IBM QRadar
7.4.3

Guide du langage de requête Ariel

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 73.

Table des matières

A propos de ce guide

Le guide d'AQL (Ariel Query Language) fournit des informations sur l'utilisation de la recherche avancée d'AQL et de l'API.

Utilisateurs concernés

Administrateurs système qui voient des données d'événement ou de flux stockées dans la base de données Ariel.

Documentation technique

Pour rechercher la documentation produit IBM® QRadar sur le Web, y compris toute la documentation traduite, accédez à IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SS42VS/welcome).

Pour plus d'informations sur l'accès à d'autres documents techniques dans la bibliothèque de produits QRadar, voir la note technique relative à l'accès à la documentation IBM (www.ibm.com/support/docview.wss?rs=0&uid=swg21614644).

Contacter le service clients

Pour contacter le service clients, voir la note technique Support and Download (en anglais) (http://www.ibm.com/support/docview.wss?uid=swg21616144).

Déclaration de pratiques de sécurité recommandées

La sécurité des systèmes informatiques implique la protection des systèmes et des informations par la prévention, la détection et la réponse aux accès non autorisés au sein comme à l'extérieur de votre entreprise. Un accès non autorisé peut se traduire par la modification, la destruction, ou une utilisation inadéquate ou malveillante de vos systèmes, y compris l'utilisation de ces derniers pour attaquer d'autres systèmes. Non. Aucun produit ou service informatique ne doit être considéré comme parfaitement sûr et aucun produit, service ou mesure de sécurité ne peut être totalement efficace contre une utilisation inappropriée ou un accès non autorisé. Les systèmes et les produits IBM sont conçus pour s'intégrer à une approche de sécurité complète, ce qui implique nécessairement des procédures opérationnelles supplémentaires, et ils peuvent avoir besoin d'autres systèmes, produit ou services pour optimiser leur efficacité. IBM NE GARANTIT EN AUCUN CAS L'IMMUNITE DES SYSTEMES, PRODUITS OU SERVICES NI L'IMMUNITE DE VOTRE ENTREPRISE CONTRE LE COMPORTEMENT MALVEILLANT OU ILLEGAL DE L'UNE DES PARTIES.

Remarque :

Diverses lois et réglementations peuvent régir l'utilisation de ce Logiciel, y compris celles relatives à la confidentialité, à la protection des données, à l'emploi, aux communications électroniques et à l'archivage. IBM QRadar ne peut être utilisé qu'à des fins légales et de façon légale. Le client accepte d'utiliser ce programme conformément aux lois, réglementations et règles en vigueur et veille à s'y conformer. Le Détenteur de la Licence déclare qu'il obtiendra ou a obtenu tous les accords, droits ou licences nécessaires à l'utilisation légale d'IBM QRadar.

Chapitre 1. Langage AQL (Ariel Query Language) dans l'interface QRadar

L'utilisation de l'AQL peut aider à améliorer les recherches avancées et à fournir des résultats précis.

Lorsque vous utilisez des requêtes AQL, vous pouvez afficher des données de tous les QRadar dans les onglets **Activité de journal** ou **Activité réseau**.

Pour utiliser AQL dans les zones de recherche, procédez comme suit :

- Dans les zones de recherche des onglets **Activité de journal** ou **Activité réseau**, utilisez la combinaison de touches Ctrl + Espace pour afficher la liste complète des fonctions AQL, des zones (propriétés) et des mots clés.
- Ctrl + Entrée permet de créer des requêtes AQL multi-ligne, ce qui rend les requêtes plus lisibles.
- En utilisant les commandes de copie (Ctrl + C) et Coller (Ctrl + V), vous pouvez copier directement vers et depuis la zone **Recherche avancée**.

  Remarque : Veillez à utiliser des guillemets appropriés lorsque vous copiez des requêtes dans la zone de recherche.

Les catégories AQL sont répertoriées avec le composant entré dans l'interface. Le tableau suivant répertorie et explique les différentes catégories :

Tableau 1. Catégories Ariel Query Language

Catégorie	**Définition**
Base de données	Nom de la base de données Ariel, ou table, que vous pouvez interroger. La base de données est `events` ou `flows`.
Mot clé	En général, les clauses SQL de base. Par exemple, `SELECT`, `OR`, `NULL`, `NOT`, AS, ASC (croissant), et plus.
Zone	Indique les informations de base que vous pouvez interroger à partir de la base de données. Exemples : `Access intent`, `VPC ID` et `domainid`.
Fonction	Nom d'une fonction utilisée pour appeler plus d'informations. Les fonctions fonctionnent sur tous les champs et bases de données. Exemples de fonctions : `DATEFORMAT`, `HOSTNAME` et `LOWER`.

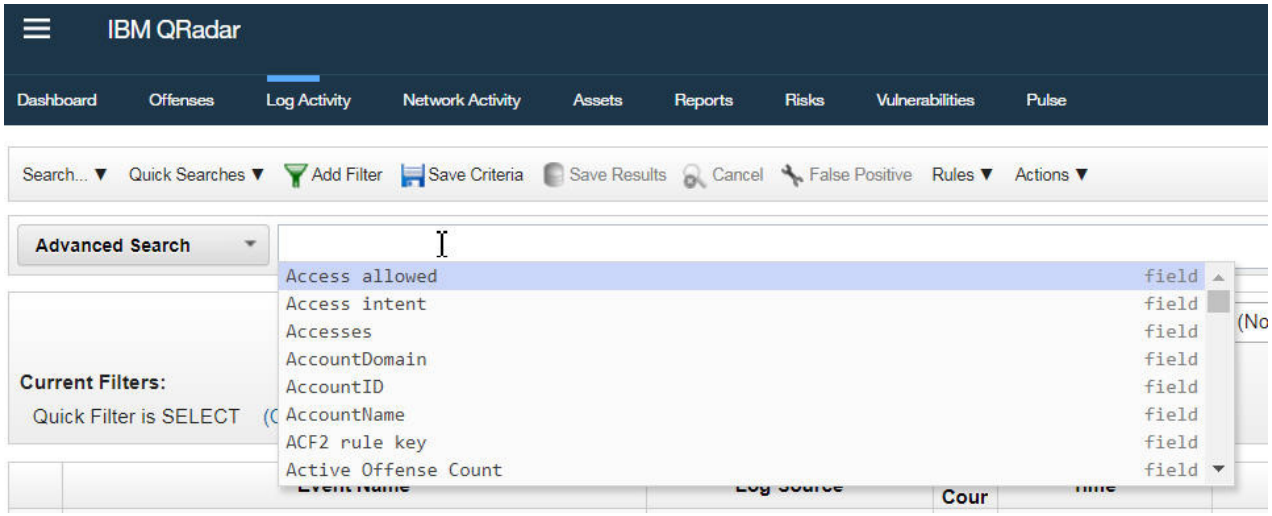

Figure 1. AQL dans la zone de recherche avancée

Chapitre 2. Structure de requête AQL

Utilisez AQL (Ariel Query Language) pour extraire, filtrer et manipuler des données d'événement et de flux de la base de données Ariel dans IBM QRadar. Vous pouvez utiliser AQL pour obtenir des données auxquelles il n'est pas toujours facile d'accéder à partir de l'interface utilisateur.

Le diagramme suivant illustre le flux d'une requête AQL.

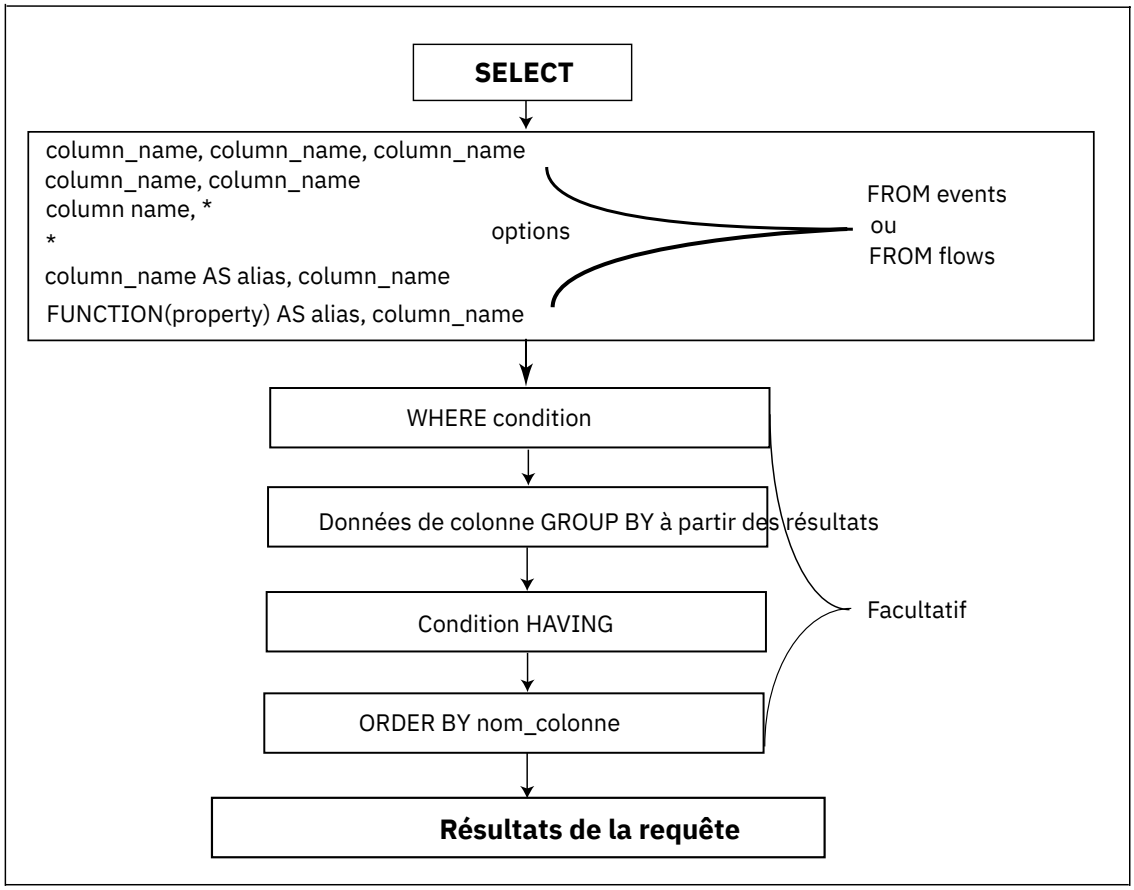

Figure 2. Flux d'une requête AQL

Structure d'une instruction AQL

Utilisez l'instruction SELECT pour sélectionner des zones d'événements ou de flux dans la base de données Ariel, affichées sous forme de colonnes. Par exemple, la requête suivante renvoie les résultats présentés dans le tableau suivant :

```
SELECT sourceip, destinationip, username, protocolid, eventcount FROM events
```

Tableau 2. Résultats de la requête AQL

sourceip	destinationip	Username	Protocolid	eventcount
192.0.2.21	198.51.100.21	Joe Ariel	233	1
192.0.2.22	198.51.100.24	Jim Ariel	233	1

Les requêtes AQL commencent par l'instruction SELECT pour sélectionner des données d'événement ou de flux dans la base de données Ariel. Vous pouvez affiner la sortie de données de l'instruction SELECT à l'aide des clauses WHERE, GROUP BY, HAVING, ORDER BY, LIMIT et LAST.

SELECT

Utilisez l'instruction SELECT pour sélectionner des zones à partir d'événements ou de flux. Par exemple, sélectionnez tous les champs des événements ou des flux en saisissant :

`SELECT * FROM events` ou `SELECT * FROM flows`

Utilisez les clauses suivantes pour filtrer et manipuler les données renvoyées par l'instruction SELECT :

WHERE

Utilisez la clause `WHERE` pour insérer une condition qui filtre la sortie, par exemple `WHERE logsourceid='65'`.

GROUP BY

Utilisez la clause `GROUP BY` pour regrouper les résultats par une ou plusieurs colonnes que vous spécifiez dans la requête (par exemple, `GROUP BY logsourceid`).

HAVING

Utilisez la clause `HAVING` pour spécifier une condition après la clause `GROUP BY`, par exemple `HAVING MAG > 3`.

ORDER BY

Utilisez la clause `ORDER BY` pour trier les résultats d'une colonne de la requête AQL dans un ordre croissant ou décroissant (par exemple, `ORDER BY username DESC`).

LIMIT

Utilisez la clause `LIMIT` pour limiter le nombre de résultats renvoyés à un nombre spécifique (par exemple, `LIMIT 50` pour limiter la sortie à 50 résultats).

LAST

Utilisez une clause LAST pour spécifier un intervalle de temps pour la requête, par exemple `LAST 1 HOURS`.

L'exemple suivant intègre toutes les clauses décrites dans la liste :

```
SELECT sourceip, destinationip, username
FROM events
WHERE username = 'test name'
GROUP by sourceip, destinationip
ORDER BY sourceip DESC
LIMIT 10
LAST 2 DAYS
```

Instruction SELECT

Utilisez l'instruction SELECT pour définir les critères que vous utilisez pour extraire des données d'événement ou de flux.

Utilisez l'instruction SELECT pour définir les colonnes (zones) que vous souhaitez générer à partir de votre requête. Vous pouvez utiliser l'instruction SELECT pour générer des données à partir d'une fonction AQL à l'aide d'un alias de colonne. En général, vous faites référence à des événements ou des flux dans votre instruction SELECT, mais vous pouvez également utiliser l'instruction SELECT avec la base de données GLOBALVIEW ou toute autre base de données à laquelle vous pouvez accéder.

Utilisez l'instruction SELECT pour sélectionner les colonnes à afficher dans la sortie de la requête.

Une instruction SELECT peut inclure les éléments suivants :

- Zones des bases de données d'événements ou de flux
- Propriétés personnalisées à partir des bases de données d'événements ou de flux
- Fonctions que vous utilisez avec des zones pour représenter des données spécifiques à renvoyer.

  Par exemple, la fonction ASSETHOSTNAME(sourceip) recherche le nom d'hôte d'un actif par adresse IP source à une heure spécifique.

Utilisez un astérisque (*) pour indiquer toutes les colonnes.

Les noms de zone et les instructions SELECT et FROM ne sont pas sensibles à la casse. Par exemple, la requête suivante utilise des observations différentes et des parses.

```
select Sourceip, DATEFORMAT(starTTime,'YYYY-MM-dd HH:mm') as startTime from
events WHERE username is noT Null GROUP BY sourceip ordER BY starttime lAsT
3 houRS
```

Les exemples suivants sont des requêtes utilisant des instructions SELECT :

- `SELECT * FROM flows`

  Renvoie toutes les colonnes de la base de données de flux.
- `SELECT sourceip, destinationip FROM events`

  Renvoie uniquement les colonnes sourceip et destinationip de la base de données des événements.
- `SELECT sourceip, * FROM flows`

  Renvoie d'abord la colonne sourceip, suivie de toutes les colonnes de la base de données de flux.
- `SELECT sourceip AS 'MY Source IPs' FROM events`

  Renvoie la colonne sourceip en tant qu'alias ou colonne renommée MY Source IPs.
- `SELECT ASSETHOSTNAME(sourceip) AS 'Host Name', sourceip FROM events`

  Renvoie la sortie de la fonction ASSETHOSTNAME en tant que nom de colonne Host Name et la colonne sourceip à partir de la base de données des événements.

Clause WHERE

Filtrez vos requêtes AQL à l'aide de clauses WHERE. La clause WHERE décrit les critères de filtrage que vous appliquez à la requête et filtre la vue résultante pour accepter uniquement les événements ou les flux qui répondent à la condition spécifiée.

Vous pouvez appliquer la clause WHERE pour ajouter une condition aux critères de recherche dans les requêtes AQL, qui filtre les résultats de la recherche.

Une condition de recherche est une combinaison d'opérateurs logiques et de comparaison qui, ensemble, font un test. Seules les lignes d'entrée qui réussissent le test sont incluses dans le résultat.

Vous pouvez appliquer les filtres suivants lorsque vous utilisez la clause WHERE dans une requête :

- Signe égal (=)
- Symbole différent de (<>)
- Symbole inférieur à (<)

- Symbole supérieur à (>)
- Symbole inférieur ou égal à (<=)
- Symbole supérieur ou égal à (>=)
- BETWEEN entre deux valeurs, par exemple, (64 ET 512)
- LIKE correspondance de casse
- ILIKE correspondance sans tenir compte de la casse
- IS NULL est vide
- AND / OR combine les conditions ou l'une ou l'autre des conditions
- TEXT SEARCH correspondance de chaîne de texte

Exemples de clauses WHERE

L'exemple de requête suivant présente les événements dont le niveau de gravité est supérieur à neuf et proviennent d'une catégorie spécifique.

```
SELECT sourceIP, category, credibility
FROM events
WHERE
severity > 9
AND
category = 5013
```

Modifiez l'ordre d'évaluation en utilisant des parenthèses. Les conditions de recherche entre parenthèses sont évaluées en premier.

```
SELECT sourceIP, category, credibility
FROM events
WHERE
(severity > 9 AND category = 5013)
OR
(severity < 5 AND credibility > 8)
```

Renvoie les événements de la base de données des événements où le texte 'typot' est détecté.

```
SELECT QIDNAME(qid)
AS EventName,
* FROM events
WHERE
TEXT SEARCH 'typot'
```

La requête suivante génère des événements à partir de la base de données d'événements où la santé est incluse dans le nom de la source de journal.

```
SELECT logsourceid, LOGSOURCEGROUPNAME(logsourceid), LOGSOURCENAME(logsourceid)
FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%%health%%'
```

La requête suivante génère des événements où l'ID de type d'unité est égal à 11 (Linux Server DSM), et où QID est égal à 44250002, qui est l'identificateur de Cron Status.

```
SELECT * FROM events
WHERE deviceType= '11'
AND qid= '44250002'
```

Clause Group By

Utilisez la clause GROUP BY pour agréger vos données par une ou plusieurs colonnes. Pour fournir des résultats significatifs de l'agrégation, généralement, l'agrégation de données est combinée à des fonctions d'agrégation sur les colonnes restantes.

