Implementing Microsoft Windows Server Failover Clustering (WSFC) and SQL Server 2012 AlwaysOn Availability Groups in the AWS Cloud

David Pae, Ulf Schoo

June 2013

(Please consult http://aws.amazon.com/windows/ for the latest version of this article)

Abstract

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying Microsoft Windows-based workloads, including Microsoft Windows Server Failover Clustering (WSFC) clusters and SQL Server 2012 AlwaysOn Availability Groups, on its reliable and secure cloud infrastructure.

WSFC clusters and AlwaysOn Availability Groups, together with Active Directory Domain Services (AD DS) and Domain Name Server (DNS) functionality, provide the underpinnings for many enterprise-class Microsoft technology–based solutions including Microsoft SharePoint 2013 and .NET applications.

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 the configuration steps involved to repeatedly and reliably deploy WSFC clusters and AlwaysOn Availability Groups in the AWS cloud.

Introduction

"A Windows Server Failover Clustering (WSFC) cluster is a group of independent servers that work together to increase the availability of applications and services. SQL Server 2012 takes advantage of WSFC services and capabilities to support AlwaysOn Availability Groups and SQL Server Failover Cluster Instances.[1]Windows Server Failover Clustering provides infrastructure features that support the high-availability and disaster recovery scenarios of hosted server applications such as Microsoft SQL Server, Microsoft Exchange, and SharePoint 2013. If a cluster node or service fails, the services that were hosted on that node can be automatically or manually transferred to another available node in a process known as failover."[2]

This guide discusses architectural considerations and configuration steps when launching the necessary AWS services such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC) to run a WSFC cluster across different subnets and Availability Zones.[3] In addition, the guide provides instructions for installing and configuring the WSFC cluster and an AlwaysOn Availability Group.

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

Note: AlwaysOn Availability Groups is a feature supported by SQL Server 2012 Enterprise Edition. You can deploy SQL Server 2012 Enterprise Edition and many other Windows server application licenses using the License Mobility through Software Assurance program.

Before You Get Started

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

---

[1] SQL Server Failover Cluster Instances (FCIs) are currently not supported on the AWS cloud
[2] http://msdn.microsoft.com/en-us/library/hh270278.aspx
[3] Running WSFC nodes in the same subnet is currently not supported on the AWS cloud

- Amazon EC2
- Amazon VPC
- Windows Server 2008 R2 and Windows Server 2012
- Windows Server Active Directory and DNS
- Windows Server Failover Clustering (WSFC)
- SQL Server 2012 AlwaysOn Availability Groups

Implementing WSFC and SQL Server 2012 AlwaysOn Availability Groups in AWS

Implementing a WSFC cluster in the AWS cloud, which is a prerequisite for deploying an AlwaysOn Availability Group, is not different from deploying both technologies in an on-premise setting as long as you meet two key requirements:

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

Keeping these two key requirements in mind, we provide instructions on how to install and configure the Windows Server Failover Cluster and an AlwaysOn Availability Group, and we call out any AWS-specific considerations.

Specifically, we walk you through the steps necessary to configure a 2-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 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 guide discusses the following topics as they relate to setting up the failover clustering environment:

- **Part 1: Launch & Configure the Virtual Network and Active Directory Infrastructure**
    - Set up the virtual network for the WSFC cluster within AWS, including subnets in two Availability Zones.
    - Configure private and public routes.
    - Launch Windows Server 2008 R2 Amazon Machine Images (AMIs) and set up and configure Active Directory and DNS.
    - Enable administrative ingress and egress into your VPC via Remote Desktop Gateway and NAT instances.
    - Configure Amazon EC2 Security Groups to control network traffic between WSFC cluster nodes and Active Directory.
- **Part 2: Launch & Configure the WSFC Cluster Nodes**
    - Create the WSFC cluster.

- **Part 3: Install & Configure a SQL Server 2012 AlwaysOn Availability Group**

    - Set up SQL Server 2012 Enterprise Edition.
    - Enable AlwaysOn High Availability.
    - Create an availability group.

When you have finished, 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:
    - 4 private subnets and 1 public subnet
    - 1 private route
    - 1 Windows Server 2008 R2–based Remote Desktop Gateway (RDGW) instance and 1 Linux-based NAT instance to enable administrative ingress and egress
    - 2 Elastic IP Addresses associated with the NAT and RDGW instances
    - 1 Windows Server 2008 R2–based instance to host the Active Directory
    - 1 Windows Server 2008 R2–based instance to host the WSFC Node and SQL Server 2012 Instance
- Security Groups to control the secure flow of traffic between the instances deployed in the Amazon VPC4

[4] Please refer to the Appendix for further details on Security Groups and Windows Firewall configurations

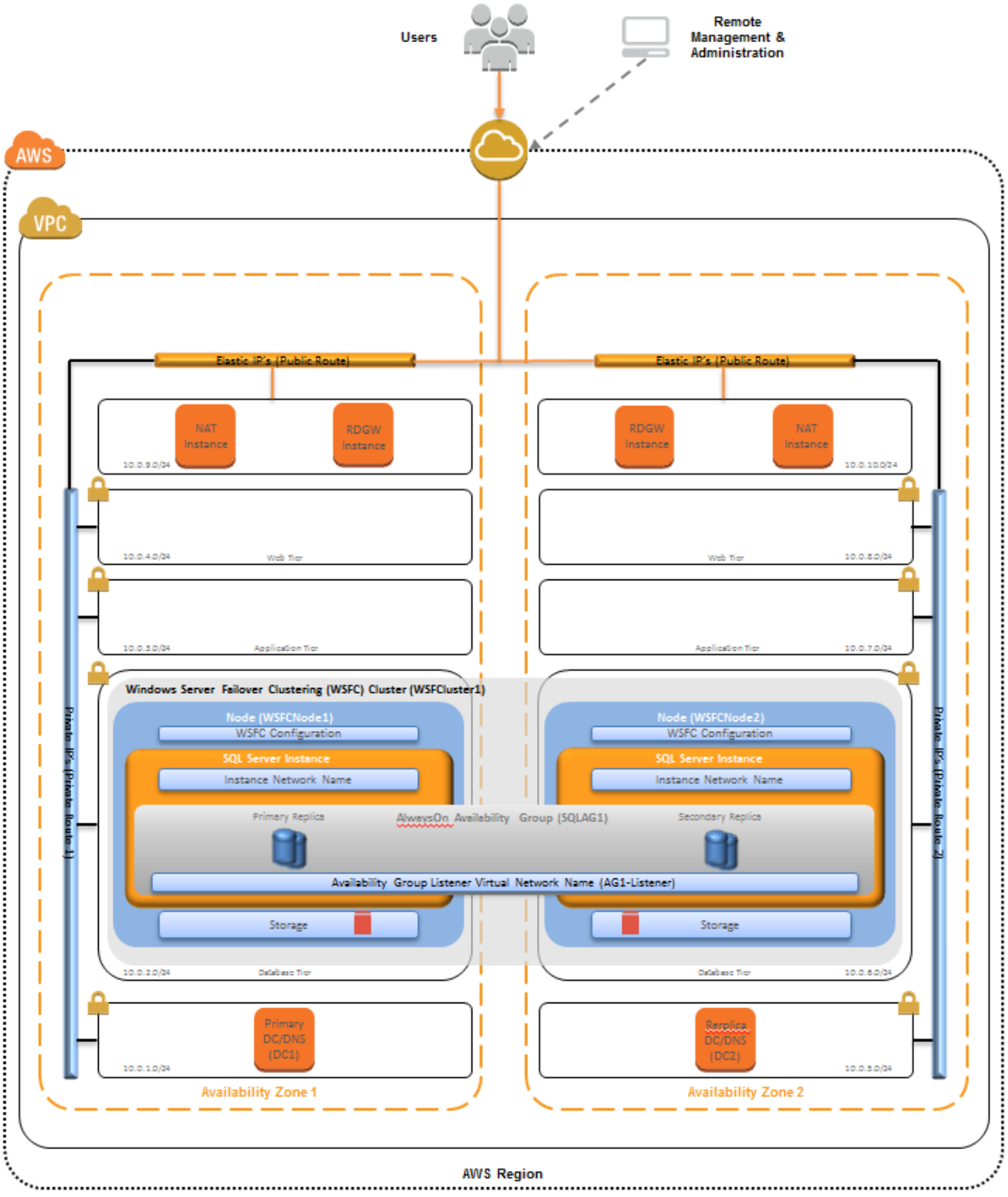


Figure 1: WSFC cluster setup across different subnets and Availability Zones

Part 1: Launch & Configure the Virtual Network & Active Directory Infrastructure

Let's start with the necessary infrastructure and virtual network setup to provide the environment in which you instantiate and configure your servers and database.

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and regions. 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.

Scripted Deployment

With this guide, we provide two sample AWS CloudFormation templates designed to help you deploy the necessary and correctly configured infrastructure predictably and repeatedly. At this point, you can simply launch the provided sample Template-1, which will provision and configure all the resources described hereafter. If you choose the scripted deployment option, no further manual configuration of the virtual network and Active Directory environment is required. Template-1 takes approximately **1 hour and 20 minutes** to complete all configuration tasks described in this section.

To launch the AWS CloudFormation template in the US-East Region, click **Launch Stack**.

Template Customization

Sample Template-1 allows for rich customization of 23 defined parameters at template launch. You can modify those parameters, change the default values, or, if you choose to edit the code of the template itself, create an entirely new set of parameters based on your specific deployment scenario. The Template-1 parameters include the following default values.

