

Dell Wyse Management Suite

版本 1.3 高可用性指南



注、小心和警告

① | **注：**“注”表示帮助您更好地使用该产品的重要信息。

△ | **小心：**“小心”表示可能会损坏硬件或导致数据丢失，并说明如何避免此类问题。

⚠ | **警告：**“警告”表示可能会造成财产损失、人身伤害甚至死亡。

1 简介.....	4
高可用性概述.....	4
2 高可用性体系结构.....	5
高可用性的系统要求.....	5
3 Windows Server 2012 上的高可用性.....	7
创建群集角色.....	7
4 在 Windows Server 2012 上实现高可用性.....	10
在 Windows Server 2012 上添加故障转移群集功能.....	10
创建文件共享见证.....	15
配置群集仲裁设置.....	16
创建群集角色.....	18
5 实现 MySQL InnoDB 的高可用性.....	21
通过 MySQL InnoDB 提供高可用性.....	21
安装 MySQL InnoDB 数据库.....	21
检查 MySQL InnoDB 服务器实例.....	21
创建 MySQL InnoDB 的群集实例.....	22
添加服务器实例至 MySQL InnoDB 群集.....	23
配置 MySQL 路由器.....	24
在 MySQL InnoDB 服务器上创建数据库和用户.....	25
6 实现 MongoDB 上的高可用性.....	26
安装 MongoDB.....	26
创建 MongoDB 数据库的副本服务器.....	27
创建数据库用户.....	27
创建 MongoDB 的 DBAdmin 用户.....	28
编辑 mongod.cfg 文件.....	28
服务器上启动复制.....	28
7 实现 Teradici 设备的高可用性.....	32
安装和配置 HAProxy.....	32
安装 Wyse Management Suite 服务器.....	34
8 在 Windows Server 2012 上安装 Wyse Management Suite.....	35
9 安装后检查.....	36
10 Troubleshooting.....	38

简介

Wyse Management Suite 版本 1.3 是下一代管理解决方案，使您能够集中配置、监控、管理和优化您的 Dell Wyse Thin Client。该解决方案有助于部署和管理高可用性设置上性能得到改进的瘦客户端。

它提供了多种高级功能选项，例如云与内部部署、使用移动应用程序随处执行管理，以及 BIOS 配置和端口锁定等增强的安全功能。其他功能包括设备发现和注册、资产和库存清单管理、配置管理、操作系统和应用程序部署、实时命令、监视、警报、报告和端点故障诊断。

Wyse Management Suite 版本 1.3 支持高可用性并且显著减少了系统停机时间。该解决方案还可保护系统，避免出现计划外停机时间，并帮助您实现所需的可用性以实现业务目标。

本指南介绍解决方案架构并说明如何在应用程序和数据库级别设置、配置和维护高可用性群集。

高可用性概述

Wyse Management Suite 版本 1.3 的高可用性解决方案包括以下任务：

- 1 查看高可用性要求 — 请参阅 [System requirements to set up high availability](#)（设置高可用性的系统要求）。
- 2 在 Microsoft Windows Server 2012 上部署高可用性 — 请参阅 [Deploy high availability on Windows Server 2012](#)（在 Windows Server 2012 上部署高可用性）。
- 3 在 MySQL InnoDB 服务器上部署高可用性 — 请参阅 [Deploy high availability on MySQL InnoDB](#)（在 MySQL InnoDB 上部署高可用性）。
- 4 在 MongoDB 上部署高可用性 — 请参阅 [Deploy high availability on MongoDB](#)（在 MongoDB 上部署高可用性）。
- 5 配置高可用性代理（适用于 Teradici 设备） — 请参阅 [Deploy high availability for Teradici servers](#)（部署 Teradici 服务器的高可用性）。
- 6 在 Windows Server 2012 上安装 Wyse Management 版本 — 请参阅 [Install Wyse Management Suite on Windows Server 2012](#)（在 Windows Server 2012 上安装 Wyse Management Suite）。
- 7 查看安装后检查 — 请参阅 [Post installation checks](#)（安装后检查）。
- 8 故障排除问题及解决办法 - 请参阅 [Troubleshooting](#)（故障排除）。

高可用性体系结构

Dell Wyse Management Suite 架构包含启用了故障转移群集的 Windows Server 2012。Windows 群集包含支持其他应用程序的主要计算机并通过利用冗余确保停机时间最短。这用于 Tomcat、Memcache、MQTT 服务的应用程序故障转移。MongoDB 数据库群集可帮助在主要数据库发生故障时，次要数据库接管工作。MySQL InnoDB 数据库群集具有内置数据库群集机制，如果主要读写数据库发生故障，次要数据库将接管工作。具有 HA 代理的 Linux 服务器是一个负载均衡器，也是适用于 EMSDK (Teradici) 服务器的高可用性服务器。本地存储库作为共享路径的一部分创建，它包含应用程序、映像、软件包，并且它不会是群集设置的一部分。

① | 注: 高可用性系统要求根据您的工作地点的基础架构可能会有所改变。

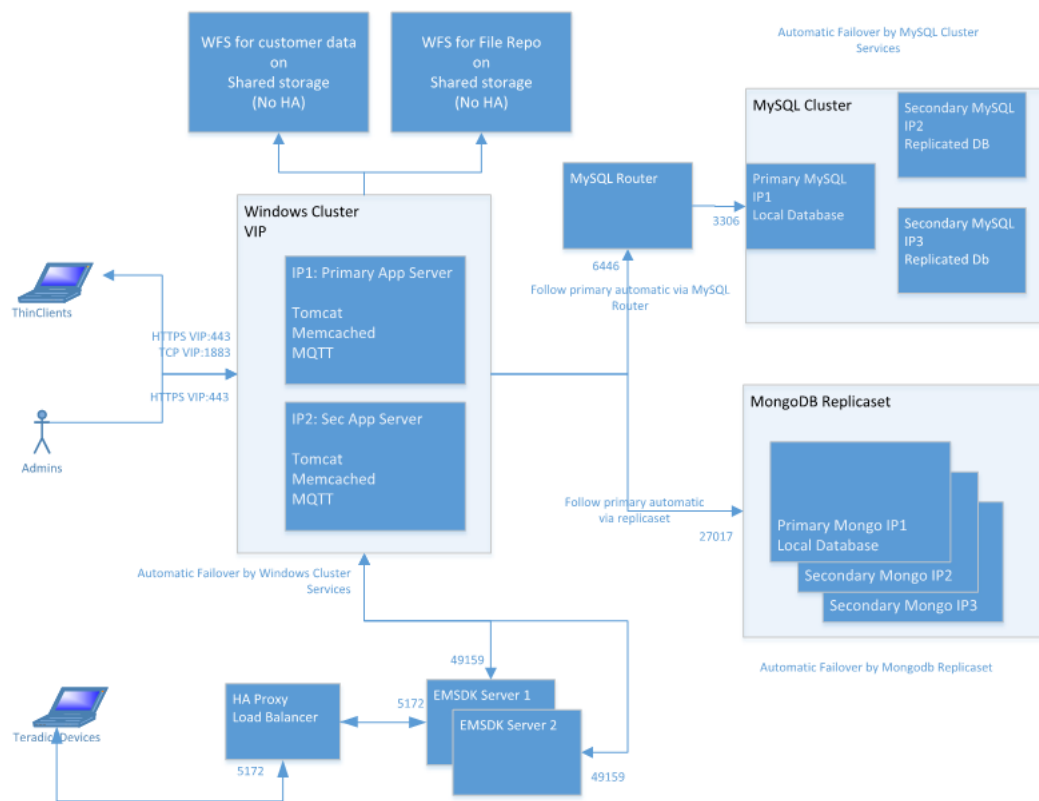


图 1: 高可用性体系结构

高可用性的系统要求

该表列出了最低硬件和软件要求并最多支持 10,000 个设备。每个 EMSDK 实例最多可支持 5000 个设备。部署可以在单个服务器上或在虚拟机监控程序环境中进行，视需要而定。

为 Wyse Management Suite 版本 1.3 设置高可用性的硬件和软件要求如下：

表. 1: 系统要求

产品	端口	协议	说明
Microsoft Windows Server 2012 R2	- 网络通信端口： - UDP：3343 - TCP：3342 - UDP：137	- 最小磁盘空间 — 40 GB - 最小系统数量 — 2 - 最小内存(RAM) — 8 GB - 最低 CPU 要求 — 4	Wyse Management Suite 托管的服务器。 支持英语、法语、意大利语、德语和西班牙语。
MySQL 群集	网络通信端口 — TCP：3306	- 最小磁盘空间 — 40 GB - 最小系统数量 — 3 - 最小内存(RAM) — 8 GB - 最低 CPU 要求 — 4	高可用性设置中的服务器。
MySQL 路由器	- 网络通信端口： 6446 6447	- 最小磁盘空间 — 40 GB - 最小系统数量 — 2 - 最小内存(RAM) — 8 GB - 最低 CPU 要求 — 4	在高可用性设置中建立通信。
MongoDB	网络通信端口 — TCP：27017	- 最小磁盘空间 — 40 GB - 最小系统数量 — 3 - 最小内存(RAM) — 8 GB - 最低 CPU 要求 — 4	数据库
EMSDK	- 网络通信端口 — TCP：5172 - TCP 49159	- 最小磁盘空间 — 40 GB - 最小系统数量 — 2 - 最小内存(RAM) — 8 GB - 最低 CPU 要求 — 4	Enterprise SDK 服务器
HAProxy	网络通信端口 — TCP：5172	- 最小磁盘空间 — 40 GB - 最小系统数量 — 1 - 最小内存(RAM) — 4 GB - 最低 CPU 要求 — 2	高可用性设置中的负载均衡器。 Ubuntu 版本 12.04 及更高版本。

① 注:

在高可用性设置期间，确保将 TCP 端口 443、8080 和 1883 添加到防火墙例外列表中。

Windows Server 2012 上的高可用性

故障转移群集是一组独立的系统，可提高群集角色的可用性和可扩展性。此功能支持硬件上或虚拟机上运行群集的多个工作负载。

故障转移群集是一组独立的系统机，可提高群集角色的可用性和可扩展性。群集服务器是互相连接形成网络的节点。如果一个或多个群集节点发生故障，其他节点将变为活动状态，并可防止网络中的系统故障转移。在群集设置过程中创建的群集角色监测以验证群集网络中的系统正常工作。如果任何系统未正常工作，则会将它们重新启动或移至另一个节点。

在 Windows Server 2012 上，用于获得高可用性的故障转移群集网络包含在运行 Windows Server 2012 的系统上配置的两个节点：节点 1 和节点 2。在故障转移群集网络中，如果作为主要主节点工作的节点 1 发生故障，节点 2 会自动作为主要节点工作。节点 1 变活动后，它将自动成为次要节点。系统具有已连接至网络的共享存储空间。

① | 注：图像中系统的 IP 地址只是示例，对于您工作场所中的每个系统各不相同。

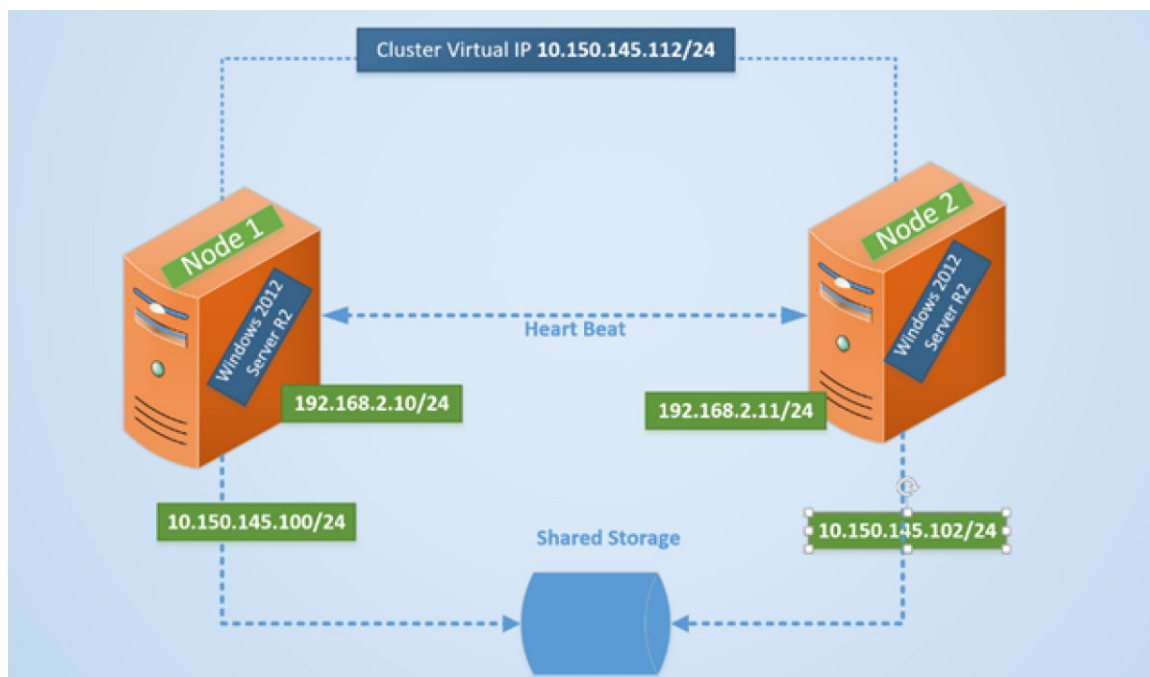


图 2: 故障转移群集设置

创建群集角色

在创建故障转移群集之后，您可以创建群集角色以管理群集工作负载。在创建群集角色之前，确保 Wyse Management Suite 已安装在服务器上并指向远程数据库。

