Microsoft Windows Server Failover Clustering (WSFC) and SQL Server AlwaysOn Availability Groups on the AWS Cloud: Quick Start Reference Deployment

Mike Pfeiffer

July 2014
Last updated: April 2015 (revisions)

Table of Contents

Abstract

This reference implementation guide includes architectural considerations and configuration steps for running Microsoft Windows Server Failover Clustering (WSFC) clusters in the Amazon Web Services (AWS) cloud. We discuss how to launch the necessary AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC), and how to run a WSFC cluster across different subnets and Availability Zones.[1] Then we provide instructions for installing, configuring, and testing the WSFC cluster and a SQL Server 2012 or 2014 AlwaysOn Availability Group.

We also provide a sample AWS CloudFormation template designed to help you deploy the necessary and correctly configured infrastructure predictably and repeatedly. This automated template deploys an Active Directory Domain Services infrastructure along with SQL Server 2012 or 2014 instances configured in a Windows Server Failover Cluster in multiple Availability Zones into an Amazon VPC.

To begin the launch process for the AWS CloudFormation template in the US-West (Oregon) region, launch the Quick Start. This stack takes approximately three (3) hours to create.

Note
You are responsible for the cost of AWS services used while running this Quick Start Reference Deployment. The cost for running the template with default settings is approximately $5.50 an hour, so you can complete the initial deployment for about $15.00. See the pricing pages of the AWS services you will be using for full details.

This guide targets IT infrastructure administrators and DevOps personnel. After reading it, you should have a good understanding of how to launch the necessary infrastructure and of the configuration steps to deploy WSFC clusters and AlwaysOn Availability Groups repeatedly and reliably in the AWS cloud.

Note: AlwaysOn Availability Groups is a feature supported by SQL Server 2012 and 2014 Enterprise Edition. You can deploy SQL Server 2012 or 2014 Enterprise Edition using the Microsoft License Mobility through Software Assurance program.

What We'll Cover

Implementing a Windows Server Failover Cluster (WSFC) cluster in the AWS cloud, which is a prerequisite for deploying an AlwaysOn Availability Group, is very similar to deploying it in an on-premises setting as long as you meet two key requirements:

- You must deploy the cluster nodes inside an Amazon VPC.
- You must deploy WSFC cluster nodes in separate subnets.

Keeping these two key requirements in mind, we provide instructions for deploying the WSFC and an AlwaysOn Availability Group. We call out any AWS-specific considerations along the way.

[1] Running WSFC nodes in the same subnet is currently not supported on the AWS cloud.

First, we provide implementation guidance for setting up AD DS in an Amazon VPC. Next, we walk you through the steps necessary to configure a two-node automatic failover cluster with a file share witness. On this cluster, we then deploy an AlwaysOn Availability Group with two availability replicas. The goal of this configuration is to protect from the failure of a single instance. Other failover cluster and availability group configurations are possible to serve either high availability (HA) or disaster recovery (DR), or both scenarios together. You should customize some of the steps to deploy a solution that best meets your business, IT, and security requirements.

This deployment includes the following steps:

- **Part 1: Implement Active Directory Domain Services**

  1. Set up the virtual network for Active Directory and the WSFC cluster within AWS, including subnets in two Availability Zones.
  2. Configure private and public routes.
  3. Launch Windows Server 2012 Amazon Machine Images (AMIs) and set up and configure Active Directory and DNS.
  4. Create and configure Amazon EC2 Security Groups to control network traffic between Active Directory Domain Controllers and Member Servers.
  5. Enable administrative ingress and egress into your Amazon VPC via Remote Desktop Gateway and NAT instances.

  This step is automated by an AWS CloudFormation template that can be launched from this guide.

- **Part 2: Launch and Configure the Server Infrastructure**

  1. Create and configure Amazon EC2 Security Groups to control network traffic between WSFC cluster nodes. Set up SQL Server 2012 or 2014 Enterprise Edition.
  2. Create the WSFC cluster.
  3. Enable AlwaysOn High Availability.

  This step is automated by an AWS CloudFormation template that can be launched from this guide.

- **Part 3: Configure a SQL Server 2012 or 2014 AlwaysOn Availability Group**

  1. Create a database.
  2. Create an AlwaysOn Availability Group.

  This step requires manually implementation steps that are covered in detail later in this guide.

When you have completed these steps, you will have deployed the following architecture and associated resources in the AWS cloud:

- One Amazon VPC
- One public route
- One Internet Gateway
- Per Availability Zone:
  - 2 private subnets and 2 public subnets
  - 2 private routes

  - 2 Windows Server 2012–based Remote Desktop Gateway (RDGW) instances and 2 Linux-based NAT instances to enable administrative ingress and egress
  - 4 Elastic IP Addresses associated with the NAT and RDGW instances
  - 2 Windows Server 2012–based instances to host the Active Directory
  - 2 Windows Server 2012–based instances to host the WSFC Nodes and SQL Server 2012 or 2014 Instances
- Security Groups to control the secure flow of traffic between the instances deployed in the Amazon VPC

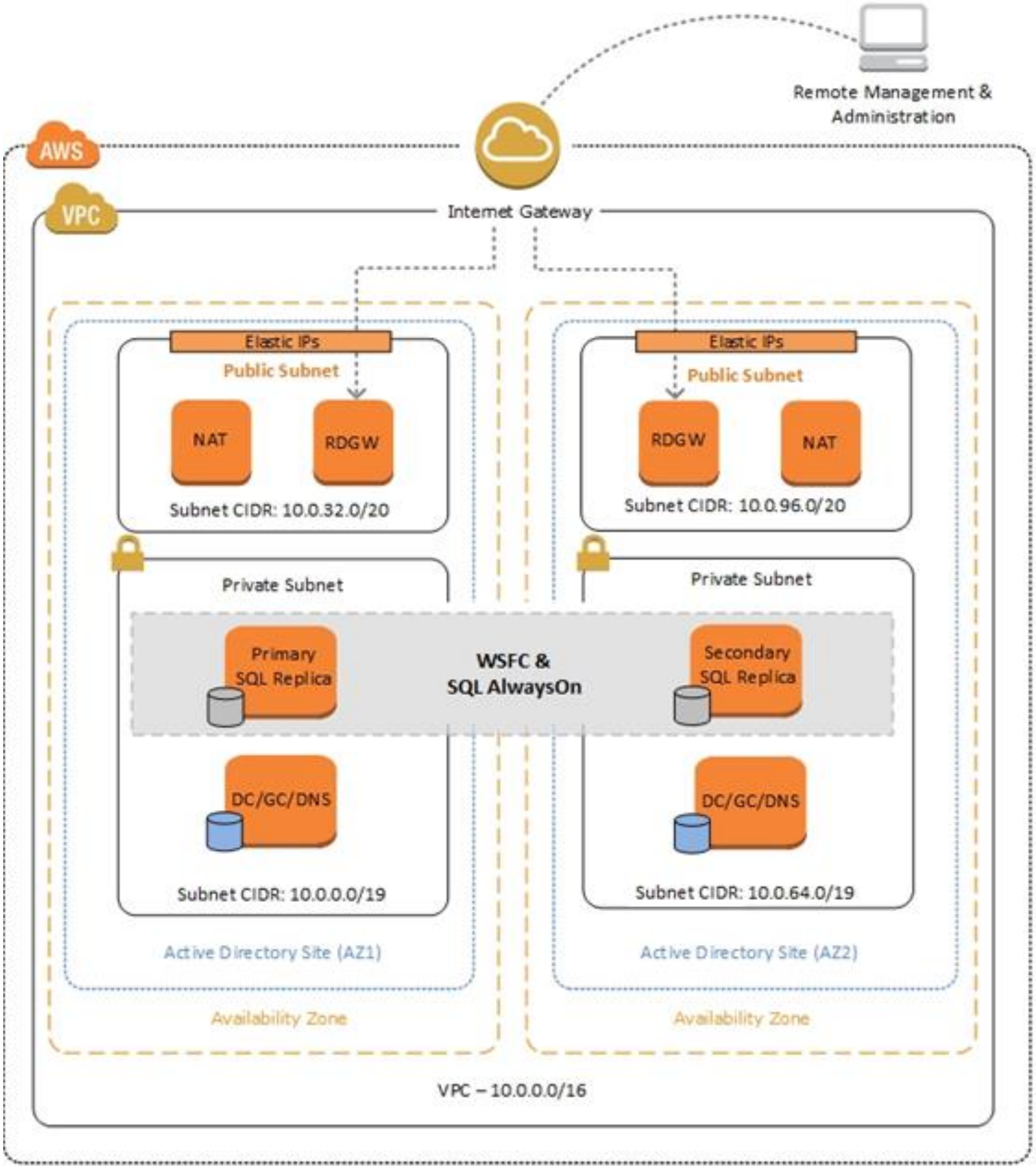


Figure 1: WSFC Cluster Set Up Across Different Subnets and Availability Zones

Automated Deployment

With this guide, we provide sample AWS CloudFormation templates designed to help you deploy the necessary and correctly configured infrastructure predictably and repeatedly.

The CloudFormation template that you can launch from this guide will perform the following tasks:

- Deploy Windows Server 2012–based instances as WSFC nodes into their respective subnets in an Amazon VPC built using the architecture provided in the Quick Start Reference Deployment for Microsoft Active Directory on AWS
- Rename the instances to a friendly NetBIOS name of your choice
- Join the Windows instances to the domain
- Deploy Windows Server 2012–based instances as WSFC nodes into separate Availability Zones
- Create a SQL Service Account (for example, sqlsa) and add it to the local Administrator Group on each WSFC node
- Install the WSFC feature on each WSFC node
- Download and install SQL Server 2012 or 2014 Enterprise Edition on each WSFC node
- Enable SQL Server AlwaysOn on each WSFC node

The following sections outline the deployment steps taken by the CloudFormation templates. We explain the approach we took to automate the deployment, but you can also use the template as a high-level guide to manually perform the tasks on base Windows server instances.

