



Microsoft SharePoint Server 2013 on the AWS Cloud: Quick Start Reference Deployment

Mike Pfeiffer

August 2014

Last updated: April 2015 ([revisions](#))

Table of Contents

Abstract	3
What We'll Cover	4
Architecture Scenarios for SharePoint Server 2013 on AWS.....	5
Amazon Virtual Private Cloud (Amazon VPC)	5
Remote Administration.....	6
Active Directory Domain Services.....	7
Server Role Architecture	9
Traditional Topologies	9
Web Tier.....	9
Application Tier	10
Database Tier	11
Streamlined Topologies	13
Front-End Servers.....	13
Batch-Processing Servers	13
Database Servers.....	13
Distributed Cache.....	13
Request Management.....	13
Specialized Workloads	13
Search.....	14
Simple Example of a Streamlined Topology.....	14
Office Web Apps	15
Intranet SharePoint Server Farm on AWS.....	16
Security	17
Security Groups.....	17

Network ACLs.....	18
Secure Website Publishing.....	18
EC2 Instance Types.....	20
Customize Your Topology at Template Launch	20
SharePoint Server 2013 Quick Start Deployment Steps	22
1. Launch Microsoft Windows Server Failover Cluster and SQL Server AlwaysOn Quick Start	23
2. Prepare a Media Volume Snapshot	23
3. Launch the SharePoint Server 2013 Stack	26
Template Customization	26
4. Test High Availability and Automatic Failover	27
Post-Configuration Tasks	38
Further Reading	38
Send Us Your Feedback.....	39
Document Revisions.....	39

Abstract

This Quick Start Reference Deployment includes architectural considerations and configurations used to build a Microsoft SharePoint Server 2013 environment on the Amazon Web Services (AWS) cloud. We discuss how to build and configure the necessary AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC) to deploy a highly available SharePoint farm across separate AWS Availability Zones.

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 a Microsoft Active Directory Domain Services infrastructure, Microsoft SQL Server 2012 or 2014 instances configured in a Windows Server Failover Cluster, and multiple Amazon EC2 instances to participate in the SharePoint Server 2013 farm in multiple Availability Zones in an Amazon VPC.

To begin the deployment process using AWS CloudFormation templates, go to [SharePoint Server 2013 Quick Start Deployment Steps](#).

Note

SharePoint Server 2013 can be deployed and licensed via the [Microsoft License Mobility through Software Assurance](#) program.



This guide targets IT infrastructure administrators and architects. After reading it, you should have a good understanding of how to launch the necessary infrastructure to support SharePoint Server 2013 on the AWS cloud.

What We'll Cover

This Quick Start Reference Deployment covers the deployment of Microsoft SharePoint Server 2013 in a highly available architecture. We provide an automated solution to deploy the Quick Start in the form of a dynamic AWS CloudFormation template that allows you to choose a highly available architecture that meets your specific requirements. The foundation for this environment utilizes Active Directory Domain Services and Microsoft SQL Server AlwaysOn Availability Groups deployed across two Availability Zones. This Quick Start discusses how the SharePoint Server 2013 environment is built so that you can deploy the automated solution or customize the provided AWS CloudFormation template as needed.

After deploying this Quick Start with the default input parameters, you will have built the following SharePoint Server 2013 environment in the AWS cloud:

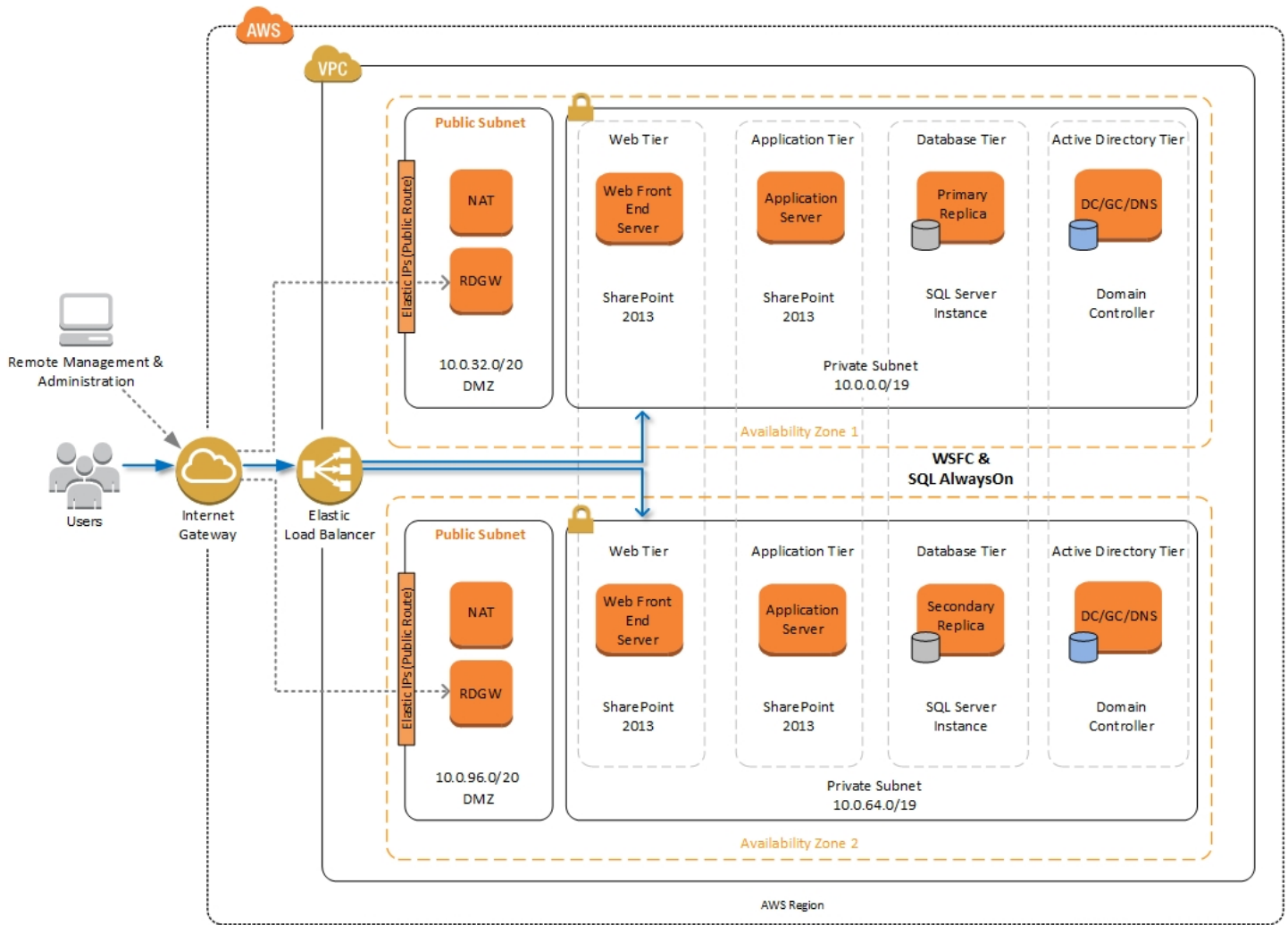


Figure 1: Highly Available SharePoint Farm on AWS

Architecture Scenarios for SharePoint Server 2013 on AWS

Amazon Virtual Private Cloud (Amazon VPC)

Amazon VPC lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. With Amazon VPC, you can define a virtual network topology closely resembling a traditional network that you might operate on your own premises. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

When deploying a Windows-based architecture on the AWS cloud, we recommend an Amazon VPC configuration that supports the following requirements:

- Critical workloads should be placed in a minimum of two Availability Zones to provide high availability.
- Internal application servers and other non-Internet facing servers should be placed in private subnets to prevent direct access to these instances from the Internet.
- RD Gateways should be deployed into public subnets in each Availability Zone for remote administration. Other components, such as reverse proxy servers, can also be placed into these public subnets if needed.

For details on the Amazon VPC design used in this reference, see the [Active Directory Domain Services Quick Start Reference Deployment guide](#).

Based on these best practices, we deploy the following base-level Amazon VPC design to support our Microsoft SharePoint 2013 infrastructure:

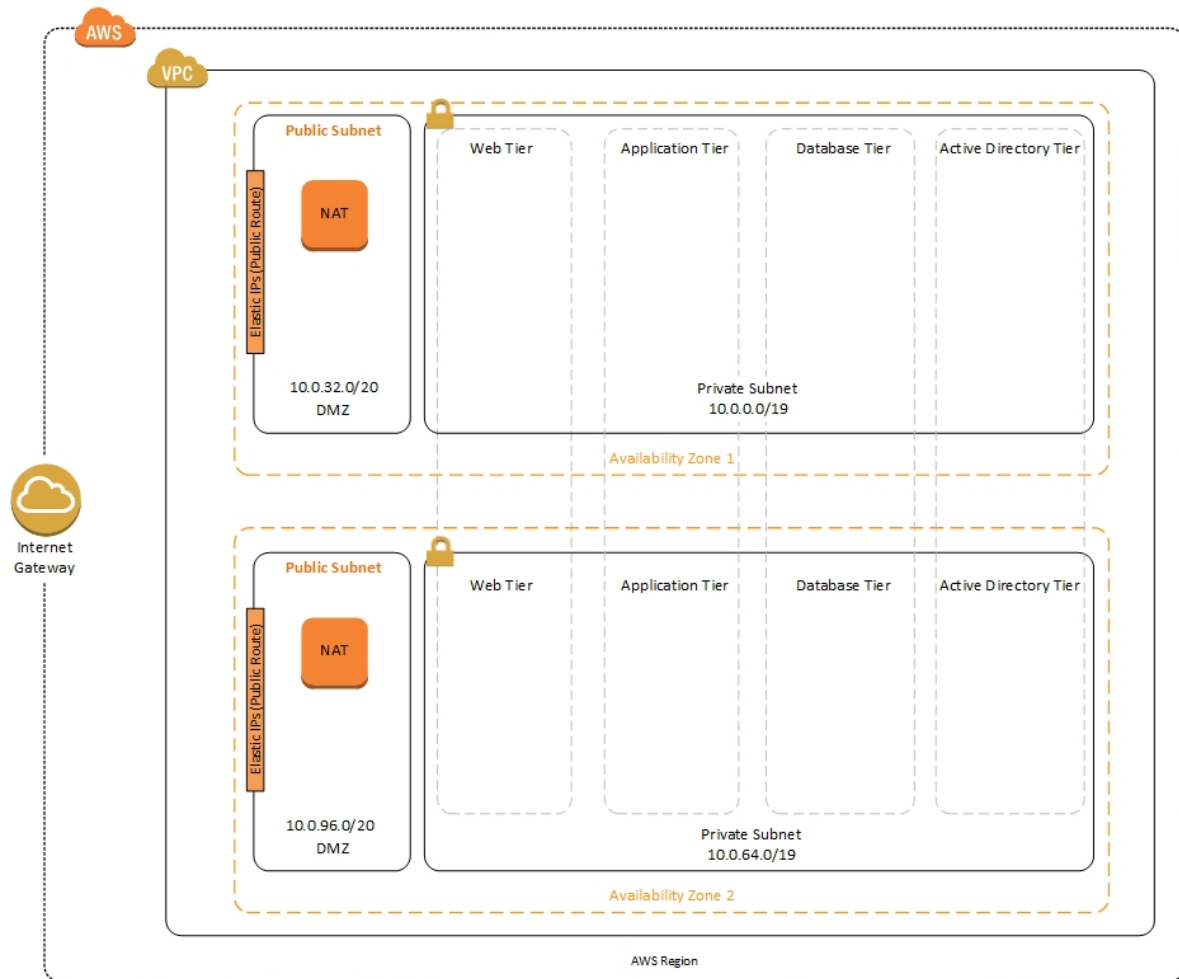


Figure 2: Tiered Amazon VPC Architecture on the AWS Cloud

As shown in Figure 2, Linux-based NAT instances are deployed into the public subnets. The public subnets have a route to the Internet directly through the Internet Gateway attached to the Amazon VPC.

Instances that will be deployed in the private subnets have no direct route to the Internet. Instead, instances in private subnets use private routes to send Internet traffic to the NAT instances in the public subnets. This architecture isolates your critical workloads from direct Internet access.

Remote Administration

As we design the architecture for a highly available SharePoint farm, we should also design for highly available and secure remote access. We can do this by deploying a Remote Desktop (RD) Gateway in each Availability Zone. In case of an Availability Zone outage, this architecture allows access to the resources that may have failed over to the other Availability Zone.

The RD Gateway uses the Remote Desktop Protocol (RDP) over HTTPS to establish a secure, encrypted connection between remote administrators on the Internet and Windows-based Amazon EC2 instances, without needing to configure a virtual private network (VPN) connection. This allows you to reduce the attack surface on your Windows-based instances while providing a remote administration solution for administrators.

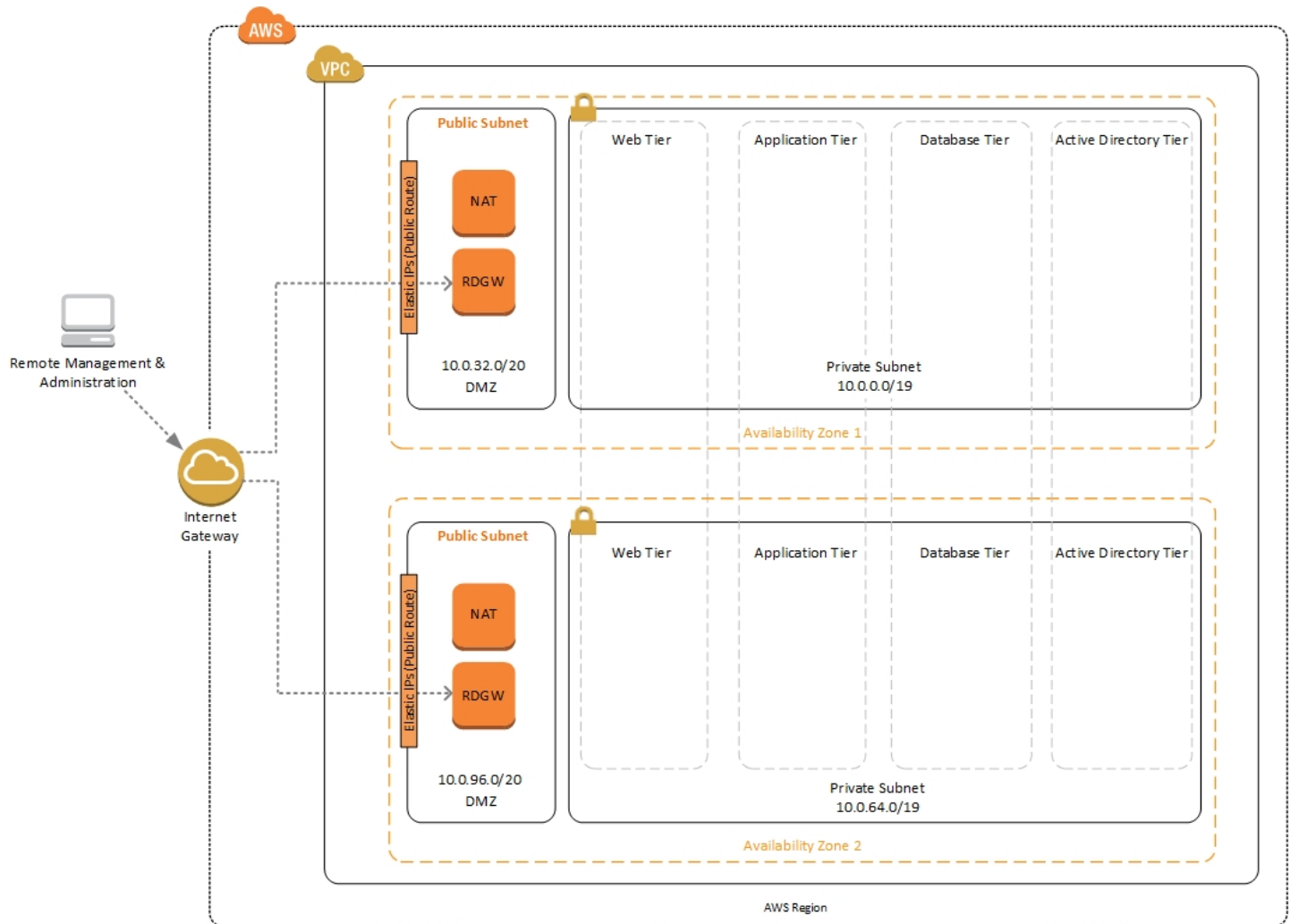


Figure 3: NAT Instances and Remote Desktop Gateways in Public Subnets

The architecture and configuration steps outlined in the Quick Start Reference Deployment for [Microsoft Remote Desktop Gateway on AWS](#) are automatically deployed by the AWS CloudFormation templates provided in this Quick Start.

