Microsoft SharePoint Server 2016 on the AWS Cloud

Quick Start Reference Deployment

AWS Quick Start team

May 2016
Last updated: July 2019 (see revisions)

This guide is also available in HTML format at
https://docs.aws.amazon.com/quickstart/latest/sharepoint/.
# Contents

About This Guide .................................................................................................................. 3
Quick Links ................................................................................................................................. 4
About Quick Starts ..................................................................................................................... 4
Overview .................................................................................................................................... 4
SharePoint Server 2016 on AWS ............................................................................................... 4
Cost and Licenses ....................................................................................................................... 5
AWS Services ............................................................................................................................. 6
Architecture ................................................................................................................................. 7
VPC Configuration ....................................................................................................................... 9
Remote Administration ............................................................................................................... 11
Active Directory Domain Services .......................................................................................... 12
Deployment Steps ...................................................................................................................... 13
Step 1. Prepare an AWS Account .............................................................................................. 13
Step 2. Download the SharePoint Software ............................................................................... 16
Step 3. Launch the SharePoint Stack .......................................................................................... 16
Step 4. Create Initial Content .................................................................................................... 19
Step 5. Make the SharePoint Databases Highly Available ....................................................... 24
Step 6. Test Automatic Failover ................................................................................................ 31
Troubleshooting ......................................................................................................................... 32
Additional Resources .................................................................................................................. 33
Appendix A: Server Role Architecture ....................................................................................... 35
  Traditional Topology ................................................................................................................ 35
    Web Tier ................................................................................................................................. 35
    Application Tier .................................................................................................................... 36
    Database Tier ....................................................................................................................... 37
  Streamlined Topology ............................................................................................................. 39
    Front-End Servers ............................................................................................................... 39
Batch-Processing Servers ........................................................................................................39
Database Servers ....................................................................................................................39
Distributed Cache ...................................................................................................................40
Request Management ............................................................................................................40
Specialized Workloads ..........................................................................................................40
Search ....................................................................................................................................40
Simple Example of a Streamlined Topology ........................................................................40
Office Online Server .............................................................................................................41
Intranet SharePoint Server Farm on AWS ...............................................................................43
Security ..................................................................................................................................45
Security Groups ........................................................................................................................45
Network ACLs ..........................................................................................................................46
Secure Website Publishing .......................................................................................................46
EC2 Instance Types ................................................................................................................48
Customize Your Topology at Template Launch ....................................................................49
Appendix B: AWS CloudFormation Template Parameters ..................................................51
GitHub Repository ..................................................................................................................56
Document Revisions ..............................................................................................................56

About This Guide
This Quick Start reference deployment guide discusses architectural considerations and configuration steps for building a Microsoft SharePoint Server 2016 environment on the Amazon Web Services (AWS) cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment.

This guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend SharePoint Server 2016 on the AWS Cloud. The guide requires basic familiarity with SharePoint Server architecture and management. For more information about SharePoint Server, including general guidance and best practices, consult the Microsoft SharePoint product documentation.
Quick Links
The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, network security, and other considerations discussed in this guide.

- If you have an AWS account, and you’re already familiar with AWS services and SharePoint, you can launch the Quick Start to build the architecture shown in Figure 2 in a new virtual private cloud (VPC) in your AWS account. The deployment takes approximately three hours. If you’re new to AWS or to this SharePoint Quick Start, please review the implementation details and follow the step-by-step instructions provided later in this guide.

- If you want to take a look under the covers, you can view the AWS CloudFormation template that automates the deployment.

About Quick Starts
Quick Starts are automated reference deployments for key enterprise workloads on the AWS Cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview
SharePoint Server 2016 on AWS
The Amazon Web Services (AWS) cloud provides a suite of infrastructure services that enable you to deploy SharePoint Server 2016 securely, affordably, and with high availability. Running SharePoint Server on the AWS Cloud gives you flexibility and agility, and you can fully customize and extend SharePoint for your business processes.

This Quick Start implementation guide walks you through the steps to automatically deploy an enterprise SharePoint Server 2016 architecture in your own AWS account. The
automatic deployment, including Active Directory and SQL Server, takes approximately three hours.

**Cost and Licenses**

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start itself.

The AWS CloudFormation template for the SharePoint Server 2016 Quick Start includes configuration parameters that you can customize, and some settings, such as the instance types and the number of instances, can greatly affect the cost of the deployment.

[Amazon Web Services](#) has published a whitepaper that shows how to estimate the cost of your SharePoint deployment. You have a wide array of options for building your SharePoint farm, and it’s not possible to cover them all in that whitepaper or in this guide. The following table offers a model based on some key assumptions. You can [open an example in the Simple Monthly Calculator](#) to change the configuration and revise any of these estimates to fit your scenario.

- It assumes that you launch the Quick Start AWS CloudFormation template with the default parameters. The traditional topology architecture shown in [Figure 2](#) includes 10 instances.
- It assumes 15 TiB of outbound data traffic per month (based on 50 MiB per day for 20,000 users). This accounts for about $1,300 of the monthly cost.
- It assumes storage and backups for about 5 TiB of data. This accounts for about $2,100 of the monthly cost.

<table>
<thead>
<tr>
<th>Model</th>
<th>Up-front cost</th>
<th>Monthly cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-Demand Instances, license Windows Server from AWS, use free trial SharePoint Server and SQL Server licenses</td>
<td>$0</td>
<td>$8,700</td>
</tr>
</tbody>
</table>

This approach represents an average of 40% savings over the typical cost to deploy an on-premises SharePoint solution. You can get an idea of the savings you may see for your specific deployment by using the [AWS TCO Calculator](#). For more information about instance pricing, see [Instance Purchasing Options](#) in the AWS documentation. Please note that AWS prices are subject to change.