Exemples de clauses GROUP BY

L'exemple de requête suivant montre les adresses IP qui ont envoyé plus d'un million d'octets dans tous les flux d'une heure spécifique.

```
SELECT sourceIP, SUM(sourceBytes)
FROM flows where sourceBytes > 1000000
GROUP BY sourceIP
```

Les résultats peuvent ressembler à ceux de la sortie suivante.

```
----------------------------------
| sourceIP | SUM_sourceBytes |
----------------------------------
| 192.0.2.0 | 4282590.0 |
| 10.105.2.10 | 4902509.0 |
| 10.103.70.243 | 2802715.0 |
| 10.103.77.143 | 3313370.0 |
| 10.105.32.29 | 2467183.0 |
| 10.105.96.148 | 8325356.0 |
| 10.103.73.206 | 1629768.0 |
----------------------------------
```

Toutefois, si vous comparez ces informations à une requête non agrégée, la sortie affiche toutes les adresses IP uniques, comme illustré dans la sortie suivante :

```
------------------------------
| sourceIP | sourceBytes |
------------------------------
| 192.0.2.0 | 1448629 |
| 10.105.2.10 | 2412426 |
| 10.103.70.243 | 1793095 |
| 10.103.77.143 | 1449148 |
| 10.105.32.29 | 1097523 |
| 10.105.96.148 | 4096834 |
| 192.0.2.0 | 2833961 |
| 10.105.2.10 | 2490083 |
| 10.103.73.206 | 1629768 |
| 10.103.70.243 | 1009620 |
| 10.105.32.29 | 1369660 |
| 10.103.77.143 | 1864222 |
| 10.105.96.148 | 4228522 |
------------------------------
```

Pour afficher le nombre maximal d'événements, utilisez la syntaxe suivante :

```
SELECT MAX(eventCount) FROM events
```

Pour afficher le nombre d'événements moyens provenant d'une adresse IP source, utilisez la syntaxe suivante :

```
SELECT AVG(eventCount), PROTOCOLNAME(protocolid)
FROM events
GROUP BY sourceIP
```

La sortie affiche les résultats suivants :

```
--------------------------------
| sourceIP | protocol |
--------------------------------
| 192.0.2.0 | TCP.tcp.ip |
| 10.105.2.10 | UDP.udp.ip |
| 10.103.70.243 | UDP.udp.ip |
| 10.103.77.143 | UDP.udp.ip |
| 10.105.32.29 | TCP.tcp.ip |
| 10.105.96.148 | TCP.tcp.ip |
| 192.0.2.0 | TCP.tcp.ip |
| 10.105.2.10 | ICMP.icmp.ip |
--------------------------------
```

Clause HAVING

Utilisez la clause HAVING dans une requête pour appliquer plus de filtres à des données spécifiques en appliquant des filtres aux résultats après la clause GROUP BY.

La clause HAVING suit la clause GROUP BY.

Vous pouvez appliquer les filtres suivants lorsque vous utilisez une clause HAVING dans une requête :

- Signe égal (=)
- Symbole différent de (<>)
- Symbole inférieur à (<)
- Symbole supérieur à (>)
- Symbole inférieur ou égal à (<=)
- Symbole supérieur ou égal à (>=)
- BETWEEN entre deux valeurs, par exemple, (64 ET 512)
- LIKE correspondance sensible à la casse
- ILIKE correspondance sans tenir compte de la casse
- SUM/AVG total ou valeur moyennes
- MAX/MIN Valeurs minimales ou maximales

Exemples de clauses HAVING

L'exemple de requête suivant affiche les résultats pour les utilisateurs qui ont déclenché des événements VPN à partir de plus de quatre adresses IP (HAVING 'Count of Source IPs' > 4) au cours des dernières 24 heures.

```
SELECT username, UNIQUECOUNT(sourceip) AS 'Count of Source IPs'
FROM events
WHERE LOGSOURCENAME(logsourceid) ILIKE '%vpn%'
AND username IS NOT NULL
GROUP BY username
HAVING "Count of Source IPs" > 4
LAST 24 HOURS
```

Remarque : Lorsque vous saisissez une requête AQL, utilisez des guillemets simples pour une comparaison de chaînes, et des guillemets pour une comparaison de valeur de propriété.

L'exemple de requête suivant affiche les résultats des événements où la crédibilité (HAVING credibility > 5) est supérieure à cinq.

```
SELECT username, sourceip, credibility
FROM events
GROUP BY sourceip
HAVING credibility > 5
LAST 1 HOURS
```

Les groupes de requêtes suivants donnent les résultats par IP source, mais affichent uniquement des résultats lorsque la magnitude (HAVING magnitude > 5) est supérieure à cinq.

```
SELECT sourceIP, magnitude
FROM events
GROUP BY sourceIP
HAVING magnitude > 5
```

Clause ORDER BY

Utilisez la clause ORDER BY pour trier la vue obtenue en fonction des résultats de l'expression. Le résultat est trié par ordre croissant ou décroissant.

Remarque : Lorsque vous saisissez une requête AQL, utilisez des guillemets simples pour une comparaison de chaînes, et des guillemets pour une comparaison de valeur de propriété.

Vous pouvez utiliser la clause ORDER BY sur une ou plusieurs colonnes.

Utilisez les clauses GROUP BY et ORDER BY dans une seule requête.

Triez en ordre croissant ou décroissant en ajoutant le mot clé ASC ou DESC à la clause ORDER BY.

Exemples de clauses ORDER BY

Pour interroger AQL pour renvoyer les résultats en ordre décroissant, utilisez la syntaxe suivante :

```
SELECT sourceBytes, sourceIP
FROM flows
WHERE sourceBytes > 1000000
ORDER BY sourceBytes DESC
```

Pour afficher les résultats par ordre croissant, utilisez la syntaxe suivante :

```
SELECT sourceBytes, sourceIP
FROM flows
WHERE sourceBytes > 1000000
ORDER BY sourceBytes ASC
```

Pour déterminer les événements anormaux les plus importants ou les adresses IP les plus consommatrices de bande passante, vous pouvez combiner les clauses GROUP BY et ORDER BY dans une seule requête. Par exemple, la requête suivante affiche l'adresse IP la plus intense de trafic en ordre décroissant :

```
SELECT sourceIP, SUM(sourceBytes)
FROM flows
GROUP BY sourceIP
ORDER BY SUM(sourceBytes) DESC
```

Avertissement :

Lorsque vous utilisez la clause GROUP BY avec un nom de colonne ou une fonction AQL, seule la première valeur est renvoyée pour la colonne GROUP BY, par défaut, même si d'autres valeurs peuvent exister.

Lorsque vous utilisez une zone de temps dans la clause ORDER BY, utilisez une zone de date et d'heure simple, telle que starttime. L'utilisation d'une zone de date et d'heure formatée peut avoir une incidence sur les performances de la recherche.

Clause LIKE

Utilisez la clause LIKE pour extraire les correspondances de chaînes partielles dans la base de données Ariel.

Vous pouvez rechercher des zones à l'aide de la clause LIKE.

Le tableau suivant indique les options de caractère générique prises en charge par l'AQL (Ariel Query Language).

Tableau 3. Options génériques prises en charge pour les clauses LIKE

Caractère générique	Description
%	Correspond à une chaîne d'au moins zéro caractères
_	Correspond à tout caractère unique

Exemples de clauses LIKE

Pour associer des noms tels que Joe, Joanne, Joseph ou tout autre nom commençant par Jo, entrez la requête suivante :

```
SELECT * FROM events WHERE userName LIKE 'Jo%'
```

Pour associer des noms commençant par Jo qui sont de 3 caractères, tels que Joe ou Jon, entrez la requête suivante :

```
SELECT * FROM events WHERE userName LIKE 'Jo_'
```

Vous pouvez entrer l'option de caractère générique à n'importe quel point de la commande, comme illustré dans les exemples suivants.

```
SELECT * FROM flows WHERE sourcePayload LIKE '%xyz'
SELECT * FROM events WHERE UTF8(payload) LIKE '%xyz%'
SELECT * FROM events WHERE UTF8(payload) LIKE '_yz'
```

Exemples de mots clés de correspondance de chaîne

Les mots clés, ILIKE et IMATCHES sont des versions insensibles à la casse de LIKE et MATCHES.

```
SELECT qidname(qid) as test FROM events WHERE test LIKE 'Information%'
SELECT qidname(qid) as test FROM events WHERE test ILIKE 'inForMatiOn%'

SELECT qidname(qid) as test FROM events WHERE test MATCHES '.*Information.*'
SELECT qidname(qid) as test FROM events WHERE test IMATCHES '.*Information.*'
```

fonction COUNT

La fonction COUNT renvoie le nombre de lignes qui satisfont la clause WHERE d'une instruction SELECT.

Si l'instruction SELECT n'a pas de clause WHERE, la fonction COUNT renvoie le nombre total de lignes dans la table.

Exemples de la fonction Count

La requête suivante renvoie le nombre de tous les événements dont la crédibilité est supérieure ou égale à 9.

```
SELECT COUNT(*) FROM events WHERE credibility >= 9
```

La requête suivante renvoie le nombre d'actifs par emplacement et adresse IP source.

```
SELECT ASSETPROPERTY('Location',sourceip)
AS location, COUNT(*)
FROM events
GROUP BY location
LAST 1 days
```

La requête suivante renvoie les noms d'utilisateur, les adresses IP source et le nombre d'événements.

```
SELECT username, sourceip,
COUNT(*) FROM events
GROUP BY username
LAST 600 minutes
```

La colonne `sourceip` est renvoyée comme `FIRST_sourceip`.

Un `sourceip` est renvoyé uniquement par `username`, même si un autre `sourceip` existe.

Remarque :

Lorsque vous utilisez la clause GROUP BY avec un nom de colonne ou une fonction AQL, seule la première valeur est renvoyée à la colonne GROUP BY, par défaut, même si d'autres valeurs existent.

Guillemets

Dans une requête AQL, les termes de requête et les colonnes interrogées nécessitent parfois des guillemets simples ou doubles pour que QRadar puisse analyser la requête.

Le tableau suivant définit quand utiliser des guillemets simples ou doubles.

Tableau 4. Type de guillemets à utiliser dans une requête

Type de guillemets	Utilisation

Célibataire	Pour spécifier une chaîne American National Standards Institute (ANSI) VARCHAR en SQL, comme les paramètres d'un opérateur LIKE ou equals (=), ou tout opérateur qui attend une chaîne VARCHAR.
	Exemples :
	`SELECT * from events WHERE sourceip = '192.0.2.0'`
	`SELECT * from events WHERE userName LIKE '%james%'`
	`SELECT * from events WHERE userName = 'james'`
	`SELECT * FROM events`
	`WHERE INCIDR('10.45.225.14', sourceip)`
	`SELECT * from events WHERE TEXT SEARCH 'my search term'`

Tableau 4. Type de guillemets à utiliser dans une requête (suite)

Type de guillemets	Utilisation

Double	Utilisez des guillemets doubles pour les éléments de requête suivants pour spécifier des noms de table et de colonne contenant des espaces ou des caractères non ASCII, et pour spécifier des noms de propriété personnalisés contenant des espaces ou des caractères non ASCII.
	Exemples :
	`SELECT "username column" AS 'User name' FROM events`
	`SELECT "My custom property name"`
	`AS 'My new alias' FROM events`
	Utilisez des guillemets pour définir le nom d'un objet système tel que la zone, la fonction, la base de données ou un alias existant.
	Exemple :
	`SELECT "Application Category", sourceIP,`
	`EventCount AS 'Count of Events'`
	`FROM events GROUP BY "Count of Events"`
	Utilisez des guillemets pour spécifier un alias existant doté d'un espace lorsque vous utilisez une clause WHERE, GROUP BY ou ORDER BY
	Exemples :
	`SELECT sourceIP, destinationIP, sourcePort,`
	`EventCount AS 'Event Count',`
	`category, hasidentity, username, payload, UtF8(payLoad),`
	`QiD, QiDnAmE(qid) FROM events`
	`WHERE (NOT (sourcePort <= 3003 OR hasidentity = 'True'))`
	`AND (qid = 5000023 OR qid = 5000193)`
	`AND (INCIDR('192.0.2.0/4', sourceIP)`
	`OR NOT INCIDR('192.0.2.0/4', sourceIP)) ORDER BY "Event Count"`
	`DESC LAST 60 MINUTES`
	`SELECT sourceIP, destinationIP, sourcePort, EventCount`
	`AS 'Event Count',`
	`category, hasidentity, username, payload, UtF8(payLoad),`
	`QiD, QiDnAmE(qid)`
	`FROM events ORDER BY "Event Count"`
	`DESC LAST 60 MINUTES`

Tableau 4. Type de guillemets à utiliser dans une requête (suite)

Type de guillemets	Utilisation

Simple ou double	Utilisez des guillemets simples pour spécifier un alias pour une définition de colonne dans une requête.
	Exemple :
	`SELECT username AS 'Name of User', sourceip AS 'IP Source' FROM events`
	Utilisez des guillemets pour spécifier un alias existant avec un espace lorsque vous utilisez une clause WHERE, GROUP BY ou ORDER BY.
	Exemple :
	`SELECT sourceIP AS 'Source IP Address', EventCount AS 'Event Count', QiD, QiDnAmE(qid) FROM events GROUP BY "Source IP Address" LAST 60 MINUTES`

Copie d'exemples de requête à partir du guide AQL

Si vous copiez et collez un exemple de requête contenant des guillemets simples ou doubles à partir du guide AQL, vous devez retaper les guillemets pour être sûr que la requête analyse la syntaxe.

Exemples de requêtes AQL

Utilisez les requêtes d'Ariel Query Language (AQL) pour extraire des données de la base de données Ariel en fonction de critères spécifiques.

Utilisez la syntaxe de requête suivante et respectez-vous l'ordre des clauses lorsque vous générez une requête AQL :

```
[SELECT *, column_name, column_name]
[FROM table_name]
[WHERE search clauses]
[GROUP BY column_reference*]
[HAVING clause]
[ORDER BY column_reference*]
[LIMIT numeric_value]
[TIMEFRAME]
```

Remarque : Lorsque vous utilisez une clause `GROUP BY` ou `ORDER BY` pour trier les informations, vous pouvez uniquement référez `column_names` à partir de votre instruction `SELECT` existante.

Remarque : Par défaut, si la valeur `TIMEFRAME` n'est pas spécifiée, la requête s'exécute sur les cinq dernières minutes des données Ariel.

N'oubliez pas d'utiliser des guillemets simples pour spécifier des valeurs littérales ou des variables et utilisez des guillemets pour les noms de colonne contenant des espaces ou des caractères non-ASCII :

Guillemets simples

Utilisez des guillemets simples lorsque vous référencez le début et la fin d'une chaîne, comme illustré dans les exemples suivants :

```
username LIKE '%User%'
sourceCIDR= '192.0.2.0'
TEXT SEARCH = 'VPN Authenticated user'
QIDNAME(qid) AS 'Event Name'
```

Guillemets doubles

Utilisez des guillemets lorsque les noms de colonne contiennent des espaces ou des caractères non-ASCII, comme illustré dans les exemples suivants :

Noms de propriété personnalisés avec des espaces, tels que « ID de sécurité de compte ».

Valeurs ayant des caractères non-ASCII.

Requêtes AQL simples

Tableau 5. Requêtes AQL simples

Commandes AQL de base	Commentaires
`SELECT * FROM events LAST 10 MINUTES`	Renvoie toutes les zones de la table des événements qui ont été envoyées au cours des 10 dernières minutes.
`SELECT sourceip,destinationip FROM events LAST 24 HOURS`	Renvoie `sourceip` et `destinationip` à partir de la table des événements qui ont été envoyés au cours des dernières 24 heures.
`SELECT * FROM events START '2017 01 01 9:00:00' STOP '2017 01 01 10:20:00'`	Renvoie tous les champs de la table des événements pendant cet intervalle de temps.
`SELECT * FROM events limit 5 LAST 24 HOURS`	Renvoie toutes les zones de la table des événements au cours des dernières 24 heures, avec une sortie limitée à cinq résultats.
`SELECT * FROM events ORDER BY magnitude DESC LAST 24 HOURS`	Renvoie toutes les zones de la table des événements envoyés au cours des dernières 24 heures, en triant le résultat de la plus grande à la plus petite magnitude.
`SELECT * FROM events WHERE magnitude >= 3 LAST 24 HOURS`	Renvoie toutes les zones de la table des événements dont l'ordre de grandeur est inférieur à trois depuis les dernières 24 heures.
`SELECT * FROM events WHERE sourceip = '192.0.2.0' AND destinationip = '198.51.100.0' START '2017 01 01 9:00:00' STOP '2017 01 01 10:20:00'`	Renvoie toutes les zones de la table des événements qui ont l'adresse IP source et la adresse IP de destination spécifiées dans la période spécifiée.
`SELECT * FROM events WHERE INCIDR('192.0.2.0/24', sourceip)`	Renvoie toutes les zones de la table des événements où l'adresse IP source se trouve dans la plage d'adresses IP CIDR spécifiée.
`SELECT * FROM events WHERE username LIKE '%roul%'`	Renvoie toutes les zones de la table des événements où le nom d'utilisateur contient l'exemple de chaîne. Les symboles de pourcentage (%) indiquent que le nom d'utilisateur peut correspondre à une chaîne de zéro ou plusieurs caractères.

Tableau 5. Requêtes AQL simples (suite)

Commandes AQL de base	Commentaires
`SELECT * FROM events WHERE username ILIKE '%ROUL%'`	Renvoie toutes les zones de la table des événements où le nom d'utilisateur contient l'exemple de chaîne, et les résultats sont insensibles à la casse. Les symboles de pourcentage (%) indiquent que le nom d'utilisateur peut correspondre à une chaîne de zéro ou plusieurs caractères.
`SELECT sourceip,category,credibility FROM events WHERE (severity > 3 AND category = 5018)OR (severity < 3 AND credibility > 8)`	Renvoie les zones `sourceip`, `category`et `credibility` à partir de la table des événements avec des niveaux de gravité spécifiques, une catégorie spécifique et un niveau de crédibilité spécifique. La clause AND permet de disposer de plusieurs types de résultats que vous souhaitez obtenir.
`SELECT * FROM events WHERE TEXT SEARCH 'firewall'`	Renvoie toutes les zones de la table des événements qui ont le texte spécifié dans la sortie.
`SELECT * FROM events WHERE username ISNOT NULL`	Renvoie toutes les zones de la table des événements où la valeur `username` n'est pas NULL.

Chapitre 3. Ariel Query Language

Ariel Query Language (AQL) est un langage de requête structuré que vous utilisez pour communiquer avec les bases de données Ariel. Utilisez AQL pour interroger et manipuler des données d'événements et de flux à partir de la base de données Ariel.

Opérateurs logiques et de comparaison AQL

Les opérateurs sont utilisés dans les instructions AQL pour déterminer toute égalité ou différence entre les valeurs. En utilisant les opérateurs dans la clause **WHERE** d'une instruction AQL, les résultats sont filtrés par les résultats qui correspondent aux conditions de la clause **WHERE**.

Le tableau suivant répertorie les opérateurs logiques et de comparaison pris en charge.

