



# Click Studios

## SQL Server Always On Availability Groups for Passwordstate High Availability

# Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>3</b>
<b>2</b>	<b>PREREQUISITES .....</b>	<b>4</b>
<b>3</b>	<b>CONFIGURE SQL SERVICES .....</b>	<b>5</b>
<b>4</b>	<b>INSTALL WINDOWS FAILOVER CLUSTER ROLE .....</b>	<b>6</b>
<b>5</b>	<b>SET UP THE FAILOVER CLUSTER (WSFC) .....</b>	<b>10</b>
<b>6</b>	<b>ENABLE ALWAYS ON AVAILABILITY GROUPS .....</b>	<b>14</b>
<b>7</b>	<b>SET UP A HIGH AVAILABILITY GROUP .....</b>	<b>15</b>
<b>8</b>	<b>CONFIGURING PASSWORDSTATE TO WORK WITH ALWAYS ON GROUP .....</b>	<b>27</b>

## 1 Overview

This document will provide instructions for configuring SQL Server Always On Availability Groups for High Availability of the Passwordstate Database.

These instructions are created using SQL Server 2017 Enterprise edition and are intended as a guide only. If you have any technical issues with SQL Server, please contact Microsoft for support.

## 2 Prerequisites

Following are some pre-requisites to installing and configuring SQL Server for Basic Availability Groups:

- You will need to have SQL Server 2012 (or above) Enterprise installed on two separate servers with the default instance set, and the default Port of 1433 configured. Neither of the machines that host SQL can be a domain controller.
- You'll also need one domain service account with "domain users" permissions on your network
- Passwordstate should be installed and communicating to a database on one of your SQL servers you have set up for this exercise
- A network share that Always On technology requires for database backups

Below is some information about a test environment used to document this process, to help you understand our instructions easier:

SQLDB1.halox.net - 10.0.0.146

- Microsoft Windows 2016 Server with SQL 2017 Standard installed
- Also hosts Passwordstate database

SQLDB2.halox.net – 10.0.0.147

- Microsoft Windows 2016 Server with SQL 2017 Standard installed

Win2k16installs.halox.net - 10.0.0.120

- Passwordstate web server

Domain privileged account:

- halox\sqlservice

Passwordstate SQL account:

- passwordstate\_user

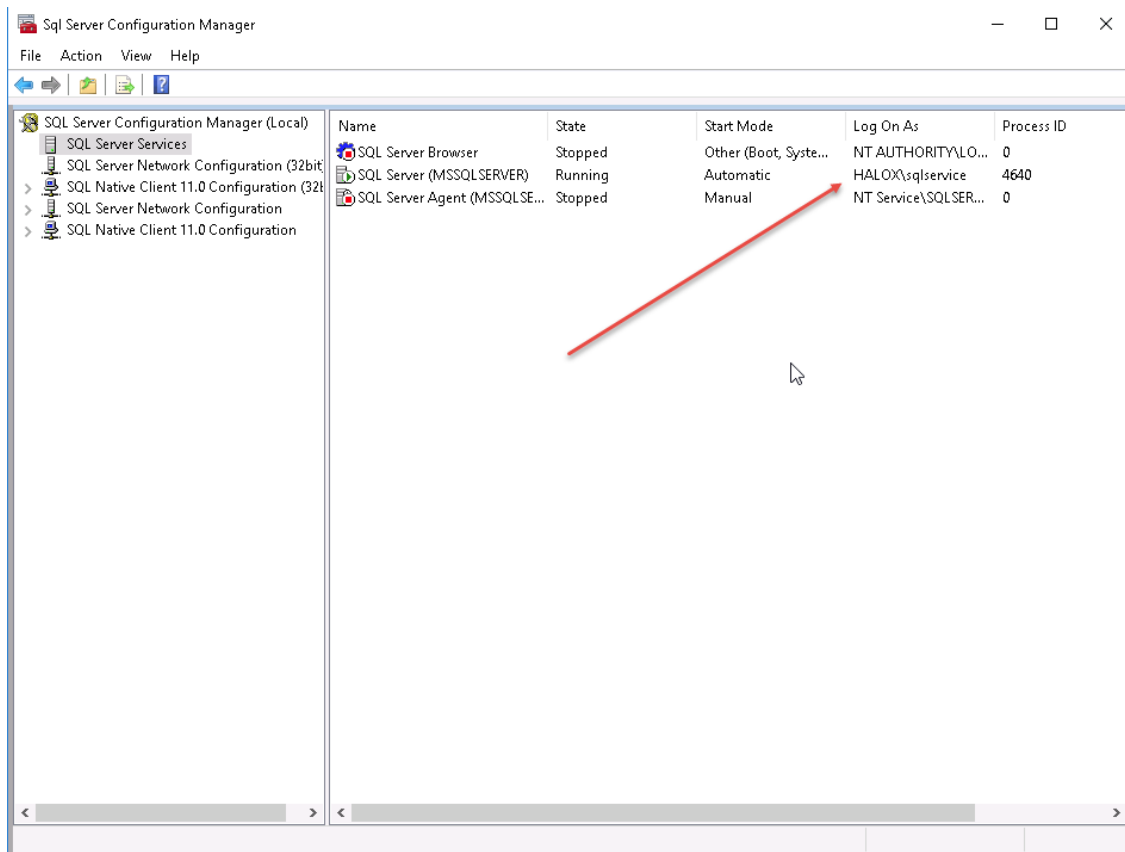
Network Share:

- [\\SQLDB1\\Backups](#)
- halox\sqlservice has full permissions to this share

### 3 Configure SQL Services

Confirm you have set your SQL Server service on both SQLDB1 and SQLDB2 to run under your domain account:

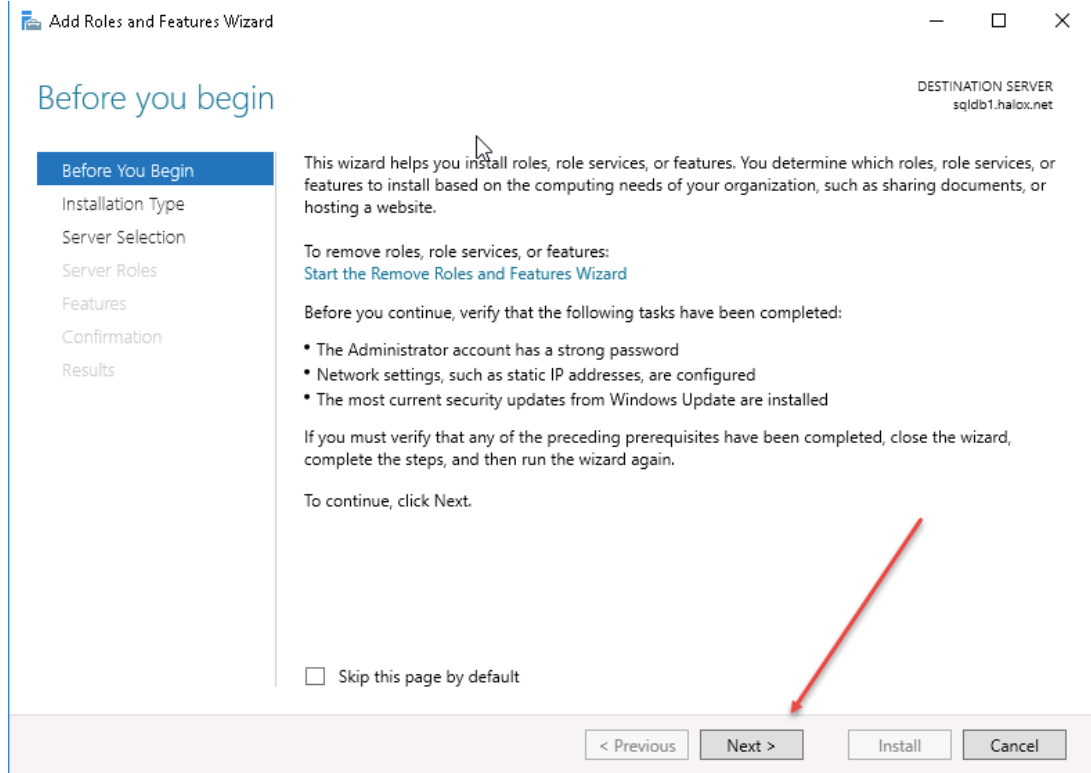
- Open **SQL Server Configuration Manager** and edit the properties of the SQL Server service, setting your domain account under the Log On tab.



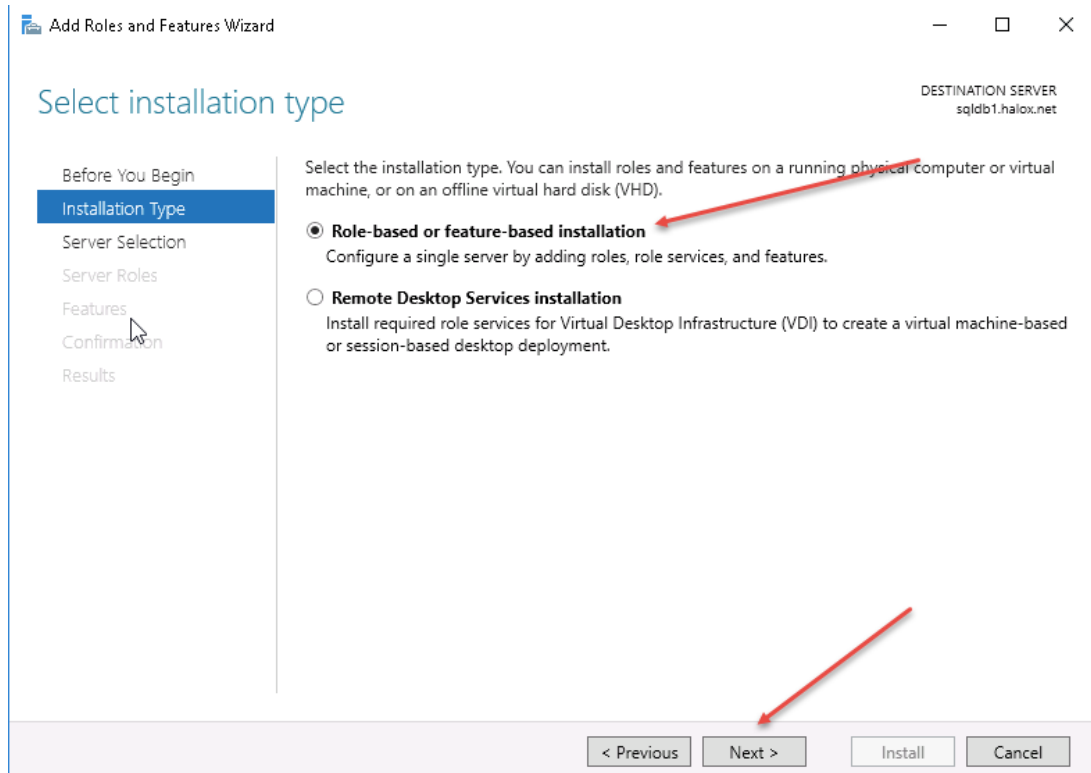
## 4 Install Windows Failover Cluster Role