After you've launched your SharePoint infrastructure using the deployment scenario in this guide, you will initially connect to your instances using a standard RDP TCP Port 3389 connection. You can then follow the steps in the Quick Start Reference Deployment for [Remote Desktop Gateway on AWS](#) to secure future connections via HTTPS.

Active Directory Domain Services

In order to provide user authentication and authorization, the Microsoft SharePoint servers in this reference architecture use Active Directory Domain Services. As you deploy your environment, you should place at least one Domain Controller in a private subnet in each Availability Zone for redundancy and high availability.

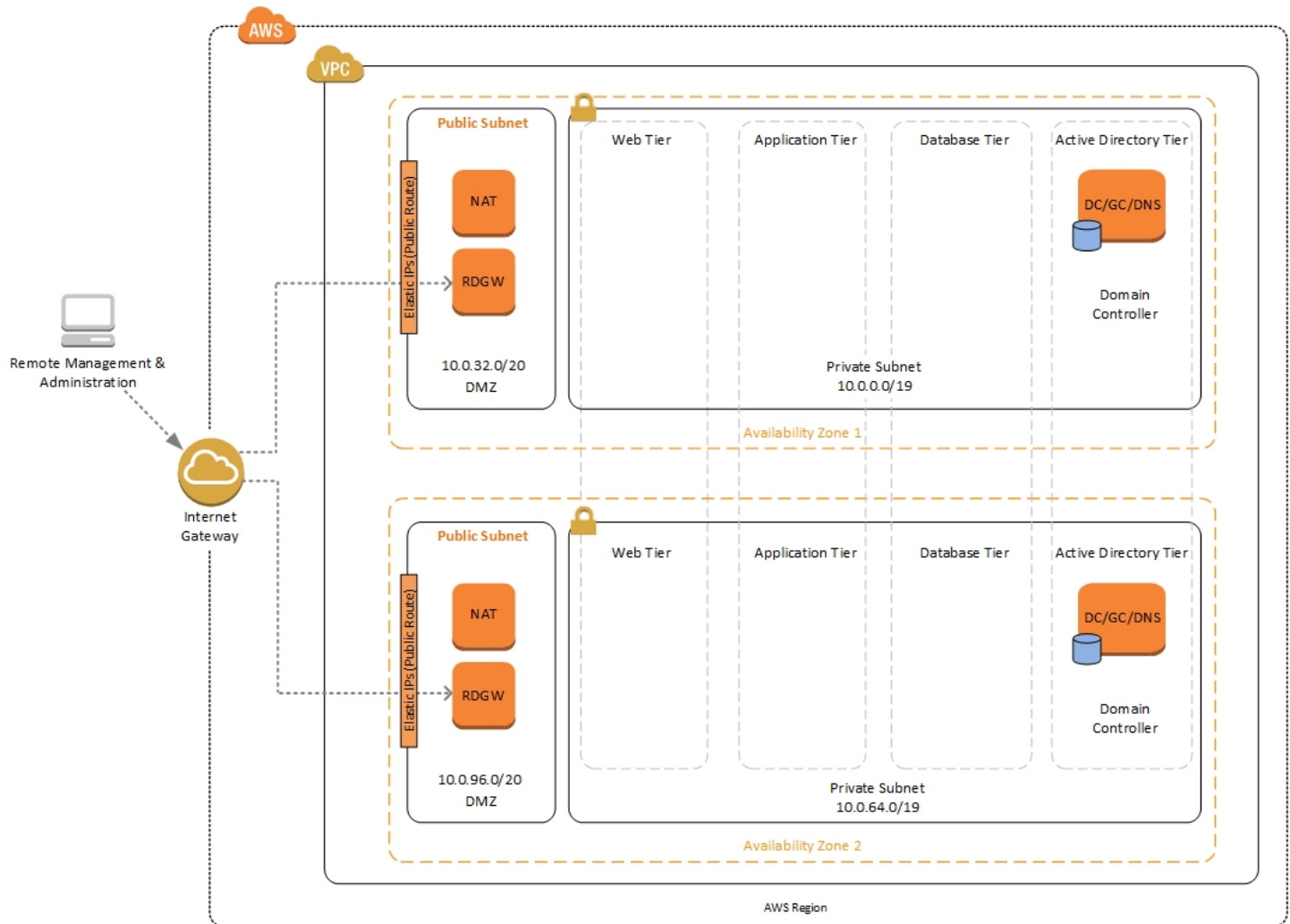


Figure 4: Domain Controllers in each Availability Zone

Notice that in Figure 4, we've now included a Domain Controller in the Active Directory tier in each Availability Zone.

There are two ways to use Active Directory Domain Services (AD DS) in the AWS cloud:

- **Cloud Only** – This is the architecture shown in Figure 4. This type of architecture means that your entire Active Directory forest exists only within the AWS cloud. With a cloud-only AD DS architecture, there are no on-premises Domain Controllers.
- **Hybrid** – The hybrid architecture takes advantage of your existing AD DS environment. You can extend your private, on-premises network to AWS so the resources in the cloud can utilize your existing AD infrastructure. We recommend that in a hybrid architecture you also deploy Domain Controllers for your existing AD forest to the AWS cloud. We recommend this configuration primarily to help ensure that the application servers deployed in AWS remain functional and available in the event of an on-premises outage.

The Quick Start Reference Deployment for [Active Directory Domain Services on AWS](#) covers all of our best practices and recommendations for deploying AD on AWS. The process outlined in this SharePoint Quick Start first launches the AD DS Quick Start to provide the foundation for the remaining infrastructure. It's responsible for building the Amazon VPC,

public and private subnets, NAT instances and Remote Desktop Gateway (RD Gateway) instances, and Domain Controllers in each Availability Zone.

Server Role Architecture

There are a number of ways to design the topology of your SharePoint farm depending on your requirements. Microsoft provides guidance for two separate architectural approaches for SharePoint 2013: **Traditional topologies** and **Streamlined topologies**. The CloudFormation template provided by this guide is built with flexibility in mind, allowing you to deploy a SharePoint farm that can be configured to operate using either architecture covered in the following two sections.

Traditional Topologies

When you build your SharePoint Server 2013 farm based on traditional topologies, you use approaches to building your architecture with web servers, application servers, and database servers as you would have with SharePoint 2010.

In a traditional farm topology, a common architecture in small environments is the two-tier design. This design utilizes two servers: one for web front-end and application services and the other for database services.

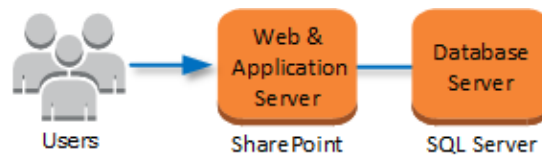


Figure 5: Two-Tier SharePoint Farm in a Traditional Topology

A traditional three-tier SharePoint architecture consists of a web-tier, an application server tier, and a database tier.

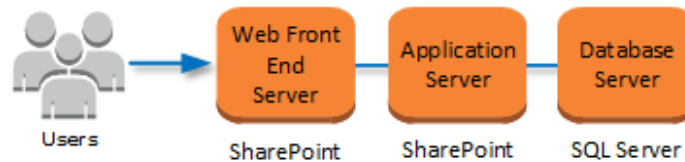


Figure 6: Three-Tier SharePoint Farm in a Traditional Topology

Detailed descriptions of each tier in a SharePoint 2013 farm built with a traditional topology are provided in the following sections.

Web Tier

The web server role responds to end-user requests for web pages. In order to provide high availability, two separate Availability Zones each host a web server instance for the SharePoint farm. Traffic to these web front end instances can be load balanced using Elastic Load Balancing or another third party load balancing solution such as HA Proxy.

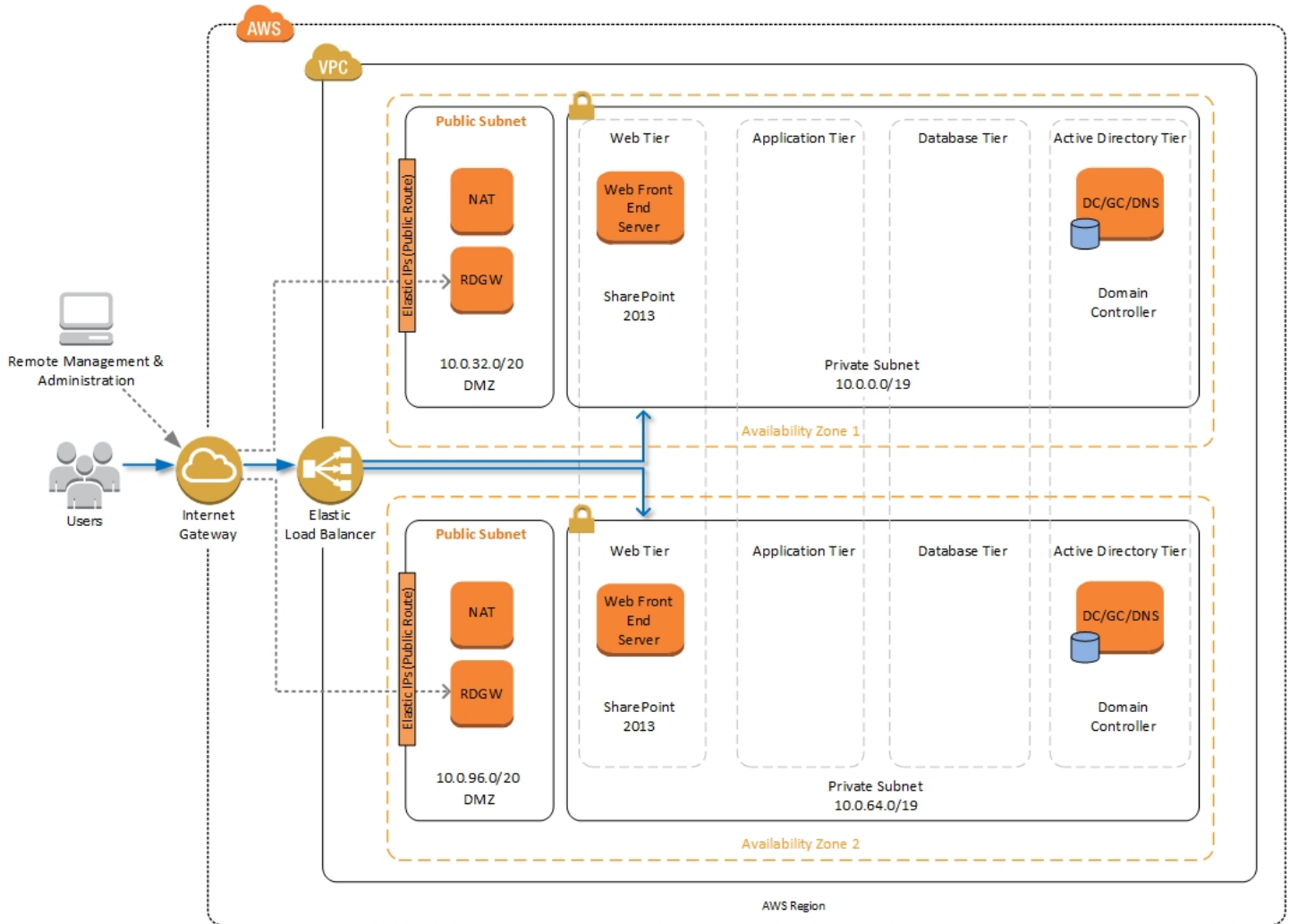


Figure 7: SharePoint Web Front End Servers in Each Availability Zone

Application Tier

The application server role runs services that enable users to access various services and features such as Microsoft Excel, Microsoft Visio or Microsoft Access. As in the web server role, you can place application servers in each Availability Zone to provide high availability for SharePoint services.

