

Dell Wyse Management Suite

Version 1.3 High Availability Guide



Notes, cautions, and warnings

 | **NOTE:** A NOTE indicates important information that helps you make better use of your product.

 | **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 | **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

© 2018 Dell Inc. or its subsidiaries. All rights reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Contents

1 Introduction.....	4
High availability overview.....	4
2 High availability architecture.....	5
System requirements for high availability.....	5
3 High availability on Windows Server 2012.....	7
Creating clustered roles.....	7
4 Achieve high availability on Windows Server 2012.....	10
Add failover cluster feature on Windows Server 2012	10
Create file share witness.....	15
Configure cluster quorum settings.....	16
Creating clustered roles.....	18
5 Achieve high availability for MySQL InnoDB.....	21
High availability with MySQL InnoDB	21
Install MySQL InnoDB database.....	21
Check MySQL InnoDB server instances.....	21
Create a cluster instance for MySQL InnoDB.....	22
Add server instance to MySQL InnoDB cluster.....	23
Configure MySQL Router.....	24
Create database and users on MySQL InnoDB server.....	25
6 Achieve high availability on MongoDB.....	26
Install MongoDB	26
Create replica servers for MongoDB database.....	27
Create database user	27
Create DBadmIn user for MongoDB	28
Edit mongod.cfg file	28
Initiate replication on the servers.....	28
7 Achieve high availability for Teradici devices.....	32
Install and configure HAProxy.....	32
Install Wyse Management Suite server.....	34
8 Install Wyse Management Suite on Windows Server 2012.....	35
9 Post installation checks	36
10 Troubleshooting.....	38

Introduction

Wyse Management Suite version 1.3 is the next generation management solution and enables you to configure, monitor, manage, and optimize your Dell Wyse thin clients. This helps you to deploy and manage thin clients on a high availability set-up with improved performance.

It offers advanced feature options such as cloud versus on-premises deployment, manage-from-anywhere by using a mobile application, and enhanced security such as BIOS configuration and port lockdown. Other features include device discovery and registration, asset and inventory management, configuration management, operating system and applications deployment, real-time commands, monitoring, alerts, reports, and troubleshooting of endpoints.

Wyse Management Suite version 1.3 supports high availability and significantly minimizes the system downtime. The solution also protects the system from unplanned downtime and helps you to achieve the required availability to meet the business goals.

This guide describes the solution architecture and explains how to set up, configure, and maintain high availability clusters at the application and database level.

High availability overview

The high availability solution for Wyse Management Suite version 1.3 includes the following tasks:

- 1 Review the high availability requirements—see [System requirements to set up high availability](#).
- 2 Deploy high availability on Microsoft Windows Server 2012—see [Deploy high availability on Windows Server 2012](#).
- 3 Deploy high availability on MySQL InnoDB servers—see [Deploy high availability on MySQL InnoDB](#).
- 4 Deploy high availability on MongoDB—see [Deploy high availability on MongoDB](#).
- 5 Configure high availability proxy (for Teradici devices)—see [Deploy high availability for Teradici servers](#).
- 6 Install Wyse Management version on Windows Server 2012—see [Install Wyse Management Suite on Windows Server 2012](#).
- 7 Review the post installation checks—see [Post installation checks](#).
- 8 Troubleshooting issues with workaround—see [Troubleshooting](#).

High availability architecture

The Dell Wyse Management Suite architecture consists of Windows Server 2012 with failover cluster enabled. The Windows cluster contains a main computer that supports other applications and ensures minimum downtime by harnessing the redundant. This is used for application failover for Tomcat, Memcache, MQTT services. MongoDB database cluster helps in the event of primary database failure the secondary database will take over. MySQL InnoDB database cluster has an inbuilt database clustering mechanism and secondary database will take over in case of primary read write database fail. Linux Server with HA Proxy is a load balancer and high availability server for EMSDK (Teradici) server. Local repository is created as part of the shared path that contains the applications, images, packages, and will not be part of the cluster set up.

NOTE: The high availability system requirements may change depending on the infrastructure at your work site.

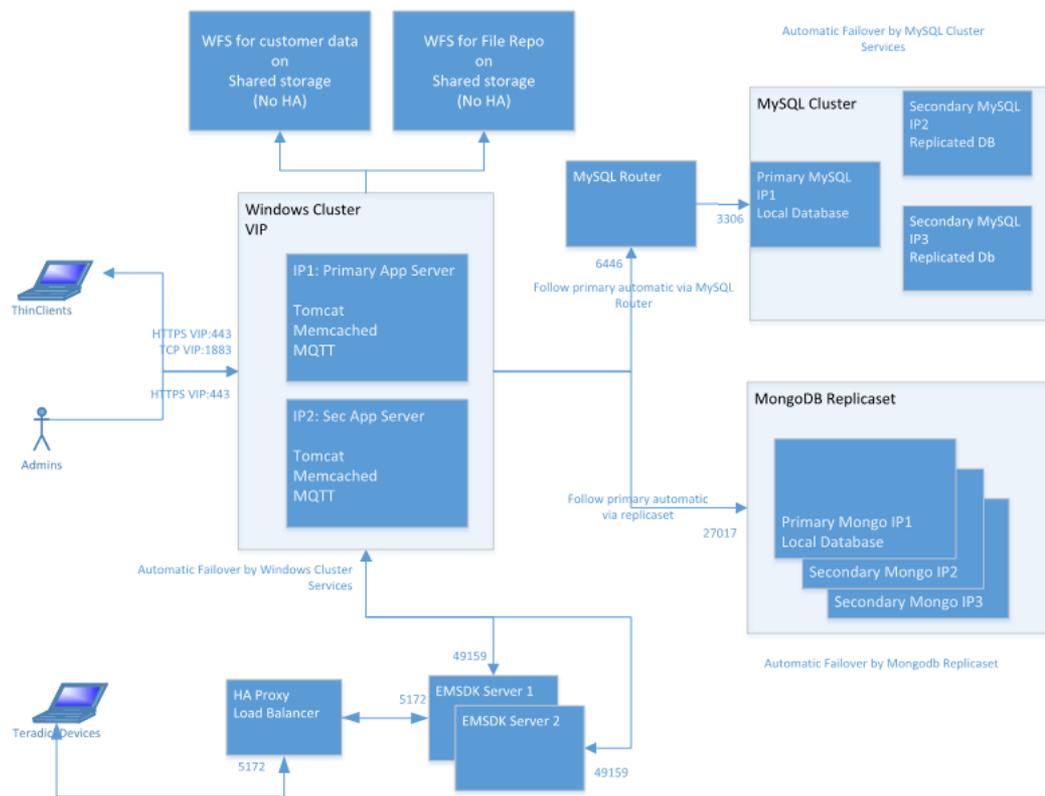


Figure 1. High availability architecture

System requirements for high availability

The table lists the minimum hardware and software requirement and supports up to 10,000 devices. Each instance of EMSDK can support a maximum of 5000 devices. The deployment can be on individual servers or on a hypervisor environment, depending on the requirement.

The hardware and software requirements to set up high availability for Wyse Management Suite version 1.3 are:

Table 1. System requirements

Product	Port	Protocol	Description
Microsoft Windows Server 2012 R2	<ul style="list-style-type: none"> • Network communication ports: <ul style="list-style-type: none"> – UDP:3343 – TCP:3342 – UDP:137 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—2 • Minimum memory (RAM)—8 GB • Minimum CPU requirements—4 	<p>Server where Wyse Management Suite is hosted.</p> <p>Supports English, French, Italian, German, and Spanish languages.</p>
MySQL Cluster	<ul style="list-style-type: none"> • Network communication port—TCP:3306 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—3 • Minimum memory (RAM)—8 GB • Minimum CPU requirements—4 	Server in the high availability setup.
MySQL Router	<ul style="list-style-type: none"> • Network communication ports: <ul style="list-style-type: none"> – 6446 – 6447 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—2 • Minimum memory (RAM)—8 GB • Minimum CPU requirements—4 	Establishes communication in the high availability setup.
MongoDB	<ul style="list-style-type: none"> • Network communication port—TCP: 27017 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—3 • Minimum memory (RAM)—8 GB • Minimum CPU requirements—4 	Database
EMSDK	<ul style="list-style-type: none"> • Network communication port—TCP: 5172 • TCP 49159 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—2 • Minimum memory (RAM)—8 GB • Minimum CPU requirements—4 	Enterprise SDK server
HAProxy	<ul style="list-style-type: none"> • Network communication port—TCP: 5172 	<ul style="list-style-type: none"> • Minimum disk space—40 GB • Minimum number of systems—1 • Minimum memory (RAM)—4 GB • Minimum CPU requirements—2 	<p>Load balancer in the high availability setup.</p> <p>Ubuntu version 12.04 and later.</p>

NOTE:

Ensure that you add the TCP ports 443, 8080 and 1883 to the firewall exception list during high availability setup.

High availability on Windows Server 2012

A failover cluster is a group of independent systems that increases the availability and scalability of clustered roles. This feature supports multiple workloads running clusters on hardware or on virtual machines.

A failover cluster is a group of systems that are independent and increases the availability and scalability of clustered roles. The clustered servers are the nodes that are connected to one another as a network. If one or more of the cluster nodes fail, other nodes become active and prevents failover of the systems in the network. The clustered roles that are created during cluster setup monitor to verify that the systems are working in the clustered network. If any of the systems are not working, they are restarted or moved to another node.

The failover cluster network for high availability on Windows Server 2012 contains two nodes, Node 1 and Node 2 that are configured on systems running Windows Server 2012. In the failover cluster network, if Node 1 that is working as the primary node fails, Node 2 starts working automatically as the primary node. After Node 1 becomes active, it automatically becomes the secondary node. The systems have a shared storage space that is connected in a network.

NOTE: The IP address of the systems in the image is an example and varies for each system at your work place.

