-endpoint`或者在 Python 中：`reader_endpoint`) -UTF-8 字符串。

  資料庫叢集的讀取器端點。資料庫叢集的讀取器端點，用來針對資料庫叢集中可用僅供讀取複本的連線進行負載平衡。當用戶端向讀取器端點請求新連線時，Neptune 會在資料庫叢集中的僅供讀取複本間分散連線請求。此功能有助於在您資料庫叢集中的多個僅供讀取複本間平衡您的讀取工作負載。

  若發生容錯移轉，且您連線的僅供讀取複本提升為主要執行個體，則會中斷您的連線。若要繼續將您的讀取工作負載傳送到叢集內的其他僅供讀取複本，您可以接著重新連線到讀取器端點。
- ReadReplicaIdentifiers (在 CLI 中：`--read-replica-identifiers`或者在 Python 中：`read_replica_identifiers`) -一個數組 ReadReplicaIdentifier-UTF-8 字符串。

  包含與此資料庫叢集相關聯的一或多個僅供讀取複本識別符。
- ReplicationSourceIdentifier (在 CLI 中：`--replication-source-identifier`或者在 Python 中：`replication_source_identifier`) -UTF-8 字符串。

  Neptune 不提供支援。
- ReplicationType (在 CLI 中：`--replication-type`或者在 Python 中：`replication_type`) -UTF-8 字符串。

  Neptune 不提供支援。
- ServerlessV2ScalingConfiguration (在 CLI 中：`--serverless-v2-scaling-configuration`或在 Python 中：`serverless_v2_scaling_configuration`) -一個無伺服器 V2 ScalingConfigurationInfo (p. 1158)對象。

  顯示 Neptune 無伺服器資料庫叢集的擴展組態。

  如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

指定此資料庫叢集的目前狀態。

- StorageEncrypted（在 CLI 中：`--storage-encrypted`，或者在 Python 中：`storage_encrypted`）-布爾值。

  指定是否要加密資料庫叢集。

- VpcSecurityGroups（在 CLI 中：`--vpc-security-groups`，或在 Python 中：`vpc_security_groups`）-VpcSecurityGroupMembership (p. 1158) VpcSecurityGroupMembership 對象的數組。

  提供資料庫叢集歸屬的 VPC 安全群組清單。

錯誤

- 資料庫ClusterAlreadyExistsFault (p. 1161)
- 資料庫ClusterNotFoundFault (p. 1162)
- 資料庫ClusterQuotaExceededFault (p. 1162)
- 資料庫ClusterSnapshotNotFoundFault (p. 1163)
- 資料庫SubnetGroupNotFoundFault (p. 1167)
- 資料庫不足ClusterCapacityFault (p. 1169)
- InsufficientStorageClusterCapacityFault (p. 1169)
- 无效ClusterSnapshotStateFault (p. 1170)
- 无效ClusterStateFault (p. 1170)
- 无效SnapshotStateFault (p. 1171)
- InvalidRestoreFault (p. 1172)
- InvalidSubnet (p. 1173)
- 无效NetworkStateFault (p. 1173)
- 公里KeyNotAccessibleFault (p. 1173)
- OptionGroupNotFoundFault (p. 1173)
- StorageQuotaExceededFault (p. 1176)
- 資料庫ClusterParameterGroupNotFoundFault (p. 1162)

描述 B（動作ClusterSnapshots）

這個 API 的亞馬遜 CLI 名稱是：`describe-db-cluster-snapshots`，Python 的名稱是：`describe_db_cluster_snapshots`。

傳回資料庫叢集快照的相關資訊。此 API 動作支援分頁。

請求

- DBClusterIdentifier（在 CLI 中：`--db-cluster-identifier`或者在 Python 中：`db_cluster_identifier`）-UTF-8 字符串。

  擷取資料庫叢集快照清單的資料庫叢集 ID。此參數無法與 `DBClusterSnapshotIdentifier` 參數一起使用。此參數不區分大小寫。

  約束:

  - 如果提供，必須符合現有 DBCluster 的識別符。

- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`）-UTF-8 字符串。

  描述特定的資料庫叢集快照識別符。此參數無法與 `DBClusterIdentifier` 參數一起使用。此值會以小寫字母字串的形式儲存。

  約束:

  - 若有提供，則必須符合現有資料庫的識別碼ClusterSnapshot。
  - 如果這是用於自動快照的識別符，也必須指定 `SnapshotType` 參數。
- Filters（在 CLI 中：`--filters`，或在 Python 中：`filters`）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- IncludePublic（在 CLI 中：`--include-public`，或者在 Python 中：`include_public`）-布爾值。

  True 表示包含公用且可由任何 Amazon 帳戶複製或還原的手動資料庫叢集快照，否則為 false。預設值為 `false`。預設值為 false。

  您可以使用 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作，以公有狀態共享手動的資料庫叢集快照。
- IncludeShared（在 CLI 中：`--include-shared`，或者在 Python 中：`include_shared`）-布爾值。

  True 表示包含來自其他 Amazon 帳戶的共用手動資料庫叢集快照，且此 Amazon 帳戶已獲授予複製或還原權限，否則為 false。預設值為 `false`。

  您可以透過 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作授予 Amazon 帳戶從另一個 Amazon 帳戶還原手動資料庫叢集快照的權限。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 `DescribeDBClusterSnapshots` 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords（在 CLI 中：`--max-records`或在 Python 中：`max_records`）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- SnapshotType（在 CLI 中：`--snapshot-type`或者在 Python 中：`snapshot_type`）-UTF-8 字符串。

  要傳回的資料庫叢集快照類型。您可以指定下列其中一個值：

  - `automated`-傳回 Amazon Neptune 為我的 Amazon 帳戶自動擷取的所有資料庫叢集快照。
  - `manual`-傳回我的 Amazon 帳戶已擷取的所有資料庫叢集快照。
  - `shared`-傳回已共用至我的 Amazon 帳戶的所有手動資料庫叢集快照。
  - `public` - 傳回已標記為公有的所有資料庫叢集快照。

  如不指定 `SnapshotType` 值，則會傳回自動和手動的資料庫叢集快照。您可以將 `IncludeShared` 參數設成 `true`，納入具有這些結果的共享資料庫叢集快照。您可以將 `IncludePublic` 參數設成 `true`，納入具有這些結果的公有資料庫叢集快照。

  `IncludeShared` 和 `IncludePublic` 參數不適用於值為 `SnapshotType` 的 `manual` 或 `automated`。當 `SnapshotType` 設為 `shared` 時，不適用 `IncludePublic` 參數。當 `SnapshotType` 設為 `public` 時，不適用 `IncludeShared` 參數。

回應

- DBClusterSnapshots（在 CLI 中：`--db-cluster-snapshots`或在 Python 中：`db_cluster_snapshots`）-數資料庫 ClusterSnapshot (p. 1108)據庫ClusterSnapshot對象的數組。

  為使用者提供資料庫叢集快照的清單。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 the section called “描述 B ClusterSnapshots” (p. 1131) 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

錯誤

- 資料庫ClusterSnapshotNotFoundFault (p. 1163)

描述 B（動作ClusterSnapshotAttributes）

這個 API 的亞馬遜 CLI 名稱是：`describe-db-cluster-snapshot-attributes`，Python 的名稱是：`describe_db_cluster_snapshot_attributes`。

傳回手動資料庫叢集快照之資料庫叢集快照屬性名稱和值的清單。

與其他 Amazon 帳戶共用快照時，會`DescribeDBClusterSnapshotAttributes`傳回授權複製或還原手動資料庫叢集快照之 Amazon 帳戶的`restore`屬性和 ID 清單。如果包含`all`在`restore`屬性的值清單中，則手動資料庫叢集快照是公用的，所有 Amazon 帳戶都可以複製或還原。

若要新增或移除 Amazon 帳戶以複製或還原手動資料庫叢集快照的存取權，或將手動資料庫叢集快照設為公開或私有，請使用 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。

請求

- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或在 Python 中：`db_cluster_snapshot_identifier`）-必需：UTF-8 字符串。

  要描述屬性之資料庫叢集快照的識別符。

回應

包含成功呼叫 the section called “描述 B ClusterSnapshotAttributes” (p. 1133) API 動作的結果。

手動資料庫叢集快照屬性用於授權其他 Amazon 帳戶複製或還原手動資料庫叢集快照。如需詳細資訊，請參閱 the section called “修改資料庫 ClusterSnapshotAttribute” (p. 1117) API 動作。

- DBClusterSnapshotAttributes（在 CLI 中：`--db-cluster-snapshot-attributes`或在 Python 中：`db_cluster_snapshot_attributes`）-數資料庫 ClusterSnapshotAttribute (p. 1110)據庫ClusterSnapshotAttribute對象的數組。

  手動資料庫叢集快照的屬性和值清單。
- DBClusterSnapshotIdentifier（在 CLI 中：`--db-cluster-snapshot-identifier`或者在 Python 中：`db_cluster_snapshot_identifier`）-UTF-8 字符串。