This SharePoint Quick Start (using free trial licenses for SQL Server and SharePoint Server) is most appropriate for a trial or proof-of-concept project.
Figure 1: Use the AWS Simple Monthly Calculator to estimate the deployment costs for SharePoint Server

By default, this Quick Start installs the evaluation edition of SharePoint Server 2016 and SQL Server provided by Microsoft. For production environments, you can license SharePoint Server and SQL Server through the Microsoft License Mobility through Software Assurance program, and use your own product key during deployment. For development and test environments, you can leverage your existing MSDN licenses using Amazon EC2 Dedicated Instances or Dedicated Hosts. For details, see the MSDN on AWS page. Note that this Quick Start doesn’t currently support deployment to Dedicated Hosts or Dedicated Instances.

AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see Getting Started with AWS.)

- **AWS CloudFormation** – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (e.g., Amazon EC2 instances) that you want. You don’t have to individually create and configure the resources or figure out dependencies—AWS CloudFormation handles all of that.

- **Amazon VPC** – The Amazon Virtual Private Cloud (Amazon VPC) service 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. 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.
• **Amazon EC2** – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.

• **NAT Gateway** – NAT Gateway is an AWS managed service that controls NAT gateway resources. A NAT gateway is a type of network address translation (NAT) device that enables instances in a private subnet to connect to the internet or to other AWS services, but prevents the internet from connecting to those instances.

• **IAM** – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can manage users, security credentials such as access keys, and permissions that control which AWS resources users can access, from a central location.

• **Amazon S3** – Amazon Simple Storage Service (Amazon S3) provides developers and IT teams with secure, durable, highly scalable, cost-effective object storage. Amazon S3 is easy to use and includes a web services interface to store and retrieve any amount of data from anywhere on the web. Object storage is not appropriate for workloads that require incremental data insertions, such as databases. However, Amazon S3 is an excellent service for storing snapshots of Amazon Elastic Block Store (Amazon EBS) volumes.

**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 2016: **traditional topology** and **streamlined topology**. The AWS CloudFormation template provided with this Quick Start is built with flexibility in mind, and lets you choose either topology for your SharePoint farm. Traditional and streamlined topologies are covered in detail in [Appendix A](#).

Deploying this Quick Start with the **default parameters** builds the following highly available SharePoint environment based on the traditional topology in the AWS Cloud.
The AWS CloudFormation template provided with this Quick Start sets up the virtual network and creates the networking resources. The template deploys a highly available architecture that includes redundant servers for Active Directory, SQL Server 2014, and SharePoint Server 2016 in two Availability Zones. Each Availability Zone includes a virtual private cloud (VPC) with two subnets, and supports remote administration. The subnets provide a public (DMZ) address space and a private address space. The public address space includes Remote Desktop (RD) Gateways and NAT gateway endpoints for outbound internet access. The private address space in each subnet hosts an Active Directory domain controller, a SharePoint Server web front-end server and application server, and a node in the SQL Server AlwaysOn Availability Group. Except for the Active Directory stack, which uses the Amazon Machine Image (AMI) for Windows Server 2016, all servers are bootstrapped from scratch using the base AMI for Microsoft Windows Server 2012 R2.
The following sections describe these components of the architecture in more detail. For more information about the server role architecture, including a detailed discussion of traditional and streamlined topologies, see Appendix A.

VPC Configuration
When deploying a Windows-based architecture on the AWS Cloud, we recommend an 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.
- Remote Desktop 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 VPC design used in this reference, see the Active Directory Domain Services Quick Start deployment guide.

Based on these best practices, the Quick Start deploys the following base-level VPC framework to support the SharePoint Server 2016 infrastructure:
As shown in Figure 3, NAT gateways are deployed into the public subnets. The public subnets have a route to the internet directly through the internet gateway attached to the 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 gateways 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 4: NAT gateways and Remote Desktop Gateways in public subnets
The AWS CloudFormation templates provided in this Quick Start automatically deploy the architecture described in the Quick Start for Remote Desktop Gateway on AWS. 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 for Remote Desktop Gateway 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 (AD DS). 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 5: Domain controllers in each Availability Zone](image-url)
Notice that in Figure 5, we’ve now included a domain controller in the Active Directory tier in each Availability Zone.

There are two ways to use AD DS in the AWS Cloud:

- Cloud only – This is the architecture shown in Figure 5. 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. In a hybrid architecture, we recommend that 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 for AD DS on AWS covers 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 VPC, public and private subnets, NAT gateway and RD Gateway instances, and domain controllers in each Availability Zone.

**Deployment Steps**

To build the SharePoint environment shown in Figure 2 on the AWS Cloud, follow these steps.

**Step 1. Prepare an AWS Account**

1. If you don’t already have an AWS account, create one at https://aws.amazon.com by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy SharePoint on AWS.

   Amazon EC2 locations are composed of Regions and Availability Zones. We currently offer twelve Regions in various geographic areas, with five more in development. Each Region includes at least two Availability Zones, which are isolated from each other with respect to power, network backbone, etc. Deploying your cloud applications across two Availability Zones helps you achieve high availability, even in the face of natural disasters that might impact a single Availability Zone.
Tip Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs, Create Key Pair**, type a name, and then choose **Create**.
Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. With Windows instances, we use the key pair to obtain the administrator password via the Amazon EC2 console, and then log in using Remote Desktop Protocol (RDP), as explained in the step-by-step instructions in the Amazon Elastic Compute Cloud User Guide.

4. Check for default subnets in the AWS Region you selected. The AWS CloudFormation template requires at least two default subnets in the Region you selected. To check, open the Amazon VPC console. In the navigation pane, choose Subnets, and then make sure that at least two of the subnets are marked as default subnets.

5. Verify that you have available Elastic IP addresses in your account. The AWS CloudFormation stack you will launch will automatically create Elastic IP addresses as needed for the SharePoint architecture. Each AWS account has a default limit of five addresses. To ensure beforehand that the AWS CloudFormation template will not fail because you’ve reached this limit, we recommend that you manually create two Elastic IP addresses in your AWS account and then delete them before you launch the AWS CloudFormation stack.