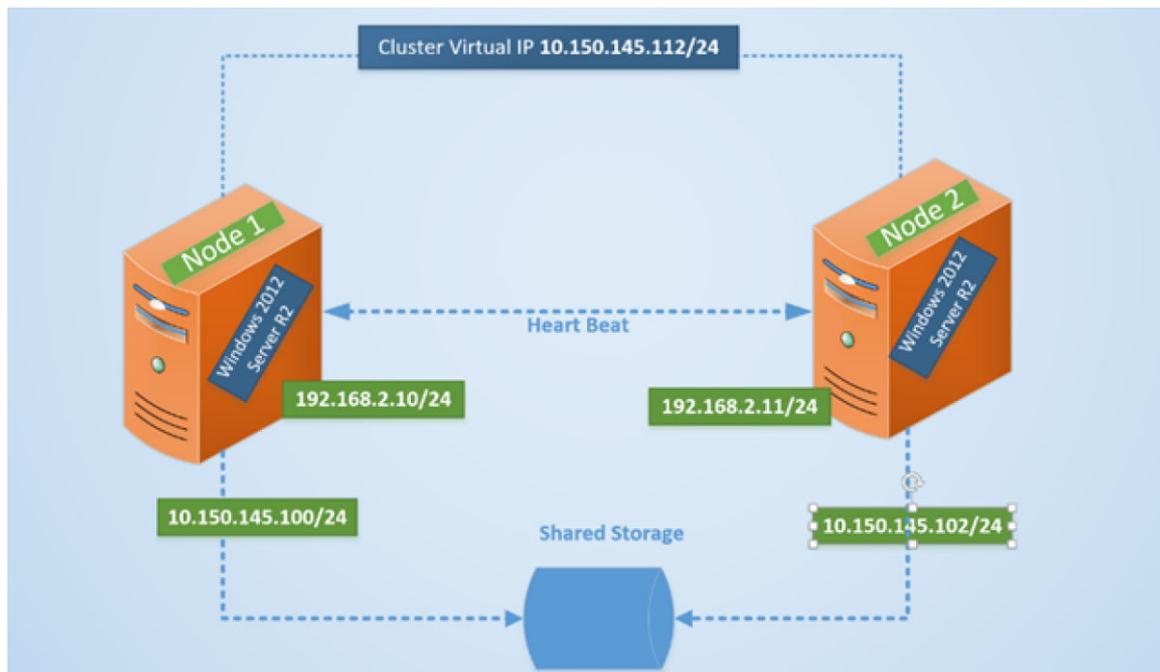


Figure 2. Failover cluster setup

Creating clustered roles

After you create the failover cluster, you can create clustered roles to host cluster workloads. Ensure that Wyse Management Suite is installed on the servers and point to the remote database before you create clustered roles.

To create a clustered role, do the following:

- 1 In Microsoft Windows Server 2012, right-click the **Start** menu and then select **Server Manager** to launch the Server Manager dashboard
- 2 Click **Failover Cluster Manager** to launch the cluster manager.

- 3 Right-click **Roles** and then select **Configure Role** to display the **High Availability Wizard** screen.

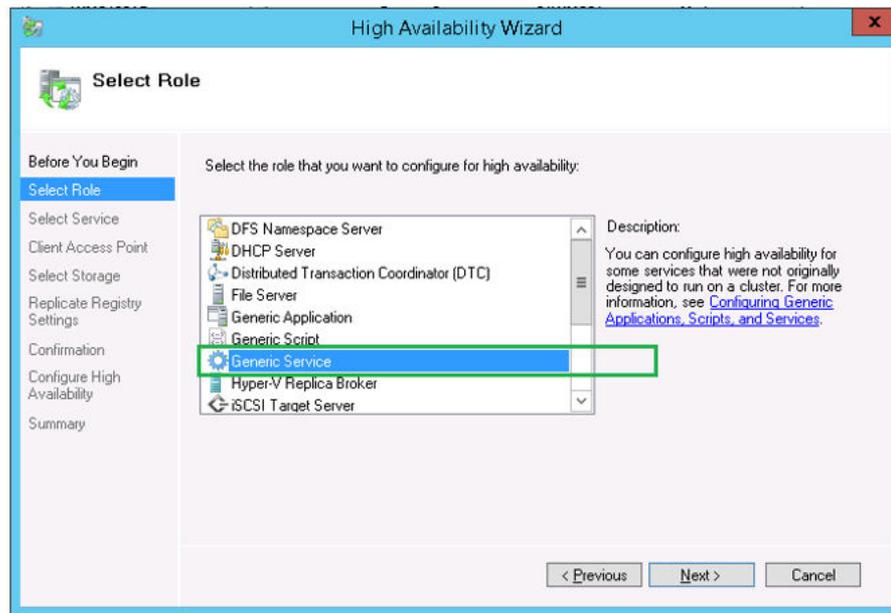


Figure 3. High availability wizard

- 4 Select **Generic Service** and then click **Next** to view the **Select Service** screen.

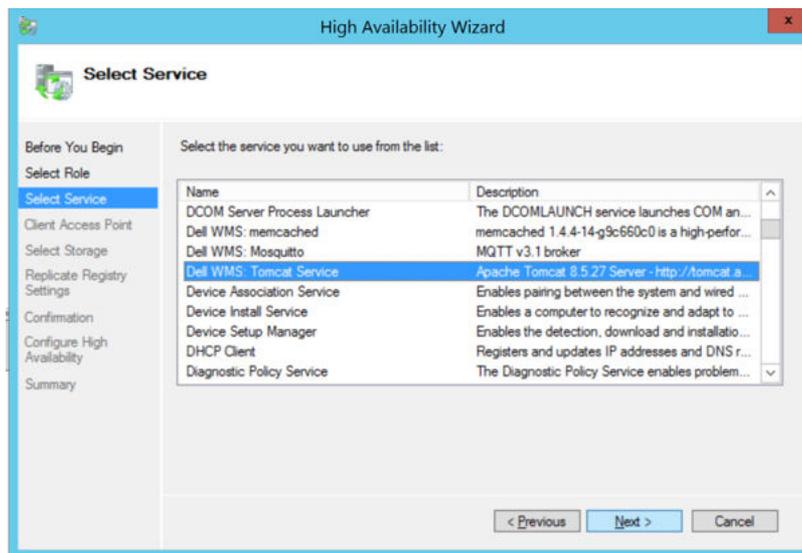


Figure 4. Select service

- 5 Select **Dell WMS: Tomcat Service** and then click **Next**.

NOTE: You can add the Wyse Management Suite version 1.3 related services to the cluster only after you install Wyse Management Suite version 1.3.

The **High Availability Wizard** screen is displayed where you need to create the client access point and establish connectivity between the Windows server 2012 and Wyse Management Suite.

- 6 Type a network name in the **Name** field and then click **Next**. The **Confirmation** screen is displayed with the network name and IP address details of the server.

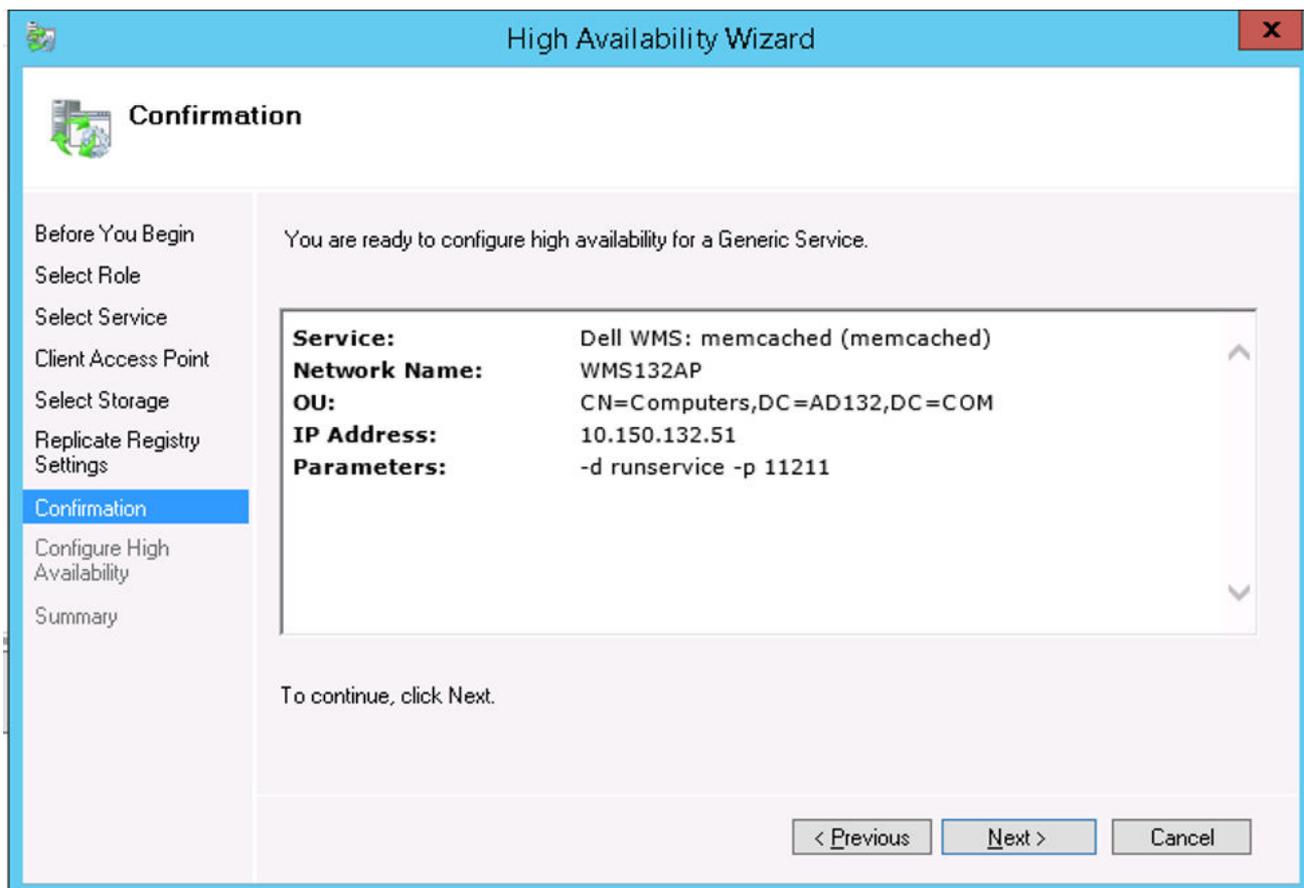


Figure 5. Confirmation

- 7 Click **Next** to complete the process.
- 8 To add other Wyse Management Suite services as part of the cluster, launch **Failover Cluster Manager**, and then go to **ActionsRoles** to display the network name that you have created.
- 9 Click on the network name, and go to **Add ResourceGeneric Service**.
- 10 Select the following services from the **New Resource Wizard** screen that needs to be added as part of the cluster:
 - a Dell WMS: Mosquitto >> MQTT Broker
 - b Dell WMS: memcached
- 11 Click **Next** to complete the task.
The Wyse Management Suite services that have been added as part of the cluster are displayed with the status **Running**.

Achieve high availability on Windows Server 2012

The following are the steps to achieve high availability on Windows Server 2012:

- 1 Add failover cluster feature on Windows Server 2012—see [Adding failover cluster feature on Windows Server 2012](#).
- 2 Create file share witness—see [Create file share witness](#).
- 3 Configure cluster Quorum—see [Configure cluster Quorum](#).
- 4 Create clustered roles—see [Create cluster roles](#).

Add failover cluster feature on Windows Server 2012

To add the failover clustering feature on the Windows server 2012, do the following:

- 1 In Microsoft Windows Server 2012, click **Start** to open the **Start** screen and then click **Server Manager** to launch the **Server Manager** dashboard.

NOTE: Server Manager is a management console in Windows Server 2012 that enables you to add server roles/features, manage, and deploy servers.

- 2 Click **Add roles and features** and select an option to configure the server based on your requirement from the **Add Roles and Feature Wizard** screen.