Parameter	Default	Description
KeyPairName	*<User Provides>*	Public/private key pairs allow you to connect securely to your instance after it launches.
ADServer1InstanceType	m1.xlarge	Amazon EC2 instance type for the first Active Directory instance.
ADServer2InstanceType	m1.xlarge	Amazon EC2 instance type for the second Active Directory instance.
ADServer1NetBIOSName	DC1	NetBIOS name of the first Active Directory server (up to 15 characters).
ADServer2NetBIOSName	DC2	NetBIOS name of the second Active Directory server (up to 15 characters).
ADServer1PrivateIp	10.0.1.10	Fixed private IP for the first Active Directory server located in AZ1.
ADServer2PrivateIp	10.0.5.10	Fixed private IP for the second Active Directory server located in AZ2.
NATInstanceType	m1.small	Amazon EC2 instance type for the NAT instances.
RDGWInstanceType	m1.large	Amazon EC2 instance type for the Remote Desktop Gateway instances.
DomainDNSName	contoso.com	Fully qualified domain name (FQDN) of the forest root domain; e.g., corp.example.com.
DomainNetBIOSName	Contoso	NetBIOS name of the domain (up to 15 characters) for users of earlier versions of Windows; e.g., CORP.
RestoreModePassword	Password123	Password for a separate administrator account when the domain controller is in restore mode. Must be at least 8 characters containing letters, numbers, and

		symbols.
DomainAdminUser	StackAdmin	User name for the account that is added as domain administrator. This is separate from the default "administrator" account.
DomainAdminPassword	Password123	Password for the domain admin user. Must be at least 8 characters containing letters and numbers.
SQLAdminUser	sqlsa	User name for the SQL Server Admin Account. This Account is a Domain User.
SQLAdminPassword	Password123	Password for the SQL admin user. Must be at least 8 characters containing letters and numbers
PublicSubnet1CIDR	10.0.9.0/24	CIDR block for the Public subnet located in AZ1.
PublicSubnet2CIDR	10.0.10.0/24	CIDR block for the Public subnet located in AZ2.
PrivateSubnet1CIDR	10.0.1.0/24	CIDR block for the Private Subnet 1 located in AZ1.
PrivateSubnet2CIDR	10.0.2.0/24	CIDR block for the Private Subnet 2 located in AZ1.
PrivateSubnet3CIDR	10.0.3.0/24	CIDR block for the Private Subnet 3 located in AZ1.
PrivateSubnet4CIDR	10.0.4.0/24	CIDR block for the Private Subnet 4 located in AZ1.
PrivateSubnet5CIDR	10.0.5.0/24	CIDR block for the Private Subnet 5 located in AZ2.
PrivateSubnet6CIDR	10.0.6.0/24	CIDR block for the Private Subnet 6 located in AZ2.
PrivateSubnet7CIDR	10.0.7.0/24	CIDR block for the Private Subnet 7 located in AZ2.
PrivateSubnet8CIDR	10.0.8.0/24	CIDR block for the Private Subnet 8 located in AZ2.
VPCCIDR	10.0.0.0/16	CIDR block for the VPC.

Figure 2: Template-1 parameters

After successfully launching Template-1, you will have the following resources of your architecture launched and properly configured, ready to deploy a WSFC cluster and an AlwaysOn Availability Group:

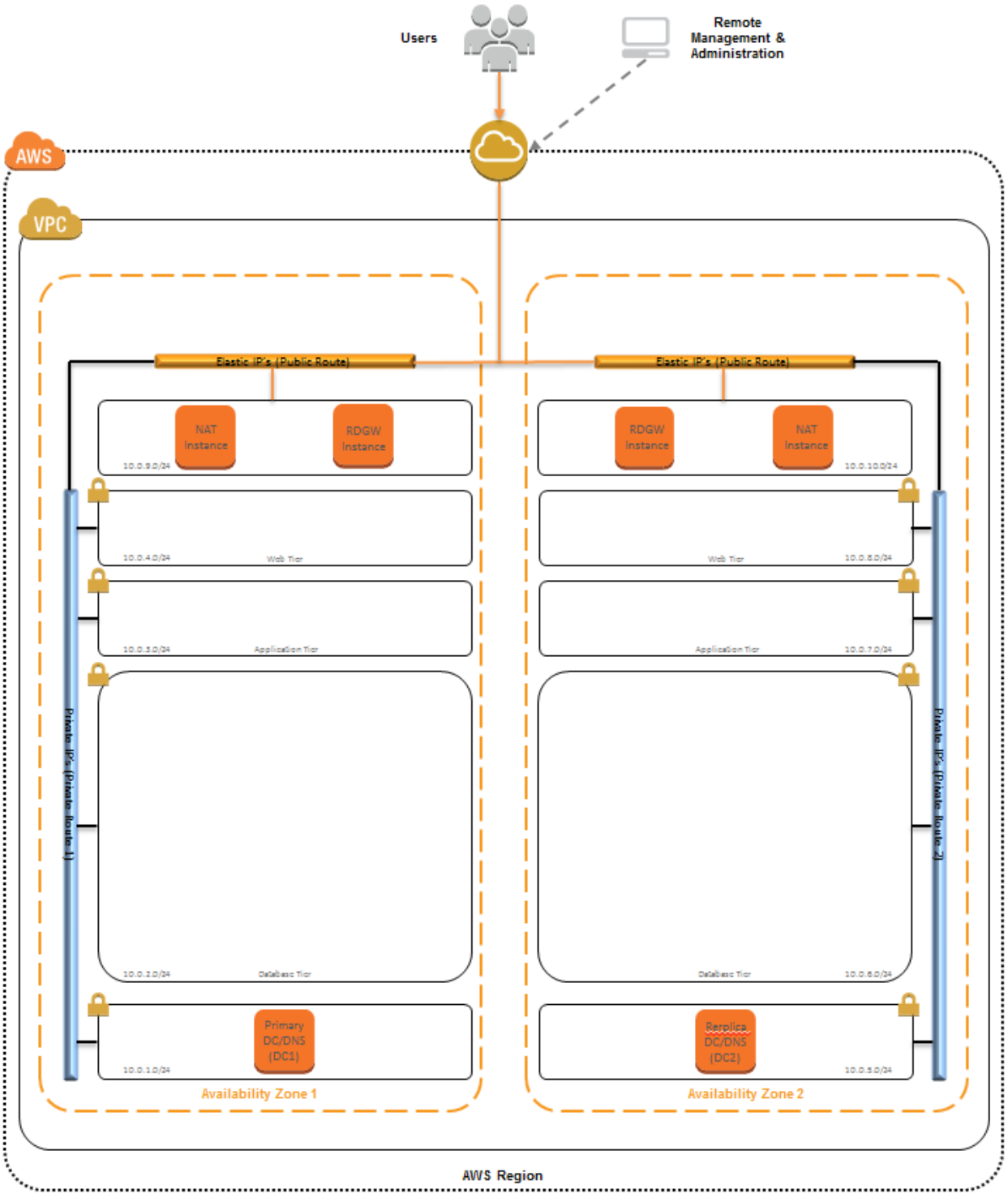


Figure 3: Virtual network, Active Directory, and administrative ingress and egress

Part 2: Launch & Configure the WSFC Cluster Nodes

After we have successfully deployed the necessary infrastructure, virtual network setup, and Active Directory, we move on and configure the WSFC nodes.

Scripted Deployment

Template-2, the second of the two sample AWS CloudFormation templates provided with this guide, is designed to help you deploy the WSFC Nodes into our architecture. Specifically, the template performs the following setup and configuration tasks for you:

- Deploys Windows Server 2008 R2–based instances as WSFC nodes into their respective subnets
- Renames the instance to a friendly NetBIOS name of your choice
- Joins the Windows instances to the domain
- Adds the SQL Service Account (e.g., sqlsa) to the local Administrator Group
- Installs the WSFC feature
- Downloads a SQL Server 2012 Enterprise Edition Evaluation copy and a required hotfix from Microsoft onto the instance
- Opens the TCP ports 1433, 1434, 4022, 5022, and 135 on the Windows Firewall

At this point, you can simply launch the provided sample Template-2, which will provision the Amazon EC2 instances. You'll configure these instances according to the instructions in this section. Template-2 takes approximately **50 minutes** to complete all the Amazon EC2 provisioning tasks described here.

To launch the AWS CloudFormation template in the US-East Region, click **Launch Stack**.

Template Customization

Sample Template-2 provides 20 defined parameters at template launch. You must modify those parameters, change the default values, or create an entirely new set of parameters based on your specific deployment scenario and values previously defined when launching Template-1. The Template-2 parameters include the following default values.

Parameter	Default	Description
KeyPairName	*<User Provides>*	Public/private key pairs allow you to connect securely to your instance after it launches.
WSFCNode1InstanceType	M2.4xlarge	Amazon EC2 instance type for the first WSFC node.
WSFCNode2InstanceType	M2.4xlarge	Amazon EC2 instance type for the second WSFC node.
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).
ADServer1PrivateIp	10.0.1.10	Fixed private IP for the first Active Directory server located in AZ1.
ADServer2PrivateIp	10.0.5.10	Fixed private IP for the second Active Directory server located in AZ2.
WSFCNode1PrivateIp	10.0.2.100	Fixed private IP for the first WSFC node located in AZ1.
WSFCNode2PrivateIp	10.0.6.100	Fixed private IP for the second WSFC node located in AZ2.
DomainDNSName	contoso.com	Fully qualified domain name (FQDN) of the forest root domain; e.g., corp.example.com.
DomainNetBIOSName	Contoso	NetBIOS name of the domain (up to 15 characters) for users of earlier versions of Windows; e.g., CORP.
DomainAdminUser	StackAdmin	User name for the account that is added as domain administrator. This is separate from the default "administrator" account.
DomainAdminPassword	Password123	Password for the domain admin user. Must be at least 8 characters containing

		letters and numbers.
SQLServiceAccount	sqlsa	User name for the SQL Server Service Account. This account is a Domain User.
SQLServicePassword	Password123	Password for the SQL Service account. Must be at least 8 characters containing letters and numbers.
DomainMemberSGID	*<User Provides>*	ID of the Domain Member Security Group.
WSFCServerSecurityGroupID	*<User Provides>*	ID of the WSFC and AlwaysOn AG Security Group.
WSFCNode1Subnet	*<User Provides>*	ID of the subnet you want to provision the first WSFC node into.
WSFCNode2Subnet	*<User Provides>*	ID of the subnet you want to provision the second WSFC node into.
VPC	10.0.0.0/16	ID of the VPC.

