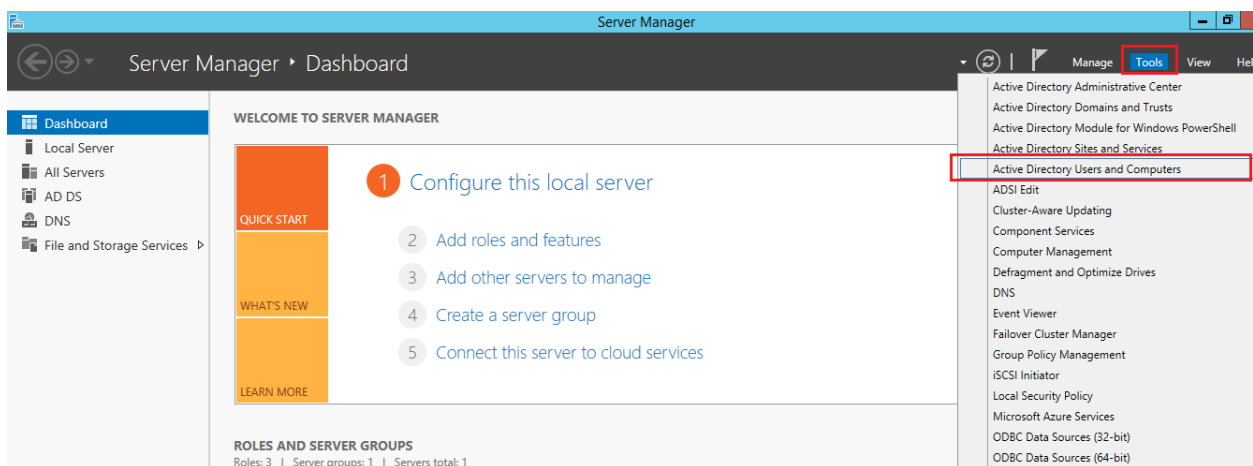


Configuring a SQL Server AlwaysOn High Availability Group

Configuring a SQL Server AlwaysOn High Availability Group

In the following steps, we will create a Service account to connect SQL server instances. The Service account will be used to connect the SQL instance from both the node servers.

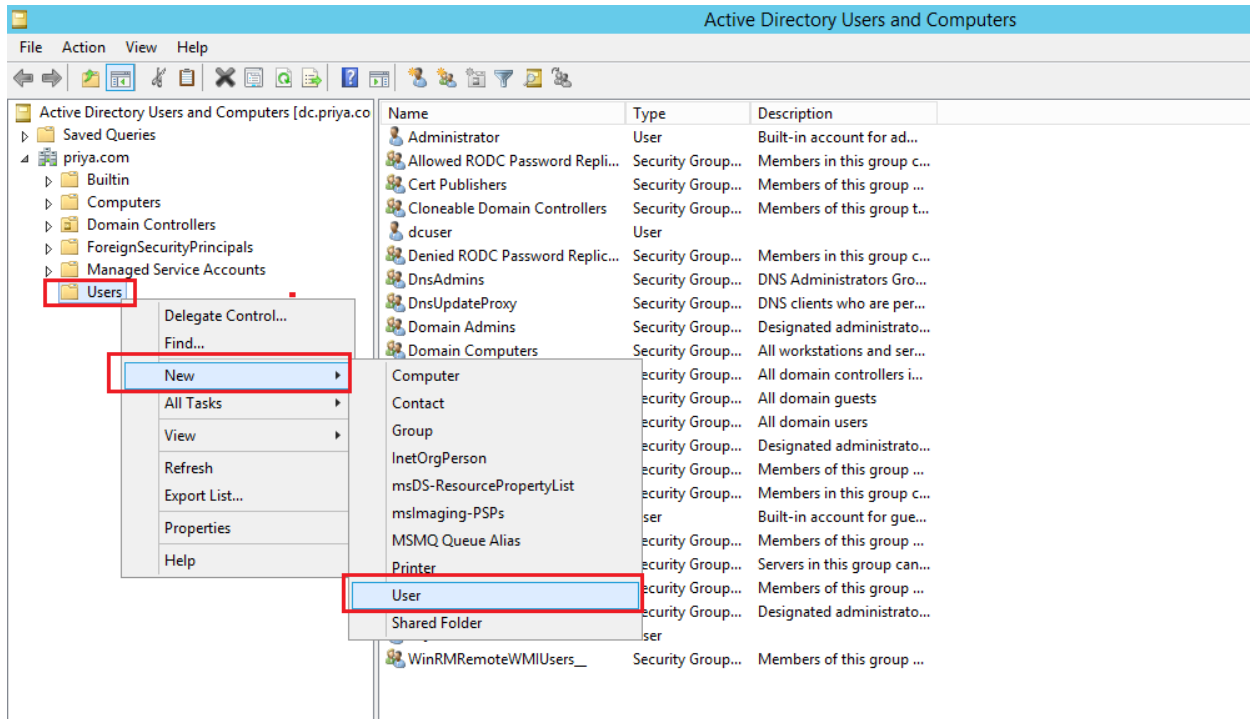


Create a Service Account

We will log in to the Domain controller and perform the following steps.

Open **Server Manager**, **Tools** and Click **Active Directory Users and Computers**.

In Active Directory Users and Computers window, **Right-Click Users > New > User** as shown in the following Fig.



In this step, we will create a Service account username, Specify the service account login name and Click Next

Create in: priya.com/Users

First name: SQL Initials:

Last name:

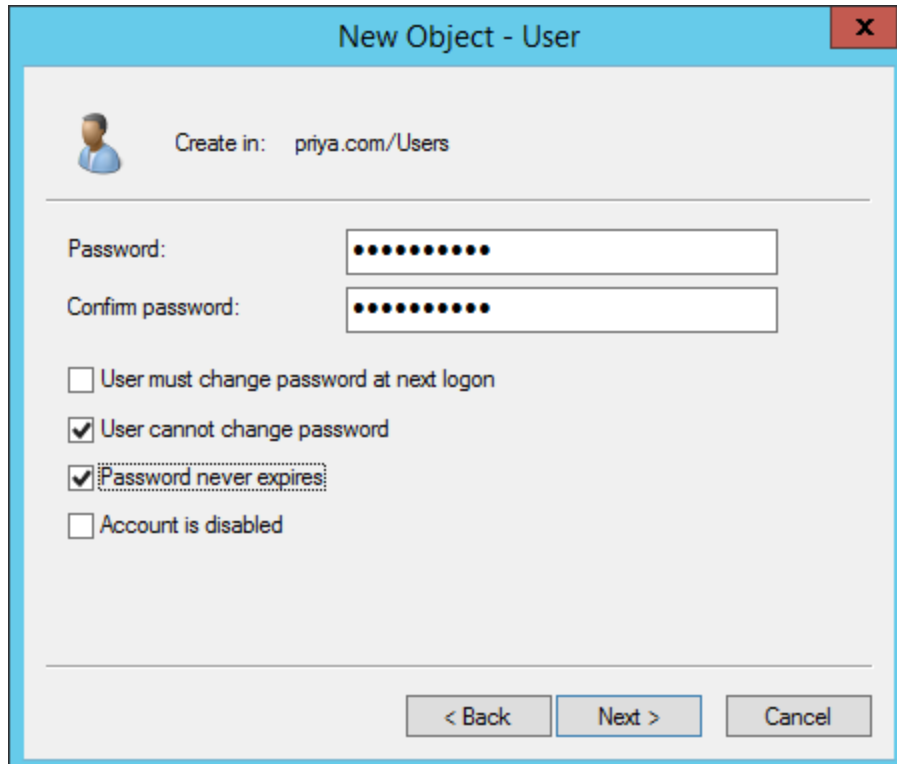
Full name: SQL

User logon name: svcaccount @priya.com

User logon name (pre-Windows 2000): PRIYA\ svcaccount

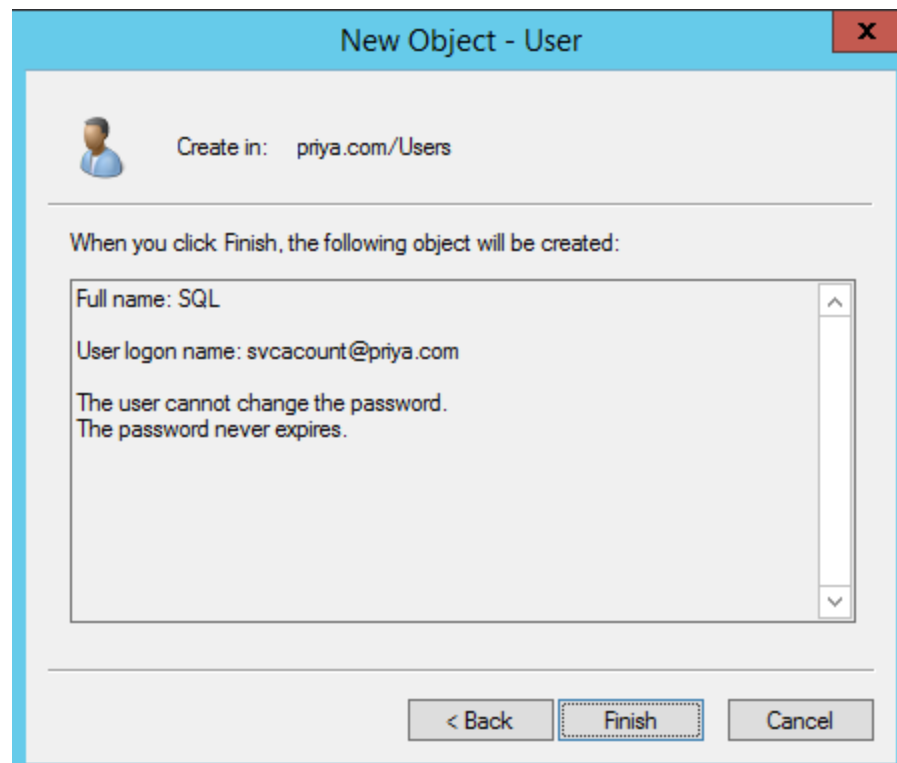
< Back Next > Cancel

Specify a Password and tick the options for **User cannot change password** and **Password never expires**. We need to select **Password never expires** option because, if the password expired, the service account login would fail, and it will cause a problem to the high availability configuration.



The dialog box is titled "New Object - User" with a close button (X) in the top right corner. It features a user icon and the text "Create in: priya.com/Users". Below this, there are two password input fields labeled "Password:" and "Confirm password:", both containing masked characters (dots). Underneath the password fields are four checkboxes: "User must change password at next logon" (unchecked), "User cannot change password" (checked), "Password never expires" (checked), and "Account is disabled" (unchecked). At the bottom, there are three buttons: "< Back", "Next >", and "Cancel".

Click **Finish** to complete the service account creation process.



This dialog box, also titled "New Object - User" with a close button (X), shows the summary of the user creation. It includes the same user icon and "Create in: priya.com/Users" text. Below this, it states "When you click Finish, the following object will be created:". A scrollable text area displays the following details: "Full name: SQL", "User logon name: svcaccount@priya.com", "The user cannot change the password.", and "The password never expires.". At the bottom, there are three buttons: "< Back", "Finish", and "Cancel".

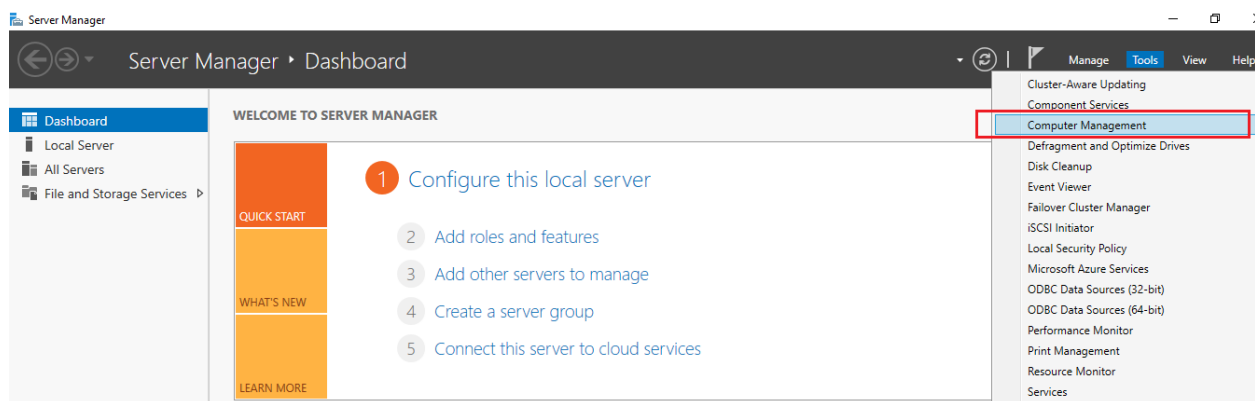
Configure Administrator permission, windows user and login for the service account

Once the Service account created on the Domain controller, we need to add service account as a member of the administrator account on both nodes.

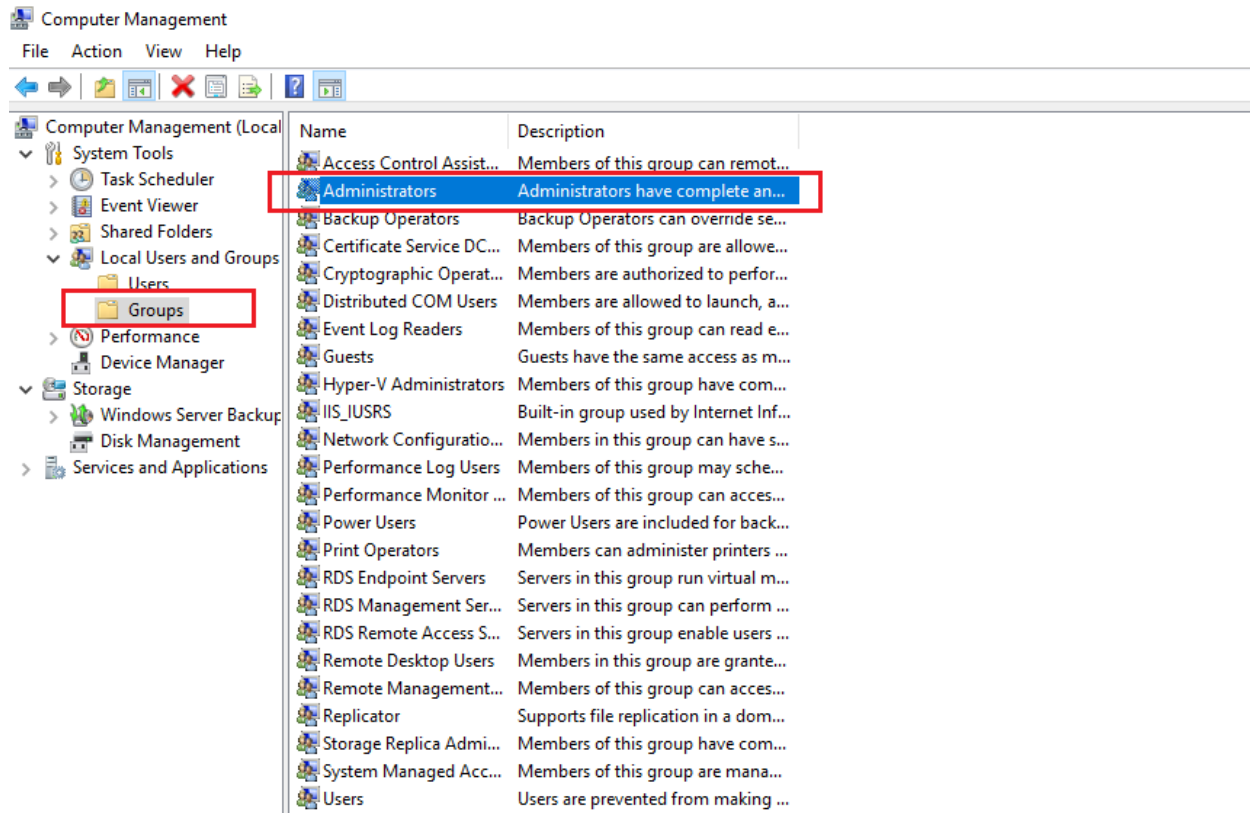
Connect to both node servers SQL1 and SQL2 and perform below steps