Tableau 6. Opérateurs logiques et de comparaison

Opérateur	Description	Exemple
*	Multiplie deux valeurs et renvoie le résultat.	`SELECT * FROM flows WHERE sourceBytes * 1024 < 1`
=	L'opérateur identique à l'opérateur compare deux valeurs et renvoie la valeur true si elles sont égales.	`SELECT * FROM EVENTS WHERE sourceIP = destinationIP`
!=	Compare deux valeurs et renvoie la valeur true si elles sont inégales.	`SELECT * FROM events WHERE sourceIP != destinationip`
< AND <=	Compare deux valeurs et renvoie la valeur true si la valeur du côté gauche est inférieure ou égale à la valeur du côté droit.	`SELECT * FROM flows WHERE sourceBytes < 64 AND destinationBytes <= 64`
> AND >=	Compare deux valeurs et renvoie la valeur true si la valeur du côté gauche est supérieure ou égale à la valeur du côté droit.	`SELECT * FROM flows WHERE sourceBytes > 64 AND destinationBytes >= 64`
/	Divise deux valeurs et renvoie le résultat.	`SELECT * FROM flows WHERE sourceBytes / 8 > 64`
+	Ajoute deux valeurs et renvoie le résultat.	`SELECT * FROM flows WHERE sourceBytes + destinationBytes < 64`
-	Soustra une valeur d'une autre et renvoie le résultat.	`SELECT * FROM flows WHERE sourceBytes - destinationBytes > 0`

Tableau 6. Opérateurs logiques et de comparaison (suite)

Opérateur	Description	Exemple

^	Prend une valeur et l'élève à la puissance spécifiée et renvoie le résultat.	`SELECT *`
		`FROM flows`
		`WHERE sourceBytes ^ 2 < 256`

%	Prend le module d'une valeur et renvoie le résultat.	`SELECT *`
		`FROM flows`
		`WHERE sourceBytes % 8 == 7`

AND	Prend le côté gauche et le côté droit d'une instruction et renvoie la valeur true si les deux sont vraies.	`SELECT *`
		`FROM events`
		`WHERE (sourceIP = destinationIP)`
		`AND (sourcePort = destinationPort)`

BETWEEN (X,Y)	Prend un côté gauche et deux valeurs et renvoie la valeur true si le côté gauche est entre les deux valeurs.	`SELECT *`
		`FROM events`
		`WHERE magnitude`
		`BETWEEN 1 AND 5`

COLLATE	Paramètre à commander par qui permet de rassembler une balise de langue BCP47.	`SELECT *`
		`FROM EVENTS ORDER BY`
		`sourceIP DESC COLLATE 'de-CH'`

IN	Indique plusieurs valeurs dans une clause WHERE. L'opérateur IN est un raccourci pour plusieurs conditions OR.	`SELECT *`
		`FROM EVENTS`
		`WHERE SourceIP IN ('192.0.2.1', '::1', '198.51.100.0')`

INTO	Crée un curseur nommé qui contient des résultats qui peuvent être interrogés à un moment différent.	`SELECT * FROM EVENTS INTO` `'MyCursor' WHERE....`

NOT	Prend dans une instruction et renvoie la valeur true si l'instruction a pour résultat false.	`SELECT * FROM EVENTS`
		`WHERE NOT`
		`(sourceIP = destinationIP)`

ILIKE	Correspond à la valeur transmise si la chaîne transmise est LIKE et n'est pas sensible à la casse. Utilisez % comme caractère générique.	`SELECT *`
		`FROM events WHERE userName`
		`ILIKE '%bob%'`

IMATCHES	Correspond si la chaîne correspond à l'expression régulière fournie et n'est pas sensible à la casse.	`SELECT *`
		`FROM events`
		`WHERE userName`
		`IMATCHES '^.bob.$'`

Tableau 6. Opérateurs logiques et de comparaison (suite)

Opérateur	Description	Exemple

LIMIT	Limite le nombre de résultats au nombre indiqué.	`SELECT *`
		`FROM events LIMIT 100`
		`START '2015-10-28 10:00'`
		`STOP '2015-10-28 11:00'`
		
		Remarque : Placez la clause LIMIT devant une clause START et STOP.

LIKE	Correspond à la valeur transmise si la chaîne transmise est LIKE mais qu'elle est sensible à la casse. Utilisez % comme caractère générique.	`SELECT *`
		`FROM events WHERE userName`
		`LIKE '%bob%'`

MATCHES	Correspond si la chaîne correspond à l'expression régulière fournie.	`SELECT *`
		`FROM events`
		`WHERE userName MATCHES`
		`'^.bob.$'`

NOT NULL	Prend une valeur et renvoie la valeur true si la valeur n'est pas nulle.	`SELECT *`
		`FROM events`
		`WHERE userName`
		`IS NOT NULL`

OR	Prend le côté gauche d'une instruction et le côté droit d'une instruction et renvoie la valeur true si l'un des deux côtés est vrai.	`SELECT *`
		`FROM events`
		`WHERE (sourceIP = destinationIP)`
		`OR (sourcePort = destinationPort)`

TEXT SEARCH	Recherche en texte intégral pour la valeur transmise.
	
	TEXT SEARCH est valide avec les opérateurs AND. Vous ne pouvez pas utiliser TEXT SEARCH avec OR ou d'autres opérateurs ; sinon, vous obtenez une erreur de syntaxe.
	
	Placez TEXT SEARCH dans la première position de la clause WHERE.
	
	Vous pouvez également effectuer des recherches en texte intégral à l'aide du filtre rapide dans l'interface utilisateur QRadar. Pour plus d'informations sur les fonctions de filtre rapide, voir le *IBM QRadar - Guide d'utilisation*.
		`SELECT *`
		`FROM events`
		`WHERE TEXT SEARCH 'firewall'`
		`AND sourceip='192.168.1.1'`
		
		`SELECT sourceip,url`
		`FROM events`
		`WHERE TEXT SEARCH`
		`'download.cdn.mozilla.net'`
		`AND sourceip='192.168.1.1'`
		`START '2015-01-30 16:10:12'`
		`STOP '2015-02-22 17:10:22'`

Exemples d'opérateurs logiques et comparatifs

- Pour rechercher des événements qui ne sont pas analysés, entrez la requête suivante :

```
SELECT * FROM events
WHERE payload = 'false'
```

- Pour rechercher des événements qui renvoient une infraction et qui ont une adresse IP source spécifique, entrez la requête suivante :

```
SELECT * FROM events
WHERE sourceIP = '192.0.2.0'
AND
hasOffense = 'true'
```

- Pour rechercher des événements qui incluent le texte « firewall », entrez la requête suivante :

```
SELECT QIDNAME(qid)
AS EventName,
* FROM events
WHERE TEXT SEARCH 'firewall'
```

Fonctions de calcul et de formatage des données AQL

Utilisez les fonctions de calcul et de formatage d'Ariel Query Language (AQL) sur les résultats de recherche extraits des bases de données Ariel.

Cette liste décrit les fonctions AQL utilisées pour les calculs et le formatage des données :

BASE64

Cible

Renvoie une chaîne codée eb base 64 représentant des données binaires.

Exemple

```
SELECT BASE64(payload)
FROM events
```

Renvoie les charges utiles pour les événements au format BASE64.

CONCAT

Cible

Concatène toutes les chaînes transmises en une seule chaîne.

Exemple

```
SELECT CONCAT(username, ':', sourceip,  ':', destinationip)
FROM events LIMIT 5
```

DATEFORMAT

Cible

Formats en millisecondes depuis 00:00:00 Coordinated Universal Time (UTC) le 1er janvier 1970 à une forme lisible par l'utilisateur.

Exemples

```
SELECT
DATEFORMAT(startTime, 'yyyy-MM-dd hh:mm:ss')
AS StartTime
FROM events
```

```
SELECT DATEFORMAT(starttime,'yyyy-MM-dd hh:mm')
AS 'Start Time',
DATEFORMAT(endtime, 'yyyy-MM-dd hh:mm')
AS Storage_time,
QIDDESCRIPTION(qid)
AS 'Event Name'
FROM events
```

Voir d'autres exemples

DOUBLE

Cible

Convertit une valeur qui représente un nombre en élément de type double.

Exemple

```
DOUBLE('1234')
```

LONG

Cible

Convertit une valeur qui représente un nombre en entier de type long.

Exemples

```
SELECT destinationip,
LONG(SUM(sourcebytes+destinationbytes))
AS TotalBytes
FROM flows
GROUP BY sourceip
```

L'exemple renvoie l'adresse IP de destination et la somme des octets source et de destination dans la colonne TotalBytes.

```
SELECT
LONG(sourceip)
AS long_ip
FROM events
INTO <cursor_name>
WHERE (long_ip & 0x<ff>000000) = 0x<hexadecimal value of IP address>000000
GROUP BY long_ip
LIMIT 20
```

Dans QRadar7.3.1, vous pouvez utiliser la fonction LONG pour convertir des adresses IP en un entier long. QRadar utilise des entiers longs avec des opérateurs bitwise pour effectuer l'arithmétique des adresses IP et le filtrage dans les requêtes AQL. Dans l'exemple, l'adresse IP source est renvoyée sous la forme d'un entier, qui est utilisé par l'opérateur bitwise AND.

Dans l'exemple, le <*ff*> correspond à <*hexadecimal value of IP address*>, qui se trouve dans la position du premier octet pour une adresse IP. <*cursor_name*> peut être n'importe quel nom que vous souhaitez utiliser.

Par exemple, si vous souhaitez renvoyer toutes les adresses IP source avec le numéro 9 dans le premier octet, remplacez la valeur hexadécimale 9, qui est identique à la valeur décimale, dans <*Valeur hexadécimale de l'adresse IP*>.

Voir d'autres exemples de la fonction longue qui sont utilisés avec des opérateurs binaires

PARSEDATETIME

Cible

Transmettez une valeur de temps à l'analyseur syntaxique, par exemple `PARSEDATETIME('time reference')`. *time reference* indique le temps d'analyse de la requête.

Exemple

```
SELECT * FROM events
START PARSEDATETIME('1 hour ago')
```

Voir d'autres exemples de fonctions de temps

PARSETIMESTAMP

Cible

Analyser la représentation du texte de la date et de l'heure et la convertir en temps d'époque UNIX.

Par exemple, analyser le format de date texte suivant :

`Thursday, August 24, 2017 3:30:32 PM GMT +01:00` et le convertir en horodatage d'époque suivant : `1503588632`.

Cette fonction permet de lancer plus facilement des appels à partir de l'API qui sont basés sur des scripts.

Exemple de fonctionnement de la conversion de format horaire

L'exemple suivant montre comment la fonction DATEFORMAT convertit le temps d'utilisation en horodatage de texte en utilisant le format de date spécifié, puis la fonction PARSETIMESTAMP est utilisée pour convertir l'horodatage du texte en format d'heure d'époque.

```
SELECT starttime, DATEFORMAT(starttime,'EEE, MMM d, "yyyy"')
AS "text time format",
PARSETIMESTAMP('EEE, MMM d, "yyyy"', "text time format")
AS 'epoch time returned' from events limit 5
```

L'exemple suivant affiche un extrait de la sortie de la requête :

```
starttime       text time format      epoch time returned
1503920389888   Mon, M08 28, "2017"   1503920389888
```

Exemple de la façon dont PARSETIMESTAMP peut être utilisé pour convertir les temps en temps d'époque afin que les calculs de temps puissent être effectués.

Dans l'exemple suivant, les événements sont renvoyés lorsque la différence de temps entre les heures de déconnexion et de connexion est inférieure à 1 heure.

Le format d'heure `EEE, d MMM yyyy HH:mm:ss.SSSZ` n'est qu'un exemple de format d'heure que vous pouvez utiliser, et `my_login` et `my_logout` sont des propriétés personnalisées dans un format d'heure connu, par exemple, `EEE, MMM d, "yy"`.

```
SELECT * from events
WHERE
PARSETIMESTAMP('EEE, d MMM yyyy HH:mm:ss.SSSZ', my_logout)
- PARSETIMESTAMP('EEE, d MMM yyyy HH:mm:ss.SSSZ', my_login)
< 3600000 last 10 days
```

Voir d'autres exemples de fonctions de temps

NOW

Cible

Renvoie l'heure en cours qui est exprimée en millisecondes depuis l'heure UTC (Temps Universel Coordonné) 00:00:00 le 1er janvier 1970.

Exemple

```
SELECT ASSETUSER(sourceip, NOW())
AS 'Asset user' FROM events
```

Trouvez l'utilisateur de l'actif à ce moment dans le temps (NOW).

LOWER

Cible

Renvoie une représentation en minuscules d'une chaîne.

Exemple

```
SELECT
LOWER(username),
LOWER(LOGSOURCENAME(logsourceid))
FROM events
```

Renvoie les noms d'utilisateur et les noms de source de journal en minuscules.

REPLACEALL

Cible

Recherche une expression régulière et remplace toutes les correspondances par du texte.

Remplace toutes les sous-séquences (*arg2*) de la séquence d'entrée correspondant au modèle (*arg1*) par la chaîne de remplacement (*arg3*).

Exemple

```
REPLACEALL('\d{16}',
username, 'censored')
```

REPLACEFIRST

Cible

Recherche une expression régulière et remplace la première correspondance par du texte.

Remplace la première sous-séquence (*arg2*) de la séquence d'entrée correspondant au modèle (*arg1*) par la chaîne de remplacement (*arg3*).

Exemple

```
REPLACEFIRST('\d{16}',
 username, 'censored')
```

STR

Cible

Convertit un paramètre en chaîne.

Exemple

```
STR(sourceIP)
```

STRLEN

Cible

Renvoie la longueur de cette chaîne.

Exemple

```
SELECT STRLEN(sourceIP),
STRLEN(username) from events
```

Renvoie la longueur de chaîne pour `sourceip` et `username`.

STRPOS

Cible

Renvoie la position (index - commece à zéro) d'une chaîne dans une autre chaîne. Recherche dans la chaîne de l'index de la sous-chaîne spécifiée. Vous pouvez éventuellement spécifier un paramètre supplémentaire pour indiquer à quelle position (index) commencer à rechercher le modèle spécifié.

La recherche de la chaîne commence au décalage spécifié et se déplace vers la fin de la chaîne.

`STRPOS(string, substring, index)`

Renvoie `-1` si la sous-chaîne est introuvable.

Exemples

```
SELECT STRPOS(username, 'name') FROM events
```

```
SELECT STRPOS(sourceip, '180', 2) FROM events)
```

SUBSTRING

Cible

Copie une plage de caractères dans une nouvelle chaîne.

Exemples

```
SELECT SUBSTRING(userName, 0, 3) FROM events
```

```
SELECT SUBSTRING(sourceip, 3, 5) FROM events
```

UPPER

Cible

Renvoie une représentation en majuscules d'une chaîne.

Exemple

```
SELECT
UPPER(username),
UPPER(LOGSOURCENAME(logsourceid))
FROM events
```

Renvoie les noms d'utilisateur et les noms de source de journal en majuscules.

UTF8

Cible

Renvoie la chaîne UTF8 d'un tableau d'octets.

Exemple

```
SELECT UTF8(payload)
FROM events
WHERE sourceip='192.0.2.0'
```

Renvoie la charge UTF8 pour les événements où l'adresse IP source est 192.0.2.0

Fonctions d'agrégation de données AQL

Les fonctions d'agrégation d'Ariel Query Language (AQL) vous aident à agréger et à manipuler les données que vous extrayez de la base de données Ariel.

Fonctions d'agrégation de données

Utilisez les fonctions AQL suivantes pour agréger des données et effectuer des calculs sur les données agrégées que vous extrayez des bases de données AQL:

AVG

Cible

Renvoie la valeur moyenne des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
AVG(magnitude)
FROM events
GROUP BY sourceip
```

COUNT

Cible

Renvoie le nombre de lignes de l'agrégat.

Exemple

```
SELECT sourceip,
COUNT(*)
FROM events
GROUP BY sourceip
```

Voir d'autres exemples

FIRST

Cible

Renvoie la première entrée des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
FIRST(magnitude)
FROM events
GROUP BY sourceip
```

GROUP BY

Cible

Crée un agrégat à partir d'une ou de plusieurs colonnes.

Pour renvoyer les valeurs autres que la première valeur par défaut, utilisez les fonctions telles que COUNT, MAX et AVG.

Exemples

```
SELECT sourceip,
COUNT(*)
FROM  events
GROUP BY sourceip, destinationip
```

```
SELECT username, sourceip,
COUNT(*) FROM events
GROUP BY username
LAST 5 minutes
```

La colonne `sourceip` est renvoyée comme `FIRST_sourceip`. Une seule colonne `sourceip` est renvoyée par `username`, même si une autre colonne `sourceip` existe.

```
SELECT username,
COUNT(sourceip),
COUNT(*) FROM events
GROUP BY username
LAST 5 minutes
```

La colonne `sourceip` est renvoyée comme `COUNT_sourceip`. Le nombre de résultats `sourceip` est renvoyé par `username`.

Voir d'autres exemples

HAVING

Cible

Utilise les opérateurs sur le résultat d'un regroupement par colonne.

Exemple

```
SELECT sourceip,
MAX(magnitude)
AS MAG
FROM events
GROUP BY  sourceip
HAVING MAG > 5
```

Voir d'autres exemples

Les recherches sauvegardées qui incluent la clause d'inclusion et qui sont utilisées pour les rapports planifiés ou les graphiques de série temporelle ne sont pas prises en charge.

LAST

Cible

Retourne la dernière entrée des lignes dans l'agrégat.