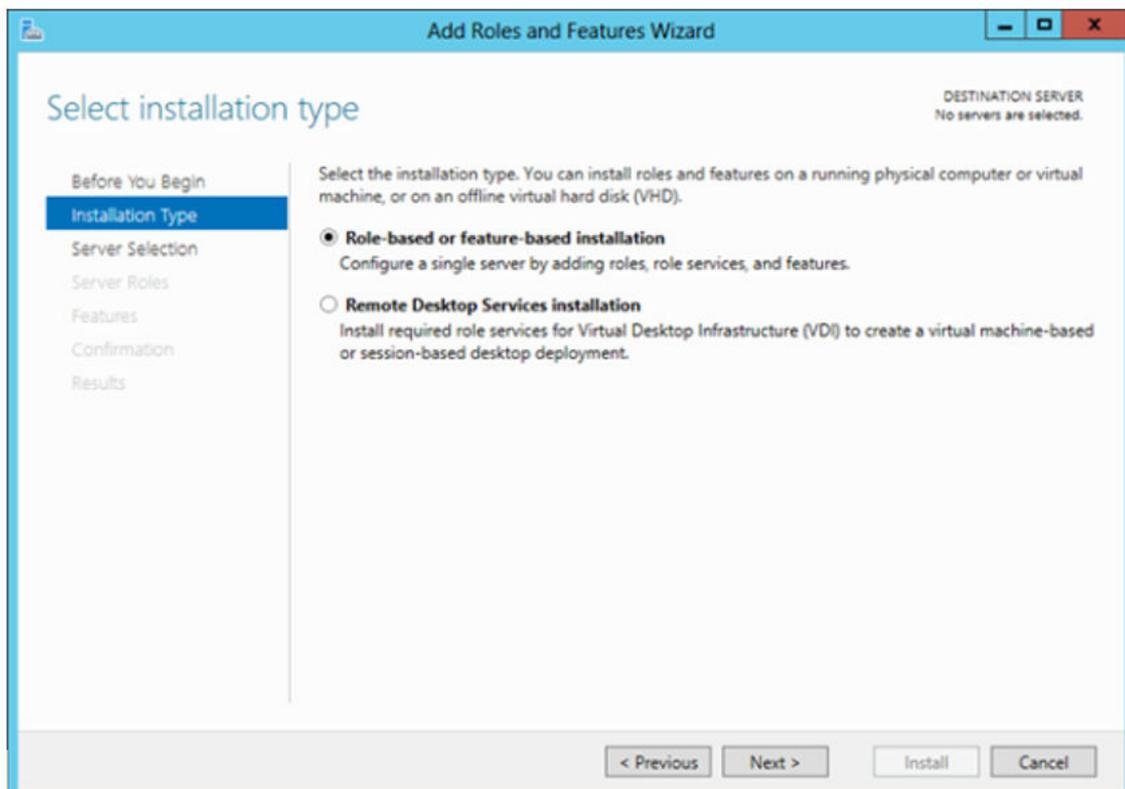


Figure 6. Role based selection

- 3 Click **Installation Type** and select **Role-based or Feature-based installation** and then click **Next** to view the list of servers in the **Select destination server** screen.

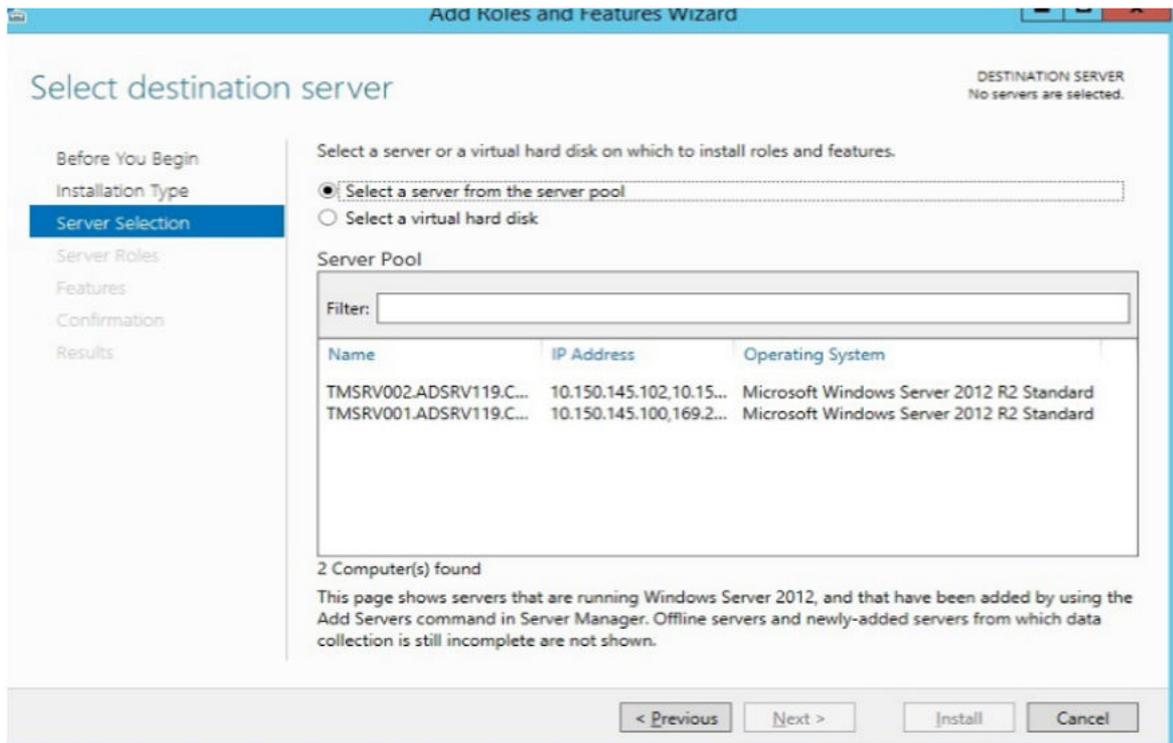


Figure 7. Select server destination

- 4 Select the server where you want to enable the failover cluster feature and then click **Next**.
- 5 Select **Failover Clustering** on the **Features** screen, and then click **Next**. After you enable the failover cluster on the servers, open the **Failover Cluster Manager** on the server at Node 1.
- 6 Click **Yes** to confirm installation, and enable the failover cluster feature on the selected server.
- 7 In the **Failover Cluster Manager** screen, click **Validate Configuration** to view the **Validate a Configuration Wizard** add the required servers or nodes to cluster.

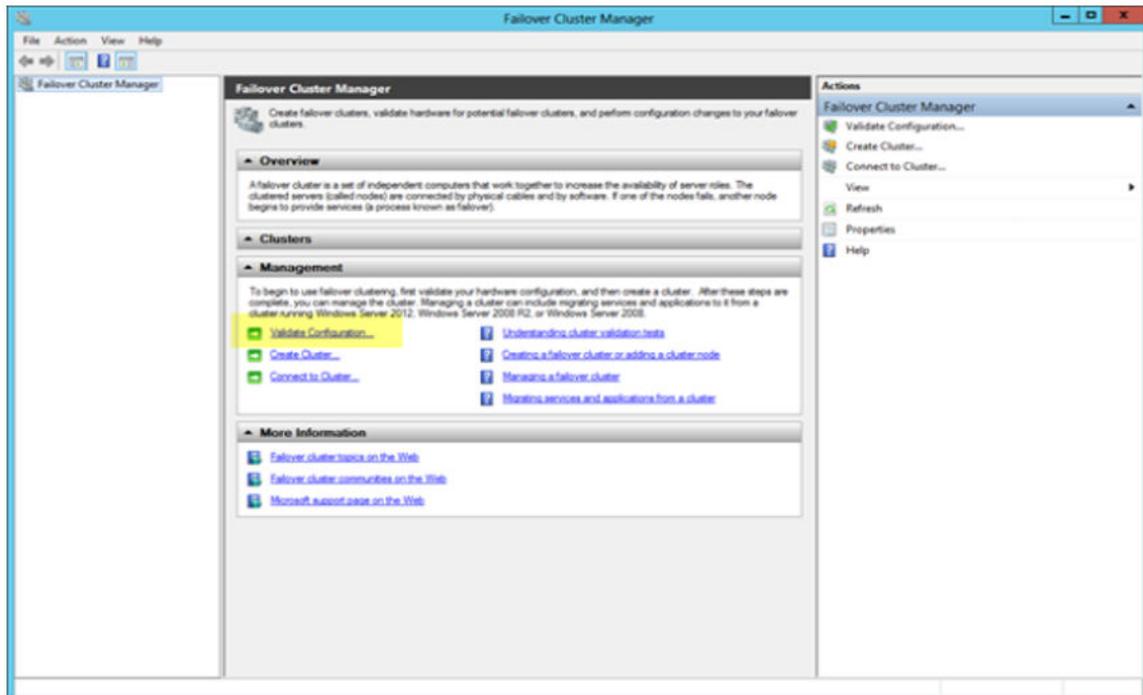


Figure 8. Failover cluster manager

- 8 Click **Select servers or cluster** and then click **Browse** to configure the servers.
- 9 Click **Next** and select **Run all tests** from the **Testing Options** screen.

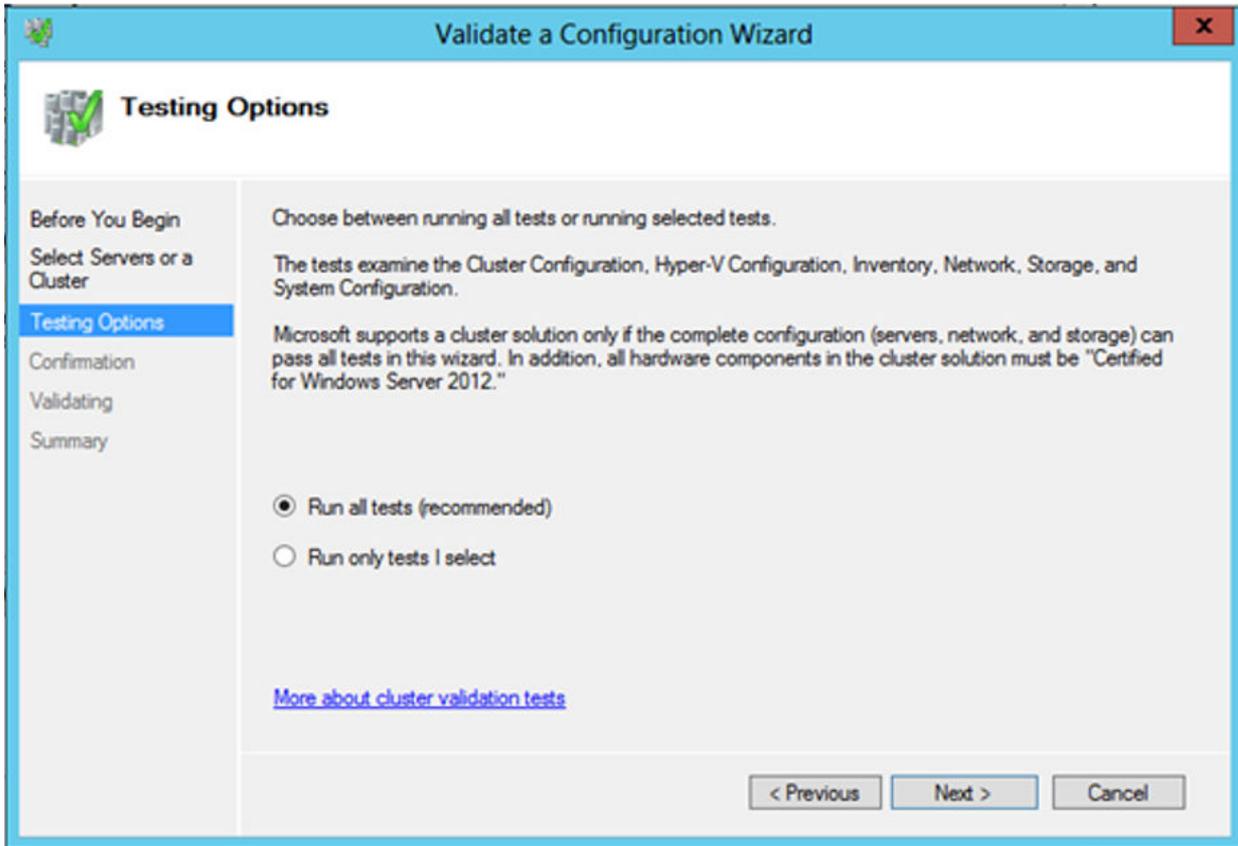


Figure 9. Testing options

- 10 Click **Next**. The **Confirmation** screen is displayed with the list of selected servers.