  屬性套用目標之手動資料庫叢集快照的識別符。

錯誤

- 資料庫ClusterSnapshotNotFoundFault (p. 1163)

Neptune 事件 API

結構：

- Event (結構) (p. 1134)
- EventCategoriesMap （結構） (p. 1134)
- EventSubscription （結構） (p. 1135)

動作:

- CreateEventSubscription （動作） (p. 1136)
- DeleteEventSubscription （動作） (p. 1138)
- ModifyEventSubscription （動作） (p. 1139)
- DescribeEventSubscriptions （動作） (p. 1141)
- AddSourceIdentifierToSubscription （動作） (p. 1142)
- RemoveSourceIdentifierFromSubscription （動作） (p. 1143)
- DescribeEvents （動作） (p. 1144)
- DescribeEventCategories （動作） (p. 1146)

Event (結構)

此資料類型在 the section called “DescribeEvents” (p. 1144) 動作中會用來作為回應元素。

欄位

- Date – 時間戳記。

  指定事件的日期和時間。
- EventCategoriesUTF-8 字串陣。 EventCategory

  指定事件類別。
- Message – UTF-8 字串。

  提供此事件的文字。
- SourceArn – UTF-8 字串。

  事件的 Amazon Resource Name (ARN)。
- SourceIdentifier – UTF-8 字串。

  提供事件來源的識別符。
- SourceType – UTF-8 字串 (有效值：`db-instance=""` | `db-parameter-group=""` | `db-security-group=""` | `db-snapshot=""` | `db-cluster=""` | `db-cluster-snapshot=""`)。

  指定此事件的來源類型。

EventCategoriesMap （結構）

包含成功呼叫 the section called “DescribeEventCategories” (p. 1146) 動作的結果。

欄位

- EventCategoriesUTF-8 字串陣。 EventCategory

  指定的來源類型事件類別
- SourceType – UTF-8 字串。

  傳回類別所屬之來源類型

EventSubscription （結構）

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

欄位

- CustomerAwsId – UTF-8 字串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId – UTF-8 字串。

  資料庫事件通知訂閱 ID。
- Enabled – 布林值。

  布林值指出訂閱是否啟用。True 指出訂閱已啟用。
- EventCategoriesListUTF-8 字串陣。 EventCategory

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn – UTF-8 字串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn – UTF-8 字串。

  事件通知訂閱的主題 ARN。
- SourceIdsListUTF-8 字串陣。 SourceId

  事件通知訂閱來源 ID 的清單。
- SourceType – UTF-8 字串。

  事件通知訂閱的來源類型。
- Status – UTF-8 字串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。
- SubscriptionCreationTime – UTF-8 字串。

  事件通知訂閱建立的時間。

`EventSubscription` 會用來做為以下項目的回應元素：

- CreateEventSubscription (p. 1136)
- ModifyEventSubscription (p. 1139)
- AddSourceIdentifierToSubscription (p. 1142)
- RemoveSourceIdentifierFromSubscription (p. 1143)
- DeleteEventSubscription (p. 1138)

CreateEventSubscription （動作）

這個 API 的亞馬遜 CLI 名稱是：`create-event-subscription`，Python 的名稱是：`create_event_subscription`。

建立事件通知訂閱。這個動作需要 Neptune 主控台、SNS 主控台或 SNS API 建立的主題 ARN (Amazon Resource Name)。若要使用 SNS 取得 ARN，您必須在 Amazon SNS 中建立一個主題並訂閱該主題。SNS 主控台中會顯示 ARN。

您可以指定要通知的 source (SourceType) 類型，提供觸發事件的 Neptune 來源清單 (SourceIds)，並為您想要通知的事件提供事件類別清單 (EventCategories)。例如，您可以指定 SourceType = DB 實例，SourceIds= 我的數據庫實例 1，我的數據庫實例 2 和 EventCategories = 可用性，Backup。

如果您同時指定 SourceType 和 SourceIds，例如 SourceType = Db-執行個體和 SourceIdentifier = MyDbIndBInce1，您將會收到指定來源的所有 Db-執行個體事件通知。如果您指定， SourceType 但未指定 SourceIdentifier，您將會收到所有 Neptune 來源的事件通知。如果您未指定和， SourceType也不指定 SourceIdentifier，則會收到從屬於您的客戶帳戶的所有 Neptune 來源所產生的事件通知。

請求

- Enabled (在 CLI 中：`--enabled`，或者在 Python 中：`enabled`) -布爾值。

  布林值；設為 true 啟用訂閱，設為 false 建立訂閱，但不啟動訂閱。
- EventCategories (在 CLI 中：`--event-categories`或者在 Python 中：`event_categories`) -一個數組 EventCategory-UTF-8 字符串。

  您要訂閱的事件類別的 SourceType 事件類別清單。您可以使用動作查看指定 SourceType 的類別清DescribeEventCategories單。
- SnsTopicArn (在 CLI 中：`--sns-topic-arn`或在 Python 中：`sns_topic_arn`) -必需：UTF-8 字符串。

  針對事件通知所建立 SNS 主題的 Amazon Resource Name (ARN)。此 ARN 是當您建立並訂閱主題時由 Amazon SNS 所建立。
- SourceIds (在 CLI 中：`--source-ids`或者在 Python 中：`source_ids`) -一個數組 SourceId-UTF-8 字符串。

  傳回事件的事件來源識別碼清單。如未指定，回應中會包含所有來源。識別碼必須以字母開頭，只能包含 ASCII 字母、數字和連字號，而且不得以連字號結尾，或連續包含兩個連字號。

  約束:
  - 如 SourceIds 果提供，也 SourceType 必須提供。
  - 如果來源類型是資料庫執行個體，則必須提供 `DBInstanceIdentifier`。
  - 如果來源類型是資料庫安全群組，則必須提供 `DBSecurityGroupName`。
  - 如果來源類型是資料庫參數群組，則必須提供 `DBParameterGroupName`。
  - 如果來源類型是資料庫快照，則必須提供 `DBSnapshotIdentifier`。
- SourceType (在 CLI 中：`--source-type`或者在 Python 中：`source_type`) -UTF-8 字符串。

  產生事件的來源類型。例如，如果您希望接到資料庫執行個體所產生的事件通知，請將此參數設為 db-instance。如未指定此值，則會傳回所有事件。

有效值：db-instance | db-cluster | db-parameter-group | db-security-group | db-snapshot | db-cluster-snapshot

- SubscriptionName（在 CLI 中：--subscription-name或在 Python 中：subscription_name）-必需：UTF-8 字符串。

  訂閱的名稱。

  限制條件：名稱必須少於 255 個字元。
- Tags（在 CLI 中：--tags或在 Python 中：tags）-Tag (p. 1150) 標籤對象的數組。

  要套用到新事件訂閱的標籤。

回應

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

- CustomerAwsId（在 CLI 中：--customer-aws-id或者在 Python 中：customer_aws_id）-UTF-8 字符串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId（在 CLI 中：--cust-subscription-id或者在 Python 中：cust_subscription_id）-UTF-8 字符串。

  資料庫事件通知訂閱 ID。
- Enabled（在 CLI 中：--enabled，或者在 Python 中：enabled）-布爾值。

  布林值指出訂閱是否啟用。True 指出訂閱已啟用。
- EventCategoriesList（在 CLI 中：--event-categories-list或者在 Python 中：event_categories_list）-一個數組 EventCategory-UTF-8 字符串。

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn（在 CLI 中：--event-subscription-arn或者在 Python 中：event_subscription_arn）-UTF-8 字符串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn（在 CLI 中：--sns-topic-arn或者在 Python 中：sns_topic_arn）-UTF-8 字符串。

  事件通知訂閱的主題 ARN。
- SourceIdsList（在 CLI 中：--source-ids-list或者在 Python 中：source_ids_list）-一個數組 SourceId-UTF-8 字符串。

  事件通知訂閱來源 ID 的清單。
- SourceType（在 CLI 中：--source-type或者在 Python 中：source_type）-UTF-8 字符串。

  事件通知訂閱的來源類型。
- Status（在 CLI 中：--status或者在 Python 中：status）-UTF-8 字符串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。

- SubscriptionCreationTime (在 CLI 中：`--subscription-creation-time`或者在 Python 中：`subscription_creation_time`) -UTF-8 字符串。

  事件通知訂閱建立的時間。

錯誤

- EventSubscriptionQuotaExceededFault (p. 1168)
- SubscriptionAlreadyExistFault (p. 1176)
- SNSInvalidTopicFault (p. 1174)
- SNSNoAuthorizationFault (p. 1174)
- SNSTopicArnNotFoundFault (p. 1175)
- SubscriptionCategoryNotFoundFault (p. 1176)
- SourceNotFoundFault (p. 1175)

DeleteEventSubscription （動作）

這個 API 的亞馬遜 CLI 名稱是：`delete-event-subscription`，Python 的名稱是：`delete_event_subscription`。

刪除事件通知訂閱。

請求

- SubscriptionName (在 CLI 中：`--subscription-name`或在 Python 中：`subscription_name`) -必需：UTF-8 字符串。