6. If necessary, request a service limit increase for the instance types used for the deployment. You might need to request an increase if you already have an existing deployment that uses the same instance types as your SharePoint architecture or if you need additional Elastic IP addresses. To do this, in the AWS Support Center, choose Create Case, Service Limit Increase, EC2 instances, and then complete the fields in the limit increase form. It can take a few days for the new service limit to become effective.
Step 2. Download the SharePoint Software
You will need access to SharePoint Server 2016 installation media in the form of an ISO disc image (.img) file. We recommend putting the .img file in an S3 bucket for the best performance, but you can also use an HTTP or HTTPS URI without Amazon S3.

Step 3. Launch the SharePoint Stack
The automated AWS CloudFormation template deploys SharePoint in multiple Availability Zones into a VPC.
1. **Launch the AWS CloudFormation template** into your AWS account.

   The template is launched in the US West (Oregon) Region by default. You can change the Region by using the Region selector in the navigation bar.

   This stack takes approximately three hours to create.

   You can also download the template to use it as a starting point for your own implementation.

2. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.

3. On the **Specify Details** page, review the parameters for the template. Provide values for the following required parameters.

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack Name</td>
<td>Requires input</td>
<td>Enter a name for this AWS CloudFormation stack. Later, you can delete the stack and all the resources associated with it.</td>
</tr>
<tr>
<td>Key Pair Name (KeyPairName)</td>
<td>Requires input</td>
<td>Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.</td>
</tr>
<tr>
<td>Restore Mode Password (RestoreModePassword)</td>
<td>Requires input</td>
<td>Password for a separate administrator account when the domain controller is in Restore Mode. This password must meet Microsoft’s default password complexity requirements.</td>
</tr>
<tr>
<td>Domain Admin Password (DomainAdminPassword)</td>
<td>Requires input</td>
<td>Password for the domain administrator user. This password must meet Microsoft’s default password complexity requirements.</td>
</tr>
<tr>
<td>Service Account Password (SQLServiceAccountPassword)</td>
<td>Requires input</td>
<td>Password for the SQL Service account. This password must meet Microsoft’s default password complexity requirements.</td>
</tr>
<tr>
<td>Installation Media ISO Image File URI (SPISOImageURI)</td>
<td>Requires input</td>
<td>S3 bucket URI that contains the ISO image file for the SharePoint Server 2016 installation media (e.g., s3://sample-bucket/microsoft/sharepoint/installation-media.img). You can also specify an HTTP/HTTPS URI (e.g., <a href="https://example.com/microsoft/sharepoint/installation-media.img">https://example.com/microsoft/sharepoint/installation-media.img</a>), but we recommend using an S3 bucket for optimal performance.</td>
</tr>
<tr>
<td>Product Key (SPKey)</td>
<td>trial key</td>
<td>The trial key for SharePoint Server 2016 is provided by default, but you can replace it with your own product key.</td>
</tr>
<tr>
<td>Farm Account Password (SPFarmAccountPassword)</td>
<td>Requires input</td>
<td>Password for the SharePoint farm account. This password must meet Microsoft’s default password complexity requirements.</td>
</tr>
</tbody>
</table>
Please make a note of these other parameters that have default values. You will need to edit or know these values in order to log in and manage the SharePoint farm.

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain DNS Name</td>
<td>example.com</td>
<td>Fully qualified domain name (FQDN) of the forest root domain.</td>
</tr>
<tr>
<td>Domain NetBIOS Name</td>
<td>example</td>
<td>The NetBIOS name (up to 15 characters) of the domain, for users of earlier versions of Windows.</td>
</tr>
<tr>
<td>Domain Admin User Name</td>
<td>StackAdmin</td>
<td>User name for the account that will be added as the domain administrator. This is separate from the default &quot;Administrator&quot; account.</td>
</tr>
<tr>
<td>Service Account Name</td>
<td>sqlsa</td>
<td>User name for the SQL Server service account. This account is a domain user.</td>
</tr>
</tbody>
</table>

For a complete list of template parameters and their descriptions, see Appendix B.

When you finish reviewing and customizing the parameters, choose Next.

**Note** You can also download the template and edit it to create your own parameters based on your specific deployment scenario.

4. On the Options page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you're done, choose Next.

5. On the Review page, review and confirm the template settings. Under Capabilities, select the check box to acknowledge that the template will create IAM resources.

6. Choose Create to deploy the stack.

7. Monitor the status of the stack. You will see that it spins off three separate sub-stacks. When the status of the “SharePoint Master” stack is CREATE_COMPLETE, the SharePoint environment is ready. The total process takes a little over three hours.
Step 4. Create Initial Content

In steps 4-6, 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 to validate our test.

After you have successfully launched the stack, remote into the environment through one of the 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 SPAPP1 server. Disable IE Enhanced Security Configuration or add http://spapp1 as a trusted site. Start Internet Explorer with administrative permissions (Run as administrator option) and navigate to SharePoint Central Administration (http://spapp1:18473/). If prompted, use the domain admin user name credentials.

2. Under Application Management, choose Manage web applications.
3. To create a new web application, choose **New**.

4. Set **Allow Anonymous** to **Yes** as shown in Figure 12, and then choose **OK**.
5. After the web application has been created, navigate back to SharePoint Central Administration and choose **Create site collections**.

6. 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 15. You can use the StackAdmin user account for this value. When you finish filling out the form, choose **OK**.
7. Now that you have created a blog, navigate to `http://spapp1`. 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, choose the gear icon, and then choose **Site settings**.
8. Under **Users and Permissions**, choose **Site permissions** to open the **Permissions** page. On the ribbon, choose **Anonymous Access**. In the **Anonymous Access** dialog box, choose **Entire Web site**, and then choose **OK**.

---

Figure 16: Modifying site settings for the blog
Step 5. Make the SharePoint Databases Highly Available

1. Establish an RDP session to the WSFCNODE1 instance. Start SQL Server Management Studio, and then choose **Connect** to connect to the local server.
2. 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, choose **Tasks**, and then choose **Back Up**. Keep the default settings, and then choose **OK** to perform the backup.

![Figure 19: Backing up a database](image)

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

![Figure 20: Naming the availability group](image)
4. Select the databases you previously backed up, and then choose **Next**.