Exemple

```
SELECT sourceip,
LAST(magnitude)
FROM events
GROUP BY sourceip
```

MIN

Cible

Renvoie la valeur minimale des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
MIN(magnitude)
FROM events
GROUP BY sourceip
```

MAX

Cible

Renvoie la valeur maximale des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
MAX(magnitude)
FROM events
GROUP BY sourceip
```

STDEV

Cible

Renvoie la valeur d'écart type de l'échantillon des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
STDEV(magnitude)
FROM events
GROUP BY sourceip
```

STDEVP

Cible

Renvoie la valeur de déviation standard de la population des lignes dans l'agrégat.

Exemple

```
SELECT sourceip,
STDEVP(magnitude)
FROM events
GROUP BY sourceip
```

SUM

Cible

Renvoie la somme des lignes de l'agrégat.

Exemple

```
SELECT sourceip,
SUM(sourceBytes)
FROM flows
GROUP BY sourceip
```

UNIQUECOUNT

Cible

Renvoie le nombre unique de la valeur dans l'agrégat.

Exemple

```
SELECT username,
UNIQUECOUNT(sourceip)
AS CountSrcIP
FROM events
GROUP BY sourceip
```

Fonctions d'extraction de données d'AQL

Utilisez les fonctions intégrées d'Ariel Query Language (AQL) pour extraire des données en utilisant des fonctions de requête de données et des propriétés d'ID de zone à partir de la base de données Ariel.

Utilisez les fonctions AQL suivantes pour extraire des données des bases de données Ariel :

Fonctions d'extraction de données

- «APPLICATIONNAME», à la page 29

APPLICATIONNAME

Cible

Renvoie les noms d'application de flux par ID application

Paramètres

ID d'application

Exemple

```
SELECT APPLICATIONNAME(applicationid)
AS 'Name of App'
FROM flows
```

Renvoie les noms des applications de la base de données de flux. Ces noms d'application sont répertoriés dans la colonne **Nmae of App**, qui est un alias.

ARIELSERVERS4EPID

Cible

Utilisez la fonction ARIELSERVERS4EPID pour spécifier l'ID du processeur d'événements lorsque vous l'utilisez avec PARAMETERS REMOTESERVERS ou PARAMETERS EXCLUDESERVERS.

Paramètres

```
ARIELSERVERS4EPID(processor_ID)
```

Les exemples suivants montrent comment utiliser la fonction ARIELSERVERS4EPID avec PARAMETERS REMOTESERVERS ou PARAMETERS EXCLUDESERVERS :

```
     PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPID(processor_ID)
```

```
     PARAMETERS REMOTESERVERS=ARIELSERVERS4EPID(processor_ID)
```

Exemples

Dans l'exemple suivant, seuls les résultats de recherche de `ARIELSERVERS4EPID(8)` sont inclus dans la sortie. Si l'ID processeur que vous spécifiez comme paramètre pour la fonction ARIELSERVERS4EPID n'est pas dans votre déploiement QRadar, la requête ne s'exécute pas.

```
SELECT ARIELSERVERS4EPID(8), ARIELSERVERS4EPID(11), processorid,
PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid
LAST 20 MINUTES
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPID(8)
```

Vous pouvez également utiliser la fonction ARIELSERVERS4EPID pour retourner les serveurs Ariel qui sont connectés à un processeur d'événements spécifique identifié par l'ID, comme illustré dans l'exemple suivant :

```
SELECT processorid, PROCESSORNAME(processorid),
ARIELSERVERS4EPID(processorid)
FROM events GROUP BY processorid
```

ARIELSERVERS4EPNAME

Cible

Vous utilisez la fonction ARIELSERVERS4EPNAME pour spécifier le nom du processeur d'événements lorsque vous l'utilisez avec PARAMETERS REMOTESERVERS ou PARAMETERS EXCLUDESERVERS.

Paramètres

```
ARIELSERVERS4EPNAME('eventprocessor_name')
```

Les exemples suivants montrent comment utiliser ARIELSERVERS4EPNAME PARAMETERS REMOTESERVERS ou PARAMETERS EXCLUDESERVERS :

```
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPNAME ('eventprocessor104')
```

```
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPNAME ('eventprocessor255')
```

Exemples

Dans l'exemple suivant, les enregistrements des serveurs associés à `eventprocessor104` sont exclus de la recherche.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid)
FROM events
```

```
GROUP BY logsourceid
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPNAME ('eventprocessor104')
```

Vous pouvez également utiliser la fonction pour renvoyer des serveurs Ariel associés à un processeur d'événements identifié par son nom.

```
SELECT PROCESSORNAME(processorid),
ARIELSERVERS4EPNAME(PROCESSORNAME(processorid))
FROM events GROUP BY processorid
```

Renvoie les serveurs Ariel pour le processeur d'événements nommé.

ASSETHOSTNAME

Cible

Recherche le nom d'hôte d'un actif à un moment donné.

Le domaine peut éventuellement être spécifié pour cibler un actif sur un domaine particulier.

```
ASSETHOSTNAME(sourceip)
```

```
ASSETHOSTNAME(sourceip, NOW())
```

```
ASSETHOSTNAME(sourceip, domainid)
```

Paramètres

Adresse IP (l'horodatage et l'ID de domaine sont facultatifs)

Si l'horodatage n'est pas spécifié, l'heure en cours est utilisée.

Exemples

```
SELECT ASSETHOSTNAME(destinationip, NOW())
AS 'Host Name'
FROM events
```

```
SELECT ASSETHOSTNAME(sourceip, NOW())
AS 'Host Name'
FROM events
```

Renvoie le nom d'hôte de l'actif au moment de la requête.

ASSETPROPERTY

Cible

Recherche une propriété pour un actif.

Le domaine peut éventuellement être spécifié pour cibler un actif sur un domaine particulier.

```
ASSETPROPERTY
('Unified Name', sourceIP, domainId)
```

Paramètres

Nom de la propriété, adresse IP

L'ID de domaine est facultatif

Exemple

```
SELECT
ASSETPROPERTY('Location',sourceip)
AS Asset_location,
COUNT(*)
```

```
AS 'event count'
FROM events
GROUP BY Asset_location
LAST 1 days
```

Renvoie l'emplacement de l'actif associé à l'adresse IP source.

ASSETUSER

Cible

Recherche l'utilisateur d'un actif à un moment ou un autre.

Le domaine peut éventuellement être spécifié pour cibler un actif dans un domaine spécifique.

```
ASSETUSER(sourceIP,NOW(), domainId)
```

Paramètres

Adresse IP (l'horodatage et l'ID de domaine sont facultatifs)

Si l'horodatage n'est pas spécifié, l'heure en cours est utilisée.

Exemple

```
SELECT
ASSETUSER(sourceip, now())
AS 'Username of Asset'
FROM events
```

Renvoie le nom d'utilisateur associé à l'adresse IP source.

CATEGORYNAME

Cible

Recherche le nom d'une catégorie par ID de catégorie.

```
CATEGORYNAME(Category)
```

Paramètres

Catégorie

Exemple

```
SELECT sourceip, category,
CATEGORYNAME(category)
AS 'Category name'
FROM events
```

Renvoie l'adresse IP source, l'ID de catégorie et le nom de catégorie

COMPONENTID

Cible

Extrait l'ID d'un composant avec un nom donné.

Par exemple, `ARIELSERVERS4EPNAME()` est un raccourci pour la fonction `ARIELSERVERS4EPID(COMPONENTID(<event_processor_name>))`.

Paramètres

```
COMPONENTID(<component_name>))
```

Exemple

```
SELECT  * from events where  processorid = COMPONENTID('eventprocessor0')
```

Extrait les événements du processeur d'événements nommé.

DOMAINNAME

Cible

Recherche le nom de domaine par l'ID de domaine.

```
DOMAINNAME(domainID)
```

Paramètres

ID de domaine

Exemple

```
SELECT sourceip, username,
DOMAINNAME(domainid)
AS 'Domain name'
FROM events
```

Renvoie l'adresse IP source, le nom d'utilisateur et les noms de domaine de la base de données d'événements

GLOBALVIEW

Cible

Renvoie les résultats de la base de données GLOBALVIEW pour le nom d'une recherche sauvegardée en fonction de l'intervalle entré.

Cette requête peut être exécutée uniquement à l'aide de l'API.

Pour plus d'informations sur l'accès à une base de données GLOBALVIEW, voir le *IBM Security QRadar - Guide d'administration*.

Paramètres

Recherche sauvegardée, intervalle (DAILY, NORMAL, HOURLY)

Exemple

```
SELECT *
FROM GLOBALVIEW
('Top Log Sources','DAILY')
LAST 2 days
```

GEO::LOOKUP

Cible

Renvoie les données d'emplacement, fournies par MaxMind, pour une adresse IP sélectionnée.

Paramètres

Adresse IP (obligatoire)

Chaîneaux (au moins une fois requis) :

`city`, `continent`, `physical_country`, `registered_country`, `represented_country`, `location`, `postal`, `subdivisions`, `traits`, `geo_json`

Exemple

```
SELECT sourceip, GEO::LOOKUP(sourceip, 'city')
AS GEO_CITY
FROM events last 10 minutes
```

GEO::DISTANCE

Cible

Renvoie la distance, en kilomètres, de deux adresses IP.

Paramètres

Adresse IP (deux adresses requises)

Exemple

```
SELECT GEO::DISTANCE(sourceip, destinationip)
AS GEO_DISTANCE
FROM events last 10 minutes
```

HOSTNAME

Cible

Retourne le nom d'hôte d'un processeur d'événements avec un certain processorID.

```
HOSTNAME(processorId)
```

Paramètres

ID de processeur

Exemple

```
SELECT HOSTNAME(processorId) FROM events
```

INCIDR

Cible

Filtre la sortie de l'instruction SELECT en référençant l'adresse IP CIDR source/destination spécifiée par INCIDR.

Paramètres

IP/CIDR, adresse IP

Exemple

```
SELECT sourceip, username
FROM events
WHERE INCIDR('172.16.0.0/16', sourceip)
```

Renvoie les colonnes source IP et nom d'utilisateur de la base de données de flux où l'adresse IP source CIDR provient du sous-réseau 172.16.0.0/16.

Voir d'autres exemples

INOFFENSE

Cible

Si un événement ou un flux appartient à l'infraction spécifiée, renvoie `true`.

Paramètres

ID de l'infraction

Exemple

```
SELECT * FROM events
 WHERE InOffense(123)
```

```
SELECT * FROM flows
 WHERE InOffense(123)
```

LOGSOURCENAME

Cible

Recherche le nom d'une source de journal par son ID.

```
LOGSOURCENAME(logsourceid)
```

Paramètres

ID source de journal

Exemple

```
SELECT * FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%mylogsourcename%'
```

Renvoie uniquement les résultats qui incluent `mylogsourcename` dans leur nom de source de journal.

```
SELECT LOGSOURCENAME(logsourceid)
AS Log_Source
FROM events
```

Renvoie l'alias de colonne **Log_source** qui affiche les noms des sources de journal à partir de la base de données des événements.

LOGSOURCEGROUPNAME

Cible

Recherche le nom d'un groupe de sources de journal par son ID groupe de sources de journal.

`LOGSOURCEGROUPNAME(deviceGroupList)`

Paramètres

Liste des groupes d'unités

Exemple

```
SELECT sourceip, logsourceid
FROM events
WHERE LOGSOURCEGROUPNAME(devicegrouplist)
ILIKE '%other%'
```

Renvoie l'adresse IP source et les ID source de journal pour les groupes de sources de journal qui ont « autre » en leur nom.

LOGSOURCETYPENAME

Cible

Recherche le nom d'un type de source de journal par type d'unité.

LOGSOURCETYPENAME(deviceType)

Paramètres

Type de dispositif

Exemple

```
SELECT LOGSOURCETYPENAME(devicetype)
AS 'Device names', COUNT(*)
FROM events
GROUP BY "Device names"
LAST 1 DAYS
```

Renvoie les noms des unités et le nombre d'événements.

Exemple de toutes les fonctions de sources de journal :

```
SELECT logsourceid,
LOGSOURCENAME(logsourceid)
AS 'Name of log source',
LOGSOURCEGROUPNAME(devicegrouplist)
AS 'Group Names',
LOGSOURCETYPENAME(devicetype)
AS 'Devices'
FROM events
GROUP BY logsourceid
```

Renvoie les noms des sources de journal, les noms de groupe de sources de journal et les noms d'unité de source de journal.

Lorsque vous utilisez la fonction GROUP BY, le premier élément de la liste GROUP BY est affiché dans les résultats.

MATCHESASSETSEARCH

Cible

Si l'actif est renvoyé dans les résultats de la recherche sauvegardée, il renvoie la valeur true.

```
MATCHESASSETSEARCH
('My Saved Search', sourceIP)
```

Paramètres

Nom de la recherche sauvegardée, adresse IP

Exemple

```
MATCHESASSETSEARCH
('My Saved Search', sourceIP)
```

NETWORKNAME

Cible

Recherche le nom de réseau à partir de la hiérarchie de réseau pour l'hôte qui est transmis.

```
NetworkName(sourceip)
```

Le domaine peut éventuellement être spécifié pour cibler un réseau dans un domaine particulier.

```
NETWORKNAME(sourceip, domainId)
```

Paramètres

Propriété hôte (domaine facultatif)

Exemples

```
SELECT NETWORKNAME(sourceip)
ILIKE 'servers'
AS 'My Networks'
FROM flows
```

Renvoie tous les réseaux ayant les serveurs de noms.

```
SELECT NETWORKNAME(sourceip, domainID)
ILIKE 'servers'
AS 'My Networks'
FROM flows
```

Renvoie les réseaux qui possèdent les serveurs de noms dans un domaine spécifique.

```
SELECT NETWORKNAME(sourceip)
AS 'Src Net',
NETWORKNAME(destinationip)
AS Dest_net
FROM events
```

Renvoie le nom de réseau associé aux adresses IP source et de destination.

OFFENSE_TIME

Nouveautés de la version 7.4.3, groupe de correctifs 1

Cible

Limite la requête aux heures applicables qu'une infraction peut être active.

Cette fonction augmente la vitesse de la requête.

Paramètres

ID de l'infraction

Exemple

```
SELECT * FROM events
 WHERE INOFFENSE(1) times OFFENSE_TIME(1)
```

PARAMETERS EXCLUDESERVERS

Cible

Filtre les critères de recherche en excluant les serveurs spécifiés.

Paramètres

[Server IP address:Port number]

Utilisez le port 32006 pour un processeur d'événements et le port 32011 pour une console.

Les paramètres acceptent une liste d'arguments séparés par une virgule. Exemple :

`"host1:port1,host2:port2,host3:port3"`.

Exemples

Dans l'exemple suivant, les résultats de recherche de `192.0.2.0` sont exclus. Pour exclure une console, vous devez utiliser `localhost` ou `127.0.0.1`. N'utilisez pas l'adresse IP de la console dans cette requête.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid)
from events
GROUP BY logsourceid
PARAMETERS EXCLUDESERVERS='192.0.2.0:32006'
```

Dans l'exemple suivant, les résultats de recherche de la console sont exclus :

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) FROM events
GROUP BY logsourceid start '2017-03-15 10:26'
STOP '2017-03-15 10:30'
PARAMETERS EXCLUDESERVERS='127.0.0.1:32011'
```

Dans l'exemple suivant, les résultats de recherche de la console sont exclus. La console est appelée `localhost` dans cet exemple.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid start '2017-03-15 10:25'
STOP '2017-03-15 10:30'
PARAMETERS EXCLUDESERVERS='localhost:32011'
```

L'exemple suivant utilise plusieurs arguments pour exclure les résultats de la recherche de la console et de deux autres serveurs.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid start '2017-04-15 10:25'
STOP '2017-04-15 10:30'
PARAMETERS EXCLUDESERVERS='127.0.0.1:32011,192.0.2.0:32006,172.11.22.31:32006'
```

Indiquez l'ID du processeur d'événements dans votre requête à l'aide de la fonction suivante :

```
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPID(processor_ID)
```

Affinez votre requête en utilisant `ARIELSERVERS4EPID` avec `PARAMETERS EXCLUDESERVERS` pour spécifier l'ID du processeur d'événements que vous souhaitez exclure de votre recherche. Vous pouvez spécifier un ou plusieurs ID de processeur d'événement.

Exemple

Dans l'exemple suivant, tous les résultats de `ARIELSERVERS4EPID(8)` sont exclus de la recherche.

```
SELECT processorid,
PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid
LAST 20 MINUTES
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPID(8)
```

Indiquez le nom du processeur d'événements dans votre requête à l'aide de la fonction suivante :

```
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPNAME ('processor_name')
```

Affinez votre requête en utilisant `ARIELSERVERS4EPNAME` avec `PARAMETERS EXCLUDESERVERS` pour spécifier le processeur d'événements par nom. Vous pouvez spécifier un ou plusieurs noms de processeur d'événement.

Exemple

Dans l'exemple suivant, les enregistrements des serveurs associés à `eventprocessor104` sont exclus de la recherche.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid)
FROM events
GROUP BY logsourceid
PARAMETERS EXCLUDESERVERS=ARIELSERVERS4EPNAME ('eventprocessor104')
```

PARAMETERS REMOTESERVERS

Cible

Utilisez la fonction PARAMETERS REMOTESERVERS pour limiter votre recherche à des serveurs spécifiques, ce qui accélère votre recherche car la recherche ne portera pas sur tous les hôtes.

Paramètres

[Server IP address:Port number]

Utilisez le port 32006 pour un processeur d'événements et le port 32011 pour une console.

Utilisez une liste séparée par des virgules pour plusieurs arguments, par exemple,

`"host1:port1,host2:port2,host3:port3"`.

Exemples

Dans l'exemple suivant, seul le serveur spécifié est recherché.

```
SELECT * FROM EVENTS START '2016-09-08 16:42'
STOP '2016-09-08 16:47'
PARAMETERS REMOTESERVERS='192.0.2.0:32006'
```

Dans l'exemple suivant, plusieurs serveurs sont spécifiés, ce qui inclut les résultats de la recherche à partir de la console et de deux autres serveurs.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid start '2017-04-15 10:25'
STOP '2017-04-15 10:30'
PARAMETERS REMOTESERVERS='127.0.0.1:32011,192.0.2.0:32006,172.11.22.31:32006'
```

Indiquez l'ID du processeur d'événements dans votre requête à l'aide de la fonction suivante :

```
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPID(processor_ID)
```

Affinez votre requête en utilisant ARIELSERVERS4EPID et PARAMETERS REMOTESERVERS pour spécifier l'ID du processeur d'événements que vous souhaitez inclure dans votre recherche. Vous pouvez spécifier un ou plusieurs ID de processeur d'événement.

Exemple

Dans l'exemple suivant, seuls les résultats de recherche de `ARIELSERVERS4EPID(8)` sont inclus dans la sortie.

```
SELECT ARIELSERVERS4EPID(8), ARIELSERVERS4EPID(11), processorid,
PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid) from events
GROUP BY logsourceid
LAST 20 MINUTES
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPID(8)
```

Remarque : Si l'ID processeur que vous spécifiez comme paramètre pour la fonction ARIELSERVERS4EPID n'est pas dans votre déploiement QRadar, la requête ne s'exécute pas.

Indiquez le nom du processeur d'événements dans votre requête à l'aide de la fonction suivante :

```
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPNAME ('eventprocessor_name')
```

Affinez votre requête en utilisant ARIELSERVERS4EPNAME et PARAMETERS REMOTESERVERS pour spécifier le nom du processeur d'événements que vous souhaitez inclure dans votre recherche. Vous pouvez spécifier un ou plusieurs noms de processeur d'événement.