  您要刪除的事件通知訂閱名稱。

回應

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

- CustomerAwsId (在 CLI 中：`--customer-aws-id`或者在 Python 中：`customer_aws_id`) -UTF-8 字符串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId (在 CLI 中：`--cust-subscription-id`或者在 Python 中：`cust_subscription_id`) -UTF-8 字符串。

  資料庫事件通知訂閱 ID。
- Enabled (在 CLI 中：`--enabled`，或者在 Python 中：`enabled`) -布爾值。

  布林值指出訂閱是否啟用。True 指出訂閱已啟用。
- EventCategoriesList (在 CLI 中：`--event-categories-list`或者在 Python 中：`event_categories_list`) -一個數組 EventCategory-UTF-8 字符串。

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn (在 CLI 中：`--event-subscription-arn`或者在 Python 中：`event_subscription_arn`) -UTF-8 字符串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn (在 CLI 中：`--sns-topic-arn`或者在 Python 中：`sns_topic_arn`) -UTF-8 字符串。

事件通知訂閱的主題 ARN。

- SourceIdsList (在 CLI 中：`--source-ids-list`或者在 Python 中：`source_ids_list`) -一個數組 SourceId-UTF-8 字符串。

  事件通知訂閱來源 ID 的清單。
- SourceType (在 CLI 中：`--source-type`或者在 Python 中：`source_type`) -UTF-8 字符串。

  事件通知訂閱的來源類型。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。
- SubscriptionCreationTime (在 CLI 中：`--subscription-creation-time`或者在 Python 中：`subscription_creation_time`) -UTF-8 字符串。

  事件通知訂閱建立的時間。

錯誤

- SubscriptionNotFoundFault (p. 1177)
- InvalidEventSubscriptionStateFault (p. 1172)

ModifyEventSubscription （動作）

這個 API 的亞馬遜 CLI 名稱是：`modify-event-subscription`，Python 的名稱是：`modify_event_subscription`。

修改現有的事件通知訂閱。請注意，您無法使用此呼叫修改來源識別符；若要變更訂閱的來源識別符，請使用 the section called “AddSourceIdentifierToSubscription” (p. 1142) 和 the section called “RemoveSourceIdentifierFromSubscription” (p. 1143) 呼叫。

您可以使用動作查看指定 SourceType 事件的事件類別清DescribeEventCategories單。

請求

- Enabled (在 CLI 中：`--enabled`，或者在 Python 中：`enabled`) -布爾值。

  布林值；設為 true 啟用訂閱。
- EventCategories (在 CLI 中：`--event-categories`或者在 Python 中：`event_categories`) -一個數組 EventCategory-UTF-8 字符串。

  您要訂閱的事件類別的 SourceType 事件類別清單。您可以使用動作查看指定 SourceType 的類別清DescribeEventCategories單。
- SnsTopicArn (在 CLI 中：`--sns-topic-arn`或者在 Python 中：`sns_topic_arn`) -UTF-8 字符串。

  針對事件通知所建立 SNS 主題的 Amazon Resource Name (ARN)。此 ARN 是當您建立並訂閱主題時由 Amazon SNS 所建立。
- SourceType (在 CLI 中：`--source-type`或者在 Python 中：`source_type`) -UTF-8 字符串。

產生事件的來源類型。例如，如果您希望接到資料庫執行個體所產生的事件通知，請將此參數設為 db-instance。如未指定此值，則會傳回所有事件。

有效值：DB-執行個體 db-parameter-group | db-security-group | 資料庫快照

- SubscriptionName（在 CLI 中：`--subscription-name`或在 Python 中：`subscription_name`）-必需：UTF-8 字符串。

  事件通知訂閱的名稱。

回應

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

- CustomerAwsId（在 CLI 中：`--customer-aws-id`或者在 Python 中：`customer_aws_id`）-UTF-8 字符串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId（在 CLI 中：`--cust-subscription-id`或者在 Python 中：`cust_subscription_id`）-UTF-8 字符串。

  資料庫事件通知訂閱 ID。
- Enabled（在 CLI 中：`--enabled`，或者在 Python 中：`enabled`）-布爾值。

  布林值指出訂閱是否啟用。True 指出訂閱已啟用。
- EventCategoriesList（在 CLI 中：`--event-categories-list`或者在 Python 中：`event_categories_list`）-一個數組 EventCategory-UTF-8 字符串。

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn（在 CLI 中：`--event-subscription-arn`或者在 Python 中：`event_subscription_arn`）-UTF-8 字符串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn（在 CLI 中：`--sns-topic-arn`或者在 Python 中：`sns_topic_arn`）-UTF-8 字符串。

  事件通知訂閱的主題 ARN。
- SourceIdsList（在 CLI 中：`--source-ids-list`或者在 Python 中：`source_ids_list`）-一個數組 SourceId-UTF-8 字符串。

  事件通知訂閱來源 ID 的清單。
- SourceType（在 CLI 中：`--source-type`或者在 Python 中：`source_type`）-UTF-8 字符串。

  事件通知訂閱的來源類型。
- Status（在 CLI 中：`--status`或者在 Python 中：`status`）-UTF-8 字符串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。
- SubscriptionCreationTime（在 CLI 中：`--subscription-creation-time`或者在 Python 中：`subscription_creation_time`）-UTF-8 字符串。

  事件通知訂閱建立的時間。

錯誤

- EventSubscriptionQuotaExceededFault (p. 1168)
- SubscriptionNotFoundFault (p. 1177)
- SNSInvalidTopicFault (p. 1174)
- SNSNoAuthorizationFault (p. 1174)
- SNSTopicArnNotFoundFault (p. 1175)
- SubscriptionCategoryNotFoundFault (p. 1176)

DescribeEventSubscriptions （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-event-subscriptions`，Python 的名稱是：`describe_event_subscriptions`。

列出客戶帳戶的所有訂閱描述。訂閱的說明包括 SubscriptionName SNSTOPicarn、CustomerID SourceType、SourceID 和狀態。 CreationTime

如果您指定 SubscriptionName，會列出該訂閱的說明。

請求

- Filters（在 CLI 中：`--filters`，或在 Python 中：`filters`）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  由先前的 DescribeOrderable數據庫InstanceOptions請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords（在 CLI 中：`--max-records`或在 Python 中：`max_records`）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- SubscriptionName（在 CLI 中：`--subscription-name`或者在 Python 中：`subscription_name`）-UTF-8 字符串。

  您要描述的事件通知訂閱名稱。

回應

- EventSubscriptionsList（在 CLI 中：`--event-subscriptions-list`，或在 Python 中：`event_subscriptions_list`）-EventSubscription (p. 1135) EventSubscription 對象的數組。

  EventSubscriptions 數據類型的列表。
- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  由先前的 DescribeOrderable數據庫InstanceOptions請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

錯誤

- SubscriptionNotFoundFault (p. 1177)

AddSourceIdentifierToSubscription （動作）

這個 API 的亞馬遜 CLI 名稱是：add-source-identifier-to-subscription，Python 的名稱是：add_source_identifier_to_subscription。

將來源識別符新增至現有的事件通知訂閱。

請求

- SourceIdentifier（在 CLI 中：--source-identifier或在 Python 中：source_identifier）-必需：UTF-8 字符串。

  要新增之事件來源的識別符。

  約束:

  - 如果來源類型是資料庫執行個體，則必須提供 DBInstanceIdentifier。
  - 如果來源類型是資料庫安全群組，則必須提供 DBSecurityGroupName。
  - 如果來源類型是資料庫參數群組，則必須提供 DBParameterGroupName。
  - 如果來源類型是資料庫快照，則必須提供 DBSnapshotIdentifier。
- SubscriptionName（在 CLI 中：--subscription-name或在 Python 中：subscription_name）-必需：UTF-8 字符串。

  您想要新增來源識別符之事件通知訂閱的名稱。

回應

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

- CustomerAwsId（在 CLI 中：--customer-aws-id或者在 Python 中：customer_aws_id）-UTF-8 字符串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId（在 CLI 中：--cust-subscription-id或者在 Python 中：cust_subscription_id）-UTF-8 字符串。

  資料庫事件通知訂閱 ID。
- Enabled（在 CLI 中：--enabled，或者在 Python 中：enabled）-布爾值。

  布林值指出訂閱是否啟用。True 指出訂閱已啟用。
- EventCategoriesList（在 CLI 中：--event-categories-list或者在 Python 中：event_categories_list）-一個數組 EventCategory-UTF-8 字符串。

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn（在 CLI 中：--event-subscription-arn或者在 Python 中：event_subscription_arn）-UTF-8 字符串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn（在 CLI 中：--sns-topic-arn或者在 Python 中：sns_topic_arn）-UTF-8 字符串。

  事件通知訂閱的主題 ARN。
- SourceIdsList（在 CLI 中：--source-ids-list或者在 Python 中：source_ids_list）-一個數組 SourceId-UTF-8 字符串。