Figure 4: Template-2 parameters

Create the WSFC Cluster

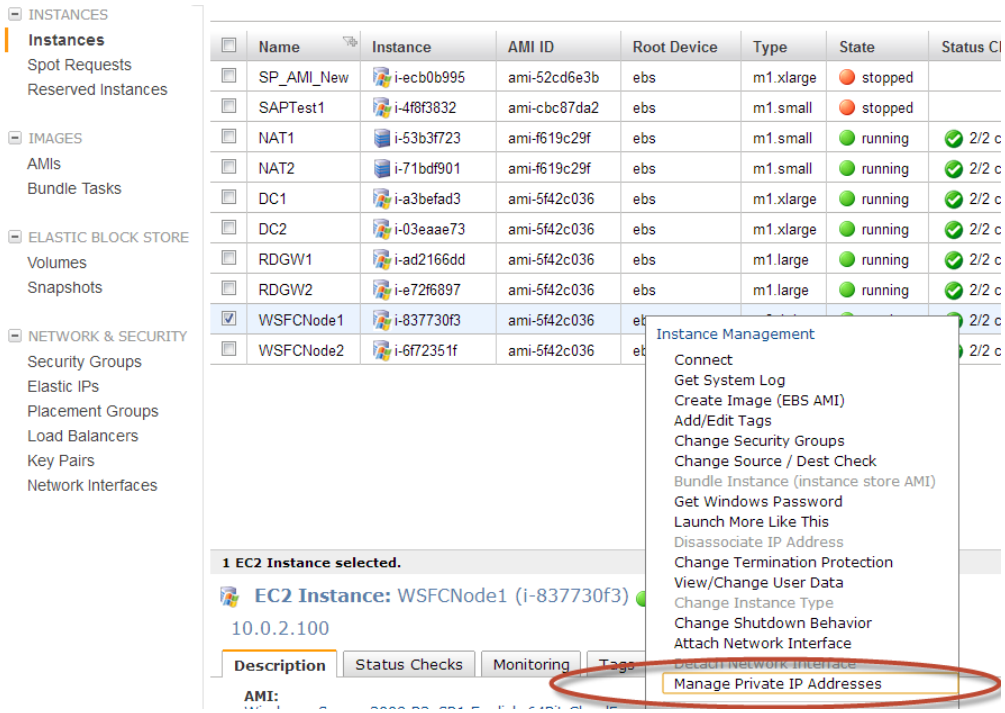
1. Before we can launch **Failover Cluster Manager**, we first have to add two secondary private IP addresses to the Amazon EC2 instances that host the WSFC nodes (WSFCNode1 and WSFCNode2). Those secondary private IP addresses will be assigned to the WSFC cluster and the AlwaysOn Availability Group listener.

   As we have chosen to assign the fixed IP address of 10.0.2.100 to WSFCNode1 and 10.0.6.100 to WSFCNode2, we will assign the secondary private IP addresses of X.X.X.101 and X.X.X.102 to the nodes, respectively.

Server NetBIOS Name	IP Address	Availability Zone
WSFCNode1	10.0.2.100 *(Primary)* 10.0.2.101 (assigned the WSFC cluster) 10.0.2.102 (assigned the availability group Listener)	AZ1 (e.g., us-east-1a)
WSFCNode2	10.0.6.100 *(Primary)* 10.0.6.101 (assigned the WSFC cluster) 10.0.6.102 (assigned the availability group Listener)	AZ2 (e.g., us-east-1b)

Figure 5: WSFC node primary and secondary private IP assignment

2. Open the Amazon EC2 console, select the first WSFC node (WSFCNode1), and then right-click to select **Manage private IP addresses**.



3. In the **Manage Private IP Addresses** dialog box, select **Assign a secondary IP address**.

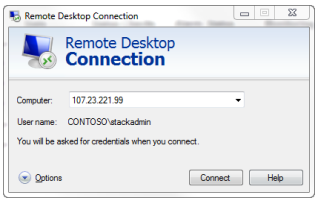
4. After entering the secondary private IP addresses, click **Yes, Update**.

5. Repeat for the second WSFC node (WSFCNode2).

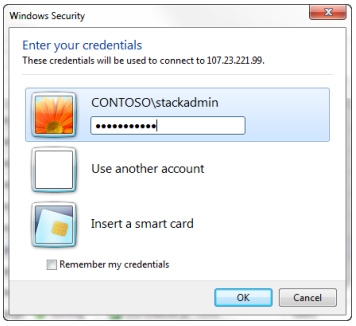
   Note: You can also perform the above steps 2–5 via the command line interface (CLI) or Windows PowerShell.

```
PS C:\Program Files (x86)\AWS Tools\PowerShell\AWSPowerShell> Register-EC2PrivateIPAddresse -NetworkInterfaceId "eni-663
58e0b" -PrivateIpAddresses "10.0.2.101"
```

6. Open the Remote Desktop Connection application (mstsc.exe) and connect to the Remote Desktop Gateway (RDGW1) in AZ1 using its Elastic IP address (e.g., 107.23.221.99).



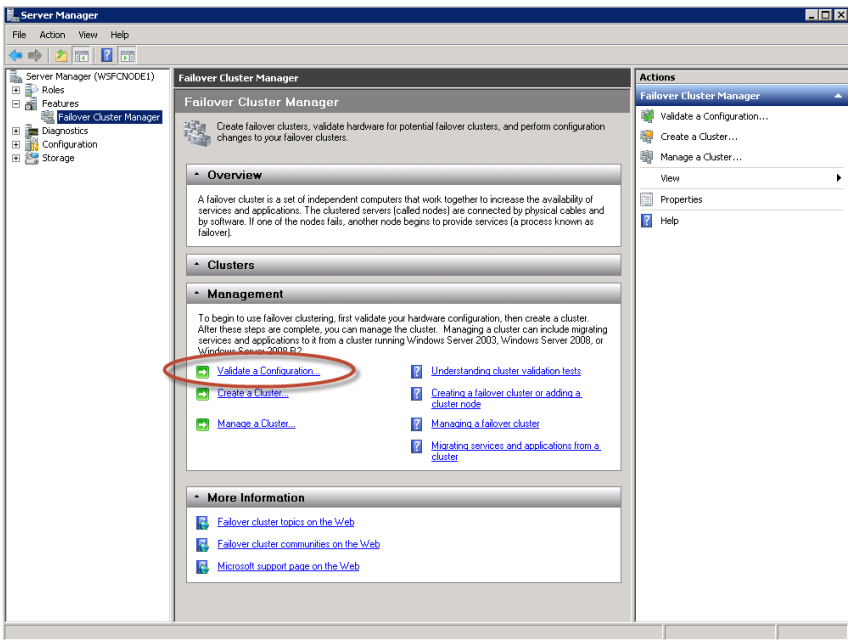
7. Your Remote Desktop Gateway is domain joined. Log in using the credentials of the Domain Admin user and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123).



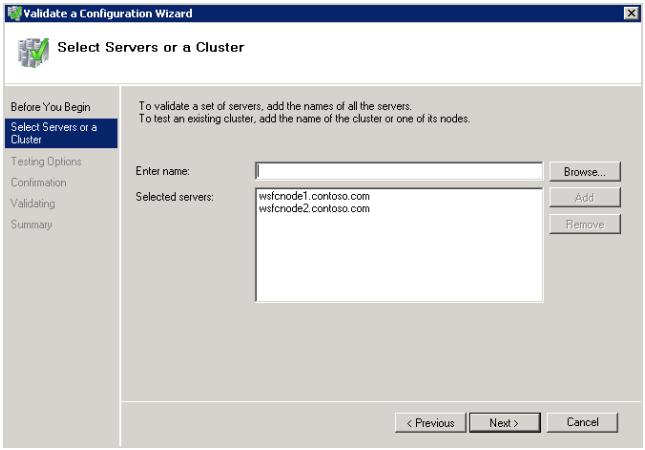
8. After successfully logging into the Remote Desktop Gateway, open the Remote Desktop Connection application, and connect to the WSFC node (WSFCNode1) in AZ1 using its NetBIOS name (e.g., WSFCNode1).

Note: You will be connecting to several instances from your Remote Desktop Gateway. We recommend pinning the Remote Desktop Connection application to your task bar to streamline this process.

9. Use the credentials of the Domain Admin User and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.
10. On the WSFCNode1 instance, open **Server Manager**, navigate to **Failover Cluster Manager**, and start the **Validate a Configuration** wizard.

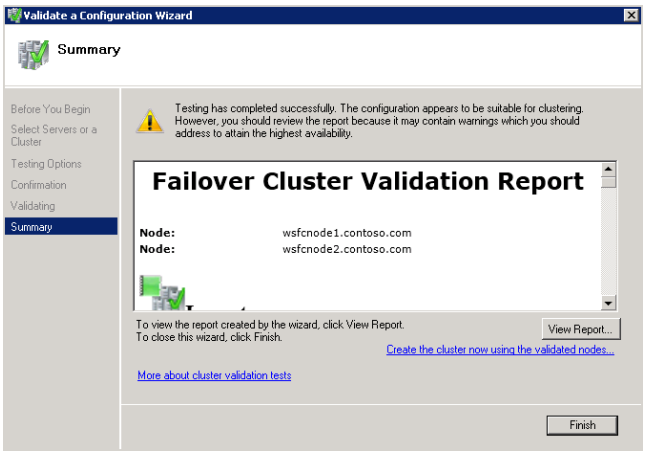


11. Select both WSFC node servers (e.g., WSFCNode1, WSFCNode2), and then click **Next**.

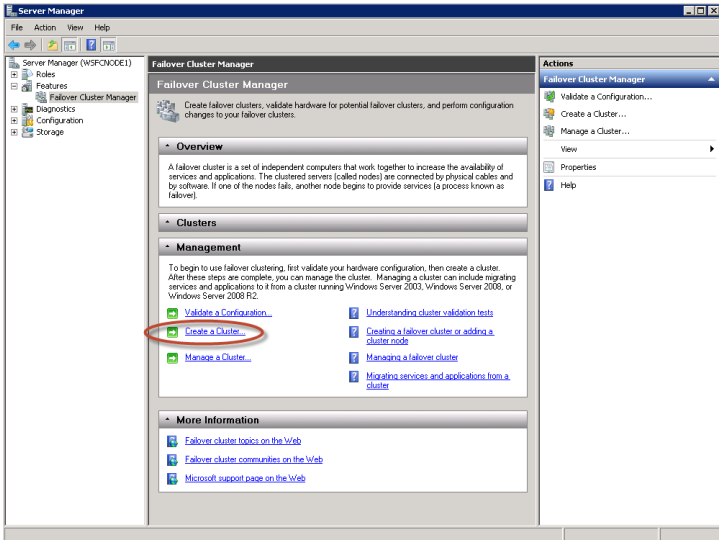


12. Select **Run all tests (recommended)**.

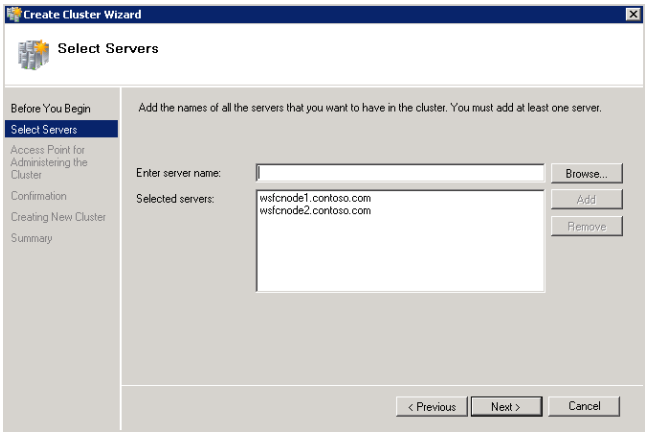
13. View the Failover Cluster Validation Report and click **Finish**. Address any errors reported and rerun the **Validate a Configuration** wizard.