Exemple

Dans l'exemple suivant, seuls les enregistrements de recherche associés à `eventprocessor104` sont inclus dans les résultats de la recherche.

```
SELECT processorid,PROCESSORNAME(processorid),
LOGSOURCENAME(logsourceid)
FROM events
GROUP BY logsourceid
PARAMETERS REMOTESERVERS=ARIELSERVERS4EPNAME ('eventprocessor104')
```

PROCESSORNAME

Cible

Renvoie le nom d'un processeur par ID processeur.

```
PROCESSORNAME(processorid)
```

Paramètres

Numéro d'ID processeur

Exemple

```
SELECT sourceip, PROCESSORNAME(processorid)
AS 'Processor Name'
FROM events
```

Renvoie l'adresse IP source et le nom du processeur de la base de données des événements.

Exemple

```
SELECT processorid, PROCESSORNAME(processorid)
FROM events WHERE processorid=104
GROUP BY processorid LAST 5 MINUTES
```

Renvoie les résultats de processeur d'événements dont l'ID processeur est égal à 104.

PROTOCOLNAME

Cible

Renvoie le nom d'un protocole par l'ID de protocole

Paramètres

Numéro d'ID de protocole

Exemple

```
SELECT sourceip, PROTOCOLNAME(protocolid)
AS 'Name of protocol'
FROM events
```

Renvoie l'adresse IP source et le nom du protocole à partir de la base de données des événements.

QIDNAME

Cible

Recherche le nom d'un QID par son QID.

```
QIDNAME(qid)
```

Paramètres

QID

Exemple

```
SELECT QIDNAME(qid)
AS 'My Event Names', qid
FROM events
```

Renvoie le nom QID et le numéro QID.

QIDESCRIPTION

Cible

Recherche la description QID par son QID.

```
QIDDESCRIPTION(qid)
```

Paramètres

QID

Exemple

```
SELECT QIDDESCRIPTION(qid)
AS 'My_Event_Names', QIDNAME(qid)
AS 'QID Name'
FROM events
```

Renvoie la description QID et le nom QID.

REFERENCEMAP

Cible

Recherche la valeur d'une clé dans une mappe de référence.

```
ReferenceMap('Value',Key,domainID)
```

Bien que le `domainID` soit facultatif, dans un environnement activé par domaine, la recherche est limitée aux seules données de référence partagées lorsque `domainID` est exclu.

Paramètres

Chaîne, Chaîne, Entier

Exemple

```
SELECT
REFERENCEMAP('Full_name_lookup', username, 5)
AS Name_of_User
FROM events
```

Recherche la clé `userName` (clé) dans la mappe de référence `Full_name_lookup` dans le domaine spécifié et renvoie le Nom complet (valeur) du nom d'utilisateur (clé).

REFERENCEMAPSETCONTAINS

Cible

Si une valeur existe pour une clé dans une mappe de référence d'ensembles, pour un domaine, elle renvoie `true`.

```
REFERENCEMAPSETCONTAINS(MAP_SETS_NAME, KEY, VALUE)
```

Paramètres

Chaîne, Chaîne, Chaîne

Exemple

```
ReferenceMapSetContains('RiskyUsersForIps','sourceIP','userName')
```

REFERENCETABLE

Cible

Recherche la valeur d'une clé de colonne dans une table identifiée par une clé de table dans une collection de tables de référence spécifique.

```
REFERENCETABLE
('testTable','value','key', domainID)
or
REFERENCETABLE
('testTable','value','key' domainID)
```

Bien que le `domainID` soit facultatif, dans un environnement activé par domaine, la recherche est limitée aux seules données de référence partagées lorsque `domainID` est exclu.

Paramètres

Chaîne, Chaîne, Chaîne (ou adresse IP), Entier

Exemple

```
SELECT
REFERENCETABLE('user_data','FullName',username, 5)
AS 'Full Name',
REFERENCETABLE('user_data','Location',username, 5)
AS Location,
REFERENCETABLE('user_data','Manager',username, 5)
AS Manager
FROM events
```

Renvoie le nom complet (valeur), l'emplacement (valeur) et le gestionnaire (valeur) de `username` (clé) à partir de `user_data`.

Voir plus d'exemples de données de référence

REFERENCESETCONTAINS

Cible

Si une valeur est incluse dans un jeu de références spécifique, renvoie `true`.

```
REFERENCESETCONTAINS
('Ref_Set', 'value', domainID)
```

Bien que le `domainID` soit facultatif, dans un environnement activé par domaine, la recherche est limitée aux seules données de référence partagées lorsque `domainID` est exclu.

Paramètres

Chaîne, Chaîne, Entier

Exemple

```
SELECT
ASSETUSER(sourceip, NOW())
AS 'Source Asset User'
FROM flows
WHERE
REFERENCESETCONTAINS('Watchusers', username, 5)
GROUP BY "Source Asset User"
LAST 24 HOURS
```

Renvoie l'utilisateur de l'actif lorsque `username` (valeur) est inclus dans le jeu de références `Watchusers`.

RULENAME

Cible

Renvoie un ou plusieurs noms de règle qui sont basés sur l'ID ou les ID de règle transmis.

```
RULENAME(creeventlist)
```

```
RULENAME(3453)
```

Paramètres

ID de règle unique ou liste d'ID de règle.

Exemple

```
SELECT * FROM events
WHERE RULENAME(creEventList)
ILIKE '%my rule name%'
```

Renvoie les événements qui déclenchont un nom de règle spécifique.

```
SELECT RULENAME(123)
FROM events
```

Renvoie le nom de la règle par l'ID de règle.

Critères de temps dans les requêtes AQL

Définissez des intervalles de temps dans vos requêtes AQL à l'aide des clauses START et STOP, ou utilisez la clause LAST pour les références de temps relatives.

Définir les paramètres d'heure transmis à la requête AQL

L'instruction SELECT prend en charge une option `arieltime` qui remplace les paramètres de temps.

Vous pouvez limiter la période pour laquelle une requête AQL est évaluée à l'aide des clauses et fonctions suivantes :

- «START», à la page 44
- «STOP», à la page 44
- «LAST», à la page 45
- «NOW», à la page 46
- «PARSEDATETIME», à la page 46

START

Vous pouvez passer un intervalle de temps à START en sélectionnant des données (à partir du moment), dans les formats suivants :

```
yyyy-MM-dd HH:mm
yyyy-MM-dd HH:mm:ss
yyyy/MM/dd HH:mm:ss
yyyy/MM/dd-HH:mm:ss
yyyy:MM:dd-HH:mm:ss
```

timezone est représenté par 'z ou Z' aux formats suivants :

```
yyyy-MM-dd HH:mm'Z'
```

```
yyyy-MM-dd HH:mm'z'
```

Utilisez START en combinaison avec STOP.

Exemples

```
SELECT *
FROM events WHERE userName IS NULL
START '2014-04-25 15:51'
STOP '2014-04-25 17:00'
```

Renvoie les résultats de : 2014-04-25 15:51:00 à 2014-04-25 16:59:59

```
SELECT *
FROM events WHERE userName IS NULL
START '2014-04-25 15:51:20'
STOP '2014-04-25 17:00:20'
```

Renvoie les résultats de : 2014-04-25 15:51:00 à 2014-04-25 17:00:59

```
SELECT * from events
START PARSEDATETIME('1 hour ago')
STOP PARSEDATETIME('now')
```

STOP est facultatif. Si vous ne l'incluez pas dans la requête, l'heure STOP est = now

STOP

Vous pouvez passer un intervalle de temps à STOP en sélectionnant les données (heure de fin), dans les formats suivants :

```
yyyy-MM-dd HH:mm
yyyy-MM-dd HH:mm:ss
yyyy/MM/dd HH:mm:ss
yyyy/MM/dd-HH:mm:ss
yyyy:MM:dd-HH:mm:ss
```

timezone est représenté par 'z ou Z' aux formats suivants :

```
yyyy-MM-dd HH:mm'Z'
```

```
yyyy-MM-dd HH:mm'z'
```

Utilisez STOP en combinaison avec START.

Exemples

```
SELECT * FROM events
WHERE username IS NULL
START  '2016-04-25 14:00'
STOP '2016-04-25 16:00'
```

```
SELECT * FROM events
WHERE username IS NULL
START '2016-04-25 15:00:30'
STOP '2016-04-25 15:02:30'
```

Utilisez n'importe quel format avec la fonction PARSEDATETIME, par exemple,

```
SELECT *
FROM events
START PARSEDATETIME('1 day ago')
```

Même si STOP n'est pas inclus dans cette requête, l'heure STOP est = now.

```
Select * FROM events
START PARSEDATETIME('1 hour ago')
STOP PARSEDATETIME('now')
```

```
SELECT * FROM events
START PARSEDATETIME('1 day ago')
```

```
Select *
FROM events
WHERE logsourceid = '69'
START '2016-06-21 15:51:00'
STOP '2016-06-22 15:56:00'
```

LAST

Vous pouvez passer un intervalle de temps à la clause LAST pour indiquer une heure spécifique à partir de la sélection de données.

Les intervalles valides sont MINUTES, HOURS et DAYS.

Exemples

```
SELECT * FROM events
LAST 15 MINUTES
```

```
SELECT * FROM events
LAST 2 DAYS
```

```
SELECT * from events
WHERE userName ILIKE '%dm%'
LIMIT 10
LAST 1 HOURS
```

Remarque : Si vous utilisez une clause LIMIT dans votre requête, vous devez la placer avant les clauses START et STOP, par exemple,

```
SELECT *
FROM events
LIMIT 100
START '2016-06-28 10:00'
STOP '2016-06-28 11:00'
```

Fonctions de temps

Utilisez les fonctions d'heure suivantes pour spécifier l'heure d'analyse de la requête.

NOW

Cible

Renvoie l'heure en cours qui est exprimée en millisecondes depuis l'heure UTC (Temps Universel Coordonné) 00:00:00 le 1er janvier 1970.

Exemple

```
SELECT ASSETUSER(sourceip, NOW())
AS 'Asset user' FROM events
```

Trouvez l'utilisateur de l'actif à ce moment dans le temps (NOW).

PARSEDATETIME

Cible

Transmettez une valeur de temps à l'analyseur syntaxique, par exemple `PARSEDATETIME('time reference')`. Ce `'time reference'` est l'heure d'analyse de la requête.

Exemple

```
SELECT * FROM events
START PARSEDATETIME('1 hour ago')
```

Formats de date et d'heure d'AQL

Utilisez les formats de date et d'heure Ariel Query Language (AQL) pour représenter les heures et les dates dans les requêtes.

Le tableau suivant répertorie les lettres qui représentent la date et l'heure dans les requêtes AQL. Cette table est basée sur le *SimpleDateFormat*.

Tableau 7. Formats de date et d'heure

Lettre	Paramètre de date ou d'heure	Présentation	Exemples

y	Année civile	Année
		L'exemple de date utilisé est : 20-juin-2016
			`DATEFORMAT(starttime,'yy-MM-dd')`
			Format de date de retour : 16-06-20
			`DATEFORMAT(starttime,'yyyy-MM-dd')`
			Format de date de retour : 2016-06-20
			`SELECT DATEFORMAT(devicetime,'yyyy-MM-dd') AS Log_Src_Date, QIDDESCRIPTION(qid) AS 'Event Name' FROM events`

Y	Semaine de la semaine	Année
		Les premiers et derniers jours d'une semaine peuvent avoir des valeurs différentes pour l'année civile.
		L'exemple de date utilisé est : 20-juin-2016
			`DATEFORMAT(starttime,'YY-MM-dd')`
			Format de date de retour : 16-06-20
			`DATEFORMAT(starttime,'YYYY-MM-dd')`
			Format de date de retour : 2016-06-20
			`SELECT DATEFORMAT(starttime,'YYYY-MM-dd hh:mm') AS 'Start Time', DATEFORMAT(endtime,'YYYY-MM-dd hh:mm') AS Storage_time, QIDDESCRIPTION(qid) AS 'Event Name' FROM events`
			Renvoie les colonnes de début, de temps de stockage et de nom d'événement

Tableau 7. Formats de date et d'heure (suite)

Lettre	Paramètre de date ou d'heure	Présentation	Exemples

M	Mois en année	Mois
		3 lettres ou plus sont interprétées comme du texte.
		2 lettres sont interprétées comme un nombre.
		L'exemple de date utilisé est : 20-juin-2016
			`DATEFORMAT(starttime,'yyyy-MMMM-dd')`
			Format de date de retour : 2016-juin-20
			`DATEFORMAT(starttime,'yyyy-MMM-dd')`
			Format de date de retour : 2016-juin-20
			`DATEFORMAT(starttime,'yyyy-MM-dd')`
			Format de date de retour : 2016-06-20

w	Semaine en année	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-ww-dd')` Format de date de retour : 2016-26-20 **Remarque :** 26 est la semaine 26 en année
W	Semaine en mois	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-WW-dd')` Format de date de retour : 2016-04-20 **Remarque :** 04 est la semaine 4 du mois
D	Jour de l'année	Numéro Jour de l'année représenté par le nombre L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-mm-DD')` Format de date de retour : 2016-06-172 **Remarque :** 172 est le nombre de jours 172 dans l'année
d	Jour du mois	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-mm-dd')` Format de date de retour : 2016-06-20
F	Jour de la semaine du mois	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-MM-FF')` Format de date de retour : 2016-06-03 **Remarque :** 03 est le jour 3 de la semaine du mois
E	Nom de la journée dans la semaine	Texte L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-MM-EE')` Format de date de retour : 2016-06-Lun
a	AM (matin) ou PM (après-midi)	Texte L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-MM-dd h a')` 2016-06-20 06 PM
H	Heure du jour (0-23)	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-MM-dd H')` Format de date de retour : 2016-06-20 18 **Remarque :** 18 heures : 18 heures
k	Heure du jour (1-24)	Numéro L'exemple de date utilisé est : 20-juin-2016	`DATEFORMAT(starttime,'yyyy-MM-dd k')` Format de date de retour : 2016-06-20 18 **Remarque :** 18 heures : 18 heures
K	Heure dans AM/PM (0-11)	Numéro L'exemple de date utilisé est : 20-juin-2016, 6 PM	`DATEFORMAT(starttime,'yyyy-MM-dd K a')` Format de date de retour : 2016-06-20 6 PM **Remarque :** K = 6 et a = PM
h	Heure à AM/PM (1-12)	Numéro Exemple de date utilisé : 20-juin-2016 6 PM	`DATEFORMAT (starttime,'yyyy-MM-dd h a')` Format de date de retour : 2016-06-20 6 PM **Remarque :** H = 6 et a = PM
m	Minute en heure	Numéro L'exemple de date utilisé est : 20-juin-2016, 6:10 PM	`DATEFORMAT(starttime,'yyyy-MM-dd h:m a')` Format de date de retour : 2016-06-20 6:10 PM **Remarque :** Deux-points ajoutés dans la requête pour formater l'heure
s	Seconde en minutes	Numéro L'exemple de date utilisé est : 20-juin-2016, 6:10:56 PM	`DATEFORMAT(starttime,'yyyy-MM-dd h:m:s a')` Format de date de retour : 2016-06-20 6:10:56 PM **Remarque :** Deux points ajoutés dans la requête pour formater l'heure

Tableau 7. Formats de date et d'heure (suite)

Lettre	Paramètre de date ou d'heure	Présentation	Exemples
S	Millisecondes	Numéro L'exemple de date utilisé est : 20-juin-2016, 6:10 PM	`DATEFORMAT(starttime,'yyyy-MM-dd h:m:ss:SSS a')` Format de date de retour : 2016-06-20 6:10:00:322 PM **Remarque :** Deux points ajoutés dans la requête pour formater l'heure
z	Fuseau horaire	Fuseau horaire général L'exemple de date utilisé est : 20-juin-2016, 6:10 PM GMT + 1	`DATEFORMAT(starttime,'yyyy-MM-dd h:m a z')` Format de date de retour : 2016-06-20 6:10 PM GMT + 1 **Remarque :** Deux-points ajoutés dans la requête pour formater l'heure
Z	Fuseau horaire	Fuseau horaire RFC 822 L'exemple de date utilisé est : 20-juin-2016, 6:10 PM GMT + 1	`DATEFORMAT(starttime,'yyyy-MM-dd h:m a Z')` Format de date de retour : 2016-06-20 6:10 PM + 0100 **Remarque :** Deux-points ajoutés dans la requête pour formater l'heure
X	Fuseau horaire	Fuseau horaire ISO 8601 L'exemple de date utilisé est : 20-juin-2016, 6:10 PM GMT + 1	`DATEFORMAT(starttime,'yyyy-MM-dd h:m a X')` Format de date de retour : 2016-06-20 6:10 PM + 01 **Remarque :** Deux-points ajoutés dans la requête pour formater l'heure

Sous-requête AQL

Utilisez une sous-requête AQL comme source de données référencée ou recherchée par la requête principale. Utilisez la clause FROM ou IN pour affiner votre requête AQL en faisant référence aux données qui sont extraites par la sous-requête.

Une *sous-requête* est une requête imbriquée ou interne qui est référencée par la requête principale. La sous-requête est disponible dans les formats suivants :

- SELECT <*field/s*> FROM (<*expression de requête AQL*>)

  Cette requête utilise la clause FROM pour rechercher la sortie (curseur) de la sous-requête.
- SELECT <*field/s*> FROM events WHERE <*field*> IN (<*expression de requête AQL*>)

  Cette requête utilise la clause IN pour indiquer les résultats de sous-requête qui correspondent aux valeurs de la recherche de sous-requête. Cette sous-requête renvoie uniquement une seule colonne. Vous pouvez spécifier la limite de résultats, mais le maximum est de 10 000 résultats.

Exemples de sous-requête

L'instruction SELECT imbriquée est entre parenthèses dans la sous-requête. La sous-requête est exécutée en premier et elle fournit les données qui sont utilisées par la requête principale. L'instruction SELECT de la requête principale extrait les noms d'utilisateur de la sortie (curseur) de la sous-requête.

```
SELECT username FROM
(SELECT * FROM events
WHERE username IS NOT NULL
LAST 60 MINUTES)
```

La requête suivante renvoie des enregistrements dans lesquels le nom d'utilisateur de la base de données Ariel correspond aux valeurs de la sous-requête.

```
SELECT * FROM events
WHERE username IN
(SELECT username FROM events
LIMIT 10 LAST 5 MINUTES) LAST 24 HOURS
```

La requête suivante renvoie des enregistrements dans lesquels l'adresse IP source de la base de données Ariel correspond à l'adresse IP de destination de la sous-requête.

```
SELECT * FROM EVENTS
WHERE sourceip IN
(SELECT destinationip FROM events)
```

La requête suivante renvoie des enregistrements dans lesquels l'adresse IP source de la base de données Ariel correspond aux adresses IP source qui sont renvoyées dans la sous-requête. La sous-requête filtre les données pour l'instruction select principale en localisant les hôtes internes qui ont interagi avec des entités à risque élevé. La requête renvoie les hôtes qui ont communiqué avec des entité à risque élevé.

```
SELECT sourceip AS 'Risky Hosts' FROM events
WHERE destinationip IN (SELECT sourceip FROM events
WHERE eventdirection = 'L2R'
AND REFERENCESETCONTAINS('CriticalWatchList', destinationip)
GROUP BY sourceip)
GROUP BY sourceip last 24 hours
```

Regroupement d'événements associés en sessions

Les événements de groupe qui sont liés au contexte dans les sessions où vous pouvez observer les séquences d'événements et les résultats de ces séquences d'événements. Découvrez l'activité utilisateur et l'activité réseau en observant la séquence des événements qui se produisent dans une session.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser des événements pour vous indiquer ce qu'un utilisateur a fait à un moment précis, mais vous pouvez utiliser des sessions transactionnelles pour vous indiquer ce que l'utilisateur a fait avant et après un événement. Les transactions vous donnent des détails complets, comme un achat sur Internet, ou une tentative de connexion non autorisée.