![Select Databases](image1)

**Figure 21: Selecting availability group databases**

5. On the **Specify Replicas** page, add WSFCNODE2 as a replica. Make sure that the check boxes for automatic failover and synchronous replication are selected, as shown in Figure 22.

![Specify an Instance of SQL Server to host a secondary replica.](image2)

**Figure 22: Specifying replicas**

6. On the **Specify Replicas** page, choose 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 23, choose **Next**.
7. On the **Select Initial Data Synchronization** page, choose **Full** and enter `\\dc1\replica` as the network share to use for synchronizing the data. Choose **Next**.

8. Accept the default settings on the remaining pages of the wizard, and then choose **Next** and **Finish** to build the availability group. Make sure that the wizard completes successfully before moving on to the next step.
9. 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 26.

10. 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 27. 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 25: Successful completion of the AlwaysOn Availability Group wizard**

**Figure 26: Forcing AD replication from Availability Zone 1 to Availability Zone 2**
11. On the SPAPP1 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

$ dbs = Get-SPDatabase | ?{$_.MultiSubnetFailover -ne $true}

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

**Figure 27: Modifying the SQL alias**

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

**Figure 28: Enabling multi-subnet failover for the SharePoint databases**
13. Navigate back to SharePoint Central Administration and choose **System Settings** in the left column. Under **Farm Management**, choose **Configure alternate access mappings**.

14. Edit the public zone URLs for your blog site collection, as shown in Figure 31. For the purposes of this test, the internet zone URL should be the DNS name of the ELB load balancer you recorded in step 12. Remember that for production, you can have a CNAME record (such as sharepoint.example.com) that resolves to the ELB load balancer DNS name.
15. 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.

**Step 6. Test Automatic Failover**

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 SPAPP1 and SPAPP2. To verify that automatic failover is functional, forcibly stop WSFCNODE1 and SPAPP1 from the Amazon EC2 console. You can stop the instances simultaneously to perform this test, as shown in Figure 32.
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 SPAPP1 has gone offline and direct HTTP traffic to SPAPP2. You can revisit the site in your web browser to confirm that everything is still working.

**Troubleshooting**

When you deploy the Quick Start, if you encounter a CREATE_FAILED error instead of the CREATE_COMPLETE status code, we recommend that you re-launch the template with **Rollback on failure** set to No. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack’s state will be retained and the instance will be left running, so you can troubleshoot the issue.

**Important** When you set **Rollback on failure** to No, you’ll continue to incur AWS charges for this stack. Please make sure to delete the stack when you’ve finished troubleshooting.

The following table lists specific CREATE_FAILED error messages you might encounter while creating the stack in AWS CloudFormation.

<table>
<thead>
<tr>
<th>Error message</th>
<th>Possible cause</th>
<th>What to do</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API: ec2: RunInstances Not authorized for images:</strong> <strong>ami-ID</strong></td>
<td>The template is referencing an AMI that has expired.</td>
<td>We refresh AMIs on a regular basis, but our schedule isn’t always synchronized with AWS AMI updates. If you get this error message, notify us, and we’ll update the template with the new AMI ID. If you’d like to fix the template yourself, you can download it and update the Mappings section with the latest AMI ID for your region.</td>
</tr>
<tr>
<td><strong>We currently do not have sufficient instance-type capacity in the AZ you requested</strong></td>
<td>One of the instance types is currently not available.</td>
<td>Switch to an instance type that supports higher capacity, or complete the request form in the AWS Support Center to increase the Amazon EC2 limit for the instance type or region. Limit increases are tied to the region they were requested for.</td>
</tr>
<tr>
<td><strong>Instance ID did not stabilize</strong></td>
<td>You have exceeded your IOPS for the region.</td>
<td>Request a limit increase by completing the request form in the AWS Support Center.</td>
</tr>
<tr>
<td><strong>System Administrator password must contain at least 8 characters</strong></td>
<td>The master password contains $ or other special characters.</td>
<td>Check the password parameters before you re-launch the Quick Start. The passwords must be at least 8 characters, consisting of uppercase and lowercase letters and numbers. Follow the guidelines for complex passwords, and avoid using special characters such as @ or $.</td>
</tr>
</tbody>
</table>
If failure is signaled or a wait condition or resource signal times out, you should remote into the affected machine and launch Event Viewer. Under **Custom Views, Administrative Events** or under **Windows Logs, Application**, look for errors of source **AWSQuickStart**. These will indicate the failing script, line number, and exception that was reported.

For additional information, see [Troubleshooting AWS CloudFormation](https://aws.amazon.com/documentation/cloudformation/) on the AWS website.

### Additional Resources

**AWS services**

- AWS CloudFormation
  [https://aws.amazon.com/documentation/cloudformation/](https://aws.amazon.com/documentation/cloudformation/)
- Amazon EC2
- Amazon VPC
  [https://aws.amazon.com/documentation/vpc/](https://aws.amazon.com/documentation/vpc/)

**Microsoft SharePoint Server**

- Configure SQL Server 2012 AlwaysOn Availability Groups for SharePoint 2013
- Windows Server Failover Clustering and SQL Server AlwaysOn Availability Groups

**Deploying Microsoft software on AWS**

- Microsoft on AWS
  [https://aws.amazon.com/microsoft/](https://aws.amazon.com/microsoft/)
- Secure Microsoft applications on AWS
- Microsoft Licensing Mobility
  [https://aws.amazon.com/windows/mslicensemobility/](https://aws.amazon.com/windows/mslicensemobility/)
- MSDN on AWS
  [https://aws.amazon.com/windows/msdn/](https://aws.amazon.com/windows/msdn/)
• Windows and .NET Developer Center on AWS
  https://aws.amazon.com/net/

Quick Start reference deployments

• AWS Quick Start home page
  https://aws.amazon.com/quickstart/

• Microsoft Active Directory on AWS
  https://docs.aws.amazon.com/quickstart/latest/active-directory-ds/

• Microsoft Remote Desktop Gateway on AWS
  https://docs.aws.amazon.com/quickstart/latest/rd-gateway/

• Microsoft SQL Server with WSFC on AWS
  https://docs.aws.amazon.com/quickstart/latest/sql/
Appendix A: Server Role Architecture

Traditional Topology

When you build your SharePoint Server 2016 farm based on traditional topologies, you build your architecture with web servers, application servers, and database servers.