Before You Get Started

Implementing a WSFC cluster and AlwaysOn Availability Groups is an advanced topic. If you are new to AWS, see the Getting Started section of the AWS documentation. In addition, you want to be familiar with the following topics:

- Amazon EC2
- Amazon VPC
- Windows Server 2012
- Windows Server Active Directory and DNS
- Windows Server Failover Clustering (WSFC)
- SQL Server AlwaysOn Availability Groups

Launch and Configure Microsoft Windows Server Failover Clustering (WSFC) on Amazon Web Services (AWS)

Part 1: Implement Active Directory Domain Services

The underlying Active Directory architecture for this deployment is based on an existing reference implementation provided in the Quick Start Reference Deployment for Microsoft Active Directory on AWS. This architecture provides a highly available Active Directory Domain Services infrastructure that supports the following best practices:

- Domain Controllers should be placed in a minimum of two Availability Zones to provide high availability.
- Domain Controllers and other non-internet facing servers should be placed in private subnets.
- Instances launched by CloudFormation templates will require internet access to connect to the AWS CloudFormation endpoint during the bootstrapping process. To support this configuration, public subnets are used to host NAT instances for outbound internet access. Remote Desktop Gateways are also deployed into the public subnets for remote administration. Other components, such as reverse proxy servers can be placed into these public subnets, if needed.

Several critical components and considerations are covered in the Active Directory reference that addresses Active Directory Site and Subnet design and how DNS and DHCP work inside an Amazon VPC.

For more details on the underlying Active Directory and network design, see the reference architecture outlined in the Quick Start Reference Deployment for Microsoft Active Directory on AWS.

Part 2: Launch and Configure the Server Infrastructure

This sections describes how to deploy SQL Server 2012 or 2014 instances in a Windows Server Failover Cluster configuration and enable SQL Server AlwaysOn in the Amazon VPC you created in Part 1.

High Availability and Disaster Recovery in the AWS Cloud

Amazon EC2 provides the ability to place instances in multiple locations composed of regions and Availability Zones. Regions are dispersed and located in separate geographic areas. Availability Zones are distinct locations within a region that are engineered to be isolated from failures in other Availability Zones and that provide inexpensive, low-latency network connectivity to other Availability Zones in the same region.

By launching your instances in separate regions, you can design your application to be closer to specific customers or to meet legal or other requirements. By launching your instances in separate Availability Zones, you can protect your applications from the failure of a single location. WSFC provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the AWS cloud.

SQL Server Enterprise Edition

Although Amazon Machine Images (AMIs) for SQL Server Express and SQL Server Web Edition are available for launch on AWS, the Enterprise edition is not available on an AMI. To install SQL Server 2012 or 2014 Enterprise Edition on AWS, you can download the trial software from Microsoft. If you choose to deploy this solution using the provided CloudFormation templates, a script will automatically connect to the Microsoft download site and install the trial software for you.

You'll find the installation software in a share on the Domain Controller in the first Availability Zone, which by default will be *\\dc1\sqlinstall\*. If you need to re-run the SQL installation, there will be a batch file on the desktop of each node called *InstallSQLEE.bat* that will launch the installer from the share. If you do re-run the installation, make sure you right-click the batch file and select Run as Administrator to start the installation.

The SQL services are configured to run under the **sqlsa** account that is created in Active Directory. This account is also added to the local administrators groups on each WSFC node.

Note: AWS does not provide installation media for Microsoft software. If you are not using the CloudFormation templates, you can set up a test or evaluation environment by downloading a trial version of SQL Server 2012 or 2014 at http://www.microsoft.com/en-us/server-cloud/products/sql-server/. For a production deployment, use your volume licensing software and mobilize the license as described in the License Mobility through Software Assurance program.

Security Groups and Firewalls

When launched, Amazon EC2 instances must be associated with a Security Group, which acts as a stateful firewall. You have complete control over the network traffic entering or leaving the Security Group, and you can build granular rules that are scoped by protocol, port number, and source or destination IP address or subnet. By default, all traffic egressing a Security Group is permitted. Ingress traffic, on the other hand, must be configured to allow the appropriate traffic to reach your instances.

In the Securing the Microsoft Platform on Amazon Web Services whitepaper, we discuss in detail the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers using Security Groups. We recommend that you tightly control ingress traffic in order to reduce the attack surface of your Amazon EC2 instances.

Domain Controllers and member servers require several Security Group rules to allow traffic for services such as AD DS replication, user authentication, Windows Time services, and Distributed File System (DFS), among others. The WSCF nodes running SQL Server will need to permit several additional ports to communicate with each other as well. Finally, instances launched into the application server tier will need to establish SQL client connections to the WSFC nodes.

If you choose to implement this solution using our CloudFormation templates, a number Security Groups and rules will be created for you. For a detailed list of port mappings, see the Appendix of the Active Directory reference, and the Appendix of this guide.

In addition to Amazon Security Groups, the Windows Firewall also needs to be modified on the SQL server instances. During the bootstrapping process, a script will run on each instance that opens the TCP ports 1433, 1434, 4022, 5022, and 135 on the Windows Firewall.

Storage on the WSFC Nodes

Storage capacity and performance is a key aspect of any production SQL Server installation. While capacity and performance will vary from one deployment to the next, we provide a reference configuration that you can use as a starting point. The CloudFormation template will deploy the WSFC nodes using the r3.2xlarge instance type by default. This is a memory optimized instance with 1 x 160 GB of SSD instance storage.

In an effort to provide highly performant and durable storage, we've also included Amazon Elastic Block Store (Amazon EBS) volumes in this reference architecture. Amazon EBS volumes are network-attached disk storage, which you can create and attach to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use a block device. Amazon EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component.

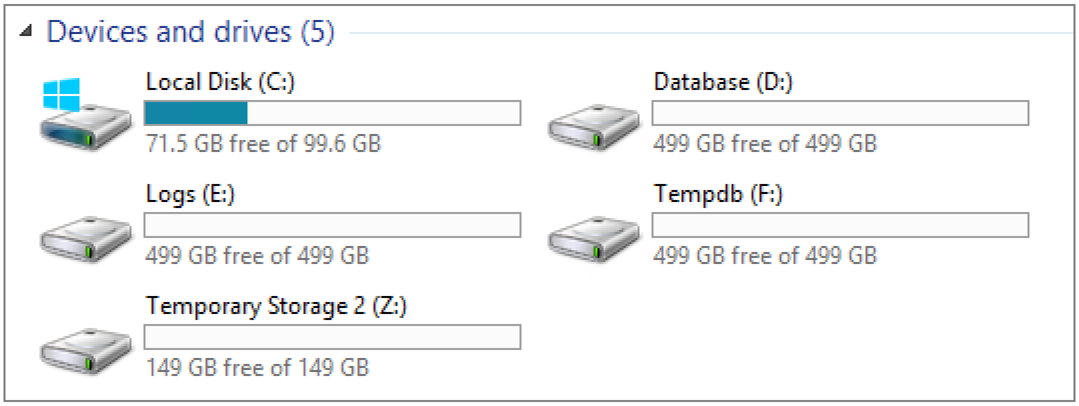
Provisioned IOPS Amazon EBS volumes offer storage with consistent and low-latency performance. They are backed by Solid-State Drives (SSDs) and are designed for applications with I/O-intensive workloads such as databases.

Amazon EBS-optimized instances, such as the r3.2xlarge, deliver dedicated throughput between Amazon EC2 and Amazon EBS. The dedicated throughput minimizes contention between Amazon EBS I/O and other traffic from your Amazon EC2 instance, providing the best performance for your Amazon EBS volumes.

On each WSFC node, we deploy three 500-GiB General Purpose (SSD) volumes to store databases, logs, TempDB, and backups. This is in addition to the root General Purpose (SSD) volume used by the operating system. The General Purpose (SSD) volume type delivers a consistent baseline of 3 IOPS/GiB, which provides a total of 1,500 IOPS per volume for SQL Server database and log volumes. This is provided as a starting point. If you need more IOPS per volume, consider using Provisioned IOPS (SSD) volumes or use disk striping within Windows.

The disk layout for SQL Server in this Quick Start uses the following Amazon EBS volumes:

- 1 SSD volume (100 GiB) for the operating system (C:)
- 1 SSD volume (500 GiB) to host the SQL Server database files (D:)
- 1 SSD volume (500 GiB) to host the SQL Server log files (E:)
- 1 SSD volume (500 GiB) to host the SQL Server TempDB and backup files (F:)



Devices and drives (5)

Local Disk (C:)
71.5 GB free of 99.6 GB

Database (D:)
499 GB free of 499 GB

Logs (E:)
499 GB free of 499 GB

Tempdb (F:)
499 GB free of 499 GB

Temporary Storage 2 (Z:)
149 GB free of 149 GB

Figure 2: WSFC Node Disk Layout

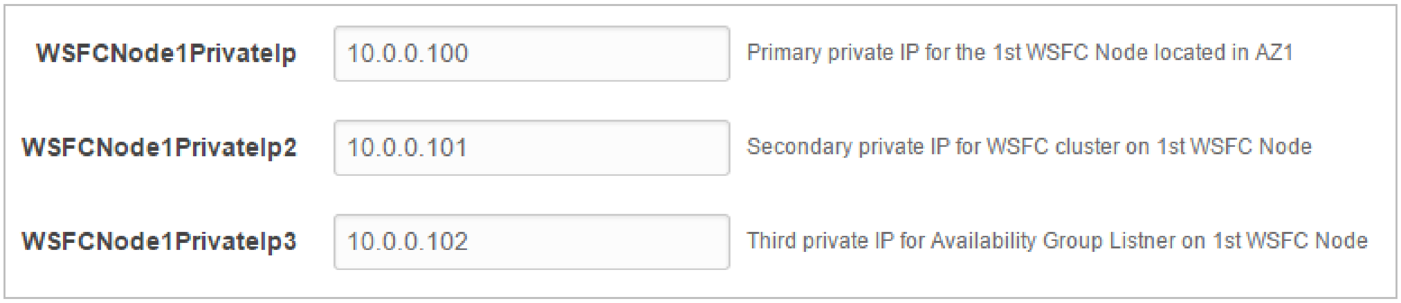
Figure 2 shows the disk layout on each SQL Server node. The Z: drive is instance storage that can be used for ephemeral data, such as the operating system page file. Keep in mind that data on instance storage will be lost when you stop your Amazon EC2 instance.