- Add Service account as an Administrator
- Create a Windows user for the Service account
- Create a Windows authentication login for a service account login in the SQL database instance

Open **Server manager** and Select **Tools** and **Computer Management**

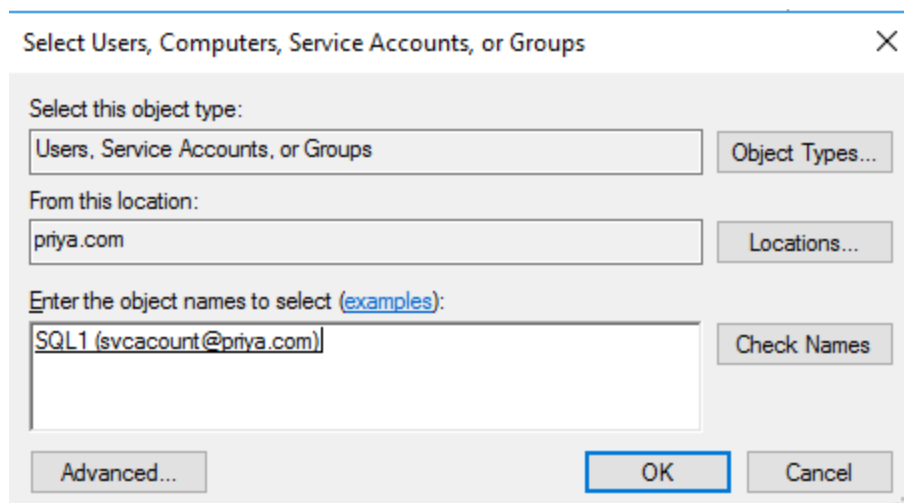


In the **Computer management window**, Select **Groups** and Right-Click **Administrators** and Go to **Properties**



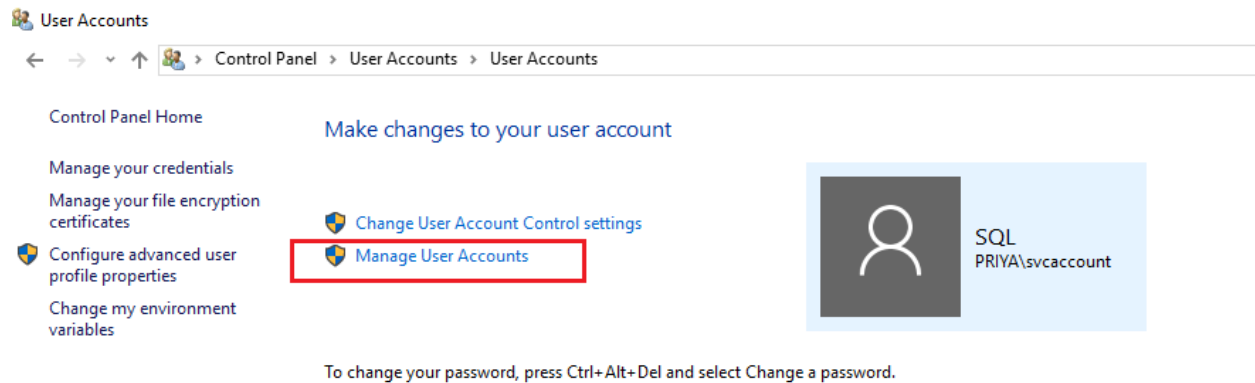
Add User, Enter the Service account login name and Check names and Click **OK**

Note – Once clicked on the Check Names, it may ask to enter username and password pop-up window. In the pop-up window, we need to register a domain controller username and password and make sure the location should select as a directory.

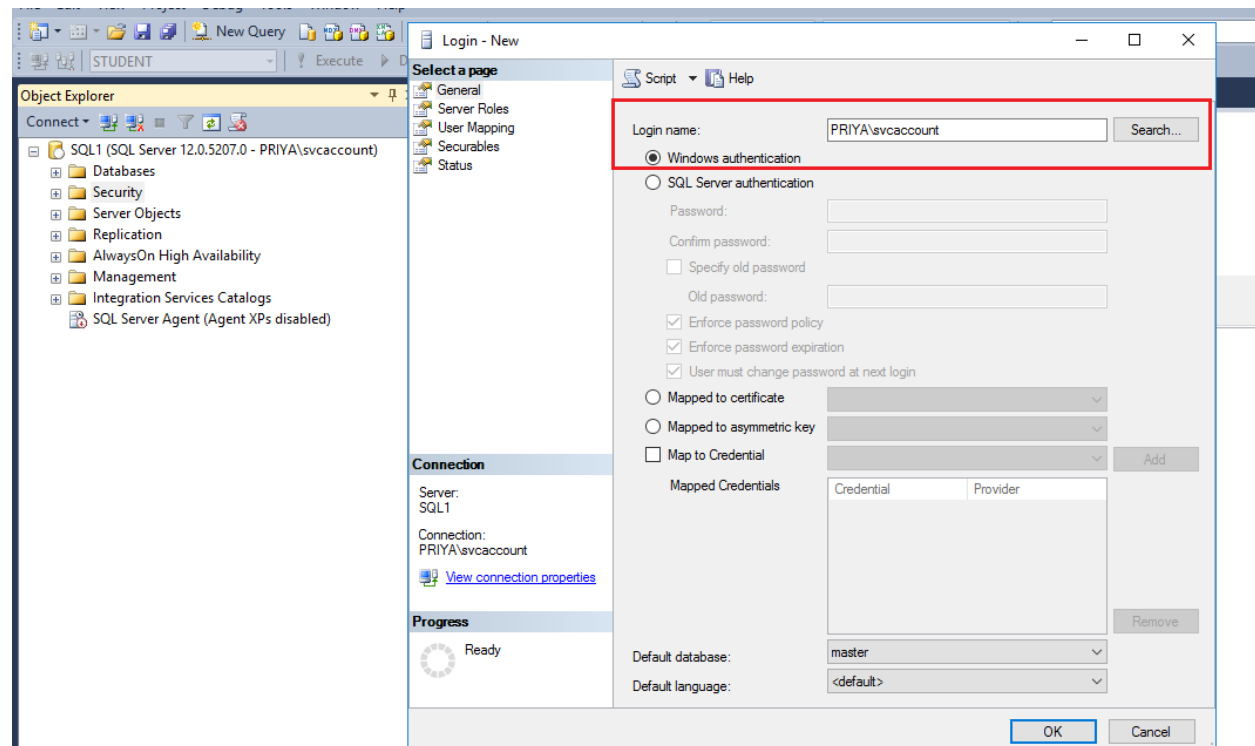


In the following Fig. We will create the Service account as a windows user in **Control Panel > User Account > Click on Manage User Accounts.**

Enter a **Service account name** and **domain name** and Click **Next**, Select **Administrator** and Finish

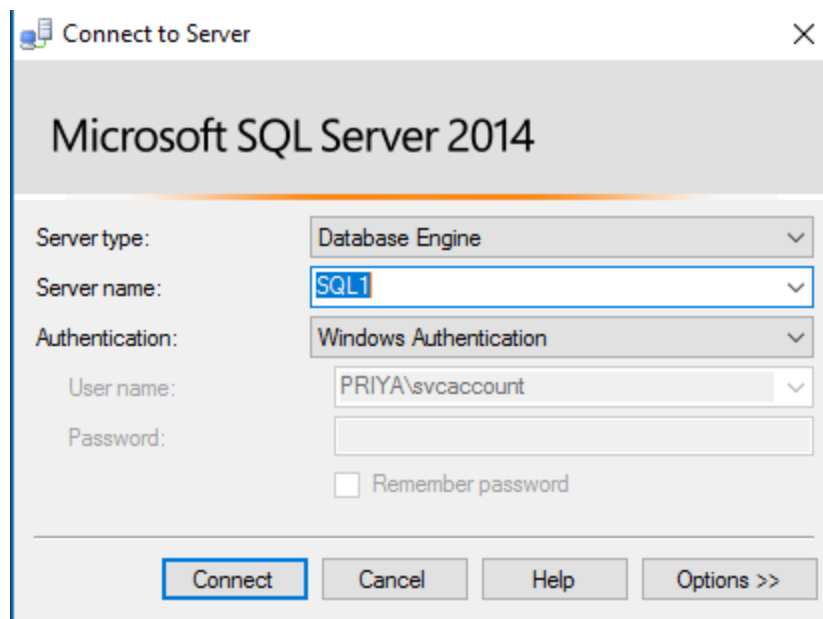


Next, we will create a windows authentication login for the service account as shown in the following fig.