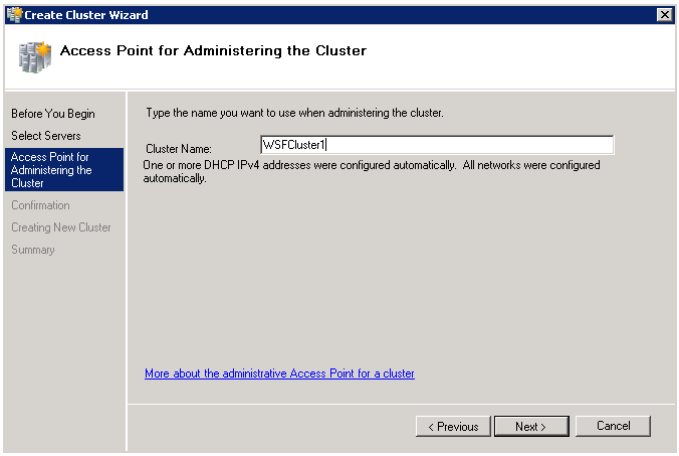
14. Proceed to **Create a Cluster**. This starts the **Create Cluster** wizard.



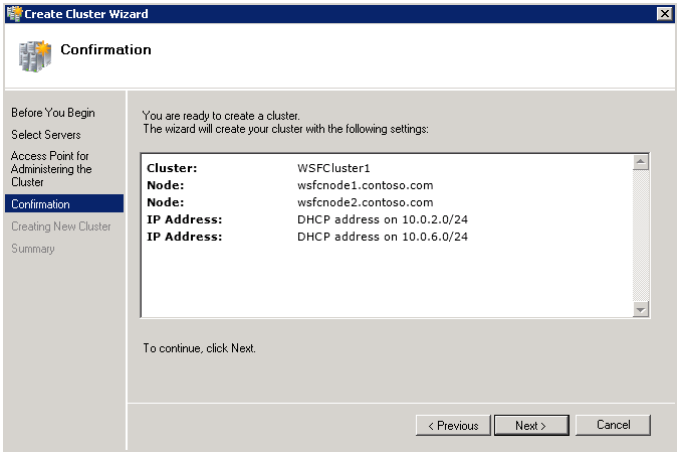
15. Select both WSFC node servers (e.g., WSFCNode1, WSFCNode2), and then click **Next**.



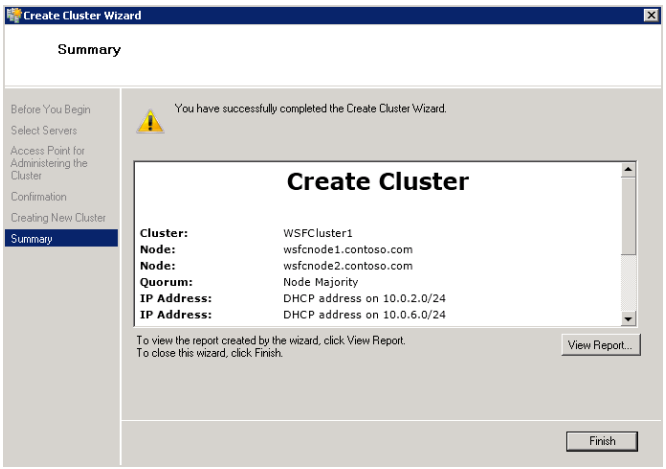
16. Select a **Cluster Name** (e.g., WSFCluster1), and then click **Next**.



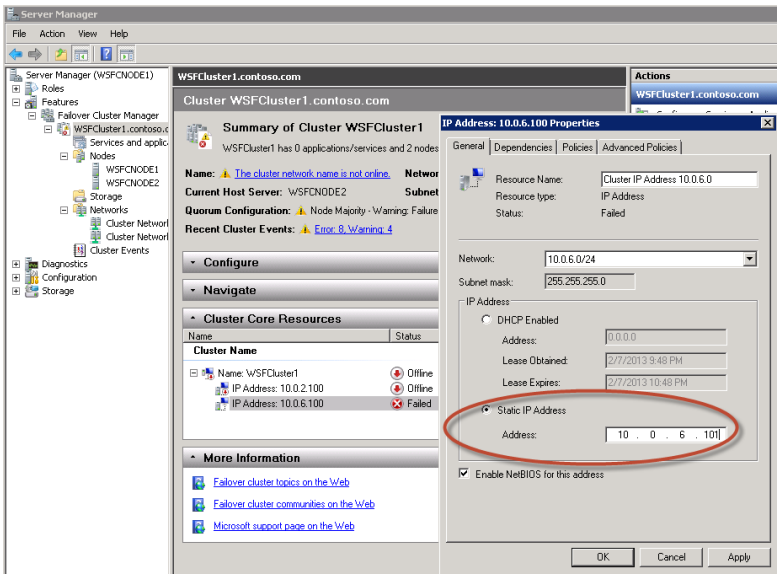
17. Review and confirm the settings you have chosen, and click **Next** to create the cluster.



18. With Step 17, the **Create Cluster** wizard finished creating your cluster. Click **Finish** to continue configuring your cluster.

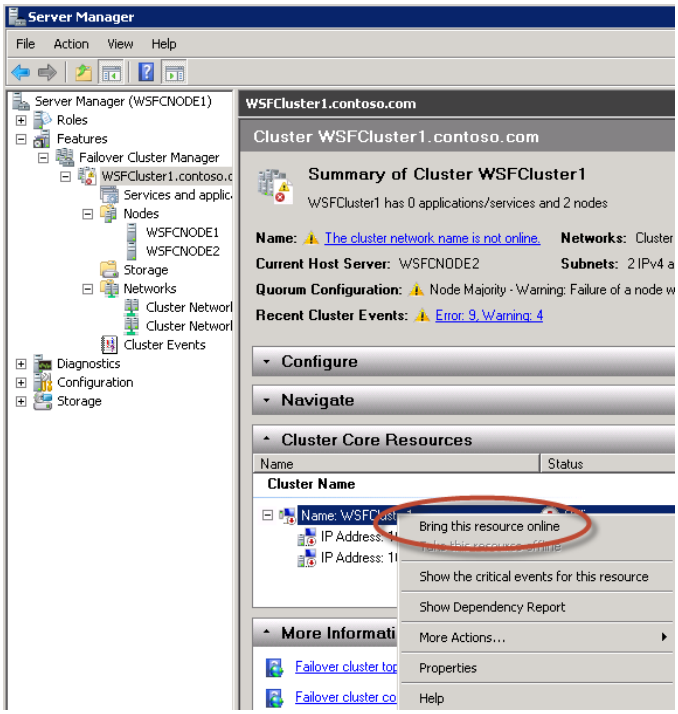


19. In **Server Manager**, navigate to the cluster you just created (e.g., WSFCluster1.contoso.com). Note the error message indicating that the cluster is not online and the IP addresses under the **Cluster Name** are either in an "Offline" or "Failed" state. Double-click or right-click to bring up the IP Address properties. Change the IP address from **DHCP enabled**" to **Static IP Address** and assign the first of the two secondary private IP addresses you have created earlier in Steps 3 and 4 (e.g., 10.0.6.101). Click **Apply**.

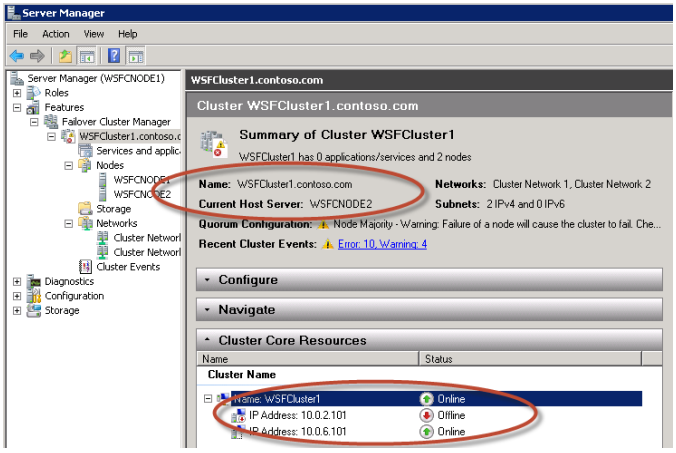


20. Repeat for the second IP address (e.g., 10.0.2.101).

21. Select the **Cluster Name**, right-click it and select **Bring this resource online**.



22. Note how the cluster network name (e.g., WSFCluster1.contoso.com) and the IP Address for the current host server (e.g., WSFCNode2) is now online. With this, we are half-way there. Note the warning next to the **Quorum Configuration** saying that the cluster is currently running in **Node Majority** mode and that failure of a node will cause the cluster to fail. We will deal with this next when we create a witness share and change **Quorum Configuration** to **Node and File Share Majority**.

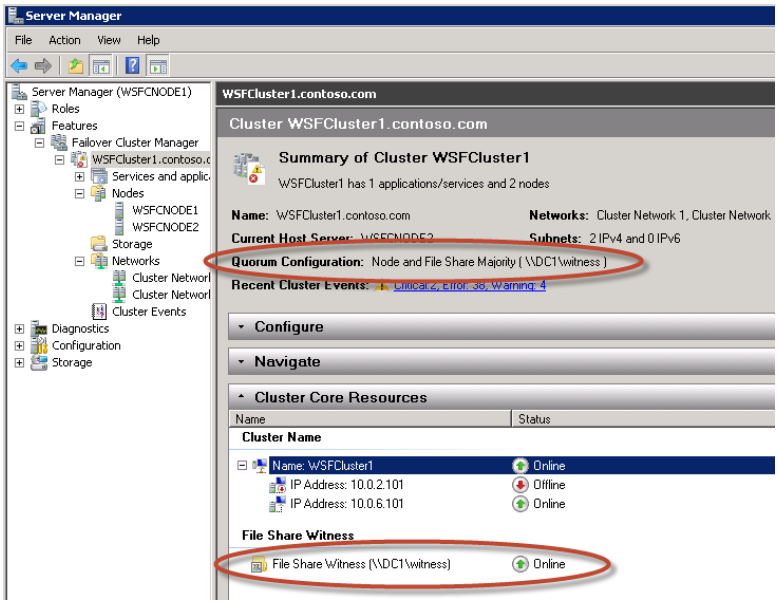


23. From Remote Desktop Gateway open the Remote Desktop Connection application and connect to the Primary Domain Controller (DC1) in AZ1 using its NetBIOS name (e.g., DC1).
24. Use the credentials of the Domain Admin User and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.