On both SQLDB1 and SQLDB2, install the Failover Cluster role by following this process:

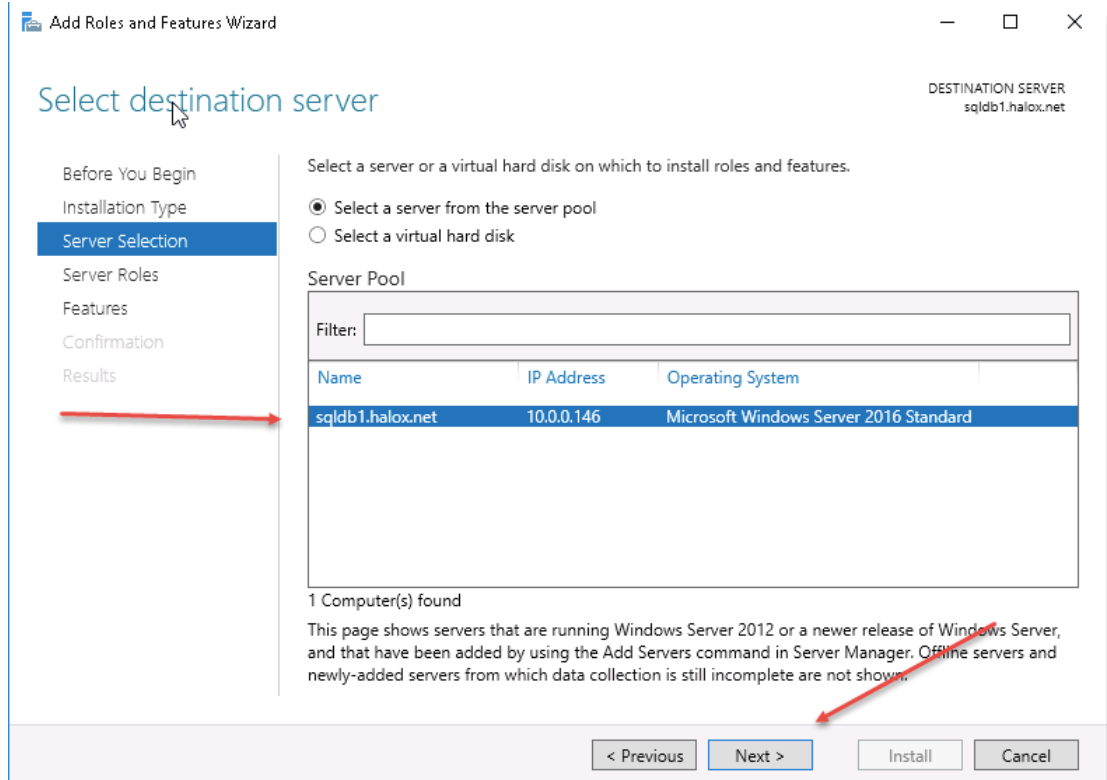
- Open **Server Manager**
- Select **Add Roles and Features**
- At the Before you Begin Page, select **Next**



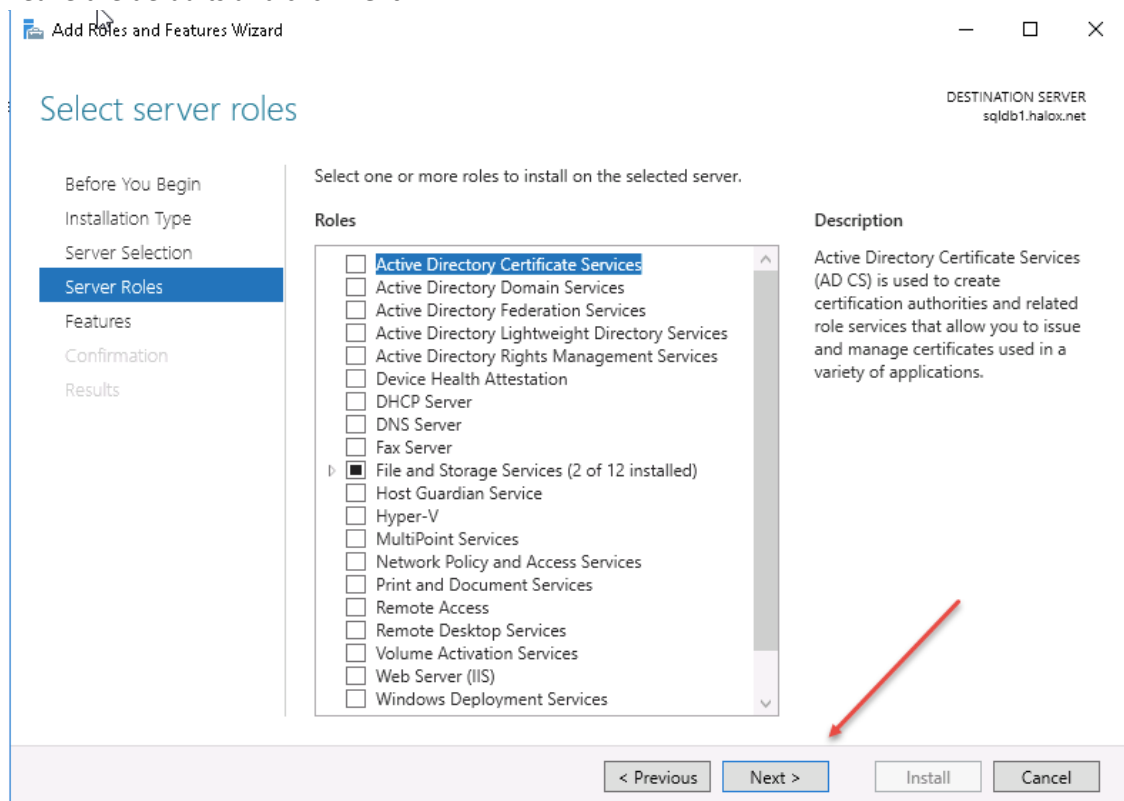
- Select **Role-based or feature-based installation** and click **Next**