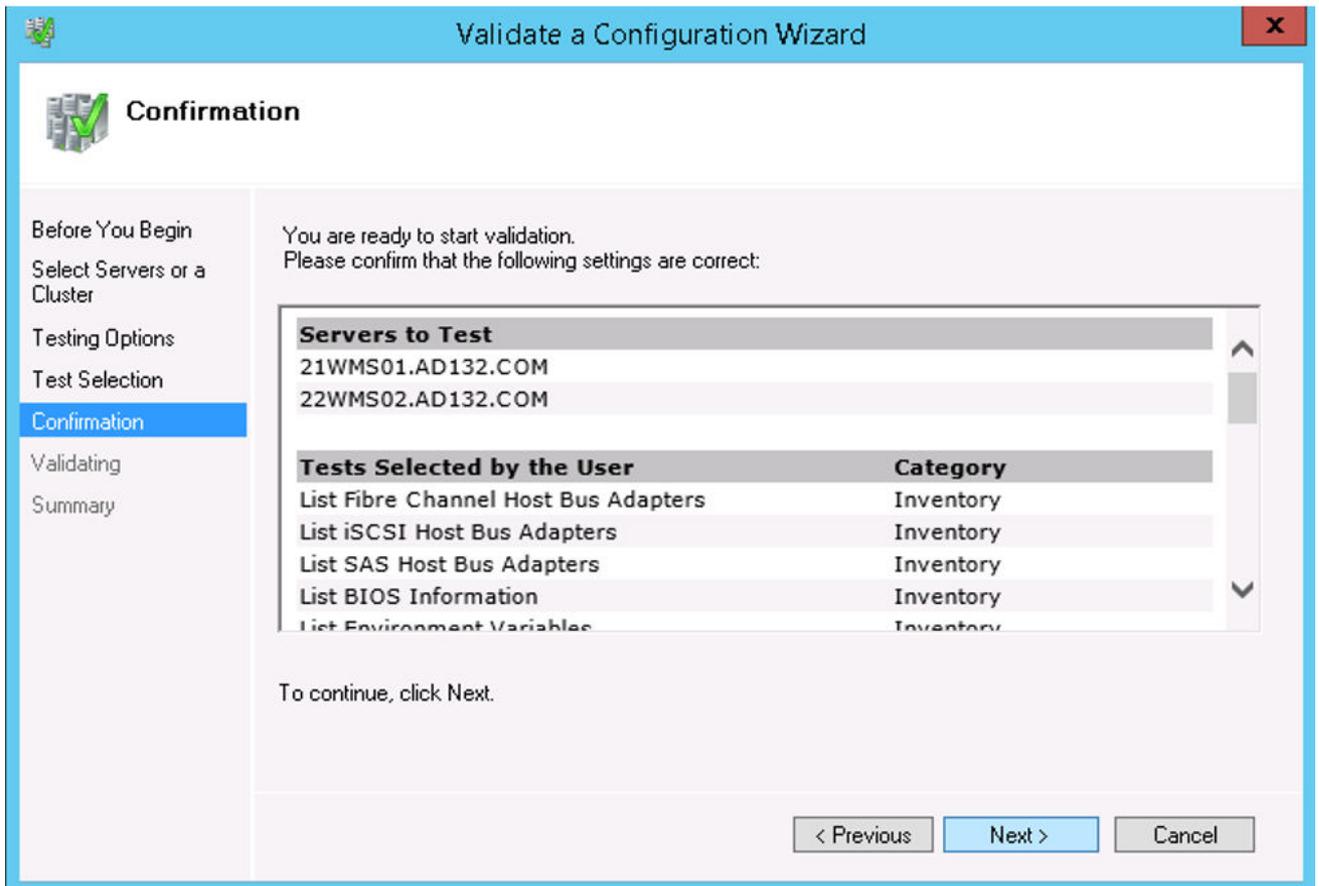


Figure 10. Confirmation

- 11 Click **Next**. The **Summary** screen is displayed with the failover cluster validation report.

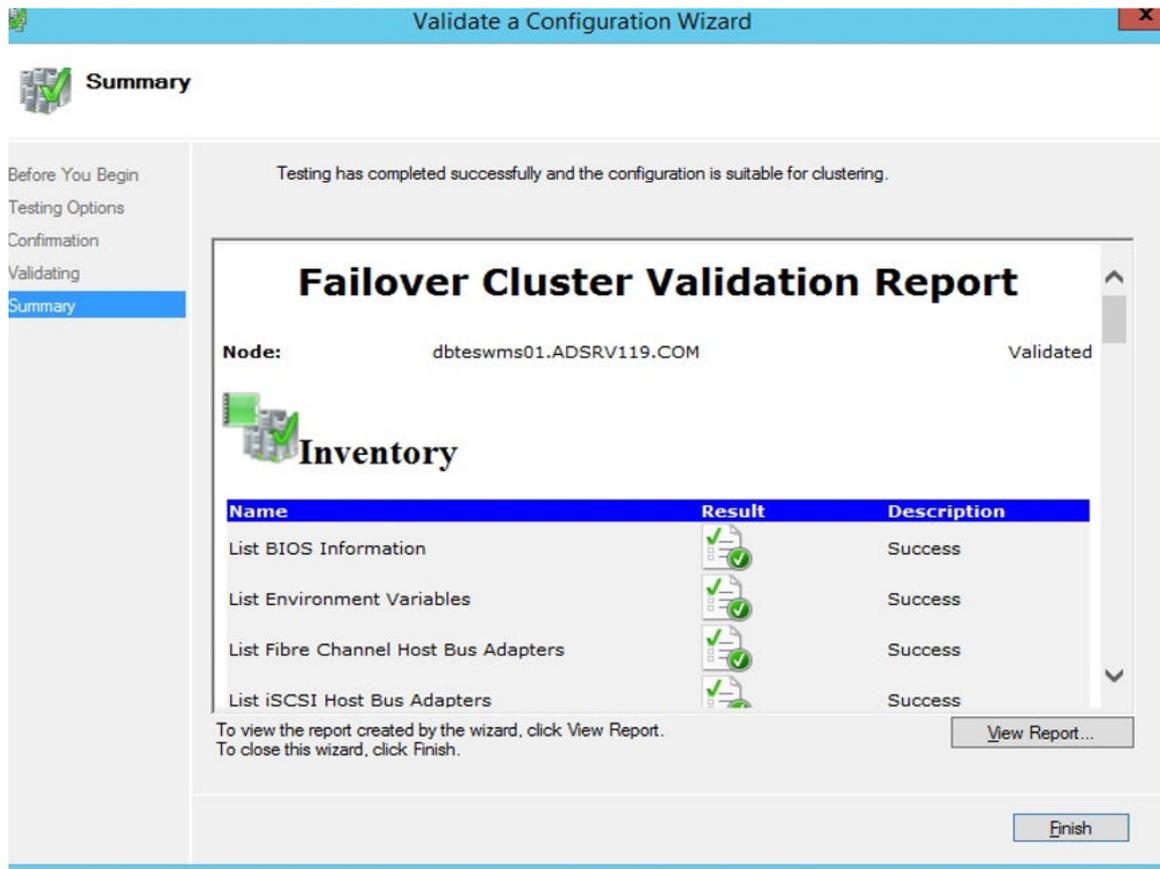


Figure 11. Test summary details

- 12 Click **View Report** to check the report. If the status is **Passed**, you can proceed with the next step. If the status is **Failed**, you must fix the errors before you proceed with the next step.

NOTE: The Create Cluster Wizard screen is displayed if there are no validation errors.

- 13 Click **Next** and type a name for the cluster in the **Cluster Name** field and then select the IP address of the system.
- 14 Click **Next** and the **Confirmation** screen is displayed.
- 15 Click **Next** to create the cluster on all the selected clustered nodes and then click **View Report** to view the warning messages.
- 16 Click **Finish** to create the failover cluster.

Create file share witness

A file share witness is a basic file share that the cluster computer has read/write access. The file share must be on a separate Windows Server 2012 in the same domain where the cluster resides.

To create a file share witness, do the following:

- 1 In Microsoft Windows Server 2012, Right-click the **Start** Menu and then select **Server Manager** to launch the Server Manager dashboard
- 2 Click the **Server Manager** icon to access the server manager.
- 3 Go to **Files and Storage ServicesShares** and then click **Tasks**.
- 4 Click **New Share**. The **New Share Wizard** is displayed.
- 5 Click **Select Profile** to create a file share and then click **Next**.
- 6 On the **Share location** screen, select the server and share location for the file share and then click **Next**.
- 7 On the **Share Name** screen, type a name in the **Share name** field and then click **Next** until the **Confirmation** screen is displayed.
- 8 Click **Create** to create the file share and the **View results** screen is displayed with the status as **Completed** which indicates that the file share witness is created without any errors.

- 9 Click **Close** to exit.

Configure cluster quorum settings

The cluster configuration database, also called the quorum, contains details as to which server should be active at any given time in a cluster set-up.

To configure the cluster quorum settings, do the following:

- 1 In Microsoft Windows Server 2012, click **Start** to open the **Start** screen and then click **Server Manager** to launch the Server Manager dashboard.
- 2 Click the **Server Manager** icon to access the server manager and then click **Failover Cluster Manager** to launch the cluster manager.
- 3 Right-click the cluster node, and go to **More Actions** **Configure Cluster Quorum Settings** to display the **Configure Cluster Quorum Wizard** screen.
- 4 Click **Next**. Select **Select the quorum witness** from the **Select Quorum Configuration Option** screen.