25. On DC1, create and share a file folder named “Witness”. Give the **Cluster Name$** (e.g., WSFCluster1$) account Read/Write permissions to the file share.
26. On either of the **Cluster Nodes** (e.g., WSFCNode1 or WSFCNode2), run Windows PowerShell as Administrator.
27. Set **Quorum Configuration** to **Node and File Share Majority** pointing to the share you created in Step 25.

```
PS C:\Windows\system32> set-clusterquorum -NodeAndFileShareMajority "\\DC1\witness"
Cluster                                  QuorumResource                           QuorumType
-------                                  --------------                           ----------
WSFCluster1                              File Share Witness                       NodeAndFileShareMajority
```

28. After this step is complete, the **File Share Witness** is now online and the **Quorum Configuration** warning disappeared, thus completing the WSFC cluster setup and configuration.



29. For Windows Server 2008 R2, install the required hotfix (Article ID: 2494036) from http://support.microsoft.com/kb/2494036. If you have chosen the scripted deployment option, this hotfix has already been downloaded onto the instance and you can install it from C:\MicrosoftDownloads\Fix348347\430171_intl_x64_zip.exe. After you apply this hotfix, you can configure a cluster node that does not have quorum votes. For a more detailed discussion about quorum mode considerations, please read on.

After completing the steps described in this section, you will have the following resources of your architecture launched and configured:

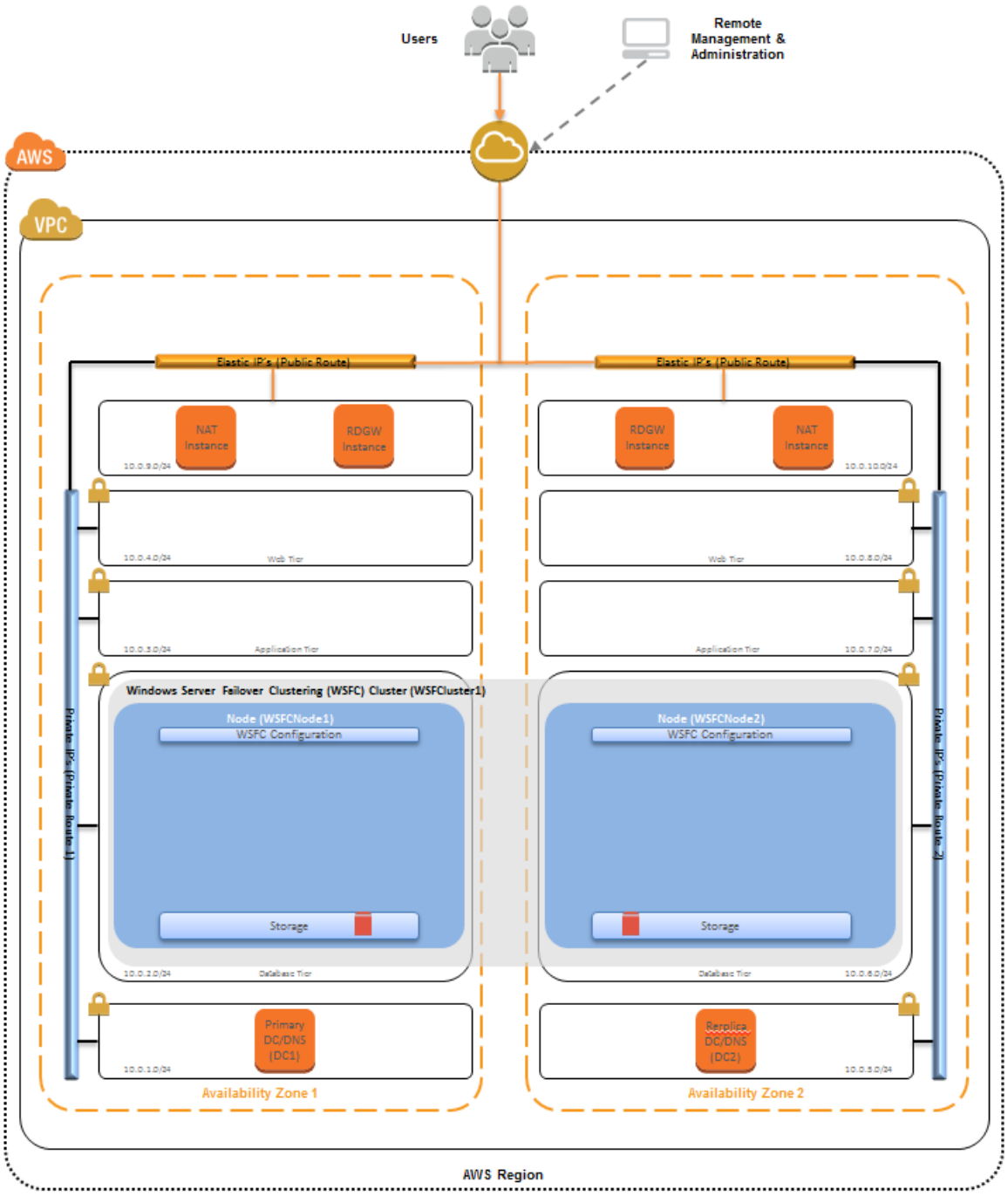


Figure 6: Infrastructure and WSFC nodes

Part 3: Install & Configure a SQL Server 2012 AlwaysOn Availability Group

After successfully launching and configuring the WSFC cluster, we can move on. The next steps are to download and install the SQL Server 2012 Enterprise Edition software from Microsoft, enable AlwaysOn high availability for your database, and create a new availability group. If you have chosen the scripted deployment option, the script downloads the SQL Server 2012 Enterprise Edition trial software from the Microsoft download site onto the instance for you.

Note: AWS does not provide installation media for Microsoft software. If you use this guide to set up a test or evaluation environment, download a trial version at http://www.microsoft.com/sqlserver/en/us/default.aspx. For a production deployment, use your volume licensing software and mobilize the license as described in the License Mobility through Software Assurance program.

Set Up SQL Server 2012 Enterprise Edition

1. From the Remote Desktop Gateway (RDGW1), open the Remote Desktop Connection application (mstsc.exe) and connect to the Primary Domain Controller (DC1) in AZ1 using its NetBIOS name (e.g., DC1).
2. Use the credentials of the Domain Admin User and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.
3. On DC1, open **Server Manager**.
4. Using Active Directory Users and Computers, create a SQL Service account user (e.g., contoso\sqlsa) as a Domain User. For this particular user, we recommend clearing **User must change password at next login** and checking **Password never expires**.
5. Create and share a folder called "Replica" on DC1 and give the SQL Service account (e.g., contoso\sqlsa) Read/Write permissions to the share.
6. From the Remote Desktop Gateway, open the Remote Desktop Connection application and connect to the first cluster node in AZ1 using its NetBIOS name (e.g., WSFCNode1).
7. Use the credentials of the Domain Admin User and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.
8. On WSFCNode1, download a trial version of SQL Server 2012 Enterprise Edition from http://www.microsoft.com/sqlserver/en/us/default.aspx, or download and mount an ISO of your volume licensing media.
9. Run setup.exe from the location where you saved the installation software download in the previous step. This launches the SQL Server Installation Center.

   Note: If you have chosen the scripted deployment option, the SQL Server 2012 Enterprise Edition trial software is already downloaded for you onto the instance. Please run setup.exe from C:\MicrosoftDownloads\x64\SQLFULL_x64_ENU.

10. In the SQL Server Installation Center, select the "Installation" topic and then click **New SQL Server stand-alone installation or add features to an existing installation**. This launches the **SQL Server Setup** wizard.

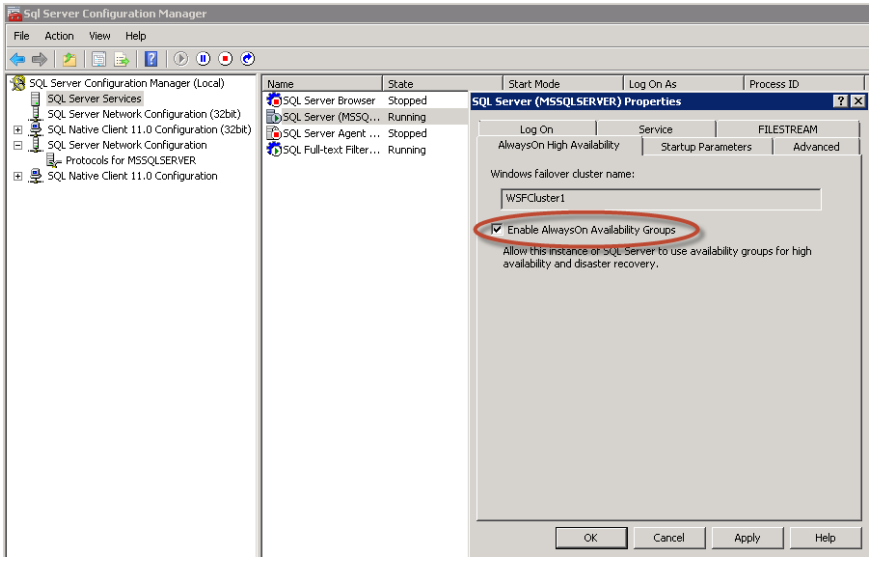
11. Follow the **SQL Server Setup** wizard and take the following actions:

Setup Wizard Page	Action	Comments
Setup Support Rules	**OK**	
Product Key	**Next**	The default action on this page is **Next**, which installs an Evaluation copy of SQL Server 2012 Enterprise Edition. If you install SQL Server 2012 Enterprise Edition for production use, please provide at this stage the product key obtained through your volume licensing agreement to mobilize the license as described in the License Mobility through Software Assurance program.
License Terms	Check **I accept the license terms** + **Next**	
Product Updates	**Next**	
Install Setup Files	**Next**	
Setup Support Rules	**Next**	If this is a clean installation, you will most likely receive a warning indicating that the **SQL Server Setup** wizard detected that the Windows Firewall is enabled. Do not disable the Windows Firewall. Please refer to the Appendix for information on how to configure the Windows Firewall to enable an AlwaysOn Availability Group
Setup Role	**Next**	"SQL Server Feature Installation"
Feature Selection	Select Features + <Next>	• Database Engine Services • SQL Server Replication • Full text and semantic…. • SQL Server Data Tools • Client Tools Connectivity • Document Components • Management Tools Basic/Complete • …any additional features you may need for your specific deployment
Installation Rules	**Next**	
Instance Configuration	**Next**	
Disk Space Requirements	**Next**	
Server Configuration	Update Account Name/Password + <Next>	Use the SQL Service Account created in Step 4 above for: • SQL Server Agent • SQL Server Database Engine
Database Engine Configuration	"Specify SQL Server administrator" <*Add Current User*>	Add other users as needed
Error Reporting	**Next**	
Installation Configuration Rules	**Next**	
Ready to Install	**Install**	
Installation Progress		
Complete	**Close**	

12. Repeat steps 6–11 above on the second cluster node (e.g., WSFCNode2).

Enable AlwaysOn High Availability

1. From the Remote Desktop Gateway (RDGW1), open the Remote Desktop Connection application (mstsc.exe) and connect to the first cluster node in AZ1 using its NetBIOS name (e.g., WSFCNode1).
2. Use the credentials of the Domain Admin User and Domain Admin Password (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.
3. Open **SQL Server Configuration Manager**.
4. Bring up the **Properties** of the SQL Server service.
5. On the **AlwaysOn High Availability** tab, check **Enable AlwaysOn Availability Groups** and click **Apply**.

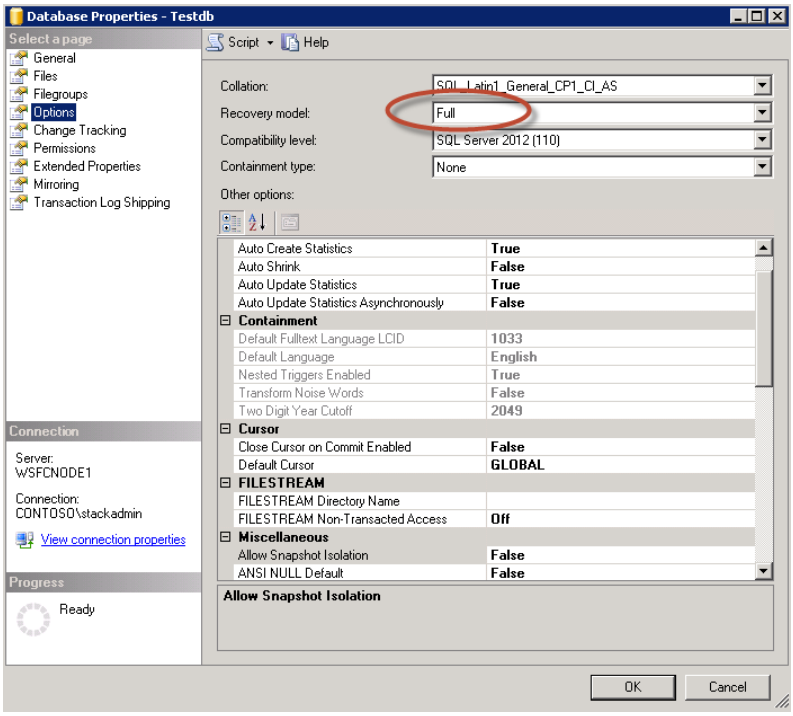


6. Restart the SQL Server (MSSQLSERVER) service.
7. Repeat Steps 1–6 above on the second cluster node (e.g., WSFCNode2).

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.

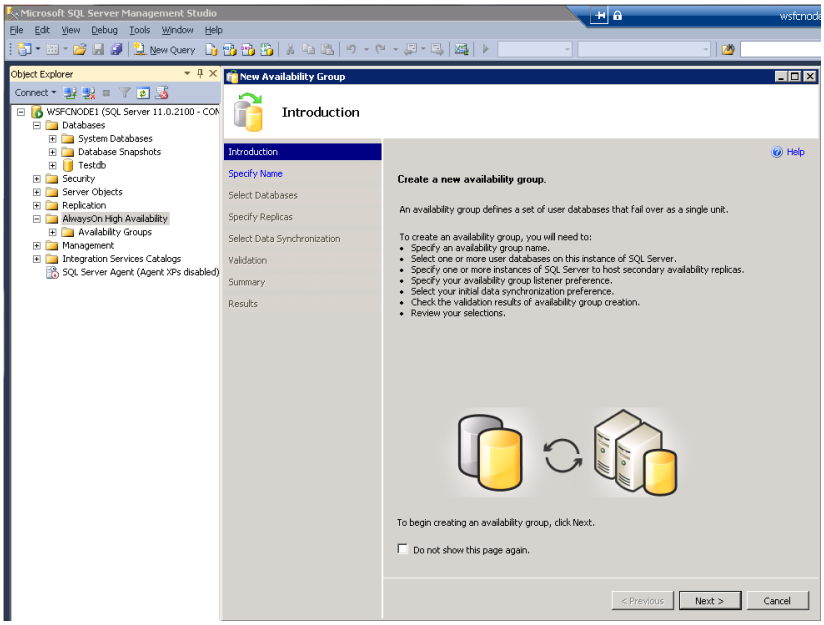
3. Set the **Recovery model** on the database to **Full**.



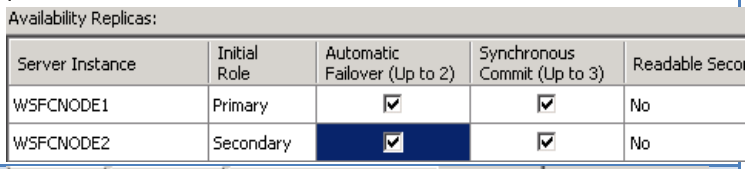
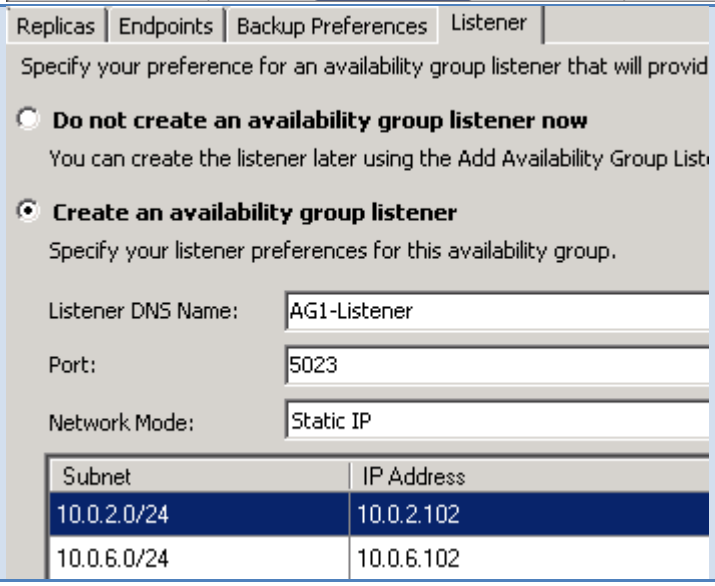
4. Backup the database

Create an availability group

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



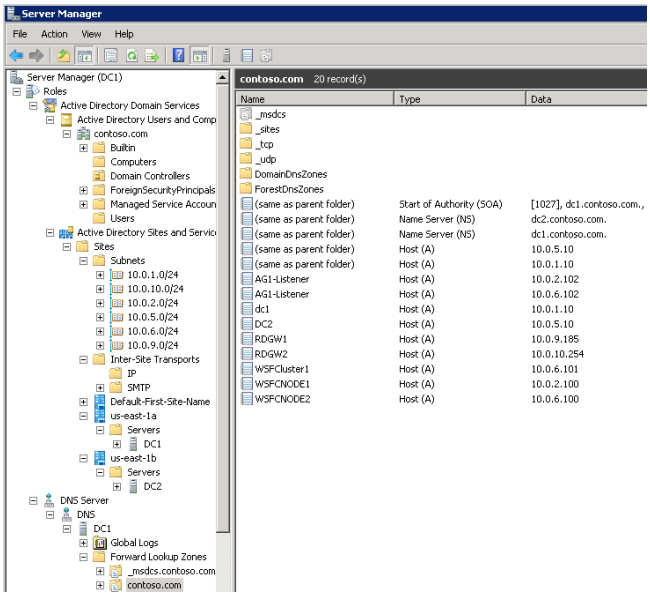
2. Follow the **New Availability Group** wizard and take the following actions:

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 in the previous paragraph + **Next**.	
Specify Replicas	Add the second cluster node (e.g., WSFCNode2) and select **Automatic Failover**	. 
Specify Replicas	On the **Listener** tab, select **Create an availability group listener**, provide a **Listener DNS Name** (e.g., AG1-Listener), and then specify the TPC port used by this listener (e.g., **5023**) and add the two subnets into which you have deployed the cluster nodes and a corresponding IPv4 address. **Note**: We are going to use the second of the secondary private IP addresses we assigned earlier to the nodes (e.g., 10.0.2.102 and 10.0.6.102).	
Select Initial Data Synchronization	Select **Full** + **Next**.	Specify the file share you created earlier on DC1 (e.g., \\DC1\Replica)
Validation	**Next**	Make sure the results show **Success** for all the validation steps
Summary	**Finish**	
Results	**Close**	