Note To build your SharePoint farm with the traditional topology, keep the Farm Topology parameter at its default setting (traditional) during deployment. For more information about customizing this parameter, see the Customize Your Topology section later in this appendix.

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

![Two-tier SharePoint farm in a traditional topology](image)

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

![Three-tier SharePoint farm in a traditional topology](image)

The following sections provide detailed descriptions of each tier in a SharePoint 2016 farm built with a traditional topology.

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 by using Elastic Load Balancing or another third-party load-balancing solution such as HA Proxy.
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.
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-end servers, 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 your databases on a single server. For larger-sized farms, you can spread your databases across multiple SQL Server instances or clusters of SQL Server instances. 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 use Microsoft License Mobility through Software Assurance to bring your own license into the cloud.

In the Quick Start 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 8 vCPUs, 60 GiB of memory, and 1 x 160 GiB of SSD instance storage. Additionally, we provide highly performant and durable storage in the form of Elastic Block Store (Amazon EBS) volumes.

Figure 37: Highly available SharePoint farm on 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 this Quick Start. There are a
The number of input parameters that enable you to control the instance type and other settings, and you can further customize the deployment to meet your specific needs. For details, take a look at the Quick Start for Microsoft WSFC and SQL Server AlwaysOn for SQL Server Enterprise on AWS.

For details on the traditional topologies and configuring services on SharePoint 2016, see the technical diagrams for SharePoint 2013 and Services on Server Install Worksheet for Traditional Topologies provided by Microsoft. (Although those links are for earlier versions of SharePoint, the information generally applies to SharePoint 2016 as well.)

**Streamlined Topology**

When building your SharePoint farm based on a streamlined topology, services and other components are distributed to maximize server resources. A streamlined architecture includes front-end servers, batch-processing servers, and database servers. Streamlined topologies introduce a new approach to farm design in SharePoint 2016. Using this type of topology allows you to scale out more easily, because the servers in the front-end and batch-processing tiers are dedicated to separate functions. When the time comes to scale out within a specific tier, you simply add an identically configured server in your environment. The following sections describe the tiers in a SharePoint 2016 farm built with a streamlined topology.

**Note** To build your SharePoint farm with a streamlined topology, set the Farm Topology parameter to streamlined during Quick Start deployment. For more information about customizing this parameter, see the Customize Your Topology section later in this appendix.

**Front-End Servers**

Components, services, and service 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 from database servers in a traditional topology. The database tier will still consist of SQL Server instances, and traditional guidance for deploying database servers remains the same.
Distributed Cache
The Distributed Cache service can run on front-end servers in small or medium environments with fewer than 10,000 users. For larger environments, Distributed Cache, which is a memory-intensive service, is typically placed on dedicated servers.

Request Management
The Request Management feature 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.

Search
As larger environments scale beyond two batch-processing servers, it is very common to place the Search role on a 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 instance (or instances) for maximum performance.

There are a number of ways to architect a large SharePoint farm using a streamlined topology. For additional details on streamlined topologies and configuring services on SharePoint 2016, see the technical diagrams for SharePoint 2013 and Services on Server Install Worksheet for Streamlined Topologies provided by Microsoft. (Although those links are for earlier versions of SharePoint, the information generally applies to SharePoint 2016 as well.)

Simple Example of a Streamlined Topology
Figure 39 shows a SharePoint 2016 architecture based on a streamlined topology running in the AWS Cloud. This architecture includes the tiers for front-end servers, batch-processing and search servers, and database servers. It also includes an additional SharePoint server in each Availability Zone dedicated to the Distributed Cache feature.
Whether you decide to use a traditional or a streamlined topology for SharePoint 2016, 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 Online Server**

The Microsoft Office Online Server allows users to view and, depending on the scenario, edit Office documents in SharePoint libraries by using a supported browser on various devices such as PCs, mobile devices, and tablets. Figure 40 shows an Office Online Server instance within the web server tier in each Availability Zone.
It is important to notice that the Office Online Server role is not installed on the SharePoint 2016 servers and must be deployed on separate servers in the environment. The Office Online Server can also be used by other enterprise products like Microsoft Exchange and Skype for Business for rendering Office documents through a browser.

The AWS CloudFormation template provided by this Quick Start allows you to choose whether to include Office Online Server in each Availability Zone in your environment. Figure 41 shows the template parameter that controls that setting. If you choose to include these servers, the Quick Start will prepare the instances for Office Online Server and handle the prerequisites. You’ll need to download, install, and configure Office Online Server manually.
If you've included Office Online Servers in your template launch, you will need to configure them to work with your SharePoint farm. For configuration steps, see Configure Office Online Server for SharePoint 2016 on the Microsoft TechNet site. You'll need to download and install the Office Online Server components from Microsoft.

**Intranet SharePoint Server Farm on AWS**

All the architecture diagrams shown up to this point represent an internet-facing Microsoft SharePoint farm. For this 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 42 shows a typical topology for an intranet SharePoint server farm running on the AWS Cloud.
Figure 42: Intranet SharePoint farm with hybrid architecture

As shown in Figure 42, we’ve added a virtual private gateway to the VPC. To enable internal network connectivity to the VPC, we’ve created a VPN tunnel from the customer gateway (an IPSec-capable device) to the virtual private gateway running in the VPC.

In addition, AWS offers the AWS 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 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 allows you to choose how to implement Elastic Load Balancing. This parameter, shown in Figure 43, is in the Amazon EC2 Configuration section. You can choose from two options: internal or external. The default setting is external.
If you are building an intranet-only farm, you can 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. NTLM authentication is not supported with Elastic Load Balancing using an HTTP listener at this time. There are also a number of third-party load-balancing solutions in the [AWS Marketplace](https://aws.amazon.com/marketplace/) that you can use 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 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.