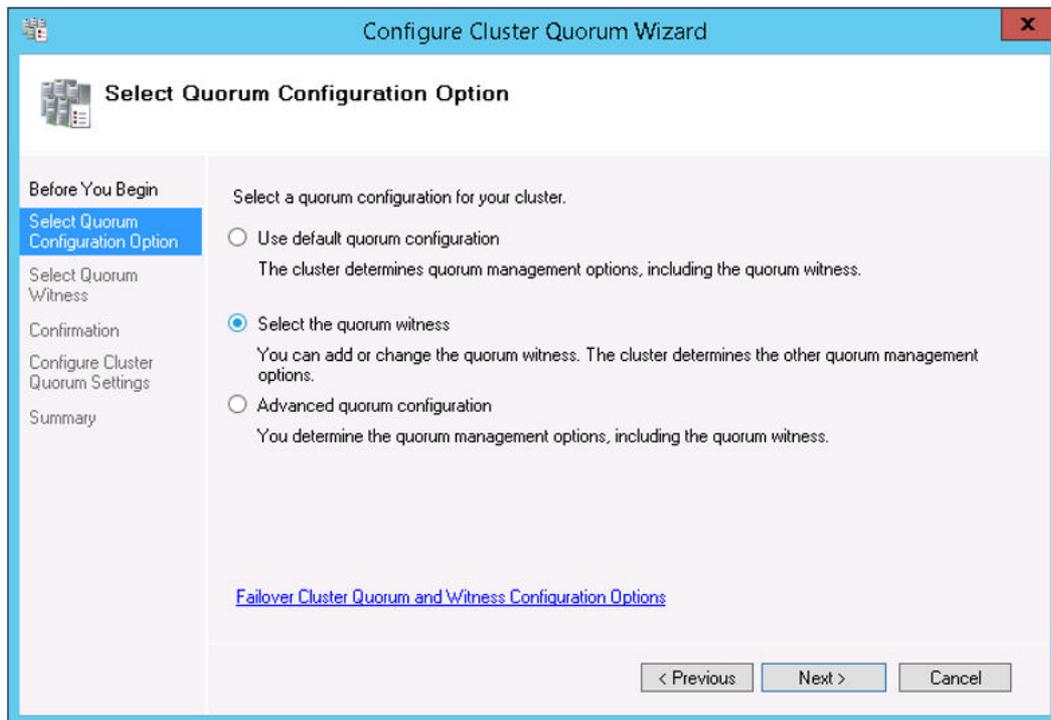


Figure 12. Quorum cluster wizard

- 5 Click **Next**. Select **All Nodes** from the **Select Voting Configuration** screen.

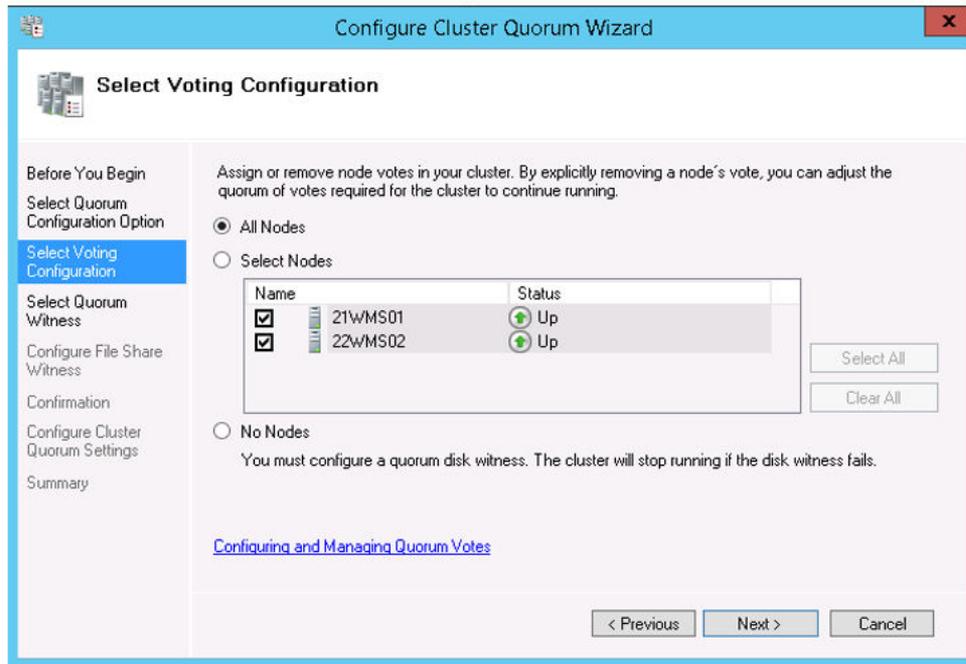


Figure 13. Select voting configuration

- 6 Click **Next** . Select **Configure a file share witness** from the **Select Quorum Witness** screen.
- 7 Click **Next** and then type the share path in the **File Share Path** field from the **Configure a file share witness** screen.

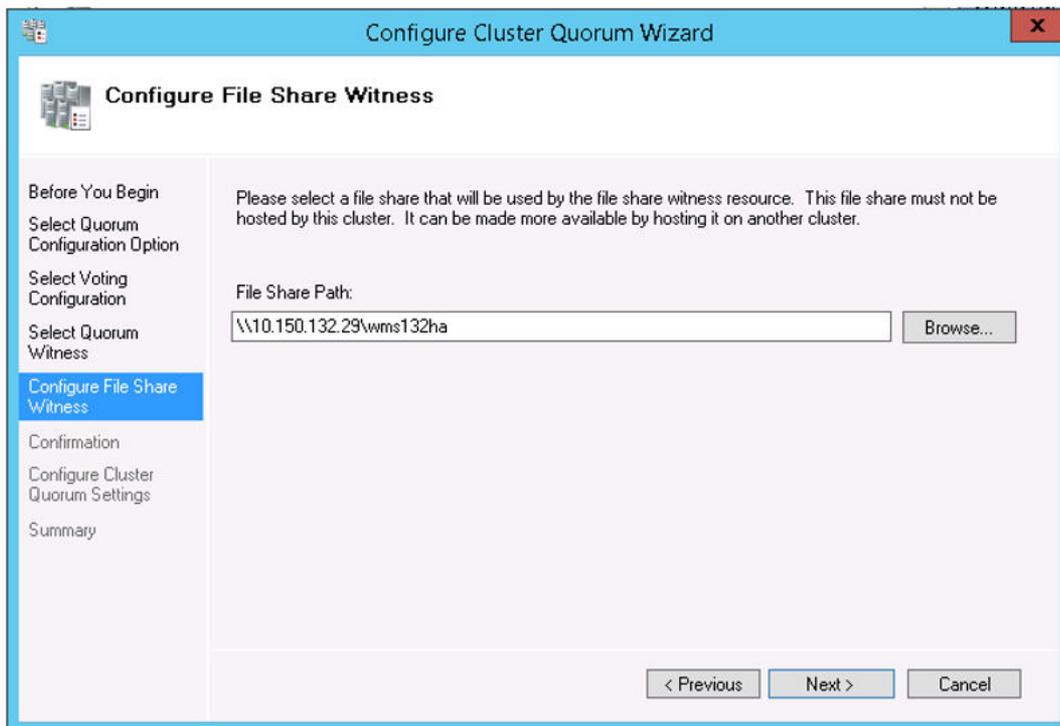


Figure 14. Configure file share witness

- 8 Click **Next** . The **Summary** screen is displayed with the configured quorum settings.

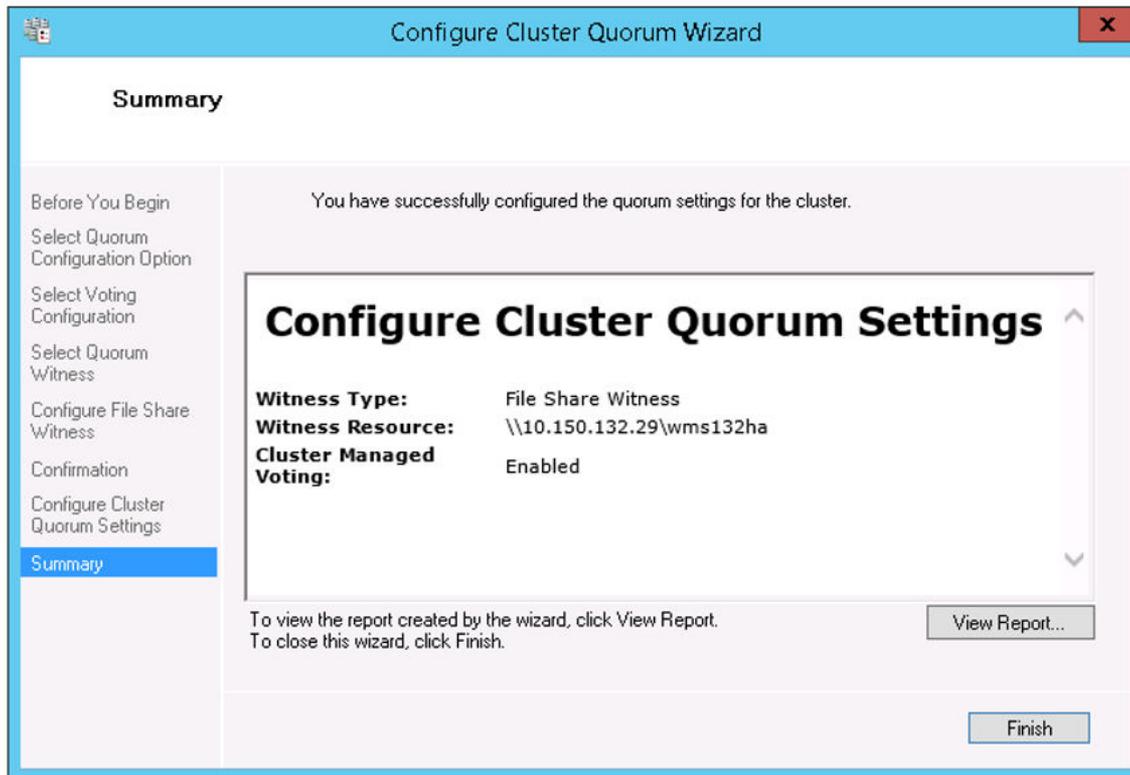


Figure 15. Summary of the quorum settings

- 9 Click **Finish** to complete the quorum settings.

Creating clustered roles

After you create the failover cluster, you can create clustered roles to host cluster workloads. Ensure that Wyse Management Suite is installed on the servers and point to the remote database before you create clustered roles.

To create a clustered role, do the following:

- 1 In Microsoft Windows Server 2012, right-click the **Start** menu and then select **Server Manager** to launch the Server Manager dashboard
- 2 Click **Failover Cluster Manager** to launch the cluster manager.
- 3 Right-click **Roles** and then select **Configure Role** to display the **High Availability Wizard** screen.