  事件通知訂閱來源 ID 的清單。

- SourceType (在 CLI 中：`--source-type`或者在 Python 中：`source_type`) -UTF-8 字符串。

  事件通知訂閱的來源類型。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。
- SubscriptionCreationTime (在 CLI 中：`--subscription-creation-time`或者在 Python 中：`subscription_creation_time`) -UTF-8 字符串。

  事件通知訂閱建立的時間。

錯誤

- SubscriptionNotFoundFault (p. 1177)
- SourceNotFoundFault (p. 1175)

RemoveSourceIdentifierFromSubscription （動作）

這個 API 的亞馬遜 CLI 名稱是：`remove-source-identifier-from-subscription`，Python 的名稱是：`remove_source_identifier_from_subscription`。

從現有的事件通知訂閱移除來源識別符。

請求

- SourceIdentifier (在 CLI 中：`--source-identifier`或在 Python 中：`source_identifier`) -必需：UTF-8 字符串。

  訂閱要移除的來源識別符，例如資料庫執行個體的資料庫執行個體識別符或安全群組的名稱。
- SubscriptionName (在 CLI 中：`--subscription-name`或在 Python 中：`subscription_name`) -必需：UTF-8 字符串。

  您想要移除來源識別符之事件通知訂閱的名稱。

回應

包含成功呼叫 the section called “DescribeEventSubscriptions” (p. 1141) 動作的結果。

- CustomerAwsId (在 CLI 中：`--customer-aws-id`或者在 Python 中：`customer_aws_id`) -UTF-8 字符串。

  與事件通知訂閱相關聯的。
- CustSubscriptionId (在 CLI 中：`--cust-subscription-id`或者在 Python 中：`cust_subscription_id`) -UTF-8 字符串。

  資料庫事件通知訂閱 ID。
- Enabled (在 CLI 中：`--enabled`，或者在 Python 中：`enabled`) -布爾值。

布林值指出訂閱是否啟用。True 指出訂閱已啟用。

- EventCategoriesList (在 CLI 中：`--event-categories-list`或者在 Python 中：`event_categories_list`) -一個數組 EventCategory-UTF-8 字符串。

  事件通知訂閱事件類別的清單。
- EventSubscriptionArn (在 CLI 中：`--event-subscription-arn`或者在 Python 中：`event_subscription_arn`) -UTF-8 字符串。

  事件訂閱的 Amazon Resource Name (ARN)。
- SnsTopicArn (在 CLI 中：`--sns-topic-arn`或者在 Python 中：`sns_topic_arn`) -UTF-8 字符串。

  事件通知訂閱的主題 ARN。
- SourceIdsList (在 CLI 中：`--source-ids-list`或者在 Python 中：`source_ids_list`) -一個數組 SourceId-UTF-8 字符串。

  事件通知訂閱來源 ID 的清單。
- SourceType (在 CLI 中：`--source-type`或者在 Python 中：`source_type`) -UTF-8 字符串。

  事件通知訂閱的來源類型。
- Status (在 CLI 中：`--status`或者在 Python 中：`status`) -UTF-8 字符串。

  事件通知訂閱的狀態。

  約束:

  可以是下列其中一項：建立 | 修改 | 刪除 | 作用中 | 無權限 | topic-not-exist

  狀態「無許可」表示 Neptune 不再具有發佈 SNS 主題的許可。狀態 "topic-not-exist" 表示主題在建立訂閱之後已刪除。
- SubscriptionCreationTime (在 CLI 中：`--subscription-creation-time`或者在 Python 中：`subscription_creation_time`) -UTF-8 字符串。

  事件通知訂閱建立的時間。

錯誤

- SubscriptionNotFoundFault (p. 1177)
- SourceNotFoundFault (p. 1175)

DescribeEvents （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-events`，Python 的名稱是：`describe_events`。

傳回過去 14 天與資料庫執行個體、資料庫安全群組、資料庫快照和資料庫參數群組有關的事件。提供名稱做為參數可取得特定資料庫執行個體、資料庫安全群組、資料庫快照或資料庫參數群組的專屬事件。根據預設，會傳回過去一小時的事件。

請求

- Duration (在 CLI 中：`--duration`或在 Python 中：`duration`) -數字 (整數) 。

  擷取事件的分鐘數。

  預設：60

- EndTime (在 CLI 中：`--end-time`，或在 Python 中：`end_time`) -時間戳。

  擷取事件的時間間隔終點，以 ISO 8601 格式指定。如需 ISO 8601 的詳細資訊，請前往 ISO8601 Wikipedia 頁面。

  範例：2009-07-08T18:00Z
- EventCategories (在 CLI 中：`--event-categories`或者在 Python 中：`event_categories`) -一個數組 EventCategory-UTF-8 字符串。

  觸發事件通知訂閱通知的事件類別清單。
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  先前 DescribeEvents 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- SourceIdentifier (在 CLI 中：`--source-identifier`或者在 Python 中：`source_identifier`) -UTF-8 字符串。

  要傳回其事件的事件來源識別碼。如未指定，回應中會包含所有來源。

  約束:
  - 如果 SourceIdentifier 提供，也 SourceType 必須提供。
  - 如果來源類型是 `DBInstance`，則必須提供 `DBInstanceIdentifier`。
  - 如果來源類型是 `DBSecurityGroup`，則必須提供 `DBSecurityGroupName`。
  - 如果來源類型是 `DBParameterGroup`，則必須提供 `DBParameterGroupName`。
  - 如果來源類型是 `DBSnapshot`，則必須提供 `DBSnapshotIdentifier`。
  - 不能以連字號結尾或連續包含兩個連字號。
- SourceType (在 CLI 中：`--source-type`或在 Python 中：`source_type`) -UTF-8 字符串 (有效值：`db-instance=""db-parameter-group=""|db-security-group="" |db-snapshot="" |db-cluster="" |db-cluster-snapshot=""`) 。

  事件擷取來源的事件來源。如未指定任何值，則會傳回所有事件。
- StartTime (在 CLI 中：`--start-time`，或在 Python 中：`start_time`) -時間戳。

  擷取事件的時間間隔起點，以 ISO 8601 格式指定。如需 ISO 8601 的詳細資訊，請前往 ISO8601 Wikipedia 頁面。

  範例：2009-07-08T18:00Z

回應

- Events (在 CLI 中：`--events`或在 Python 中：`events`) -事件 (p. 1134) 事件對象的數組。

  the section called “事件” (p. 1134) 執行個體的清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

前一個 Events 請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 MaxRecords 指定的值為止。

DescribeEventCategories （動作）

這個 API 的亞馬遜 CLI 名稱是：describe-event-categories，Python 的名稱是：describe_event_categories。

顯示所有事件來源類型或特定來源類型 (如果指定) 的類別清單。

請求

- Filters（在 CLI 中：--filters，或在 Python 中：filters）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- SourceType（在 CLI 中：--source-type或者在 Python 中：source_type）-UTF-8 字符串。

  產生事件的來源類型。

  有效值：DB-執行個體 db-parameter-group | db-security-group | 資料庫快照

回應

- EventCategoriesMapList（在 CLI 中：--event-categories-map-list，或在 Python 中：event_categories_map_list）-EventCategoriesMap (p. 1134) EventCategoriesMap 對象的數組。

  EventCategoriesMap 數據類型的列表。

其他 Neptune API

結構：

- CloudwatchLogsExportConfiguration （結構） (p. 1147)
- 資料庫EngineVersion (結構) (p. 1147)
- EngineDefaults （結構） (p. 1148)
- PendingMaintenanceAction （結構） (p. 1148)
- PendingModifiedValues （結構） (p. 1149)
- PendingCloudwatchLogsExports （結構） (p. 1149)
- ResourcePendingMaintenanceActions （結構） (p. 1150)
- UpgradeTarget （結構） (p. 1150)
- Tag (結構) (p. 1150)

動作:

- AddTagsToResource （動作） (p. 1151)
- ListTagsForResource （動作） (p. 1151)
- RemoveTagsFromResource （動作） (p. 1152)
- ApplyPendingMaintenanceAction （動作） (p. 1152)
- DescribePendingMaintenanceActions （動作） (p. 1153)

- 描述 BEngineVersions （ 動作 ） (p. 1154)

CloudwatchLogsExportConfiguration （ 結構 ）

特定資料庫執行個體或資料庫叢集的哪些 CloudWatch 日誌類型要匯出至 Loggs 的組態設定。

`EnableLogTypes`和`DisableLogTypes`陣列會決定哪些記錄要匯出 (或不匯出) 至 CloudWatch記錄。

欄位

- DisableLogTypes – UTF-8 字串陣列。

  要停用的日誌類型清單。
- EnableLogTypes – UTF-8 字串陣列。

  要啟用的日誌類型清單。

資料庫EngineVersion (結構)