The [Securing the Microsoft Platform on Amazon Web Services](https://aws.amazon.com/whitepapers/security/) whitepaper discusses the different methods for securing your AWS infrastructure in detail. 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 a 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 44 shows the default ACL configuration for a VPC subnet.

![Network ACL: Default (replace)](default-acl-config-table.png)

<table>
<thead>
<tr>
<th>Rule #</th>
<th>Port (Service)</th>
<th>Protocol</th>
<th>Source</th>
<th>Allow/Deny</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>ALL</td>
<td>ALL</td>
<td>0.0.0.0/0</td>
<td>ALLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0.0.0/0</td>
<td>DENY</td>
</tr>
</tbody>
</table>

You may choose to keep the default network ACL configuration or 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 use SharePoint Server to host a public website. In this scenario, you can add another 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 that are running in a private subnet.

Figure 45 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.

![Diagram of web application publishing with a reverse proxy server](image)

**Figure 45: 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 does not set up an environment for website publishing, but after the deployment, you may choose to add reverse proxy servers and configure the environment that’s illustrated in Figure 45.

**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 follow 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.

<table>
<thead>
<tr>
<th>Role</th>
<th>Processor</th>
<th>RAM</th>
<th>Boot volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web front-end server / front-end server</td>
<td>64-bit, 4 cores</td>
<td>12 GiB</td>
<td>80 GiB</td>
</tr>
<tr>
<td>Application server / batch processing / back end</td>
<td>64-bit, 4 cores</td>
<td>12 GiB</td>
<td>80 GiB</td>
</tr>
<tr>
<td>Database server (fewer than 1,000 users)</td>
<td>64-bit, 4 cores</td>
<td>8 GiB</td>
<td>80 GiB</td>
</tr>
<tr>
<td>Database server (between 1,000 and 10,000 users)</td>
<td>64-bit, 8 cores</td>
<td>16 GiB</td>
<td>80 GiB</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Role</th>
<th>EC2 instance type</th>
<th>Boot volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web front-end server / front-end server</td>
<td>c4.2xlarge (8 vCPU, 15 GiB memory)</td>
<td>120 GiB (EBS/GP2)</td>
</tr>
<tr>
<td>Application server / batch processing / back end</td>
<td>c3.2xlarge (8 vCPU, 15 GiB memory)</td>
<td>120 GiB (EBS/GP2)</td>
</tr>
<tr>
<td>Database server</td>
<td>r3.2xlarge (8 vCPU, 61 GiB memory)</td>
<td>120 GiB (EBS/GP2)</td>
</tr>
</tbody>
</table>

Amazon Elastic Block Store (Amazon EBS) volumes are used as the boot volume for each instance. Notice that we use the EBS General Purpose (gp2) volume type. This is an SSD-backed EBS volume that is used as the default boot volume type for all Amazon EC2 instances. These gp2 volumes provide a consistent baseline of 3 IOPS/GiB 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

When you launch the template, you can use the Farm Topology (SPTopology) parameter to define the SharePoint Server farm topology for the Quick Start deployment. This parameter provides two options: traditional (default) and streamlined.

![SPTopology Parameter](image)

**Figure 46: Choosing a farm topology**

- When you choose to build a traditional topology, the Quick Start sets up one application server and one web server in each Availability Zone.

  The traditional topology provides the minimum number of servers for high availability. The servers named SPWFE1 and SPWFE2 will receive HTTP requests from Elastic Load Balancing. The servers named SPAPP1 and SPAPP2 can provide application server or batch-processing functionality. This option is intended to be used to deploy a farm based on the traditional topology, but the servers can be specialized to also align with a streamlined topology.

- When you choose to build a streamlined topology, the Quick Start provides dedicated servers for additional functions in each Availability Zone.

  The streamlined topology provides enough infrastructure for a large farm. This option provides a set of instances in each Availability Zone for front-end server, Distributed Cache, batch-processing servers, and search servers. After deployment, you can modify the server roles to better accommodate your environment’s needs. For example, you can repurpose the search servers as specialized workload servers.

As your SharePoint servers are launched, the servers are renamed, joined to the Active Directory domain, and the SharePoint Server 2016 prerequisites are installed on each server. The farm is created after SharePoint is installed on the first server, and the remaining servers are installed and joined to the farm in the appropriate order.
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://spapp1:18473/) to configure your farm components, services, and service applications.

The default value for the Farm Topology (SPTopology) parameter is traditional. If you launch the AWS CloudFormation template and accept the default parameters, you will deploy the architecture illustrated in Figure 2.
Appendix B: AWS CloudFormation Template Parameters

The following tables provide a complete list of parameters provided in the AWS CloudFormation template for this Quick Start.

**Network Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key Pair Name</strong> (KeyPairName)</td>
<td><strong>Requires input</strong></td>
<td>Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your region.</td>
</tr>
<tr>
<td><strong>Availability Zones</strong> (AvailabilityZones)</td>
<td><strong>Requires input</strong></td>
<td>The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.</td>
</tr>
<tr>
<td><strong>Private Subnet 1 CIDR</strong> (PrivateSubnet1CIDR)</td>
<td>10.0.0.0/19</td>
<td>CIDR block for the Active Directory server tier located in Availability Zone 1.</td>
</tr>
<tr>
<td><strong>Private Subnet 2 CIDR</strong> (PrivateSubnet2CIDR)</td>
<td>10.0.64.0/19</td>
<td>CIDR block for the Active Directory server tier located in Availability Zone 2.</td>
</tr>
<tr>
<td><strong>Public Subnet 1 CIDR</strong> (PublicSubnet1CIDR)</td>
<td>10.0.32.0/20</td>
<td>CIDR block for the public (DMZ) subnet located in Availability Zone 1.</td>
</tr>
<tr>
<td><strong>Public Subnet 2 CIDR</strong> (PublicSubnet2CIDR)</td>
<td>10.0.96.0/20</td>
<td>CIDR block for the public (DMZ) subnet located in Availability Zone 2.</td>
</tr>
<tr>
<td><strong>VPC CIDR</strong> (VPCCIDR)</td>
<td>10.0.0.0/16</td>
<td>CIDR block for the VPC.</td>
</tr>
</tbody>
</table>