IP Addressing on the WSFC Nodes

In order to support Windows Server Failover Clustering and AlwaysOn Availability Group listeners, each node hosting the SQL Server instances participating in the cluster will need to have a total of three (3) IP addresses assigned.

- One IP address will be used as the primary IP address for the instance.
- A second IP address will act as the WSFC IP resource.
- A third IP address will be used to host the AlwaysOn Availability Group listener.

When launching the CloudFormation template, you will be given the opportunity to specify these addresses for each node. By default, the 10.0.0.0/19 and 10.0.64.0/19 CIDR blocks are used for the private subnets.



WSFCNode1PrivateIp	10.0.0.100	Primary private IP for the 1st WSFC Node located in AZ1
WSFCNode1PrivateIp2	10.0.0.101	Secondary private IP for WSFC cluster on 1st WSFC Node
WSFCNode1PrivateIp3	10.0.0.102	Third private IP for Availability Group Listner on 1st WSFC Node

Figure 3: Defining WSFC Node IP Addresses when Launching the CloudFormation Template

Windows Server Failover Clustering

Once your Windows Server 2012 instances have been deployed and domain joined, you're ready to build the cluster. The CloudFormation templates carry out this task when deploying the second node. Assuming the default template parameters are used, the following PowerShell commands are executed to complete this task:

```
Install-WindowsFeature failover-clustering -IncludeManagementTools

New-Cluster -Name WSFCluster1 -Node WSFCNODE1,WSFCNODE2 -StaticAddress
10.0.0.101,10.0.64.101
```

The first command runs on each instance during the bootstrapping process. It installs the required components and management tools for the Failover Clustering services. The second command runs near the end of the bootstrapping process on the second node and is responsible for creating the cluster and for defining the server nodes and IP addresses.

Since there will be an even number of servers configured in the cluster, we need a third resource to maintain a majority vote to keep the cluster online in the event of an individual server failure. For this, we'll utilize the file share witness resource, which requires us to modify the cluster settings to *Node and File Share Majority*. The first step in making this configuration change is to create the share. The CloudFormation template uses the Domain Controller in the first Availability Zone to host this share. We do this to minimize the number of instances required to get you up and running. However, for production environments, any domain joined server can be used for this task.

The CloudFormation template will create a folder called C:\witness on the first DC, using the share name *witness*. The end result, assuming you've chosen the default parameters, will give you a file share defined as \\dc1\witness. The Active Directory computer account that will be created when forming the cluster (WSFCNODE1) will be given NTFS permissions to access the share. After that, the cluster will be updated to use the share as the file share witness resource.

```
Set-ClusterQuorum -NodeAndFileShareMajority \\dc1\witness
```

With SQL Server installed and both nodes operating in a Windows Server Failover Cluster, we can move on to the next step of enabling SQL AlwaysOn.

AlwaysOn Configuration

After SQL Server Enterprise Edition has been installed, and the Windows Server Failover Cluster is built, the next step is to enable SQL AlwaysOn. The CloudFormation template provided by this guide will enable SQL Server AlwaysOn for you. This is one of the last steps taken by the automated solution, and it is done with a simple PowerShell command.

```
Enable-SqlAlwaysOn -ServerInstance WSFCNODE1
```

This command should be run on each node, and the proper server name needs to be provided as a parameter value for the ServerInstance parameter. CloudFormation will run this command for each node.

As you move on to creating an Availability Group, you'll need to provide a network share used to perform an initial data synchronization. As you progress through the New Availability Group wizard, a full backup for each selected database will be taken and placed in the share. The secondary node will connect to the share and restore the database backups before joining the Availability Group.

To accommodate this initial synchronization, The CloudFormation templates will create a folder called C:\replica on the first DC, using the share name *replica*. The end result, assuming you've chosen the default parameters, will give you a file share defined as \\dc1\replica. The **sqlsa** Active Directory user account will be given NTFS permissions to this share, as this is the account under which the SQL services run. Again, for production environments, you may want to use a dedicated file server to host this share rather than a Domain Controller.

Since it is up to you to decide what databases need to be created, our automated solution ends after enabling SQL Server AlwaysOn. After deploying this solution, you can move on to creating your databases and making them highly available by creating an AlwaysOn Availability Group. This process is covered in Part 3 of this guide.

Automation: Implement Microsoft WSFC and SQL Server Enterprise

This automated template deploys an Active Directory Domain Services infrastructure along with SQL Server 2012 or 2014 instances configured in a Windows Server Failover Cluster in multiple Availability Zones into an Amazon VPC.

To launch the AWS CloudFormation template in the US-West (Oregon) region, **launch the Quick Start**.

This stack takes approximately three (3) hours to create.

Note

You are responsible for the cost of AWS services used while running this Quick Start Reference Deployment. The cost for running the template with default settings is approximately $5.50 an hour, so you can complete the initial deployment for about $15.00. See the pricing pages of the AWS services you will be using for full details.

Template Customization

This automation allows for rich customization of parameters at template launch. You can modify these parameters, change the default values, or, if you choose to edit the code of the template itself, you can create an entirely new set of parameters based on your specific deployment scenario. The parameters include the following default values:

Parameter	Default	Description
KeyPairName	*<User Provided>*	Public/private key pairs allow you to connect securely to your instance after it launches.
ADInstanceType	m3.xlarge	Amazon EC2 instance type for the first Active Directory Instance
AD2InstanceType	m3.xlarge	Amazon EC2 instance type for the second Active Directory Instance

NATInstanceType	t2.small	Amazon EC2 instance type for the NAT Instances
RDGWInstanceType	m3.xlarge	Amazon EC2 instance type for the Remote Desktop Gateway Instance
WSFCNode1InstanceType	r3.2xlarge	Amazon EC2 instance type for the first WSFC Node
WSFCNode2InstanceType	r3.2xlarge	Amazon EC2 instance type for the second WSFC Node
DomainDNSName	example.com	Fully qualified domain name (FQDN) of the forest root domain, e.g. example.com
DomainNetBIOSName	example	NetBIOS name of the domain (up to 15 characters) for users of earlier versions of Windows, e.g. EXAMPLE
ADServerNetBIOSName1	DC1	NetBIOS name of the first AD Server (up to 15 characters)
ADServerNetBIOSName2	DC2	NetBIOS name of the second AD Server (up to 15 characters)
WSFCNode1NetBIOSName	WSFCNode1	NetBIOS name of the first WSFC Node (up to 15 characters)
WSFCNode2NetBIOSName	WSFCNode2	NetBIOS name of the second WSFC Node (up to 15 characters)
RestoreModePassword	*<User Provided>*	Password for a separate Administrator account when the domain controller is in Restore Mode. This must be a complex password that's at least 8 characters long.
DomainAdminUser	stackadmin	User name for the account that will be added as Domain Administrator. This is separate from the default "Administrator" account
DomainAdminPassword	*<User Provided>*	Password for the domain admin user. This must be a complex password that's at least 8 characters long.
SQLServiceAccount	sqlsa	User name for the SQL Server Service Account. This Account is a Domain User
SQLServiceAccountPassword	*<User Provided>*	Password for the SQL Service account This must be a complex password that's at least 8 characters long.
UserCount	25	Total number of test user accounts to create in Active Directory
DMZ1CIDR	10.0.32.0/20	CIDR Block for the Public DMZ Subnet located in AZ1
DMZ2CIDR	10.0.96.0/20	CIDR Block for the Public DMZ Subnet located in AZ2
PrivSub1CIDR	10.0.0.0/19	CIDR block for the Private Subnet located in AZ1
PrivSub2CIDR	10.0.64.0/19	CIDR block for the Private Subnet located in AZ2
VPCCIDR	10.0.0.0/16	CIDR Block for the Amazon VPC
AD1PrivateIp	10.0.0.10	Fixed private IP for the first Active Directory server located in AZ1
AD2PrivateIp	10.0.64.10	Fixed private IP for the second Active Directory server located in AZ2
WSFCNode1PrivateIp	10.0.0.100	Primary private IP for the first WSFC Node located in AZ1
WSFCNode1PrivateIp2	10.0.0.101	Secondary private IP for WSFC cluster on first WSFC Node
WSFCNode1PrivateIp3	10.0.0.102	Third private IP for Availability Group Listener on first WSFC Node
WSFCNode2PrivateIp	10.0.64.100	Primary private IP for the second WSFC Node located in AZ2
WSFCNode2PrivateIp2	10.0.64.101	Secondary private IP for WSFC cluster on second WSFC Node
WSFCNode2PrivateIp3	10.0.64.102	Third private IP for Availability Group Listener on second WSFC Node
SQLServerVersion	2014	Version of SQL Server to install on WSFC Nodes. Options include either 2014 or 2012

Note: If you have already deployed Active Directory Domain Services on AWS, you can launch this SQL Server infrastructure into an existing Amazon VPC using Template_1_SQL_AlwaysOn.template.