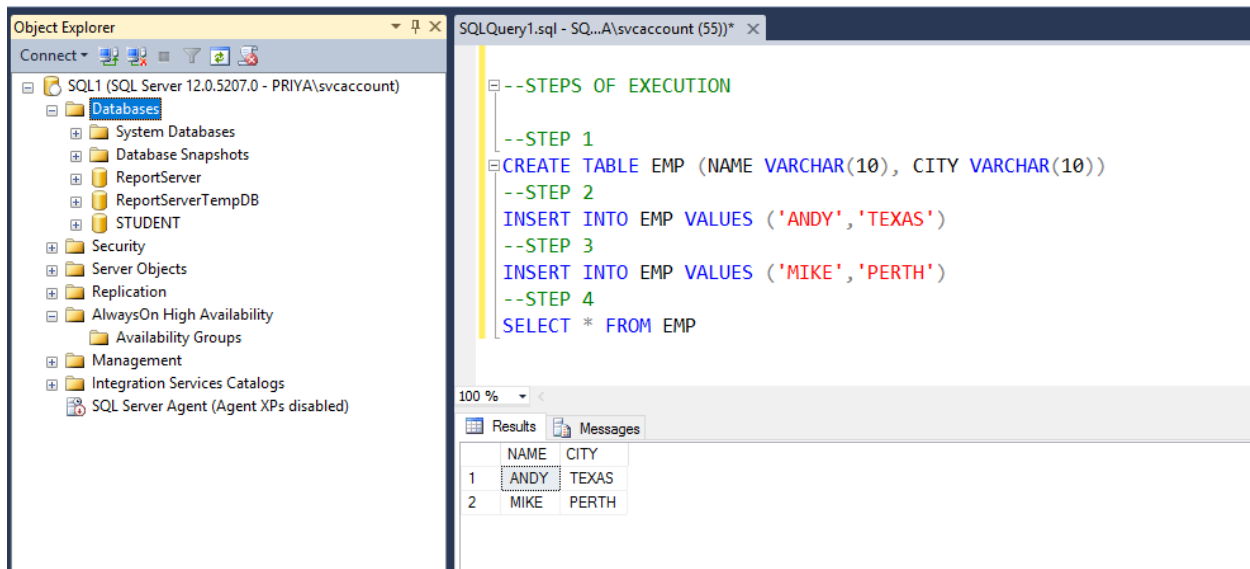
Take remote desktop connection of SQL1 with the service account windows user, connect to the SQL Server database engine instance SQL1 with default windows authentication login as shown in the following fig.

Enable AlwaysOn Feature



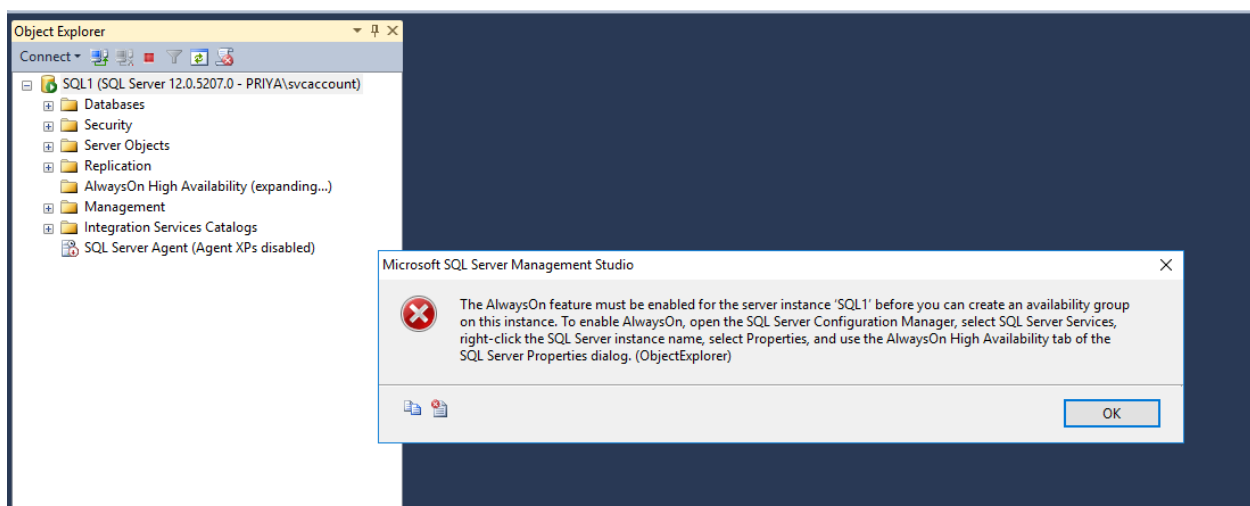
Once SQL Server instance is connected, we need to decide database which will take part in SQL Server AlwaysOn High Availability.

In this demonstration, we will create a new database **STUDENT**, and then we will create a table and add a few records into it as shown in the following fig.



Next, Expand SQL Server AlwaysOn High Availability option, we will get an error while expanding the AlwaysOn option

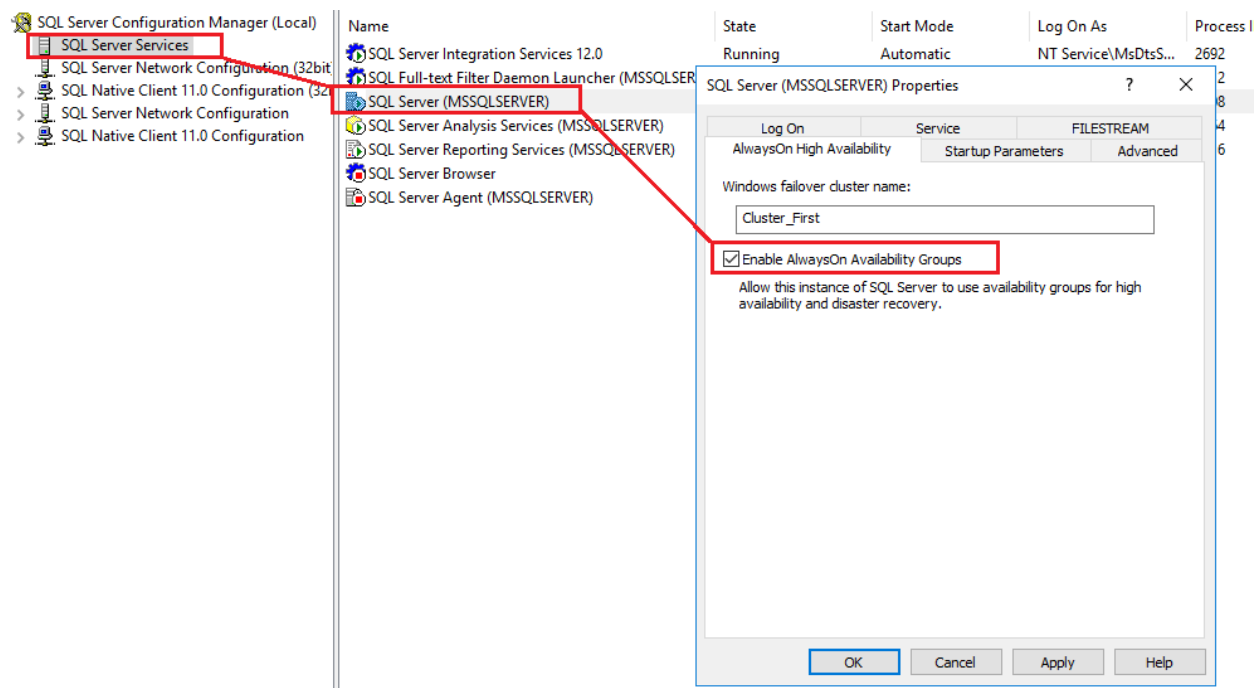
The AlwaysOn feature must be enabled for the server instance 'SQL1' before you can create an availability group on this instance. To enable AlwaysOn, open the SQL Server Configuration Manager, select SQL Server Services, right-click the SQL Server instance name, select Properties, and use the AlwaysOn High Availability tab of the SQL Server Properties dialog. (ObjectExplorer)



To enable AlwaysOn feature, **Open SQL Server Configuration Manager, Right-click SQL Server instance and go to properties, Select AlwaysOn High**

Availability tab section and tick checkbox for **Enable AlwaysOn AvailabilityGroups**

Note: – We also need to make sure nodes where we will enable AlwaysOn feature is part of Windows Failover Cluster.