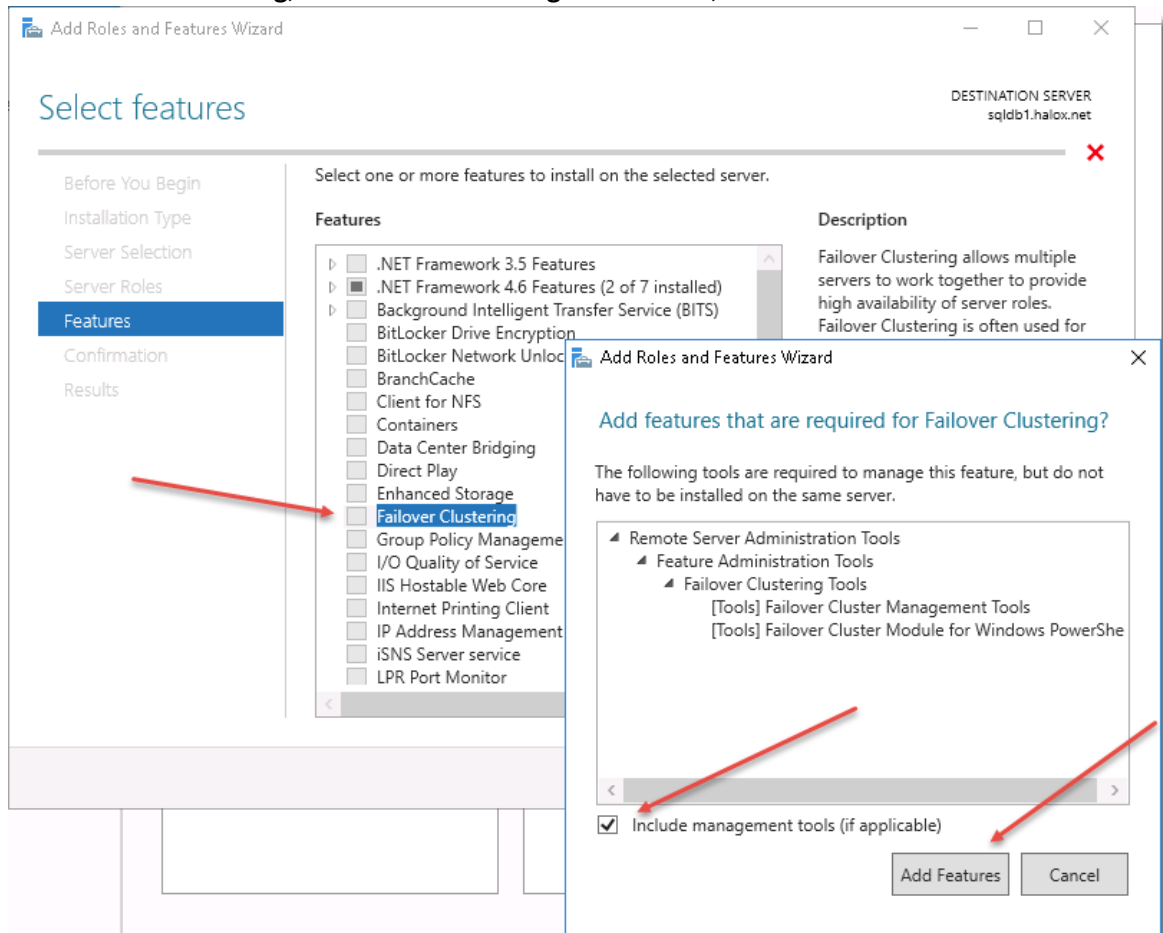
- Select your server and click **Next**



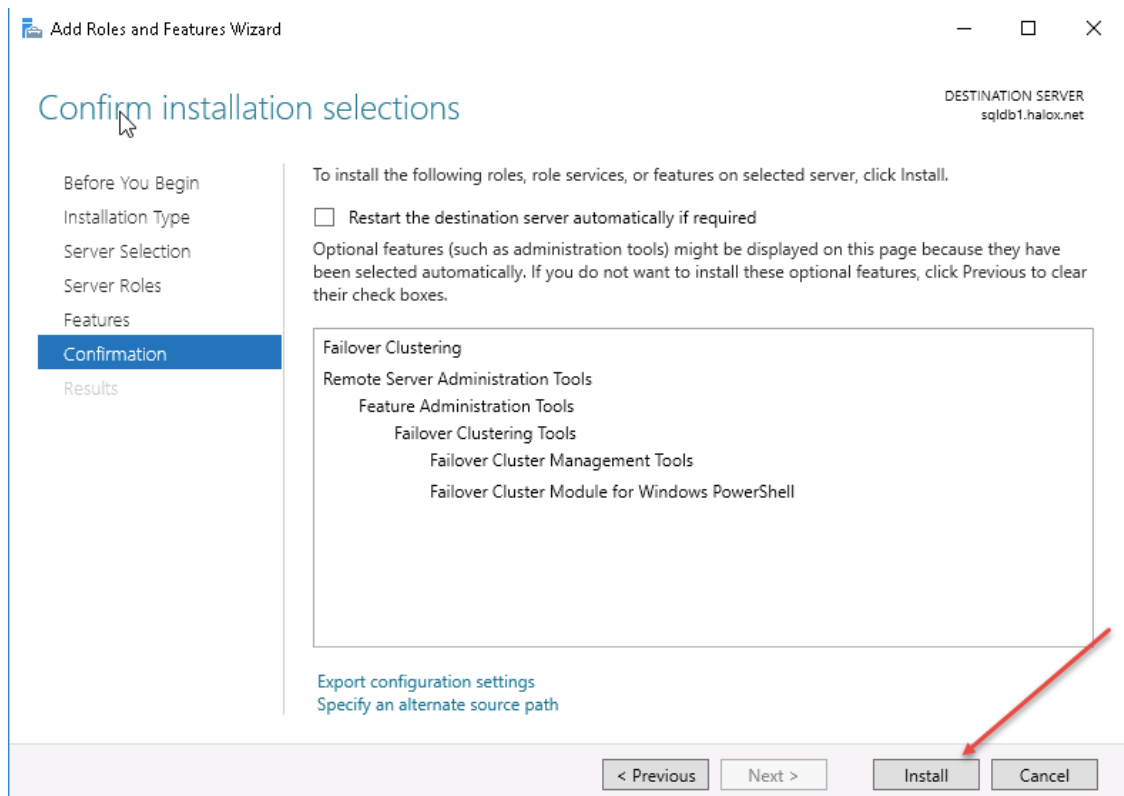
- Leave the defaults and click **Next**



- Tick **Failover Clustering**, Select **Include Management Tools**, Click **Add Features** and then click **Next**



- Click **Install**





## Click Studios

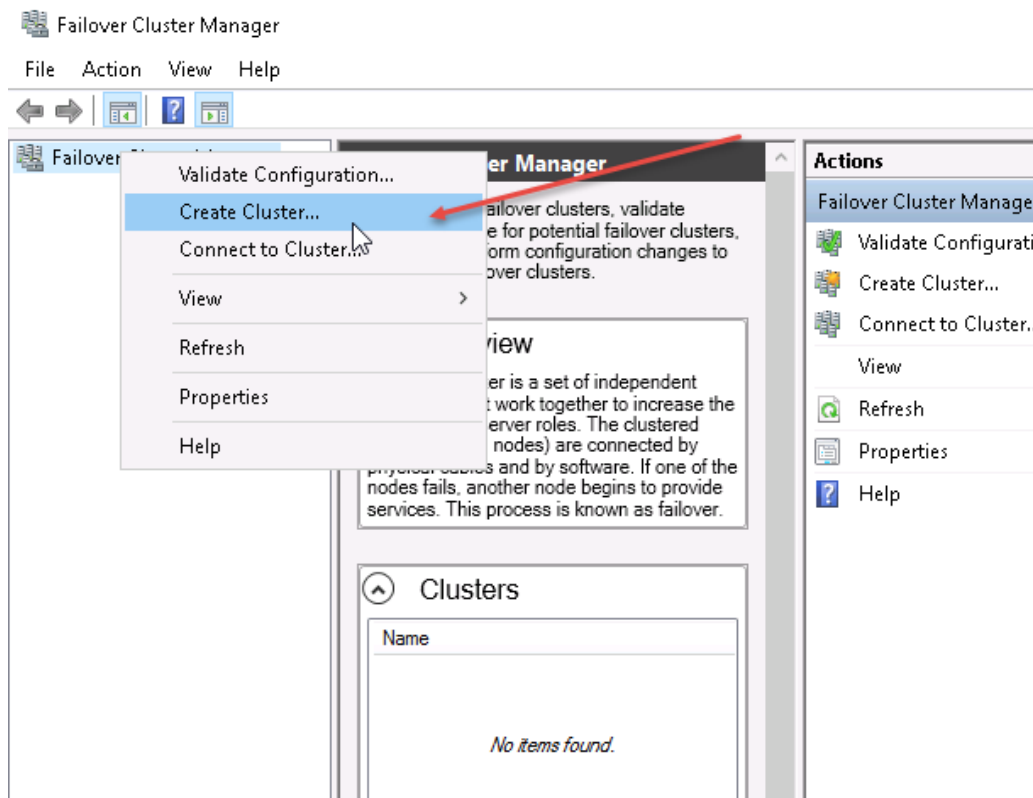
---

- When the installation has finished, reboot the server
- Ensure you repeat this process on both servers

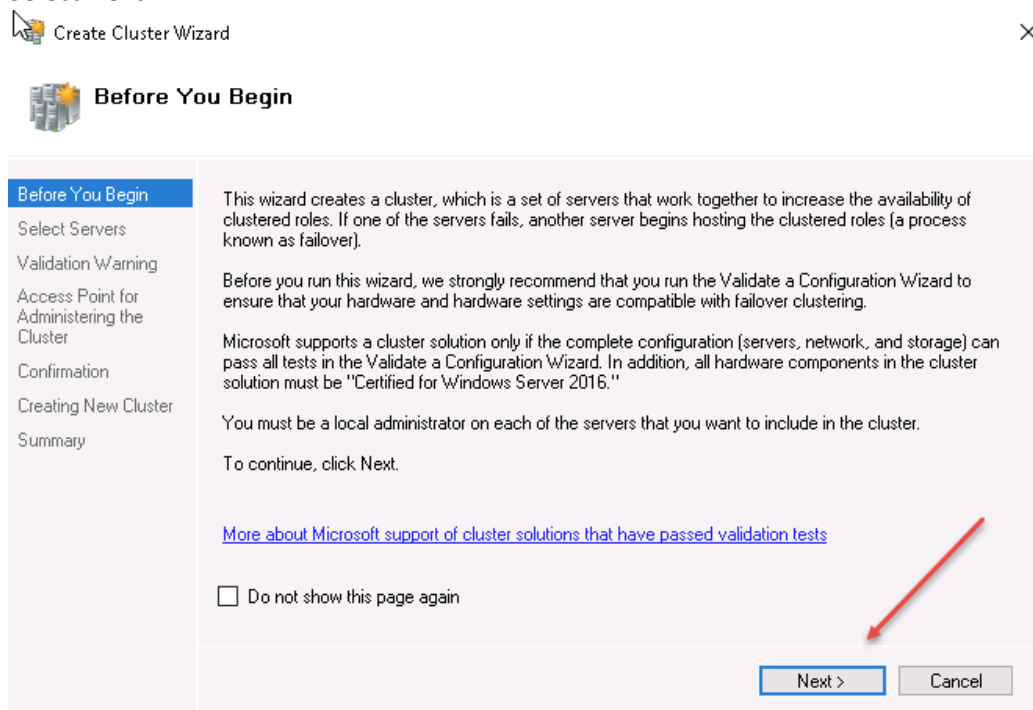
## 5 Set up the Failover Cluster (WSFC)

Set up a Windows Server Failover Cluster which includes both SQLDB1 and SQLDB2. To do this:

- On **SQLDB1**, open **Server Manager**
- Go to **Tools -> Failover Cluster Manager**
- Right click **Failover Cluster Manager** and select **Create Cluster**:



- **Select Next**



- Add **SQLDB1** and **SQLDB2** and click **Next**

The screenshot shows the 'Create Cluster Wizard' window, specifically the 'Select Servers' step. The left sidebar contains a list of steps: 'Before You Begin', 'Select Servers' (highlighted), 'Validation Warning', 'Access Point for Administering the Cluster', 'Confirmation', 'Creating New Cluster', and 'Summary'. The main area has a title bar with a close button and the text 'Select Servers'. Below the title bar, there is a section 'Before You Begin' with the instruction: 'Add the names of all the servers that you want to have in the cluster. You must add at least one server.' The 'Enter server name:' field is empty. The 'Selected servers:' list contains two entries: 'sqldb1.halox.net' and 'sqldb2.halox.net'. To the right of the list are buttons for 'Browse...', 'Add', and 'Remove'. At the bottom right, there are buttons for '< Previous', 'Next >' (highlighted with a red arrow), and 'Cancel'. A red arrow points from the 'Selected servers:' list to the 'Next >' button.

- Click **No** to running the validation tests and click **Next**

The screenshot shows the 'Create Cluster Wizard' window, specifically the 'Validation Warning' step. The left sidebar contains a list of steps: 'Before You Begin', 'Select Servers', 'Validation Warning' (highlighted), 'Access Point for Administering the Cluster', 'Confirmation', 'Creating New Cluster', and 'Summary'. The main area has a title bar with a close button and the text 'Validation Warning'. Below the title bar, there is a section 'Before You Begin' with a warning icon and the text: 'For the servers you selected for this cluster, the reports from cluster configuration validation tests appear to be missing or incomplete. Microsoft supports a cluster solution only if the complete configuration (servers, network and storage) can pass all the tests in the Validate a Configuration wizard. Do you want to run configuration validation tests before continuing?' There are two radio button options: 'Yes. When I click Next, run configuration validation tests, and then return to the process of creating the cluster.' and 'No. I do not require support from Microsoft for this cluster, and therefore do not want to run the validation tests. When I click Next, continue creating the cluster.' The 'No' option is selected. Below the options is a link: 'More about cluster validation tests'. At the bottom right, there are buttons for '< Previous', 'Next >' (highlighted with a red arrow), and 'Cancel'. A red arrow points from the 'No' option to the 'Next >' button.

- Add in the name of your cluster and the static IP Address it will be assigned. This will create a virtual computer object in Active Directory and a Host entry for this object in DNS.