要创建群集角色，请执行以下操作：

- 1 在 Microsoft Windows Server 2012 中，右键单击**开始菜单**，然后选择**服务器管理器**以启动“服务器管理器”仪表板
- 2 单击**故障转移群集管理器**以启动群集管理器。
- 3 右键单击**角色**，然后选择**配置角色**以显示高可用性向导屏幕。

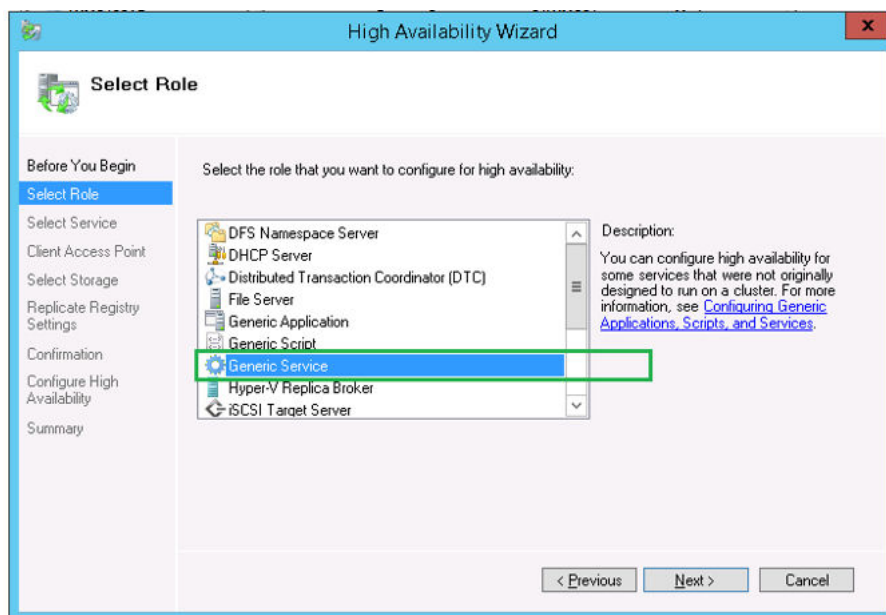


图 3: 高可用性向导

- 4 选择通用服务，然后单击下一步以查看选择服务屏幕。

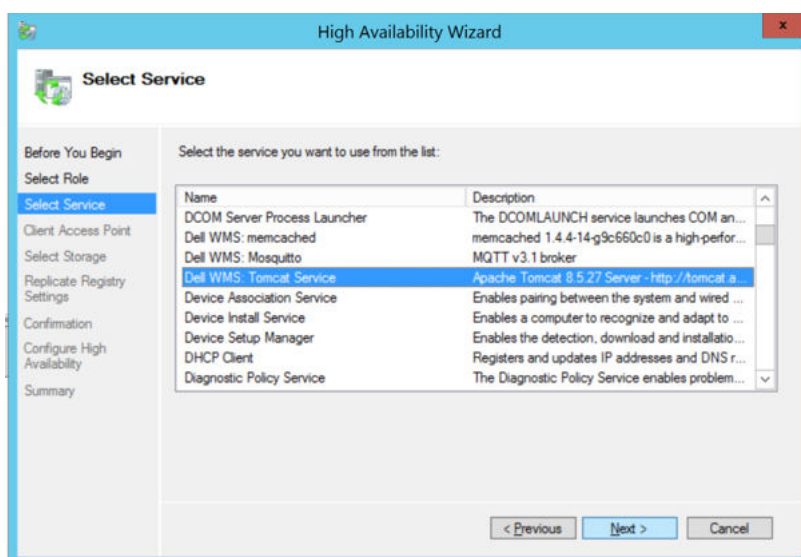


图 4: 选择服务

- 5 选择 **Dell WMS: Tomcat 服务**，然后单击下一步。

注: 只有在安装 Wyse Management Suite 版本 1.3 之后，才能将与 Wyse Management Suite 版本 1.3 相关的服务添加到群集。

此时将显示高可用性向导屏幕，您需要在其中创建客户端访问点以及在 Windows Server 2012 和 Wyse Management Suite 之间建立连接。

- 6 在名称字段中键入网络名称，然后单击下一步。此时将显示确认屏幕，其中含有服务器的网络名称和 IP 地址详细信息。

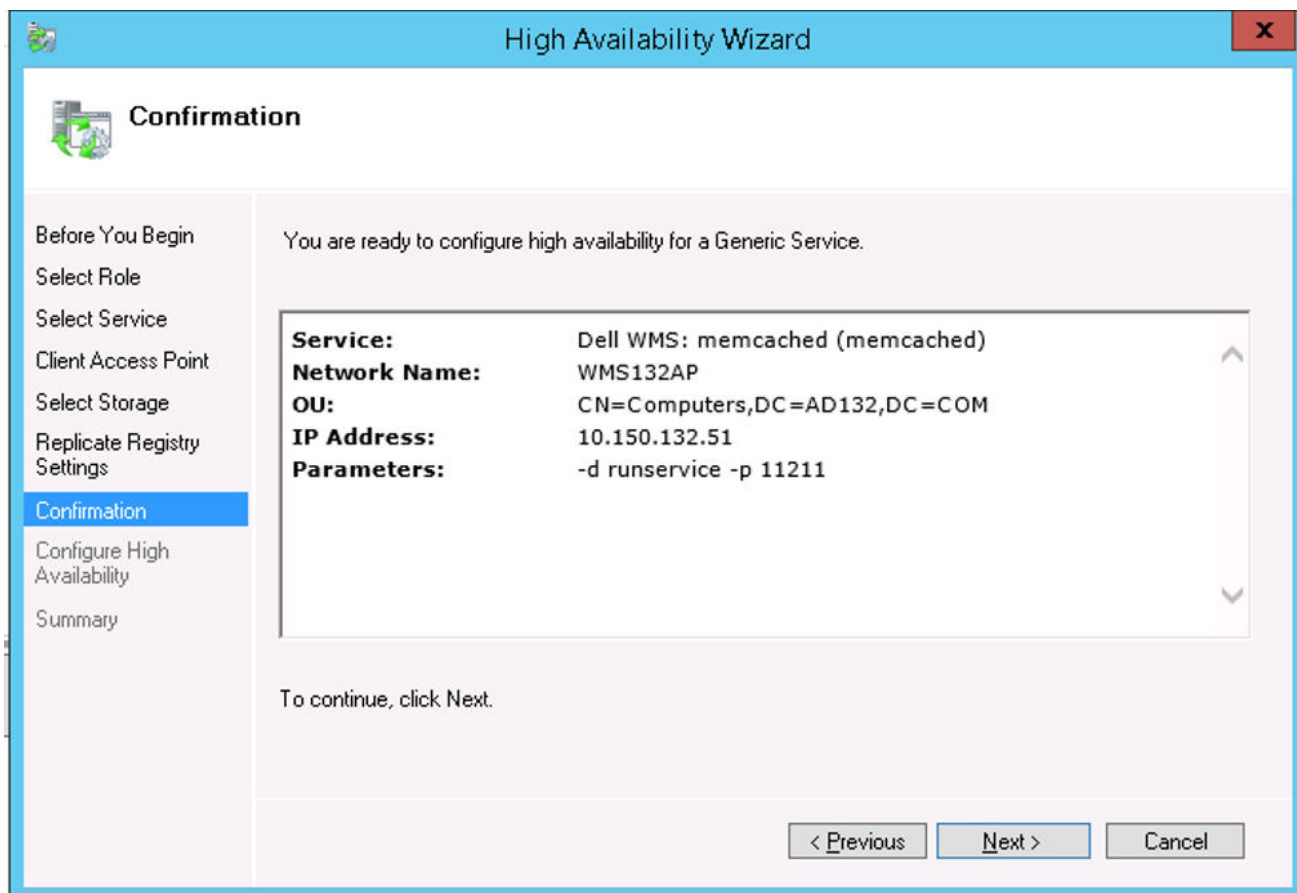


图 5: 确认

- 7 单击下一步以完成该过程。
- 8 要添加其他 Wyse Management Suite 服务作为群集的一部分，请启动故障转移群集管理器，然后转至操作角色以显示您已创建的网络名称。
- 9 单击网络名称，然后转至添加资源通用服务。
- 10 从新建资源向导屏幕中，选择以下服务（需要添加它们作为群集的一部分）：
 - a Dell WMS: Mosquitto >> MQTT Broker
 - b Dell WMS: memcached
- 11 单击下一步以完成任务。
已添加为群集一部分的 Wyse Management Suite 服务以正在运行状态显示。

在 Windows Server 2012 上实现高可用性

以下是在 Windows Server 2012 上实现高可用性的步骤：

- 1 在 Windows Server 2012 上添加故障转移群集功能 - 请参阅 [Adding failover cluster feature on Windows Server 2012](#) (在 Windows Server 2012 上添加故障转移群集功能)。
- 2 创建文件共享见证 - 请参阅 [Create file share witness](#) (创建文件共享见证)。
- 3 配置群集仲裁 - 请参阅 [Configure cluster Quorum](#) (配置群集仲裁)。
- 4 创建群集角色 - 请参阅 [Create cluster roles](#) (创建群集角色)。

在 Windows Server 2012 上添加故障转移群集功能

要在 Windows Server 2012 上添加故障转移群集功能，请执行以下操作：

- 1 在 Microsoft Windows Server 2012 中，单击开始以打开开始屏幕，然后单击服务器管理器以启动服务器管理器仪表板。
① 注：服务器管理器是 Windows Server 2012 中的管理控制台，可让您添加服务器角色/功能、管理和部署服务器。
- 2 单击添加角色和功能，然后从添加角色和功能向导屏幕中选择一个选项，根据您的要求配置服务器。