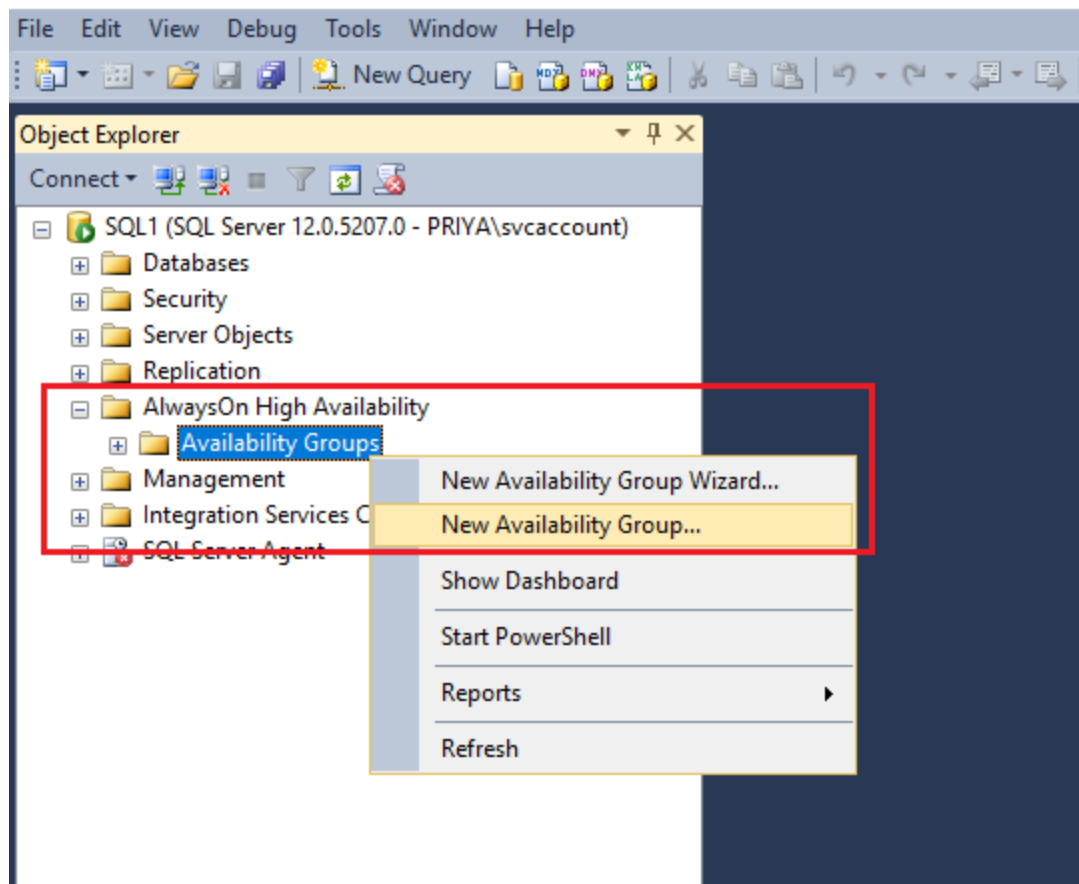


Once we enabled, Apply and Click **OK**. Restart the SQL Server services.

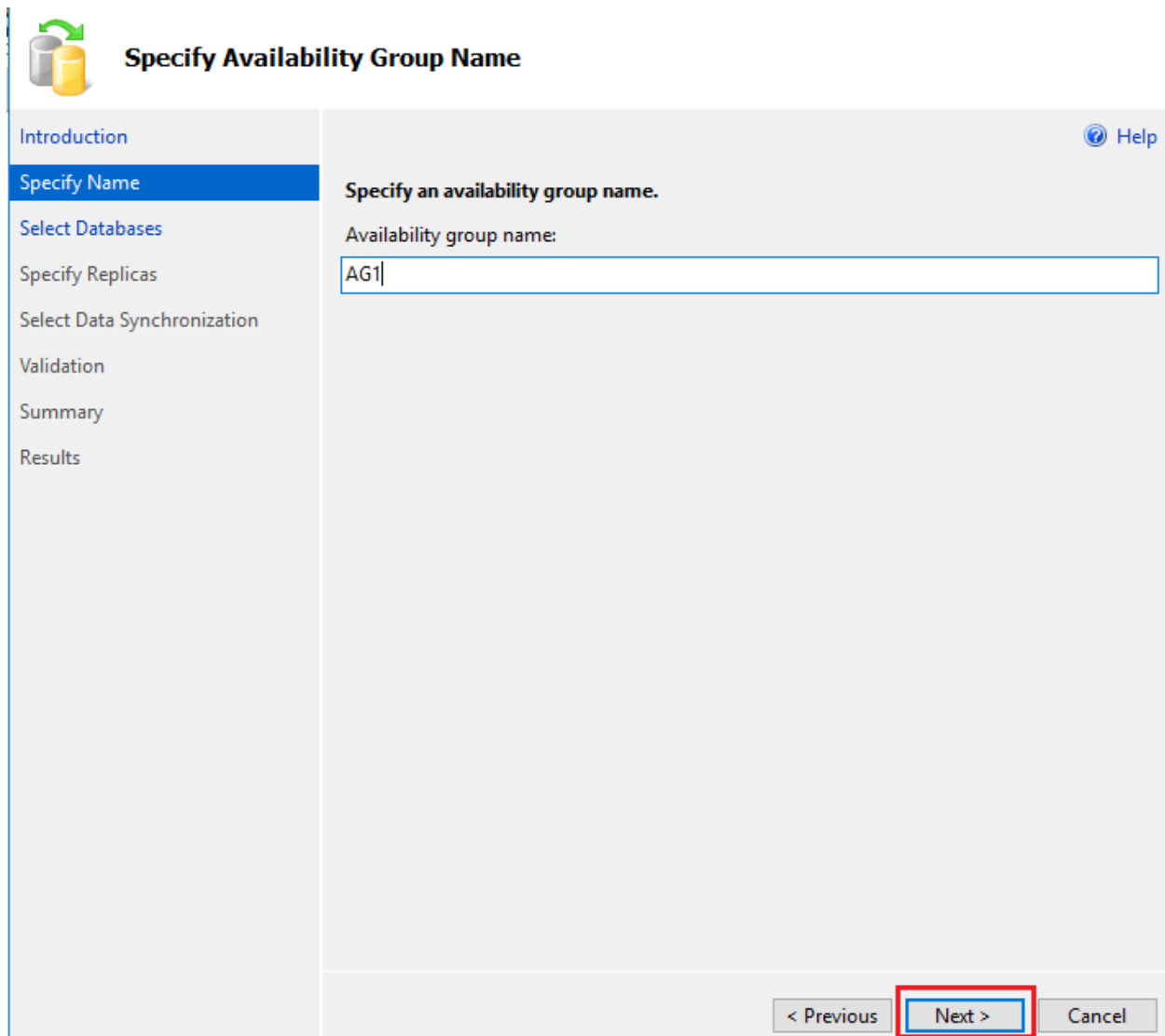
As shown in the following fig. we will able to use SQL Server AlwaysOn High Availability feature. Right-click **Availability Groups** and Click **New Availability Group**.