L'ID de session est unique et est affecté aux événements de la même session. Vous définissez la session en fonction de paramètres tels que le temps, le nom d'utilisateur, la connexion ou tout autre critère. Vous utilisez la clause SESSION BY pour créer les sessions uniques.

Par exemple, utilisez les sessions transactionnelles pour effectuer les tâches suivantes :

- Définissez une activité utilisateur basée sur des événements d'accès Web incluant une combinaison unique d'activités.
- Regroupage les événements par une session de comportement utilisateur spécifique telle que les visites de site Web, les téléchargements ou les courriers électroniques envoyés.
- Enregistrez lorsque les utilisateurs se connectant et se déconnectant de votre réseau, et pendant combien de temps ils se connectant pour. La déconnexion ferme la transaction associée lancée par la connexion.
- Sélectionnez une activité que vous souhaitez suivre et définissez les critères de l'activité de session.

Procédure

1. Pour créer des sessions, utilisez la clause SESSION BY en utilisant le format suivant.

   `SESSION BY <TimeExpression> <AQL_expression_list> BEGIN <booleanExpression> END <booleanExpression>`

   Le tableau suivant décrit les paramètres de session.

Tableau 8.

Paramètres de session	**Description**
Time <*TimeExpression*>	Durée
<*AQL_expression_list*>	Liste d'expressions AQL
BEGIN <*booleanExpression*>	Démare une nouvelle session
END <*booleanExpression*>	La clause END est facultative et est utilisée pour terminer la session.

Le SessionId change lorsqu'une valeur d'expression AQL change ou lorsque le paramètre BEGIN ou END *booleanExpression* est TRUE.

2. Pour tester un exemple, procédez comme suit :
   a) Pour accéder à la page IBM QRadar **Documentation de l'API**, dans le menu **Aide**, cliquez sur **API interactive pour les développeurs**.
   b) Cliquez sur **8.0** ou sur la version la plus haute pour développer le menu.
   c) Cliquez sur **/ariel** > **/searches**.
   d) Cliquez sur l'onglet **Publication**.
   e) Entrez votre requête AQL dans la zone **Valeur** pour le paramètre **query_expression**.
   Exemple :

```
Select sessionID, DATEFORMAT(starttime, 'YYYY-MM-dd HH:mm:ss')
start_time, username, sourceip, category from events
into <your_Cursor_Name> where username is not null
SESSION BY starttime username, sourceip
BEGIN category=16001
start '2016-09-14 14:20' stop '2016-09-14 14:50'
```

   <*your_cursor_name*> est un nom que vous souhaitez utiliser pour la sortie de résultats.
   f) Cliquez sur **Try it out**.

   Si la requête s'exécute sans erreur, le code de réponse est 201.
   g) Cliquez sur **/ariel** > **/ searches >** > **/{search_id}** > **/results**

   La page **8.0 - GET - /ariel/searches/{search_id}/results** s'ouvre.
   h) Dans la zone **Valeur** du paramètre **search_id**, entrez <*votre_nom_curseur*>.
   i) Sélectionnez **text/table** pour le type Mime.
   j) Cliquez sur **Try it out**.

Tableau 9. Résultats de la requête

sessionId	start_time	username	IP de la source	category
1	2016-09-14 14:42:03	admin	9.23.121.97	16003
1	2016-09-14 14:42:09	admin	9.23.121.97	16003
2	2016-09-14 14:42:10	admin	127.0.0.1	16003
2	2016-09-14 14:42:11	admin	127.0.0.1	16003
3	2016-09-14 14:42:27	joe_blogs	9.23.121.98	16001
4	2016-09-14 14:44:11	joe_blogs	9.23.121.98	16001
5	2016-09-14 14:44:35	root	127.0.0.1	4017
5	2016-09-14 14:44:35	root	127.0.0.1	3014
5	2016-09-14 14:44:55	root	127.0.0.1	4017
5	2016-09-14 14:44:55	root	127.0.0.1	3014

Les catégories représentent des activités spécifiques dans vos journaux d'événements. Une nouvelle session est démarrée pour chaque modification du nom d'utilisateur et des valeurs d'adresse IP source, par exemple, voir `sessionid` 2 et `sessionid` 5.

De plus, une nouvelle session est créée pour la catégorie 16001, qui se produit dans `sessionid` 3 et `sessionid` 4.

Exemple

Dans cet exemple, les événements sont renvoyés et regroupés par ID session unique, où l'utilisateur joe_blogs se connecte et démarre un processus entre 4 PM et 11:30 le 25 novembre.

```
select sessionId,DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss')
start_time,username,sourceip,category from events into <cursor_name>
where username='joe_blogs'
SESSION BY starttime username, sourceip
BEGIN category=16001
END category=16003
start '2016-11-25 16:00'
stop '2016-11-25 23:30'
```

Une session est démarrée lorsque vous obtenez un événement où l'expression BEGIN est satisfaite OU l'événement précédent termine la session.

Une session est arrêtée lorsque vous obtenez un événement où l'expression END est vraie OU l'événement suivant lance une nouvelle session.

La catégorie d'événement 16001 indique un événement de connexion ou de déconnexion de l'utilisateur sur la console, et la catégorie d'événement 16003 indique qu'un utilisateur a lancé un processus, comme le démarrage d'une sauvegarde ou la génération d'un rapport. Pour obtenir la liste des catégories d'événements, voir IBM QRadar *Administration Guide*.

Affinements de requêtes transactionnelles

Affinez les requêtes AQL transactionnelles à l'aide de l'expression EXPLICIT avec les expressions BEGIN et END. Utilisez également les expressions TIMEOUT et TIMEWINDOW pour spécifier des intervalles de temps.

Utilisez l'expression EXPLICIT avec les expressions BEGIN et END pour appliquer un filtrage plus précis à vos requêtes transactionnelles.

Par exemple, vous pouvez utiliser l'expression BEGIN avec l'expression EXPLICIT END pour capturer plusieurs tentatives de connexion infructueuses (BEGIN), suivies d'une connexion réussie (EXPLICIT END).

Utilisez les expressions TIMEOUT et TIMEWINDOW pour appliquer des filtres de temps aux sessions dans vos requêtes transactionnelles.

Expressions

Utilisez les expressions décrites dans la section suivante pour affiner votre requête AQL transactionnelle :

Tableau 10. Expressions de requête transactionnelle AQL

Expressions de requête	Description
BEGIN	Une session est démarrée lorsque vous obtenez un événement où l'expression BEGIN est satisfaite ou que l'événement précédent met fin à la session.
EXPLICIT BEGIN	Lance une nouvelle session uniquement si l'expression EXPLICIT BEGIN est vraie.
END	Une session est arrêtée lorsque vous obtenez un événement où l'expression END est vraie ou que l'événement suivant démarre une nouvelle session.
EXPLICIT END	Ferme la session en cours uniquement si l'expression EXPLICIT END est vraie.

Tableau 10. Expressions de requête transactionnelle AQL (suite)	
Expressions de requête	**Description**
TIMEOUT	Ferme la session lorsque la période TIMEOUT spécifiée s'écoule à partir du moment où l'événement précédent s'est produit au moment où l'événement en cours s'est produit.
TIMEWINDOW	Effectue le suivi de la session. Ferme la session lorsque la période TIMEWINDOW spécifiée s'écoule à partir du moment où le premier événement s'est produit au moment où l'événement en cours s'est produit.

Syntaxe

```
SESSION BY
<TimeExpression> <ExpressionList>
[EXPLICIT] BEGIN <booleanExpression>
[EXPLICIT] END <booleanExpression>
TIMEOUT <IntegerLiteral millieseconds>
TIMEWINDOW <IntegerLiteral SECONDS|MINUTES|HOURS|DAYS>
```

Les exemples suivants illustrent les exemples de résultats que vous obtenez en utilisant différentes combinaisons des expressions de requête disponibles:

Expressions BEGIN et END

Une expression BEGIN démarre une session lorsqu'un événement correspond à l'expression BEGIN ou que l'événement précédent termine la session.

Une expression END termine une session lorsque l'expression END est vraie pour un événement ou que l'événement suivant démarre une nouvelle session.

En utilisant l'expression EXPLICIT avec les expressions BEGIN et END, vous appliquez un filtre plus précis qui affine l'ensemble de résultats.

Voir les exemples de requêtes et de résultats suivants.

L'exemple de requête suivant utilise les expressions BEGIN et END.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss')
start_time, username, sourceip,
category from events into TR1
where username = 'user_x'
SESSION BY starttime username, sourceip
BEGIN category=16001
END category=16003
start '2016-12-10 16:00' stop '2016-12-10 23:30'
```

La catégorie d'événement 16001 indique un événement de connexion ou de déconnexion de l'utilisateur sur la console, et la catégorie d'événement 16003 indique qu'un utilisateur a lancé un processus, comme le démarrage d'une sauvegarde ou la génération d'un rapport.

Le tableau suivant présente les résultats de la requête utilisant BEGIN et END.

Tableau 11. Résultats de la requête BEGIN et END

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06	user_x	10.2.2.10	16001
1	2016-12-10 16:00:06	user_x	10.2.2.10	16003
2	2016-12-10 16:00:06	user_x	10.2.2.10	16003
3	2016-12-10 16:00:10	user_x	10.2.2.10	16001

Tableau 11. Résultats de la requête BEGIN et END (suite)

sessionId	start_Time	user name	IP de la source	category
3	2016-12-10 16:00:10	user_x	10.2.2.10	16003
4	2016-12-10 16:00:11	user_x	10.2.2.10	16003
3	2016-12-10 16:00:11	user_x	10.2.2.10	16001
3	2016-12-10 16:00:11	user_x	10.2.2.10	16003

Remarque : `Sessionid` 2 est constitué d'un seul événement qui le ferme (catégorie 16003). Une session ayant un événement est une exception et peut se produire.

Expressions EXPLICIT BEGIN et END

Les événements sont ignorés lorsqu'une session n'est pas démarrée et qu'un événement n'est pas un événement `EXPLICIT BEGIN`.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss')
start_time, username, sourceip,
category from events into TR2
where username='user_x'
SESSION BY starttime username, sourceip
EXPLICIT BEGIN category=16001
END category=16003 start '2016-12-10 16:00'
stop '2016-12-10 23:30'
```

Le tableau suivant présente les résultats de la requête qui utilise EXPLICIT BEGIN et END.

Tableau 12. Résultats de la requête EXPLICIT BEGIN et END

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06	user_x	10.2.2.10	16001
1	2016-12-10 16:00:06	user_x	10.2.2.10	16003
2	2016-12-10 16:00:07	user_x	10.2.2.10	16001
2	2016-12-10 16:00:07	user_x	10.2.2.10	16003
3	2016-12-10 16:00:11	user_x	10.2.2.10	16001
3	2016-12-10 16:00:11	user_x	10.2.2.10	16003
3	2016-12-10 16:00:11	user_x	10.2.2.10	16003
4	2016-12-10 16:00:14	user_x	10.2.2.10	16001
5	2016-12-10 16:00:15	user_x	10.2.2.10	16001
5	2016-12-10 16:00:15	user_x	10.2.2.10	16003

Seuls les événements répondant à l'expression `EXPLICIT BEGIN` sont renvoyés.

`Sessionid` 2 et `Sessionid` 4 dans EXPLICIT BEGIN et END ne répondent pas à l'expression `EXPLICIT BEGIN`.

BEGIN et EXPLICIT END

Fermez la session en cours uniquement si l'expression EXPLICIT END est vraie. Il n'y a plus de contrôle des événements BEGIN dans la session lorsque l'expression EXPLICIT END est vraie.

Plusieurs événements BEGIN dans une même session peuvent être associés à une expression `EXPLICIT END`. Par exemple, vous pouvez utiliser l'expression `EXPLICIT END` pour compter plusieurs tentatives

de connexion ayant échoué, suivies d'une connexion réussie au cours d'un intervalle de temps spécifique (délai d'attente de session).

L'exemple de requête suivant utilise les expressions BEGIN et EXPLICIT END.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss')
start_time, username, sourceip,
category from events into TR3
where username = 'user_x'
SESSION BY starttime username, sourceip
BEGIN category=16001
EXPLICIT END category=16003
start '2016-12-10 16:00'
stop '2016-12-10 23:30'
```

Le tableau suivant présente les résultats de la requête utilisant les expressions BEGIN et EXPLICIT END.

Tableau 13. Résultats de la requête BEGIN et EXPLICIT END

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06	user_x	10.2.2.10	16001
1	2016-12-10 16:00:06	user_x	10.2.2.10	16003
2	2016-12-10 16:00:07	user_x	10.2.2.10	16003
2	2016-12-10 16:00:10	user_x	10.2.2.10	16001
2	2016-12-10 16:00:10	user_x	10.2.2.10	16003
3	2016-12-10 16:00:11	user_x	10.2.2.10	16001
3	2016-12-10 16:00:11	user_x	10.2.2.10	16003
4	2016-12-10 16:00:12	user_x	10.2.2.10	16003
4	2016-12-10 16:00:12	user_x	10.2.2.10	16001
4	2016-12-10 16:00:12	user_x	10.2.2.10	16003
5	2016-12-10 16:00:13	user_x	10.2.2.10	16001
4	2016-12-10 16:00:11	user_x	10.2.2.10	16003

EXPLICIT BEGIN et EXPLICIT END

Les événements sont ignorés lorsqu'une session n'est pas démarrée et qu'un événement n'est pas un événement EXPLICIT BEGIN.

Fermez la session en cours uniquement si l'expression EXPLICIT END est vraie. Il n'y a plus de contrôle des événements BEGIN dans la session lorsque l'expression EXPLICIT END est vraie.

L'exemple de requête suivant utilise les expressions EXPLICIT BEGIN et EXPLICIT END.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss')
start_time, username, sourceip,
category from events into TR4
where username = 'user_x'
SESSION BY starttime username, sourceip
EXPLICIT BEGIN category=16001
EXPLICIT END category=16003
start '2016-12-10 16:00'
stop '2016-12-10 23:30'
```

Le tableau suivant présente les résultats de la requête qui utilise les expressions EXPLICIT BEGIN et EXPLICIT END.

Tableau 14. Résultats de la requête EXPLICIT BEGIN et EXPLICIT END

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06	user_x	10.2.2.10	16001
1	2016-12-10 16:00:06	user_x	10.2.2.10	16003
2	2016-12-10 16:00:10	user_x	10.2.2.10	16001
2	2016-12-10 16:00:10	user_x	10.2.2.10	16003
3	2016-12-10 16:00:11	user_x	10.2.2.10	16001
3	2016-12-10 16:00:12	user_x	10.2.2.10	16001
3	2016-12-10 16:00:12	user_x	10.2.2.10	16003
4	2016-12-10 16:00:13	user_x	10.2.2.10	16001
4	2016-12-10 16:00:14	user_x	10.2.2.10	16001
4	2016-12-10 16:00:14	user_x	10.2.2.10	16003
5	2016-12-10 16:00:15	user_x	10.2.2.10	16001
5	2016-12-10 16:00:15	user_x	10.2.2.10	16003

TIMEOUT

Ferme la session lorsque la période TIMEOUT spécifiée s'écoule à partir du moment où l'événement précédent s'est produit au moment où l'événement en cours s'est produit. L'événement en cours fait partie d'une nouvelle session. La valeur TIMEOUT est spécifiée en millisecondes.

L'exemple de requête suivant utilise l'expression TIMEOUT.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss.SSS')
start_time, username, sourceip,
category from events into TR5
where username='user_x'
SESSION BY starttime username, sourceip
BEGIN category=16001
EXPLICIT END category=16003
TIMEOUT 3600
start '2016-12-10 16:00'
stop '2016-12-10 23:30'
```

Le tableau suivant présente les résultats de la requête qui utilise l'expression TIMEOUT.

Tableau 15. Résultats de la requête TIMEOUT

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06.716	user_x	10.2.2.10	16003
2	2016-12-10 16:00:10.328	user_x	10.2.2.10	16001

`Sessionid` 1 est arrêté et `sessionid` 2 est démarré, car le TIMEOUT de 3600 est dépassé.

TIMEWINDOW

Effectue le suivi de la session. Ferme la session lorsque la période TIMEWINDOW spécifiée s'écoule à partir du moment où le premier événement s'est produit au moment où l'événement en cours s'est produit. L'événement en cours fait partie d'une nouvelle session. La valeur TIMEWINDOW peut être spécifiée en secondes, minutes, heures ou jours.

L'exemple de requête suivant utilise l'expression TIMEWINDOW.

```
Select sessionId,
DATEFORMAT(starttime,'YYYY-MM-dd HH:mm:ss.SSS')
start_time, username, sourceip,
category from events into TR6
where username='user_x'
SESSION BY starttime username, sourceip
BEGIN category=16001
EXPLICIT END category=16003
TIMEWINDOW 3000
start '2016-12-10 16:00'
stop '2016-12-10 23:30'
```

Le tableau suivant présente les résultats de la requête qui utilise l'expression TIMEWINDOW.

Tableau 16. Résultats de la requête TIMEWINDOW

sessionId	start_Time	user name	IP de la source	category
1	2016-12-10 16:00:06.415	user_x	10.2.2.10	16001
1	2016-12-10 16:00:06.433	user_x	10.2.2.10	16003
2	2016-12-10 16:00:06.716	user_x	10.2.2.10	16003
3	2016-12-10 16:00:10.328	user_x	10.2.2.10	16001
3	2016-12-10 16:00:06.328	user_x	10.2.2.10	16003

`Sessionid1` se trouve dans l'expression TIMEWINDOW, mais `sessionid` 2 est arrêté car le TIMEWINDOW de 3600 est dépassé.

Logique conditionnelle dans les requêtes AQL

Vous pouvez soumettre une logique conditionnelle dans les requêtes AQL en utilisant des expressions IF et CASE.

Vous pouvez utiliser dans vos requêtes AQL une logique conditionnelle afin de fournir des options alternatives selon que l'évaluation de la condition de clause renvoie true ou false.

Instructions CASE

Les expressions CASE renvoie la valeur booléenne true ou false. Lorsqu'une expression renvoie la valeur true, la valeur de l'expression CASE est renvoyée et le traitement est arrêté. Si elle renvoie false, la valeur de la clause ELSE est renvoyée.

Dans l'exemple suivant, lorsque le nom de l'utilisateur est root, la valeur de l'expression CASE retournée est `Admin root`. Lorsque le nom de l'utilisateur est admin, la valeur de l'expression CASE retournée est `Admin user`. Si les expressions CASE renvoient la valeur booléenne false, la valeur de la clause ELSE est renvoyée.