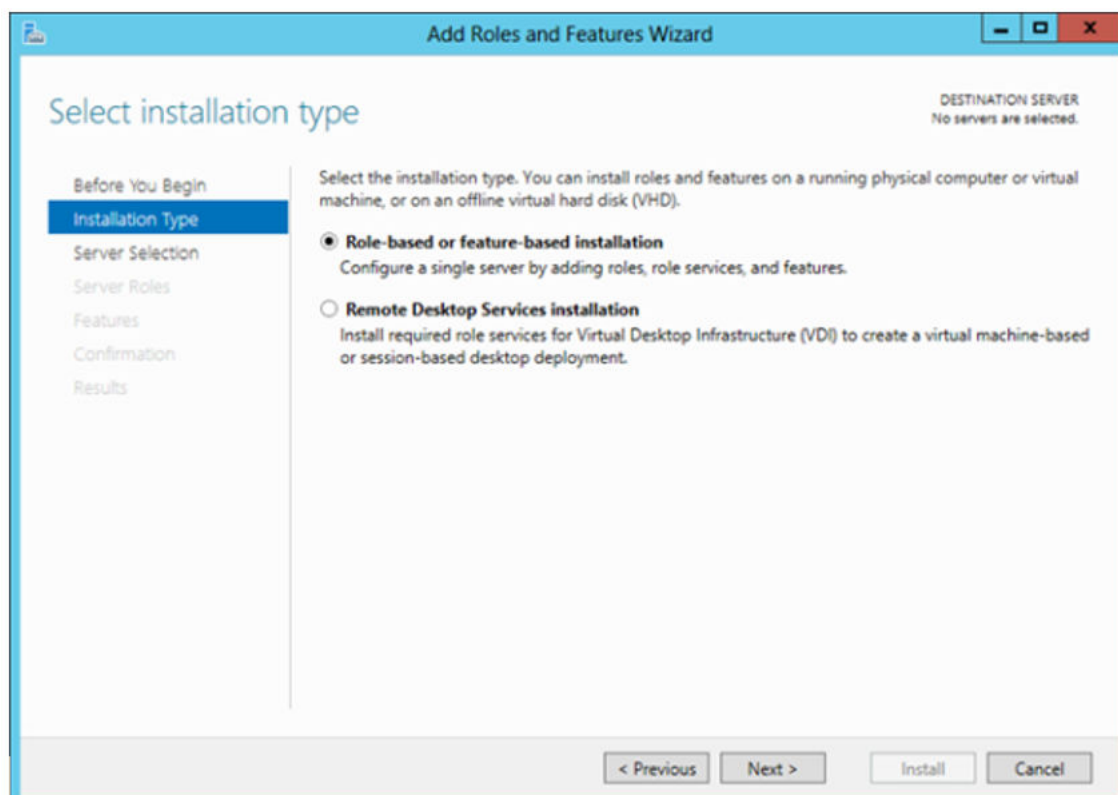


图 6: 基于角色的选项

- 3 单击安装类型并选择基于角色或基于功能的安装，然后单击下一步以在选择目标服务器屏幕中查看服务器列表。

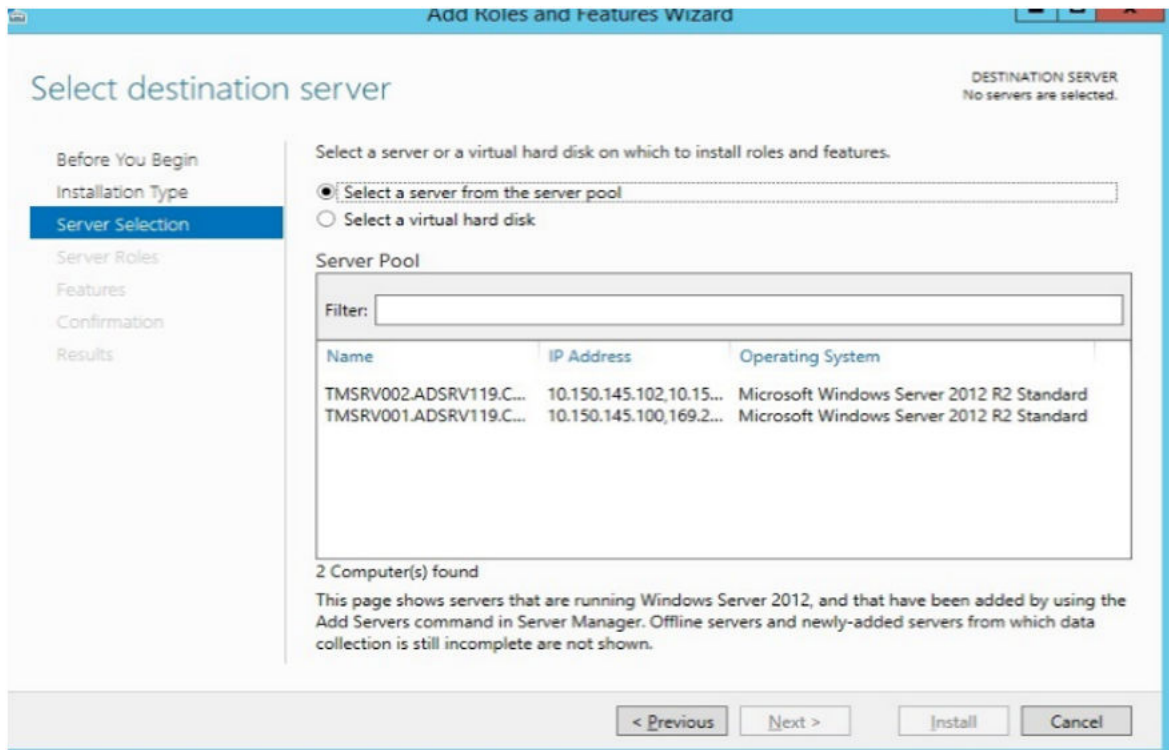


图 7: 选择服务器目标

- 4 选择要在其中启用故障转移群集功能的服务器，然后单击下一步。
- 5 在功能屏幕上选择**故障转移群集**，然后单击下一步。在服务器上启用故障转移群集之后，在节点 1 中的服务器上打开**故障转移群集管理器**。
- 6 单击**是**以确认安装，并在所选服务器上启用故障转移群集功能。
- 7 在**故障转移群集管理器**屏幕中，单击**验证配置**以查看**验证配置向导**，以将所需的服务器或节点添加到群集。