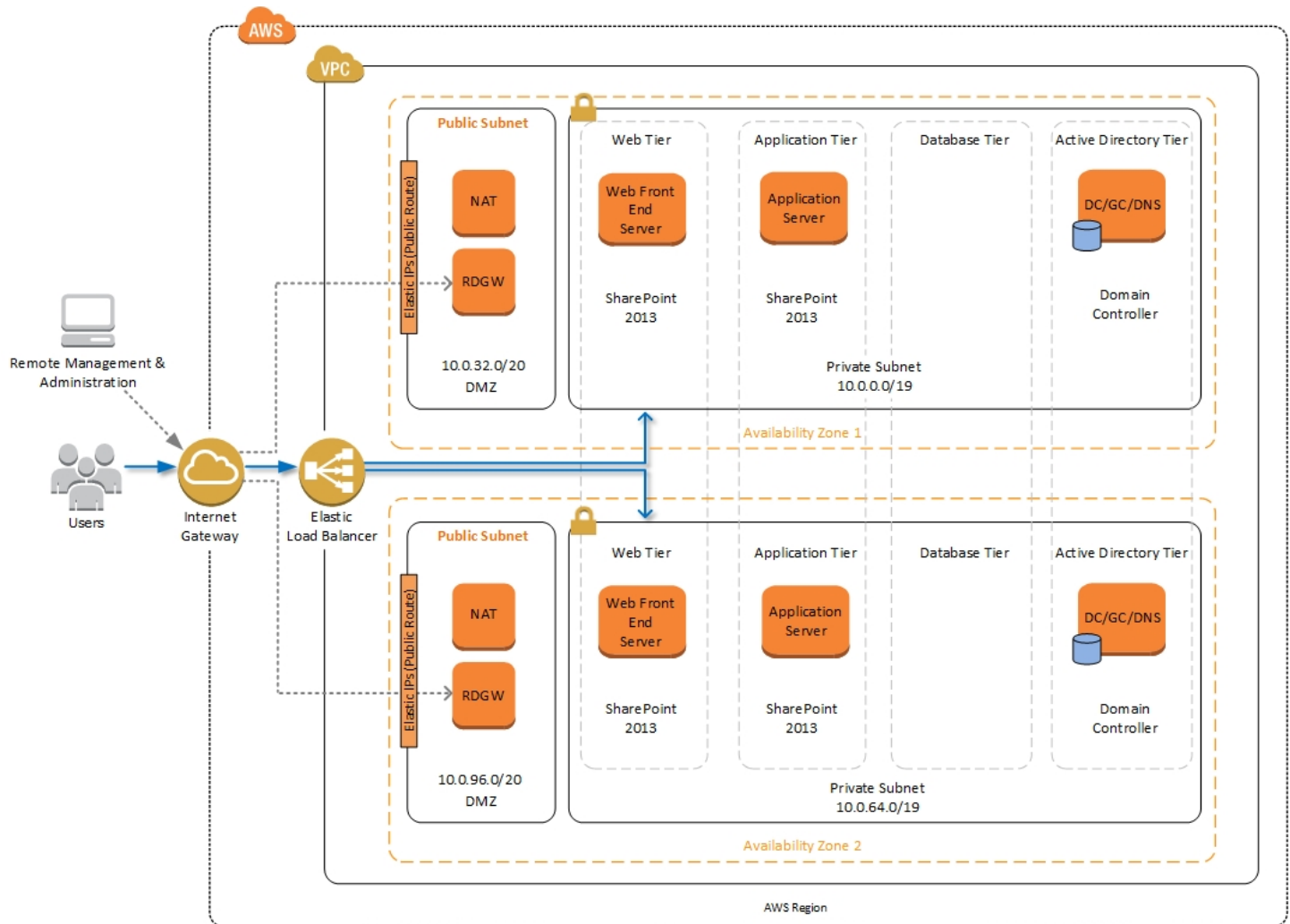


Figure 8: SharePoint Web Front Ends and Application Servers in Each Availability Zone

Unlike web servers, the application servers do not need to be load balanced with an external service like Elastic Load Balancing. You can create redundancy for application services by hosting those services on application servers in each Availability Zone. End-users are sent to web front ends, and those servers reach back to application servers as needed.

Database Tier

The database server role stores content and service data so your SharePoint farm can utilize SQL Server in a number of ways. For small or medium sized environments, you may be able to place all of your databases on a single server. For larger sized farms, you can spread your databases across multiple SQL Servers or clusters of SQL Servers. We recommend using SQL Server Enterprise in your SharePoint deployment, as it meets the performance, high availability, and reliability requirements for an enterprise application.

Amazon Machine Images (AMIs) for SQL Server Express, SQL Server Web Edition, and SQL Server Standard are available for launch on AWS. To install SQL Server 2012 or 2014 Enterprise Edition on AWS, you can utilize [Microsoft License Mobility through Software Assurance](#) to bring your own license into the cloud.

In the Quick Start Reference Deployment for [Microsoft WSFC and SQL Server AlwaysOn on AWS](#), we provide an example of how you can deploy an AlwaysOn Availability Group to provide high availability for your databases. Our default SQL Server configuration uses the r3.2xlarge instance type, which is a memory optimized instance with eight vCPUs, 60GiB of memory, and 1 x 160 GB of SSD instance storage. Additionally, we provide highly performant and durable storage in the form of Elastic Block Store (Amazon EBS) volumes.

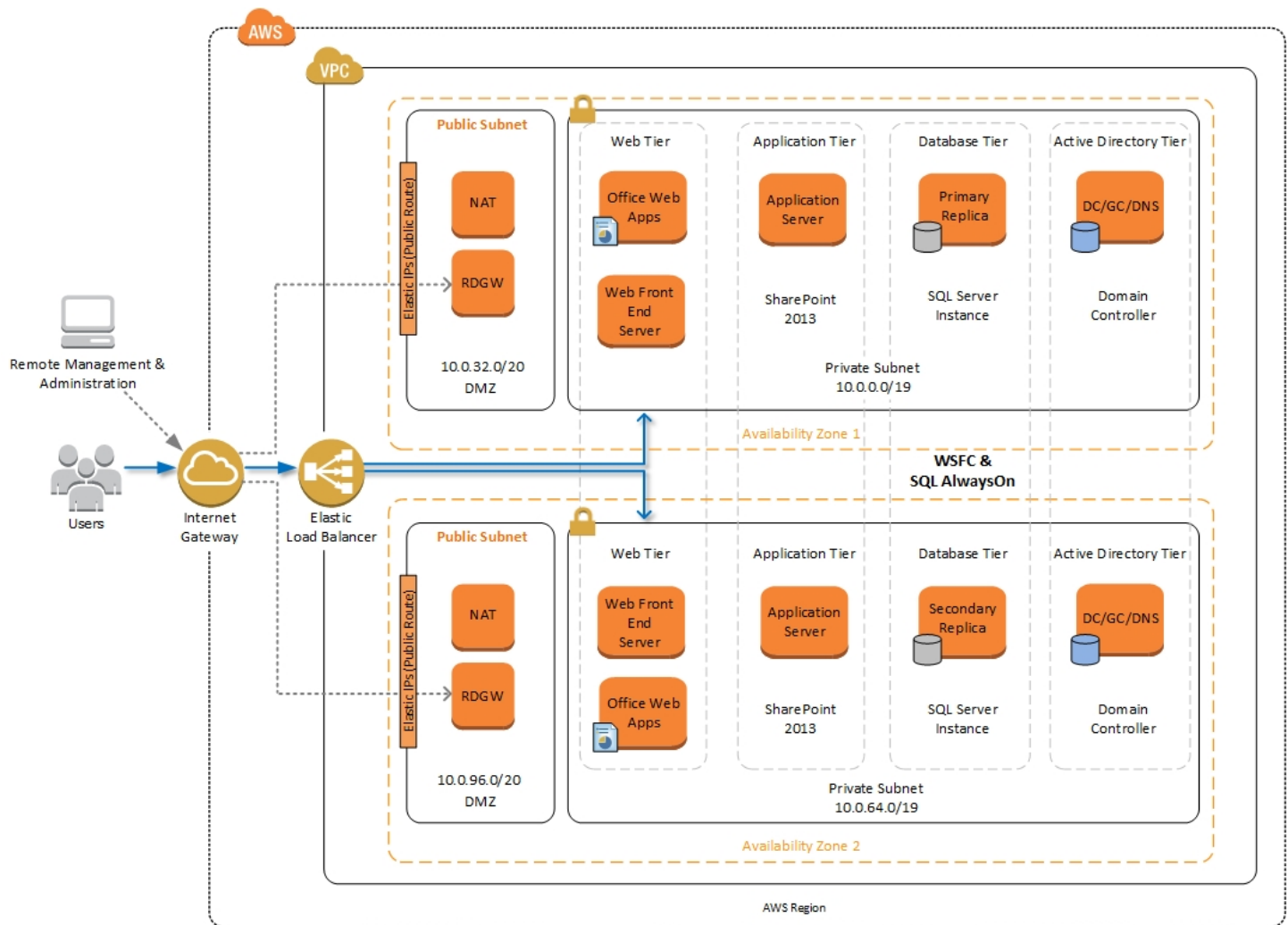


Figure 9: Highly Available SharePoint Farm in AWS

The Microsoft WSFC and SQL Server AlwaysOn Quick Start is used automatically as the database tier for your SharePoint Server farm when you launch the automated solution provided in this guide. There are a number of input parameters that allow you to control the instance type and other settings, and you can further customize the deployment to meet your specific needs. For more details, take a look at the Quick Start Reference Deployment for [Microsoft WSFC and SQL Server AlwaysOn on AWS](#) for SQL Server Enterprise on AWS.

For more details on the traditional topologies and configuring services on SharePoint 2013, see the [technical diagrams for SharePoint 2013](#) and [Services on Server Install Worksheet for Traditional Topologies](#) provided by Microsoft.

Streamlined Topologies

When building your SharePoint farm based on streamlined topologies, services and other components are distributed to maximize server resources. Streamlined architectures include front-end servers, batch-processing servers, and database servers. Streamlined topologies introduce a new approach to farm design in SharePoint 2013. Using this type of topology allows you to scale out easier, as the servers in the front-end and batch-processing tiers are standardized. When the time comes to scale out within a specific tier, you simply add an identically configured server in your environment. Detailed descriptions of each tier in a SharePoint 2013 farm built with a streamlined topology are provided in the following sections.

Front-End Servers

Components, services and services applications that serve end-user requests directly are placed on front-end servers. Front-end servers are optimized for fast performance.

Batch-Processing Servers

Batch-processing or “Back-End” servers provide a middle-tier of servers running components, services, and service applications that process background tasks. Batch-processing servers can tolerate more resource intensive tasks since end-users do not interface with these servers directly.

Database Servers

Database servers in a streamlined topology are no different than database servers in a traditional topology. The database tier will still consist of SQL Servers and traditional guidance for deploying database servers remains the same.

Distributed Cache

Distributed cache can run on front-end servers in small or medium environments with less than 10,000 users. For larger environments, Distributed Cache, which is a memory intensive service, is typically placed on dedicated servers.

Request Management

Request management gives SharePoint the ability to route incoming requests based on routing rules. The Request Management component can be run on front-end servers, installed together in tandem on Distributed Cache servers, or on dedicated servers.

Specialized Workloads

Some organizations will use other service applications such as Excel Calculation or Performance Point very heavily. In this scenario, these services are placed on dedicated servers.

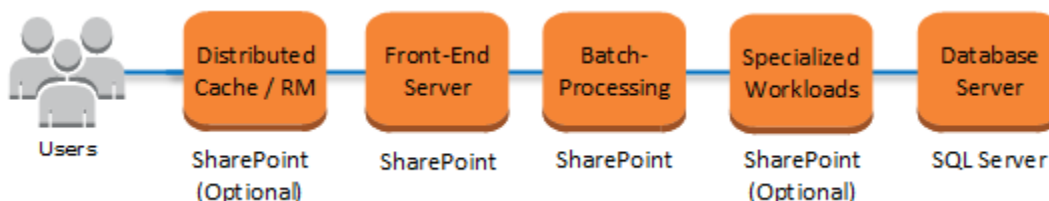


Figure 10: Visualizing the Streamlined Topology

Search

As larger environments scale beyond two batch-processing servers, it is very common to place the Search role on dedicated server, as the Search workload consumes a lot of system resources. You can optionally configure the Search role to utilize databases on a separate SQL server(s) for maximum performance.

There are a number of ways to architect a large SharePoint farm using a Streamlined topology. For additional details on the Streamlined topologies and configuring services on SharePoint 2013, see the [Technical diagrams for SharePoint 2013](#) and [Services on Server Install Worksheet for Streamlined Topologies](#) provided by Microsoft.

Simple Example of a Streamlined Topology

Figure 11 shows a SharePoint 2013 architecture based on a Streamlined topology running in the AWS cloud. This architecture includes the tiers for Front-End servers, Batch-Processing servers, and Database servers. It also includes an additional SharePoint server in each Availability Zone dedicated to the Distributed Cache feature.

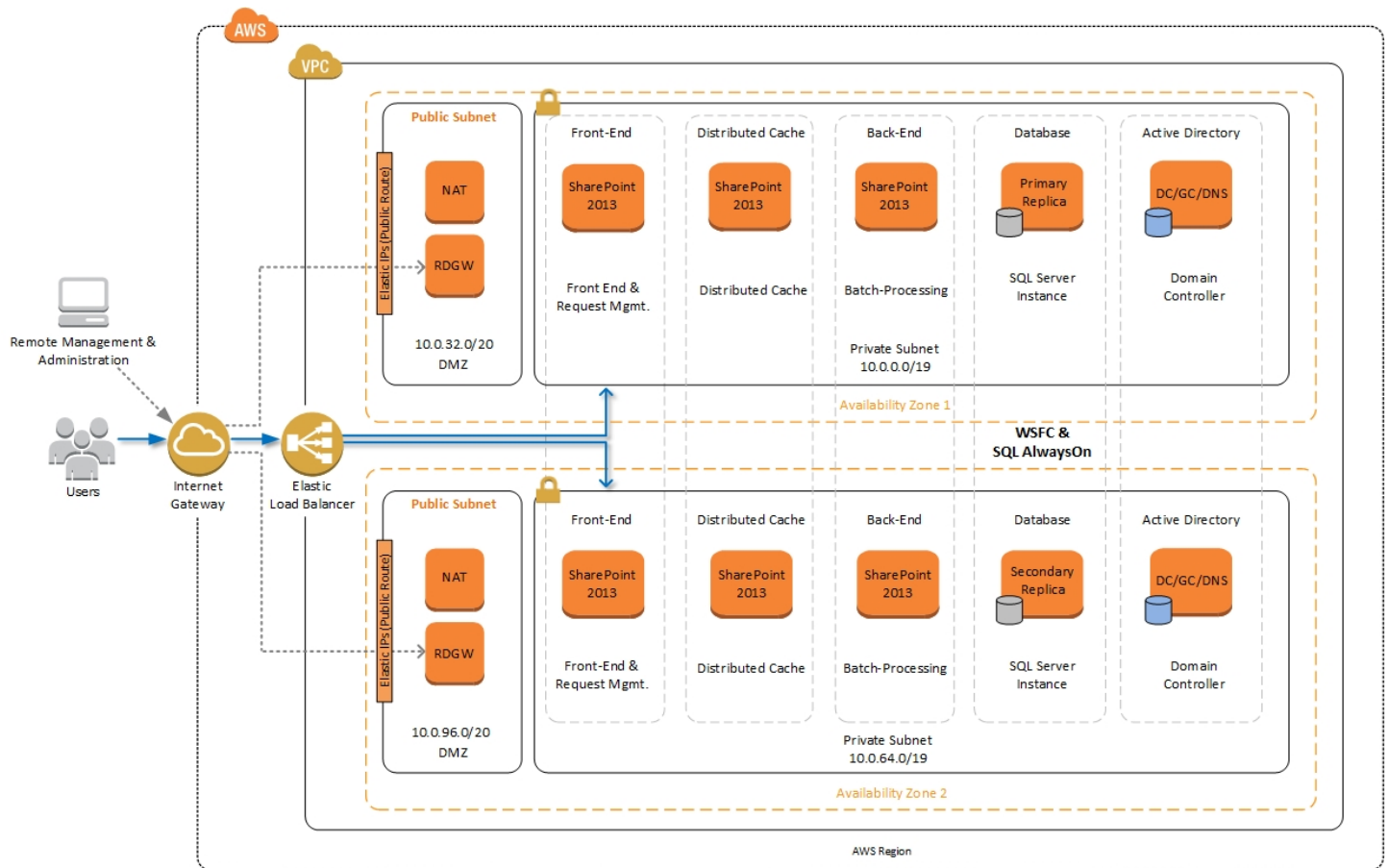


Figure 11: Example Streamlined Topology in the AWS Cloud

Whether you decide to use a Traditional or Streamlined SharePoint 2013 topology, the AWS CloudFormation template launched from this guide will automatically utilize the Microsoft WSFC and SQL Server AlwaysOn Quick Start for the database tier.