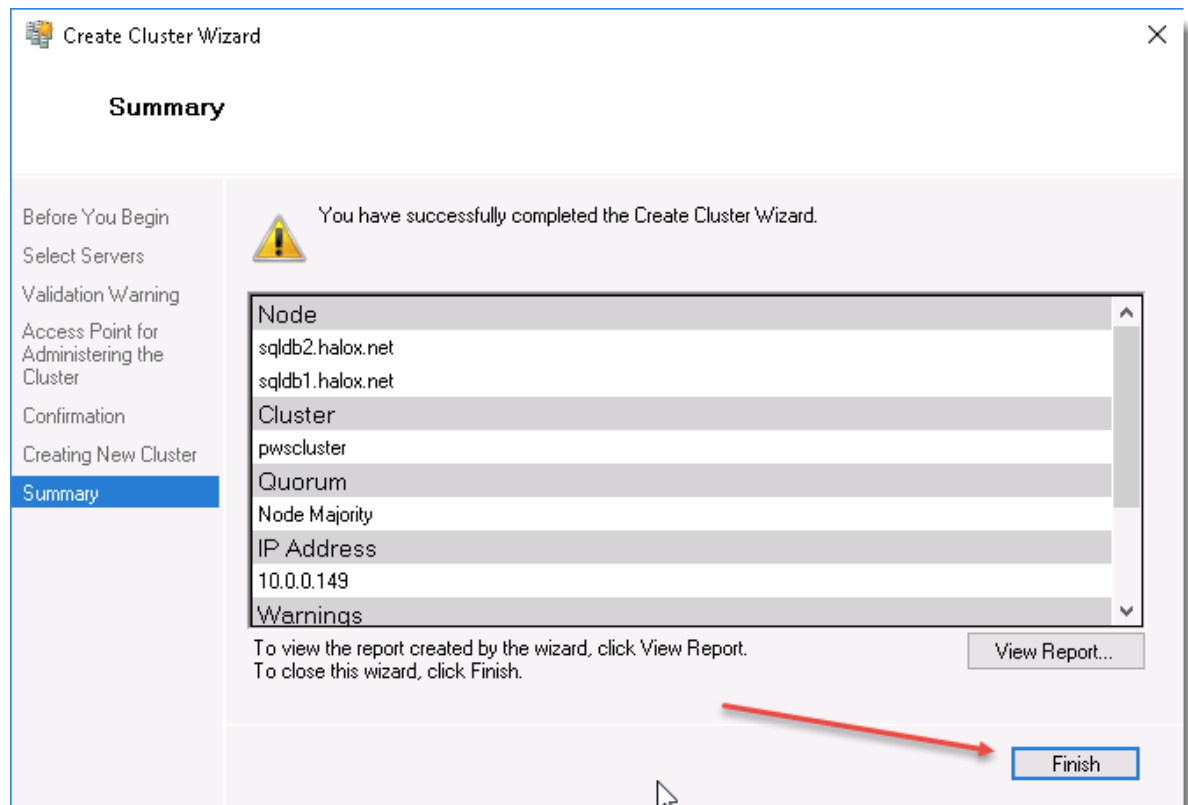
The screenshot shows the 'Create Cluster Wizard' window at the 'Access Point for Administering the Cluster' step. The left sidebar contains a list of steps: 'Before You Begin', 'Select Servers', 'Validation Warning', 'Access Point for Administering the Cluster' (highlighted), 'Confirmation', 'Creating New Cluster', and 'Summary'. The main area has a title bar with a cluster icon and the text 'Access Point for Administering the Cluster'. Below this, it says 'Type the name you want to use when administering the cluster.' and 'Cluster Name: pwscluster'. A red arrow points to the 'Cluster Name' text box. Below the text box is a warning icon and text: 'The NetBIOS name is limited to 15 characters. One or more IPv4 addresses could not be configured automatically. For each network to be used, make sure the network is selected, and then type an address.' Below this is a table with two columns: 'Networks' and 'Address'. The table has one row with a checked checkbox in the 'Networks' column, the text '10.0.0.0/24', and the IP address '10 . 0 . 0 . 149'. A red arrow points to the IP address field. At the bottom right are three buttons: '< Previous', 'Next >' (highlighted), and 'Cancel'.

	Networks	Address
<input checked="" type="checkbox"/>	10.0.0.0/24	10 . 0 . 0 . 149

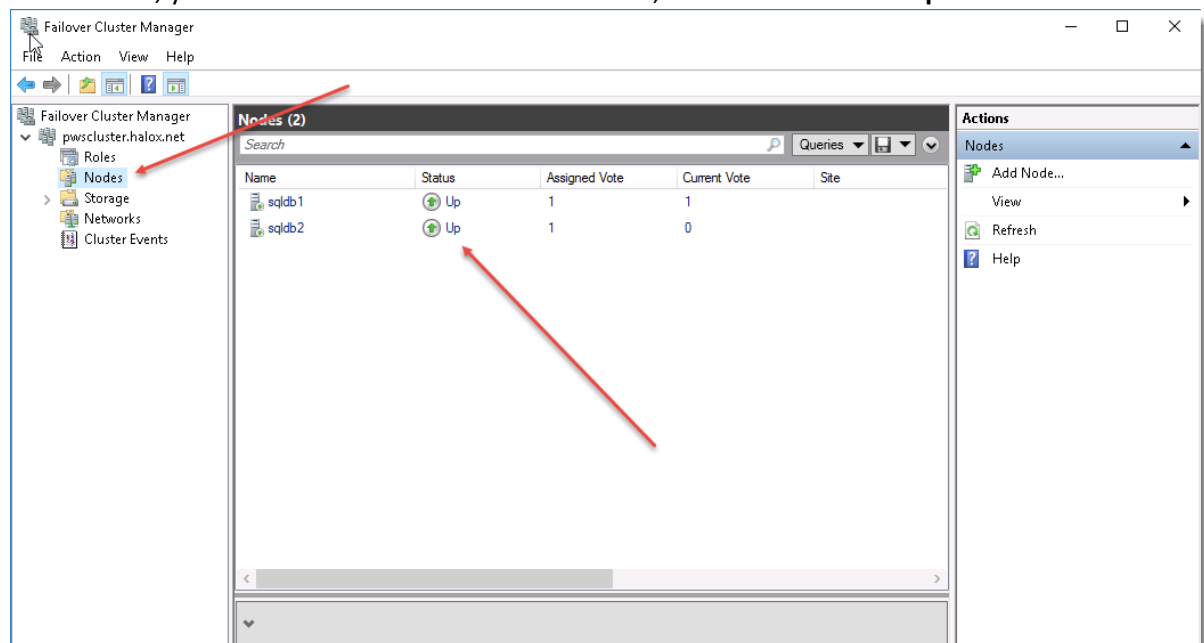
- Click **Next**

The screenshot shows the 'Create Cluster Wizard' window at the 'Confirmation' step. The left sidebar contains a list of steps: 'Before You Begin', 'Select Servers', 'Validation Warning', 'Access Point for Administering the Cluster', 'Confirmation' (highlighted), 'Creating New Cluster', and 'Summary'. The main area has a title bar with a cluster icon and the text 'Confirmation'. Below this, it says 'You are ready to create a cluster. The wizard will create your cluster with the following settings:'. Below this is a list box containing the following items: 'Cluster', 'pwscluster', 'Node', 'sqldb2.halox.net', 'sqldb1.halox.net', 'Cluster registration', and 'DNS and Active Directory Domain Services'. Below the list box is a checked checkbox and the text 'Add all eligible storage to the cluster. To continue, click Next.' A red arrow points to the 'Next >' button at the bottom right. The bottom right has three buttons: '< Previous', 'Next >' (highlighted), and 'Cancel'.

- Click **Finish**



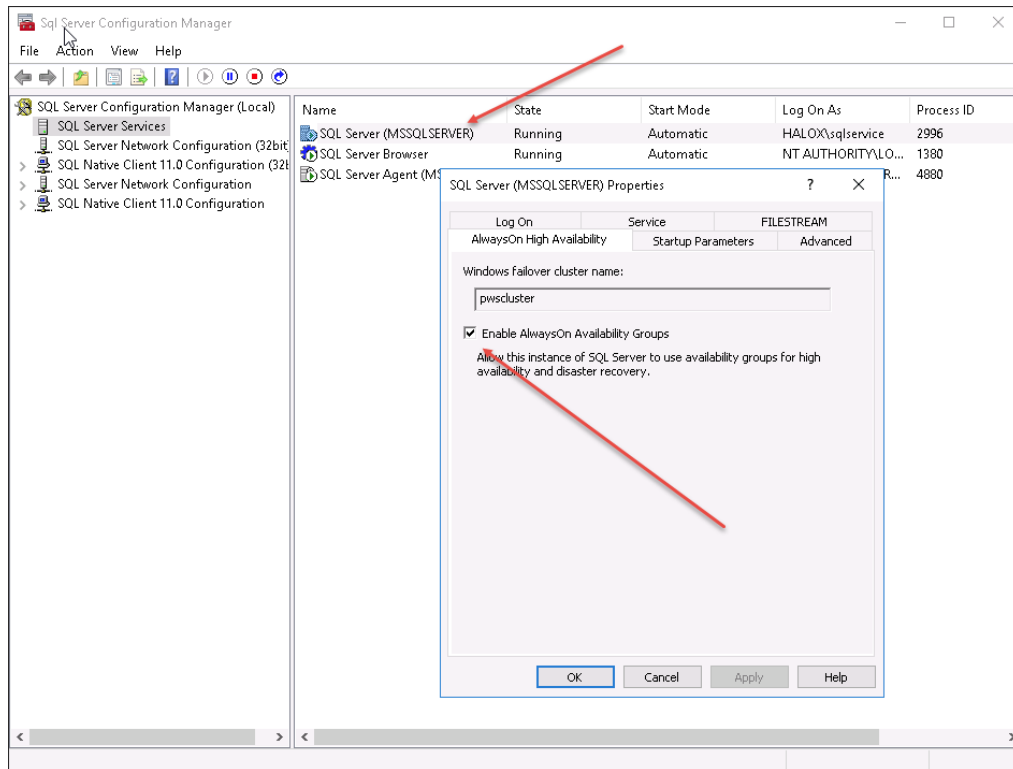
- This process will automatically create the cluster on SQLDB2, so if you log into any of your database servers now, you will see the cluster has been created, and both nodes are **Up**:



## 6 Enable Always On Availability Groups

To enable **Always On High Availability Groups** in SQL , perform the following steps on both SQLDB1 and SQLDB2

- Open **SQL Server Configuration Manager**, go to the properties of the **SQL Server** service, and tick the **Enable AlwaysOn Availability Groups** option. Click **OK** to the warning about the services needing to be restarted.