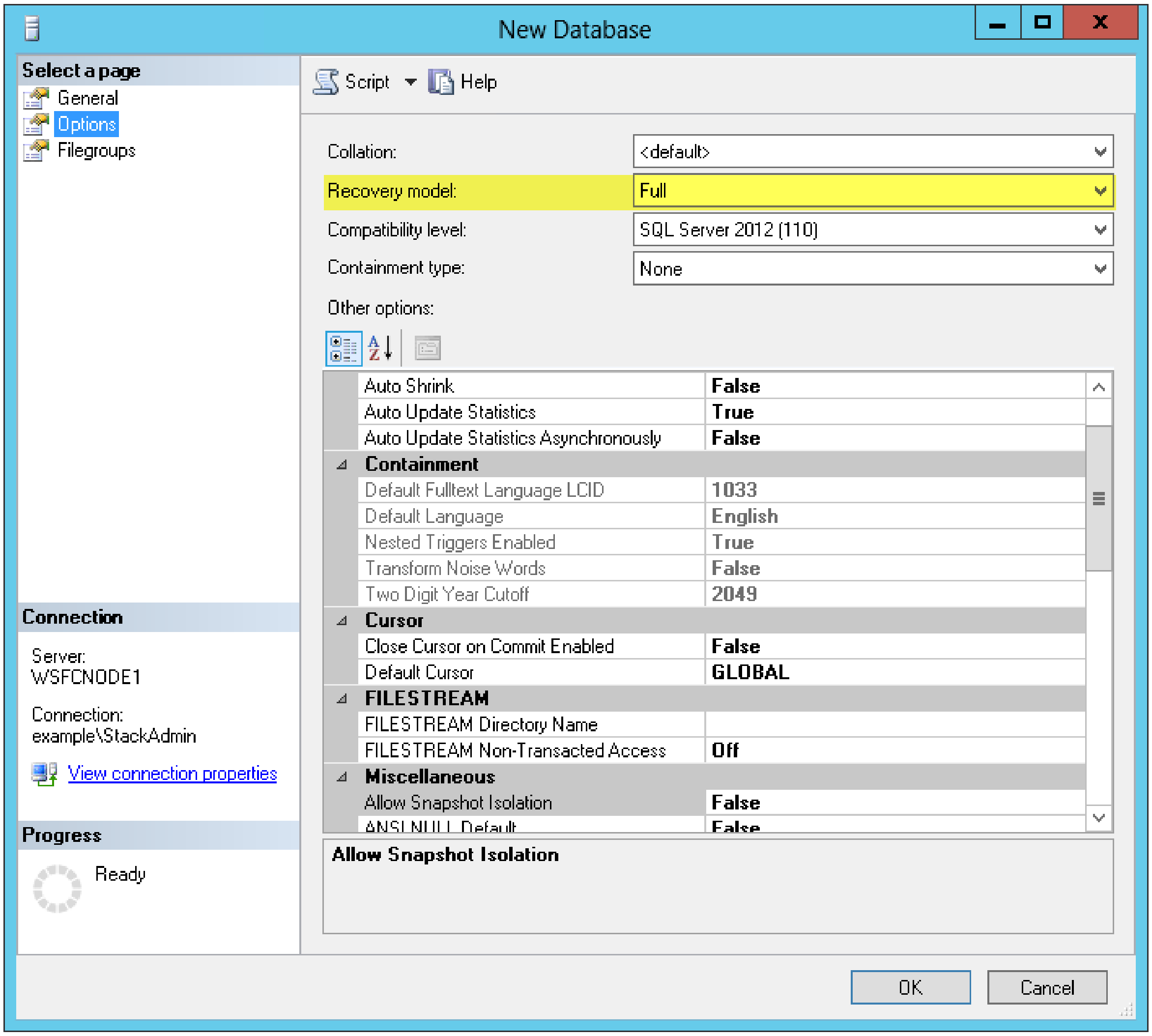
Part 3: Configure a SQL Server AlwaysOn Availability Group

After you have successfully deployed the necessary infrastructure from Part 1 and Part 2, you can configure the WSFC nodes using the following steps.

Create a Test Database or Attach an Existing Database

1. Using **SQL Server Management Studio**, connect to the first cluster node (e.g., WSFCNode1).
2. Create a new database or attach a test database.

3. Ensure the **Recovery model** on the database is set to **Full**.



Server:
WSFCNODE1

Connection:
example\StackAdmin

Collation: <default>

Recovery model: Full

Compatibility level: SQL Server 2012 (110)

Containment type: None

Other options:

Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False
Containment	
Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049
Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL
FILESTREAM	
FILESTREAM Directory Name	
FILESTREAM Non-Transacted Access	Off
Miscellaneous	
Allow Snapshot Isolation	False
ANSI NULL Default	False

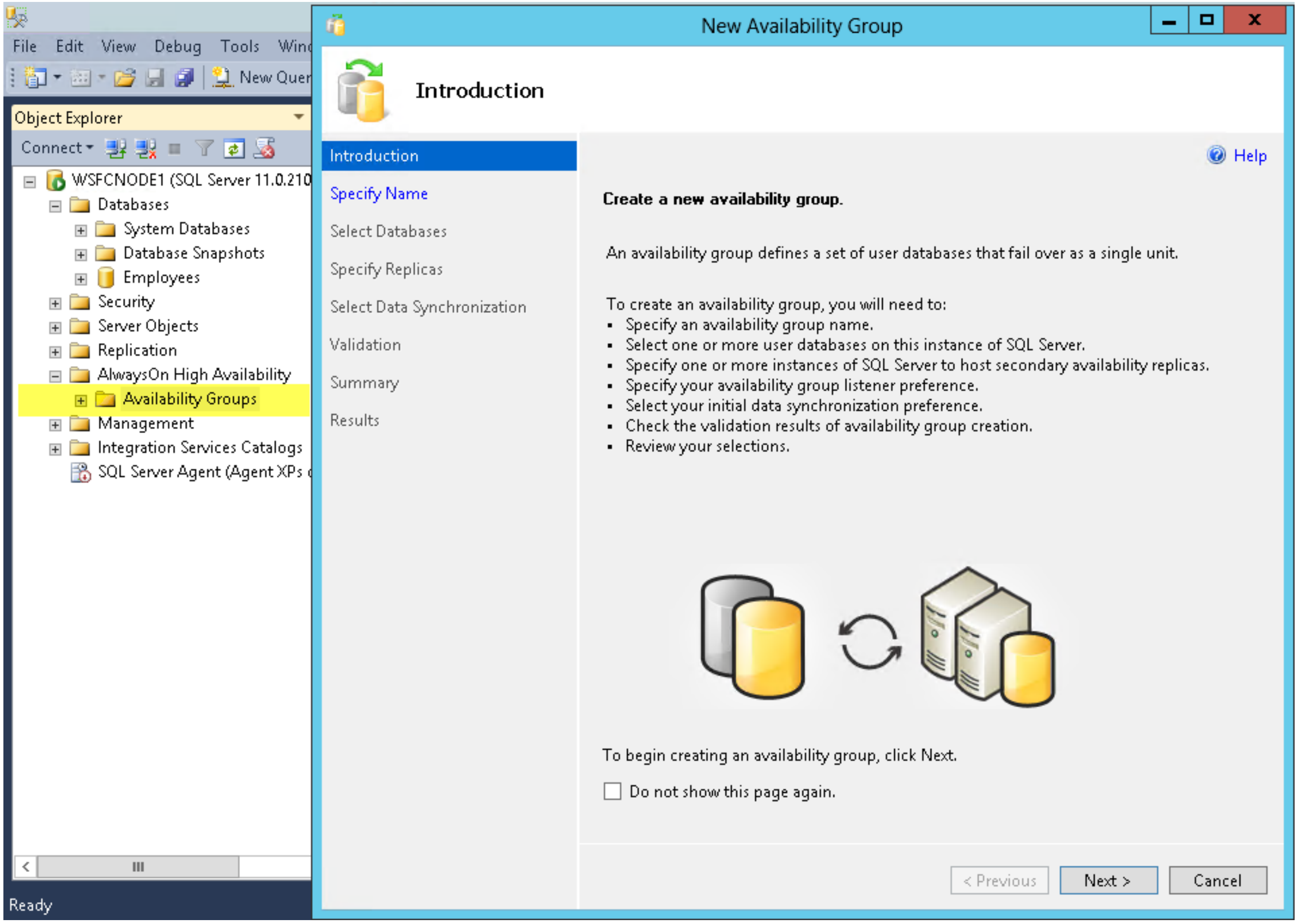
Allow Snapshot Isolation

Figure 4: Creating a New Database in SQL Management Studio

4. Back up the database. Right click on the database in SQL Management Studio and select **Tasks** > **Backup**.

Create an availability group

1. In **Object Explorer**, right-click **AlwaysOn High Availability** and launch the **New Availability Group** wizard.



Create a new availability group.

An availability group defines a set of user databases that fail over as a single unit.

To create an availability group, you will need to:

- Specify an availability group name.
- Select one or more user databases on this instance of SQL Server.
- Specify one or more instances of SQL Server to host secondary availability replicas.
- Specify your availability group listener preference.
- Select your initial data synchronization preference.
- Check the validation results of availability group creation.
- Review your selections.

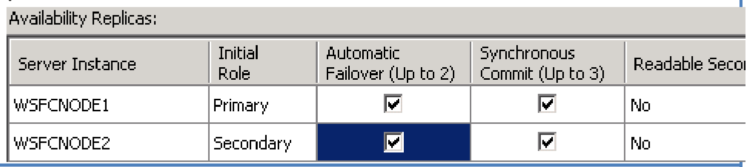
To begin creating an availability group, click Next.

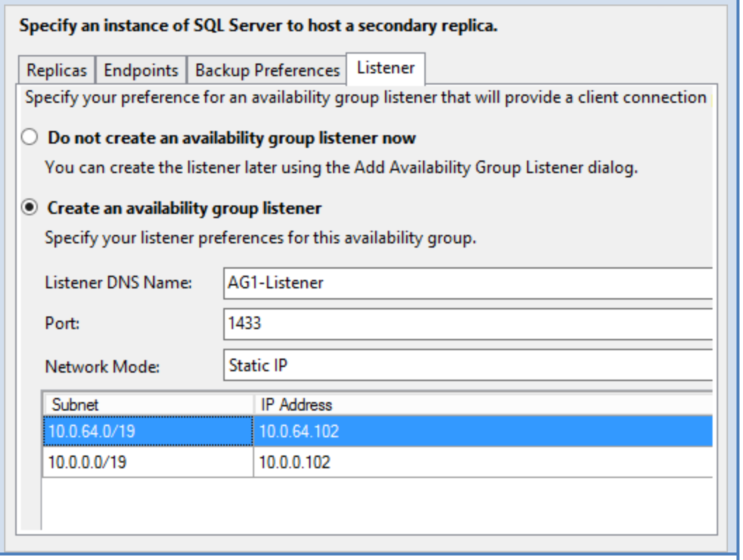
□ Do not show this page again.