Office Web Apps

The Microsoft Office Web Apps server allows users to view and, depending on the scenario, edit Office documents in SharePoint libraries using a supported browser on various devices such as PCs, mobile devices, and tablets. Figure 12, shows an Office Web Apps server within the web server tier in each Availability Zone.

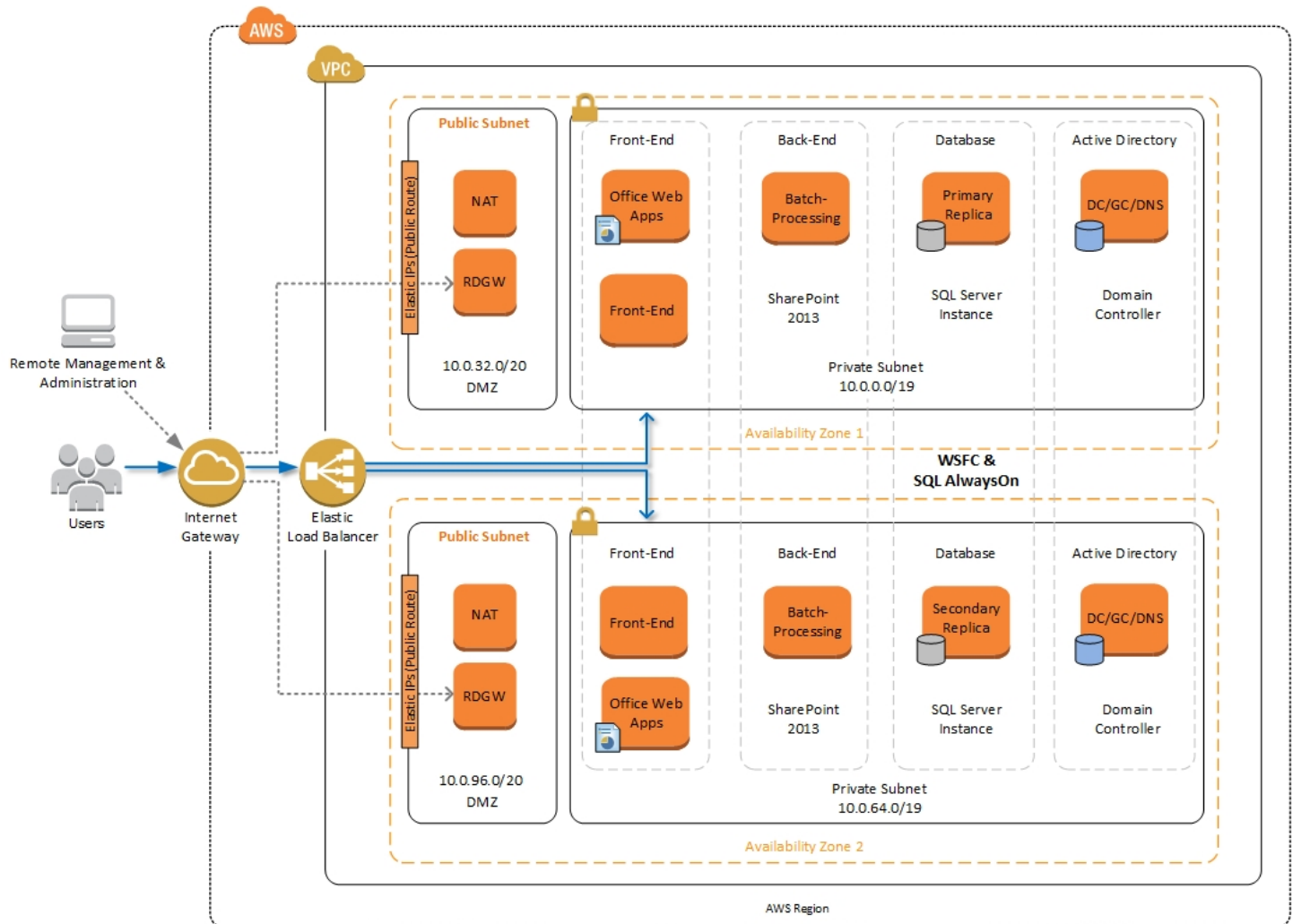


Figure 12: Highly Available SharePoint Farm with Office Web Apps Servers in AWS

It is important to notice that the Office Web Apps server role is not installed on the SharePoint 2013 servers and must be deployed on separate servers in the environment. The Office Web Apps server can also be used by other enterprise products like Microsoft Exchange 2013 and Lync 2013 for rendering Office documents through a browser.

The CloudFormation template provided by this Quick Start allows you to choose whether or not to include Office Web App servers in your environment. If you choose to include these servers, you'll need to download and install the Office Web Apps component manually.

DomainNetBIOSName	<input type="text" value="example"/>	NetBIOS name of the domain (upto 15 characters) for users of earlier versions of Windows e.g. CORP
IncludeELB	<input type="text" value="external"/>	Specify whether or not you want to include an Elastic Load Balancer. Allowed values are internal, external, or none
IncludeOfficeWebApps	<input type="text" value="true"/>	To include an Office Web Apps Server in each AZ, set this parameter to true.
KeyPairName	<input type="text" value="MyKeyPair"/>	Public/private key pairs allow you to securely connect to your instance after it launches

Figure 13: Specifying Office Web Apps Options in Template Parameters

If you do choose to include Office Web Apps servers in your deployment, you will need to perform some post configuration tasks to include them in your SharePoint 2013 farm. These post configuration tasks are covered later in this document.

Intranet SharePoint Server Farm on AWS

All of the architecture diagrams shown up to this point represent an Internet facing Microsoft SharePoint farm. For the Internet facing farm scenario, external users access SharePoint through external Elastic Load Balancing. For a non-Internet facing SharePoint server farm scenario, you'll still want to include a load balancer for the front-end server tier, but in this case the load balancer will be accessible only from the internal network. Figure 14 shows a typical topology for an intranet SharePoint server farm running on the AWS cloud.

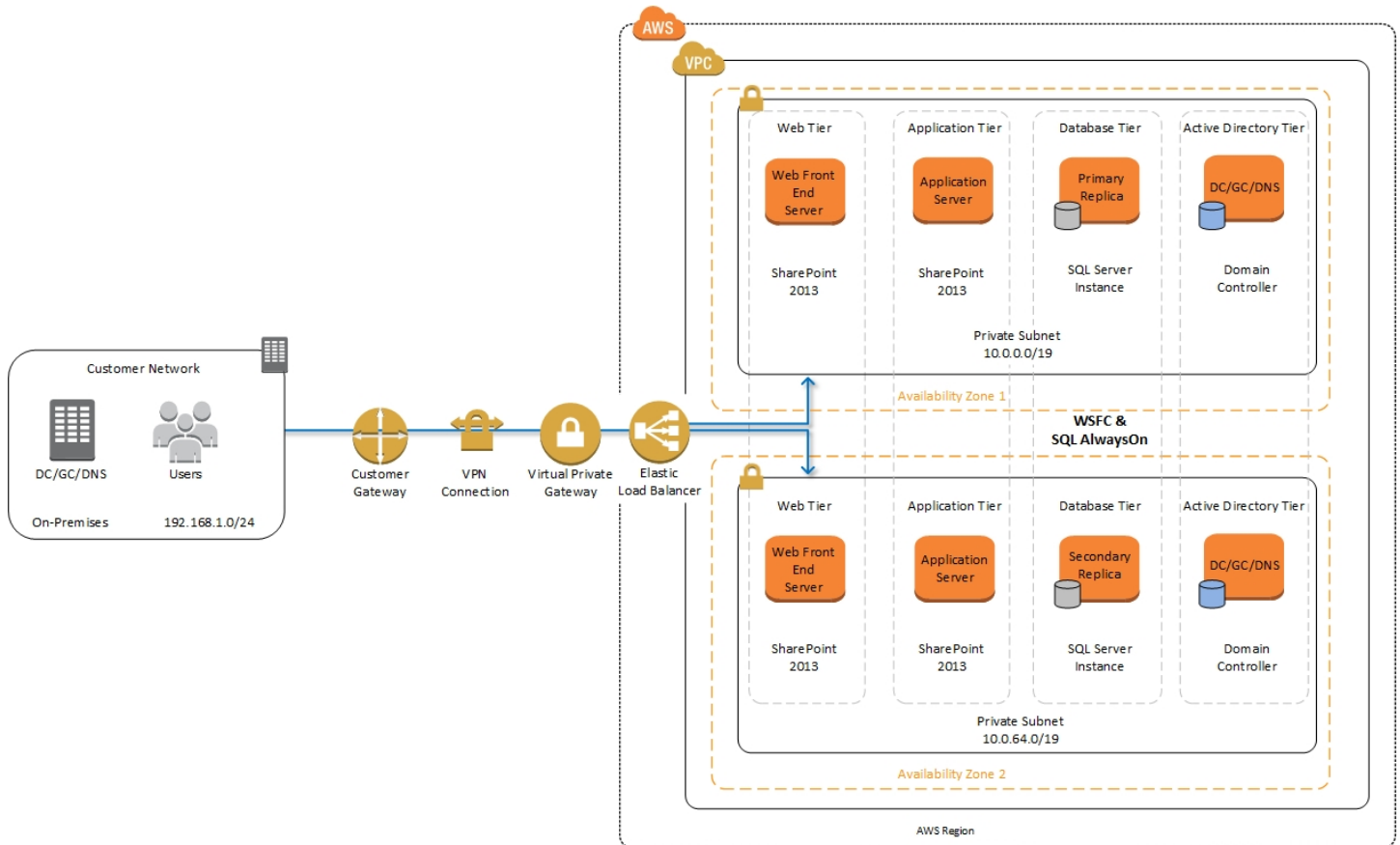


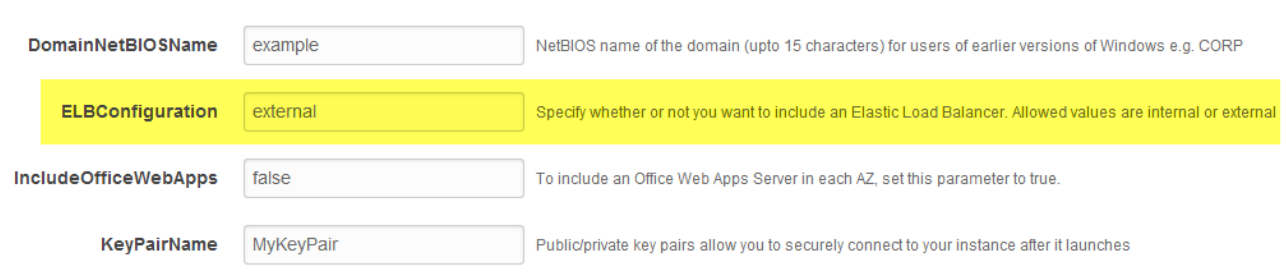
Figure 14: Intranet SharePoint Farm with Hybrid Architecture



As shown in Figure 14, we've added a virtual private gateway to the Amazon VPC. To enable internal network connectivity to the Amazon VPC, we've created a VPN tunnel from the Customer Gateway (an IPSec capable device) to the VPG running in the Amazon VPC.

In addition, AWS offers the Direct Connect service which allows you to create a direct network connection from your data center into the AWS cloud. In either case, once you have internal network connectivity into the Amazon VPC from your on-premises environment, you can simply provision internal Elastic Load Balancing to spread incoming traffic to front-end servers across each Availability Zone. Elastic Load Balancing will also provide high availability in the event of a server failure. If a web front end server is unavailable, requests will be sent to one that is online.

The AWS CloudFormation template provided in this Quick Start will allow you to choose how to implement Elastic Load Balancing. You can choose from two options: internal or external. The default setting is initialized to external.



DomainNetBIOSName	example	NetBIOS name of the domain (upto 15 characters) for users of earlier versions of Windows e.g. CORP
ELBConfiguration	external	Specify whether or not you want to include an Elastic Load Balancer. Allowed values are internal or external
IncludeOfficeWebApps	false	To include an Office Web Apps Server in each AZ, set this parameter to true.
KeyPairName	MyKeyPair	Public/private key pairs allow you to securely connect to your instance after it launches

Figure 15: Specifying Elastic Load Balancing Options in Template Parameters

If you are building an intranet only farm, you can simply deploy your SharePoint environment using the provided AWS CloudFormation template and, upon completion, connect your on-premises environment to AWS using either VPN or AWS Direct Connect.

Note

You must use Forms-Based or Kerberos Authentication for your SharePoint servers when load balancing with Elastic Load Balancing (ELB). NTLM authentication is not supported with ELB at this time. There are a number of third party load balancing solutions in the [AWS Marketplace](#) that can be used as an alternative.

Security

As with any enterprise application deployment, a Microsoft SharePoint Server farm on AWS should implement strict security controls. AWS provides a comprehensive set of security features that allow you to control the flow of traffic through your Amazon VPC and associated subnets and ultimately to each Amazon EC2 instance. These features allow you to reduce the attack surface of your environment while providing both end-user access to SharePoint content and applications and administrator access for securely managing the Windows server infrastructure. These security features and approaches are covered in this section.

Security Groups

When launched, Amazon EC2 instances must be associated with at least one Security Group which acts as a stateful firewall. You have complete control over the network traffic entering or leaving your Security Groups, and you can build granular rules that are scoped by protocol, port number, and source/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.

Network ACLs

A network access control list (ACL) is a set of permissions that can be attached to any network subnet in an Amazon VPC to provide stateless filtering of traffic. Network ACLs can be used for inbound or outbound traffic and provide an effective way to blacklist a CIDR block or individual IP addresses. These ACLs can contain ordered rules to allow or deny traffic based upon IP protocol, service port, or source or destination IP address. Figure 16 shows the default ACL configuration for an Amazon VPC subnet.

Network ACL: Default (replace)				
Inbound:				
Rule #	Port (Service)	Protocol	Source	Allow/Deny
100	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL	ALL	0.0.0.0/0	DENY
Outbound:				
Rule #	Port (Service)	Protocol	Destination	Allow/Deny
100	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL	ALL	0.0.0.0/0	DENY

Figure 16: Default Network ACL Configuration for an Amazon VPC Subnet

You may choose to keep the default network ACL configuration, or you may choose to lock it down with more specific rules to restrict traffic between subnets at the network level. Typically, Network ACLs will mirror your Security Group rules. One benefit of multiple layers of network security (Security Groups and Network ACLs) is that each layer can be managed by a separate group in your organization. If a server administrator inadvertently exposes unnecessary network ports on a Security Group, a network administrator could supersede this configuration by blocking that traffic at the Network ACL layer.

Secure Website Publishing

Some organizations may utilize SharePoint server to host a public website. In this scenario, you can add an additional layer of security by placing reverse proxy servers into your public subnet to provide additional security and threat management. In this configuration, the public subnet acts like the DMZ that you would typically use in a physical network environment. Web page requests from Internet-based users would be sent to these reverse proxy servers, which would then establish a connection to your web front end servers running in a private subnet.

Figure 16 shows an example of publishing SharePoint web front end servers, located in a private subnet, through a reverse proxy server deployed into a public subnet.