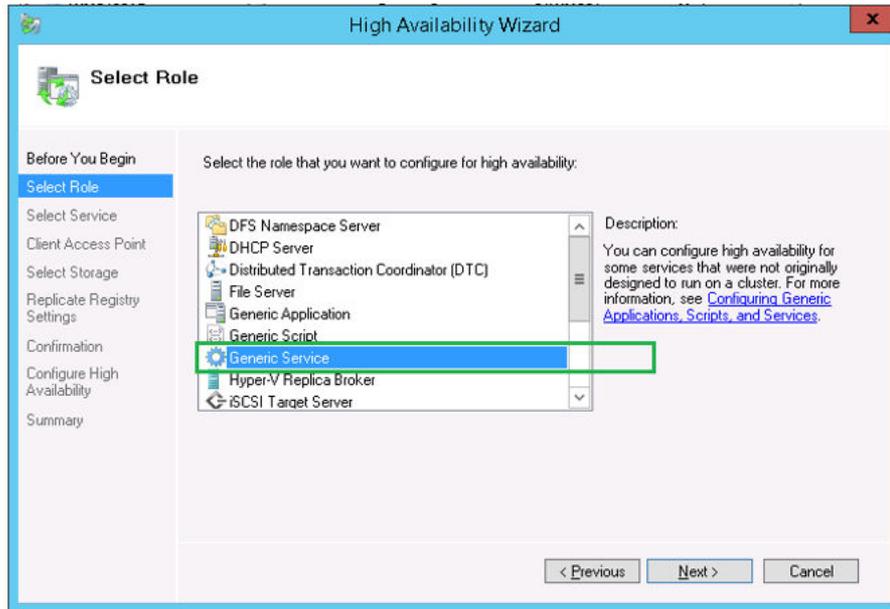


Figure 16. High availability wizard

- 4 Select **Generic Service** and then click **Next** to view the **Select Service** screen.

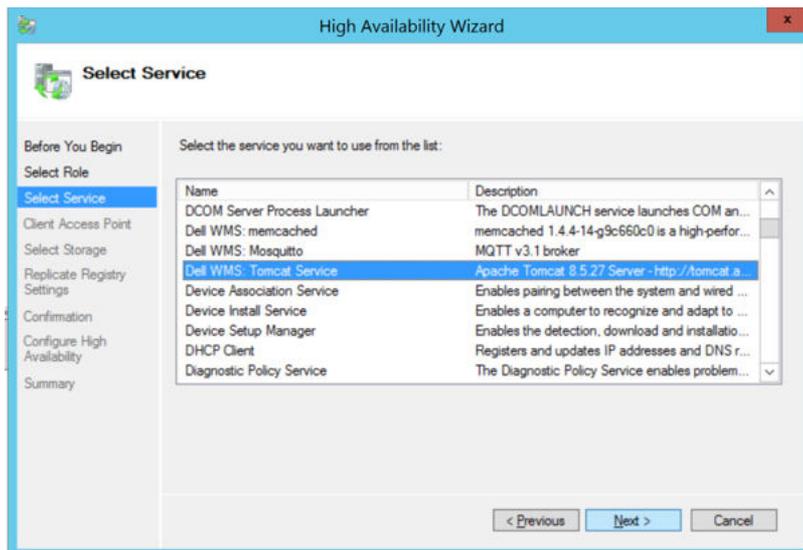


Figure 17. Select service

- 5 Select **Dell WMS: Tomcat Service** and then click **Next**.

NOTE: You can add the Wyse Management Suite version 1.3 related services to the cluster only after you install Wyse Management Suite version 1.3.

The **High Availability Wizard** screen is displayed where you need to create the client access point and establish connectivity between the Windows server 2012 and Wyse Management Suite.

- 6 Type a network name in the **Name** field and then click **Next**. The **Confirmation** screen is displayed with the network name and IP address details of the server.

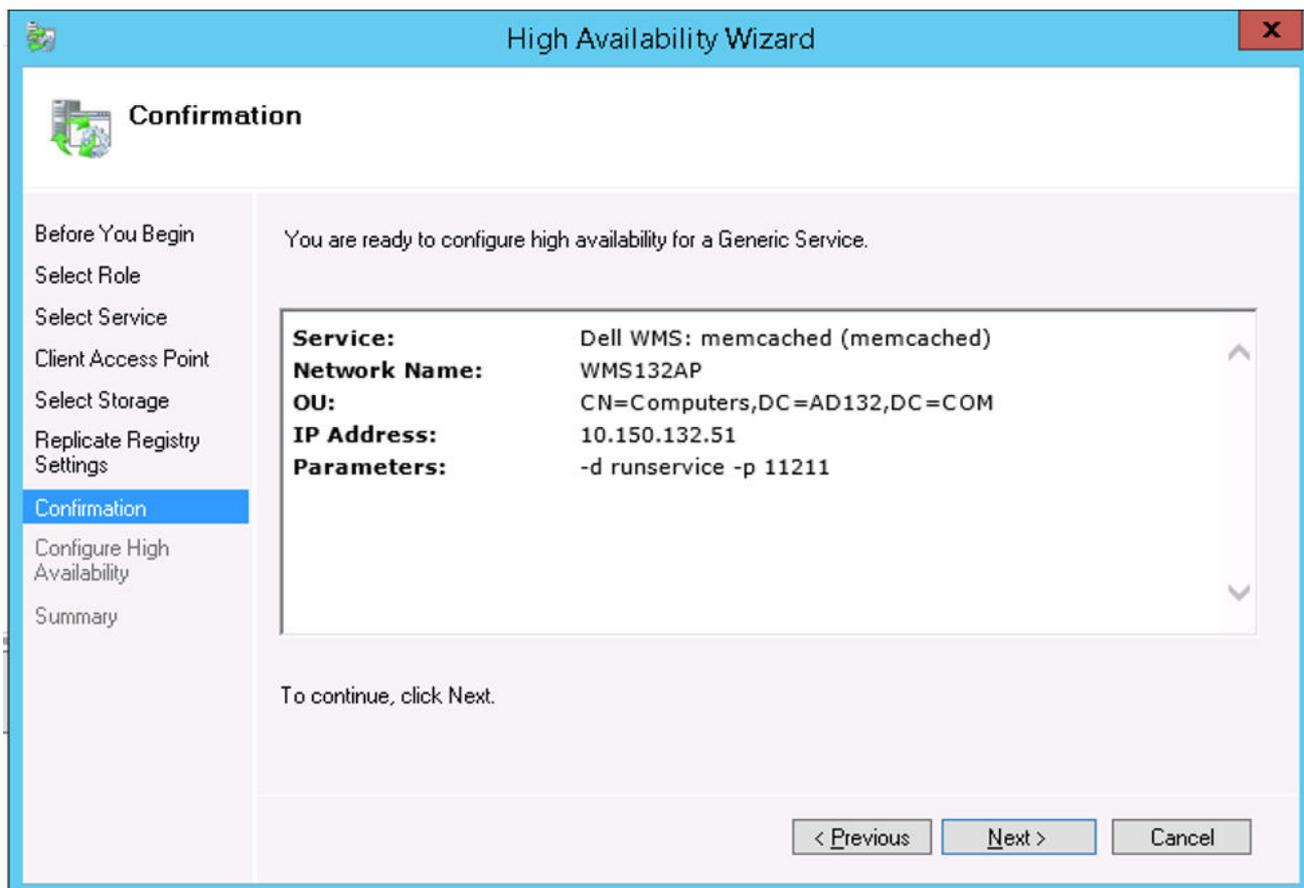


Figure 18. Confirmation

- 7 Click **Next** to complete the process.
- 8 To add other Wyse Management Suite services as part of the cluster, launch **Failover Cluster Manager**, and then go to **ActionsRoles** to display the network name that you have created.
- 9 Click on the network name, and go to **Add ResourceGeneric Service**.
- 10 Select the following services from the **New Resource Wizard** screen that needs to be added as part of the cluster:
 - a Dell WMS: Mosquitto >> MQTT Broker
 - b Dell WMS: memcached
- 11 Click **Next** to complete the task.
The Wyse Management Suite services that have been added as part of the cluster are displayed with the status **Running**.

Achieve high availability for MySQL InnoDB

The following steps explain how to achieve high availability for MySQL InnoDB:

- 1 Check MySQL InnoDB server instance—see [Create MySQL InnoDB cluster](#).
- 2 Add server or node to MySQL InnoDB—see [Adding server or node to MySQL InnoDB cluster](#).
- 3 Create MySQL Router—see [Creating MySQL Router](#)

High availability with MySQL InnoDB

The MySQL InnoDB cluster provides a complete high availability solution for MySQL. The client application is connected to the primary node by using the MySQL router. If the primary node fails, a secondary node is automatically promoted to the role of primary node, and the MySQL router routes the requests to the new primary node.

The components of the MySQL InnoDB cluster are:

- MySQL server
- MySQL router

Install MySQL InnoDB database

To install MySQL InnoDB, see dev.mysql.com.

To set up the environment as per the high availability setup, see dev.mysql.com.

Check MySQL InnoDB server instances

Before you add MySQL InnoDB to the cluster setup, verify that MySQL InnoDB is created as per the cluster requirements.

You must login as **root** user to run the commands and restart the system each time you run a set of commands.

Run the following commands to verify that the MySQL InnoDB server instance meets the configured cluster requirements:

NOTE: The IP Address is different for each system that is used at your work place and the following commands are used only as an example.

- 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')`

```

C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
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_checksun         | CRC32         | NONE           | Update the server variable             |
| enforce_gtid_consistency | OFF           | ON             | Update read-only variable and restart the server |
| gtid_node                | 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> _

```

Figure 19. MySQL command prompt

To check that the MySQL InnoDB is created on all the three cluster nodes, run the following commands at the command prompt:

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

Create a cluster instance for MySQL InnoDB

After you have installed MySQL InnoDB instance on the servers, create a cluster instance.

To create a cluster for MySQL InnoDB, do the following:

- 1 Login as administrator user from the command prompt. This user account should have administrative privileges. For example, **DBadmin**. The following screen shows an example of logging in as root user.

```

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 ?
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> _

```

Figure 20. Login prompt

- 2 Run the following command to create a cluster with a unique name. For example, **MySQLCluster**.

```
MySql JS> var cluster = dba.createCluster('MySQLCluster')
```

- 3 Run the following command to check the status of the cluster.

```
MySql JS> Cluster.status()
```

The status of the created cluster is displayed as **ONLINE** which indicates that the cluster is created successfully.

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

Figure 21. Confirmation screen

Add server instance to MySQL InnoDB cluster

You must add server instance to the MySQL InnoDB cluster as primary or secondary.

Do the following to add a server instance to the MySQL InnoDB cluster:

- 1 Login as **DB Admin** user from the command prompt.
- 2 Run the following command to add a server instance to the MySQL InnoDB cluster:

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

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

NOTE: The IP address and the port numbers are only examples and will vary based on the system that you will be using at your work place.

- 3 Run the following command to check the status of the server instance:

```
cluster.status()
```

NOTE: All the nodes should display the status as **ONLINE** which indicates that the nodes have been added successfully to the MySQL InnoDB cluster set up.

```
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] >
```

Figure 22. Cluster status

Configure MySQL Router

MySQL Router establishes communication network between Wyse Management Suite and MySQL InnoDB.