Figure 5: Creating a New Availability Group in SQL Management Studio

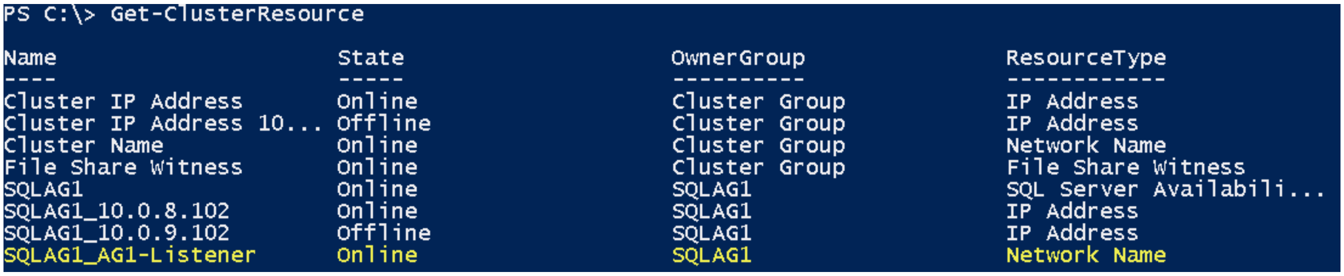
2. Follow the **New Availability Group** wizard and take the following actions:

New Availability Group Wizard Page	Action	Comments
Introduction	**Next**	
Specify Availability Group Name	Enter “SQLAG1” + **Next**.	
Select Databases	Select the database you created or attached previously + **Next**.	

<table>
<tr><td><b>Specify Replicas</b></td><td>Add the second cluster node (e.g., WSFCNode2) and select <b>Automatic Failover</b></td><td>.

Availability Replicas:
<table>
<tr><th>Server Instance</th><th>Initial Role</th><th>Automatic Failover (Up to 2)</th><th>Synchronous Commit (Up to 3)</th><th>Readable Seco</th></tr>
<tr><td>WSFCNODE1</td><td>Primary</td><td>☑</td><td>☑</td><td>No</td></tr>
<tr><td>WSFCNODE2</td><td>Secondary</td><td>☑</td><td>☑</td><td>No</td></tr>
</table></td></tr>
<tr><td><b>Specify Replicas</b></td><td>On the <b>Listener</b> tab, select <b>Create an availability group listener</b>, provide a <b>Listener DNS Name</b> (e.g., AG1-Listener), and then specify the TPC port used by this listener (e.g., <b>1433</b>) and add the two private subnets into which you have deployed the cluster nodes and a corresponding IPv4 address.

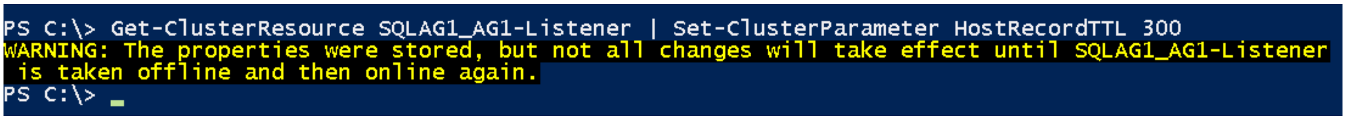
<b>Note</b>: We are going to use the second of the secondary private IP addresses we assigned earlier to the nodes (e.g., 10.0.0.102 and 10.0.64.102).</td><td>
<b>Specify an instance of SQL Server to host a secondary replica.</b>
Replicas | Endpoints | Backup Preferences | Listener
Specify your preference for an availability group listener that will provide a client connection
○ <b>Do not create an availability group listener now</b>
You can create the listener later using the Add Availability Group Listener dialog.
● <b>Create an availability group listener</b>
Specify your listener preferences for this availability group.
Listener DNS Name: AG1-Listener
Port: 1433
Network Mode: Static IP
<table>
<tr><th>Subnet</th><th>IP Address</th></tr>
<tr><td>10.0.64.0/19</td><td>10.0.64.102</td></tr>
<tr><td>10.0.0.0/19</td><td>10.0.0.102</td></tr>
</table></td></tr>
<tr><td><b>Select Initial Data Synchronization</b></td><td>Select <b>Full</b> + <b>Next</b>.</td><td>Specify the replica file share created on DC1 (e.g., \\DC1\Replica)</td></tr>
<tr><td><b>Validation</b></td><td><b>Next</b></td><td>Make sure the results show <b>Success</b> for all the validation steps</td></tr>
<tr><td><b>Summary</b></td><td><b>Finish</b></td><td></td></tr>
<tr><td><b>Results</b></td><td><b>Close</b></td><td></td></tr>
</table>

3. Run Windows PowerShell as Administrator and change the availability group Listener Host Record TTL to 300.

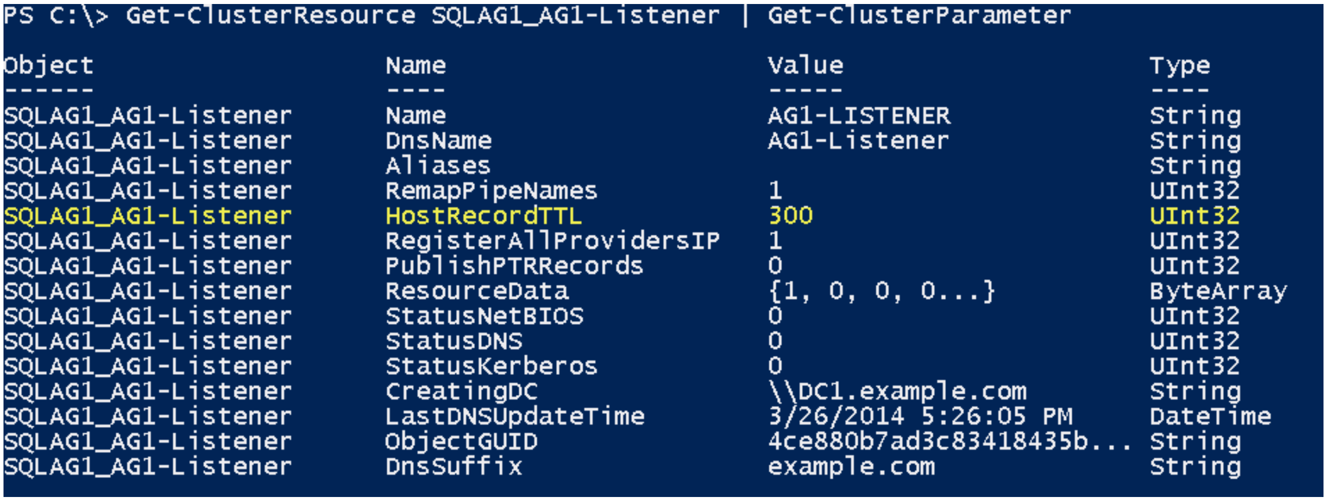


```
PS C:\> Get-ClusterResource

Name                      State           OwnerGroup      ResourceType
----                      -----           ----------      ------------
Cluster IP Address        Online          Cluster Group   IP Address
Cluster IP Address 10...  Offline         Cluster Group   IP Address
Cluster Name              Online          Cluster Group   Network Name
File Share Witness        Online          Cluster Group   File Share Witness
SQLAG1                    Online          SQLAG1          SQL Server Availabili...
SQLAG1_10.0.8.102         Online          SQLAG1          IP Address
SQLAG1_10.0.9.102         Offline         SQLAG1          IP Address
SQLAG1_AG1-Listener       Online          SQLAG1          Network Name
```



```
PS C:\> Get-ClusterResource SQLAG1_AG1-Listener | Set-ClusterParameter HostRecordTTL 300
WARNING: The properties were stored, but not all changes will take effect until SQLAG1_AG1-Listener
 is taken offline and then online again.
PS C:\> _
```

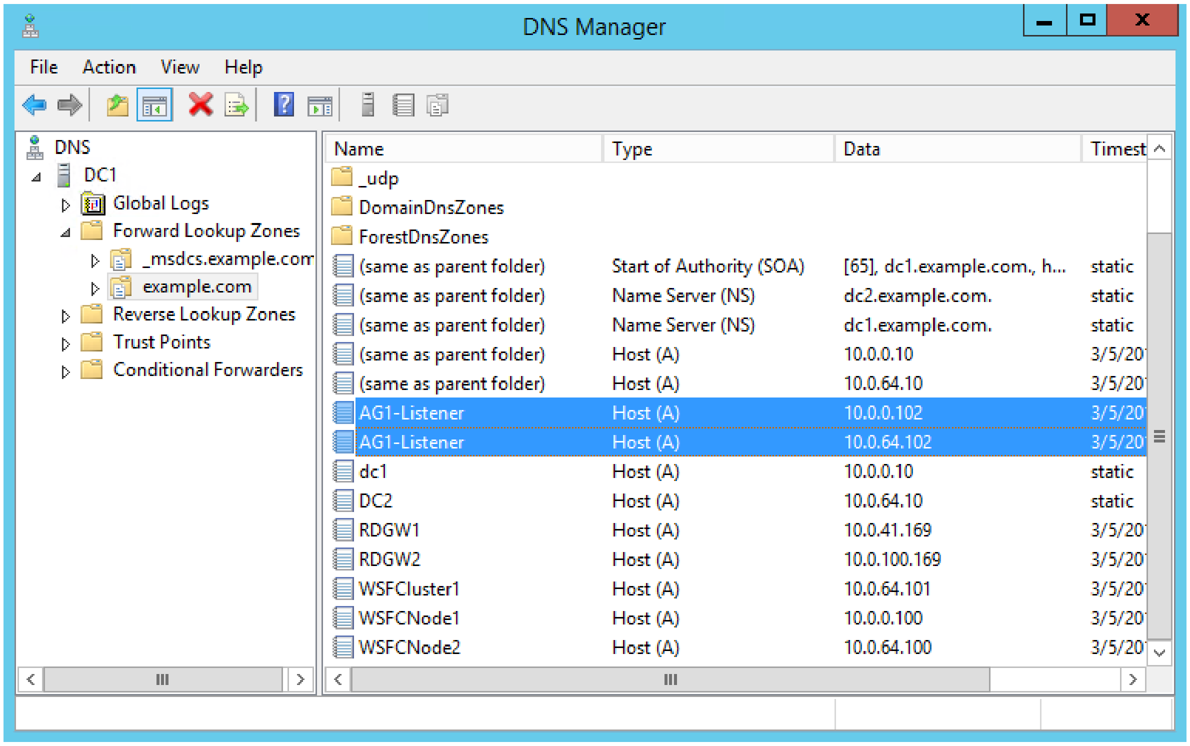


```
PS C:\> Get-ClusterResource SQLAG1_AG1-Listener | Get-ClusterParameter

Object                   Name                     Value                    Type
------                   ----                     -----                    ----
SQLAG1_AG1-Listener      Name                     AG1-LISTENER             String
SQLAG1_AG1-Listener      DnsName                  AG1-Listener             String
SQLAG1_AG1-Listener      Aliases                                           String
SQLAG1_AG1-Listener      RemapPipeNames           1                        UInt32
SQLAG1_AG1-Listener      HostRecordTTL            300                      UInt32
SQLAG1_AG1-Listener      RegisterAllProvidersIP   1                        UInt32
SQLAG1_AG1-Listener      PublishPTRRecords        0                        UInt32
SQLAG1_AG1-Listener      ResourceData             {1, 0, 0, 0...}          ByteArray
SQLAG1_AG1-Listener      StatusNetBIOS            0                        UInt32
SQLAG1_AG1-Listener      StatusDNS                0                        UInt32
SQLAG1_AG1-Listener      StatusKerberos           0                        UInt32
SQLAG1_AG1-Listener      CreatingDC               \\DC1.example.com        String
SQLAG1_AG1-Listener      LastDNSUpdateTime        3/26/2014 5:26:05 PM     DateTime
SQLAG1_AG1-Listener      ObjectGUID               4ce880b7ad3c83418435b... String
SQLAG1_AG1-Listener      DnsSuffix                example.com              String
```