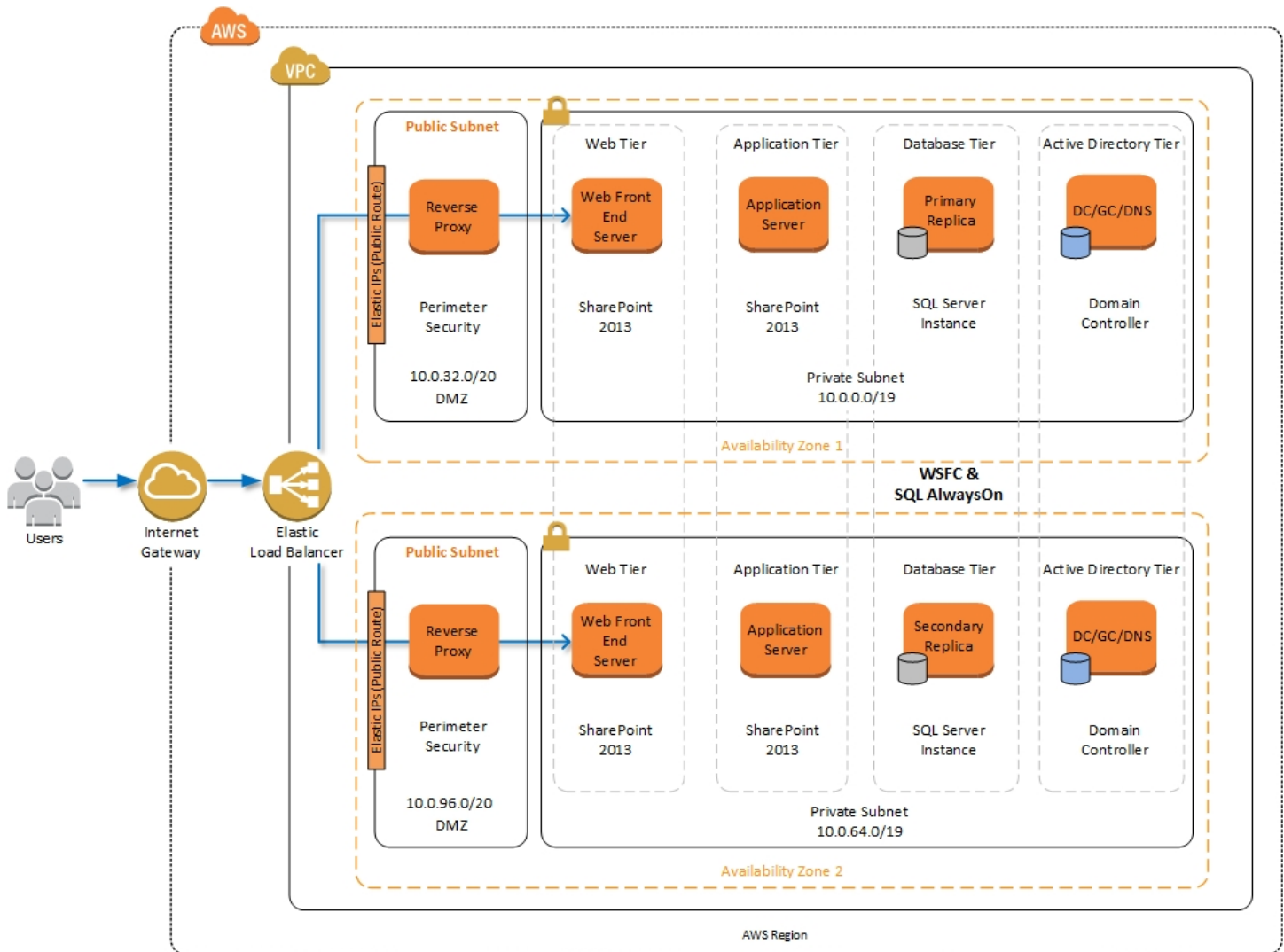


Figure 17: Web Application Publishing with a Reverse Proxy Server

A benefit of this architecture is that it provides the ability to pre-authenticate users at the perimeter of your network while shielding your internal SharePoint servers from the public Internet. Several third-party appliances and applications can be used for this task. Microsoft's Web Application Proxy role in Windows Server 2012 R2 also provides support for publishing your SharePoint resources to the Internet.

The AWS CloudFormation template provided by this Quick Start allows you to choose the most appropriate architecture based on your requirements. If you choose to deploy a publicly accessible SharePoint farm, **we do not add additional reverse proxy servers into the DMZ for you**. We simply create an external elastic load balancer that is reachable from the Internet which communicates directly with your SharePoint web servers.

After the deployment, you can launch additional infrastructure in the public subnets and update the ELB to utilize new reverse proxy or threat management servers.

EC2 Instance Types

Properly planning for capacity and sizing servers is a key aspect of every enterprise application deployment. As such, it is important that you choose the appropriate Amazon EC2 instance type for each server role in your Microsoft SharePoint deployment. Since each deployment is different, you will need to use Microsoft's detailed guidance on how to properly size your environment based on the number of users and workloads involved. As a starting point, consider the minimum requirements for each server role.

The following values are based on minimum requirements for all server roles operating in a three-tier farm.

Role	Processor	RAM	Boot Volume
Web Front-End Server / Front-End Server	64-bit, 4 cores	12 GB	80 GB
Application Server / Batch-Processing / Back-End	64-bit, 4 cores	12 GB	80 GB
Database Server (fewer than 1,000 users)	64-bit, 4 cores	8 GB	80 GB
Database Server (between 1,000 and 10,000 users)	64-bit, 8 cores	16 GB	80 GB

The Quick Start uses the following instance types by default. This gives you additional capacity over the absolute minimum requirements as a starting point.

Role	EC2 Instance Type	Boot Volume
Web Front End Server / Front-End Server	c3.2xlarge (8 vCPU, 15 GiB Memory)	100 GiB (EBS/GP2)
Application Server / Batch-Processing / Back-End	c3.2xlarge (8 vCPU, 15 GiB Memory)	100 GiB (EBS/GP2)
Database Server	r3.2xlarge (8 vCPU, 61 GiB Memory)	100 GiB (EBS/GP2)

Amazon Elastic Block Store (Amazon EBS) volumes are used as the boot volume for each instance. Notice that we utilize the EBS General Purpose (GP2) volume type. This is an SSD-backed EBS volume that is now used as the default boot volume type for all Amazon EC2 instances. These GP2 volumes provide a consistent baseline of 3 IOPS/GB and are burstable up to 3,000 IOPS.

When you launch the AWS CloudFormation template in this guide, you'll be given the opportunity to adjust these instance types.

Customize Your Topology at Template Launch

Since your SharePoint Server 2013 farm design depends greatly on your own requirements, our automated solution allows you to choose how many SharePoint servers to deploy into your environment. At a minimum, we want to provide a highly available architecture, so each role should be present in each Availability Zone.

The screenshot shows a configuration form for a SharePoint Server 2013 farm. It includes three input fields:

- SPFarmAccount**: A text input field containing 'spfarm' with the label 'User name for the SP Farm account.'
- SPFarmAccountPassword**: A password input field containing '*****' with the label 'Password for the SP Farm account. Must be at least 8 characters containing letters, numbers and symbols.'
- SPServersPerAZ**: A text input field containing '2' with the label 'Specify the number of SharePoint servers to deploy in each AZ. Allowed values are 2, 3, or 4'. This field is highlighted with a yellow background.

Figure 18: Choosing the Number of SharePoint Servers at Template Launch

When launching the template, you can use the **SPServersPerAZ** parameter to define the number of servers in each Availability Zone, as shown in Figure 18. Valid options are 2, 3, and 4.

- **Two SharePoint servers per Availability Zone** provide the minimum number of servers for high availability. This option can be used to deploy a farm based on either the Traditional or Streamlined topologies. The servers named SP1 and SP2 will receive http requests from Elastic Load Balancing. The servers named SP3 and SP4 can provide application server or batch-processing functionality.
- **Three SharePoint servers per Availability Zone** will give you a third set of servers for dedicated services. This option is more likely to be used for Streamlined topologies. The servers named SP1 and SP2 will receive http requests from Elastic Load Balancing. The servers named SP3 and SP4 can provide application server or batch-processing functionality. The servers named SP5 and SP6 can be used as dedicated Distributed Cache or Specialized Workload servers.
- **Four SharePoint servers per Availability Zone** provide enough infrastructure for a large farm in a Streamlined topology. This option provides a set of instances in each Availability Zone for Distributed Cache and Request Management, Front-End servers, Batch-Processing servers, and Specialized Workload servers.

As your SharePoint servers are launched, the servers are renamed, joined to the Active Directory domain, and the SharePoint 2013 prerequisites are installed on each server. The farm is created after the installation of SharePoint on the first server, and the remaining servers are installed and joined to the farm. The automated solution is complete after this step. After your stack has been created successfully, you can RDP into your environment and navigate to SharePoint Central Administration (<http://sp1:18473/>) to configure your farm components, services, and service applications.

The default value for **SPServersPerAZ** is 2. If you launch the AWS CloudFormation template and accept the default parameters, you will deploy the following architecture that was introduced at the beginning of this guide.

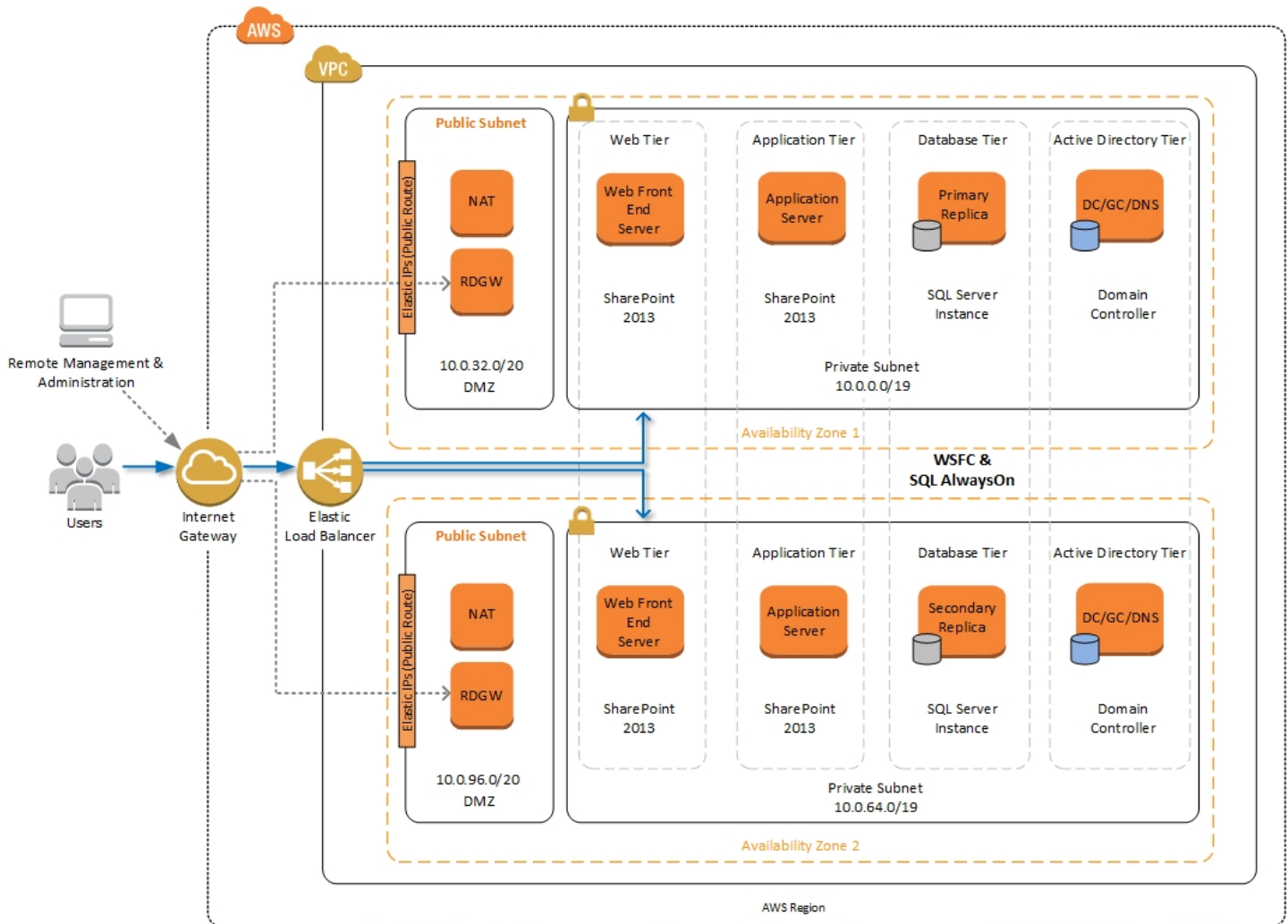


Figure 19: Highly Available SharePoint Farm on AWS

SharePoint Server 2013 Quick Start Deployment Steps

This Quick Start deploys in three simple steps, followed by an optional testing step. First, we need to launch the Microsoft WSFC and SQL Server AlwaysOn Quick Start. This step provides the foundational elements for the environment, which include the Amazon VPC spanning two Availability Zones, two Active Directory Domain Controllers, and two Microsoft SQL Server Enterprise nodes configured in a Windows Server Failover Cluster.

After the foundation is in place, we need to create an Amazon EBS volume that contains the Microsoft SharePoint Server 2013 installation media. In order to install SharePoint Server 2013 on Windows Server 2012 R2, you should ensure that you use the SharePoint Server 2013 bits with SP1 slipstreamed into the media. Currently, there is not a trial version of SharePoint Server 2013 with SP1 publicly available. Therefore, in order to install the most current version of SharePoint server on the latest Windows operating system, you must first download your licensed SharePoint Server 2013 media from Microsoft and create an Amazon EBS volume snapshot that can be used by every SharePoint server in the farm to perform the installation.

After you've created your SharePoint media volume, you can launch the final AWS CloudFormation template which will build your SharePoint Server 2013 farm, and test your servers. Instructions for each step are covered in the following sections.

1. Launch Microsoft Windows Server Failover Cluster and SQL Server AlwaysOn Quick Start

The Microsoft WSFC and SQL Server AlwaysOn Quick Start provides the Active Directory Domain Services and Database Server tiers for our SharePoint 2013 environment. This Quick Start Reference Deployment includes an AWS CloudFormation template that launches an Active Directory infrastructure along with clustered SQL Server 2012 or 2014 instances. The Quick Start Reference Deployment for [Microsoft WSFC and SQL AlwaysOn on AWS](#) provides architectural considerations and configuration steps for running WSFC clusters in the AWS cloud and instructions for configuring and testing the WSFC cluster and a SQL Server 2012 or 2014 AlwaysOn Availability Group. This stack takes approximately three hours to create.