Configuring Availability Group

An AlwaysOn Availability Group contains a set of user databases; we need to select user databases which will be part of the Availability group. These databases called as an Availability database.




Specify **Availability Group name** and click **Next**




The image shows a SQL Server Enterprise Manager wizard titled "Specify Availability Group Name". The wizard has a left-hand pane with a list of steps: Introduction, Specify Name (which is the current step and highlighted in blue), Select Databases, Specify Replicas, Select Data Synchronization, Validation, Summary, and Results. The main area of the wizard is titled "Specify an availability group name." and contains a text box labeled "Availability group name:" with the text "AG1" entered. A "Help" icon is in the top right corner. At the bottom right, there are three buttons: "< Previous", "Next >" (which is highlighted with a red rectangle), and "Cancel".

As shown in fig. we will use database **STUDENT**. To proceed further, we need to perform one essential pre-requisite step, a Full backup of the database. So, take a Full backup of STUDENT and click **Next**



Select Databases

[Introduction](#)
[Specify Name](#)
Select Databases
[Specify Replicas](#)
[Select Data Synchronization](#)
[Validation](#)
[Summary](#)
[Results](#)


[Help](#)

Select user databases for the availability group.

User databases on this instance of SQL Server:

	Name	Size	Status
<input type="checkbox"/>	ReportServer	15.3 MB	Full backup is required
<input type="checkbox"/>	ReportServerTempDB	5.3 MB	Full recovery mode is required
<input checked="" type="checkbox"/>	STUDENT	5.0 MB	Meets prerequisites

Refresh

< Previous

Next >

Cancel

Next, **Specify Replica** step has four sections. **Replica, Endpoint, Backup Preferences, and Listener**

Replica– Replica is a server. There are one primary replica and multiple replicas. In SQL server 2012, It supports up to 4 secondary replicas, while in SQL Server 2014, it supports up to 8 replicas.

The primary replica is primary source server or production server. A secondary replica is a server which maintain a backup copy of the primary server availability database. On the Primary replica, it allows to perform Read and write operations while on the secondary replica only read operations.

In this case, SQL1 is our primary replica and SQL2 is a secondary replica. So, we will add SQL2 as a secondary replica as shown in the following fig.

Initial Role – It specifies the role of replica whether Primary or secondary.

Automatic Failover – It failovers functional role from primary replica to secondary replica by an automated method with no data loss.

Synchronous Commit – In the Synchronous-commit mode, failover can happen by either Automatic or manual way with no data loss. So, in this case, we will use synchronous commit with automatic failover.

The screenshot shows the 'Specify Replicas' step of the 'New Availability Group' wizard. The left sidebar contains a list of steps: Introduction, Specify Name, Select Databases, Specify Replicas (highlighted), Select Data Synchronization, Validation, Summary, and Results. The main area is titled 'Specify an instance of SQL Server to host a secondary replica.' and has tabs for Replicas, Endpoints, Backup Preferences, and Listener. The 'Replicas' tab is active, showing a table of 'Availability Replicas'.

Server Instance	Initial Role	Automatic Failover (Up to 2)	Synchronous Commit (Up to 3)	Readable Secondary
SQL1	Primary	<input type="checkbox"/>	<input type="checkbox"/>	No

Below the table are three buttons: 'Add Replica...' (highlighted with a red box), 'Add Azure Replica...', and 'Remove Replica'. Below these buttons is a 'Summary for the replica hosted by SQL1' section.

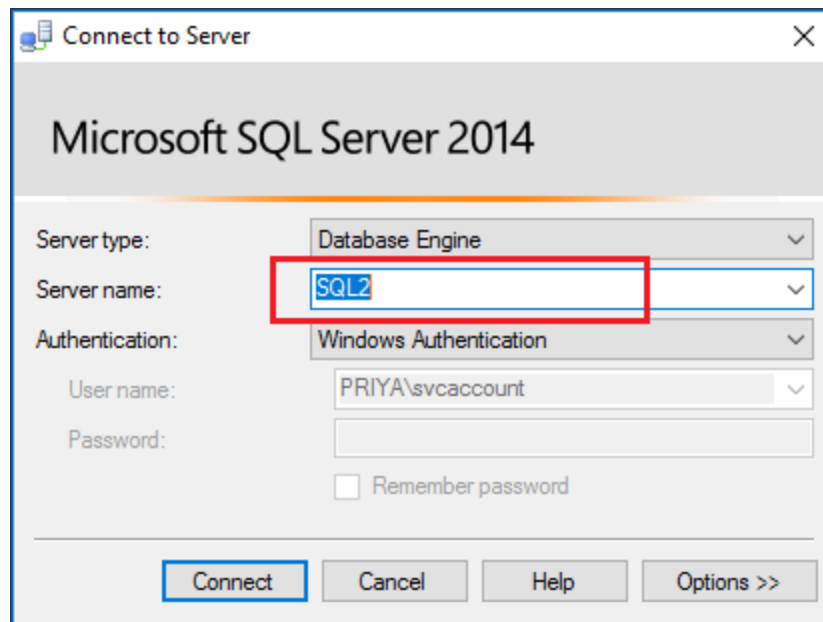
Summary for the replica hosted by SQL1

Replica mode: Asynchronous commit
This replica will use asynchronous-commit availability mode and support only forced failover (with possible data loss).

Readable secondary: No
In the secondary role, this availability replica will not allow any connections.

At the bottom right, there are three buttons: '< Previous' (highlighted with a blue box), 'Next >', and 'Cancel'.

Once clicked **Add Replica**, connect to the server, Select SQL2 and Connect



Once we add secondary server SQL2 as a secondary replica, the primary role of SQL2 is secondary.

New Availability Group

Specify Replicas

Introduction
Specify Name
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 2)	Synchronous Commit (Up to 2)	Readable Secondary
SQL1	Primary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No
SQL2	Secondary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Yes

< >

Add Replica... Add Azure Replica... Remove Replica


Summary for the replica hosted by SQL2

Replica mode: Synchronous commit with automatic failover
This replica will use synchronous-commit availability mode and 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

< Previous Next > Cancel

In the Endpoint section, it will show configured endpoint URL, port, endpoint name, and the service account name for SQL1 and SQL2 nodes.



Specify Replicas

[Introduction](#)
[Specify Name](#)
[Select Databases](#)
[Specify Replicas](#)
[Select Data Synchronization](#)
[Validation](#)
[Summary](#)
[Results](#)

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

Replicas | **Endpoints** | Backup Preferences | Listener

Endpoint values:

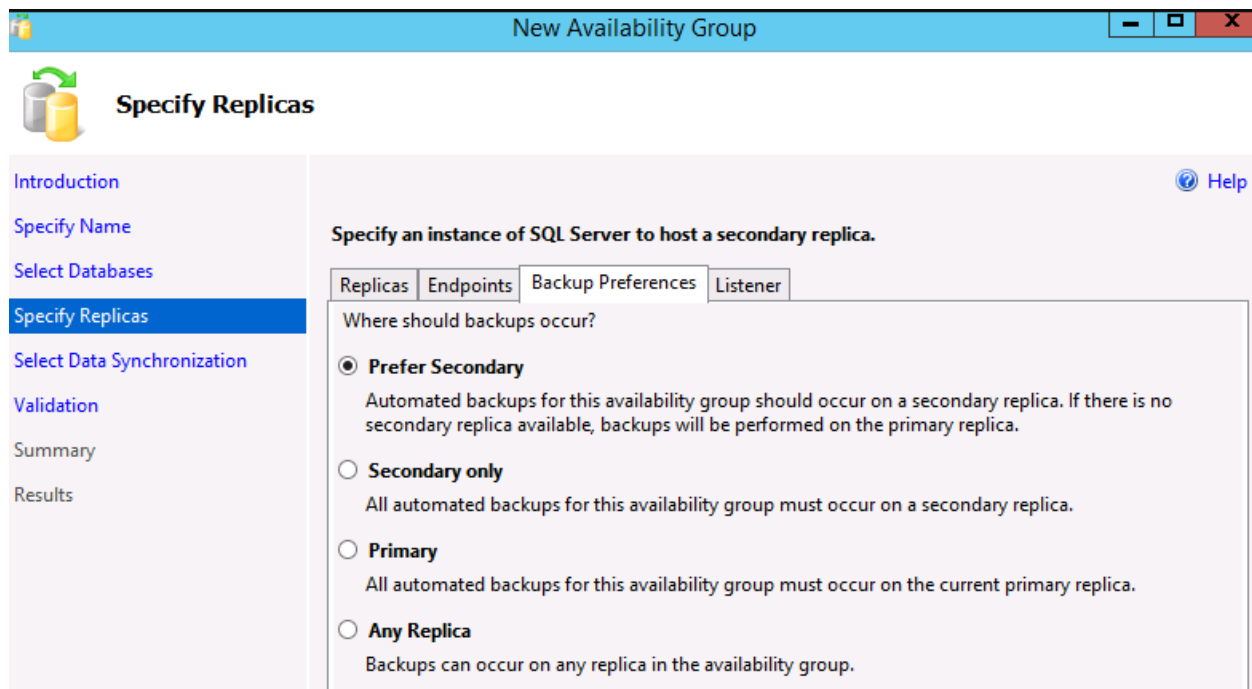
Server Name	Endpoint URL	Port Number	Endpoint Name	Encrypt Data	SQL Server Service Account
SQL1	TCP://sql1.priya.com:5022	5022	Hadr_endpoint	<input checked="" type="checkbox"/>	PRIYA\svcaccount
SQL2	TCP://sql2.priya.com:5022	5022	Hadr_endpoint	<input checked="" type="checkbox"/>	PRIYA\svcaccount