Figure 6: Modifying the TTL in Windows PowerShell

4. From the Remote Desktop Gateway (RDGW1), open the Desktop Connection application (mstsc.exe) and connect to the Primary Domain Controller (DC1) in AZ1 using its NetBIOS name (e.g., DC1).
5. Use the credentials of the Domain Admin User and Domain Admin Password to log into the instance.
6. On DC1, open **Server Manager**.
7. Check DNS to ensure all availability group Listener (e.g., AG1-Listener) IP addresses are listed.



Name	Type	Data	Timest
_udp			
DomainDnsZones			
ForestDnsZones			
(same as parent folder)	Start of Authority (SOA)	[65], dc1.example.com., h...	static
(same as parent folder)	Name Server (NS)	dc2.example.com.	static
(same as parent folder)	Name Server (NS)	dc1.example.com.	static
(same as parent folder)	Host (A)	10.0.0.10	3/5/20
(same as parent folder)	Host (A)	10.0.64.10	3/5/20
AG1-Listener	Host (A)	10.0.0.102	3/5/20
AG1-Listener	Host (A)	10.0.64.102	3/5/20
dc1	Host (A)	10.0.0.10	static
DC2	Host (A)	10.0.64.10	static
RDGW1	Host (A)	10.0.41.169	3/5/20
RDGW2	Host (A)	10.0.100.169	3/5/20
WSFCluster1	Host (A)	10.0.64.101	3/5/20
WSFCNode1	Host (A)	10.0.0.100	3/5/20
WSFCNode2	Host (A)	10.0.64.100	3/5/20

Figure 7: Verifying DNS Configuration

Note: Client connectivity to an Availability Group database can be established via the Availability Group listener. The Availability Group Listener (in this case, AG1-Listener) is a virtual network name to which clients can connect. This configuration allows clients to connect to a database without knowing the name of an individual server in the WSFC cluster. The Availability Group listener can share TCP port 1433 with an individual SQL Server instance. However, when running multiple side-by-side SQL Server instances, you will need to use a non-standard port to avoid a port conflict.

The Security Groups and ingress rules created by the CloudFormation template permit all required traffic between WSFC nodes and client connections to TCP port 1433 from the remaining server tiers within the Amazon VPC. See the Appendix for a detailed list of port mappings.

After completing the steps in this section, you will have a Windows Server Failover Clustering (WSFC) cluster and SQL Server 2012 AlwaysOn Availability Group successfully deployed in the AWS cloud.

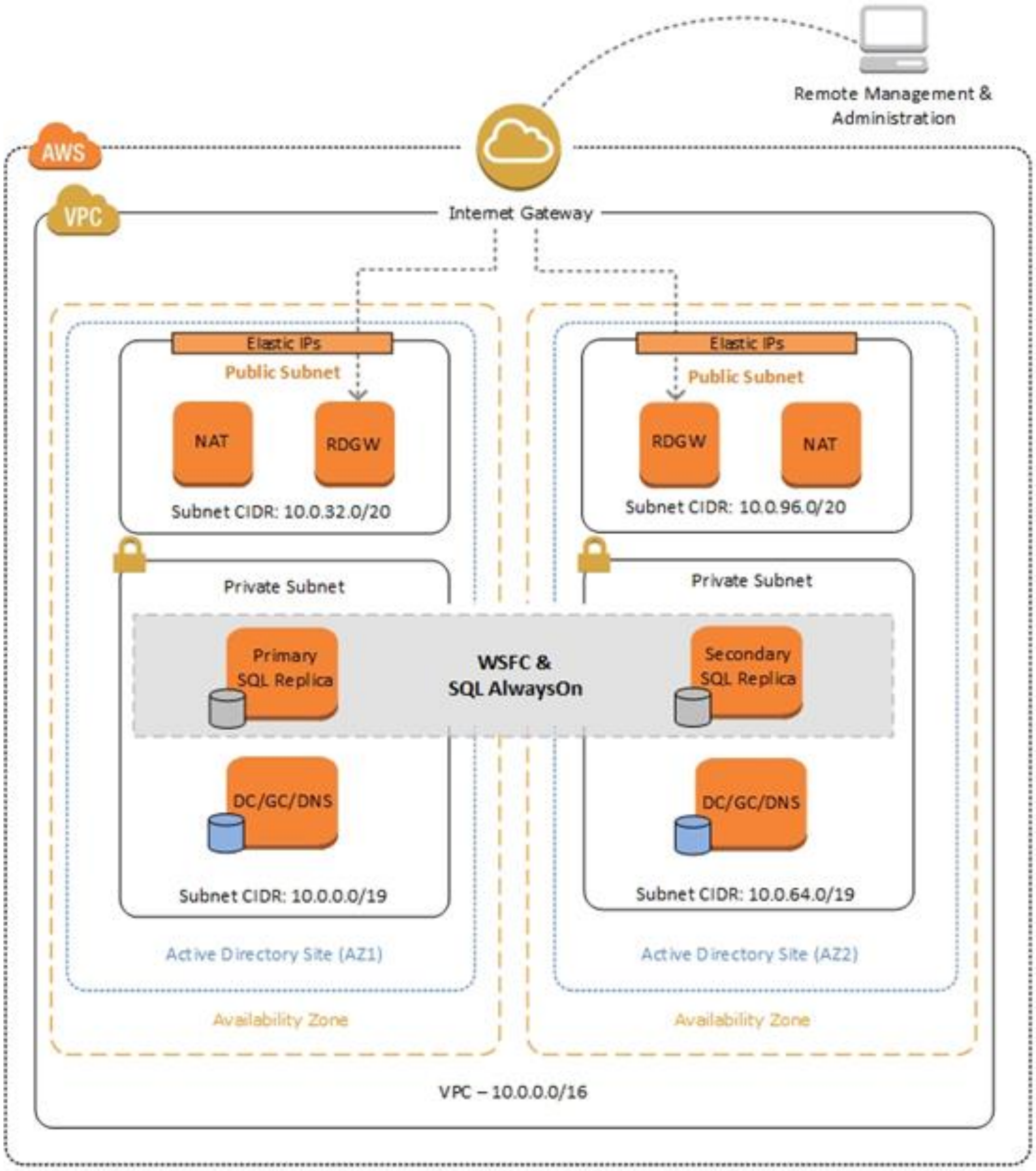
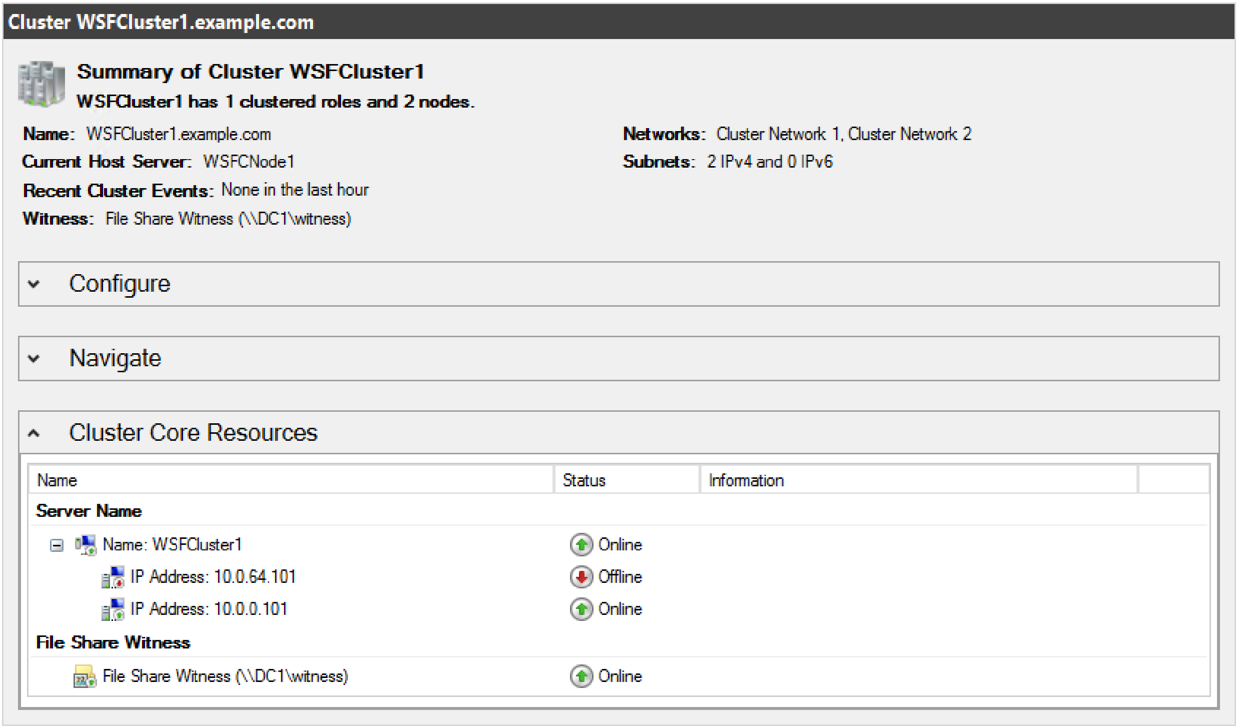


Figure 8: WSFC Cluster and AlwaysOn Availability Group Deployed in the AWS Cloud

Test Your WSFC Cluster and AlwaysOn Availability Group Deployed in the AWS Cloud

Before putting the availability group into production, you should test your deployment and familiarize yourself with the cluster's behavior during a high availability automatic failover or a disaster recovery event.