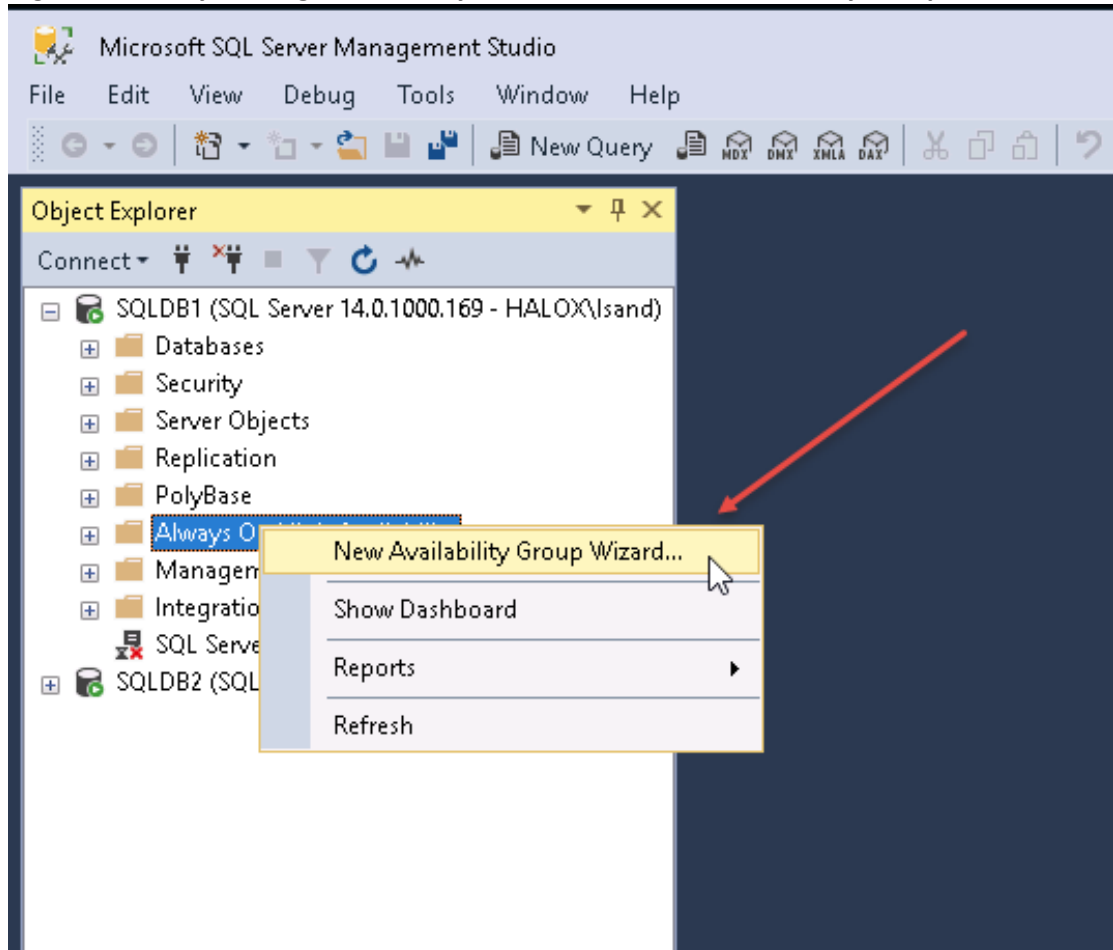
- Restart the **SQL Service** service on both servers after making this change

## 7 Set up a High Availability Group

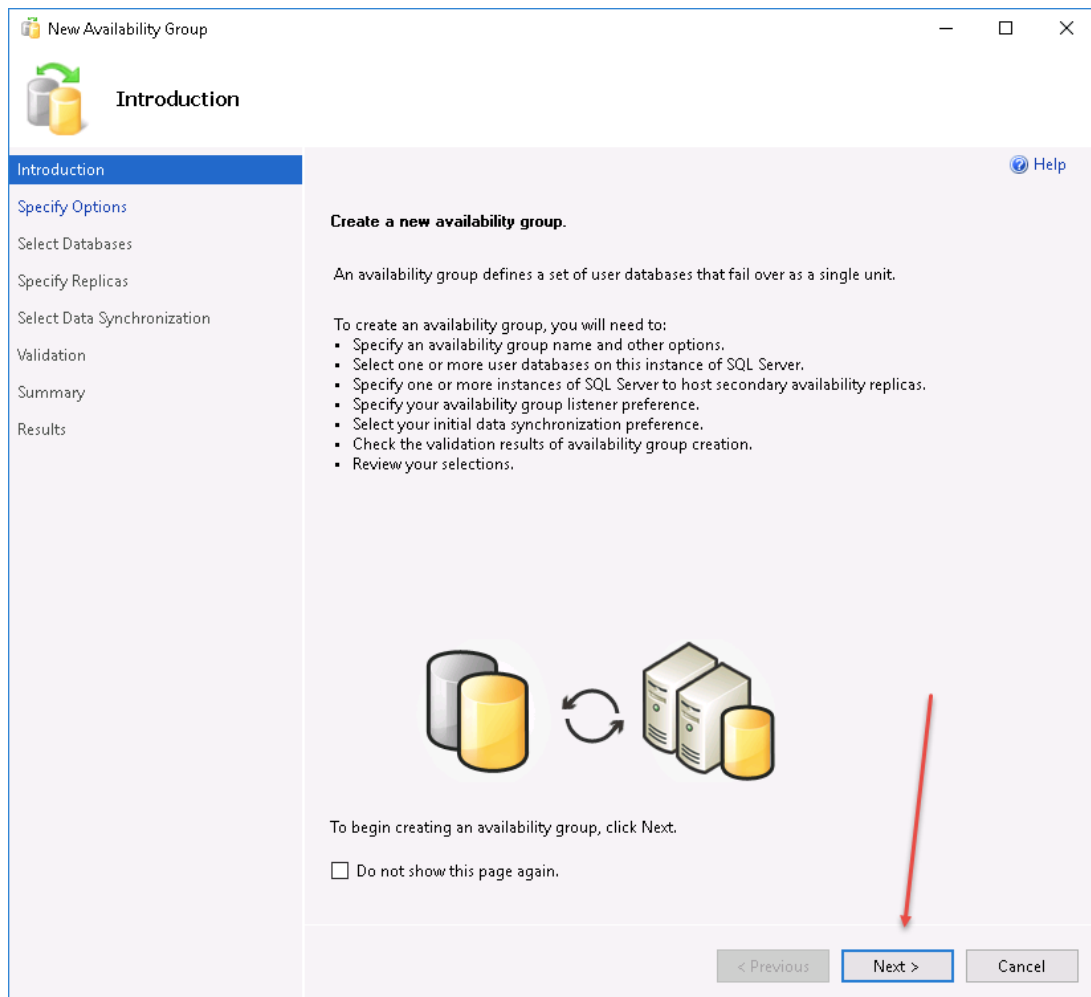
We will now create a High Availability Group which will automatically failover to a working database in the event one of them become unavailable. This process will create the Availability Group on both SQL servers, and it will synchronize your Passwordstate database between them.

It will also create a **Listener**, which is a virtual computer object in Active Directory. This will be used in Passwordstate to ensure your web site is always available.

- On **SQLDB1** open **SQL Management Studio Tools**
- Right click **Always On High Availability** and launch the **New Availability Group Wizard**



- Click **Next**





- Type in a name of the Availability Group as **Passwordstate**, select the Cluster Type as **Windows Server Failover Cluster**, and tick the option for **Database Level Health Detection**. Click **Next**.

New Availability Group

Specify Availability Group Options

Introduction

Specify Options

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Specify availability group options