此資料類型在 the section called “描述 BEngineVersions” (p. 1154) 動作中會用來作為回應元素。

欄位

- DBEngineDescription – UTF-8 字串。

  資料庫引擎描述。
- DBEngineVersionDescription – UTF-8 字串。

  資料庫引擎版本描述。
- DBParameterGroupFamily – UTF-8 字串。

  資料庫引擎的資料庫參數群組系列名稱。
- Engine – UTF-8 字串。

  資料庫引擎名稱。
- EngineVersion – UTF-8 字串。

  資料庫引擎版本編號。
- ExportableLogTypes – UTF-8 字串陣列。

  資料庫引擎可匯出至記錄檔的 CloudWatch記錄檔類型。
- SupportedTimezones-時時區 (p. 1158)區對象的數組。

  此引擎針對 `CreateDBInstance` 動作之 `Timezone` 參數支援的時區清單。
- SupportsGlobalDatabases – 布林值。

  指出您是否可以搭配特定資料庫引擎版本使用 Aurora 全域資料庫的值。
- SupportsLogExportsToCloudwatchLogs – 布林值。

  指出引擎版本是否支援將指定的記錄類型匯出 ExportableLogTypes 至 CloudWatch 記錄檔的值。
- SupportsReadReplica – 布林值。

  指出資料庫引擎版本是否支援僅供讀取複本。
- ValidUpgradeTarget – 一個 UpgradeTarget (p. 1150) UpgradeTarget 物件陣列。

此資料庫引擎版本可升級的引擎版本清單。

EngineDefaults （結構）

包含成功呼叫 the section called “DescribeEngineDefaultParameters” (p. 1101) 動作的結果。

欄位

- DBParameterGroupFamily – UTF-8 字串。

  指定要套用引擎預設參數的資料庫參數群組系列名稱。
- Marker – UTF-8 字串。

  先前 EngineDefaults 請求提供的可選分頁令牌。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- Parameters—參數 (p. 1085) 參數物件的陣列。

  包含引擎預設參數的清單。

`EngineDefaults` 會用來做為以下項目的回應元素：

- DescribeEngineDefaultParameters (p. 1101)
- DescribeEngineDefaultClusterParameters (p. 1102)

PendingMaintenanceAction （結構）

提供資源待處理維護動作的資訊。

欄位

- Action – UTF-8 字串。

  可供資源使用的待處理維護動作類型。
- AutoAppliedAfterDate – 時間戳記。

  套用動作時的維護時段日期。維護動作會在此日期後的第一個維護時段中，套用到資源。如已指定此日期，則忽略任何 `next-maintenance` 選擇使用請求。
- CurrentApplyDate – 時間戳記。

  待處理維護動作套用至資源時的有效日期。此日期會考慮從 the section called “ApplyPendingMaintenanceAction” (p. 1152) API、`AutoAppliedAfterDate` 和 `ForcedApplyDate` 收到的選擇使用請求。如未收到選擇使用請求，且沒有任何項目指定為 `AutoAppliedAfterDate` 或 `ForcedApplyDate`，則此值為空白。
- Description – UTF-8 字串。

  提供維護動作更多詳細資訊的描述。
- ForcedApplyDate – 時間戳記。

  自動套用維護動作時的日期。維護動作會在此日期套用至資源，無論資源的維護時段為何。如已指定此日期，則忽略任何 `immediate` 選擇使用請求。
- OptInStatus – UTF-8 字串。

  指出資源已收到的選擇使用請求類型。

PendingModifiedValues （結構）

此資料類型在 the section called “ModifyDBInstance” (p. 1069) 動作中會用來作為回應元素。

欄位

- AllocatedStorage – 數字 (整數)。

  包含要套用或目前正在套用之資料庫執行個體的新 `AllocatedStorage` 大小。
- BackupRetentionPeriod – 數字 (整數)。

  指定保留自動化備份的待處理天數。
- CACertificateIdentifier – UTF-8 字串。

  指定資料庫執行個體 CA 憑證的識別符。
- DBInstanceClass – UTF-8 字串。

  包含要套用或目前正在套用之資料庫執行個體的新 `DBInstanceClass`。
- DBInstanceIdentifier – UTF-8 字串。

  包含要套用或目前正在套用之資料庫執行個體的新 `DBInstanceIdentifier`。
- DBSubnetGroupName – UTF-8 字串。

  資料庫執行個體的新資料庫子網路群組。
- EngineVersion – UTF-8 字串。

  指示資料庫引擎版本。
- Iops – 數字 (整數)。

  指定要套用或目前正在套用之資料庫執行個體的新佈建 IOPS 值。
- MultiAZ – 布林值。

  指出單一可用區資料庫執行個體要變更為異地同步備份部署。
- PendingCloudwatchLogsExports – PendingCloudwatchLogsExports (p. 1149) 物件。

  此`PendingCloudwatchLogsExports`結構指定已啟用和已啟用和已啟用哪些政策類型。 CloudWatch
- Port – 數字 (整數)。

  指定資料庫執行個體的待處理連接埠。
- StorageType – UTF-8 字串。

  指定要和資料庫執行個體建立關聯的儲存體類型。

PendingCloudwatchLogsExports （結構）

組態仍待處理的日誌類型清單。換言之，這些日誌類型正在進行啟用或停用中。

欄位

- LogTypesToDisable – UTF-8 字串陣列。

  正在進行啟用的日誌類型。啟用這些記錄檔之後，這些記錄類型就會匯出至 CloudWatch 記錄檔。
- LogTypesToEnable – UTF-8 字串陣列。

正在進行停用的日誌類型。停用後，這些記錄類型就不會匯出至 CloudWatch 記錄檔。

ResourcePendingMaintenanceActions （結構）

描述資源的待處理維護動作。

欄位

- PendingMaintenanceActionDetails – 一個 PendingMaintenanceAction (p. 1148) PendingMaintenanceAction 物件陣列。

  清單，提供資源待處理維護動作的詳細資訊。
- ResourceIdentifier – UTF-8 字串。

  具有待處理維護動作的資源 ARN。

`ResourcePendingMaintenanceActions` 會用來做為以下項目的回應元素：

- ApplyPendingMaintenanceAction (p. 1152)

UpgradeTarget （結構）

資料庫執行個體可升級的資料庫引擎版本。

欄位

- AutoUpgrade – 布林值。

  指出目標版本是否已套用至任何 AutoMinorVersionUpgrade 設定為 true 的來源資料庫執行個體的值。
- Description – UTF-8 字串。

  資料庫執行個體可升級的資料庫引擎版本。
- Engine – UTF-8 字串。

  升級目標資料庫引擎名稱。
- EngineVersion – UTF-8 字串。

  升級目標資料庫引擎的版本編號。
- IsMajorVersionUpgrade – 布林值。

  值，指出資料庫引擎是否升級到主要版本。
- SupportsGlobalDatabases – 布林值。

  指示是否可在目標引擎版本中使用 Neptune 全域資料庫的值。

Tag (結構)

中繼資料，指派給由金鑰/值對組成的 Amazon Neptune 資源。

欄位

- Key – UTF-8 字串。

鍵是標籤的必要名稱。字串值長度可以是 1 到 128 個 Unicode 字元，不可在前面加上aws:或rds:。該字符串只能包含一組 Unicode 字母，數字，空格，'_'，'。 '、'/'、' = '、' + '、'-'(Java regex："^ ([\\ p {L}\\ p {Z}\\ p {N} _.: /=+\\-] *) $」)。

- Value – UTF-8 字串。

  值是標籤的選用值。字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上aws:或rds:。該字符串只能包含一組 Unicode 字母，數字，空格，'_'，'。 '、'/'、' = '、' + '、'-'(Java regex："^ ([\\ p {L}\\ p {Z}\\ p {N} _.: /=+\\-] *) $」)。

AddTagsToResource （動作）

這個 API 的亞馬遜 CLI 名稱是：add-tags-to-resource，Python 的名稱是：add_tags_to_resource。

將中繼資料標籤新增到 Amazon Neptune 資源。這些標籤也可以搭配成本分配報告，用來追蹤與 Amazon Neptune 資源相關聯的成本，或用在 Amazon Neptune 之 IAM 政策的條件陳述式中。

請求

- ResourceName（在 CLI 中：--resource-name或在 Python 中：resource_name）-必需：UTF-8 字符串。

  要新增標籤的 Amazon Neptune 資源。這個值是 Amazon Resource Name (ARN)。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。
- Tags（在 CLI 中：--tags或在 Python 中：tags）-必需：Tag (p. 1150)標籤對象的數組。

  要指派給 Amazon Neptune 資源的標籤。

回應

- 無回應參數。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)
- 資料庫SnapshotNotFoundFault (p. 1166)
- 資料庫ClusterNotFoundFault (p. 1162)

ListTagsForResource （動作）

這個 API 的亞馬遜 CLI 名稱是：list-tags-for-resource，Python 的名稱是：list_tags_for_resource。

列出 Amazon Neptune 資源的所有標籤。

請求

- Filters（在 CLI 中：--filters，或在 Python 中：filters）-篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援此參數。
- ResourceName（在 CLI 中：--resource-name或在 Python 中：resource_name）-必需：UTF-8 字符串。

具有要列出之標籤的 Amazon Neptune 資源。這個值是 Amazon Resource Name (ARN)。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。