Status Refresh

Backup Preferences – Backup preferences indicate the backup location of the **availability group**. It provides four backup options.

- **Prefer Secondary** – Automatic backup will occur on a secondary replica
- **Secondary only** – Automatic backup must occur on a secondary replica
- **Primary** – Automatic backup must occur on a primary replica
- **Any Replica** – Backup will occur on any replica

On the secondary replica, we can only take a log backup, and Copy-only backup, Differential, and Full back up are not allowed on a secondary replica. While on the primary replica, we can perform Full, log, and Differential backups.



Listener– Listener is a server through which nodes get to communicate with each other. The Listener has all information about availability group. We need to select **Create an Availability group listener** option and specify Name, Port and static IP as shown in the following fig.

usual port is 1433



Specify Replicas

Introduction

Specify Name

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

☐ 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: AGLName

Port: 16333

Network Mode: Static IP

Subnet	IP Address
10.0.0.0/8	10.0.0.4

< Previous

Next >

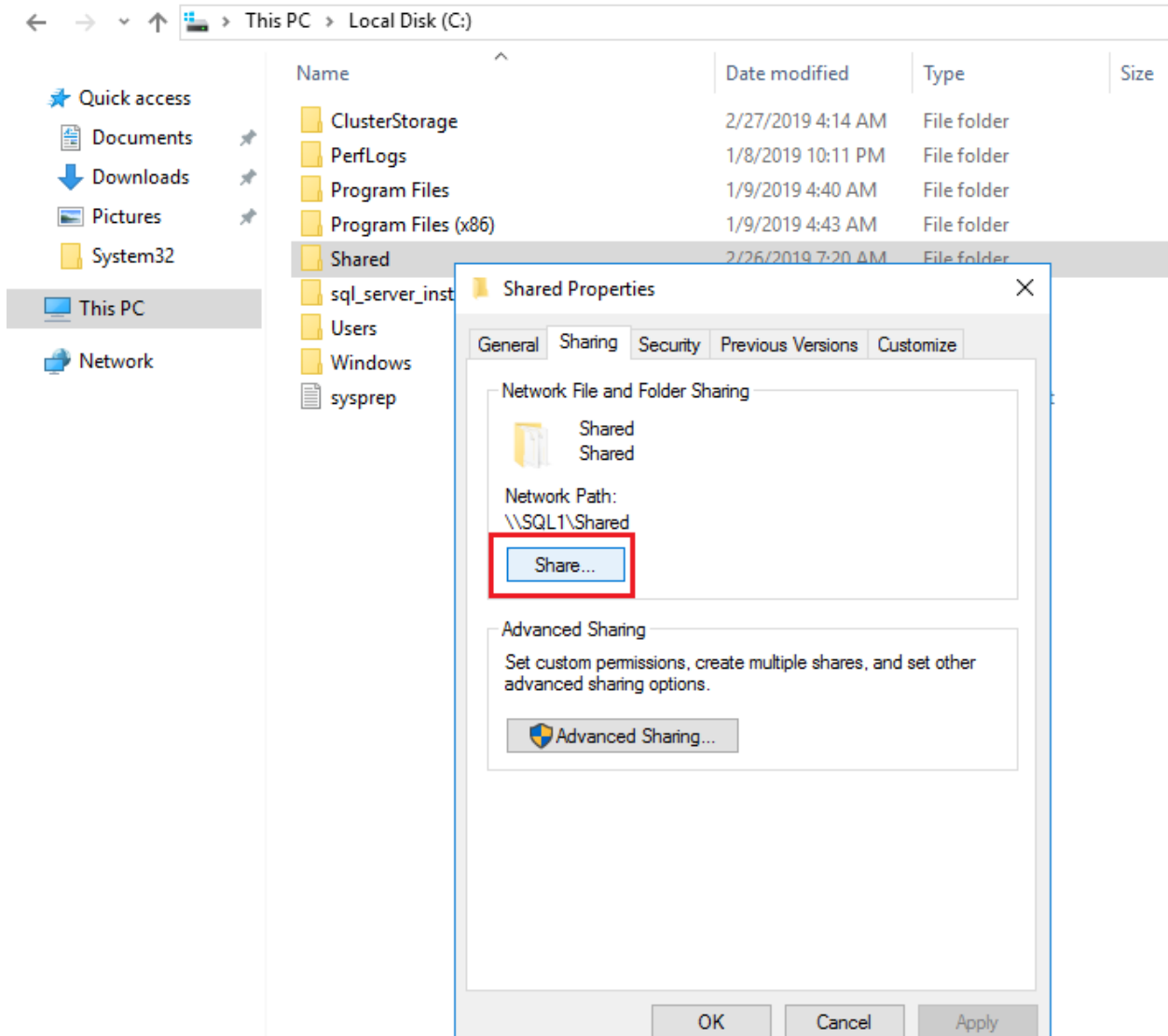
Cancel

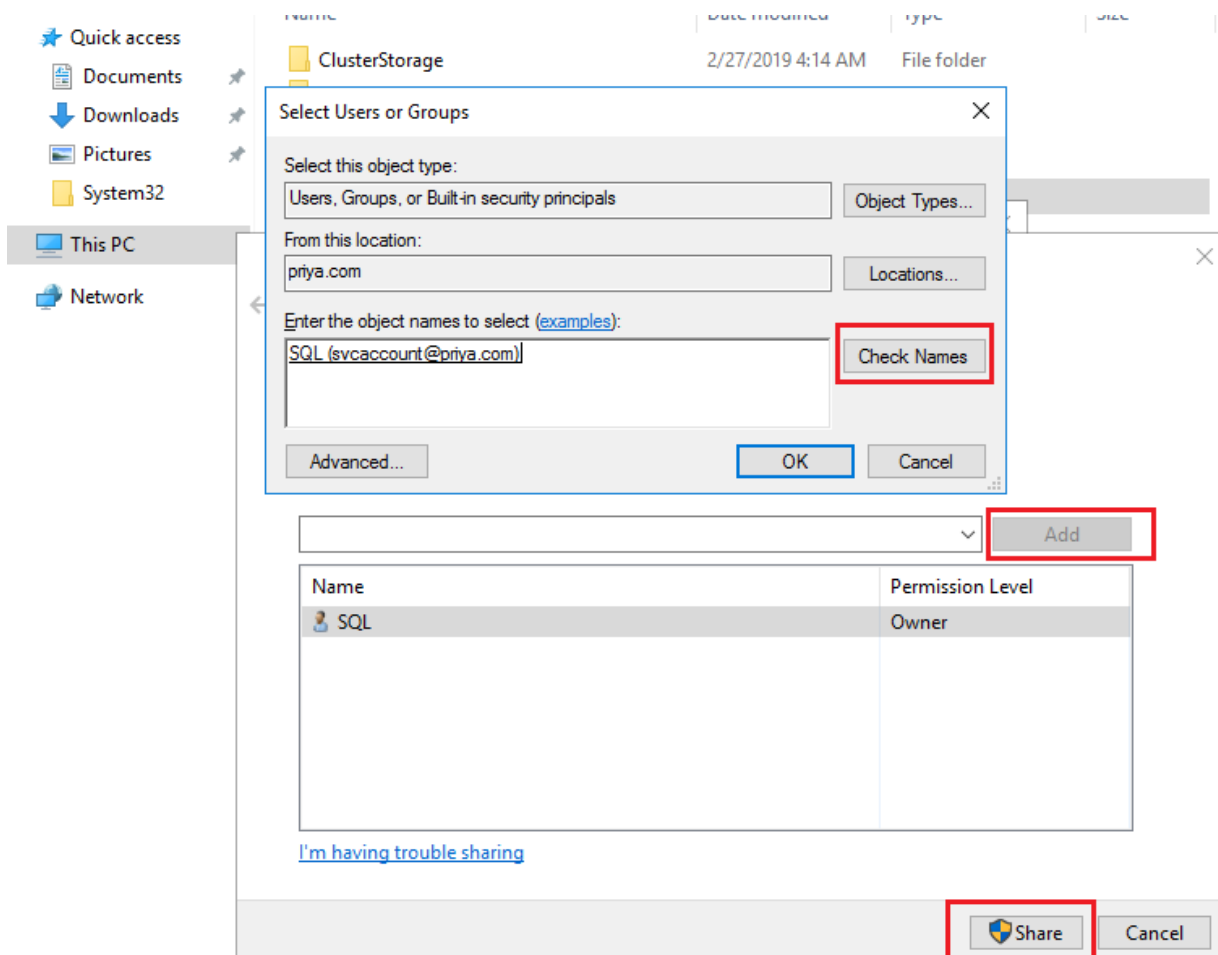
In **Data Synchronization step**, Select **Full** where it performs Full and Log backup and puts it into shared location path.

Create a Shared location

Connect to node SQL1 and choose desired drive location and folder, **Right-click folder** and **Go to Properties, Select Security** and click **Share**. It will show a pop-up window to add a user to whom, share permission need to grant.

Provide the service account name and Click **Add**; it will show pop-up window to check names, once service account name found in the domain name, click **Share**.







Select Initial Data Synchronization

[Introduction](#)
[Specify Name](#)
[Select Databases](#)