Availability group name: Passwordstate

Cluster type: Windows Server Failover Cluster

☒ Database Level Health Detection

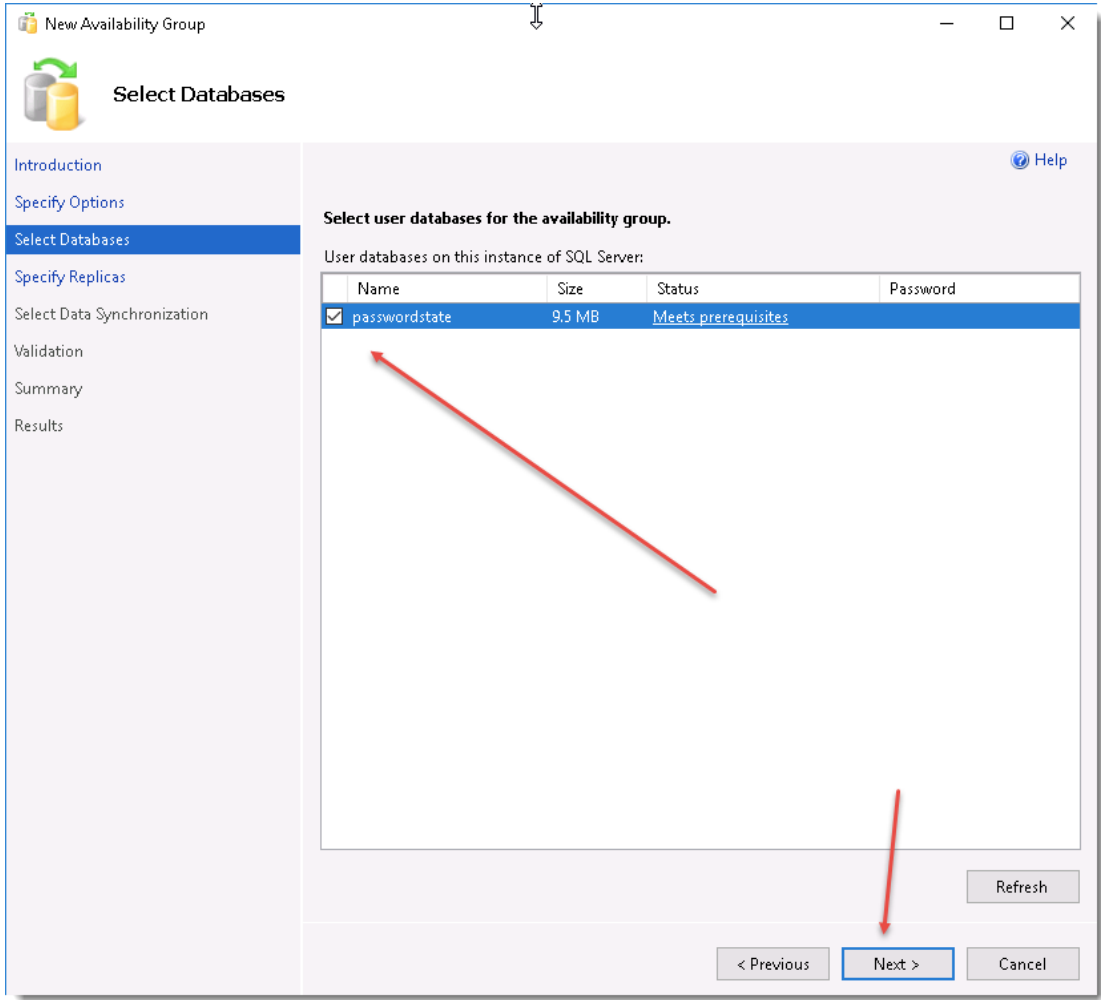
☐ Per Database DTC Support

< Previous

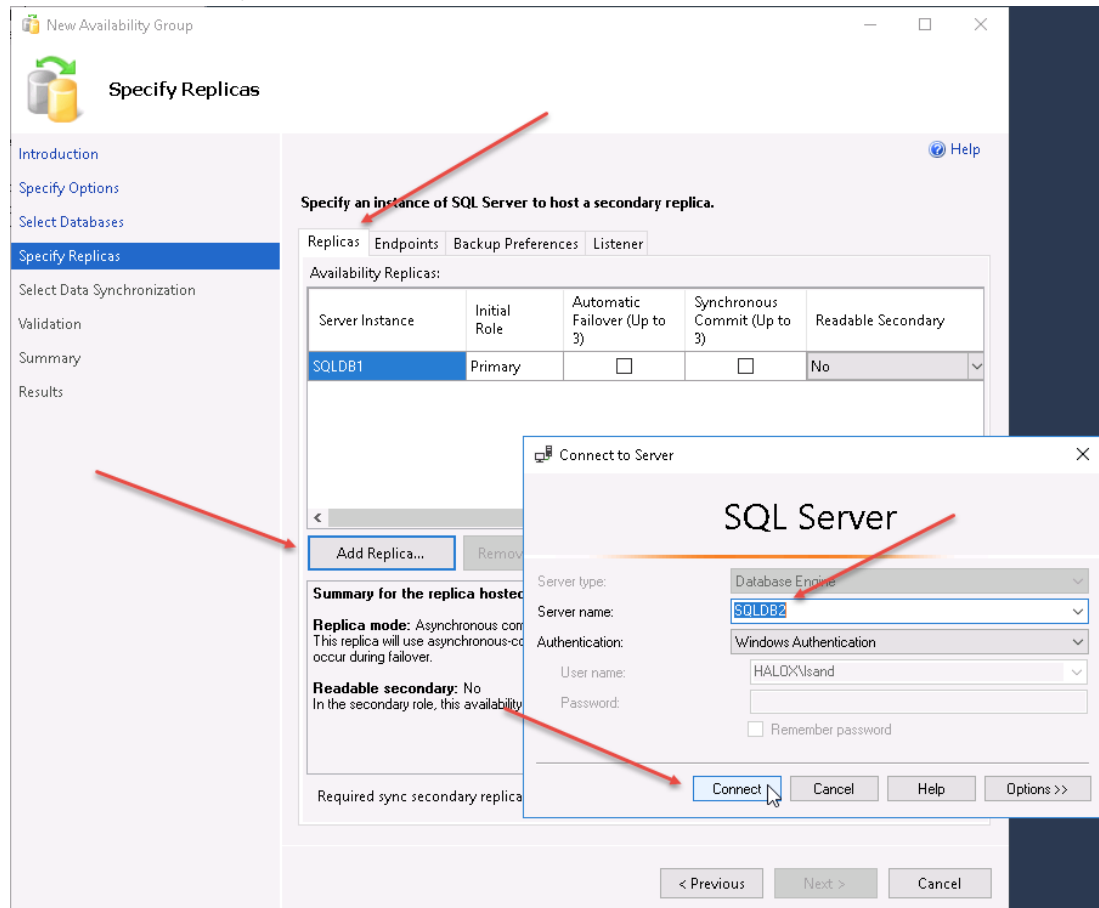
Next >

Cancel

- Tick the **Passwordstate** database and click **Next**:



- Under the **Relicas** tab, click the **Add Replica** button, connect to your second SQL Server, which in this case is **SQLDB2**, and then click **Connect**



- Select the **Automatic Failover** option, **Synchronous Commit**, and **Yes** as the Readable Secondary for both databases.

New Availability Group

## Specify Replicas

Introduction  
Specify Options  
Select Databases  
**Specify Replicas**  
Select Data Synchronization  
Validation  
Summary  
Results

Help

Specify an instance of SQL Server to host a secondary replica.

Replicas | Endpoints | Backup Preferences | Listener

Availability Replicas:

Server Instance	Initial Role	Automatic Failover (Up to 3)	Synchronous Commit (Up to 3)	Readable Secondary
SQLDB1	Primary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Yes
SQLDB2	Secondary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Yes

Add Replica... Remove Replica

**Summary for the replica hosted by SQLDB2**

**Replica mode:** Synchronous commit with automatic failover  
This replica will use synchronous-commit availability mode and will support both automatic failover and manual failover.

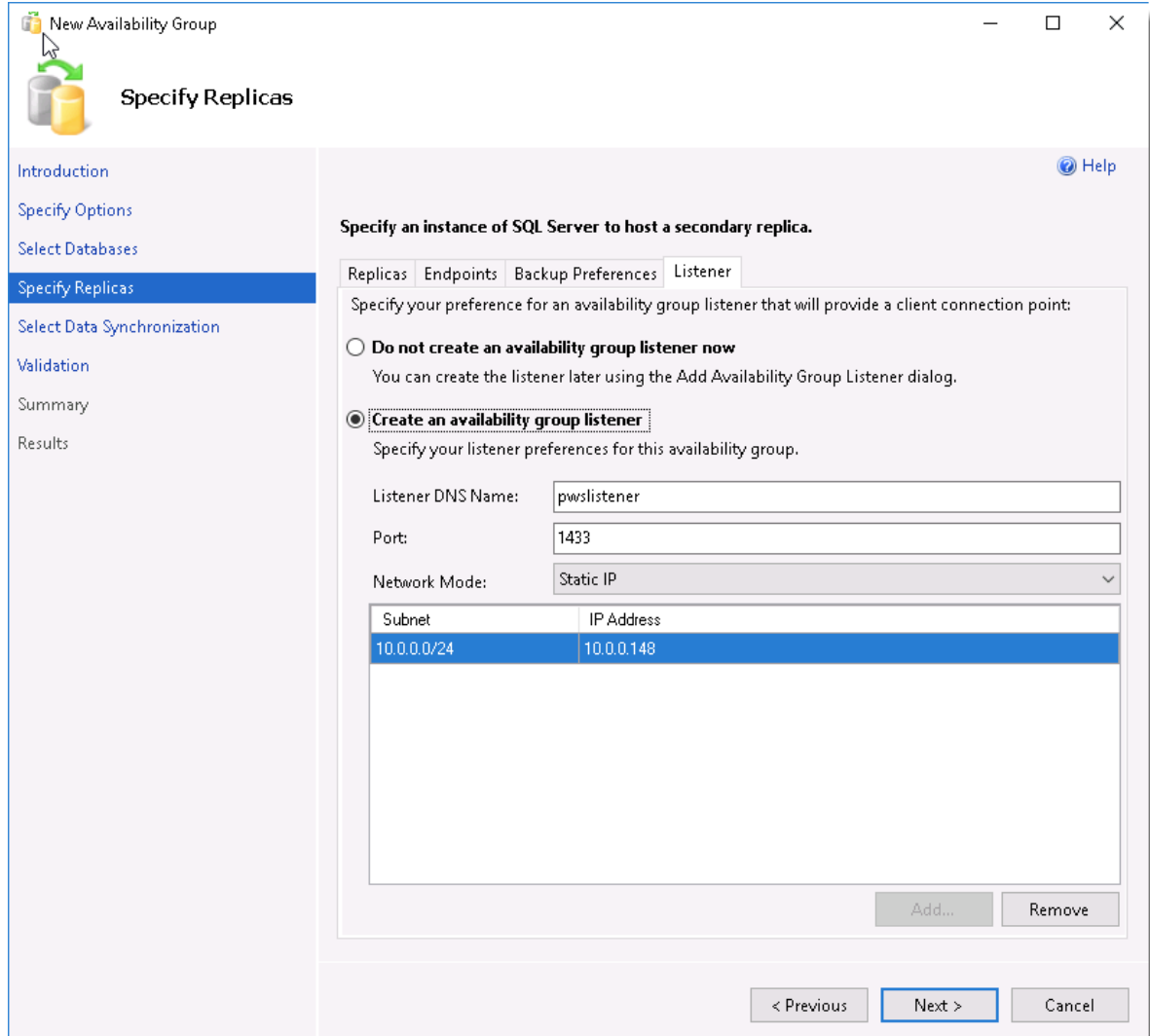
**Readable secondary:** Yes  
In the secondary role, this availability replica will allow all connections for read access, including connections running with older clients.

Required sync secondary replicas to commit: 0

< Previous Next > Cancel

- Under the **Listener** tab, enable the **Create an availability group listener** option and enter a DNS name of a server which will provide a client connection point.

This will create a virtual computer object in Active Directory and will also create a Host DNS record for this Active Directory object. Enter the port as **1433** and enter a static **IP Address** that will be set.



New Availability Group

Specify Replicas

Introduction  
Specify Options  
Select Databases  
Specify Replicas  
Select Data Synchronization  
Validation  
Summary  
Results

Help

Specify an instance of SQL Server to host a secondary replica.

Replicas Endpoints Backup Preferences Listener

Specify your preference for an availability group listener that will provide a client connection point:

☐ Do not create an availability group listener now  
You can create the listener later using the Add Availability Group Listener dialog.

☒ **Create an availability group listener**  
Specify your listener preferences for this availability group.

Listener DNS Name: pwslistener

Port: 1433

Network Mode: Static IP

Subnet	IP Address
10.0.0.0/24	10.0.0.148

Add... Remove

< Previous Next > Cancel

- Enable the **Full database and log backup** option and set the value of the share on your primary SQL server

The screenshot shows the 'New Availability Group' wizard in SQL Server Enterprise Manager. The title bar reads 'New Availability Group'. Below the title bar is a navigation pane on the left with the following steps: Introduction, Specify Options, Select Databases, Specify Replicas, **Select Data Synchronization** (highlighted), Validation, Summary, and Results. The main area is titled 'Select Initial Data Synchronization' and contains the following content:


**Select your data synchronization preference.**

- ☐ **Automatic seeding**  
SQL Server automatically creates databases for every selected secondary replica. Automatic seeding requires that the data and log file paths are the same on every SQL Server instance participating in the availability group.
- ☒ **Full database and log backup**  
Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group. Make sure the file share is accessible to all replicas and is mounted to the same directory on all Linux replicas.  
Specify the file share path in Windows format:  
   
Specify the file share location in Linux format:
- ☐ **Join only**  
Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary.
- ☐ **Skip initial data synchronization**  
Choose this option if you want to perform your own database and log backups of each primary database.