回應

- TagList（在 CLI 中：`--tag-list`或在 Python 中：`tag_list`）-Tag (p. 1150) 標籤對象的數組。

  作業傳回的標籤清 ListTagsForResource 單。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)
- 資料庫SnapshotNotFoundFault (p. 1166)
- 資料庫ClusterNotFoundFault (p. 1162)

RemoveTagsFromResource （動作）

這個 API 的亞馬遜 CLI 名稱是：`remove-tags-from-resource`，Python 的名稱是：`remove_tags_from_resource`。

移除 Amazon Neptune 資源中的中繼資料標籤。

請求

- ResourceName（在 CLI 中：`--resource-name`或在 Python 中：`resource_name`）-必需：UTF-8 字符串。

  要移除標籤的 Amazon Neptune 資源。這個值是 Amazon Resource Name (ARN)。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。
- TagKeys（在 CLI 中：`--tag-keys`或在 Python 中：`tag_keys`）-必需：UTF-8 字符串的數組。

  要移除之標籤的標籤金鑰 (名稱)。

回應

- 無回應參數。

錯誤

- 資料庫InstanceNotFoundFault (p. 1164)
- 資料庫SnapshotNotFoundFault (p. 1166)
- 資料庫ClusterNotFoundFault (p. 1162)

ApplyPendingMaintenanceAction （動作）

這個 API 的亞馬遜 CLI 名稱是：`apply-pending-maintenance-action`，Python 的名稱是：`apply_pending_maintenance_action`。

將待處理維護動作套用到資源 (例如，資料庫執行個體)。

請求

- ApplyAction (在 CLI 中：`--apply-action`或在 Python 中：`apply_action`) -必需：UTF-8 字符串。

  要套用至此資源的待處理維護動作。

  有效值：`system-update`、`db-upgrade`
- OptInType (在 CLI 中：`--opt-in-type`或在 Python 中：`opt_in_type`) -必需：UTF-8 字符串。

  值，指定選擇使用請求的類型，或復原選擇使用請求。無法復原 `immediate` 類型的選擇使用請求。

  有效值：
  - `immediate` - 立即套用維護動作。
  - `next-maintenance` - 在下次資源維護時段期間套用維護動作。
  - `undo-opt-in` - 取消任何現有的 `next-maintenance` 選擇使用請求。
- ResourceIdentifier (在 CLI 中：`--resource-identifier`或在 Python 中：`resource_identifier`) -必需：UTF-8 字符串。

  要套用待處理維護動作之資源的 Amazon Resource Name (ARN)。如需建立 ARN 的資訊，請參閱建構 Amazon Resource Name (ARN)。

回應

描述資源的待處理維護動作。

- PendingMaintenanceActionDetails (在 CLI 中：`--pending-maintenance-action-details`，或在 Python 中：`pending_maintenance_action_details`) -PendingMaintenanceAction (p. 1148) PendingMaintenanceAction 對象的數組。

  清單，提供資源待處理維護動作的詳細資訊。
- ResourceIdentifier (在 CLI 中：`--resource-identifier`或者在 Python 中：`resource_identifier`) -UTF-8 字符串。

  具有待處理維護動作的資源 ARN。

錯誤

- ResourceNotFoundFault (p. 1174)

DescribePendingMaintenanceActions （動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-pending-maintenance-actions`，Python 的名稱是：`describe_pending_maintenance_actions`。

傳回至少有一個待處理維護動作的資源清單 (例如，資料庫執行個體)。

請求

- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  篩選條件，指定傳回待處理維護動作的一或多種資源。

  支援的篩選條件：
  - `db-cluster-id` - 接受資料庫叢集識別符及資料庫叢集 Amazon Resource Name (ARN)。結果清單只包含這些 ARN 找到的資料庫叢集待處理維護動作。
  - `db-instance-id` - 接受資料庫執行個體識別符和資料庫執行個體 ARN。結果清單只包含這些 ARN 找到的資料庫執行個體待處理維護動作。

- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 `DescribePendingMaintenanceActions` 請求提供的選用分頁字符。如已指定此參數，則回應僅包含超出標記的記錄，不超過 `MaxRecords` 指定的記錄數量。
- MaxRecords（在 CLI 中：`--max-records`或在 Python 中：`max_records`）-數字（整數）。

  要在回應中包含的記錄數量上限。若存在的記錄比 `MaxRecords` 值指定的更多，則稱為「標記」的分頁字符會包含在回應中，讓您可以擷取剩餘的結果。

  預設：100

  限制條件：最小 20，最大 100。
- ResourceIdentifier（在 CLI 中：`--resource-identifier`或者在 Python 中：`resource_identifier`）-UTF-8 字符串。

  傳回待處理維護動作的資源 ARN。

回應

- Marker（在 CLI 中：`--marker`或者在 Python 中：`marker`）-UTF-8 字符串。

  前一個 `DescribePendingMaintenanceActions` 請求提供的選用分頁字符。如已指定此參數，則回應僅包含超出標記的記錄，不超過 `MaxRecords` 指定的記錄數量。
- PendingMaintenanceActions（在 CLI 中：`--pending-maintenance-actions`，或在 Python 中：`pending_maintenance_actions`）-ResourcePendingMaintenanceActions (p. 1150) ResourcePendingMaintenanceActions 對象的數組。

  資源的待處理維護動作清單。

錯誤

- ResourceNotFoundFault (p. 1174)

描述 BEngineVersions（動作）

這個 API 的亞馬遜 CLI 名稱是：`describe-db-engine-versions`，Python 的名稱是：`describe_db_engine_versions`。

傳回可用的資料庫引擎清單。

請求

- DBParameterGroupFamily（在 CLI 中：`--db-parameter-group-family`或者在 Python 中：`db_parameter_group_family`）-UTF-8 字符串。

  要傳回詳細資訊的特定資料庫參數群組系列名稱。

  約束:
  - 如果提供，則必須與現有的數據庫匹配ParameterGroupFamily。
- DefaultOnly（在 CLI 中：`--default-only`，或者在 Python 中：`default_only`）-布爾值。

  指出只會傳回指定引擎或引擎預設版本和主要版本的組合。
- Engine（在 CLI 中：`--engine`或者在 Python 中：`engine`）-UTF-8 字符串。

  要傳回的資料庫引擎。

- EngineVersion (在 CLI 中：`--engine-version`或者在 Python 中：`engine_version`) -UTF-8 字符串。

  要傳回的資料庫引擎版本。

  範例：`5.1.49`
- Filters (在 CLI 中：`--filters`，或在 Python 中：`filters`) -篩選條件 (p. 1157) 過濾器對象的數組。

  目前不支援。
- ListSupportedCharacterSets (在 CLI 中：`--list-supported-character-sets`，或者在 Python 中：`list_supported_character_sets`) -布爾值。

  如已指定此參數，而請求的引擎支援 `CreateDBInstance` 的 `CharacterSetName` 參數，則回應會包含每個引擎版本支援的字元集清單。
- ListSupportedTimezones (在 CLI 中：`--list-supported-timezones`，或者在 Python 中：`list_supported_timezones`) -布爾值。

  如已指定此參數，而請求的引擎支援 `CreateDBInstance` 的 `TimeZone` 參數，則回應會包含每個引擎版本支援的時區清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。
- MaxRecords (在 CLI 中：`--max-records`或在 Python 中：`max_records`) -數字 (整數) 。

  要在回應中包含的記錄數量上限。如有 `MaxRecords` 以上的值可用，則稱為標記的分頁字符會包含在回應中，以便擷取下列結果。

  預設：100

  限制條件：最小 20，最大 100。

回應

- DBEngineVersions (在 CLI 中：`--db-engine-versions`或在 Python 中：`db_engine_versions`) -數資料庫EngineVersion (p. 1147)據庫EngineVersion 對象的數組。

  `DBEngineVersion` 元素的清單。
- Marker (在 CLI 中：`--marker`或者在 Python 中：`marker`) -UTF-8 字符串。

  前一個請求提供的選用分頁字符。若指定此參數，則回應只會包含超過標記的記錄，直到 `MaxRecords` 指定的值為止。

常見的 Neptune 資料類型

結構：

- AvailabilityZone (結構) (p. 1156)
- 資料庫 SecurityGroupMembership (結構) (p. 1156)
- DomainMembership (結構) (p. 1156)
- DoubleRange (結構) (p. 1156)
- Endpoint (結構) (p. 1157)
- Filter (結構) (p. 1157)

- Range (結構) (p. 1157)
- 無伺服器 V2 ScalingConfiguration (結構) (p. 1158)
- 無伺服器 V2 ScalingConfigurationInfo (結構) (p. 1158)
- Timezone (結構) (p. 1158)
- VpcSecurityGroupMembership (結構) (p. 1158)

AvailabilityZone (結構)

指定可用區域。

欄位

- Name – UTF-8 字串。

  可用區域的名稱。

資料庫 SecurityGroupMembership (結構)

指定指定資料庫安全群組中的成員資格。

欄位

- DBSecurityGroupName – UTF-8 字串。

  資料庫安全群組的名稱。
- Status – UTF-8 字串。

  資料庫安全群組的狀態。

DomainMembership (結構)

與資料庫執行個體相關聯的 Active Directory 網域成員資格記錄。

欄位

- Domain – UTF-8 字串。

  Active Directory 網域的識別符。
- FQDN – UTF-8 字串。

  Active Directory 網域的完整網域名稱。
- IAMRoleName – UTF-8 字串。

  對 Directory Service 執行 API 呼叫時要使用的 IAM 角色名稱。
- Status – UTF-8 字串。

  資料庫執行個體 Active Directory 網域成員資格的狀態，例如已加入、加入待定、故障等等。

DoubleRange (結構)

double 值的範圍。

欄位

- From – 數字 (雙位數)。

  範圍內的最小值。
- To – 數字 (雙位數)。

  範圍內的最大值。

Endpoint (結構)

指定連線端點。

如需代表 Amazon Neptune less Bunverless Serverless, AverlessDBClusterEndpoint.