```
SELECT CASE username
WHEN 'root'
THEN 'Admin root'
WHEN 'admin'
THEN 'Admin user'
ELSE 'other' END FROM events
```

Lorsque l'instruction WHEN renvoie la valeur true, l'instruction THEN est alors traitée. Sinon, le traitement s'arrête.

Instructions IF, THEN, ELSE

Les instructions entre THEN et ELSE sont traitées lorsque l'instruction IF renvoie la valeur true.

Dans cet exemple, lorsque la condition IF est vraie, `'ADMIN'` est renvoyé lorsque le nom d'utilisateur est `'root'`, sinon le nom d'utilisateur est renvoyé par le journal des événements.

```
SELECT sourceip,
IF username = 'root'
THEN 'ADMIN'
ELSE username AS user FROM events
```

Dans l'exemple suivant, si le journal n'a pas de nom d'utilisateur, celui-ci provient du modèle d'actif. Autrement, le nom d'utilisateur est renvoyée à partir du journal d'événements.

```
SELECT sourceip,
IF username IS NULL
THEN ASSETUSER(sourceip)
ELSE username AS username FROM events
GROUP BY username
LAST 2 DAYS
```

Opérateurs Bitwise dans les requêtes AQL

Améliorer la capacité de filtrage et les performances de vos requêtes AQL qui incluent des adresses IP en utilisant des opérateurs au niveau du bit. Spécifiez des filtres au niveau de l'octet d'adresse IP pour renvoyer des résultats spécifiques.

En filtrant les octets dans une adresse IP, vous pouvez affiner les critères de recherche d'adresse IP.

Par exemple, pour rechercher des types d'unité spécifiques dont le dernier octet d'une adresse IP source se termine par 100, par exemple x.y.z.100, vous pouvez utiliser la requête suivante:

```
SELECT LONG(sourceip)AS long_ip,
sourceip
FROM events into <cursor_name>
WHERE (long_ip & 0x000000ff)=0x00000064
GROUP BY long_ip
ORDER BY long_ip
```

Dans l'exemple, le <*sourceip*> est renvoyé sous la forme d'un entier. L'entier est utilisé par l'opérateur bitwise AND. La valeur hexadécimale <*ff*> dans la position du dernier octet de l'adresse IP source spécifie un filtre dans la position d'octet d'adresse IP correspondante de 0x000000 <*Valeur hexadécimale de l'octet d'adresse IP*>>. Dans ce cas, la valeur hexadécimale <*64*> est remplacée par la valeur décimale 100 dans l'adresse IP.

Le résultat est toutes les adresses IP source qui se terminent par 100. Les résultats peuvent être une liste pour un type d'unité spécifique pour une société, si le dernier octet de toutes les adresses IP est 100.

Les exemples suivants présentent les scénarios à utiliser lorsque vous effectuez une recherche avec des opérateurs bitwise.

Exemples AND (&) au niveau du bit

Renvoie toutes les adresses IP correspondant à 10.xxx.xxx.xxx

```
SELECT LONG(sourceip)AS long_ip,
sourceip
FROM events into t1
WHERE (long_ip & 0xff000000)=0x0a000000
GROUP BY long_ip
LIMIT 50
```

Renvoie toutes les adresses IP correspondant à xxx.100.xxx.xxx

```
SELECT LONG(sourceip)AS long_ip,
sourceip
FROM events into t2
WHERE (long_ip & 0x00ff0000)=0x0064000
GROUP BY long_ip
ORDER BY long_ip
```

Renvoie toutes les adresses IP correspondant à xxx.xxx.220.xxx

```
SELECT LONG(sourceip)AS long_ip,
sourceip
FROM events into t3
WHERE (long_ip & 0x0000ff00)=0x000dc00
GROUP BY long_ip
ORDER BY long_ip
```

Renvoie toutes les adresses IP correspondant à xxx.xxx.xxx.1

```
SELECT LONG(sourceip)AS long_ip,
sourceip
FROM events
WHERE (long_ip & 0x000000ff)=0x0000001
GROUP BY long_ip
ORDER BY long_ip
```

Exemples NOT (~) au niveau du bit

Utilisez les exemples suivants pour convertir chaque valeur 1 bit en valeur 0 bit, ou chaque valeur 0 bit à une valeur 1 bit, dans un modèle binaire donné.

```
SELECT ~123456789
FROM events
LIMIT 1
```

Renvoie 123456790

```
SELECT ~0
FROM events
LIMIT 1
```

Renvoie -1

```
SELECT ~2147483647
FROM events
LIMIT 1
```

Renvoie - 2147483648

Exemples OR au niveau du bit

Utilisez les exemples suivants pour comparer deux bits. Si les deux bits ont une valeur « 1 », la requête renvoie un 1. Si les deux bits ont une valeur « 0 », la requête renvoie un 0.

```
SELECT destinationip,
LONG(destinationip),
sourceip,
LONG(sourceip)AS source_ip,
LONG(destinationip)|source_ip
FROM events
WHERE destinationip='127.0.0.1'
LIMIT 1
```

```
SELECT destinationip,
LONG(destinationip),
sourceip,
~LONG(sourceip)AS not_source_ip,
LONG(destinationip)|not_source_ip
FROM events
WHERE destinationip='127.0.0.1'
LIMIT 1
```

```
SELECT-2147483648|2147483647
FROM events
LIMIT 1
```

Renvoie -1

Exemples XOR au niveau du bit

Les exemples suivants peuvent être utilisés pour prendre des modèles 2 bits, ou une paire de bits de chaque position, et les convertir en 1 ou 0. Si les bits sont différents, le résultat dans cette position est 1. Si les bits sont identiques, le résultat dans cette position est 0.

```
SELECT 2147483647#2147483647
FROM events
LIMIT 1
```

Renvoie 0

```
SELECT 12345#6789
AS A,
(~12345 & 6789)|(12345 & ~6789)
AS B
FROM events
LIMIT 1
```

Renvoie 10940, 10940

Exemples ShiftLeft

Le nombre d'emplacements à déplacer est donné comme second argument à l'opérateur de décalage.

```
SELECT -1<<1
AS A
FROMS events
LIMIT 1
```

Renvoie -2

```
SELECT 16<<1
AS A
FROMS events
LIMIT 1
```

Renvoie 128

Exemples ShiftRight

L'opérateur >> utilise le bit de signe, qui est le plus gauche, pour remplir les positions de fin après le décalage. Si le nombre est négatif, alors 1 est utilisé comme filtre et si le nombre est positif, alors 0 est utilisé comme filtre.

```
SELECT 16>>3
AS A
FROMS events
LIMIT 1
```

Renvoie 2

```
SELECT -32768>>15
AS A
FROMS events
LIMIT 1
```

Renvoie -1

Exemple de ShiftRightUnsigned

Toujours remplir 0 quel que soit le signe du nombre.

```
SELECT -1>>>33
FROM events
LIMIT 1
```

Renvoie 2147483647

Dividing par la puissance de 2.

```
SELECT (20+44)>>>1 A,
(20+44)>>>2 B,
(20+44)>>>3 C,
(20+44)>>>4 D,
(20+44)>>>5 E
FROM events
LIMIT 1
```

Adresses IP CIDR dans les requêtes AQL

Vous pouvez insérer des adresses IP CIDR (IPv4 ou IPv6) dans vos instructions AQL pour demander une plage d'adresses IP, une adresse IP source, une adresse IP de destination ou vous pouvez exclure des adresses IP CIDR spécifiques.

Exemples d'adresses IP CIDR dans les requêtes AQL

Requête par adresse IP source CIDR ou par adresse IP CIDR de destination.

```
SELECT * FROM flows
WHERE INCIDR('10.100.100.0/24',sourceip)
```

```
SELECT * FROM flows
WHERE INCIDR('10.100.100.0/24',destinationip)
```

```
SELECT * FROM flows
WHERE INCIDR('ff02:0:0:0:0:1:ff2f:29d6',destinationv6)
```

Requête pour les flux ayant une adresse IP source ou cible CIDR 10.100.100.0/24

```
SELECT * FROM flows
WHERE INCIDR('10.100.100.0/24',sourceip)
OR INCIDR('10.100.100.0/24',destinationip)
```

Requête pour les événements où 192.168.222.0/24 n'est pas l'adresse IP source CIDR.

```
SELECT *
FROM events
WHERE NOT INCIDR('192.168.222.0/24',sourceip)
```

Requête pour les flux où 192.168.222.0/24 n'est pas l'adresse IP CIDR de destination.

```
SELECT *
FROM flows
WHERE NOT INCIDR('192.168.222.0/24',destinationip)
```

Propriétés personnalisées dans les requêtes AQL

Vous pouvez appeler directement une propriété personnalisée dans vos instructions AQL. Si la propriété personnalisée contient des espaces, vous devez utiliser des guillemets pour encapsuler la propriété personnalisée.

Vous devez activer une propriété personnalisée avant de pouvoir l'utiliser dans une instruction AQL.

Si la propriété personnalisée n'est pas activée, vous pourrez exécuter votre requête AQL, mais vous n'obtiendrez pas de résultats.

Exemple de propriété personnalisée

```
SELECT Bluecoat-cs-host, sourceip, Bluecoat-cs-uri
FROM events
WHERE LOGSOURCEGROUPNAME(devicegrouplist)
ILIKE '%Proxies%'
AND Bluecoat-cs-host ILIKE '%facebook.com%'
GROUP BY sourceip
```

`Bluecoat-cs-host` est le nom d'hôte à partir de l'URL du client qui est demandée.

`Bluecoat-cs-uri` est l'URL d'origine demandée.

Exemples de requêtes de performances système

Vous pouvez utiliser ou éditer des exemples de requêtes AQL de performances système à exécuter dans votre réseau.

Utilisez les exemples de requête suivants pour obtenir des informations sur les performances du système dans votre réseau ou modifier ces exemples pour créer vos propres requêtes personnalisées.

Utilisation du disque et utilisation de l'unité centrale

```
SELECT Hostname, "Metric ID", AVG(Value)
AS Avg_Value, Element
FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%%health%%'
AND
"Metric ID"='SystemCPU'
OR
"Metric ID"='DiskUtilizationDevice'
GROUP BY Hostname, "Metric ID", Element
ORDER BY Hostname last 20 minutes
```

Cette requête génère les colonnes **Hostname**, **MetricID**, **Avg_Value** et **Element**.

La colonne **Avg_Value** renvoie une valeur moyenne pour l'utilisation de l'UC et l'utilisation du disque.

Utilisation du disque par partition

```
SELECT Hostname, AVG(Value) AS Disk_Usage, Element
FROM events
where LOGSOURCENAME(logsourceid)
ILIKE '%%health%%'
and "Metric ID"='DiskUsage'
GROUP BY Hostname, Element
ORDER BY Hostname
LAST 2 HOURS
```

Cette requête génère les colonnes **Hostname**, **Disk_Usage** et **Element**.

La colonne **Disk_Usage** renvoie une valeur d'utilisation du disque pour les répertoires répertoriés dans la colonne **Element**.

Utilisation du disque en gigaoctets (Go) par partition

```
SELECT element
AS Partiton_Name,
MAX(value/(1024*1024*1024))
AS 'Gigabytes_Used'
FROM events
WHERE "Metric ID"='DiskSpaceUsed'
GROUP BY element
ORDER BY Gigabytes_Used DESC
LAST 2 DAYS
```

Cette requête génère les colonnes **Partition_Name** et **Gigabytes_Used** à partir de la base de données des événements.

La colonne **Gigabytes_Used** renvoie une valeur pour les gigaoctets utilisés par chaque partition répertoriée dans la colonne **Gigabytes_Used** des deux derniers jours.

Copie d'exemples de requête à partir du guide AQL

Si vous copiez et collez un exemple de requête contenant des guillemets simples ou doubles à partir du guide AQL, vous devez retaper les guillemets pour être sûr que la requête analyse la syntaxe.

Exemples de requêtes d'événements et de flux

Utilisez ou éditez des exemples de requête pour créer des événements et des requêtes de flux que vous pouvez utiliser pour vos recherches AQL.

Utilisez les exemples de requête suivants pour obtenir des informations sur les événements et les flux de votre réseau ou modifier ces exemples pour générer vos propres requêtes personnalisées.

Taux d'événements et débits de flux pour des hôtes spécifiques

```
SELECT AVG(Value), "Metric ID", Hostname
FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%%health%%'
AND ("Metric ID"='FlowRate' OR "Metric ID"='EventRate')
GROUP BY "Metric ID", Hostname
LAST 15 minutes
```

Cette requête génère les colonnes **AVG_Value**, **ID Metric** et **Hostname** à partir de la base de données d'événements ou de flux depuis les 15 dernières minutes.

La colonne **AVG_Value** renvoie une valeur pour le débit moyen ou le débit d'événements au cours des 15 dernières minutes pour l'hôte nommé dans la colonne **Hostname**.

Débit EPS par source de journal

```
SELECT logsourcename(logsourceid)
AS 'MY Log Sources',
SUM(eventcount) / 2.0*60*60
AS EPS_Rates
FROM events
GROUP BY logsourceid
ORDER BY EPS_Rates DESC
LAST 2 HOURS
```

Cette requête génère les colonnes **My Log Sources** et **EPS_Rates** à partir des événements.

La colonne **My Log Sources** renvoie les noms des sources de journal et la colonne **EPS-Rates** renvoie les débits EPS pour chaque source de journal au cours des deux dernières heures.

Nombre d'événements et types d'événement par jour

```
SELECT
DATEFORMAT( devicetime, 'dd-MM-yyyy')
AS 'Date of log source',
QIDDESCRIPTION(qid)
AS 'Description of event', COUNT(*)
FROM events
WHERE devicetime >( now() -(7*24*3600*1000) )
GROUP BY "Date of log source", qid
LAST 4 DAYS
```

Cette requête génère les colonnes d'événement **Date of log source**, **Description of event** et **count** à partir des événements.

La date de l'événement, la description de l'événement et le nombre d'événements sont renvoyés pour les quatre derniers jours.

Surveillance du trafic local vers le flux à distance par réseau

```
SELECT sourceip,
LONG(SUM(sourcebytes+destinationbytes))
AS TotalBytes
FROM flows
WHERE flowdirection= 'L2R'
AND NETWORKNAME(sourceip)
ILIKE 'servers'
GROUP BY sourceip
ORDER BY TotalBytes
```

Cette requête génère les colonnes **sourceip** et **TotalBytes**.

La colonne **TotalBytes** renvoie la somme des octets source et de destination qui croise de local à distant.

Surveillance du trafic local vers le trafic local par réseau

```
SELECT sourceip,
LONG(SUM(sourcebytes+destinationbytes))
AS TotalBytes
FROM flows
WHERE flowdirection= 'R2L'
AND NETWORKNAME(sourceip)
ILIKE 'servers'
GROUP BY sourceip
ORDER BY TotalBytes
```

Cette requête génère les colonnes **sourceip** et **TotalBytes**.

La colonne **TotalBytes** renvoie la somme des octets source et de destination de l'emplacement distant au local.

Copie d'exemples de requête à partir du guide AQL

Si vous copiez et collez un exemple de requête contenant des guillemets simples ou doubles à partir du guide AQL, vous devez retaper les guillemets pour être sûr que la requête analyse la syntaxe.

Exemples de requêtes de données de référence

Utilisez des requêtes AQL pour obtenir des données à partir d'ensembles de références, de mappes de référence ou de tables de référence. Vous pouvez créer et remplir des données de référence en utilisant des règles pour remplir des ensembles de références, en utilisant des flux de menaces externes, par exemple, LDAP Threat Intelligence App, ou en utilisant des fichiers de données importés pour votre ensemble de référence.

Utilisez les exemples suivants pour vous aider à créer des requêtes pour extraire des données de vos données de référence.

Utiliser des tables de référence pour obtenir des métadonnées externes pour les noms d'utilisateur qui s'affichent dans les événements

```
SELECT
REFERENCETABLE('user_data','FullName',username) AS 'Full Name',
REFERENCETABLE('user_data','Location',username) AS 'Location',
REFERENCETABLE('user_data','Manager',username) AS 'Manager',
UNIQUECOUNT(username) AS 'Userid Count',
UNIQUECOUNT(sourceip) AS 'Source IP Count',
COUNT(*) AS 'Event Count'
FROM events
WHERE qidname(qid)ILIKE '%logon%'
```

```
GROUP BY "Full Name", "Location", "Manager"
LAST 1 days
```

Utilisez la table de référence pour obtenir des données externes telles que le nom complet, l'emplacement et le nom du gestionnaire pour les utilisateurs qui se sont connectés au réseau au cours des dernières 24 heures.

Obtenir les ID utilisateur globaux pour les utilisateurs dans les événements marqués pour une activité suspecte

```
SELECT
REFERENCEMAP('GlobalID_Mapping',username) AS 'Global ID',
REFERENCETABLE('user_data','FullName', 'Global ID') AS 'Full Name',
UNIQUECOUNT(username),
COUNT(*) AS 'Event count'
FROM events
WHERE RULENAME(creEventlist)
ILIKE '%suspicious%'
GROUP BY "Global ID"
LAST 2 days
```

Dans cet exemple, les utilisateurs individuels ont plusieurs comptes dans le réseau. L'organisation requiert une vue unique de l'activité d'un utilisateur. Utilisez les données de référence pour mapper les ID utilisateurs locaux à un ID global. La requête renvoie les comptes utilisateur qui sont utilisés par un ID global pour les événements marqués comme suspects.

Utilisez une recherche de mappe de référence pour extraire des noms d'utilisateur globaux pour les noms d'utilisateur renvoyés dans des événements