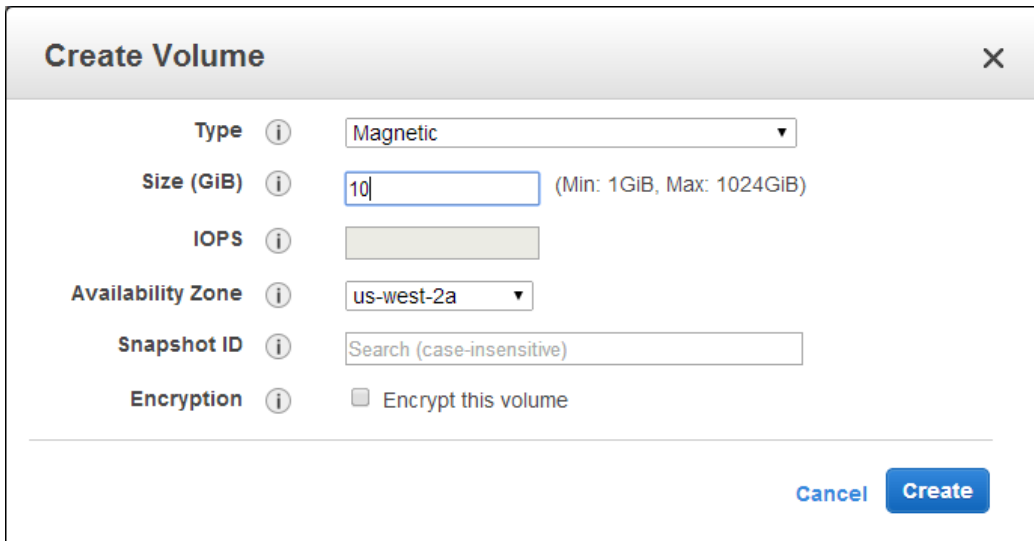
To launch the Microsoft WSFC and SQL Server AlwaysOn AWS CloudFormation template in the US-West (Oregon) region, [launch the Quick Start](#).

You can also download the AWS CloudFormation template directly from https://s3.amazonaws.com/quickstart-reference/microsoft/sql/latest/templates/SQL_AlwaysOn_Master.template.

2. Prepare a Media Volume Snapshot

After completing Step 1, you will have a fully functional Active Directory Domain Services deployment, including SQL Server Enterprise instances. The next step is to create a new Amazon EBS volume snapshot that will be used to store the SharePoint installation media. Follow the steps below to create the volume snapshot.

1. In the Amazon EC2 console, navigate to **Elastic Block Store > Volumes** and click **Create Volume**.



The screenshot shows the 'Create Volume' dialog box with the following fields and values:

- Type: Magnetic
- Size (GiB): 10 (Min: 1GiB, Max: 1024GiB)
- IOPS: (empty)
- Availability Zone: us-west-2a
- Snapshot ID: Search (case-insensitive)
- Encryption: Encrypt this volume

Buttons: Cancel, Create

Figure 20: Creating a new Amazon EBS Volume

2. With the newly created volume selected, click the **Actions** drop-down list and select **Attach Volume**.
3. Choose the Remote Desktop Gateway (RDGW1) in the first Availability Zone as the instance this volume will be attached to, and then click **Attach**.

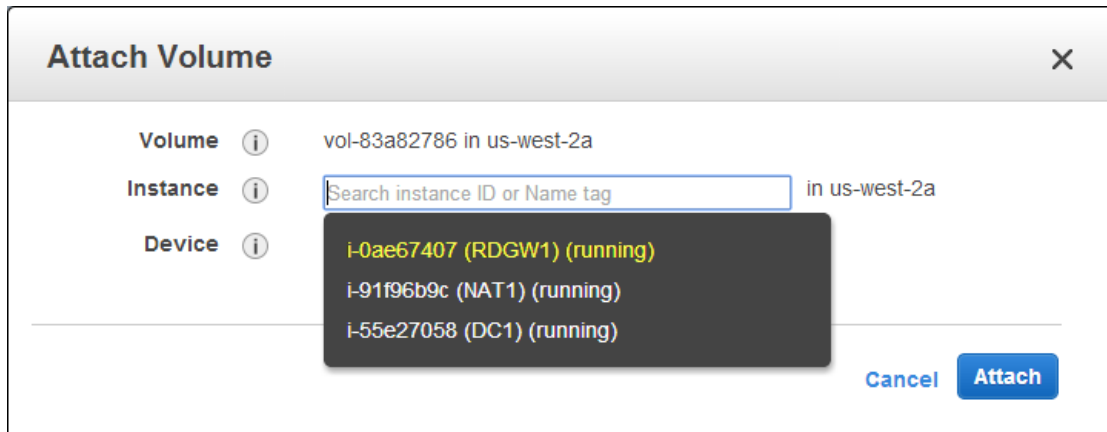


Figure 21: Attaching the new EBS Volume to the RDGW1 Instance

- Open the Remote Desktop Connection client (mstsc.exe) and connect to the RDGW1 instance. You can find the Elastic IP for the RDGW1 instance in the Amazon EC2 console, under Instances.

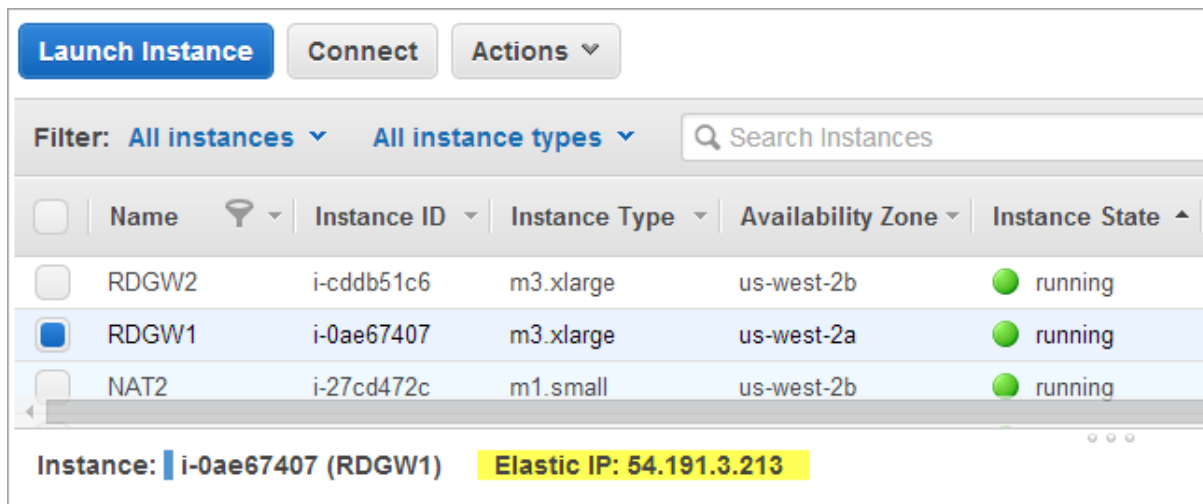


Figure 22: RDGW1 Elastic IP on the Amazon EC2 Console

- When you are connected to the desktop of the RDGW1 server, you are ready to download **SharePoint 2013 with SP1**. The source of your download depends on your current licensing agreement with Microsoft. By default, the installation uses a trial key for the deployment. Keep in mind that in order to use a licensed version of SharePoint Server, you must utilize [License Mobility Through Software Assurance](#).
- The download will likely be in the form of an ISO image. Windows Server 2012 provides native support for ISO files, so you can simply double-click the file to mount the image. Next, you should copy the files from the installation media to your empty Amazon EBS volume.

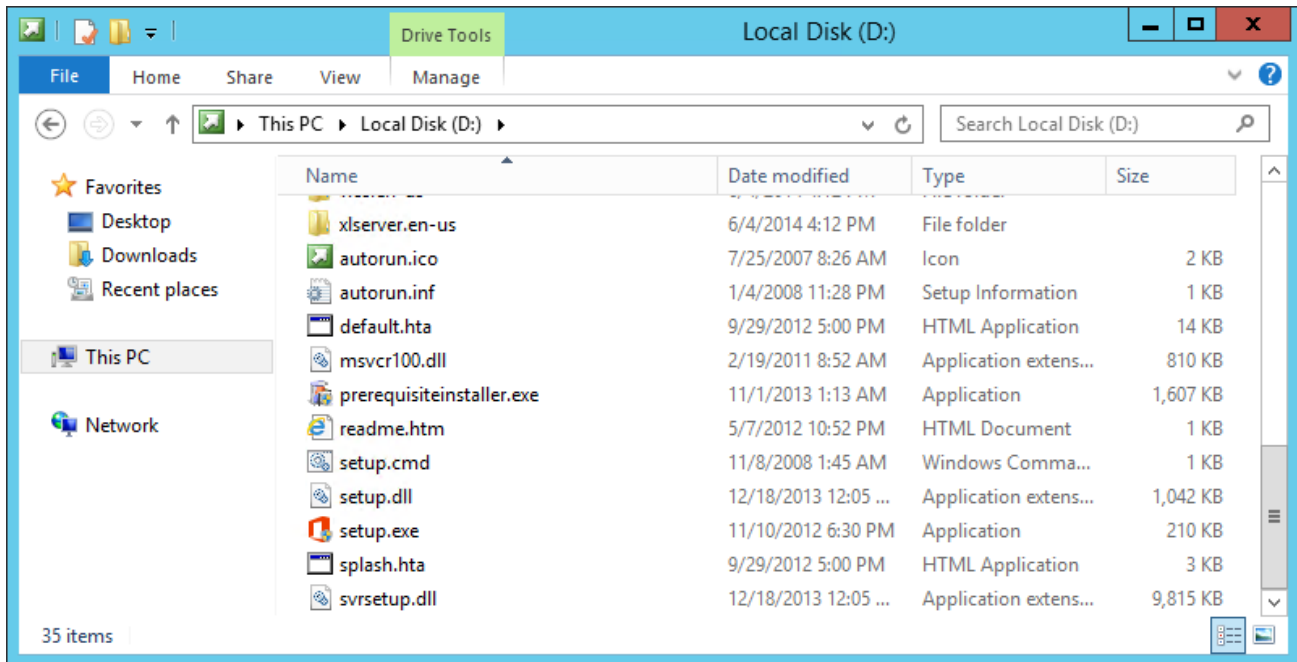


Figure 23: Copying the SharePoint Media to the Empty Amazon EBS Volume

Note: Make sure that you copy the files into the root of the EBS volume as shown in Figure 23. We recommend that you use the XCOPY command to remove the read-only attribute when copying the files, as outlined in [KB323002](#) on the Microsoft Support website.

- Now that the media has been copied to the Amazon EBS volume, we are ready to create a snapshot of this volume. In the Amazon EC2 console, navigate to **Elastic Block Store > Volumes** and select the volume you created previously. From the **Actions** drop down menu select **Create Snapshot**.

Provide a Name and Description for the Snapshot and click **Create**.

Create Snapshot ✕

Volume ⓘ vol-c4a629c1

Name ⓘ

Description ⓘ

Encrypted ⓘ No

Figure 24: Creating an Amazon EBS Snapshot

After the snapshot is created, navigate to **Elastic Block Store > Snapshots** and make a note of the snapshot ID, which will be required input when you launch the AWS CloudFormation template for SharePoint.

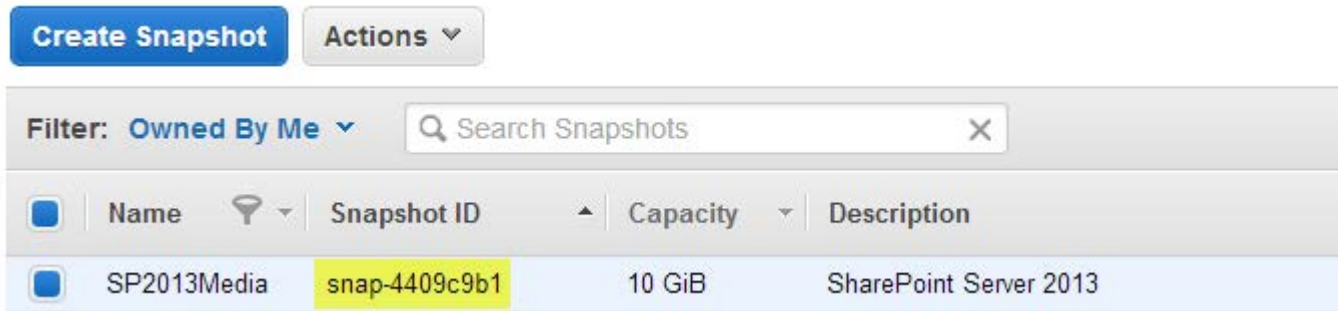


Figure 25: Amazon EBS Snapshot ID

3. Launch the SharePoint Server 2013 Stack

This automated AWS CloudFormation template deploys SharePoint 2013 servers in multiple Availability Zones into an Amazon VPC. Please ensure you've created your media snapshot as explained in the previous section before launching the stack.

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

This stack takes approximately one hour 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 \$9.00 an hour, and you can complete the initial deployment for about \$25.00. See the pricing pages of the AWS services you will be using for full details.

You can also download the template directly from https://s3.amazonaws.com/quickstart-reference/microsoft/sharepoint/latest/templates/Template_1_SharePoint_2013.template.

Template Customization

This automation allows for rich customization of 30 defined 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
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)
DomainAdminUser	stackadmin	User name for the account that will be 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, numbers and symbols.
SPFarmAccount	spfarm	User name for the SP Farm account
SPFarmAccountPassword	Password123	Password for the SP Farm account. Must be at least 8 characters.
SPKey	NQTMW-K63MQ-39G6H-B2CH9-FRDWJ	The Product Key for SharePoint 2013. The trial key is provided by default. You can replace with your own key.
ELBConfiguration	external	Specify whether or not you want to include an Elastic Load Balancer. Allowed values are internal or external.
IncludeOfficeWebApps	false	To include an Office Web Apps Server in each AZ, set this parameter to true
SPInstanceType	c3.2xlarge	Amazon EC2 instance type for the SharePoint Servers
OFFICEWEBInstanceType	m3.xlarge	Amazon EC2 instance type for the Office Web Apps Servers
SPServersPerAZ	2	Specify the number of SharePoint servers to deploy in each Amazon AZ. Allowed values are 2, 3, or 4.
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
PrivateSubnet1	<User Provided>	ID of the private subnet in AZ1 (e.g., subnet-a0246dcd)
PrivateSubnet2	<User Provided>	ID of the private subnet in AZ2 (e.g., subnet-a0246dcd)
SPSnapShotID	<User Provided>	ID of the volume snapshot containing the SharePoint Server 2013 installation media (e.g., snap-dc834329)
DomainMemberSGID	<User Provided>	ID of the Domain Member Security Group (e.g., sg-7f16e910)
VPCID	<User Provided>	ID of the Amazon VPC (e.g., vpc-0343606e)
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
ADServerNetBiosName1	DC1	NetBIOS name of the existing Domain Controller in AZ1
ADServerNetBiosName2	DC2	NetBIOS name of the existing Domain Controller in AZ2

4. Test High Availability and Automatic Failover

In this section, we'll walk you through testing high availability and automatic failover of your SharePoint servers. We'll assume that you've used the default parameter values in the AWS CloudFormation template with an externally facing ELB load balancer. In this scenario, we'll assume that the SharePoint farm is hosting a public facing website, and we'll set up a simple blog in order to validate our test.

After you have successfully launched the stack, remote into the environment through one of the Remote Desktop Gateway (RD Gateway) instances. You can retrieve the Elastic IP Address for each RD Gateway instance from the Amazon EC2 console. You can use the [Remote Desktop Gateway Quick Start](#) to fully configure your RD Gateway instances, or you can simply connect to the desktop of your RD Gateway instances, and then start a new RDP client to connect internally to your servers.