To install MySQL Router, do the following:

- 1 Login to the Windows Server 2012 to install MySQL Router. For more information, see [MySQL Router Installation](#)
- 2 Select **MySQL Router** from the **Select Products and Features** screen and then click **Next** until the **Installation Complete** screen is displayed.
- 3 Browse to `\ProgramData\MySQL\MySQL Router` directory, and open the file `mysqlrouter.conf` to check that the bootstrap property with all the configured MySQL servers are part of cluster setup.

```

mysqlrouter - Notepad
File Edit Format View Help
# 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_oqj610zm2w3p
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

```

Figure 23. Bootstrap server address

Create database and users on MySQL InnoDB server

You must create the database and user accounts with administrator privileges on MySQL InnoDB server.

To create database on MySQL InnoDB server, run the following SQL commands:

```

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;

```

NOTE: Instead of IP Address, you can type the Wildcard for Network /Subnet or Multiple Single host entry where Wyse Management Suite application server will be installed.

Achieve high availability on MongoDB

The following steps explain how to achieve high availability on MongoDB:

- 1 Install MongoDB—see [Installing MongoDB](#).
- 2 Create replica servers—see [Creating Replica servers](#).
- 3 Create Stratus users—see [Creating Stratus user account](#).
- 4 Create root user—see [Creating root user for MongoDB](#).
- 5 Edit MongoDB configuration file—see [Editing MongoDB configuration file](#).

Install MongoDB

To install MongoDB on all the three nodes, do the following:

NOTE: For information on installing MongoDB see—[Install MongoDB](#)

- 1 Copy the MongoDB installation files on a system.
- 2 Create two folders `Data\log` and `data\db` on a secondary drive other than Drive C.

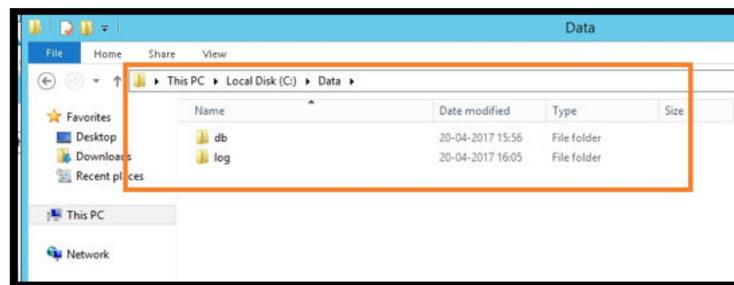


Figure 24. Data files

- 3 Go to the folder where you have copied the MongoDB installation files, and create a file `mongod.cfg` from the command prompt.

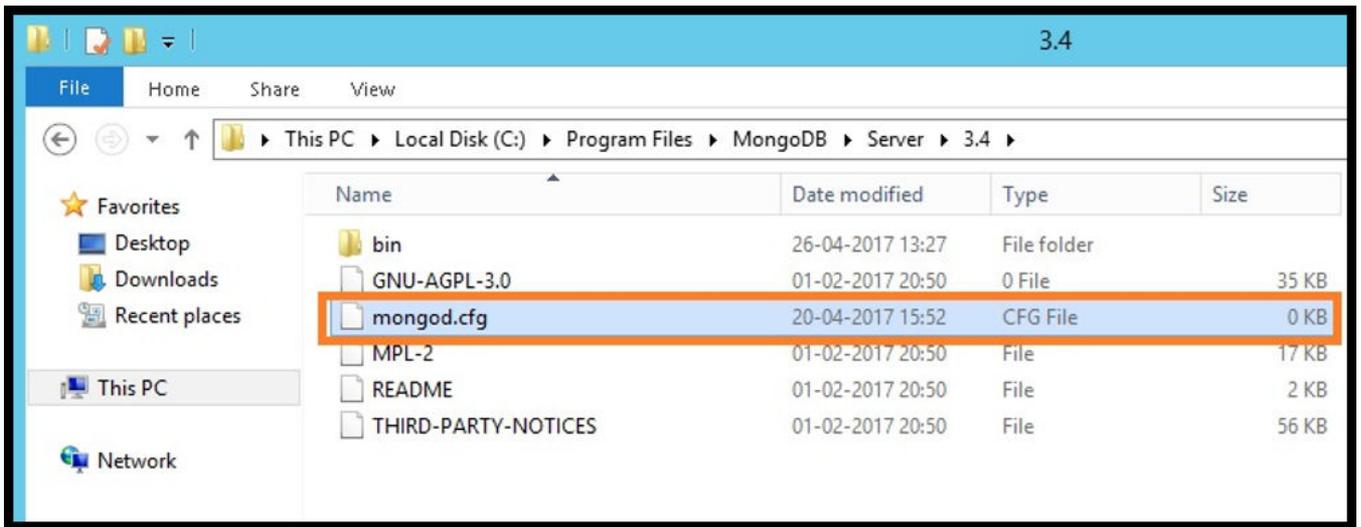


Figure 25. mongod.cfg file

- 4 Open the `mongod.cfg` file in a text editor, and add the following entries:
 - a `SystemLog:destination: file`
 - b `path: c:\data\log\mongod.log`
 - c `Storage: dbpath: c:\data\db`
- 5 Save the file.
- 6 Login to MongoDB server.
- 7 Run the following command to start the MongoDB service:
 - a `C:\MongoDB\bin>.\mongod.exe --config c:\Mongodb\mongod.cfg --install`
 - b `C:\MongoDB\bin>net start mongodb`

The message **MongoDB service is starting** is displayed.

- 8 Change the working directory to `\MongoDB\bin`.
- 9 Run `Mongo . exe` at the command prompt to complete the MongoDB installation.

Create replica servers for MongoDB database

You must create replica servers to avoid any system failures. The replica servers should have the capacity to store multiple distributed read operations.

For more information to create replica servers, see Deploy a Replica Server Set at docs.mongodb.com/manual.

Create database user

Create an user, for example, DBUser using the Wyse Management Suite to access MongoDB.

NOTE: The database user and password are examples and can be created using a different name and password at your work place.

Run the following command to create the 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" } ]
})
```

Create DBadmin user for MongoDB

Login to the MongoDB using the user account created in the previous section. The DBadmin user is created with the administrative privileges.

Run the following command to create the DBadmin user:

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

Edit mongod.cfg file

You must edit the `mongod.cfg` file to enable the security for the MongoDB database.

- 1 Login to MongoDB as root user that you have already created and run the following command:
`mongo -uroot -pAdmin#123 admin`
- 2 Go to `\data\bin\mongod.cfg` directory, and open `mongod.cfg` file in a text editor.
- 3 Edit `mongod.cfg` file as shown in the following command:

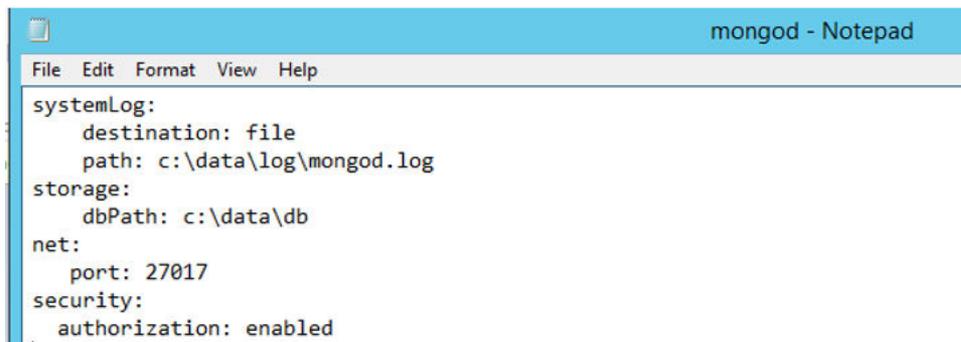


Figure 26. Edit `mongod.cfg`

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

NOTE: The port numbers will change depending on the system at the work place.

- 4 Save `mongod.cfg` and exit.

Initiate replication on the servers

Ensure that you disable firewall on Windows and stop Tomcat servers if they are running.

- 1 Login to MongoDB as root user that you have already created and run the following command:
`mongo -uroot -pAdmin#123 admin`

- 2 Go to `\data\bin\mongod.cfg` directory, and open `mongod.cfg` file in a text editor.
- 3 Add the following three lines in the `mongod.cfg` file:

```
keyFile: c:\data\log\mongod.key.txt  
  
replication:  
  
replSetName: wms
```

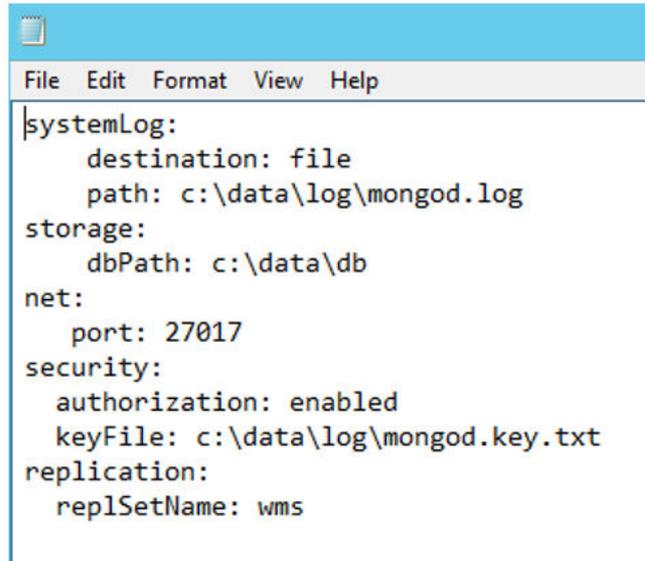


Figure 27. Enabling security

- 4 Create `mongod.key.txt` file and copy on all the three servers.

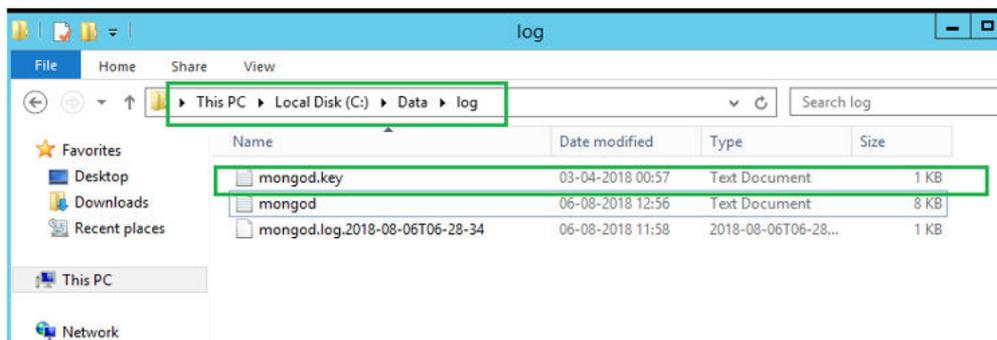


Figure 28. Copy the mongod key file

- 5 After you copy the file, stop the mongod service by running the following command:

```
net stop mongodb
```
- 6 Initiate replication on the primary node of the MongoDB cluster logging in using DBadmIn user and then run the following command:

```
rs.initiate();
```
- 7 Check the replication status by running the following command:

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