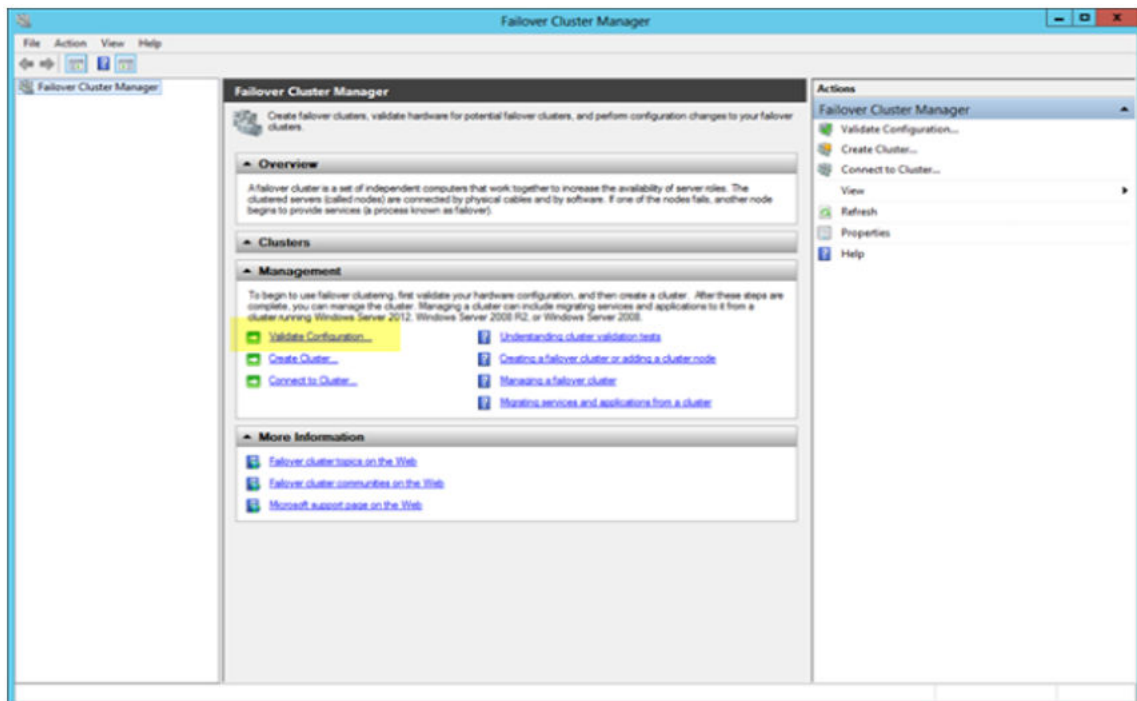


图 8: 故障转移群集管理器

- 8 单击**选择 服务器或群集**，然后单击**浏览**以配置服务器。
- 9 单击**下一步**，然后从**测试选项**屏幕中选择**运行所有测试**。

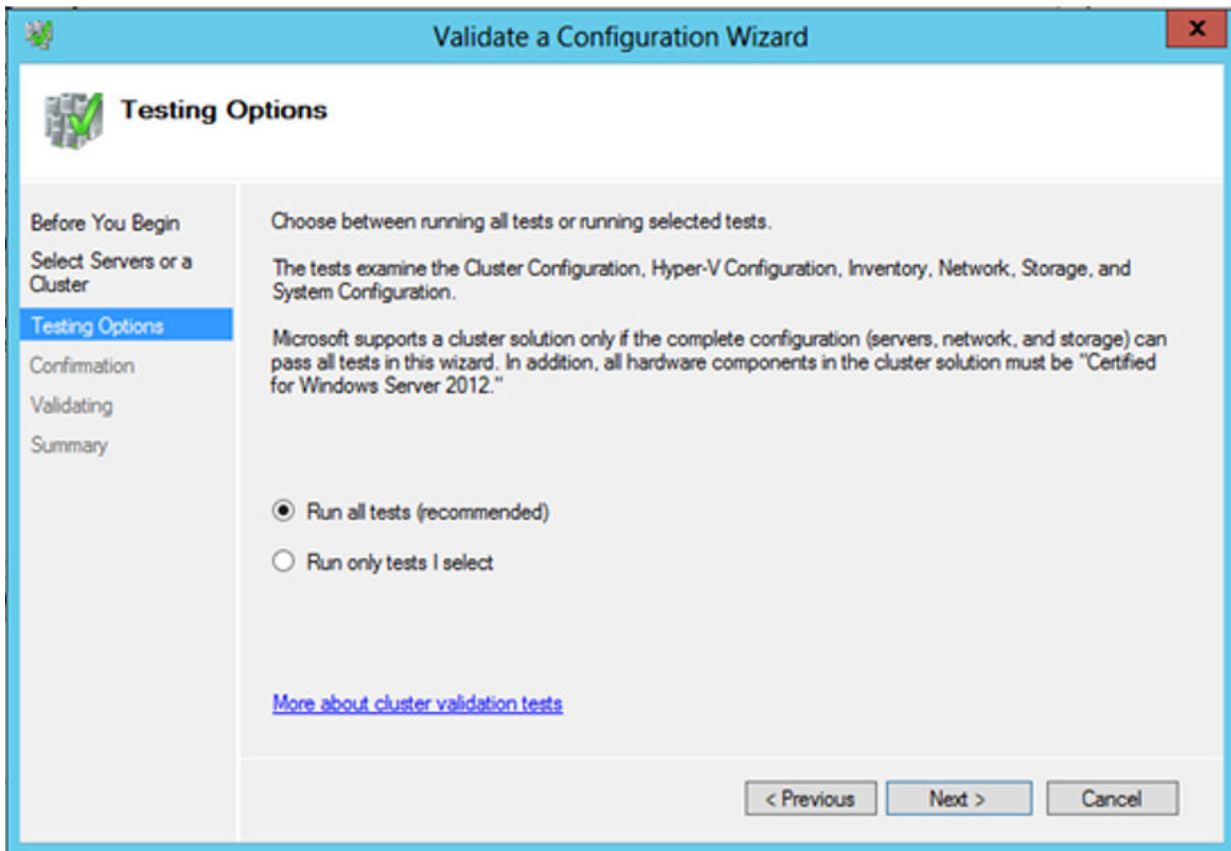


图 9: 测试选项

- 10 单击下一步。确认屏幕显示所选服务器列表。

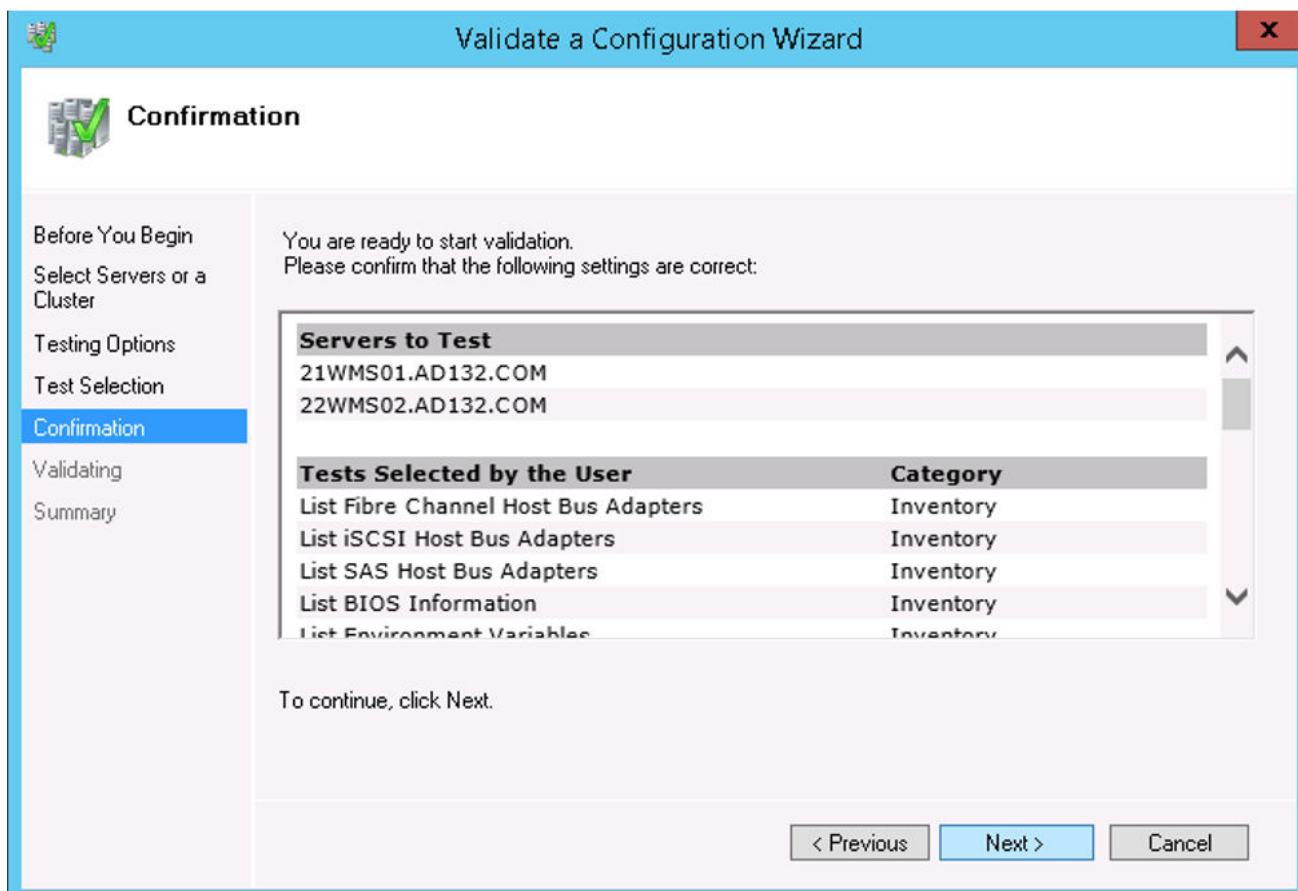


图 10: 确认

- 11 单击下一步。此时将显示摘要屏幕，其中含有故障转移群集验证报告。

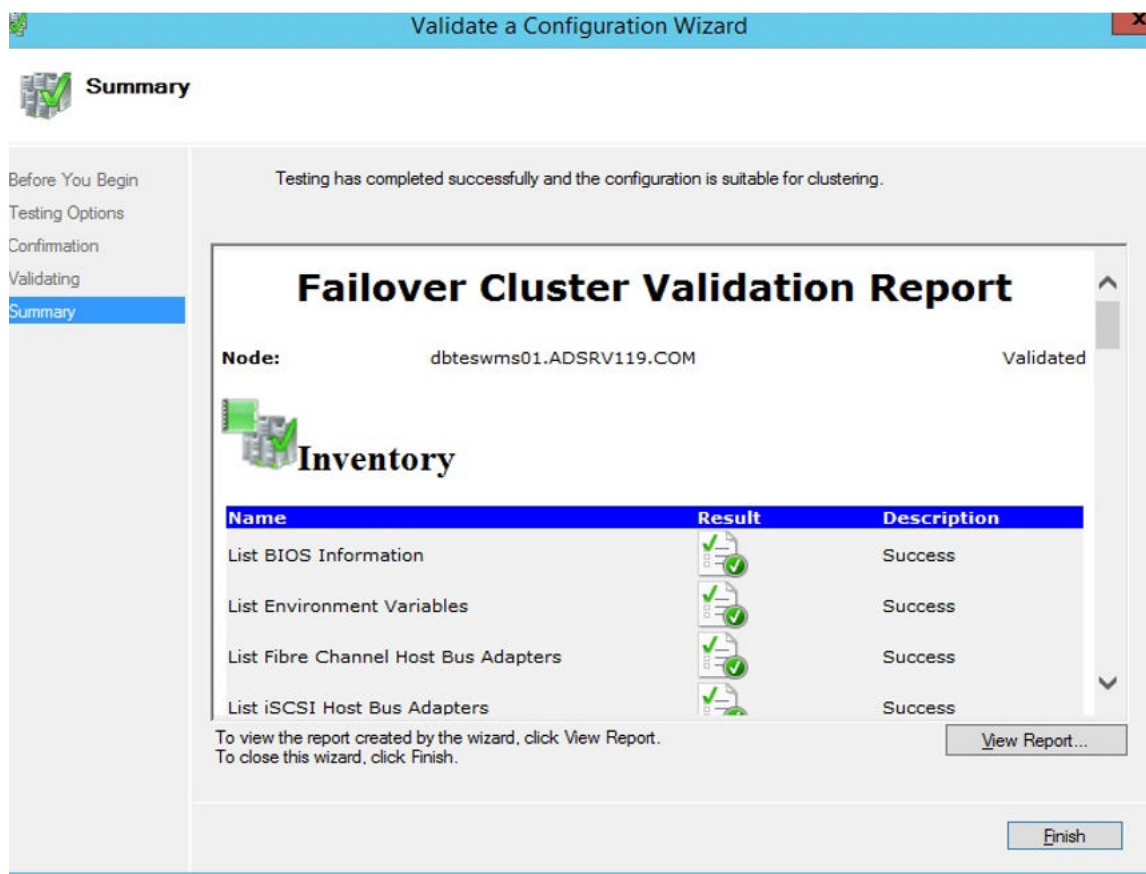


图 11: 测试摘要详细信息

- 12 单击**查看报告**以检查报告。如果状态是**通过**，您可以继续执行下一步。如果状态是**失败**，您必须在继续执行下一步之前先修复错误。

① **注:** 如果没有验证错误，则会显示创建群集向导屏幕。

- 13 单击**下一步**并在**群集名称**字段中键入群集的名称，然后选择系统的 IP 地址。
- 14 单击**下一步**，将显示**确认**屏幕。
- 15 单击**下一步**以在所选群集节点上创建群集，然后单击**查看报告**以查看警告消息。
- 16 单击**完成**以创建故障转移群集。

创建文件共享见证

文件共享见证是群集计算机具有读/写访问权限的基本文件共享。文件共享必须位于群集所在域内的独立 Windows Server 2012 上。要创建文件共享见证，请执行以下操作：

- 1 在 Microsoft Windows Server 2012 中，右键单击**开始**菜单，然后选择**服务器管理器**以启动“服务器管理器”仪表板
- 2 单击**服务器管理器**图标以访问服务器管理器。
- 3 请转至**文件和存储服务共享**，然后单击**任务**。
- 4 单击**网络共享**。将显示**新共享向导**。
- 5 单击**选择配置文件**以创建文件共享，然后单击**下一步**。
- 6 在**共享位置**屏幕中，选择文件共享的服务器和共享位置，然后单击**下一步**。
- 7 在**共享名称**屏幕中，在**共享名称**字段中键入名称，然后单击**下一步**，直到显示**确认**屏幕。
- 8 单击**创建**以创建文件共享，并且**查看结果**屏幕将会出现并显示**已完成**状态，表示已创建文件共享见证且未发生任何错误。
- 9 单击**关闭**以退出。

配置群集仲裁设置

群集配置数据库（也称为仲裁）包含一些详细信息，例如在任何给定的时间，群集设置中的哪个服务器应该处于活动状态。要配置群集仲裁设置，请执行以下操作：

- 1 在 Microsoft Windows Server 2012 中，单击**开始**以打开**开始**屏幕，然后单击**服务器管理器**以启动“服务器管理器”仪表板。
- 2 单击**服务器管理器**图标以访问服务器管理器，然后单击**故障转移群集管理器**以启动群集管理器。
- 3 右键单击群集节点，然后转至**更多操作配置群集仲裁设置**以显示**配置群集仲裁**向导屏幕。
- 4 单击**下一步**。从**选择仲裁配置选项**屏幕中选择**选择仲裁见证**。