1. Establish an RDP session to the SP1 server. Start Internet Explorer with administrative permissions and navigate to **SharePoint Central Administration** (<http://sp1:18473/>).
2. Under **Application Management**, click **Manage web applications**.

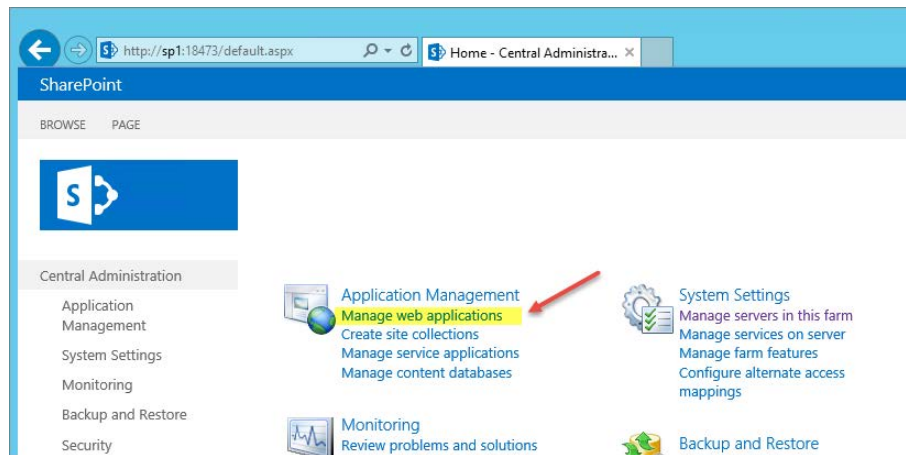


Figure 26: SharePoint Central Administration

3. Click **New** to create a new web application.

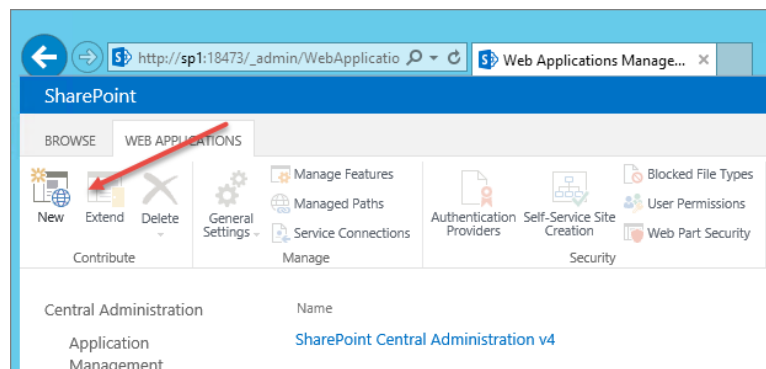


Figure 27: Create a New Web Application

4. You need to change only one setting in the **Create New Web Application** dialog box. Set **Allow Anonymous** to **Yes** as shown in Figure 28, and then click **OK**.

Figure 28: The Create New Web Application Dialog Box

- After the web application has been created, navigate back to SharePoint Central Administration and click **Create site collections**.

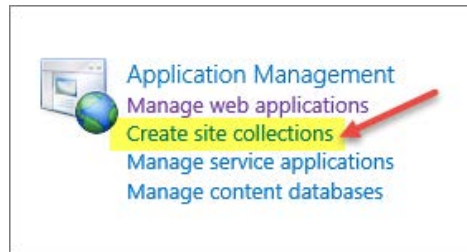


Figure 29: Creating Site Collections

- Provide a **Title** for your site, and then select the **Blog** template on the **Collaboration** tab. You'll also need to define a **Primary Site Collection Administrator** on this page, as shown in Figure 31. You can use the StackAdmin user account for this value. When you finish filling out the form, click **OK**.

Title and Description
Type a title and description for your new site. The title will be displayed on each page in the site.

Title:

Description:

Web Site Address
Specify the URL name and URL path to create a new site, or choose to create a site at a specific path.

To add a new URL Path go to the [Define Managed Paths](#) page.

URL:

Template Selection

Select experience version:

Select a template:

- Collaboration
- Enterprise
- Publishing
- Custom
- Team Site
- Blog**
- Developer Site
- Project Site

Figure 30: Creating a Blog Site

Primary Site Collection Administrator
Specify the administrator for this site collection. Only one user login can be provided; security groups are not supported.

User name:

Figure 31: Setting the Primary Site Collection Administrator

- Now that you have created a blog, navigate to <http://sp1>. Note that this site is listening on the default HTTP port 80, so make sure that your browser does not autocomplete the port number for Central Administration in the URL. In the upper-right corner, click the gear icon, and then click **Site settings**.

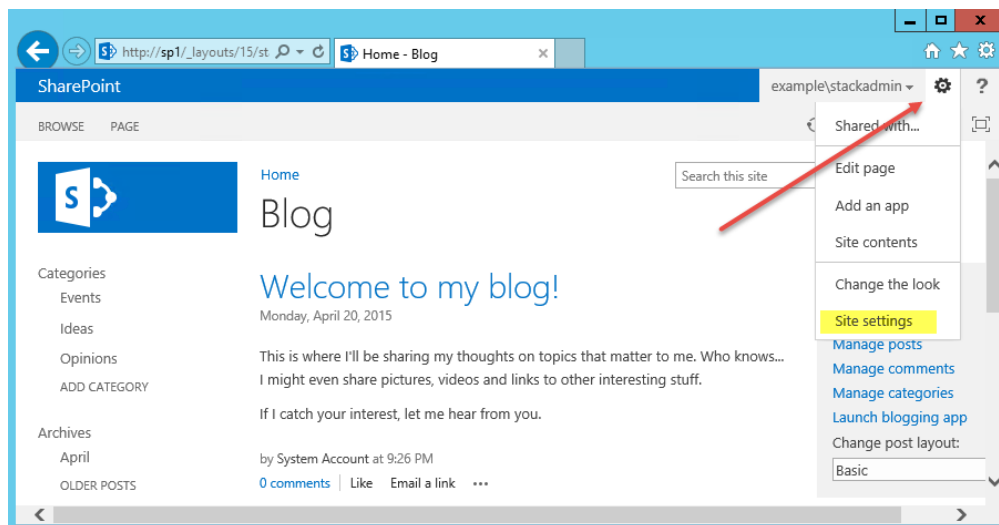


Figure 32: Modifying Site Settings for the Blog

- Under **Site Settings**, click **Site permissions** to open the **Permissions** page. On the ribbon, click **Anonymous Access**. In the **Anonymous Access** dialog box, click **Entire Web site**, and then click **OK**.

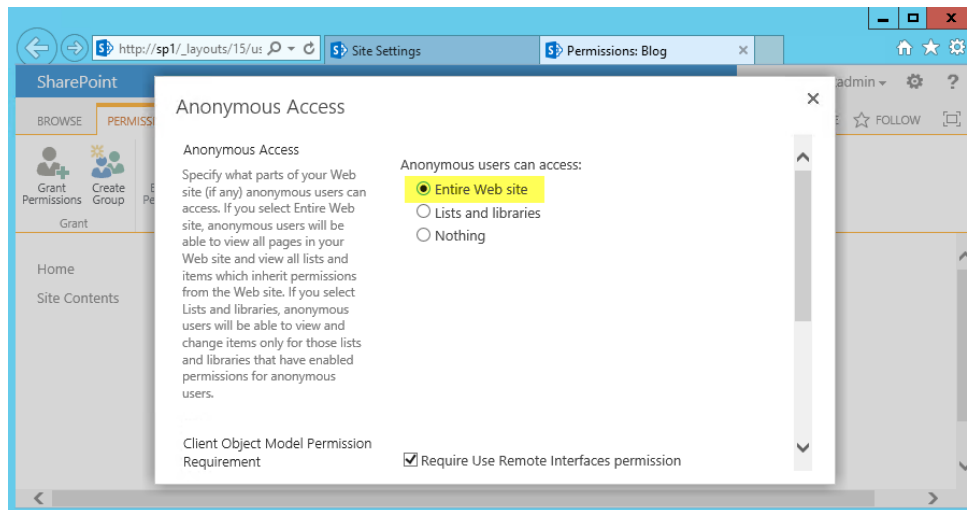


Figure 33: Enabling Anonymous Access

- At this point, we're ready to make our SharePoint databases highly available. Establish an RDP session to the WSFCNODE1 instance. Start SQL Server Management Studio, and then click **Connect** to connect to the local server.

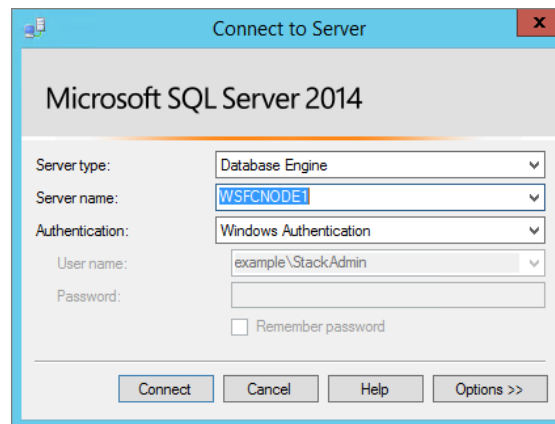


Figure 34: Connecting to WSFCNODE1

- Expand the **Databases** node in the Object Explorer and make a backup of each SharePoint database. The databases you'll need to back up are AdminDB, SPConfigDB, and WSS_Content. To make a backup, right-click the database name, click **Tasks**, and then click **Back Up**. Keep the default settings, and then click **OK** to perform the backup.

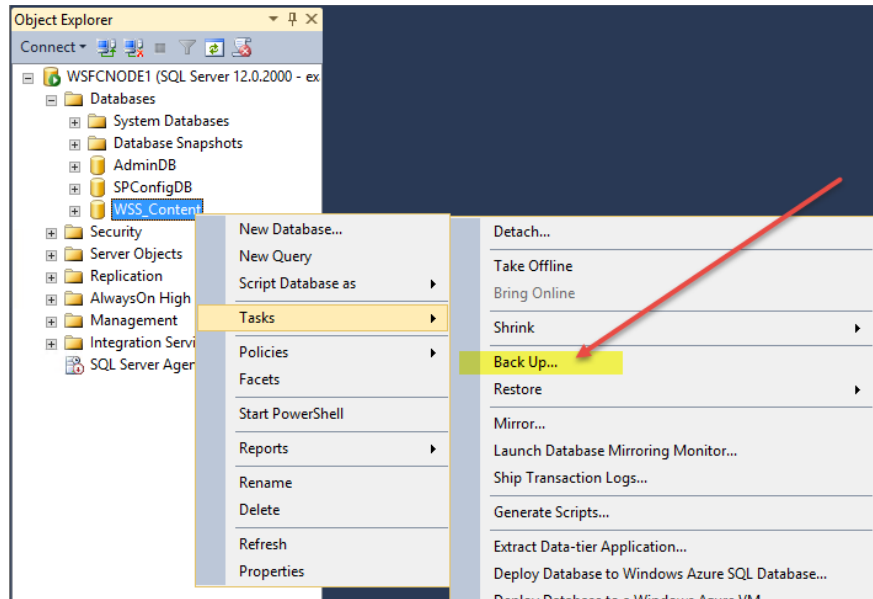


Figure 35: Backing up a database

- When the databases have been backed up, right-click **AlwaysOn High Availability** in the Object Explorer, and then click **New Availability Group Wizard**. Provide a name for the availability group, and click **Next**. In this example, we'll use **SharepointAG** as the name of the group.

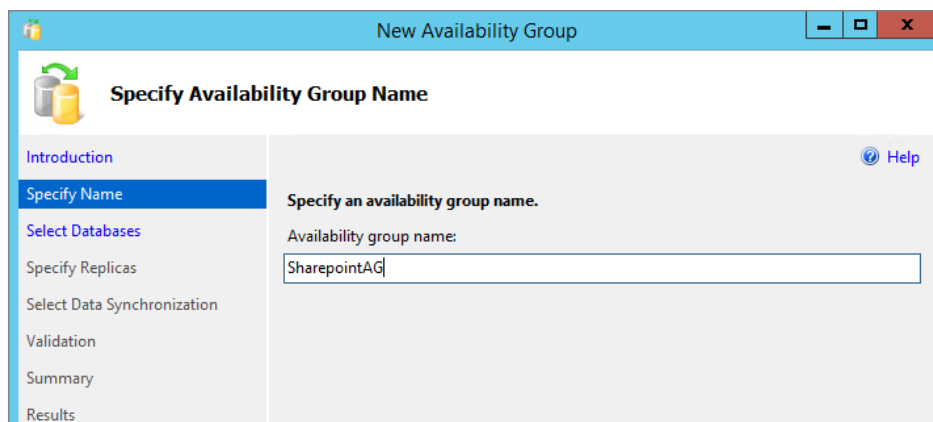


Figure 36: Naming the Availability Group

- Select the databases you previously backed up, and then click **Next**.

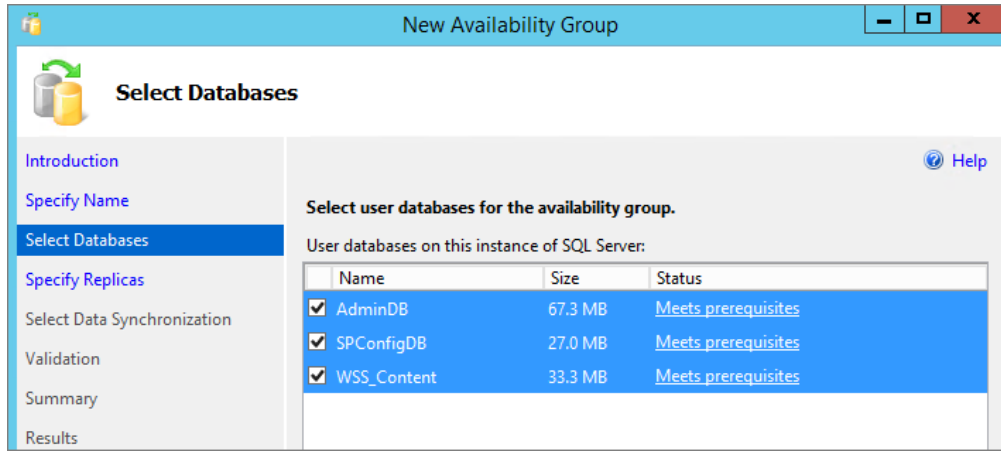


Figure 37: Selecting Availability Group Databases

13. On the **Specify Replicas** page, add WSFCNODE2 as a replica. Ensure that the check boxes for automatic failover and synchronous replication are selected, as shown in Figure 38.