```

ums:OTHER>
ums:PRIMARY>
ums:PRIMARY> rs.status();
{
  "set" : "ums",
  "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
}

```

Figure 29. Replication status

- 8 Start mongod service and add the secondary nodes to the second and third nodes in the MongoDB cluster:

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

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

NOTE: The port numbers will differ based on the systems at your network and systems.

- 9 After you add the nodes in the MongoDB cluster, check the replication status by running the following commands for the primary and secondary nodes:

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

```

"set" : "wms",
"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

Figure 30. Status in primary server

```

"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

Figure 31. Secondary server status

Achieve high availability for Teradici devices

Wyse Management Suite uses the HAProxy hosted on the Ubuntu server 16.04.1 LTS to perform load balancing between the EMSDK servers. HAProxy is a load balancer proxy that can also provide high availability based on how it is configured. It is a popular open source software for TCP/HTTP Load Balancer, and proxy solution which runs on Linux operating system. The most common use is to improve the performance and reliability of a server environment by distributing the workload across multiple servers.

The following points explain how to achieve high availability for Teradici devices using HAProxy on Linux operating system:

- There will be only one instance of Teradici server as part of high availability with Wyse Management Suite.
- Teradici device support requires installation of EMSDK. EMSDK is a software component provided by Teradici that is integrated into Wyse Management Suite. Wyse Management Suite Installer installs EMSDK can be installed on Wyse Management Suite server or on a separate server. You need minimum of two instances of EMSDK to support more than 5000 devices, and all EMSDK servers should be on remote servers.
- Only one instance of EMSDK can be installed per server.
- Teradici Device support requires a PRO license.
- High availability of Teradici will be provided through HAProxy.
- If Teradici server goes down, device will reconnect automatically to the next available EMSDK server.

Install and configure HAProxy

HAProxy which is the load balancer for ThreadX 5x devices is configured on Ubuntu Linux version 16.04.1 with HAProxy version 1.6.

Do the following to install and configure HAProxy on Ubuntu Linux system:

- 1 Log in to Ubuntu system using the user credentials used during the installation of Ubuntu operating system.
- 2 Run the following commands to install 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 Run the following command to take backup of the original configuration:

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

- 4 Edit the HAProxy configuration file in a suitable text editor by running the following commands:

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

Add the following entries in the configuration file:

```
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>
```

NOTE: Administrator must add additional back end servers beyond the total number of client's capacity to have seamless failover.

- 5 Save the changes to the `haproxy.cfg` file by typing CTRL+O.

The following text is a sample HAProxy configuration file:

```
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 Validate the HAProxy configuration by running the following command:

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

If the configuration is valid, the message **Configuration is Valid** is displayed.

- 7 Restart HAProxy service by running the following command:

```
Sudo service haproxy restart
```

- 8 Stop HAProxy by running the following command:

```
serviceSudo service haproxy stop
```

Install Wyse Management Suite server

Ensure that the following components are configured before you install Wyse Management Suite server:

- Windows Failover Cluster on two Nodes
- MongoDB Server is running with replica set
- MySQL InnoDB Cluster set-up is running
- MySQL Router is installed on the two Nodes

Do the following to install Wyse Management Suite server:

- 1 Launch the Wyse Management Suite v1.3 installer screen.
- 2 Select **Custom Type** and **Teradici EMSDK** and then click **Next**.
- 3 Select the **External MongoDB** option (MongoDB Cluster with Replica set that you have created. For example, wms. Type the Remote Primary Mongo DB server information, Port number, and MongoDB Username and password in the respective fields and then click **Next**.
- 4 Select the **External MariaDB** option for MySQL. Use MySQL router address (Local Host if it is installed on Wyse Management Suite server node).

NOTE: Ensure that the Stratus user account is created on MySQL server.

- 5 Type MySQL Router information in the **External Maria DB Server** fields with the port number. Type the MySQL database user account information that you have created initially. The **Port Selection** screen is displayed with the port details. This port is used by MySQL Router. The default port is 6466. However, you can also change the port number.
- 6 Type the user name that has administrative privileges and e-mail address with the Teradici EMSDK port number and CIFS user account information.
- 7 Type the Destination Installation folder path and Shared UNC path for the local repository and then click **Next**. The message **The installation was successful** is displayed.

NOTE: The shared UNC path should be kept out of both the Windows Server where Wyse Management Suite application is installed.

NOTE: Before you install Wyse Management Suite application on Node 2, make sure to delete the 'Data' folder present in the Wyse Management Suite Local Repository; which was created during installation on Node 1. After 'Data' folder is deleted from the shared UNC WMS Local Repository path, you can install Wyse Management Suite Application in the Node 2 of the Windows Cluster.

Install Wyse Management Suite on Windows Server 2012

To install the Wyse Management Suite on a private cloud, do the following:

- 1 Double-click the installer package.
- 2 On the **Welcome** screen, read the license agreement, and click **Next**.
- 3 Select the **Setup Type** you want to install, and click **Next**. The available options are:
 - Typical—Requires minimum user interaction and installs embedded databases.
 - Custom—Requires maximum user interactions and is recommended for advanced users.
- 4 Select the **Setup Type** as **Custom**, and click **Next**.
The **Mongo Database Server** page is displayed.
- 5 Select the **External Mongo DB** option. Provide user name, password, database server details, and the port details, and click **Next**.

NOTE: The port field populates the default port which can be changed.

- 6 Click **Next** until the message **The Installation was Successful** is displayed.

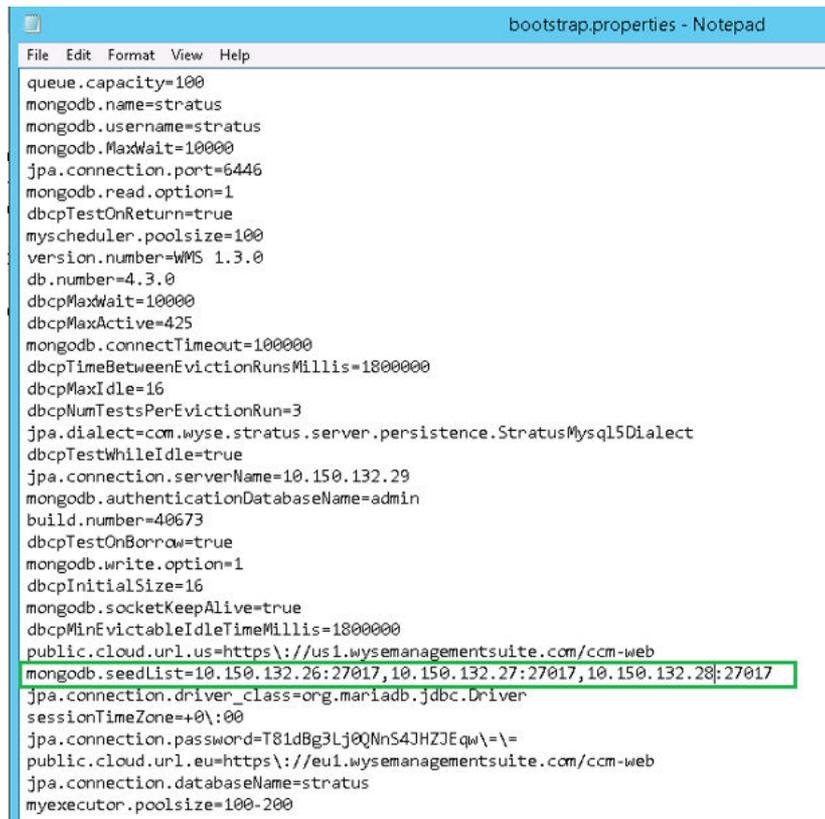
NOTE: Before you install Wyse Management Suite on server or node 2, make sure to delete the \Data folder from the local repository which is created during installation on server or node 1.

Post installation checks

Do the following to check the high availability for Wyse Management Suite version 1.3:

- Launch the Wyse Management Suite admin portal and check whether you are able to login using the web interface.
- Edit the `bootstrap.properties` file in the Tomcat sever in under the `\Dell\WMS\Tomcat-8\webapps\ccm-web\WEB-INF\classes` folder for MongoDB as follows:

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



```
bootstrap.properties - Notepad
File Edit Format View Help
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
public.cloud.url.us=https://us1.wysemanagementsuite.com/ccm-web
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=T81dBg3Lj0QnN54JHZJEqW\=\=
public.cloud.url.eu=https://eu1.wysemanagementsuite.com/ccm-web
jpa.connection.databaseName=stratus
myexecutor.poolsize=100-200
```

Figure 32. Edit bootstrap.properties file

- Login to MongoDB and update `bootstrapProperties` table with **Windows Cluster Virtual IP/Hostname of Access Point** as value for the following attributes:

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

Do the following to make changes in the MongoDB tables:

- 1 In the Stratus database access **Collections** and then select the **bootstrapProperties** table.
- 2 Update the MySQL tables and restart the Tomcat on both the nodes. Manually update `mysql` database table to retain the `ServerIp` in the `ServersInCluster` table to be active by running the following command:

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

NOTE: Ensure that there is only one record in `serversInCluster` table and if there are more than one record, delete the excess records.

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

Troubleshooting

This section provides troubleshooting information for Wyse Management Suite version 1.3 for the cluster set up.

- Problem: Where is the Wyse Management Suite log file located to check server installation issues.
Workaround: The log file is in the **%temp% WMSInstall.log** folder.
- Problem: Where is the Tomcat service related log file located to check the application related issues.

Workaround: The log file is in the **\Program Files\DELL\WMS\Tomcat-8\stratus.log** folder.

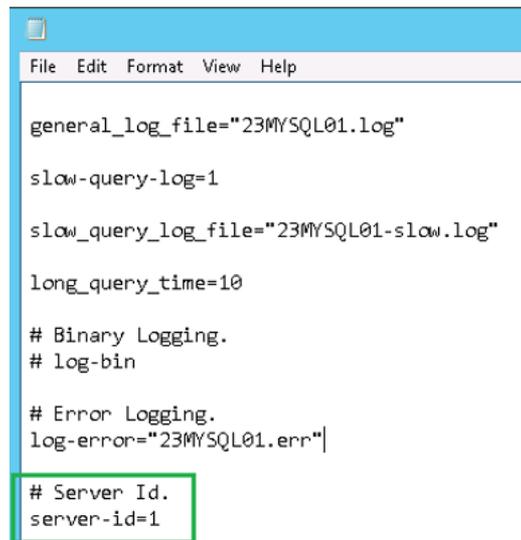
- Problem: If any of the server or node in the cluster stops working and is not part of the MySQL InnoDB cluster.

Workaround: Perform the following steps at the command prompt:

```
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
```

- Problem: If you do not add the server ID in the MySQL InnoDB cluster, an error message is displayed.

Workaround: Change the server ID entries in the **my.conf** file located in the **\ProgramData\MySQL\MySQL Server 5.7** directory.



```
File Edit Format View Help

general_log_file="23MYSQL01.log"

slow-query-log=1

slow_query_log_file="23MYSQL01-slow.log"

long_query_time=10

# Binary Logging.
# log-bin

# Error Logging.
log-error="23MYSQL01.err"

# Server Id.
server-id=1
```

Figure 33. change server ID