欄位

- Address – UTF-8 字串。

  指定資料庫執行個體的 DNS 地址。
- HostedZoneId – UTF-8 字串。

  指定當您建立託管區域時，Amazon Route 53 指派的 ID。
- Port – 數字 (整數)。

  指定資料庫引擎接聽的連接埠。

Filter (結構)

目前不支援此類型。

欄位

- Name – 必要：UTF-8 字串。

  目前不支援此參數。
- Values— 必要：值陣列。UTF-8

  目前不支援此參數。

Range (結構)

整數值範圍。

欄位

- From – 數字 (整數)。

  範圍內的最小值。
- Step – 數字 (整數)。

  範圍的間距值。例如，如果範圍是從 5,000 到 10,000，間距值為 1,000，則有效值為自 5,000 開始，每次加 1,000 的值。即使 7,500 在範圍內，卻不是範圍內的有效值。有效值為 5,000、6,000、7,000、8,000...
- To – 數字 (整數)。

範圍內的最大值。

無伺服器 V2 ScalingConfiguration (結構)

包含 Neptune less 無伺服器資料庫叢集的擴展組態。

如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。

欄位

- MaxCapacity – 數字 (雙位數)。

  NNeptune less Serverless 叢集中資料庫執行個體中資料庫執行個體的 NCOU 數量上限。指定 NCU 值時可使用 0.5 的增量，例如 40、40.5、41 等。
- MinCapacity – 數字 (雙位數)。

  NNeptune less Serverless 叢集中資料庫執行個體中資料庫執行個體的 NCOU 數量下限。指定 NCU 值時可使用 0.5 的增量，例如 8、8.5、9、9 等。

無伺服器 V2 ScalingConfigurationInfo (結構)

顯示 Neptune 無伺服器資料庫叢集的擴展組態。

如需詳細資訊，請參閱 Amazon Neptune 使用者指南中的使用 Amazon Neptune 無伺服器。

欄位

- MaxCapacity – 數字 (雙位數)。

  NNeptune less Serverless 叢集中資料庫執行個體中資料庫執行個體的 NCOU 數量上限。指定 NCU 值時可使用 0.5 的增量，例如 40、40.5、41 等。
- MinCapacity – 數字 (雙位數)。

  NNeptune less Serverless 叢集中資料庫執行個體中資料庫執行個體的 NCOU 數量下限。指定 NCU 值時可使用 0.5 的增量，例如 8、8.5、9、9 等。

Timezone (結構)

與 the section called “DBInstance” (p. 1051) 相關聯的時區。

欄位

- TimezoneName – UTF-8 字串。

  時區的名稱。

VpcSecurityGroupMembership (結構)

此資料類型在 VPC 安全群組成員資格中用為查詢的回應元素。

欄位

- Status – UTF-8 字串。

VPC 安全群組的狀態。

- VpcSecurityGroupId – UTF-8 字串。

  VPC 安全群組的名稱。

針對個別 API 的 Neptune 例外狀況

例外狀況：

- AuthorizationAlreadyExistsFault （結構） (p. 1160)
- AuthorizationNotFoundFault （結構） (p. 1161)
- AuthorizationQuotaExceededFault （結構） (p. 1161)
- CertificateNotFoundFault （結構） (p. 1161)
- 資料庫ClusterAlreadyExistsFault (結構) (p. 1161)
- 資料庫ClusterNotFoundFault (結構) (p. 1162)
- 資料庫ClusterParameterGroupNotFoundFault (結構) (p. 1162)
- 資料庫ClusterQuotaExceededFault (結構) (p. 1162)
- 資料庫ClusterRoleAlreadyExistsFault (結構) (p. 1162)
- 資料庫ClusterRoleNotFoundFault (結構) (p. 1162)
- 資料庫ClusterRoleQuotaExceededFault (結構) (p. 1163)
- 資料庫ClusterSnapshotAlreadyExistsFault (結構) (p. 1163)
- 資料庫ClusterSnapshotNotFoundFault (結構) (p. 1163)
- 資料庫InstanceAlreadyExistsFault (結構) (p. 1163)
- 資料庫InstanceNotFoundFault (結構) (p. 1164)
- 資料庫LogFileNotFoundFault (結構) (p. 1164)
- 資料庫ParameterGroupAlreadyExistsFault (結構) (p. 1164)
- 資料庫ParameterGroupNotFoundFault (結構) (p. 1164)
- 資料庫ParameterGroupQuotaExceededFault (結構) (p. 1164)
- 資料庫SecurityGroupAlreadyExistsFault (結構) (p. 1165)
- 資料庫SecurityGroupNotFoundFault (結構) (p. 1165)
- 資料庫SecurityGroupNotSupportedFault (結構) (p. 1165)
- 資料庫SecurityGroupQuotaExceededFault (結構) (p. 1165)
- 資料庫SnapshotAlreadyExistsFault (結構) (p. 1166)
- 資料庫SnapshotNotFoundFault (結構) (p. 1166)
- 資料庫SubnetGroupAlreadyExistsFault (結構) (p. 1166)
- 資料庫SubnetGroupDoesNotCoverEnough可用空間 (結構) (p. 1166)
- 資料庫SubnetGroupNotAllowedFault (結構) (p. 1166)
- 資料庫SubnetGroupNotFoundFault (結構) (p. 1167)
- 資料庫SubnetGroupQuotaExceededFault (結構) (p. 1167)
- 資料庫SubnetQuotaExceededFault (結構) (p. 1167)
- 資料庫UpgradeDependencyFailureFault (結構) (p. 1167)
- DomainNotFoundFault （結構） (p. 1168)
- EventSubscriptionQuotaExceededFault （結構） (p. 1168)
- GlobalClusterAlreadyExistsFault （結構） (p. 1168)

- GlobalClusterNotFoundFault （結構） (p. 1168)
- GlobalClusterQuotaExceededFault （結構） (p. 1168)
- InstanceQuotaExceededFault （結構） (p. 1169)
- 資料庫不足ClusterCapacityFault (結構) (p. 1169)
- 資料庫不足InstanceCapacityFault (結構) (p. 1169)
- InsufficientStorageClusterCapacityFault （結構） (p. 1169)
- 無效ClusterEndpointStateFault (結構) (p. 1170)
- 無效ClusterSnapshotStateFault (結構) (p. 1170)
- 無效ClusterStateFault (結構) (p. 1170)
- 無效InstanceStateFault (結構) (p. 1170)
- 無效ParameterGroupStateFault (結構) (p. 1170)
- 無效SecurityGroupStateFault (結構) (p. 1171)
- 無效SnapshotStateFault (結構) (p. 1171)
- 無效SubnetGroupFault (結構) (p. 1171)
- 無效SubnetGroupStateFault (結構) (p. 1171)
- 無效SubnetStateFault (結構) (p. 1172)
- InvalidEventSubscriptionStateFault （結構） (p. 1172)
- InvalidGlobalClusterStateFault （結構） (p. 1172)
- InvalidOptionGroupStateFault （結構） (p. 1172)
- InvalidRestoreFault （結構） (p. 1172)
- InvalidSubnet （結構） (p. 1173)
- 無效電腦NetworkStateFault （結構） (p. 1173)
- KMSKeyNotAccessibleFault 結構 (p. 1173)
- OptionGroupNotFoundFault （結構） (p. 1173)
- PointInTimeRestoreNotEnabledFault （結構） (p. 1174)
- ProvisionedIopsNotAvailableInAZ 錯誤 （結構） (p. 1174)
- ResourceNotFoundFault （結構） (p. 1174)
- 社群網站InvalidTopicFault (結構) (p. 1174)
- 社群網站NoAuthorizationFault (結構) (p. 1174)
- 社群網站TopicArnNotFoundFault (結構) (p. 1175)
- SharedSnapshotQuotaExceededFault （結構） (p. 1175)
- SnapshotQuotaExceededFault （結構） (p. 1175)
- SourceNotFoundFault （結構） (p. 1175)
- StorageQuotaExceededFault （結構） (p. 1176)
- StorageTypeNotSupportedFault （結構） (p. 1176)
- SubnetAlreadyInUse （結構） (p. 1176)
- SubscriptionAlreadyExistFault （結構） (p. 1176)
- SubscriptionCategoryNotFoundFault （結構） (p. 1176)
- SubscriptionNotFoundFault （結構） (p. 1177)