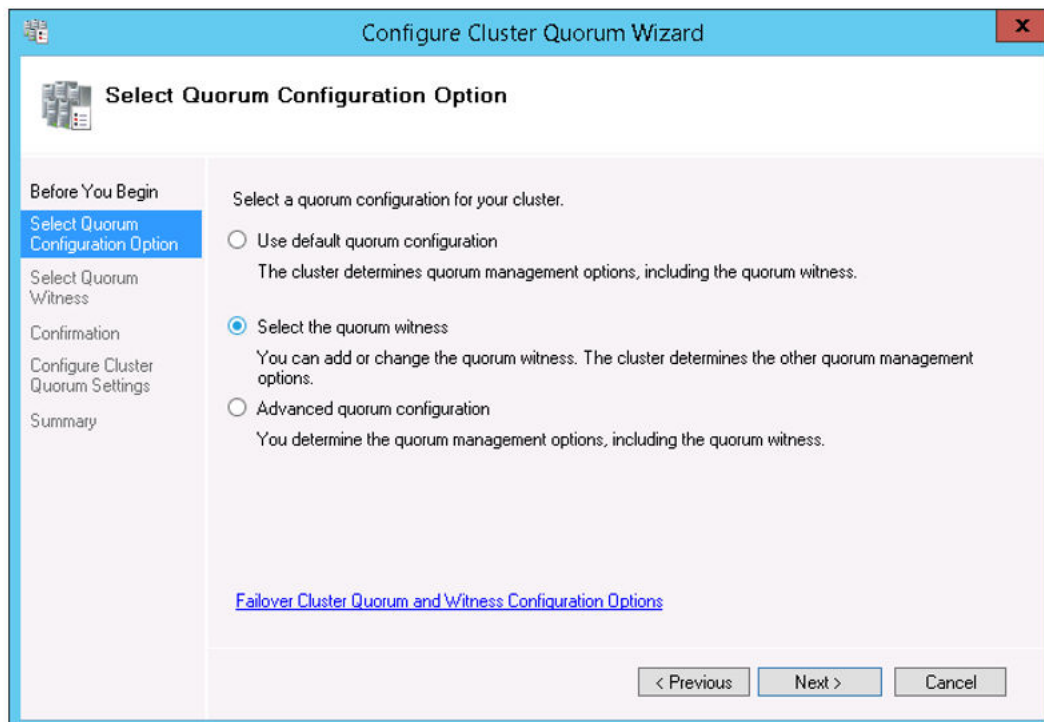


图 12: 仲裁群集向导

- 5 单击**下一步**。从**选择投票配置**屏幕中选择**所有节点**。

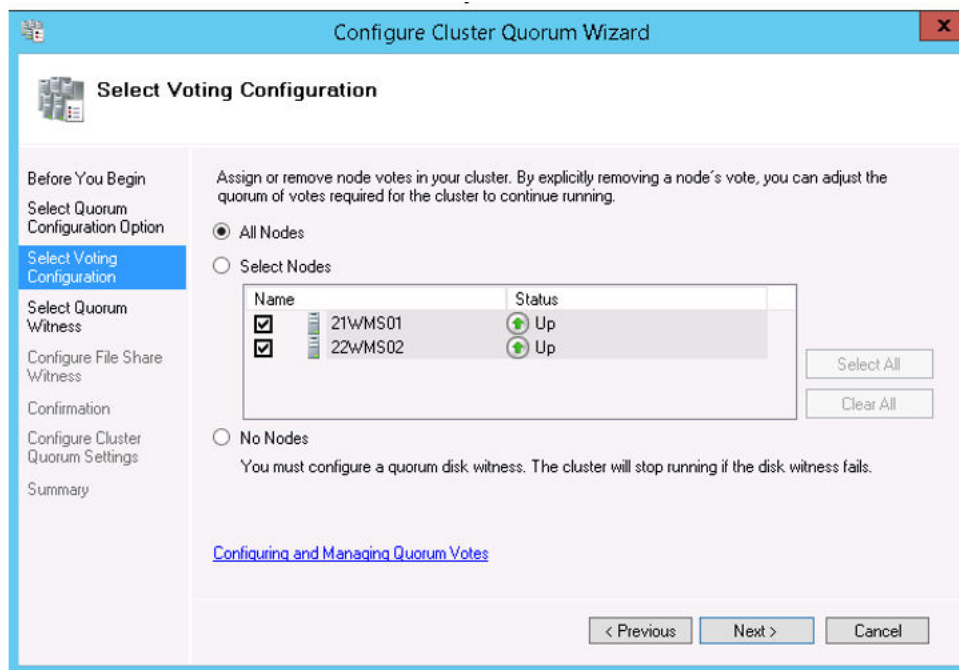


图 13: 选择投票配置

- 6 单击下一步。从选择仲裁见证屏幕中选择配置文件共享见证。
- 7 单击下一步，然后在配置文件共享见证屏幕的文件共享路径字段中，键入共享路径。

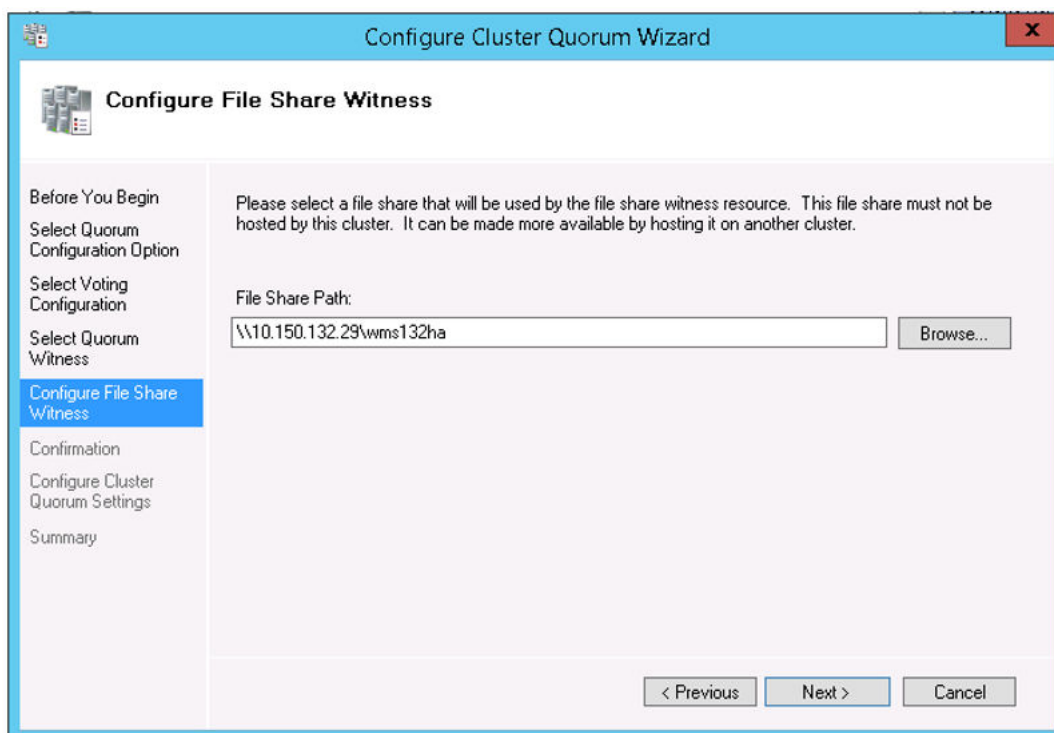


图 14: 配置文件共享见证

- 8 单击下一步。将显示具有已配置仲裁设置的摘要屏幕。

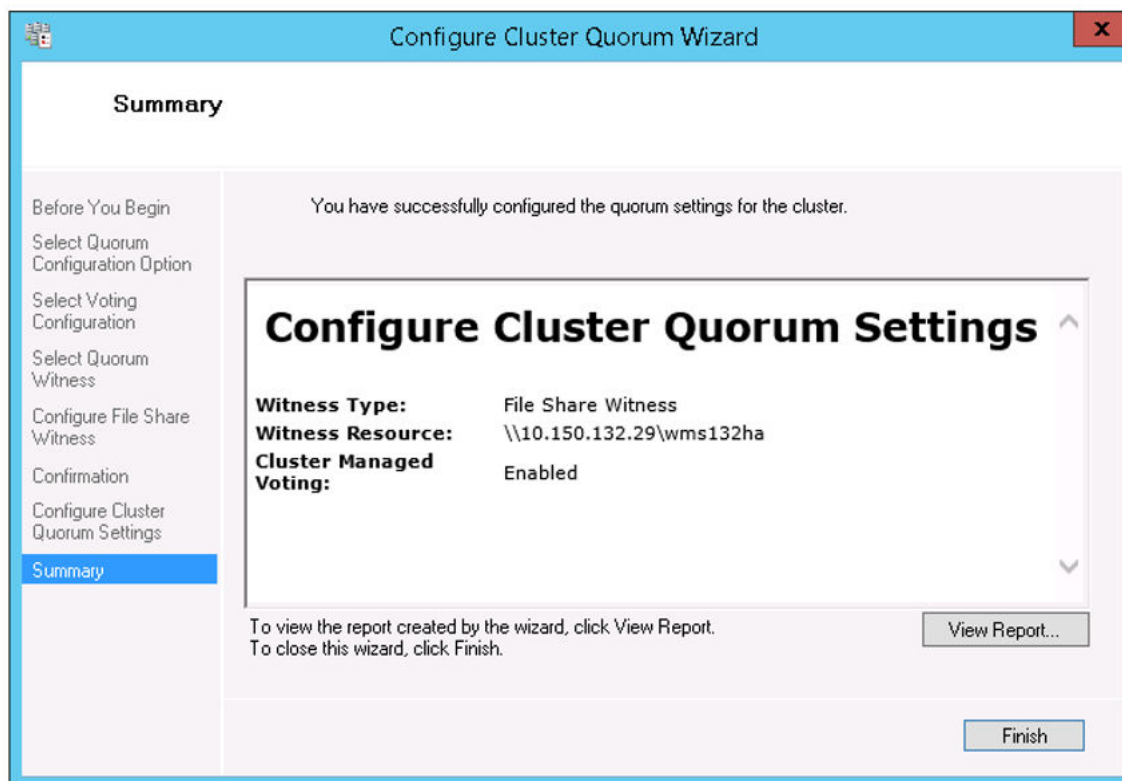


图 15: 仲裁设置摘要

- 9 单击**完成**以完成仲裁设置。

创建群集角色

在创建故障转移群集之后，您可以创建群集角色以管理群集工作负载。在创建群集角色之前，确保 Wyse Management Suite 已安装在服务器上并指向远程数据库。

要创建群集角色，请执行以下操作：

- 1 在 Microsoft Windows Server 2012 中，右键单击**开始**菜单，然后选择**服务器管理器**以启动“服务器管理器”仪表板
- 2 单击**故障转移群集管理器**以启动群集管理器。
- 3 右键单击**角色**，然后选择**配置角色**以显示高可用性向导屏幕。

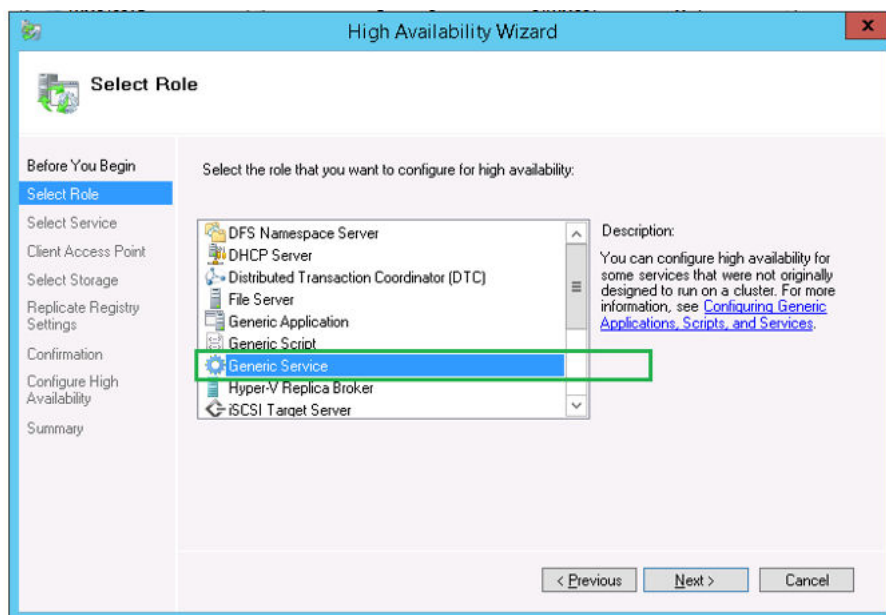


图 16: 高可用性向导

- 4 选择通用服务，然后单击下一步以查看选择服务屏幕。

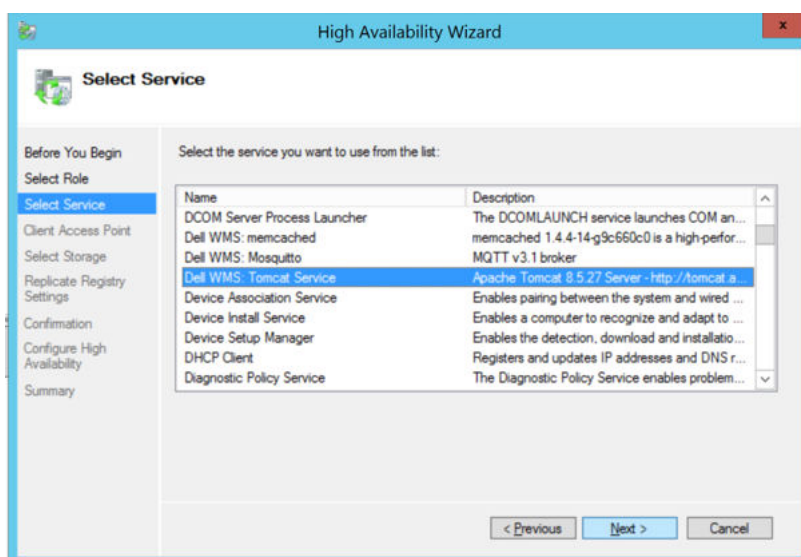


图 17: 选择服务

- 5 选择 **Dell WMS: Tomcat 服务**，然后单击下一步。

注: 只有在安装 Wyse Management Suite 版本 1.3 之后，才能将与 Wyse Management Suite 版本 1.3 相关的服务添加到群集。

此时将显示高可用性向导屏幕，您需要在其中创建客户端访问点以及在 Windows Server 2012 和 Wyse Management Suite 之间建立连接。

- 6 在名称字段中键入网络名称，然后单击下一步。此时将显示确认屏幕，其中含有服务器的网络名称和 IP 地址详细信息。

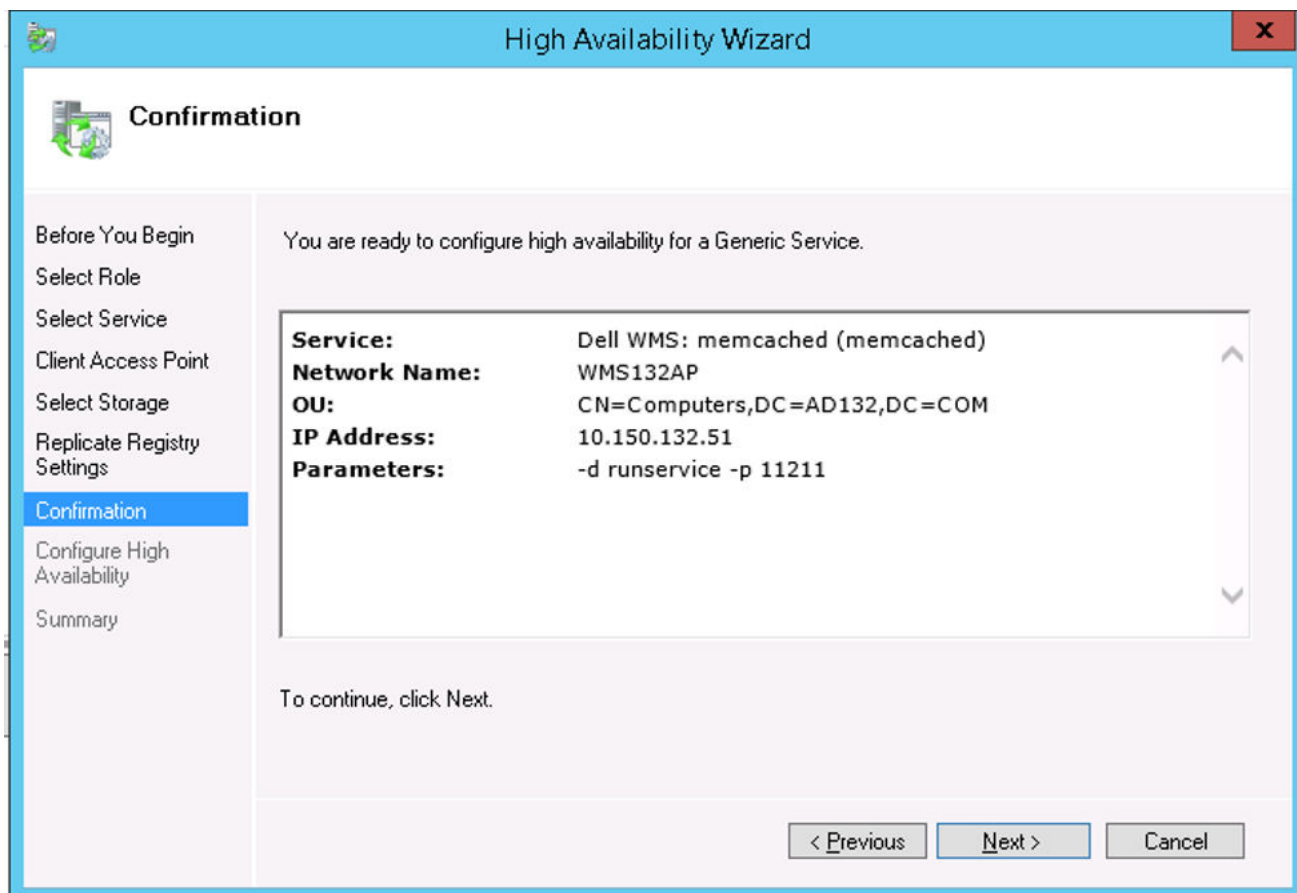


图 18: 确认

- 7 单击下一步以完成该过程。
- 8 要添加其他 Wyse Management Suite 服务作为群集的一部分，请启动故障转移群集管理器，然后转至操作角色以显示您已创建的网络名称。
- 9 单击网络名称，然后转至添加资源通用服务。
- 10 从新建资源向导屏幕中，选择以下服务（需要添加它们作为群集的一部分）：
 - a Dell WMS: Mosquitto >> MQTT Broker
 - b Dell WMS: memcached
- 11 单击下一步以完成任务。
已添加为群集一部分的 Wyse Management Suite 服务以正在运行状态显示。

实现 MySQL InnoDB 的高可用性

以下步骤解释了如何实现 MySQL InnoDB 的高可用性：

- 1 检查 MySQL InnoDB 服务器实例 — 请参阅 [Create MySQL InnoDB cluster](#) (创建 MySQL InnoDB 群集)。
- 2 将服务器或节点添加到 MySQL InnoDB — 请参阅 [Adding server or node to MySQL InnoDB cluster](#) (将服务器或节点添加到 MySQL InnoDB 群集)。
- 3 创建 MySQL 路由器 — 请参阅 [Creating MySQL Router](#) (创建 MySQL 路由器)

通过 MySQL InnoDB 提供高可用性

MySQL InnoDB 群集为 MySQL 提供了完整的高可用性解决方案。客户端应用程序通过使用 MySQL 路由器连接到主要节点。如果主要节点出现故障，辅助节点会自动提升为主要节点角色，并且 MySQL 路由器会将请求路由到新的主要节点。

MySQL InnoDB 群集的组件有：

- MySQL 服务器
- MySQL 路由器

安装 MySQL InnoDB 数据库

要安装 MySQL InnoDB，请参阅 [dev.mysql.com](#)。

要根据高可用性设置来设置环境，请参阅 [dev.mysql.com](#)。

检查 MySQL InnoDB 服务器实例

在将 MySQL InnoDB 添加到群集设置之前，请验证是否根据群集要求创建了 MySQL InnoDB。

您必须以 **root** 用户身份登录才能运行命令，并在每次运行一组命令之后重新启动系统。

运行以下命令以验证 MySQL InnoDB 服务器实例是否符合已配置的群集要求：

① | **注：** 在您的工作场所中使用的每个系统的 IP 地址各不相同，因此以下命令仅用作示例。

- To check that the MySQL InnoDB is created as per the requirements, run the following commands at the command prompt:
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address1')`
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address2')`
- `mysql-js> dba.checkInstanceConfiguration('root@IP Address3')`

```
MySQL Shell 8.0.11
Copyright (c) 2016, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type '\help' or '? ' for help; '\quit' to exit.

MySQL JS> dba.configureLocalInstance('root@10.150.132.23:3306')
Please provide the password for 'root@10.150.132.23:3306': *****
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as 23MySQL01
Clients and other cluster members will communicate with it through this address by default. If this is not correct, the report_host MySQL system variable should be changed.

Some configuration options need to be fixed:
+-----+-----+-----+-----+
| Variable | Current Value | Required Value | Note |
+-----+-----+-----+-----+
| binlog_checksum | CRC32 | NONE | Update the server variable |
| enforce_gtid_consistency | OFF | ON | Update read-only variable and restart the server |
| gtid_mode | OFF | ON | Update read-only variable and restart the server |
| log_bin | 0 | 1 | Update read-only variable and restart the server |
| log_slave_updates | 0 | ON | Update read-only variable and restart the server |
| master_info_repository | FILE | TABLE | Update read-only variable and restart the server |
| relay_log_info_repository | FILE | TABLE | Update read-only variable and restart the server |
| transaction_write_set_extraction | OFF | XXHASH64 | Update read-only variable and restart the server |
+-----+-----+-----+-----+

The following variable needs to be changed, but cannot be done dynamically: 'log_bin'
Detecting the configuration file...
Found configuration file at standard location: C:\ProgramData\MySQL\MySQL Server 5.7\my.ini
Do you want to modify this file? [y/N]: y
Do you want to perform the required configuration changes? [y/n]: y
Configuring instance...
The instance '10.150.132.23:3306' was configured for cluster usage.
MySQL server needs to be restarted for configuration changes to take effect.
MySQL JS> _
```

