Configuring SQL Server AlwaysOn Availability Groups

If you run multiple SQL Server Enterprise instances on Compute Engine, you can configure those instances to use Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups to support high availability and disaster recovery.

This tutorial shows you how to create a cluster of instances with SQL Server Enterprise with the necessary network settings and configure those instances to use Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups. You can use SQL Server Enterprise instances created from either the SQL Server public images (/compute/docs/instances/windows#sql_server) or Microsoft License Mobility (/compute/docs/instances/windows/ms-licensing).

Prerequisites

This tutorial assumes that you understand the following items:

- You are familiar with Compute Engine VPC networks (/vpc/docs) and firewall rules (/vpc/docs/firewalls).
- You are familiar with creating Windows instances (/compute/docs/instances/windows/creating-managing-windows-instances) on Compute Engine.
- You are familiar with setting up Active Directory (/compute/docs/tutorials/setup-active-directory) on Compute Engine.

Setting up the VPC network

Create a custom mode VPC network with three subnets. Then, set the firewall rules to allow traffic between internal virtual machines (VMs). You can use an existing network for this task if necessary, but we recommend you isolate systems into different networks and subnets. This tutorial expects that you create the following VPC network and subnet configuration:

1. Create a custom mode VPC network.

```
gcloud compute networks create wsfcnet --subnet-mode custom
```

2. Add three subnets to the VPC network.

```
gcloud compute networks subnets create wsfcsubnet1 --network wsfcnet \
  --region us-central1 --range 10.0.0.0/24
```

```
gcloud compute networks subnets create wsfcsubnet2 --network wsfcnet \
  --region us-central1 --range 10.1.0.0/24
```

```
gcloud compute networks subnets create wsfcsubnet3 --network wsfcnet \
  --region us-central1 --range 10.2.0.0/24
```

3. Create a firewall rule to allow traffic between the instances on internal IP addresses on the new VPC network.

```
gcloud compute firewall-rules create allow-internal-ports \
  --network wsfcnet --allow tcp:1-65535,udp:1-65535,icmp \
  --source-ranges 10.0.0.0/24,10.1.0.0/24,10.2.0.0/24
```

4. Create a firewall rule to allow RDP on port `3389` on the VPC network.

```
gcloud compute firewall-rules create allow-rdp --network wsfcnet \
  --allow tcp:3389 --source-ranges 0.0.0.0/0
```

Creating a Windows domain controller

Create a Windows domain controller. For this tutorial, the domain is `dbeng.com` and the name of the domain controller is `dc-windows` at IP address `10.2.0.100`. The domain controller is using the `wsfcsubnet3` subnet.

1. Create an instance to use as a domain controller. For this tutorial, specify a small `n1-standard-2` machine type and the latest image from the `windows-2016` image family.

   ```
   gcloud compute instances create dc-windows --machine-type n1-standard-2 \
     --boot-disk-type pd-ssd --image-project windows-cloud \
     --image-family windows-2016 --boot-disk-size 200GB \
     --zone us-central1-f --subnet wsfcsubnet3 --private-network-ip=10.2.0.100
   ```

2. Generate a password (/compute/docs/instances/windows/creating-passwords-for-windows-instances#generating_a_password) so that you can connect to the domain controller VM by using a local account. Note the username and password for future use.

3. Using RDP (/compute/docs/instances/connecting-to-instance#windows), connect to the domain controller VM with your local account username and password.

4. On the instance, run PowerShell as an administrator to open the PowerShell terminal.

5. Set up an Administrator user.

   a. Run the following command, then enter a password for use with the Administrator account.

      ```
      PS C:\> $Password = Read-Host -AsSecureString
      ```

      Note the password that you entered.

   b. Set the Administrator account password.

      ```
      PS C:\> Set-LocalUser -Name Administrator -Password $Password
      ```

   c. Enable the Administrator account.

      ```
      PS C:\> Enable-LocalUser -Name Administrator
      ```

6. Set the following variables:

```
PS C:\> $DomainName = "dbeng.com";
```

```
PS C:\> $DomainMode = "Win2012R2";
```

```
PS C:\> $ForestMode = "Win2012R2";
```

```
PS C:\> $DatabasePath = "C:\Windows\NTDS";
```

```
PS C:\> $LogPath = "C:\Windows\NTDS";
```

```
PS C:\> $SysvolPath = "C:\Windows\SYSVOL";
```

7. Install the following Active Directory tools:

```
PS C:\> Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools
```

```
PS C:\> Install-ADDSForest -CreateDnsDelegation:$false -DatabasePath $DatabaseP
```

8. If the instance does not restart automatically after it creates the domain, restart the instance manually.
9. After the instance restarts, reconnect by using the `Administrator` account with the newly created domain name (`dbeng.com` in this tutorial).

10. Create a domain account, called `sql_service`, on which to run SQL Server:

    a. Securely enter a password for use with the SQL Server service account.