AuthorizationAlreadyExistsFault （結構）

傳回的 HTTP 狀態碼：400。

指定的 CIDRIP 或 EC2 安全群組已獲指定的資料庫安全群組授權。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

AuthorizationNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

指定的 CIDRIP 或 EC2 安全群組未獲指定的資料庫安全群組授權。

Neptune 可能不會透過 IAM 授權代您執行必要的動作。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

AuthorizationQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

已達到資料庫安全群組授權配額。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

CertificateNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

CertificateIdentifier不代表現有的憑證。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

使用者的資料庫叢集已有指定的識別符。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBClusterIdentifier 不代表現有的資料庫叢集。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterParameterGroupNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBClusterParameterGroupName 不代表現有的資料庫叢集代表現有的資料庫群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：403。

使用者嘗試建立新的資料庫叢集，但使用者已經達到允許的資料庫叢集配額上限。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterRoleAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

指定的 IAM 角色 Amazon Resource Name (ARN) 與指定的資料庫叢集相關聯。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterRoleNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

指定的 IAM 角色 Amazon Resource Name (ARN) 與指定的資料庫叢集不相關聯。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterRoleQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：400。

您已超過可與指定資料庫叢集相關聯之 IAM 角色的最大數量。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterSnapshotAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

使用者的資料庫叢集快照已有指定的識別符。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ClusterSnapshotNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBInClusterSnapshotIdentifier In代表現有的資料庫叢集快照。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫InstanceAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

使用者的資料庫執行個體已有指定的識別符。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫InstanceNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DB InInInstanceIdentifier 代表現有的資料庫執行個體。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫LogFileNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

LogFileName不代表現有的資料庫記錄檔。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ParameterGroupAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

有同名的資料庫參數群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ParameterGroupNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBParameterGroupName 不代表現有的資料庫參數群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫ParameterGroupQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出允許的資料庫參數群組數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SecurityGroupAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

具有在 DB 中指定的名稱的數據庫安全組SecurityGroupName已存在。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SecurityGroupNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBInSecurityGroupName In代表現有的資料庫安全性群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SecurityGroupNotSupportedFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫安全群組不允許此動作。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SecurityGroupQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出允許的資料庫安全群組數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SnapshotAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫SnapshotIdentifier已由現有快照使用。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SnapshotNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBSnapshotIdentifier 不代表現有的資料庫快照。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetGroupAlreadyExistsFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫SubnetGroupName已由現有的資料庫子網路群組使用。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetGroupDoesNotCoverEnough可用空間 (結構)

傳回的 HTTP 狀態碼：400。

除非只有一個可用區域，否則資料庫子網路群組中的子網路至少應覆蓋兩個可用區域。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetGroupNotAllowedFault (結構)

傳回的 HTTP 狀態碼：400。

指示在建立與來源執行個體位於相同區域的僅供讀取複本時，不SubnetGroup 應指定資料庫。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetGroupNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

DBSubnetGroupName 不代表現有的資料庫子網路群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetGroupQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出允許的資料庫子網路群組數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫SubnetQuotaExceededFault (結構)

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出資料庫子網路允許的子網路數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫UpgradeDependencyFailureFault (結構)

傳回的 HTTP 狀態碼：400。

因為無法修改資料庫依賴的資源，所以資料庫升級失敗。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

DomainNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

Domain 不表示現有的 Active Directory 網域。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

EventSubscriptionQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

您已超過可以訂閱的事件數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

GlobalClusterAlreadyExistsFault （結構）

傳回的 HTTP 狀態碼：400。

`GlobalClusterIdentifier`已存在。選擇新的全域資料庫 ID (唯一名稱)，以建立新的全域資料庫叢集。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

GlobalClusterNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

`GlobalClusterIdentifier`不會參照現有的全域資料庫叢集。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

GlobalClusterQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

此帳戶的全域資料庫叢集數目已經達到允許的上限。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InstanceQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出允許的資料庫執行個體數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫不足ClusterCapacityFault (結構)

傳回的 HTTP 狀態碼：403。

資料庫叢集沒有足夠的容量，無法執行目前的操作。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

資料庫不足InstanceCapacityFault (結構)

傳回的 HTTP 狀態碼：400。

指定的資料庫執行個體類別無法提供指定的可用區域使用。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InsufficientStorageClusterCapacityFault （結構）

傳回的 HTTP 狀態碼：400。

目前的動作沒有足夠的儲存體可用。您可以更新您的子網路群組，使用有更多儲存體可用的不同可用區域，來解決這個錯誤。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效ClusterEndpointStateFault (結構)

傳回的 HTTP 狀態碼：400。

端點處於此狀態時，無法在端點上執行要求的作業。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效ClusterSnapshotStateFault (結構)

傳回的 HTTP 狀態碼：400。

提供的值不是有效的資料庫叢集快照狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效ClusterStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫叢集不為有效狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效InstanceStateFault (結構)

傳回的 HTTP 狀態碼：400。

指定的資料庫執行個體不為「可用」狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效ParameterGroupStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫參數群組正在使用，或為無效狀態。如果您想嘗試刪除參數群組，您無法刪除此狀態的參數群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效SecurityGroupStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫安全群組的狀態不允許刪除。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效SnapshotStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫快照的狀態不允許刪除。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效SubnetGroupFault (結構)

傳回的 HTTP 狀態碼：400。

指示資料庫SubnetGroup 不屬於相同來源執行個體之現有跨區域僅供讀取複本的 VPC。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效SubnetGroupStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫子網路群組因為正在使用，所以無法刪除。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效SubnetStateFault (結構)

傳回的 HTTP 狀態碼：400。

資料庫子網路不為「可用」狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InvalidEventSubscriptionStateFault （結構）

傳回的 HTTP 狀態碼：400。

事件訂閱為無效狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InvalidGlobalClusterStateFault （結構）

傳回的 HTTP 狀態碼：400。

全域叢集處於無效狀態，無法執行要求的作業。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InvalidOptionGroupStateFault （結構）

傳回的 HTTP 狀態碼：400。

選項群組不為「可用」狀態。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InvalidRestoreFault （結構）

傳回的 HTTP 狀態碼：400。

無法從 vpc 備份還原到非 vpc 資料庫執行個體。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

InvalidSubnet （結構）

傳回的 HTTP 狀態碼：400。

請求的子網路無效，或已請求的多個子網路，不全在共同的 VPC 中。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

無效電腦NetworkStateFault （結構）

傳回的 HTTP 狀態碼：400。

因為使用者的變更，資料庫子網路群組在建立後不會覆蓋所有可用區域。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

KMSKeyNotAccessibleFault 結構

傳回的 HTTP 狀態碼：400。

存取 KMS 金鑰發生錯誤。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

OptionGroupNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

找不到指定的選項群組。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

PointInTimeRestoreNotEnabledFault （結構）

傳回的 HTTP 狀態碼：400。

來源資料庫InstanceIdentifier是指BackupRetentionPeriod等於 0 的資料庫執行個體。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

ProvisionedIopsNotAvailableInAZ 錯誤（結構）

傳回的 HTTP 狀態碼：400。

在指定的可用區域中無法使用佈建 IOPS。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

ResourceNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

找不到指定的資源 ID。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

社群網站InvalidTopicFault (結構)

傳回的 HTTP 狀態碼：400。

SNS 主題無效。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

社群網站NoAuthorizationFault (結構)

傳回的 HTTP 狀態碼：400。

沒有 SNS 授權。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

社群網站TopicArnNotFoundFault (結構)

傳回的 HTTP 狀態碼：404。

找不到 SNS 主題的 ARN。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SharedSnapshotQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

您已超過可共享手動資料庫快照的帳戶數量上限。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SnapshotQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

請求會造成使用者超出允許的資料庫快照數目。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SourceNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

找不到來源。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

StorageQuotaExceededFault （結構）

傳回的 HTTP 狀態碼：400。

請求會造成使用者超過所有資料庫執行個體允許的可用儲存量。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

StorageTypeNotSupportedFault （結構）

傳回的 HTTP 狀態碼：400。

StorageType指定的不能與資料庫執行個體相關聯。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SubnetAlreadyInUse （結構）

傳回的 HTTP 狀態碼：400。

資料庫子網路已在可用區域中使用。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SubscriptionAlreadyExistFault （結構）

傳回的 HTTP 狀態碼：400。

此訂閱已存在。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SubscriptionCategoryNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

找不到指定的訂閱類別。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

SubscriptionNotFoundFault （結構）

傳回的 HTTP 狀態碼：404。

找不到指定的訂閱。

欄位

- message – UTF-8 字串。

  說明問題詳細資訊的訊息。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。