```
SELECT
QIDNAME(qid) as 'Event name',
starttime AS Time,
sourceip AS 'Source IP',
destinationip AS 'Destination IP',
username AS 'Event Username',
REFERENCEMAP('GlobalID_Mapping', username) AS 'Global User'
FROM events
WHERE "Global User" = 'John Ariel'
LAST 1 days
```

Utilisez la mappe de référence pour rechercher les noms d'utilisateur globaux pour les noms d'utilisateur renvoyés dans les événements. Utilisez la clause WHERE pour renvoyer uniquement les événements de l'utilisateur global John Ariel. John Ariel peut avoir quelques noms d'utilisateur différents mais ces noms d'utilisateur sont mappés à un utilisateur global, par exemple, dans un système de mappage d'identité externe, vous pouvez mapper un utilisateur global à plusieurs noms d'utilisateur utilisés par le même utilisateur global.

Surveillance de l'utilisation élevée du réseau par les utilisateurs

```
SELECT
LONG(REFERENCETABLE('PeerGroupStats', 'average',
REFERENCEMAP('PeerGroup',username)))
AS PGave,
LONG(REFERENCETABLE('PeerGroupStats', 'stdev',
REFERENCEMAP('PeerGroup',username)))
AS PGstd,
SUM(sourcebytes+destinationbytes) AS UserTotal
FROM flows
WHERE flowtype = 'L2R'
GROUP BY UserTotal
HAVING UserTotal > (PGAve+ 3*PGStd)
```

Renvoie les noms d'utilisateur lorsque l'utilisation du flux est trois fois supérieure à l'utilisateur moyen.

Vous avez besoin d'un ensemble de référence pour stocker l'utilisation réseau des homologues par nom d'utilisateur et nombre total d'octets.

Cotes et catégories de menaces

```
SELECT
REFERENCETABLE('ip_threat_data','Category',destinationip)
AS 'Threat Category',
REFERENCETABLE('ip_threat_data','Rating', destinationip)
AS 'Threat Rating',
UNIQUECOUNT(sourceip)
AS 'Source IP Count',
UNIQUECOUNT(destinationip)
AS 'Destination IP Count'
FROM events
GROUP BY "Threat Category", "Threat Rating" LAST 24 HOURS
```

Renvoie la catégorie de menace et la cote de menace.

Vous pouvez rechercher des données de menace de table de référence et les inclure dans vos recherches.

Copie d'exemples de requête à partir du guide AQL

Si vous copiez et collez un exemple de requête contenant des guillemets simples ou doubles à partir du guide AQL, vous devez retaper les guillemets pour être sûr que la requête analyse la syntaxe.

Exemples de requêtes de surveillance d'utilisateur et de réseau

Utilisez des exemples de requête pour vous aider à créer vos requêtes AQL d'utilisateur et de surveillance de réseau.

Utilisez les exemples suivants pour surveiller vos utilisateurs et votre réseau, ou vous pouvez modifier les requêtes en fonction de vos besoins.

Rechercher les utilisateurs ayant utilisé le réseau privé virtuel pour accéder au réseau à partir de trois adresses IP ou plus dans une période de 24 heures

```
SELECT username,
UNIQUECOUNT(sourceip)
AS 'Source IP count'
FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%VPN%'
AND username IS NOT NULL
GROUP BY username
HAVING "Source IP count" >= 3
ORDER BY "Source IP count"
DESC
LAST 24 HOURS
```

Cette requête génère les colonnes **username** et **Source IP count**.

La colonne **username** renvoie les noms des utilisateurs qui ont utilisé le réseau privé virtuel pour accéder au réseau à partir de trois adresses IP ou plus au cours des dernières 24 heures.

Rechercher les utilisateurs ayant utilisé le réseau privé virtuel depuis plus d'un emplacement géographique en 24 heures

```
SELECT username, UNIQUECOUNT(geographiclocation)
AS 'Count of locations'
FROM events
WHERE LOGSOURCENAME(logsourceid)
ILIKE '%VPN%'
AND geographiclocation <> 'other location'
AND username
```

```
IS NOT NULL
GROUP BY username
HAVING "Count of locations" > 1
ORDER BY "Count of locations"
DESC
LAST 3 DAYS
```

Cette requête génère les colonnes **username** et **Count of locations**.

La colonne **username** renvoie les noms des utilisateurs qui ont utilisé le réseau privé virtuel de plus d'un emplacement qui n'est pas appelé « autre emplacement » au cours des dernières 24 heures.

Surveillance du trafic local vers le flux à distance par réseau

```
SELECT sourceip,
LONG(SUM(sourcebytes+destinationbytes))
AS TotalBytes
FROM flows
WHERE flowdirection= 'L2R'
AND NETWORKNAME(sourceip)
ILIKE 'servers'
GROUP BY sourceip
ORDER BY TotalBytes
```

Cette requête génère les colonnes **sourceip** et **TotalBytes**.

La colonne **TotalBytes** renvoie la somme des octets source et de destination qui croise de local à distant.

Surveillance du trafic local vers le trafic local par réseau

```
SELECT sourceip,
LONG(SUM(sourcebytes+destinationbytes))
AS TotalBytes
FROM flows
WHERE flowdirection= 'R2L'
AND NETWORKNAME(sourceip)
ILIKE 'servers'
GROUP BY sourceip
ORDER BY TotalBytes
```

Cette requête génère les colonnes **sourceip** et **TotalBytes**.

La colonne **TotalBytes** renvoie la somme des octets source et de destination de l'emplacement distant au local.

Utilisation de l'application par nom d'application, utilisateurs et flux de flux

```
SELECT sourceip
AS Source_IP,
FIRST(destinationip)
AS Destination_IP,
APPLICATIONNAME(applicationid)
AS Application,
DATEFORMAT(lastpackettime, 'dd-MM-yyyy hh:m:ss')
AS 'Start Time',
FIRST(sourcebytes)
AS Source_Bytes,
ASSETUSER(sourceip, NOW()) AS Src_Asset_User
FROM flows
GROUP BY Source_IP
ORDER BY Source_Bytes DESC
```

Cette requête génère des données sur les utilisateurs de l'actif, les noms d'application et les données de flux. Utilisez cette requête pour signaler une activité utilisateur spécifique ou l'utilisation d'une application, ou pour générer une variante de cette requête afin d'obtenir les résultats souhaités.

Emplacement des actifs

```
SELECT ASSETPROPERTY('Location',sourceip)
AS asset_location,
COUNT(*)
FROM events
GROUP BY asset_location
LAST 1 days
```

Cette requête génère les colonnes **asset_location** et **count**.

La colonne **asset location** renvoie l'emplacement des actifs.

Copie d'exemples de requête à partir du guide AQL

Si vous copiez et collez un exemple de requête contenant des guillemets simples ou doubles à partir du guide AQL, vous devez retaper les guillemets pour être sûr que la requête analyse la syntaxe.

Zones d'événement, de flux et simarc pour les requêtes AQL

Utilisez Ariel Query Language (AQL) pour extraire des zones spécifiques des événements, des flux et des tables simarc dans la base de données Ariel.

Zones d'événement prises en charge pour les requêtes AQL

Les zones d'événement que vous pouvez interroger sont répertoriées dans le tableau suivant.

Tableau 17. Zones d'événement prises en charge pour les requêtes AQL

Nom de zone	Description
adekey	Ade key
adevalue	Ade value
category	Catégorie de niveau inférieur
creEventList	Règle personnalisée avec correspondance
crédibilité	Crédibilité
destinationMAC	Adresse MAC de destination
destinationPort	Port de destination
destinationv6	Destination IPv6
destinationaddress	Adresse de destination
destinationIP	IP de destination
sourceaddress	Adresse source
deviceTime	Heure de la source de journal
deviceType	Type de source de journal
devicegrouplist	Liste des groupes d'unités
domainID	ID de domaine
duration	Durée
endTime	Heure de stockage
eventCount	Nombre d'événements

Tableau 17. Zones d'événement prises en charge pour les requêtes AQL (suite)	
Nom de zone	**Description**
eventDirection	Direction de l'événement : **local-to-Local (L2L)** **local-to-remote (L2R)** **remote-to-local (R2L)** **remote-to-remote (R2R)**
geographiclocation	emplacement géographique
sourcegeographiclocation	Emplacement géographique source
destinationgeographiclocation	Emplacement géographique de destination
hasIdentity	Possède une identité
hasOffense	Associé à l'infraction
highLevelCategory	Catégorie de niveau supérieur
identityhostname	Nom d'hôte de l'identité
identityip	Adresse IP d'identité
isduplicate	Est en double
isCREEvent	Est un événement de règle personnalisé
logsourceid	ID source de journal
ampleur	Magnitude
pcappacket	Paquet PCAP
partialMatchList	Liste partielle des correspondances
payload	Contenu utile
postNatDestinationIP	Adresse IP de destination après NAT
postNatDestinationPort	Port de destination après NAT
postNatSourceIP	IP source après NAT
postNatSourcePort	Port source après NAT
preNatDestinationIP	Adresse IP de destination avant NAT
preNatDestinationPort	Port de destination avant NAT
preNatSourceIP	Adresse IP source avant NAT
preNatSourcePort	Port source avant NAT
protocolid	Protocole
processorId	ID du processeur d'événements
qid	ID de nom d'événement
qideventid	ID d'événement
pertinence	Pertinence
gravité	Gravité

Tableau 17. Zones d'événement prises en charge pour les requêtes AQL (suite)	
Nom de zone	**Description**
sourceIP	IP source
sourceMAC	Adresse MAC source
sourcePort	Port source
sourcev6	Adresse IPv6 source
startTime	Heure de début
isunparsed	Événement non analysé
userName	Nom d'utilisateur

Zones de flux prises en charge pour les requêtes AQL

Les zones de flux que vous pouvez interroger sont répertoriées dans le tableau suivant.

Tableau 18. Zones de flux prises en charge pour les requêtes AQL	
Nom de zone	**Description**
applicationId	ID d'application
Catégorie	Catégorie
crédibilité	Crédibilité
destinationASN	ASN de destination
destinationBytes	Octets cible
destinationDSCP	DSCP de destination
destinationFlags	Indicateurs de destination
destinationIP	IP de destination
destinationIfIndex	Index conditionnel de destination
destinationPackets	Paquets cible
destinationPayload	Contenu de destination
destinationPort	Port de destination
destinationPrecedence	Priorité de la destination
destinationv6	Destination IPv6
domainID	ID de domaine
fullMatchList	Liste complète des correspondances
firstPacketTime	Heure du premier paquet
flowBias	Biais du flux

Tableau 18. Zones de flux prises en charge pour les requêtes AQL (suite)	
Nom de zone	**Description**
flowDirection	Direction du flux **local-to-local (L2L)** **local-to-remote (L2R)** **remote-to-local (R2L)** **remote-to-remote (R2R)**
flowInterfaceID	ID de l'interface de flux
flowSource	Source de flux
flowType	Type de flux
geographic	Correspond à l'emplacement géographique
hasDestinationPayload	Contient un contenu de destination
hasOffense	A un contenu d'infraction
hasSourcePayload	Contient un contenu source
icmpCode	Code ICMP
icmpType	Type ou code ICMP
flowInterface	Interface du flux
intervalId	ID intervalle
isDuplicate	Événement en double
lastPacketTime	Heure du dernier paquet
partialMatchList	Liste partielle des correspondances
protocolId	ID protocole
qid	Qid
processorID	ID du processeur d'événements
pertinence	Pertinence
retentionBucket	Compartiment de conservation factice
gravité	Gravité
sourceASN	ASN source
sourceBytes	Octets source
sourceDSCP	DSCP source
sourceFlags	Indicateurs source
sourceIP	IP source
sourceIfIndex	Index conditionnel source
sourcePackets	Paquets source
sourcePayload	Contenu source
sourcePort	Port source

Tableau 18. Zones de flux prises en charge pour les requêtes AQL (suite)	
Nom de zone	**Description**
sourcePrecedence	Priorité de la source
sourcev6	Adresse IPv6 source
startTime	Heure de début
viewObjectPair	Afficher la paire d'objets

Zones simarc prises en charge pour les requêtes AQL

Les zones simarc que vous pouvez interroger sont répertoriées dans le tableau suivant.

Tableau 19. Zones simarc prises en charge pour les requêtes AQL	
Nom de zone	**Description**
destinationPort	Créateur de clé de port de destination
destinationType	Créateur de clé de type de destination
deviceId	Créateur de la clé d'unité
direction	Créateur de la clé de direction
eventCount	Créateur de clé de nombre d'événements
eventFlag	Créateur de la clé d'indicateur
applicationId	Créateur de la clé d'ID application
flowCount	Créateur de clé de comptage de flux
destinationBytes	Créateur de clé d'octets de destination
flowSource	Créateur de clé source de flux
sourceBytes	Créateur de clé d'octets source
lastPacketTime	Créateur de clé de temps
protocolId	Créateur de la clé de protocole
source	Créateur de clé source
sourceType	Créateur de clé de type source
sourceRemoteNetwork	Créateur de clé de réseau distant source
destinationRemoteNetwork	Créateur de la clé de réseau distant de destination
sourceCountry	Créateur de la clé géographique source
destinationCountry	Créateur de la clé géographique de destination
destination	Créateur de clé de destination

Remarques

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Cependant, il est de la responsabilité de l'utilisateur d'évaluer et de vérifier le fonctionnement d'un produit, d'un programme ou d'un service nonIBM .

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.</lines>
Pour le Canada, veuillez adresser votre courrier à :
<lines>IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7 Canada

Pour obtenir des informations sur les licences relatives aux produits utilisant des jeux de caractères codés sur deux octets (DBCS), contactez le service de la propriété intellectuelle IBM de votre pays ou envoyez vos demandes de renseignements, par écrit, à :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT. IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE NON-CONTREFACON ET D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Toute référence à ces informations sur des sites Web non-IBM est fournie par souci de commodité uniquement et ne constitue en aucun cas une adhésion au contenu de ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
92066 Paris-La Défense Cedex 50
USA

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du document IBM Customer Agreement, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance et les exemples client ne sont présentés qu'à des fins d'illustration. Les résultats des performances réelles peuvent varier en fonction des configurations et des conditions de fonctionnement spécifiques.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les prix IBM affichés sont des prix de détail suggérés par IBM. Ils sont à jour et peuvent être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp., dans de nombreux pays. Les autres noms de produit et de service peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page web "Copyright and trademark information" à l'adresse www.ibm.com/legal/copytrade.shtml.

Java™ et tous les logos et marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

VMware, le logo VMware, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server et VMware vSphere sont des marques de VMware, Inc. ou de ses filiales aux Etats-Unis et/ou dans certains autres pays.

Dispositions pour la documentation du produit

Les droits d'utilisation relatifs à ces publications sont soumis aux dispositions suivantes.

Domaine d'application

Ces dispositions viennent s'ajouter à toute autre condition d'utilisation applicable au site web IBM.

Usage personnel

Vous pouvez reproduire ces informations pour votre usage personnel, non commercial, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez pas distribuer ou publier tout ou partie de ces publications ou en produire des oeuvres dérivées sans le consentement exprès d'IBM.

Usage commercial

Vous pouvez reproduire, distribuer et afficher ces publications uniquement au sein de votre entreprise, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez pas reproduire, distribuer ou afficher tout ou partie de ces publications en dehors de votre entreprise ou en tirer des oeuvres dérivées, sans le consentement exprès d'IBM.

Droits

Excepté les droits d'utilisation expressément accordés dans le présent document, aucun autre droit, licence ou autorisation, implicite ou explicite, n'est accordé pour ces publications ou autres informations, données, logiciels ou droits de propriété intellectuelle contenus dans ces publications.

IBM se réserve le droit de retirer les autorisations accordées ici si IBM estime, à sa discrétion, que l'utilisation des publications devient préjudiciable à ses intérêts ou qu'à son avis les instructions ci-dessus n'ont pas été respectées.

Vous ne pouvez télécharger, exporter ou réexporter ces informations qu'en total accord avec toutes les lois et règlements applicables dans votre pays, y compris les lois et règlements américains relatifs à l'exportation.

IBM N'OCTROIE AUCUNE GARANTIE SUR LE CONTENU DE CES PUBLICATIONS. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU TACITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES PUBLICATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Déclaration de confidentialité en ligne d'IBM

Les Logiciels IBM, y compris les Logiciels sous forme de services ("Offres Logiciels"), peuvent utiliser des cookies ou d'autres technologies pour collecter des informations sur l'utilisation des produits, améliorer l'acquis utilisateur, personnaliser les interactions avec celui-ci, ou dans d'autres buts. Bien souvent, aucune information personnelle identifiable n'est collectée par les Offres Logiciels. Certaines Offres Logiciels vous permettent cependant de le faire. Si la présente Offre Logiciels utilise des cookies pour collecter des informations personnelles identifiables, des informations spécifiques sur cette utilisation sont fournies ci-dessous.

Selon la configuration déployée, la présente Offre Logiciels peut utiliser des cookies de session et des cookies persistants destinés à collecter le nom et le mot de passe des utilisateurs pour les fonctions de gestion des session et d'authentification. Ces cookies peuvent être désactivés, mais leur désactivation empêchera l'utilisation de la fonctionnalité qui leur est associée.

Si les configurations déployées de cette Offre Logiciels vous permettent, en tant que client, de collecter des informations permettant d'identifier les utilisateurs par l'intermédiaire de cookies ou par d'autres techniques, vous devez solliciter un avis juridique sur la réglementation applicable à ce type de collecte, notamment en termes d'information et de consentement.

Pour plus d'informations sur l'utilisation à ces fins des différentes technologies, y compris celle des cookies, consultez la Déclaration de confidentialité d'IBM à l'adresse http://www.ibm.com/privacy et la section « Cookies, pixels espions et autres technologies » de la Déclaration IBM de confidentialité sur Internet à l'adresse http://www.ibm.com/privacy/details/.

Règlement général sur la protection des données (RGPD)

Il incombe au client de veiller à sa propre conformité aux différentes lois et réglementations, y compris au Règlement général sur la protection des données (RGPD) de l'Union européenne. Il relève de la seule responsabilité du client de consulter les services juridiques compétents aussi bien pour identifier et interpréter les lois et règlements susceptibles d'affecter son activité, que pour toute action à entreprendre pour se mettre en conformité avec ces lois et réglementations. Les produits, services et autres fonctionnalités décrits ici ne sont pas adaptés à toutes les situations des clients et peuvent présenter une disponibilité limitée. IBM ne donne aucun avis juridique, comptable ou d'audit et ne garantit pas que ses produits ou services assurent la conformité de ses clients par rapport aux lois applicables.

En savoir plus sur le niveau de préparation au RGPD IBM et sur nos offres et fonctionnalités RGPD ici : https://ibm.com/gdpr

Index

Caractères spéciaux

A

B

C

D

F

G

H

L

S