    ```
    PS C:> $Password = Read-Host -AsSecureString
    ```

    Note the password that you entered. Use this password for the SQL Server service account.

    b. Create the SQL Server service account and set its password.

    ```
    PS C:\> New-LocalUser -Name "sql_service" -Description "SQL Agent and SQL
    ```

The domain controller is configured, and you can disconnect from the `dc-windows` instance. Next, create the instances for the availability group (#create_instances_for_availability_group).

Creating instances for your availability group

Create SQL Server instances for an availability group. These instances use the following IP address configurations with alias IPs (/vpc/docs/configure-alias-ip-ranges):

- Instance name: `cluster-sql1`
  - Internal IP address: `10.0.0.4`
  - Windows Failover Cluster name: `cluster-ag`
  - Failover Cluster IP address: `10.0.0.5` (alias IP)
  - Availability Group name: `cluster-listener`
  - Availability Group Listener address: `10.0.0.6` (alias IP)
- Instance name: `cluster-sql2`
  - Internal IP address: `10.1.0.4`
  - Windows Failover Cluster name: `cluster-ag`

- Failover Cluster IP address: `10.1.0.5` (alias IP)
- Availability Group name: `cluster-listener`
- Availability Group Listener address: `10.1.0.6` (alias IP)

Create the SQL Server instances using one of the following methods:

- Create a SQL Server instance using a public image (/compute/docs/instances/sql-server/creating-sql-server-instances)
- Bring your existing licenses to Compute Engine (/compute/docs/instances/windows/ms-licensing) using license mobility through Software Assurance and apply those licenses on top of Windows Server (/compute/docs/instances/windows#windows_server) public images.

SQL Server public images   Existing licenses (#existing-licenses)

Create two instances from public SQL Server images. For this example, name the instances `cluster-sql1` and `cluster-sql2`. Specify a `200GB` boot disk size and an `n1-highmem-4` machine type. SQL Server instances usually require more compute resources than the domain controller instance. If you need additional compute resources later, you can change the machine type (/compute/docs/instances/changing-machine-type-of-stopped-instance) for these instances. If you need additional storage space, add a disk or resize a persistent boot disk (/compute/docs/disks/add-persistent-disk). In larger availability groups, you can create several instances.

Additionally, include the `--metadata sysprep-specialize-script-ps1` flag to run a PowerShell command during instance creation that installs the `Failover-Clustering` feature.

For this example, create two instances using the following commands:

```
gcloud compute instances create cluster-sql1 --machine-type n1-highmem-4 \
  --boot-disk-type pd-ssd --boot-disk-size 200GB \
  --image-project windows-sql-cloud --image-family sql-ent-2016-win-2016 \
  --zone us-central1-a \
  --network-interface "subnet=wsfcsubnet1,private-network-ip=10.0.0.4,aliases=10.0.0
  --can-ip-forward --metadata sysprep-specialize-script-ps1="Install-WindowsFeature


gcloud compute instances create cluster-sql2 --machine-type n1-highmem-4 \
  --boot-disk-type pd-ssd --boot-disk-size 200GB \
  --image-project windows-sql-cloud --image-family sql-ent-2016-win-2016 \
  --zone us-central1-f \
  --network-interface "subnet=wsfcsubnet2,private-network-ip=10.1.0.4,aliases=10.1.0
  --can-ip-forward --metadata sysprep-specialize-script-ps1="Install-WindowsFeature
```

For larger availability groups, you can create additional instances with the appropriate IP addresses.

After you create the instances, configure them so that they can function as an availability group.

1. Connect to both instances using RDP connections (/compute/docs/instances/connecting-to-instance#windows).
2. Change both instances to use static IP addresses and set the netmask to `255.255.0.0`. Open a PowerShell terminal as an administrator and set the static IP addresses to static. These commands might end your remote desktop connection and if so, you will have to connect again:
   - Instance 1:

     ```
     PS C:\> netsh interface ip set address name=Ethernet static 10.0.0.4 255.2
     ```

     ```
     PS C:\> netsh interface ip set dns Ethernet static 10.2.0.100
     ```

     ```
     PS C:\> netsh advfirewall firewall add rule name="Open Port 5022 for Avail
     ```

     ```
     PS C:\> netsh advfirewall firewall add rule name="Open Port 1433 for SQL S
     ```

   - Instance 2:

     ```
     PS C:\> netsh interface ip set address name=Ethernet static 10.1.0.4 255.2
     ```

```
PS C:\> netsh interface ip set dns Ethernet static 10.2.0.100
```

```
PS C:\> netsh advfirewall firewall add rule name="Open Port 5022 for Avail
```

```
PS C:\> netsh advfirewall firewall add rule name="Open Port 1433 for SQL S
```

3. Add both instances to the Windows domain. Open a PowerShell as an administrator and run the following `Add-Computer` command on both instances:

   ```
   PS C:\> Add-Computer -DomainName "dbeng.com" -Credential "dbeng.com\Administrat
   ```

   The command prompts you for your credentials. When the command finishes running, the instance restarts.