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

```
PS C:\Windows\system32> Get-ClusterResource

Name                                State                               Group                               ResourceType
----                                -----                               -----                               ------------
Cluster IP Address                  Online                              Cluster Group                       IP Address
Cluster IP Address 10.0.6.0         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 Availability Group
SQLAG1_10.0.2.102                   Online                              SQLAG1                              IP Address
SQLAG1_10.0.6.102                   Offline                             SQLAG1                              IP Address
SQLAG1_AG1-Listener                 Online                              SQLAG1                              Network Name


PS C:\Windows\system32> 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                 RemapPipeNames                      1                                   UInt32
SQLAG1_AG1-Listener                 HostRecordTTL                       1200                                UInt32
SQLAG1_AG1-Listener                 RegisterAllProvidersIP              1                                   UInt32
SQLAG1_AG1-Listener                 PublishPTRRecords                   0                                   UInt32
SQLAG1_AG1-Listener                 TimerCallbackAdditionalThreshold    5                                   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.contoso.com                   String
SQLAG1_AG1-Listener                 LastDNSUpdateTime                   2/8/2013 10:50:18 PM                DateTime
SQLAG1_AG1-Listener                 ObjectGUID                          09e388b2d28d2e4888169Oebe2dab253    String


PS C:\Windows\system32> Get-ClusterResource "SQLAG1_AG1-Listener" | Set-ClusterParameter HostRecordTTL 300

PS C:\Windows\system32> 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                 RemapPipeNames                      1                                   UInt32
SQLAG1_AG1-Listener                 HostRecordTTL                       300                                 UInt32
SQLAG1_AG1-Listener                 RegisterAllProvidersIP              1                                   UInt32
SQLAG1_AG1-Listener                 PublishPTRRecords                   0                                   UInt32
SQLAG1_AG1-Listener                 TimerCallbackAdditionalThreshold    5                                   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.contoso.com                   String
SQLAG1_AG1-Listener                 LastDNSUpdateTime                   2/8/2013 10:50:18 PM                DateTime
SQLAG1_AG1-Listener                 ObjectGUID                          09e388b2d28d2e4888169Oebe2dab253    String


PS C:\Windows\system32>
```

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 (e.g., UID: Contoso\StackAdmin and Password: Password123) to log into the instance.
6. On DC1, open **Server Manager**.
7. Check DNS to ensure all availability group Listeners (e.g., AG1-Listener) IP addresses are listed.



8. This completes the setup of the AlwaysOn Availability Groups.

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:

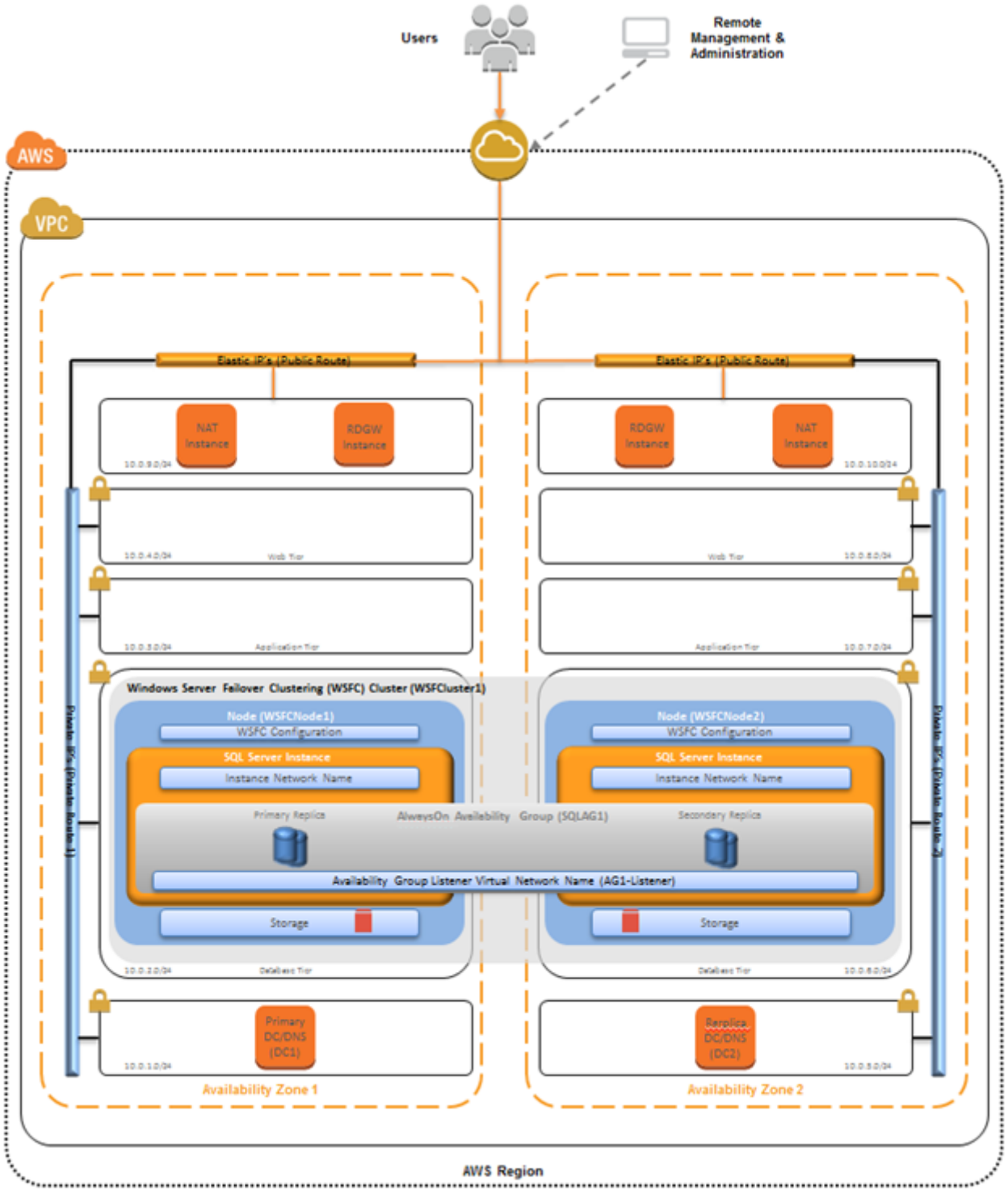
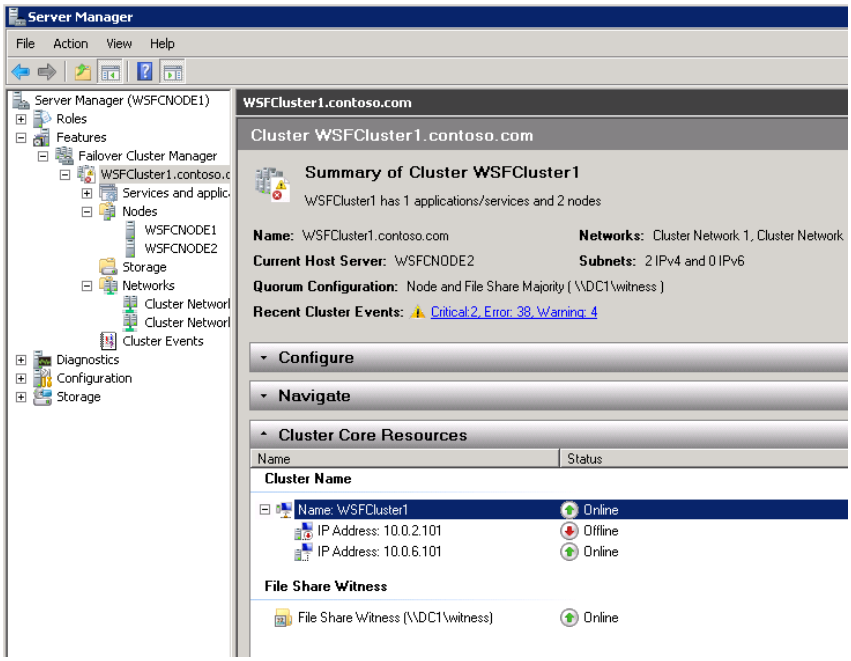


Figure 7: WSFC cluster and AlwaysOn Availability Group deployed in the AWS cloud

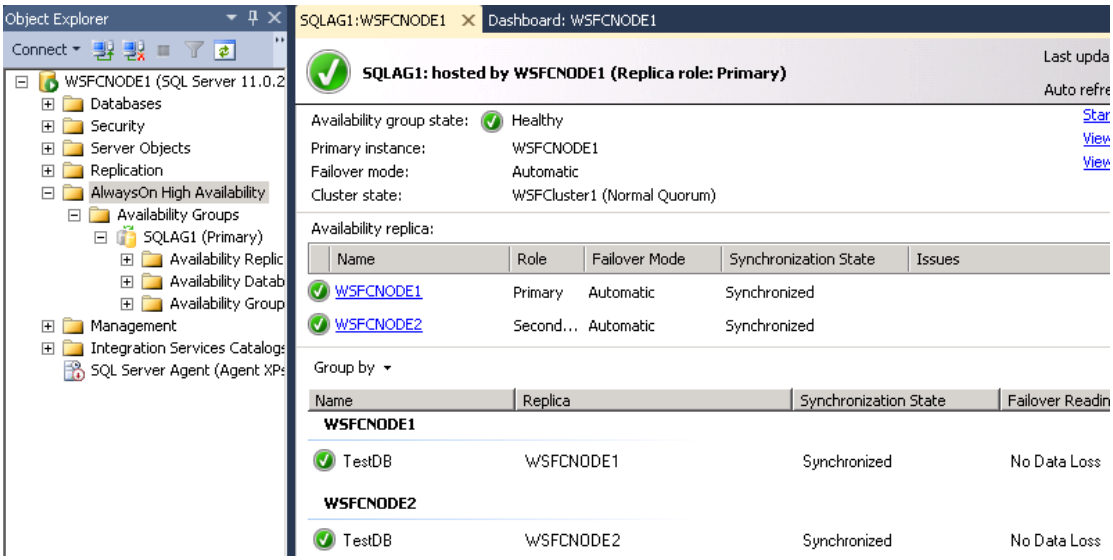
Testing Your WSFC Cluster and AlwaysOn Availability Group Deployed in the AWS Cloud

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

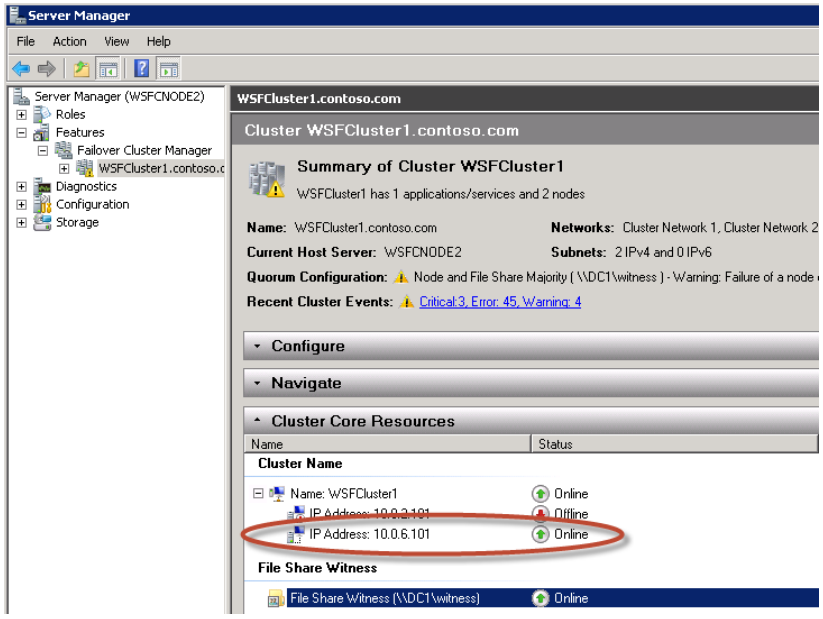
1. Open the Remote Desktop Connection application (mstsc.exe), connect to the Remote Desktop Gateway (RDGW1) in AZ1, and then connect to your to the WSFC node (WSFCNode1) in AZ1.
2. On the WSFCNode1 instance, open **Server Manager**, navigate to **Failover Cluster Manager** to view the **Cluster Core Resources**. Make sure the cluster name (e.g., WSFCluster1), one of the two listed IP addresses (e.g., IP Address: 10.0.2.101 or IP Address: 10.0.6.101) and the File Share Witness (e.g., \\DC1\witness) are Online.