图 19: MySQL 命令提示符

要检查是否已在所有三个群集节点上创建 MySQL InnoDB，请在命令提示符下运行以下命令：

- mysql-js> dba.checkInstanceConfiguration('root@IPAddress1:3306')
- mysql-js> dba.checkInstanceConfiguration('root@IPAddress2:3306')
- mysql-js> dba.checkInstanceConfiguration('root@IPAddress3:3306')

创建 MySQL InnoDB 的群集实例

在服务器上安装 MySQL InnoDB 实例之后，创建群集实例。
要创建 MySQL InnoDB 的群集，请执行以下操作：

- 从命令提示符下，以管理员用户的身份登录。此用户帐户应具有管理权限。例如，DBAdmin。下面的屏幕显示了以 root 用户的身份登录的示例。

```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
> "status": "ok"

MySQL JS> \connect root@10.150.132.23:3306
Creating a session to 'root@10.150.132.23:3306'
Enter password: *****
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 7
Server version: 5.7.22-log MySQL Community Server (GPL)
No default schema selected; type \use <schema> to set one.
MySQL [10.150.132.23] JS> _
```

图 20: 登录提示符

- 运行以下命令创建具有唯一名称的群集。例如，MySQLCluster。
MySql JS> var cluster = dba.createCluster('MySQLCluster')
- 运行以下命令以检查群集状态。
MySql JS> Cluster.status()
已创建群集的状态显示为**联机**，表示已成功创建群集。


```
Select C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe

MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS> dba.getCluster()
<Cluster:MySQLCluster>

MySQL [10.150.132.23] JS> Cluster.status()
{
  "clusterName": "MySQLCluster",
  "defaultReplicaSet": {
    "name": "default",
    "primary": "10.150.132.23:3306",
    "ssl": "DISABLED",
    "status": "OK_NO_TOLERANCE",
    "statusText": "Cluster is NOT tolerant to any failures.",
    "topology": {
      "10.150.132.23:3306": {
        "address": "10.150.132.23:3306",
        "mode": "R/W",
        "readReplicas": {},
        "role": "HA",
        "status": "ONLINE"
      }
    }
  },
  "groupInformationSourceMember": "mysql://root@10.150.132.23:3306"
}

MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS>
MySQL [10.150.132.23] JS>
```

图 21: 确认屏幕

添加服务器实例至 MySQL InnoDB 群集

必须将服务器实例添加到 MySQL InnoDB 群集作为主要节点或次要节点。

执行以下操作来将服务器实例添加到 MySQL InnoDB 群集：

- 1 从命令提示符下，以数据库管理员用户身份登录。
- 2 运行以下命令，将服务器实例添加到 MySQL InnoDB 群集：

```
cluster.addInstance('root@IPAddress2:3306')
```

```
cluster.addInstance('root@IPAddress3:3306')
```

① 注: IP 地址和端口号仅用作示例，它们因在您的工作场所中要使用的系统而异。

- 3 运行以下命令以检查服务器实例的状态：

```
cluster.status()
```

① 注: 所有节点都应显示为联机状态，表示节点已成功添加到 MySQL InnoDB 群集设置。

```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
MySQL [10.150.132.231 JS>
MySQL [10.150.132.231 JS> var cluster = dba.getCluster<
MySQL [10.150.132.231 JS> dba.getCluster<
<Cluster:MySQLCluster>

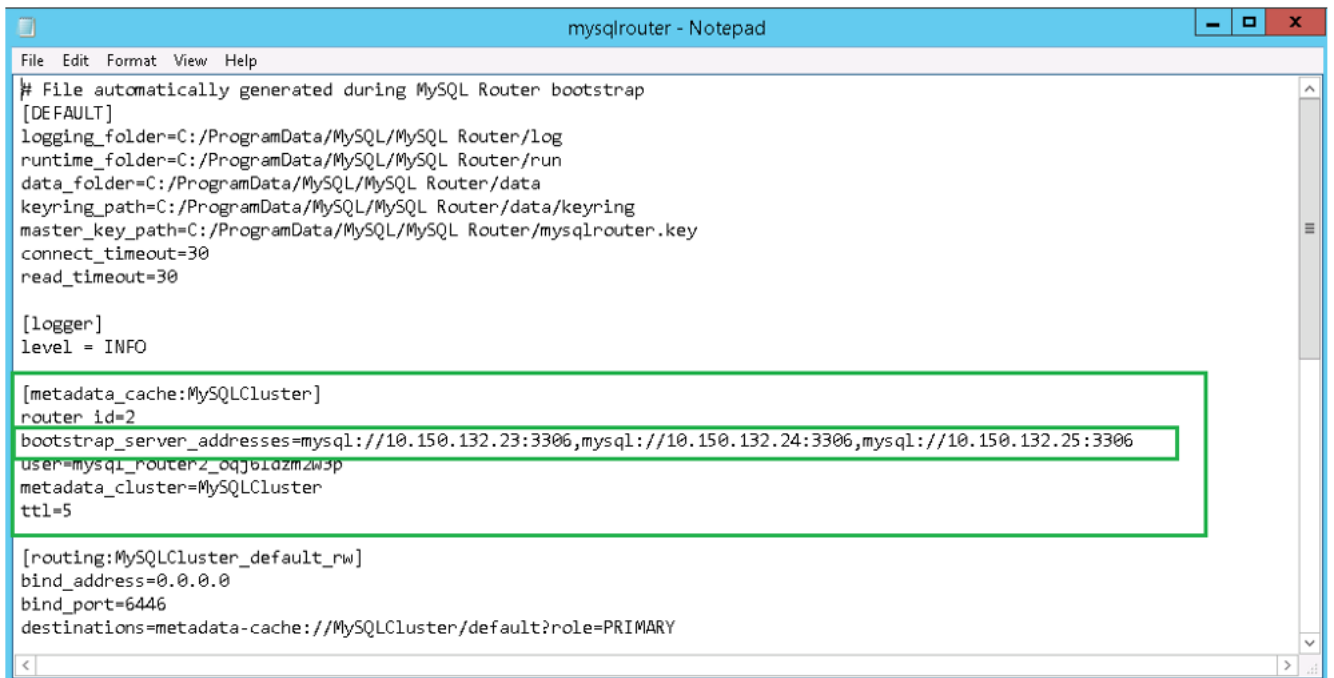
MySQL [10.150.132.231 JS> Cluster.status<
<
  "clusterName": "MySQLCluster",
  "defaultReplicaSet": {
    "name": "default",
    "primary": "10.150.132.23:3306",
    "ssl": "DISABLED",
    "status": "OK",
    "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
    "topology": {
      "10.150.132.23:3306": {
        "address": "10.150.132.23:3306",
        "mode": "R/W",
        "readReplicas": {},
        "role": "HA",
        "status": "ONLINE"
      },
      "10.150.132.24:3306": {
        "address": "10.150.132.24:3306",
        "mode": "R/O",
        "readReplicas": {},
        "role": "HA",
        "status": "ONLINE"
      },
      "10.150.132.25:3306": {
        "address": "10.150.132.25:3306",
        "mode": "R/O",
        "readReplicas": {},
        "role": "HA",
        "status": "ONLINE"
      }
    }
  },
  "groupInformationSourceMember": "mysql://root@10.150.132.23:3306"
>
MySQL [10.150.132.231 JS>
MySQL [10.150.132.231 JS>
```

图 22: 群集状态

配置 MySQL 路由器

MySQL 路由器在 Wyse Management Suite 和 MySQL InnoDB 之间建立通信网络。
要安装 MySQL 路由器，请执行下列操作：

- 1 登录到 Windows Server 2012 以安装 MySQL 路由器。有关更多信息，请参阅 [MySQL Router Installation](#)（MySQL 路由器安装）。
- 2 从**选择产品和功能**屏幕中选择 **MySQL 路由器**，然后单击下一步，直到显示**安装完成**屏幕。
- 3 浏览到 `\ProgramData\MySQL\MySQL Router` 目录，然后打开文件 `mysqlrouter.conf`，检查引导程序属性是否含有属于群集设置一部分的所有已配置 MySQL 服务器。



```
# File automatically generated during MySQL Router bootstrap
[DEFAULT]
logging_folder=C:/ProgramData/MySQL/MySQL Router/log
runtime_folder=C:/ProgramData/MySQL/MySQL Router/run
data_folder=C:/ProgramData/MySQL/MySQL Router/data
keyring_path=C:/ProgramData/MySQL/MySQL Router/data/keyring
master_key_path=C:/ProgramData/MySQL/MySQL Router/mysqlrouter.key
connect_timeout=30
read_timeout=30

[logger]
level = INFO

[metadata_cache:MySQLCluster]
router_id=2
bootstrap_server_addresses=mysql://10.150.132.23:3306,mysql://10.150.132.24:3306,mysql://10.150.132.25:3306
user=mysql_router2_oqjb10zm2w3p
metadata_cluster=MySQLCluster
ttl=5

[routing:MySQLCluster_default_rw]
bind_address=0.0.0.0
bind_port=6446
destinations=metadata-cache://MySQLCluster/default?role=PRIMARY
```

图 23: Bootstrap 服务器地址

在 MySQL InnoDB 服务器上创建数据库和用户

必须在 MySQL InnoDB 服务器上创建具有管理员权限的数据库和用户帐户。

要在 MySQL InnoDB 服务器上创建数据库，请运行以下 SQL 命令：

```
Create Database stratus DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_unicode_ci;
CREATE USER 'STRATUS'@'LOCALHOST';
CREATE USER 'STRATUS'@'IP ADDRESS';
SET PASSWORD FOR 'STRATUS'@'LOCALHOST' = PASSWORD <db_password>;
SET PASSWORD FOR 'STRATUS'@ <IP_Address> = PASSWORD <db_password>;
GRANT ALL PRIVILEGES ON *.* TO 'STRATUS'@<IP_Address> IDENTIFIED BY <db_password> WITH GRANT
OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'STRATUS'@'LOCALHOST' IDENTIFIED BY <db_password> WITH GRANT
OPTION;
```

① 注: 您可为 Network /Subnet 键入 Wildcard，或者键入将在其中安装 Wyse Management Suite 应用程序服务器的 Multiple Single 主机条目，而不是 IP 地址。

实现 MongoDB 上的高可用性

以下步骤解释了如何在 MongoDB 上实现高可用性：

- 1 安装 MongoDB - 请参阅 [Installing MongoDB](#) (安装 MongoDB)。
- 2 创建副本服务器 - 请参阅 [Creating Replica servers](#) (创建副本服务器)。
- 3 创建 Stratus 用户 - 请参阅 [Creating Stratus user account](#) (创建 Stratus 用户帐户)。
- 4 创建根用户, 请参阅 [Creating root user for MongoDB](#) (创建 MongoDB 的根用户)。
- 5 编辑 MongoDB 配置文件 - 请参阅 [Editing MongoDB configuration file](#) (编辑 MongoDB 配置文件)。

安装 MongoDB

要在所有三个节点上安装 MongoDB, 请执行以下操作：

① 注: 有关安装 MongoDB 的信息, 请参阅 [Install MongoDB](#) (安装 MongoDB)