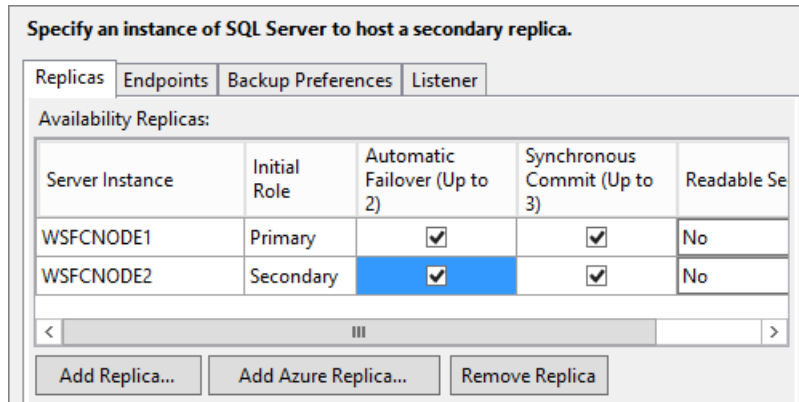


Figure 38: Selecting Availability Group Databases

14. On the **Specify Replicas** page, click the **Listener** tab. Provide a listener DNS name, the port number to listen on (which will be 1433), and the IP address for each WSFC node. Based on the template default settings, the IP addresses should be 10.0.0.102 for WSFCNODE1, and 10.0.64.102 for WSFCNODE2. When you’ve filled out the page as shown in Figure 39, click **Next**.

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

Replicas Endpoints Backup Preferences **Listener**

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

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

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

Listener DNS Name:

Port:

Network Mode:

Subnet	IP Address
10.0.64.0/19	10.0.64.102
10.0.0.0/19	10.0.0.102

Figure 39: Configuring the Availability Group Listener

15. On the **Select Initial Data Synchronization** page, click **Full** and enter `\\dc1\replica` as the network share to use for synchronizing the data. Click **Next**.

New Availability Group

Select Initial Data Synchronization

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

Select your data synchronization preference.

Full
Starts data synchronization by performing full database and log backups for each selected database. These databases are restored to each secondary and joined to the availability group.
Specify a shared network location accessible by all replicas:

Join only
Starts data synchronization where you have already restored database and log backups to each secondary server. The selected databases are joined to the availability group on each secondary. This action will be skipped for Azure replicas.

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

Figure 40: Configuring the Availability Group listener

16. Accept the default settings on the remaining pages of the wizard, and then click **Next** and **Finish** to build the availability group. Ensure that the wizard completes successfully before moving on to the next step.

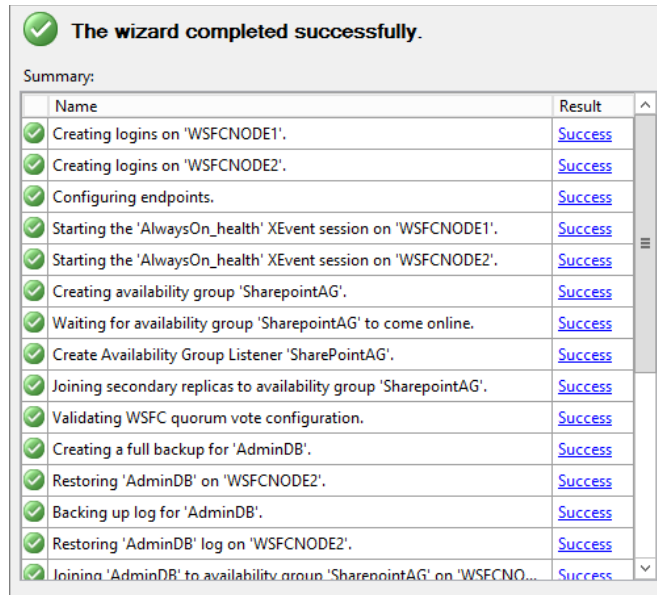


Figure 41: Successful Completion of the AlwaysOn Availability Group Wizard

17. Now that the availability group has been created, you should force AD replication from DC1 to DC2 to ensure that the DNS records for your availability group listener can be resolved in the secondary AD site in Availability Zone 2. Connect to DC1 and run the command **repadmin /syncall /A /e /P** as shown in Figure 42.

```
Administrator: Command Prompt
C:\>repadmin /syncall /A /e /P
Syncing all NC's held on DC1.
Syncing partition: DC=ForestDnsZones,DC=example,DC=com
CALLBACK MESSAGE: The following replication is in progress:
  From: 09fd8956-9a8b-4915-803e-497181330555._msdcs.example.com
  To   : 8f125e85-d40e-4f1a-ac8d-8ba7c09c07c3._msdcs.example.com
CALLBACK MESSAGE: The following replication completed successfully:
  From: 09fd8956-9a8b-4915-803e-497181330555._msdcs.example.com
  To   : 8f125e85-d40e-4f1a-ac8d-8ba7c09c07c3._msdcs.example.com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.
```

Figure 42: Forcing AD Replication from Availability Zone 1 to Availability Zone 2

18. Next you'll need to update the SQL client alias on each SharePoint server. Use the command **cliconfg** on each server to bring up the **SQL Server Client Network Utility** shown in Figure 43. On the **Alias** tab, modify the **SQL** alias to resolve to the availability group listener DNS name *instead* of the WSFCNODE1 server. You might need to restart the SharePoint services or restart your SharePoint servers for the change to take effect.

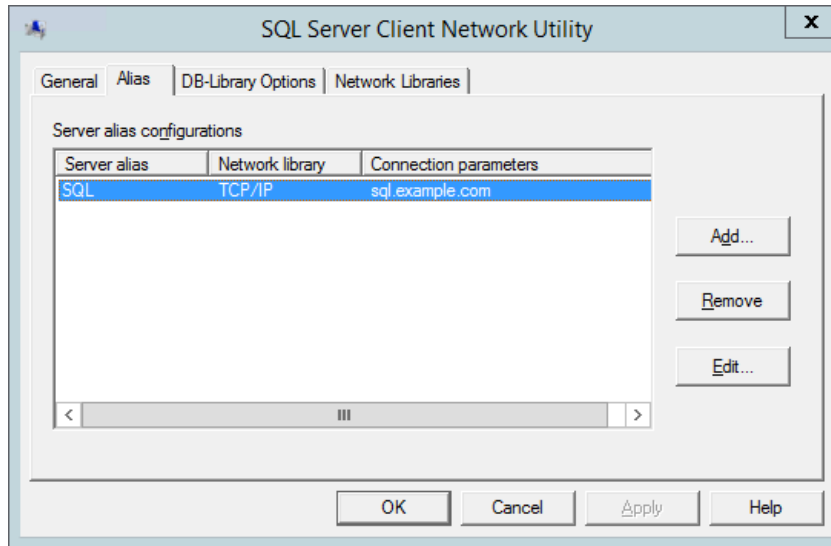


Figure 43: Modifying the SQL Alias

19. On the SP1 server, run Windows PowerShell with administrative permissions and execute the following PowerShell code to enable multi-subnet failover for the SharePoint databases.

```
Add-PSSnapin Microsoft.SharePoint.PowerShell

$dbcs = Get-SPDatabase | ?{$_MultiSubnetFailover -ne $true}

foreach ($db in $dbcs) {
    $db.MultiSubnetFailover = $true
    $db.Update()
}
```

Figure 44: Enabling Multi-Subnet Failover for the SharePoint Databases

20. Navigate to the Amazon EC2 console. In the navigation pane, under **Network & Security**, click **Load Balancers**. Record the DNS name of the ELB load balancer that was created by the AWS CloudFormation template.

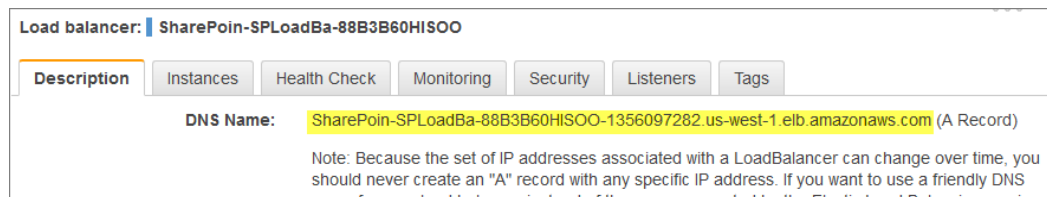


Figure 45: Retrieving the DNS name of the ELB Load Balancer

21. Navigate back to SharePoint Central Administration and click **System Settings** in the left column. Under **Farm Management**, click **Configure alternate access mappings**.

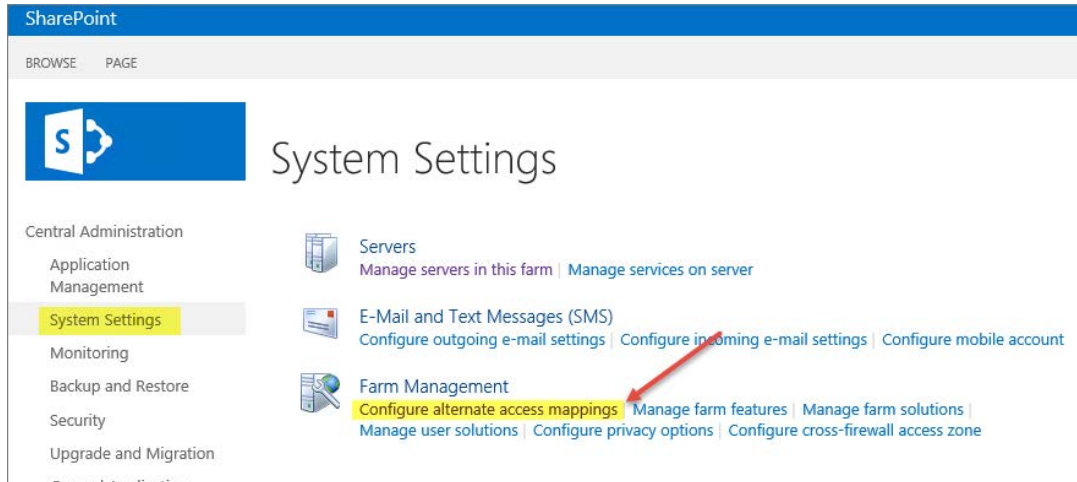


Figure 46: Configure Alternate Access Mappings Option

22. Edit the public zone URLs for your blog site collection, as shown in Figure 47. For the purposes of this test, the Internet zone URL should be the DNS name of the ELB load balancer you recorded in step 20. Remember that for production, you can have a CNAME record (such as `sharepoint.example.com`) that resolves to the ELB load balancer DNS name.

Figure 47: Configuring Alternate Access Mappings

23. At this point, you should be able to access your SharePoint-based blog **externally** by using the ELB load balancer DNS name. Visit the site to confirm that it is publicly available.
24. After your externally facing SharePoint site is available, you can test automatic failover. The primary database server should be `WSFCNODE1`, and the ELB load balancer will be distributing HTTP requests across `SP1` and `SP2`. To verify that automatic failover is functional, forcibly stop `WSFCNODE1` and `SP1` from the Amazon EC2 console. You can stop the instances simultaneously to perform this test, as shown in Figure 48.

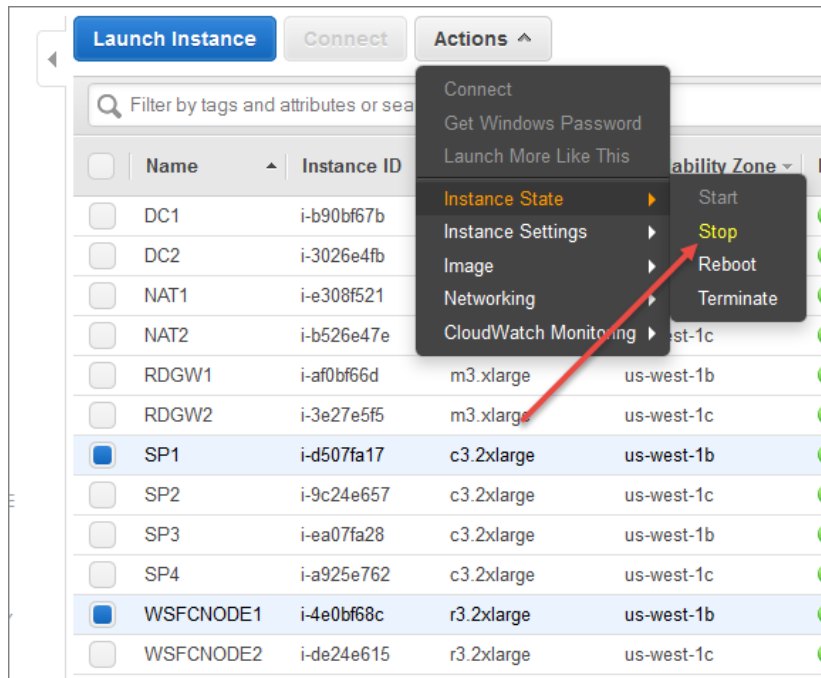


Figure 48: Stopping the Instances in Availability Zone 1

25. After you've simulated a failure by stopping the instances, the SharePoint databases should fail over automatically to WSFCNODE2, and the ELB load balancer should detect that SP1 has gone offline and direct HTTP traffic to SP2. You can revisit the site in your web browser to confirm that everything is still working.

Post-Configuration Tasks

If you've included Microsoft Office Web Apps servers in your template launch, you will need to configure them to work with your SharePoint farm. For configuration steps, see [Configure Office Web Apps for SharePoint 2013](#) on the Microsoft TechNet site. You'll need to download and install the Office Web Apps Server components from Microsoft.

Further Reading

- Microsoft on AWS:
 - <http://aws.amazon.com/microsoft/>
- Amazon EC2 Windows Guide:
 - <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/>
- AWS for Windows and .NET Developer Center:
 - <http://aws.amazon.com/net>
- Microsoft License Mobility:

- <http://aws.amazon.com/windows/mslicenseability>
- Whitepapers:
 - Active Directory Reference Architecture
https://s3.amazonaws.com/quickstart-reference/microsoft/activedirectory/latest/doc/Microsoft_Active_Directory_Quick_Start.pdf
 - Implementing Microsoft Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups in the AWS Cloud
https://s3.amazonaws.com/quickstart-reference/microsoft/sql/latest/doc/Microsoft_WSFC_and_SQL_AlwaysOn_Quick_Start.pdf
 - Remote Desktop Gateway Reference Deployment
https://s3.amazonaws.com/quickstart-reference/microsoft/rdgateway/latest/doc/Microsoft_Remote_Desktop_Gateway_Quick_Start.pdf
 - Securing the Microsoft Platform on AWS
http://media.amazonwebservices.com/AWS_Microsoft_Platform_Security.pdf
- Microsoft content:
 - Configure SQL Server 2012 AlwaysOn Availability Groups for SharePoint 2013
[http://technet.microsoft.com/en-us/library/jj715261\(v=office.15\).aspx](http://technet.microsoft.com/en-us/library/jj715261(v=office.15).aspx)
 - Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups
<http://msdn.microsoft.com/library/hh213417.aspx>

Send Us Your Feedback

Please post your feedback or questions on the [AWS Quick Start Discussion Forum](#).

Document Revisions

Date	Change	In section
April 2015	Added information about testing high availability and automatic failover of SharePoint servers	Step 4. Test High Availability and Automatic Failover
March 2015	Optimized the underlying Amazon VPC design to support expansion and to reduce complexity	Architecture diagram and template updates
December 2014	Added information about enabling MultiSubnetFailover before modifying the SQL Client Alias.	Post-configuration Tasks

© 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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.