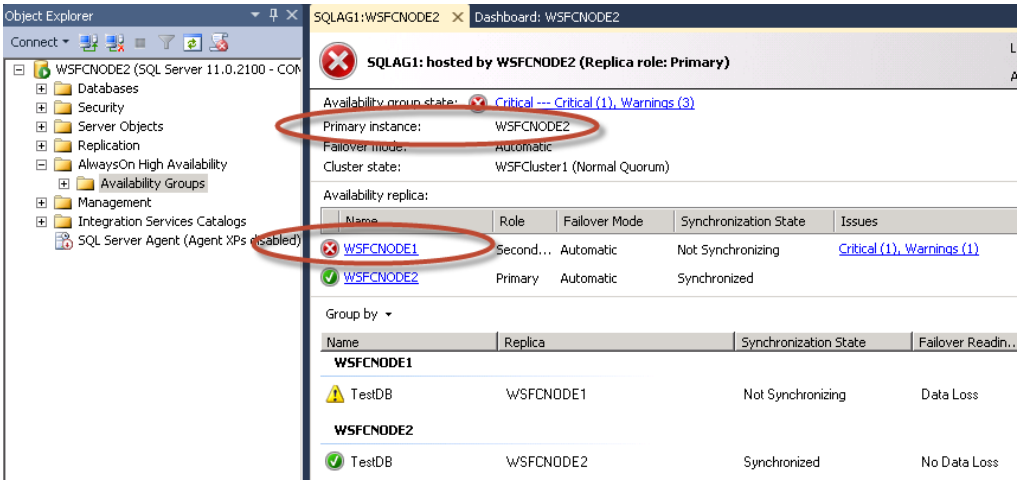
3. Open **SQL Server Management Studio**; in **Object Explorer**, navigate to the **AlwaysOn High Availability** node, and right-click to bring up the Dashboard. Launch the Dashboard for the availability group you created earlier (e.g., SQLAG1)
4. In the Dashboard, view the **Availability Replicas**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



5. Make sure that the primary instance and the IP address in the **Cluster Core Resource** window of **Server Manager** are coordinated. That means, if the primary instance is WSFCNode1, then IP address 10.0.2.101 should be Online.
6. Open the **AWS Management Console** and bring up the 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, open **Server Manager**, navigate to **Failover Cluster Manager** to view the **Cluster Core Resources**. Note that now the IP address previously Offline (e.g., IP address: 10.0.6.101) is now Online.



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**.**



12. At this point, you can start the WSFCNode1 instance again in the 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.

Conclusion

WSFC provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the AWS cloud and "SQL Server AlwaysOn takes advantage of WSFC and provides an integrated, flexible solution that increases application availability…"[5]

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.
- 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 case of a failure of all the instances participating in the cluster (e.g., WSFCNode1 and the Witness file share we placed on the primary Domain

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

Controller) and that are located in the primary Availability Zone (AZ1). In such a scenario, the cluster would fail as it loses quorum and applications lose connectivity. Manual disaster recovery steps that include restarting the cluster service and forcing quorum on a single node (e.g., Net.exe /forcequorum) are necessary to restore application availability.

We recommend you consult the Microsoft 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.amazonwebservices.com/AWSEC2/latest/WindowsGuide/Welcome.html?r=7870
- Microsoft AMIs for Windows and SQL Server:
  - http://aws.amazon.com/windows
  - http://aws.amazon.com/amis/Microsoft?browse=1
  - http://aws.amazon.com/amis/6258880392999312 (SQL Server)
- AWS Windows and .NET Developer Center:
  - http://aws.amazon.com/net
- Microsoft License Mobility:
  - http://aws.amazon.com/windows/mslicensemobility
- Whitepapers/Articles:
  - "Deploy a Microsoft SharePoint 2010 Server Farm in the AWS Cloud in 6 Simple Steps" – http://aws.amazon.com/articles/9982940049271604
  - "Microsoft SharePoint 2010 on AWS: Advanced Implementation Guide" – http://media.amazonwebservices.com/AWS_SharePoint_Reference_Implementation_Guide.pdf
  - "Microsoft SharePoint Server on AWS: Reference Architecture" – http://awsmedia.s3.amazonaws.com/SharePoint_on_AWS_Reference_Architecture_White_Paper.pdf
  - "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, the necessary security groups are configured for you by the provided AWS CloudFormation Templates and are listed here for your reference:

Subsystem Port Mappings

Subsystem	Associated With	Inbound Interface	Port(s)
FirstDomainControllerSG	DC1	DomainMemberSG	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		PrivateSubnet5CIDR (subnet where the Replica DC is deployed into)	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ1)	TCP3389, (ICMP -1)
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)
SecondDomainControllerSG	DC2	DomainMemberSG	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		FirstDomainControllerSG (Security Group where the Primary DC is deployed into)	UDP123, TCP135, UDP138, TCP445, UDP445, TCP464, UDP464, TCP49152-65535, UDP49152-65535, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP54, UDP53, TCP88, UDP67, UDP2535
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)

		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389, (ICMP -1)
DomainMemberSG	WSFCNode1, WSFCNode2, RDGW1, RDGW2	PrivateSubnet1CIDR (subnet where Primary DC is deployed into)	TCP53, UDP53, TCP49152-65535, UDP49152-65535
		PrivateSubnet5CIDR (subnet where Primary DC is deployed into)	TCP53, UDP53, TCP49152-65535, UDP49152-65535
		PublicSubnet1CIDR (subnet where the Remote Desktop Gateway is deployed in AZ1)	TCP3389
		PublicSubnet2CIDR (subnet where the Remote Desktop Gateway is deployed in AZ2)	TCP3389
NAT1SecurityGroup	NAT1	0.0.0.0/0	TCP22
		PrivateSubnet1CIDR (subnet where the Primary DC is deployed into)	ALL1-65535
		PrivateSubnet6CIDR (subnet where the WSFCNode1 is deployed into)	ALL1-65535
NAT2SecurityGroup	NAT2	0.0.0.0/0	TCP22
		PrivateSubnet5CIDR (subnet where the Replica DC is deployed into)	ALL1-65535
		PrivateSubnet6CIDR (subnet where the WSFCNode2 is deployed into)	ALL1-65535
RDGWSecurityGroup	RDGW1, RDGW2	0.0.0.0/0 *	TCP3389
WSFCSecurityGroup	WSFCNode1, WSFCNode2	PrivateSubnet2CIDR (subnet where the WSFCNode1 is deployed into)	ICMP-1, TCP135, TCP137, UDP137, TCP445, TCP1433, TCP3343, UDP3343, TCP5022, TCP49152-65535, UDP49152-65535
		PrivateSubnet6CIDR	ICMP-1, TCP135, TCP137, UDP137, TCP445, TCP1433, TCP3343, UDP3343, TCP5022, TCP49152-65535, UDP49152-65535

NOTE: It is important that RDP **never be** opened up to the entire Internet—not even for testing purposes or temporarily. Always restrict ports and source traffic to the minimum necessary to support the functionality of the application. For a further discussion of securing Remote Desktop Gateway, see the "Securing the Microsoft Platform on Amazon Web Services" whitepaper.

Windows Firewall Configuration

In addition to the EC2 Security Group, which acts as a firewall that controls the traffic allowed to reach one or more instances, you will also have to configure the Windows Firewall as some ports necessary for the cluster nodes to communicate are blocked at that level by default.

On both WSFC cluster nodes (e.g., WSFCNode1, WSFCNode2), the following ports need to be open:

TCP1433, TCP1434, TCP4022, TCP5022, TCP135

Additional Resources

Templates for a Windows Server 2012 based deployment

1. Sample Template-1 (Template_1_Infrastructure_with_AD_2012.template). This will deploy the following AWS resources:
    a. One Amazon VPC spanning two Availability Zones (AZ)
    b. One public route
    c. One Internet Gateway
    d. Per Availability Zone:
        i. 4 private subnets and 1 public subnet
        ii. 1 private route
        iii. 1 Windows Server 2012–based Remote Desktop Gateway (RDGW) instance and 1 Linux-based
        iv. NAT instance to enable administrative ingress and egress
        v. 2 Elastic IP Addresses associated with the NAT and RDGW instances
        vi. 1 Windows Server 2012–based instance to host the Active Directory
2. Sample Template-2 (Template_2_DataTier_PIOPS_2012.template). This will deploy the following AWS resources:
    a. Per Availability Zone:
        i. 1 Windows Server 20012-based instance to host the WSFC nodes and SQL Server 2012
        ii. AlwaysOn Availability Group
        iii. 6 200GiB EBS Volumes with configurable provisioned IOPS (default set to 500) each to create 2 stripe sets (Raid0) arrays configured as follows:
            1. 4 disks (800 GiB) to host the SQL Server database files
            2. 2 disks (400 GiB) to host the SQL Server log files
3. Sample Template-3 (Template_3_APPServer_Demo_2012.template). This will deploy the following AWS resource:
    a. 1 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.