4. Reconnect to your instances using RDP (/compute/docs/instances/connecting-to-instance#windows), to set the SQL Server service accounts:

   a. Open **SQL Server Configuration Manager**.

   b. Select the **SQL Server services** tab, right-click **SQL Server**, and then click **Properties**.

   c. Set the account and password for `sql_service`.

The instances are now created for the availability group. Next, configure the Failover Cluster Manager (#configure_failover_cluster).

Configuring the Failover Cluster Manager

Enable failover clustering on the instances in your availability group, and configure one instance to act as the Failover Cluster Manager. Enable AlwaysOn High Availability on all instances in the group.

1. Reconnect to your instances using RDP (/compute/docs/instances/connecting-to-instance#windows), but use the Domain Administrator credentials. For this example, the domain is `dbeng` and the administrator account is `Administrator`. If you are using Chrome RDP for Google Cloud, in the **Options** menu, under the **Certificates** list, delete the existing RDP certificates for these addresses.

2. Select one of your instances and configure it to run as the Failover Cluster Manager.

   a. Open PowerShell as an administrator and set variables that reflect your cluster environment. For this example, set the following variables:

   ```
   PS C:\> $node1 = "cluster-sql1"
   ```

   ```
   PS C:\> $node2 = "cluster-sql2"
   ```

   ```
   PS C:\> $nameWSFC = "cluster-dbclus" #Name of cluster
   ```

   ```
   PS C:\> $ipWSFC1 = "10.0.0.5" #IP address of cluster in subnet 1
   ```

   ```
   PS C:\> $ipWSFC2 = "10.1.0.5" #IP address of cluster in subnet 2
   ```

   b. Create the failover cluster:

   ```
   PS C:\> New-Cluster -Name $nameWSFC -Node $node1, $node2 -NoStorage -Stati
   ```

   c. Enable AlwaysOn High Availability for both nodes in the cluster:

   ```
   PS C:\> Enable-SqlAlwaysOn -ServerInstance $node1 -Force
   ```

```
PS C:\> Enable-SqlAlwaysOn -ServerInstance $node2 -Force
```

3. On the secondary instance where there is no Cluster Manager, create a backup folder at `C:\SQLBackup` and share the folder as `\\cluster-sql2\SQLBackup` to the `sql_service` account for both read and write.
4. On both instances, create two folders at `C:\SQLData` and `C:\SQLLog`. Use these folders for the database data and log files.

The Failover Cluster Manager is ready. Next, create the availability group (#create_availability_group).

Creating the availability group

Create a test database and configure it to work with a new availability group. Alternatively, you can instead specify an existing database for the availability group.

1. If you do not have a database configured already, create a test database. On the Cluster Manager instance, run the SQL Server Management Studio and create a test database with the following SQL command:

```
CREATE DATABASE TestDB
ON PRIMARY (NAME = 'TestDB_Data', FILENAME='C:\SQLData\TestDB_Data.mdf', SIZE =
LOG ON (NAME = 'TestDB_Log', FILENAME='C:\SQLLog\TestDB_Log.ldf', SIZE = 256MB,
GO
USE [TestDB]
Exec dbo.sp_changedbowner @loginame = 'sa', @map = false;
  ALTER DATABASE [TestDB] SET RECOVERY FULL;
  GO
  BACKUP DATABASE TestDB to disk = '\\cluster-sql2\SQLBackup\TestDB.bak' WITH I
GO
```

2. On the Cluster Manager instance, run SQL Server Management Studio.
3. Right-click **AlwaysOn High Availability** and select **New Availability Group Wizard**.

4. On the **Specify Name** page, set an availability group name. For this example, specify `cluster-ag`.
5. On the **Select Databases** page, specify which database you want to replicate. For this example, specify the `TestDB` database.
6. On the **Specify Replicas** page, set both instances as replicas with automatic failover and synchronous commit.
7. On the **Select Data Synchronization** page, specify the network share to keep the backup of the database for initial synchronization. For this example, specify `\\cluster-sql2\SQLBackup`.
8. The **Validation** page generates a warning because there is no listener, but you can ignore this warning.
9. After the wizard finishes, right-click the new availability group and select **Add Listener**.
10. Specify the parameters for this listener:
    - **Listener DNS Name**: `cluster-listener`.
    - **Network Port**: `1433`.
    - **Network Mode**: `Static IP`.
11. Add two subnet and IP address fields. For this example, use the following subnet and IP address pairs:
    - `10.0.0.0/16` and `10.0.0.6`
    - `10.1.0.0/16` and `10.1.0.6`

Now you can connect to SQL Server using `cluster-listener` as the name of the SQL Server database instead of the name of the instances. This connection points to the instance that is currently active.

What's next

- If your instances require more compute resources, change the machine type (/compute/docs/instances/changing-machine-type-of-stopped-instance) of the instance.

- If you require more storage space, add a disk or resize persistent boot disk (/compute/docs/disks/add-persistent-disk).