At the bottom of the wizard are three buttons: '< Previous', 'Next >', and 'Cancel'. A red arrow points from the 'Specify the file share location in Linux format' text box to the 'Full database and log backup' option.

- Ensure all of the tests have succeeded, and click **Next**

New Availability Group



Validation

Introduction

Specify Options

Select Databases

Specify Replicas

Select Data Synchronization

**Validation**

Summary

Results

Help

**Results of availability group validation.**

	Name	Result
✓	Checking for free disk space on the server instance that hosts secondary replica SQLDB2	<a href="#">Success</a>
✓	Checking if the selected databases already exist on the server instance that hosts second...	<a href="#">Success</a>
✓	Checking for the existence of the database files on the server instance that hosts seconda...	<a href="#">Success</a>
✓	Checking whether the endpoint is encrypted using a compatible algorithm	<a href="#">Success</a>
✓	Checking shared network location	<a href="#">Success</a>
✓	Checking replica availability mode	<a href="#">Success</a>
✓	Checking the listener configuration	<a href="#">Success</a>

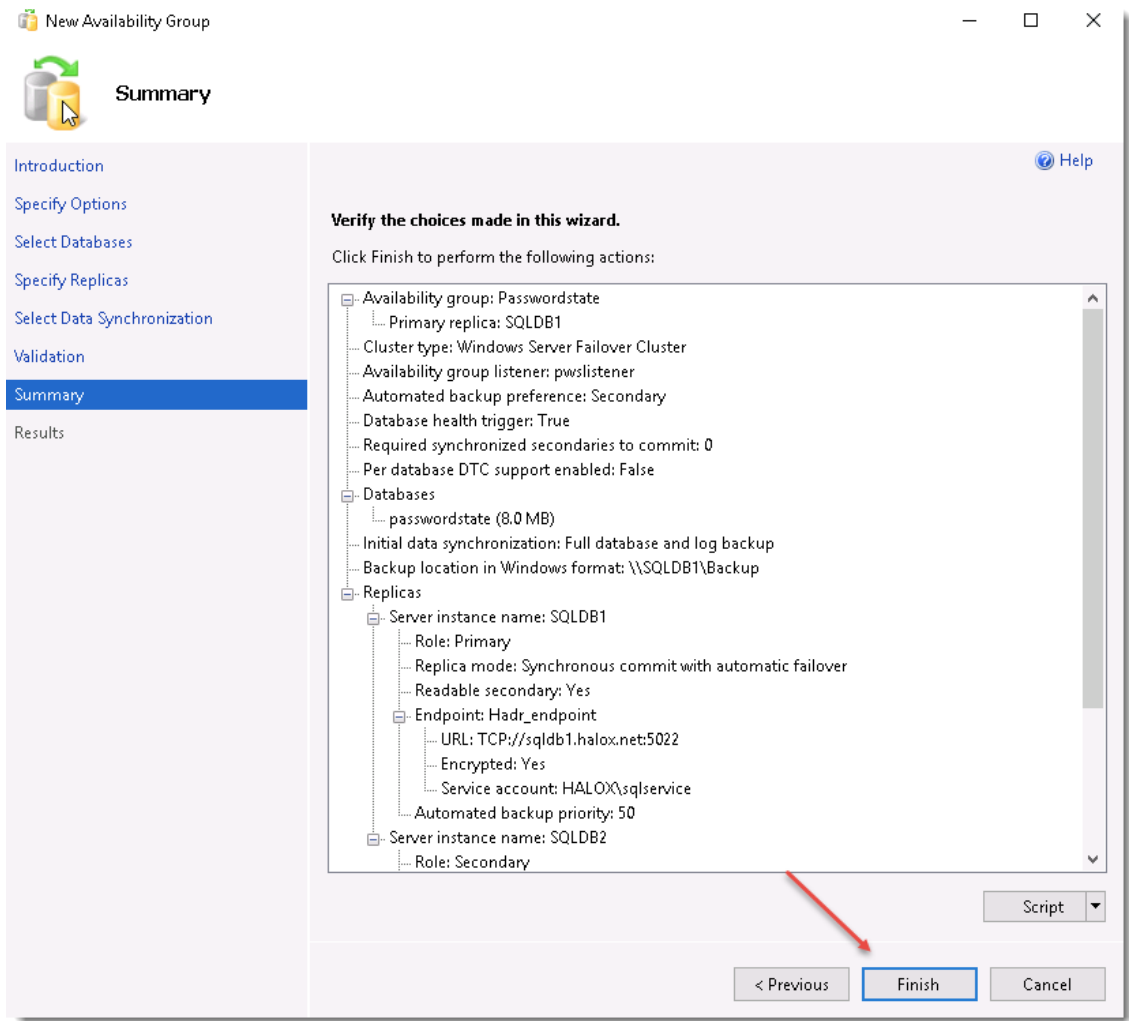
Re-run Validation

< Previous

**Next >**

Cancel


- Review the summary, and click **Finish**





- You should now see a successfully completed wizard

New Availability Group

Results

Introduction

Specify Options

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Help

✓

The wizard completed successfully.

Summary:

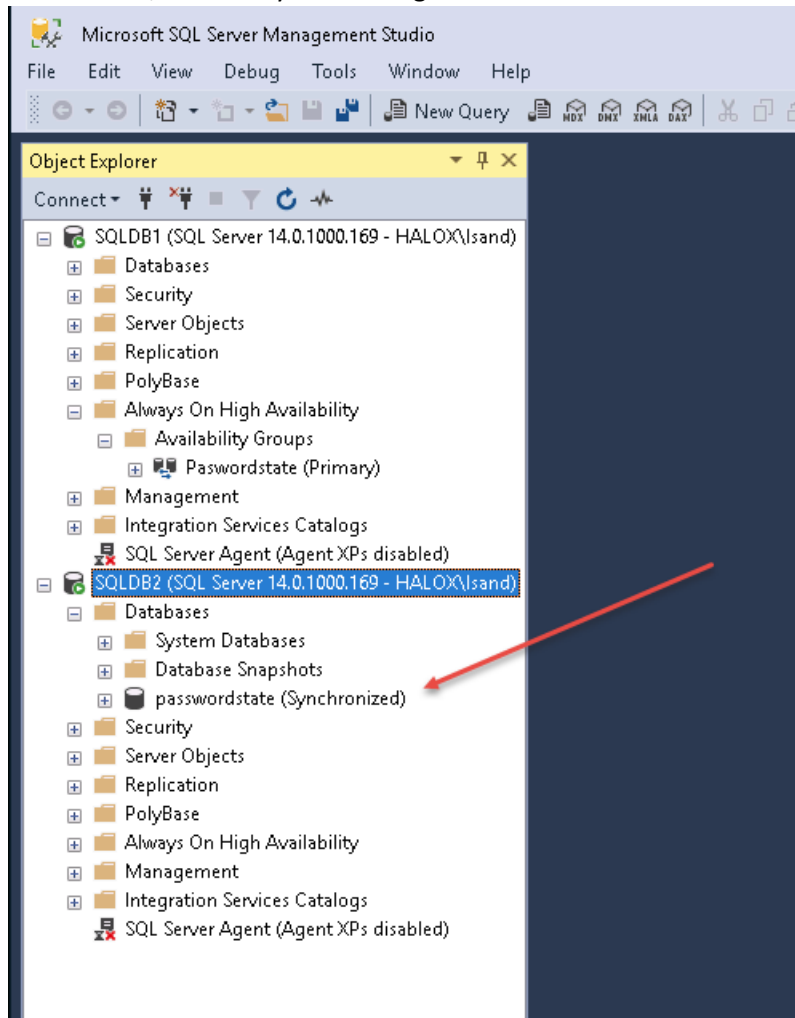
Name	Result
✓ Configuring endpoints.	<a href="#">Success</a>
✓ Starting the 'AlwaysOn_health' extended events session on 'SQLDB1'.	<a href="#">Success</a>
✓ Configuring endpoints.	<a href="#">Success</a>
✓ Starting the 'AlwaysOn_health' extended events session on 'SQLDB2'.	<a href="#">Success</a>
✓ Creating availability group 'Paswordstate'.	<a href="#">Success</a>
✓ Waiting for availability group 'Paswordstate' to come online.	<a href="#">Success</a>
✓ Creating Availability Group Listener 'ag-listener1'.	<a href="#">Success</a>
✓ Joining secondaries to availability group 'Paswordstate'.	<a href="#">Success</a>
✓ Validating Windows Failover Cluster quorum vote configuration.	<a href="#">Success</a>
✓ Creating a full backup for 'passwordstate'.	<a href="#">Success</a>
✓ Restoring 'passwordstate' on 'SQLDB2'.	<a href="#">Success</a>
✓ Backing up log for 'passwordstate'.	<a href="#">Success</a>
✓ Restoring 'passwordstate' log on 'SQLDB2'.	<a href="#">Success</a>
✓ Joining 'passwordstate' to availability group 'Paswordstate' on 'SQLDB2'.	<a href="#">Success</a>