1. Open the Remote Desktop Connection application (mstsc.exe), connect to the Remote Desktop Gateway (RDGW1) in AZ1, and then connect to the WSFC node (WSFCNode1) in AZ1.
2. On the WSFCNode1 instance, open the **Failover Cluster Manager** to view the **Cluster Core Resources**. Make sure the cluster name (e.g., WSFCluster1), and one of the two listed IP addresses (e.g., IP Address: 10.0.0.101 or IP Address: 10.0.64.101) and the File Share Witness (e.g., \\DC1\witness) are Online.



Cluster WSFCluster1.example.com

Summary of Cluster WSFCluster1

WSFCluster1 has 1 clustered roles and 2 nodes.

Name: WSFCluster1.example.com

Current Host Server: WSFCNode1

Recent Cluster Events: None in the last hour

Witness: File Share Witness (\\DC1\witness)

Networks: Cluster Network 1, Cluster Network 2

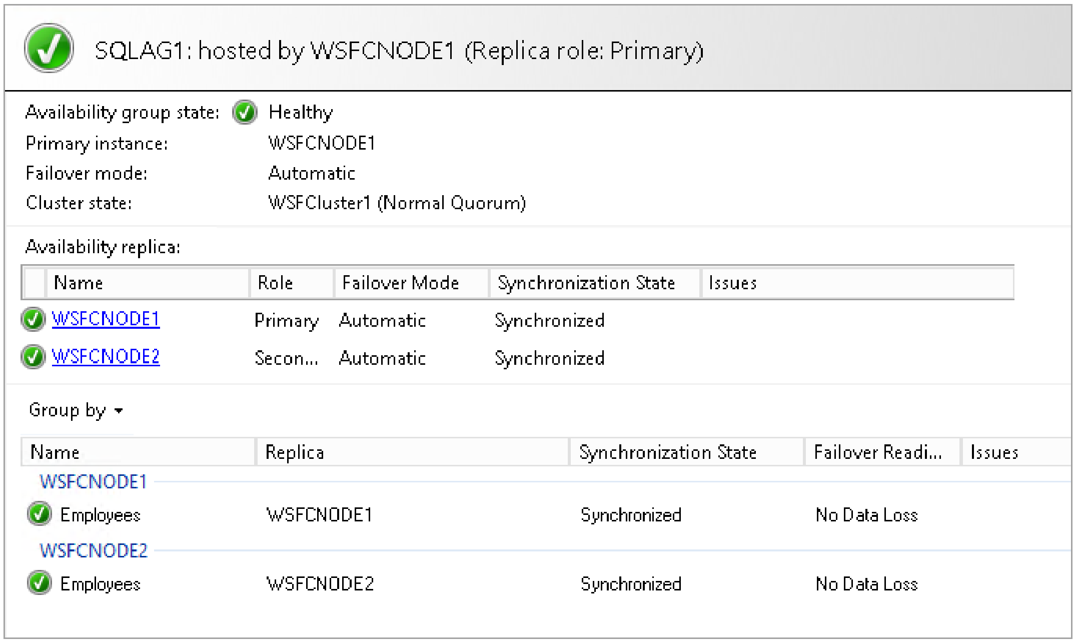
Subnets: 2 IPv4 and 0 IPv6

Cluster Core Resources

Name	Status	Information
Server Name		
Name: WSFCluster1	Online	
IP Address: 10.0.64.101	Offline	
IP Address: 10.0.0.101	Online	
File Share Witness		
File Share Witness (\\DC1\witness)	Online	

Figure 9: Viewing the Failover Cluster Manager

3. Open **SQL Server Management Studio**; in **Object Explorer**, navigate to the **AlwaysOn High Availability** node, and right-click to bring up the Dashboard. Launch the Dashboard for the availability group you created earlier (e.g., SQLAG1).
4. In the Dashboard, view the **Availability Replicas** and make sure their synchronization state is Synchronized**.** Right click on **AlwaysOn High Availability** and select **Show Dashboard.**



SQLAG1: hosted by WSFCNODE1 (Replica role: Primary)

Availability group state:	✓ Healthy
Primary instance:	WSFCNODE1
Failover mode:	Automatic
Cluster state:	WSFCluster1 (Normal Quorum)

Availability replica:

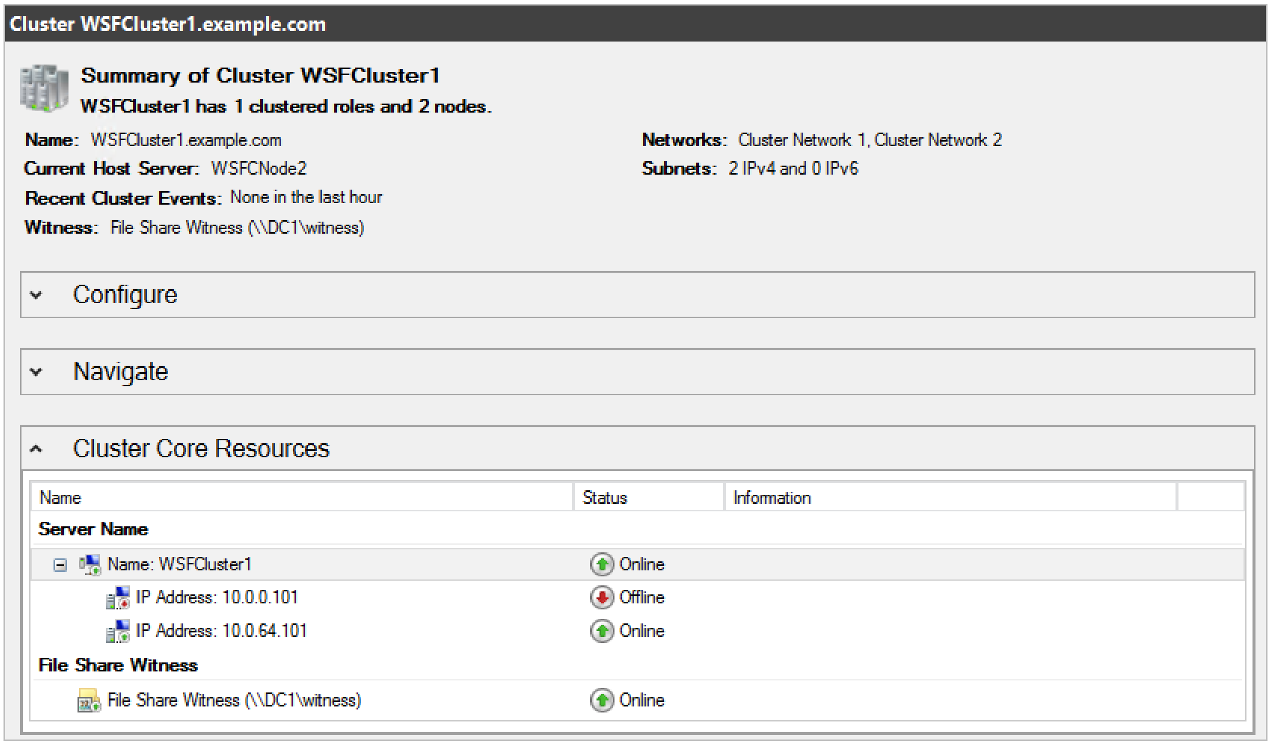
Name	Role	Failover Mode	Synchronization State	Issues
✓ WSFCNODE1	Primary	Automatic	Synchronized	
✓ WSFCNODE2	Secon...	Automatic	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Readi...	Issues
WSFCNODE1				
✓ Employees	WSFCNODE1	Synchronized	No Data Loss	
WSFCNODE2				
✓ Employees	WSFCNODE2	Synchronized	No Data Loss	

Figure 10: Viewing the AlwaysOn High Availability Dashboard with All Nodes Synchronized

5. Make sure that the primary instance and the IP address in the **Cluster Core Resource** window of **Server Manager** are coordinated. In other words, if the primary instance is WSFCNode1, then IP address 10.0.0.101 should be Online. If you need to move the cluster core resources to WSFCNode1, you can do so through PowerShell using the *Get-ClusterGroup 'Cluster Group' | Move-ClusterGroup -Node WSFCNode1* command.
6. Open the **AWS Management Console** and bring up the Amazon EC2 Dashboard.
7. Stop the primary instance (e.g., WSFCNode1).
8. Open the Remote Desktop Connection application (mstsc.exe), connect to the Remote Desktop Gateway (RDGW2) in AZ2, and then connect to your to the WSFC node (WSFCNode2) in AZ2.
9. On the WSFCNode2 instance, use the **Failover Cluster Manager** to view the **Cluster Core Resources**. Note that now the IP address previously Offline (e.g., IP address: 10.0.64.101) is now Online.