- 1 复制系统上的 MongoDB 安装文件。
- 2 在 Drive C 以外的辅助驱动器上创建两个文件夹：Data\log 和 data\db。

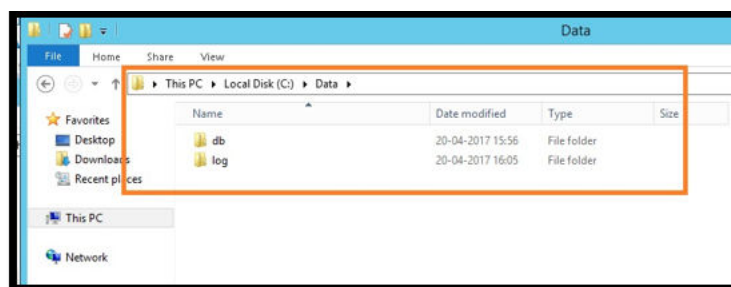


图 24: 数据文件

- 3 转至您在其中复制 MongoDB 安装文件的文件夹, 并从命令提示符中创建一个文件 mongod.cfg。

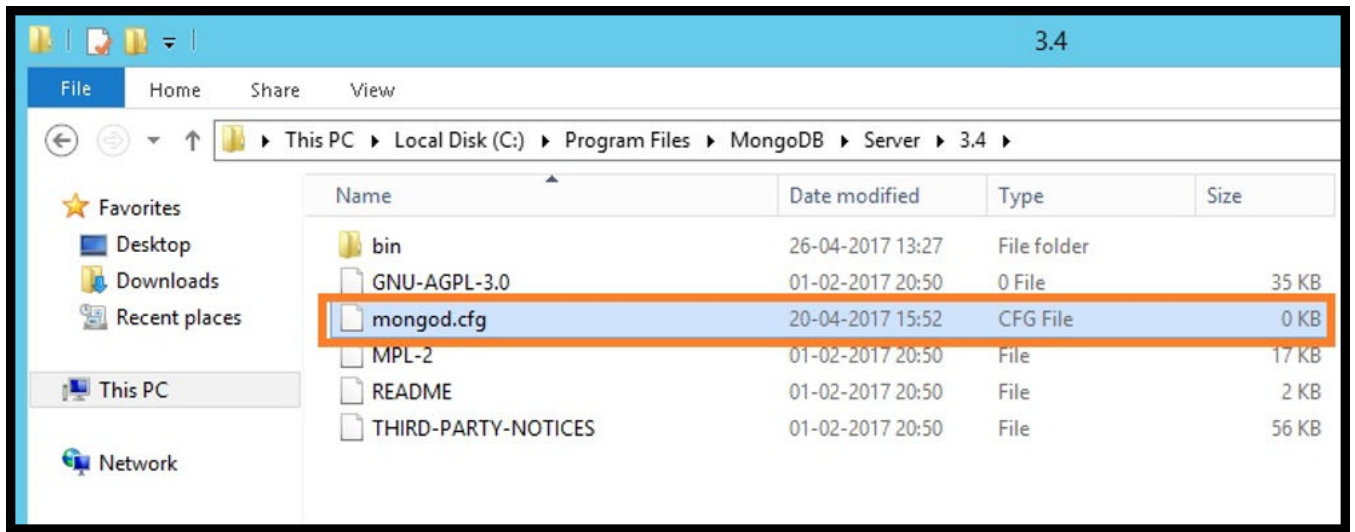


图 25: mongod.cfg 文件

- 4 在文本编辑器中打开 mongod.cfg 文件，添加以下条目：
 - a SystemLog:destination: file
 - b path: c:\data\log\mongod.log
 - c Storage: dbpath: c:\data\db
 - 5 保存文件。
 - 6 登录到 MongoDB 服务器。
 - 7 运行以下命令以启动 MongoDB 服务：
 - a C:\MongoDB\bin>.\mongod.exe --config c:\MongoDB\mongod.cfg --install
 - b C:\MongoDB\bin>net start mongod
- 将会显示消息 **MongoDB 服务正在启动**。
- 8 将工作目录更改为 \MongoDB\bin。
 - 9 在命令提示符下，运行 Mongo.exe 以完成 MongoDB 安装。

创建 MongoDB 数据库的副本服务器

您必须创建副本服务器以避免任何系统故障。副本服务器应具有存储多个分布式读取操作的容量。

有关创建副本服务器的更多信息，请参阅 Deploy a Replica Server Set（部署副本服务器集），网址为 docs.mongodb.com/manual。

创建数据库用户

使用 Wyse Management Suite 创建用户（例如，DBUser）以访问 MongoDB。

① | 注：数据库用户和密码仅用作示例，可在您的工作场所使用不同的名称和密码创建。

运行以下命令来创建 DBUser：

```
db.createUser( {
  user: "DBUser",
  pwd: <db_password>,
  roles: [ { role: "userAdminAnyDatabase", db: "admin" },
    { role: "dbAdminAnyDatabase", db: "admin" },
    { role: "readWriteAnyDatabase", db: "admin" },
    { role: "dbOwner", db: "DBUser" } ]
})
```

创建 MongoDB 的 DBadmin 用户

使用在上一节中创建的用户帐户登录到 MongoDB。将创建具有管理权限的 DBadmin 用户。

运行以下命令来创建 DBadmin 用户：

```
mongo -uDBUser -pPassword admin
use admin
db.createUser( {
  user: "DBAdmin",
  pwd: <DBAdmin user password>,
  roles: [ { role: "DBAdmin", db: "admin" } ]
})
```

编辑 mongod.cfg 文件

您必须编辑 mongod.cfg 文件以启用 MongoDB 数据库的安全性。

- 1 作为您已创建的根用户登录到 MongoDB，然后运行以下命令：

```
mongo -uroot -pAdmin#123 admin
```

- 2 转至 \data\bin\mongod.cfg 目录，并在文本编辑器中打开 mongod.cfg 文件。
- 3 编辑 mongod.cfg 文件，如以下示例中所示：

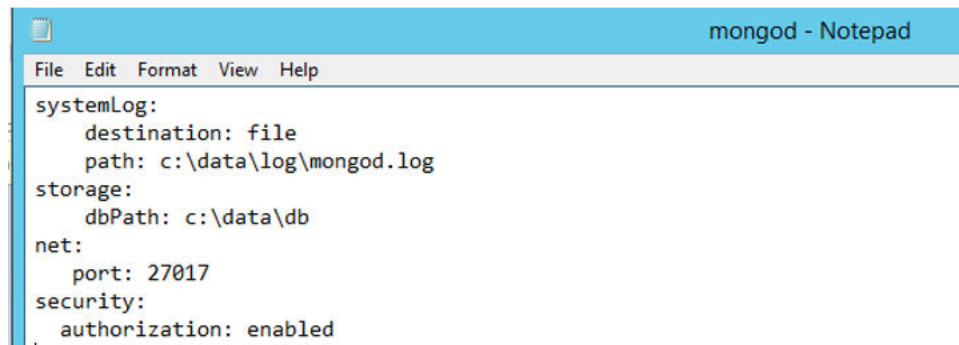


图 26: 编辑 mongod.cfg

```
systemLog:
destination: file
path: c:\data\log\mongod.log
storage:
dbPath: c:\data\db\Mongo
net:
port: 27017
security:
authorization: enabled
```

① | 注: 端口号将根据工作场所中的系统更改。

- 4 保存 mongod.cfg 并退出。

服务器上启动复制

请确保您在 Windows 上禁用防火墙并停止 Tomcat 服务器（如果它们正在运行）。

- 1 作为您已创建的根用户登录到 MongoDB，然后运行以下命令：

```
mongo -uroot -pAdmin#123 admin
```

2 转至 `\data\bin\mongod.cfg` 目录，并在文本编辑器中打开 `mongod.cfg` 文件。

3 在 `mongod.cfg` 文件中添加下面三行：

```
keyFile: c:\data\log\mongod.key.txt
```

```
replication:
```

```
  replSetName: wms
```

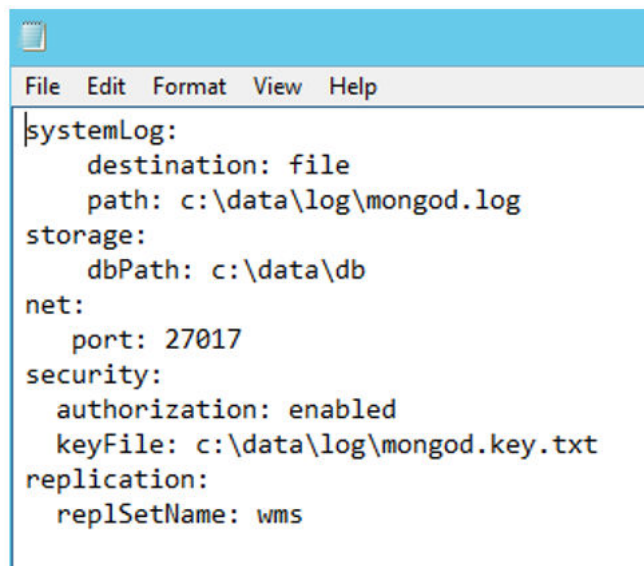


图 27: 启用安全保护

4 创建 `mongod.key.txt` 文件并在所有三个服务器上复制。

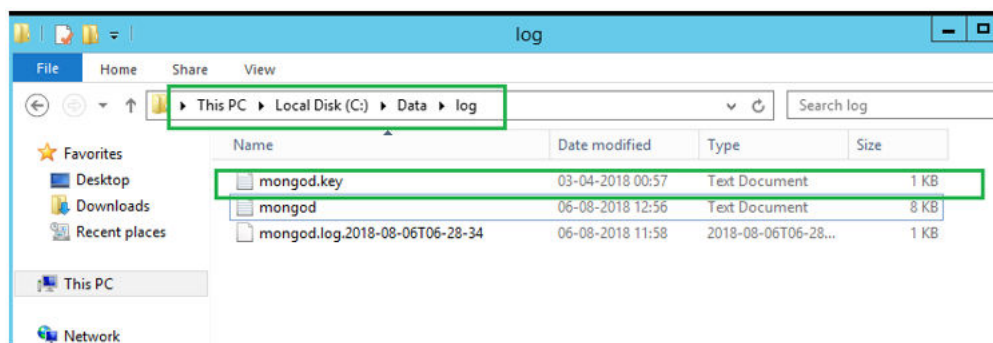


图 28: 复制 `mongod` 密钥文件

5 当您复制文件后，通过运行以下命令停止 `mongod` 服务：

```
net stop mongod
```

6 使用 `DBAdmin` 身份登录后，在 `MongoDB` 群集的主要节点上启动复制，然后运行以下命令：

```
rs.initiate();
```

7 通过运行以下命令检查复制状态：

```
rs.status();
```

```

{
  "set" : "wms",
  "date" : ISODate("2018-08-06T09:12:23.235Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1533546742, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "26MONGODB01:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 445,
      "optime" : {
        "ts" : Timestamp(1533546742, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2018-08-06T09:12:22Z"),
      "infoMessage" : "could not find member to sync from",
      "electionTime" : Timestamp(1533546710, 2),
      "electionDate" : ISODate("2018-08-06T09:11:50Z"),
      "configVersion" : 1,
      "self" : true
    }
  ],
  "ok" : 1
}

```

图 29: Replication status（复制状态）

- 8 启动 mongod 服务并将次要节点添加到 MongoDB 群集中的第二个节点和第三个节点：

```
rs.add("IPAddress2:27017")
```

```
rs.add("IPAddress3:27017")
```

① | 注: 端口号会因您网络和系统中的系统而异。

- 9 在 MongoDB 群集中添加节点之后，通过针对主要节点和次要节点运行以下命令，检查复制状态：

```
rs.status();
```

```

{ "set" : "ums",
  "date" : ISODate("2018-08-06T09:20:22.109Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1533547215, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "26MONGODB01:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 924,
      "optime" : {
        "ts" : Timestamp(1533547215, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
      "electionTime" : Timestamp(1533546710, 2),
      "electionDate" : ISODate("2018-08-06T09:11:50Z"),
      "configVersion" : 3,
      "self" : true
    }
  ]
}

```

PRIMARY MONGO DB Server Details

图 30: 主服务器状态

```

"configVersion" : 3,
"self" : true
},
{
  "_id" : 1,
  "name" : "10.150.132.27:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 14,
  "optime" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
  "optimeDurableDate" : ISODate("2018-08-06T09:20:15Z"),
  "lastHeartbeat" : ISODate("2018-08-06T09:20:22.007Z"),
  "lastHeartbeatRecv" : ISODate("2018-08-06T09:20:21.129Z"),
  "pingMs" : NumberLong(2),
  "syncingTo" : "26MONGODB01:27017",
  "configVersion" : 3
},
{
  "_id" : 2,
  "name" : "10.150.132.28:27017",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 6,
  "optime" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1533547215, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-08-06T09:20:15Z"),
  "optimeDurableDate" : ISODate("2018-08-06T09:20:15Z"),
  "lastHeartbeat" : ISODate("2018-08-06T09:20:22.013Z"),
  "lastHeartbeatRecv" : ISODate("2018-08-06T09:20:21.914Z"),
  "pingMs" : NumberLong(1),
  "configVersion" : 3
}
]
}

```

SECONDARY MONGO DB Servers' Details

Active

图 31: 次要服务器状态

实现 Teradici 设备的高可用性

Wyse Management Suite 使用托管在 Ubuntu 服务器 16.04.1 LTS 上的 HAProxy，在 EMSDK 设备之间执行负载平衡。HAProxy 是一个负载平衡器代理，根据它的配置方式，它还可以提供高可用性。它是适用于 TCP/HTTP 负载平衡器的热门开放源代码软件，也是可在 Linux 操作系统上运行的代理解决方案。最常见的用途是通过将工作负载分散在多台服务器上，提高服务器环境的性能和可靠性。以下要点说明了如何在 Linux 操作系统上使用 HAProxy 实现 Teradici 设备的高可用性：

- 只有一个 Teradici 服务器实例可以成为 Wyse Management Suite 高可用性的一部分。
- Teradici 设备支持需要安装 EMSDK。EMSDK 是集成到 Wyse Management Suite 中的一个软件组件，由 Teradici 提供。Wyse Management Suite 安装程序会安装 EMSDK，它可以安装 Wyse Management Suite 服务器上，也可以安装单独的服务器上。您至少需要 EMSDK 的两个实例来支持超过 5000 个设备，并且所有 EMSDK 服务器都应该在远程服务器上。
- 每台服务器上只能安装 EMSDK 的一个实例。
- Teradici 设备支持需要 PRO 许可证。
- Teradici 的高可用性将通过 HAProxy 提供。
- 如果 Teradici 服务器停机，则设备会自动重新连接到下一个可用的 EMSDK 服务器。