**Microsoft SharePoint Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ELB Configuration</strong> (ELBConfiguration)</td>
<td>external</td>
<td>How to configure the ELB load balancer. Options are external or internal. For more information, see the section on Intranet SharePoint Server farms in Appendix A.</td>
</tr>
<tr>
<td><strong>Include Office Online Servers</strong></td>
<td>false</td>
<td>Set to true to include an Office Online Server in each Availability Zone. For more information, see the section on Office Online Servers in Appendix A.</td>
</tr>
<tr>
<td><strong>Office Online Server Instance Type</strong> (OOSInstanceType)</td>
<td>m4.xlarge</td>
<td>Amazon EC2 instance type for the Office Online Server instances.</td>
</tr>
<tr>
<td><strong>Installation Media ISO Image File URI</strong> (SPISOImageURI)</td>
<td><strong>Requires input</strong></td>
<td>Amazon S3 URI to bucket that contains the ISO image file for the SharePoint Server 2016 installation media from step 2 of the deployment instructions (e.g., s3://sample-bucket/microsoft/sharepoint/installation-media.img). You</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
--- | --- | ---
**can also specify an HTTP/HTTPS URI (e.g., https://example.com/microsoft/sharepoint/installation-media.i** **mg), but we recommend using an S3 bucket for optimal performance.**

**Product Key** (SPKey) | **trial key** | The trial key for SharePoint Server 2016 is provided by default, but you can replace it with your own product key.

**Farm Topology** (SPTopology) | traditional | The topology for the SharePoint Server farm to be deployed. The two options are traditional and streamlined. For more information, see the section on customizing your topology in Appendix A.

**Farm Account Name** (SPFarmAccount) | spfarm | User name for the SharePoint Server farm account.

**Farm Account Password** (SPFarmAccountPassword) | **Requires input** | Password for the SharePoint farm account. This password must meet Microsoft's default password complexity requirements.

**SharePoint Server Instance Type** (SPInstanceType) | c4.2xlarge | Amazon EC2 instance type for the SharePoint web front-end servers.

---

### Microsoft Active Directory Configuration:

### Parameter label (name) | Default | Description
--- | --- | ---
**Domain Controller 1 Instance Type** (ADServer1InstanceType) | m4.xlarge | Amazon EC2 instance type for the first Active Directory instance.

**Domain Controller 1 NetBIOS Name** (ADServer1NetBIOSName) | DC1 | NetBIOS name of the first Active Directory server (up to 15 characters).

**Domain Controller 1 Private IP Address** (ADServer1PrivateIP) | 10.0.0.10 | Fixed private IP for the first Active Directory server located in Availability Zone 1.

**Domain Controller 2 Instance Type** (ADServer2InstanceType) | m4.xlarge | Amazon EC2 instance type for the second Active Directory instance.

**Domain Controller 2 NetBIOS Name** (ADServer2NetBIOSName) | DC2 | NetBIOS name of the second Active Directory server (up to 15 characters).

**Domain Controller 2 Private IP Address** (ADServer2PrivateIP) | 10.0.64.10 | Fixed private IP for the second Active Directory server located in Availability Zone 2.

**Domain Admin User Name** (DomainAdminUser) | StackAdmin | User name for the account that will be added as the domain administrator. This is separate from the default "Administrator" account.
<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Admin Password (DomainAdminPassword)</td>
<td>Requires input</td>
<td>Password for the domain administrator user. This password must meet Microsoft's default password complexity requirements.</td>
</tr>
<tr>
<td>Domain DNS Name (DomainDNSName)</td>
<td>example.com</td>
<td>Fully qualified domain name (FQDN) of the forest root domain.</td>
</tr>
<tr>
<td>Domain NetBIOS Name (DomainNetBIOSName)</td>
<td>example</td>
<td>The NetBIOS name (up to 15 characters) of the domain, for users of earlier versions of Windows.</td>
</tr>
<tr>
<td>Restore Mode Password (RestoreModePassword)</td>
<td>Requires input</td>
<td>Password for a separate administrator account when the domain controller is in Restore Mode. This password must meet Microsoft's default password complexity requirements.</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version (SQLServerVersion)</td>
<td>2014</td>
<td>The version of SQL Server to install on WSFC nodes. Supported versions are 2012 and 2014.</td>
</tr>
<tr>
<td>Service Account Name (SQLServiceAccount)</td>
<td>sqlsa</td>
<td>User name for the SQL Server service account. This account is a domain user.</td>
</tr>
<tr>
<td>Service Account Password (SQLServiceAccountPassword)</td>
<td>Requires input</td>
<td>Password for the SQL Server service account, which must meet Microsoft's default password complexity requirements.</td>
</tr>
<tr>
<td>Amazon-Provided SQL Server License (SQLLicenseProvided)</td>
<td>no</td>
<td>Set to yes to use the license-included SQL Server AMI from AWS. For more information about licensing options, see the Cost and Licenses section.</td>
</tr>
<tr>
<td>Data Volume Size (Volume1Size)</td>
<td>500</td>
<td>Volume size for the SQL Server data drive, in GiB.</td>
</tr>
<tr>
<td>Data Volume Type (Volume1Type)</td>
<td>gp2</td>
<td>Volume type (gp2 or io1) for the SQL Server data drive.</td>
</tr>
<tr>
<td>Data Volume IOPS (Volume1Iops)</td>
<td>1000</td>
<td>Provisioned IOPS for the SQL Server data drive. This setting applies only when the Data Volume Type parameter is set to io1.</td>
</tr>
<tr>
<td>Logs Volume Size (Volume2Size)</td>
<td>500</td>
<td>Volume size for the SQL Server logs drive, in GiB.</td>
</tr>
<tr>
<td>Logs Volume Type (Volume2Type)</td>
<td>gp2</td>
<td>Volume type (gp2 or io1) for the SQL Server logs drive.</td>
</tr>
<tr>
<td>Logs Volume IOPS (Volume2Iops)</td>
<td>1000</td>
<td>Provisioned IOPS for the SQL Server logs drive. This setting applies only when the Logs Volume Type parameter is set to io1.</td>
</tr>
<tr>
<td>TempDB Volume Size (Volume3Size)</td>
<td>500</td>
<td>Volume size for the SQL Server tempdb drive, in GiB.</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
--- | --- | ---
TempDB Volume Type (Volume3Type) | gp2 | Volume type (gp2 or io1) for the SQL Server tempdb drive.
TempDB Volume IOPS (Volume3Iops) | 1000 | Provisioned IOPS for the SQL Server tempdb drive. This setting applies only when the TempDB Volume Type parameter is set to io1.