Cluster WSFCluster1.example.com

Summary of Cluster WSFCluster1

WSFCluster1 has 1 clustered roles and 2 nodes.

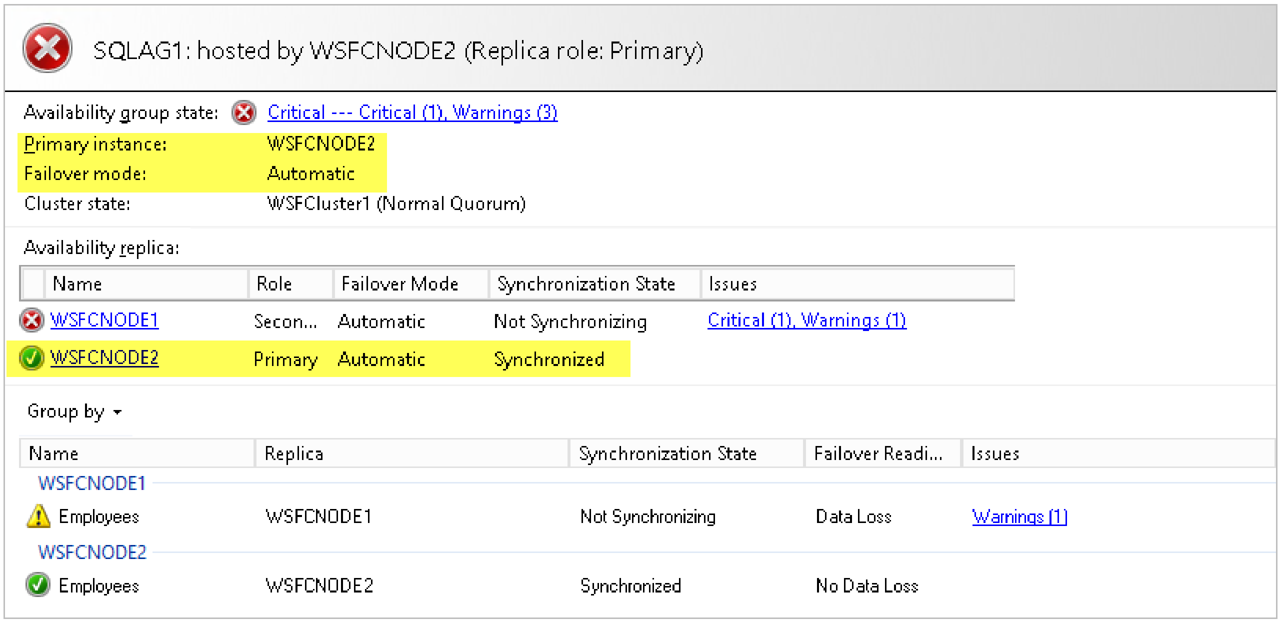
Name: WSFCluster1.example.com
Current Host Server: WSFCNode2
Recent Cluster Events: None in the last hour
Witness: File Share Witness (\\DC1\witness)

Networks: Cluster Network 1, Cluster Network 2
Subnets: 2 IPv4 and 0 IPv6

Name	Status	Information
Server Name		
Name: WSFCluster1	Online	
IP Address: 10.0.0.101	Offline	
IP Address: 10.0.64.101	Online	
File Share Witness		
File Share Witness (\\DC1\witness)	Online	

Figure 11: Viewing the Failover Cluster Manager After WSFCNode1 Goes Down

10. Open **Microsoft SQL Server Management Studio**. In **Object Explorer**, navigate to the **AlwaysOn High Availability** node, and right-click to bring up the Dashboard. Launch the Dashboard for the availability group you created earlier (e.g., SQLAG1).
11. In the Dashboard, view the **Availability Replicas.** Note that now the primary instance has switched to WSFCNode2 and that the Synchronization State of WSFCNode1 is Not Synchronizing**.**



SQLAG1: hosted by WSFCNODE2 (Replica role: Primary)

Availability group state: ✗ Critical --- Critical (1), Warnings (3)
Primary instance: WSFCNODE2
Failover mode: Automatic
Cluster state: WSFCluster1 (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Issues
✗ WSFCNODE1	Secon...	Automatic	Not Synchronizing	Critical (1), Warnings (1)
✓ WSFCNODE2	Primary	Automatic	Synchronized	

Group by ▾

Name	Replica	Synchronization State	Failover Readi...	Issues
WSFCNODE1				
⚠ Employees	WSFCNODE1	Not Synchronizing	Data Loss	Warnings (1)
WSFCNODE2				
✓ Employees	WSFCNODE2	Synchronized	No Data Loss	

Figure 12: Viewing the AlwaysOn High Availability Dashboard with WSFCNode1 Offline

12. At this point, you can start the WSFCNode1 instance again in the Amazon EC2 Dashboard. Once the instance is online, use the **Failover Availability Group** wizard in the **Availability Group** Dashboard and switch the primary instance back to WSFCNode1.

Note
We recommend you use *MultiSubnetFailover = True* in your SQL client connection string. MultiSubnetFailover enables faster failover for all Availability Groups in SQL Server 2012 and will significantly reduce failover time for single and multi-subnet AlwaysOn topologies. If you have legacy clients that need to connect to an Availability Group listener and cannot use the *MultiSubnetFailover* property, we recommend that you change the RegisterAllProvidersIP setting to 0 by using the **Set-ClusterParameter** cmdlet.

Conclusion

WSFC provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the AWS cloud, while SQL Server AlwaysOn leverages WSFC to increase application availability.[2]

In this guide, we walked you through the steps to implement the necessary infrastructure in the AWS cloud to set up and configure WSFC and an AlwaysOn Availability Group. The resulting sample implementation supports the following scenarios:

- Protect from failure of a single instance.
- Provide automatic failover between the cluster nodes.

[2] Windows Server Failover Clustering (WSFC) with SQL Server

- Protect from failure of the instance placed in the secondary Availability Zone (AZ2) and automatically failover to AZ1.

However, the sample implementation does not provide automatic failover in every case. For example, the loss of AZ1, which contains the primary node and file share witness, would prevent automatic failover to AZ2. This is because the cluster would fail as it loses quorum. Manual disaster recovery steps that include restarting the cluster service and forcing quorum on WSFCNode2 are necessary to restore application availability in this scenario.

We recommend you consult the Microsoft SQL Server documentation and customize some of the steps described in this guide or add additional ones (e.g., deploy additional cluster nodes and configure them as readable secondary replicas) to deploy a solution that best meets your high availability (HA) and disaster recovery (DR) application availability requirements.

Further Reading

- Microsoft on AWS:
  - http://aws.amazon.com/microsoft/
- Amazon EC2 Windows Guide:
  - http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/
- AWS Windows and .NET Developer Center:
  - http://aws.amazon.com/net
- Microsoft License Mobility:
  - http://aws.amazon.com/windows/mslicensemobility
- Whitepapers/Articles:
  - Active Directory Domain Services on the AWS Cloud
    http://docs.aws.amazon.com/quickstart/latest/active-directory-ds/welcome.html
  - Microsoft SharePoint Server 2013 on the AWS Cloud: Quick Start Deployment Guide
    http://docs.aws.amazon.com/quickstart/latest/sharepoint/
  - Secure Microsoft Applications on AWS
    http://media.amazonwebservices.com/AWS_Microsoft_Platform_Security.pdf

Appendix

Amazon EC2 Security Group configuration

AWS provides a set of building blocks (e.g., Amazon EC2 and Amazon VPC) that customers can use to provision infrastructure for their applications. In this model, some security capabilities such as physical security are the responsibility of AWS and are highlighted in the AWS security whitepaper. Other areas such as controlling access to applications fall on the application developer and the tools provided in the Microsoft platform.

If you have followed the scripted deployment option in Part 2, the necessary security groups for SQL Server are configured for you by the provided AWS CloudFormation Template and are listed here for your reference:

Subsystem Port Mappings

Subsystem	AssociatedWith	Inbound Interface	Port(s)
WSFCSecurityGroup	WSFCNode1, WSFCNode2	PrivSub1CIDR (the private subnet in AZ1)	ICMP-1, TCP135, TCP137, UDP137, TCP445, TCP1433, TCP3343, UDP3343, TCP5022, TCP49152-65535, UDP49152-65535
		PrivSub2CIDR (the private subnet in AZ2)	ICMP-1, TCP135, TCP137, UDP137, TCP445, TCP1433, TCP3343, UDP3343, TCP5022, TCP49152-65535, UDP49152-65535
WSFCClientSecurityGroup	WSFCNode1, WSFCNode2	PrivSub1CIDR	TCP1433
		PrivSub2CIDR	TCP1433

Additional Resources

We have a supplementary template available that can be deployed into this architecture. It creates one Windows Server 2012–based instance to host a sample application that can test your cluster and allow you to see the failover occur between the different nodes in your deployment.

To view a demo of this application being used to test the cluster, watch the Windows Server Failover Clustering and SQL Server 2012 AlwaysOn Availability Groups in AWS cloud video on YouTube.

This server should be launched into one of the private subnets in your deployment. Once deployed, RDP to the server to run the application. There will be a folder on the desktop called SQLBlasterDemo containing the application SQLDAGTester.exe.

To launch this AWS CloudFormation template in the US West region, **launch the Quick Start**.

Note
You are responsible for the cost of AWS services used while running this Quick Start Reference Deployment. The cost for running the template with default settings is approximately $5.50 an hour, so you can complete the initial deployment for about $15.00. See the pricing pages of the AWS services you will be using for full details.

Send Us Your Feedback

Please post your feedback or questions on the AWS Quick Start Discussion Forum.

Document Revisions

Date	Change	In sections
April 2015	Updated the storage configuration on the WSFC nodes.	Storage on the WSFC Nodes
March 2015	Optimized the underlying Amazon VPC design to support expansion and to reduce complexity.	Architecture diagram and template updates
November 2014	In the sample template, changed the default type for **NATInstanceType** to **t2.small** to support the EU (Frankfurt) region.	Automation (template customization table)

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.