[Specify Replicas](#)
Select Data Synchronization
[Validation](#)
[Summary](#)
[Results](#)

[Help](#)

Select your data synchronization preference.

☒ **Full**
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.

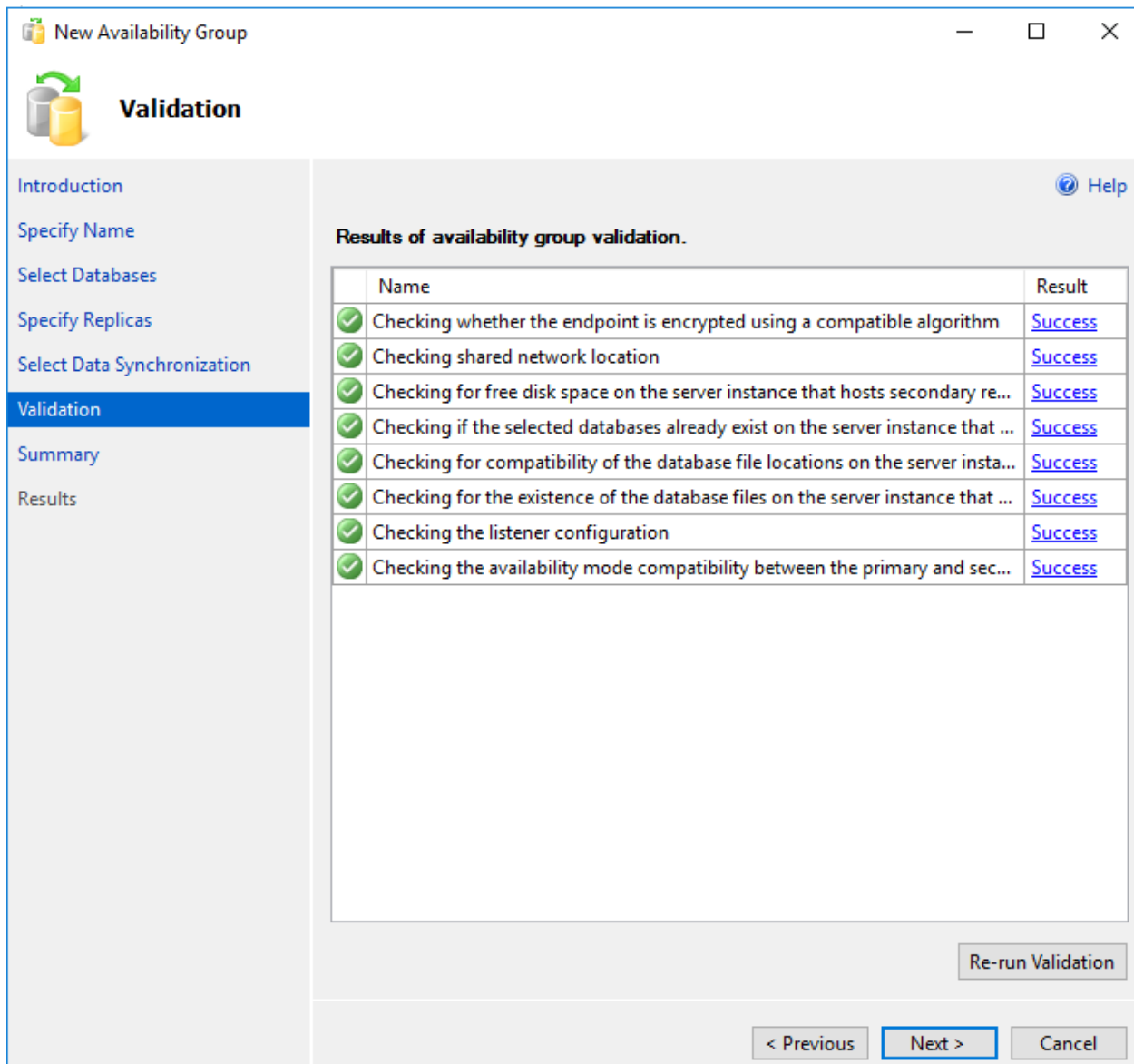
Specify a shared network location accessible by all replicas:
 [Browse...](#)

☐ **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. This action will be skipped for Azure replicas.

☐ **Skip initial data synchronization**
Choose this option if you want to perform your own database and log backups of each primary database.

[< Previous](#) [Next >](#) [Cancel](#)

In the **Validation step**, SQL Server performs validations against any required actions, and It will provide warning or error messages for missing required steps.



It performed all validation steps successfully, and wizard operation completed as shown in the following fig.



Results

Introduction

Specify Name

Select Databases

Specify Replicas

Select Data Synchronization

Validation

Summary

Results

Help



The wizard completed successfully.

Summary:

Name	Result
Configuring endpoints.	Success
Starting the 'AlwaysOn_health' XEvent session on 'SQL1'.	Success
Starting the 'AlwaysOn_health' XEvent session on 'SQL2'.	Success
Creating availability group 'AG1'.	Success
Waiting for availability group 'AG1' to come online.	Success
Joining secondary replicas to availability group 'AG1'.	Success
Validating WSFC quorum vote configuration.	Success
Creating a full backup for 'STUDENT'.	Success
Restoring 'STUDENT' on 'SQL2'.	Success
Backing up log for 'STUDENT'.	Success
Restoring 'STUDENT' log on 'SQL2'.	Success
Joining 'STUDENT' to availability group 'AG1' on 'SQL2'.	Success

< Previous

Next >

Close

After successful completion, we will verify the database at secondary replica SQL2. As we can observe, Database STUDENT is in Synchronizing state, and it is in sync with the primary replica, and we can read data successfully.