安装和配置 HAProxy

HAProxy 是 ThreadX 5x 设备的负载均衡器，在具有 HAProxy 版本 1.6 的 Ubuntu Linux 版本 16.04.1 上配置。

执行以下操作，在 Ubuntu Linux 系统上安装并配置 HAProxy：

- 1 使用在安装 Ubuntu 操作系统期间使用的用户凭据，登录到 Ubuntu 系统。
- 2 运行以下命令来安装 HAProxy

```
sudo apt-get install software-properties-common

sudo add-apt-repository ppa:vbernat/haproxy-1.6

sudo apt-get update

sudo apt-get install haproxy
```

- 3 运行以下命令以备份原始配置：

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/ haproxy.cfg.original
```

- 4 通过运行以下命令，在适当的文本编辑器中编辑 HAProxy 配置文件：

```
sudo nano /etc/haproxy/haproxy.cfg
```

在配置文件中，添加以下条目：

```
Global section: Maxconn <maximum number of connections>

Frontend tcp-in: bind :5172

Back end servers: server :5172

maxconn <maximum number of connections per Teradici device proxy server>
```

① | 注：管理员必须会添加超出客户机容量总数的额外后端服务器以获得无缝的故障转移。

- 5 键入 CTRL+O 保存对 haproxy.cfg 文件的更改。

以下文本是示例 HAProxy 配置文件：

```
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    daemon
    #maxconn is maximum allowed connections
    maxconn 60000

defaults
    log          global
    mode          tcp
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend fe_teradici_5172
    bind :5172
    mode tcp
    backlog 4096
    maxconn 70000
    default_backend be_teradici_5172

backend be_teradici_5172
    mode tcp
    option log-health-checks
    option tcplog
    balance leastconn
    server emsdk1 :5172 check server emsdk2 5172 check : timeout queue 5s timeout server
86400s
    option srvtcpka

#frontend fe_teradici_5172
#replace IP with IP of your Linux proxy machine bind Eg: 10.150.105.119:5172

#default_backend servers

#backend servers
#Add your multiple back end windows machine ip with 5172 as port
# maxconn represents number of connection- replace 10 with limit # (below 20000)
# "server1" "server2" are just names and not keywords

#server server1 10.150.105.121:5172 maxconn 20000 check
#server server2 10.150.105.124:5172 maxconn 20000 check
```

- 6 通过运行以下命令验证 HAProxy 配置：

```
sudo haproxy -f /etc/haproxy/haproxy.cfg -c
```

如果配置有效，则会显示消息：**配置有效。**

- 7 运行以下命令重新启动 HAProxy 服务：

```
Sudo service haproxy restart
```

- 8 运行以下命令以停止 HAProxy：

```
serviceSudo service haproxy stop
```

安装 Wyse Management Suite 服务器

确保在安装 Wyse Management Suite 服务器之前配置以下组件：

- 在两个节点上配置 Windows 故障转移群集
- 带副本集的 MongoDB 服务器正在运行
- MySQL InnoDB 群集设置正在运行
- 在两个节点上安装了 MySQL 的路由器

执行以下操作以安装 Wyse Management Suite 服务器：

- 1 启动 Wyse Management Suite v1.3 安装程序屏幕。
- 2 选择**自定义类型**和 **Teradici EMSDK**，然后单击**下一步**。
- 3 选择**外部 MongoDB** 选项（您已创建的带副本集的 MongoDB 群集）。例如，wms。在相应字段中键入远程主要 MongoDB 服务器信息、端口号和 MongoDB 用户名及密码，然后单击**下一步**。
- 4 选择适用于 MySQL 的**外部 MariaDB** 选项使用 MySQL 路由器地址（如果路由器安装在 Wyse Management Suite 服务器节点上，则使用本地主机）。

① | **注：**确保在 MySQL 服务器上创建 Stratus 用户帐户。

- 5 在**外部 Maria DB 服务器**字段中键入 MySQL 路由器信息，然后键入端口号。键入您最初创建的 MySQL 数据库用户帐户信息。此时会显示**端口选择**屏幕，其中含有端口详细信息。此端口由 MySQL 路由器使用。默认端口为 6466。但是，您也可以更改端口号。
- 6 键入具有管理权限的用户名和电子邮件地址及 Teradici EMSDK 端口号和 CIFS 用户帐户信息。
- 7 键入本地存储库的目标安装文件夹路径和共享 UNC 路径，然后单击**下一步**。此时会显示**安装成功**消息。

① | **注：**共享 UNC 路径应该在安装 Wyse Management Suite 应用程序的 Windows 服务器外部。

① | **注：**在节点 2 上安装 Wyse Management Suite 应用程序之前，请确保从在节点 1 上安装期间创建的 Wyse Management Suite 本地存储库中删除 ‘Data’ 文件夹。从共享 UNC WMS 本地存储库路径中删除 ‘Data’ 文件夹之后，就可以在 Windows 群集的节点 2 上安装 Wyse Management Suite 应用程序了。

在 Windows Server 2012 上安装 Wyse Management Suite

要在私有云中安装 Wyse Management Suite，请执行以下操作：

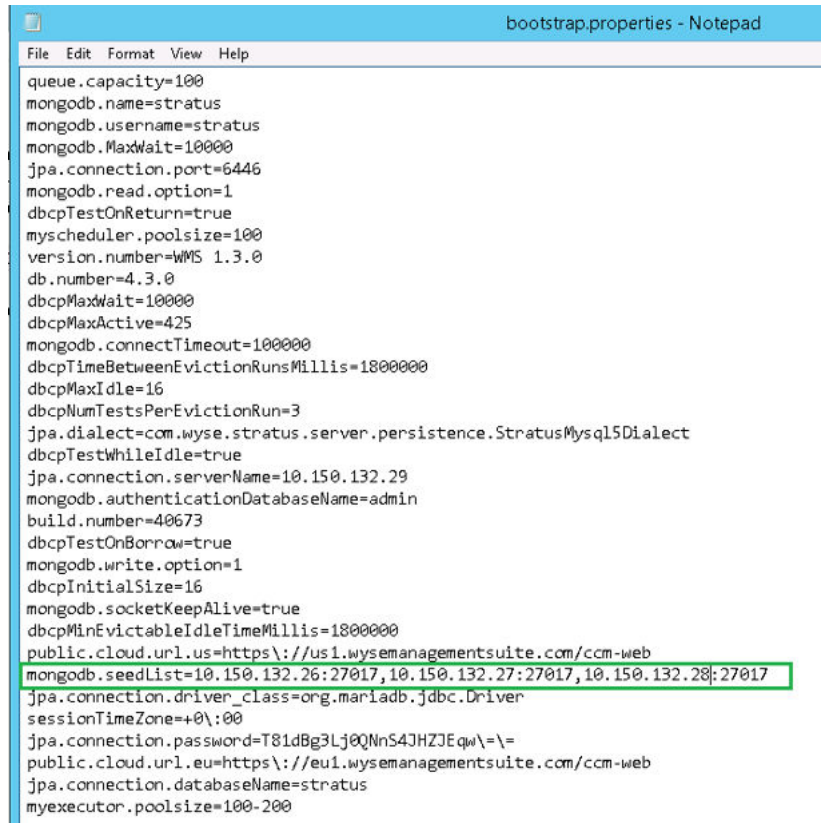
- 1 双击安装程序包。
- 2 在欢迎屏幕上，阅读许可协议并单击下一步。
- 3 选择要执行的**安装类型**，并单击下一步。可用的选项有：
 - 典型 — 所需用户交互最少，会安装嵌入式数据库。
 - 自定义 — 所需用户交互最多，推荐高级用户使用。
- 4 选择**自定义**作为**安装类型**，并单击下一步。
此时将显示 **Mongo 数据库服务器** 页面。
- 5 选择**外部 mongo DB** 选项。提供用户名、密码、数据库服务器详情和端口详情，随后单击下一步。
注：“端口”字段会填充默认端口，可以修改。
- 6 单击下一步，直到显示消息**安装成功**。
注：在服务器上或节点 2 上安装 Wyse Management Suite 之前，请确保从在服务器上或节点 1 上安装期间创建的本地存储库中删除 \Data 文件夹。

安装后检查

通过执行以下操作来检查 Wyse Management Suite 版本 1.3 的高可用性：

- 启动 Wyse Management Suite 管理门户并检查您是否能够使用 Web 界面登录。
- 编辑 Tomcat 服务器中 \Dell\WMS\Tomcat-8\webapps\ccm-web\WEB-INF\classes 文件夹（对于 MongoDB）下的 bootstrap.properties 文件，如下所示：

```
mongodb.seedList = MongoDBServer1_IP:27017, MongoDBServer2_IP:27017, MongoDBServer3_IP:27017
```



```
queue.capacity=100
mongodb.name=stratus
mongodb.username=stratus
mongodb.MaxWait=10000
jpa.connection.port=6446
mongodb.read.option=1
dbcpTestOnReturn=true
myscheduler.poolsize=100
version.number=WMS 1.3.0
db.number=4.3.0
dbcpMaxWait=10000
dbcpMaxActive=425
mongodb.connectTimeout=100000
dbcpTimeBetweenEvictionRunsMillis=1800000
dbcpMaxIdle=16
dbcpNumTestsPerEvictionRun=3
jpa.dialect=com.wyse.stratus.server.persistence.StratusMySQL5Dialect
dbcpTestWhileIdle=true
jpa.connection.serverName=10.150.132.29
mongodb.authenticationDatabaseName=admin
build.number=40673
dbcpTestOnBorrow=true
mongodb.write.option=1
dbcpInitialSize=16
mongodb.socketKeepAlive=true
dbcpMinEvictableIdleTimeMillis=1800000
mongodb.seedList=10.150.132.26:27017,10.150.132.27:27017,10.150.132.28:27017
jpa.connection.driver_class=org.mariadb.jdbc.Driver
sessionTimeZone=+0\:00
jpa.connection.password=T81dBg3Lj0QNnS4JHZJEqw\=\=
public.cloud.url.eu=https\://eu1.wysemanagementsuite.com/ccm-web
jpa.connection.databaseName=stratus
myexecutor.poolsize=100-200
```

图 32: 编辑 bootstrap.properties 文件

- 登录到 MongoDB 并更新 bootstrapProperties 表，并使用接入点的 Windows 群集虚拟 IP/主机名作为以下属性的值：

```
Stratusapp.server.url
Stratus.external.mqtt.url
Memcached.Servers
Mqtt.server.url
```

通过执行以下操作在 MongoDB 表中进行更改：

- 1 在 Stratus 数据库中，访问 集合，然后选择 bootstrapProperties 表。
- 2 在两个节点上更新 MySQL 表并重新启动 Tomcat。通过运行以下命令手动更新 MySQL 数据库表，让 ServersInCluster 中的 ServerIp 保持处于活动状态：

```
Update serversInCluster set ServerIp = '<VIP address of Windows Cluster>';
```

① | 注: 确保 **ServersInCluster** 表中只有一个记录, 如果有多个记录, 则删除多余的记录。

3 `Update queueLock set IpInLock = '<VIP address of Windows Cluster>';`

Troubleshooting

本节提供有关适用于群集设置的 Wyse Management Suite 版本 1.3 的故障排除信息。

- 问题：用于检查服务器安装问题的 Wyse Management Suite 日志文件位于何处。

解决方法：该日志文件位于 %temp% WMSInstall.log 文件夹中。

- 问题：与用于检查服务应用程序问题的日志文件相关的 Tomcat 服务位于何处。

解决方法：日志文件位于 \Program Files\DELL\WMS\Tomcat-8\stratus.log 文件夹。

- 问题：如果群集中的任何服务器或节点停止工作并且不是 MySQL InnoDB 群集的一部分。

解决方法：在命令提示符下执行以下步骤：

```
var cluster = dba.rebootClusterFromCompleteOutage(); #Reboot the cluster instance
dba.configureLocalInstance('root@Server_IPAddress:3306') #Reconfigure the local instance
cluster.addInstance('root@Server_IPAddress:3306') #Add the cluster instance back to the network
```

- 问题：如果您不在 MySQL InnoDB 群集中添加服务器 ID，则会显示错误消息。

解决方法：在 \ProgramData\MySQL\MySQL Server 5.7 目录中 my.conf 文件中，更改服务器 ID 条目。

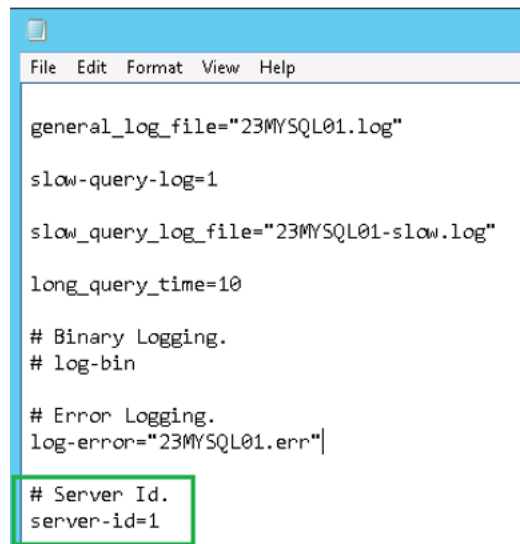


图 33: 更改服务器 ID