< Previous

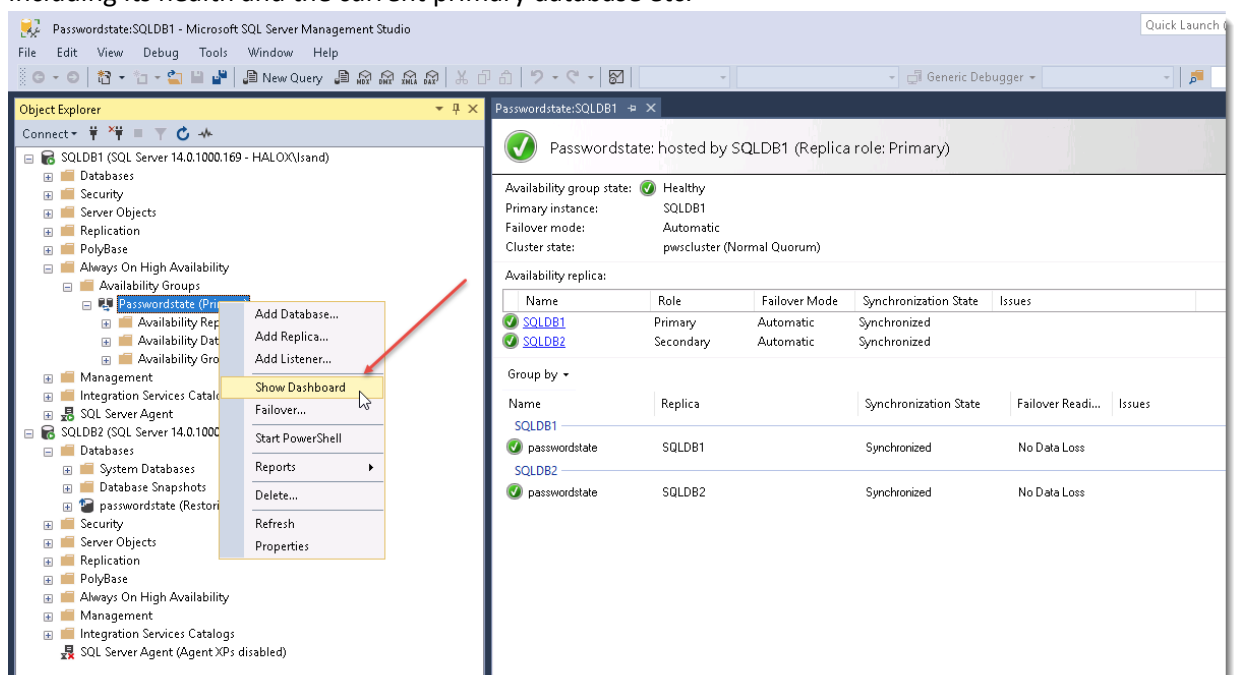
Next >

Close

- If you now connect to your secondary server, you will now see the Passwordstate database has been added, and it is synchronizing



- If you run open the Always On Dashboard, you will also see information about this new Group including its health and the current primary database etc.



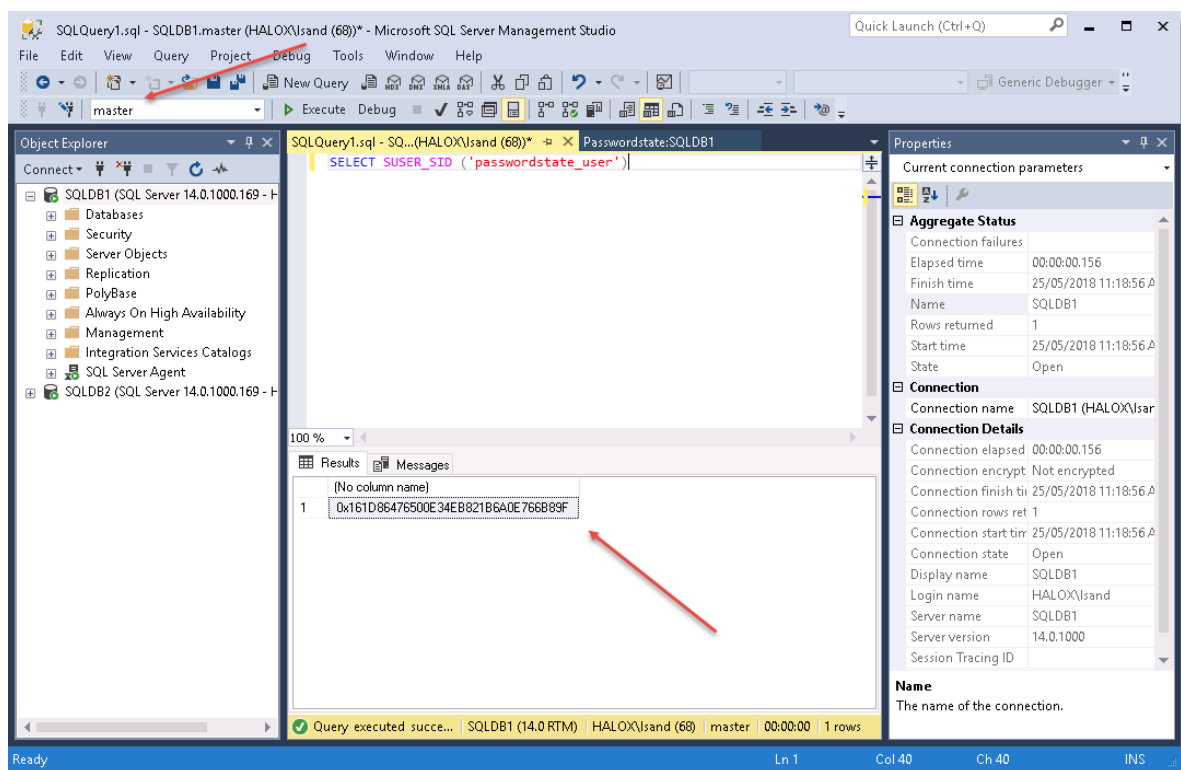
## 8 Configuring Passwordstate to work with Always On Group

The **passwordstate\_user** account is used to connect the Passwordstate website to the database. To ensure Passwordstate can connect correctly, we need to ensure the passwordstate\_user account has the same **SID** on both database servers. This process below will need to be completed to ensure if the databases failover, then the Passwordstate website will still be accessible.

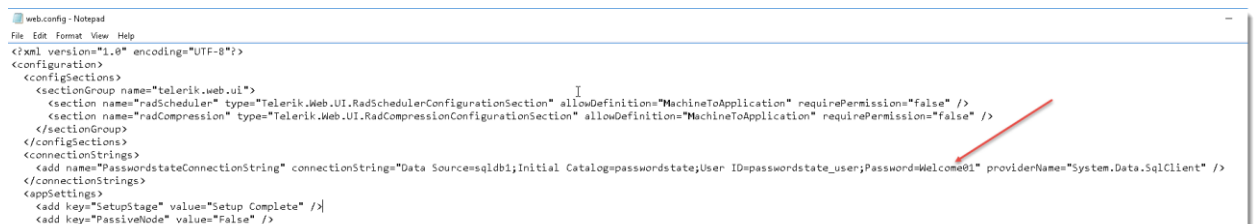
- When connected to your **SQLDB1** server, run the following query in **SQL Management Studio Tools**:

```
SELECT SUSER_SID ('passwordstate_user')
```

- Take note of the output and copy it into clipboard



- On your Passwordstate web server open Notepad "As Administrator", open the **c:\inetpub\Passwordstate\web.config** file, and take note of the password being used in the connection string



- Next connect to **SQLDB2** and run the following command in **SQL Management Studio Tools**, which will create the passwordstate\_user account on this server with the same **SID**. Ensure you insert the correct SID into this statement that you discovered above. Also ensure the password is set correctly:

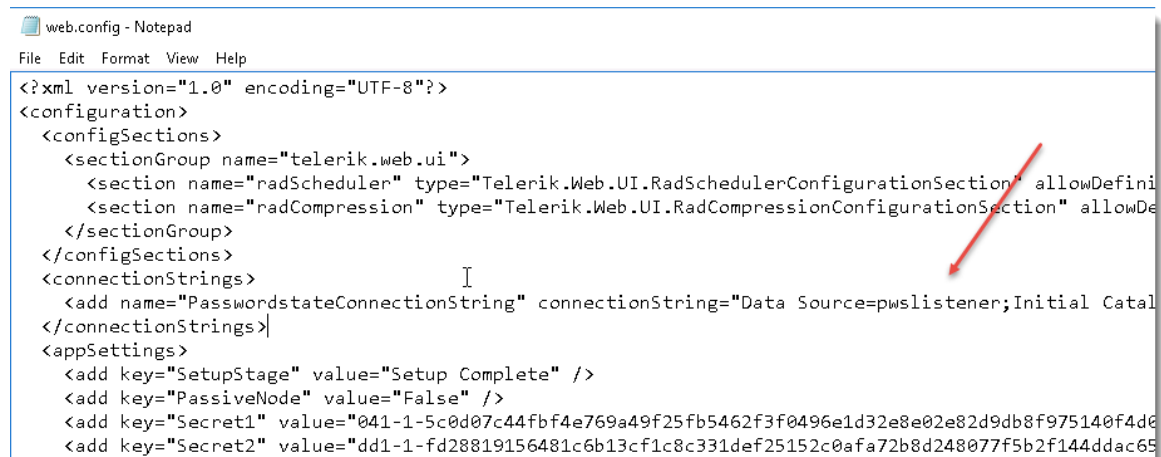
```
USE [master]
```

```
GO
```

```
CREATE LOGIN passwordstate_user WITH PASSWORD=N'Welcome01',
SID=0x161D86476500E34EB821B6A0E766B89F, DEFAULT_DATABASE=passwordstate,
DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
```

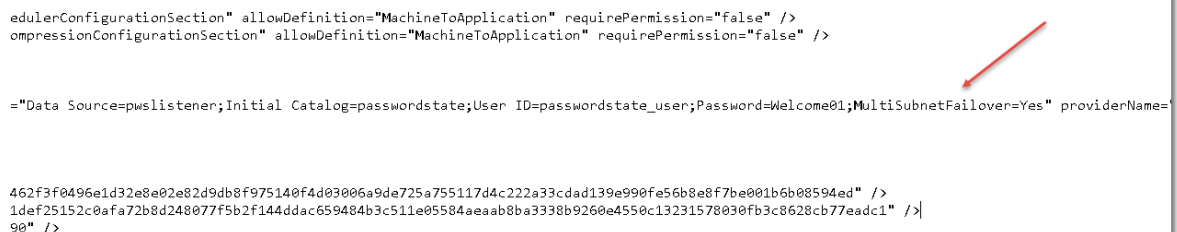
```
GO
```

- Back on your Passwordstate webserver, change the Data Source to be your new listener instead of SQLDB1



```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=pwslister;Initial Catalog=..." />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="PassiveNode" value="False" />
    <add key="Secret1" value="041-1-5c0d07c44fbf4e769a49f25fb5462f3f0496e1d32e8e02e82d9db8f975140f4d6..." />
    <add key="Secret2" value="dd1-1-fd28819156481c6b13cf1c8c331def25152c0afa72b8d248077f5b2f144ddac65..." />
  </appSettings>
</configuration>
```

- If your database servers are located on different subnets, you will also need to insert **;MultiSubnetFailover=Yes** into your connection string as per below screenshot:



```

...
<add name="PasswordstateConnectionString" connectionString="Data Source=pwslister;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=Welcome01;MultiSubnetFailover=Yes" providerName="Microsoft.SqlServer.jdbc" />
...

```

- Save your web.config file and the install is now complete

If your Primary SQL Server is unavailable for any reason, then it will automatically failover to the second server. The time it takes to failover will depend on network bandwidth and database sizes, but for a small database it is only a few seconds.