### WSFC Configuration:

| Parameter label (name) | Default | Description
--- | --- | ---
WSFC Node 1 Instance Type (WSFCNode1InstanceType) | r4.xlarge | Amazon EC2 instance type for the first WSFC node.
WSFC Node 1 NetBIOS Name (WSFCNode1NetBIOSName) | WSFCNode1 | NetBIOS name of the first WSFC node (up to 15 characters).
WSFC Node 1 Private IP Address 1 (WSFCNode1PrivateIP1) | 10.0.0.100 | Primary private IP for the first WSFC node located in Availability Zone 1.
WSFC Node 1 Private IP Address 2 (WSFCNode1PrivateIP2) | 10.0.0.101 | Secondary private IP for the WSFC cluster on the first WSFC node.
WSFC Node 1 Private IP Address 3 (WSFCNode1PrivateIP3) | 10.0.0.102 | Third private IP for the Availability Group Listener on the first WSFC node.
WSFC Node 2 Instance Type (WSFCNode2InstanceType) | r4.xlarge | Amazon EC2 instance type for the second WSFC node.
WSFC Node 2 NetBIOS Name (WSFCNode2NetBIOSName) | WSFCNode2 | NetBIOS name of the second WSFC node (up to 15 characters).
WSFC Node 2 Private IP Address 1 (WSFCNode2PrivateIP1) | 10.0.64.100 | Primary private IP for the second WSFC node located in Availability Zone 1.
WSFC Node 2 Private IP Address 2 (WSFCNode2PrivateIP2) | 10.0.64.101 | Secondary private IP for the WSFC cluster on the second WSFC node.
WSFC Node 2 Private IP Address 3 (WSFCNode2PrivateIP3) | 10.0.64.102 | Third private IP for the Availability Group Listener on the second WSFC node.
Instance Type for Cluster Node 3 (WSFCNode3InstanceType) | r4.2xlarge | Amazon EC2 instance type for the third WSFC node.
<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cluster Node 3 NetBIOS Name</strong> (WSFCNode3NetBIOSName)</td>
<td>WSFCNode3</td>
<td>NetBIOS name of the third WSFC node (up to 15 characters).</td>
</tr>
<tr>
<td><strong>Cluster Node 3 Private IP Address 1</strong> (WSFCNode3PrivateIP1)</td>
<td>10.0.64.100</td>
<td>Primary private IP for the third WSFC node located in Availability Zone 1.</td>
</tr>
<tr>
<td><strong>Cluster Node 3 Private IP Address 2</strong> (WSFCNode3PrivateIP2)</td>
<td>10.0.64.101</td>
<td>Secondary private IP for the WSFC cluster on the third WSFC node.</td>
</tr>
<tr>
<td><strong>Cluster Node 3 Private IP Address 3</strong> (WSFCNode3PrivateIP3)</td>
<td>10.0.64.102</td>
<td>Third private IP for the Availability Group Listener on the third WSFC node.</td>
</tr>
</tbody>
</table>

**Microsoft Remote Desktop Gateway Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of RDGW hosts</strong> (NumberOfRDGWHosts)</td>
<td>1</td>
<td>The number of Remote Desktop Gateway instances to create. You can choose 1-4 instances.</td>
</tr>
<tr>
<td><strong>Allowed Remote Desktop Gateway External Access CIDR</strong> (RDGWCIDR)</td>
<td>Requires input</td>
<td>Allowed CIDR block for external access to the Remote Desktop Gateway instances. We recommend that you set this value to a trusted CIDR block.</td>
</tr>
<tr>
<td><strong>Remote Desktop Gateway Server Instance Type</strong> (RDGWInstanceType)</td>
<td>m4.xlarge</td>
<td>EC2 instance type for the Remote Desktop Gateway instance.</td>
</tr>
</tbody>
</table>

**AWS Quick Start Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quick Start S3 Bucket Name</strong> (QSS3BucketName)</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.</td>
</tr>
<tr>
<td><strong>Quick Start S3 Key Prefix</strong> (QSS3KeyPrefix)</td>
<td>quickstart-microsoft-sharepoint/</td>
<td>The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.</td>
</tr>
</tbody>
</table>
GitHub Repository

You can visit our GitHub repository to download the templates and scripts for this Quick Start, to post your feedback, and to share your customizations with others.

Document Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Changes</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2019</td>
<td>Streamlined the deployment instructions</td>
<td>Step 2</td>
</tr>
<tr>
<td>March 2018</td>
<td>Updated Active Directory to use the Windows Server 2016 AMI; added Quick Start configuration parameters</td>
<td>Appendix B</td>
</tr>
<tr>
<td>May 2016</td>
<td>Added:</td>
<td>Changes throughout templates and guide</td>
</tr>
<tr>
<td></td>
<td>- Support for SharePoint Server 2016</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Dedicated streamlined topology servers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Simpler installation media consumption process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Updated parameter names</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- New parameter labels and grouping</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Support for latest AD stack and NAT gateways</td>
<td></td>
</tr>
<tr>
<td>April 2015</td>
<td>Added information about testing high availability and automatic failover of SharePoint servers</td>
<td>Steps 4-6</td>
</tr>
<tr>
<td>March 2015</td>
<td>Optimized the underlying VPC design to support expansion and to reduce complexity</td>
<td>Architecture diagram and template updates</td>
</tr>
<tr>
<td>August 2014</td>
<td>Initial publication</td>
<td>–</td>
</tr>
</tbody>
</table>