

# Migrating a SharePoint application using the AWS Server Migration Service

by [Chris Munns](#) | on 15 MAY 2020 | in [Amazon EC2](#), [AWS Migration Hub](#), [AWS Server Migration Service](#) | [Permalink](#) |

[Share](#)

*This post is contributed by Ashwini Rudra, Solutions Architect; Rajesh Rathod, Sr. Product Manager; Vivek Chawda, Senior Software Engineer, EC2 Enterprise*

Many AWS customers are migrating on-premises SharePoint workloads to AWS for greater reliability, faster performance, and lower costs. While planning the migration, customers are looking for tools and methodologies that reduce the time to migrate, application downtime, and performance disruption. They use continuous replication to optimize cost and effort required to migrate applications reliably.

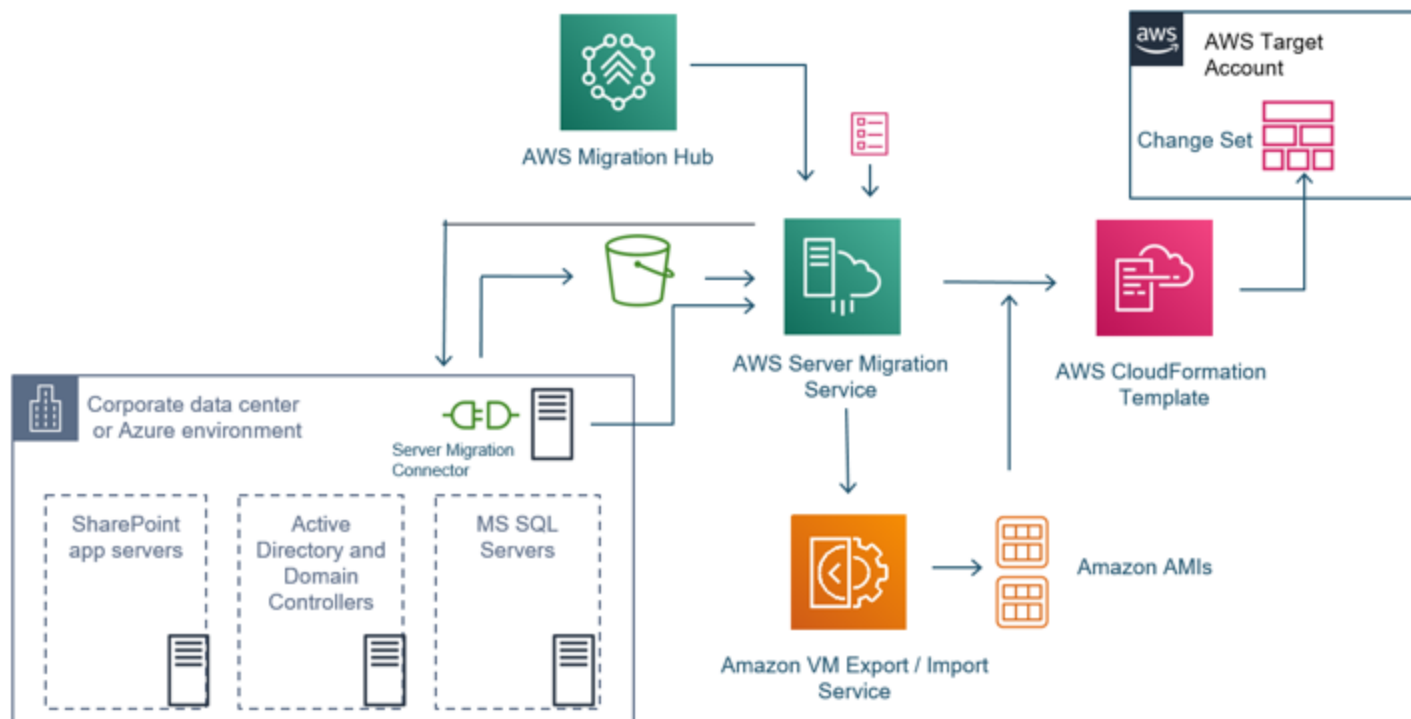
To accelerate these migrations, AWS provides a comprehensive set of tools. In this blog post, we explore how to use [AWS Server Migration Service](#) (AWS SMS) for migrating a SharePoint application from on-premises to AWS.

## Overview of solution

AWS Server Migration Service is an agentless service. It makes it easier and faster to migrate thousands of workloads from on-premises or Microsoft Azure to AWS. In this article, we discuss one of the approaches and steps to migrate SharePoint farm using AWS SMS.

AWS SMS also supports migrating a group of servers organized as an application. This can simplify migration of applications with complex dependencies that must be migrated together. This service provides a customized replication schedule designed to simplify migration at scale. This also tracks the progress of each migration using [AWS Migration Hub](#).

For SharePoint migrations, it facilitates migration failovers quickly. After the initial sync, the migration uses an incremental change capture approach to synchronize changes made to the on-premises SharePoint servers. This method also reduced required network bandwidth for the migration.



## SharePoint Migration Architecture

Here is how this service and solution works:

## Walkthrough

A basic SharePoint deployment is a 3-tier architecture comprising of web frontend servers, application servers, and backend SQL database servers. It also includes authentication services servers with Active Directory domain controllers.

To migrate this application, you must deploy a Server Migration Connector, which is a preconfigured virtual machine. This connector creates a server catalog. Based on selected server and configuration, it takes snapshots of virtual machines and stores them in S3 buckets.

In the background, AWS VM import/export service converts these snapshots into Amazon Machine Images (AMIs). Using these AMIs, you can configure launch settings where you define an order of application launch. You can also select instance types and set user-defined PowerShell scripts.

At the end, you define the cloud network topology: a VPC, subnets, and security groups. With these launch settings, SMS creates an [AWS CloudFormation](#) template, which can launch the SharePoint application in the target AWS account.

To migrate a SharePoint using SMS:

1. Install and register the SMS connector.
2. Create an application from SMS server catalog.

3. Configure replication settings for your SharePoint farm.
4. Configure launch settings.
5. Start the replication.
6. Launch the SharePoint application in AWS.

## Install and register the Server Migration Connector

The Server Migration Connector is a VM that you install in your on-premises virtualization environment. The supported platforms are VMware vSphere, Microsoft Hyper-V/SCVMM, and Microsoft Azure. Follow the links below to install in your environment:

- [Install Server Migration Connector on VMWare](#)
- [Install Server Migration Connector on Hyper-V](#)
- [Install Server Migration Connector on Azure](#)

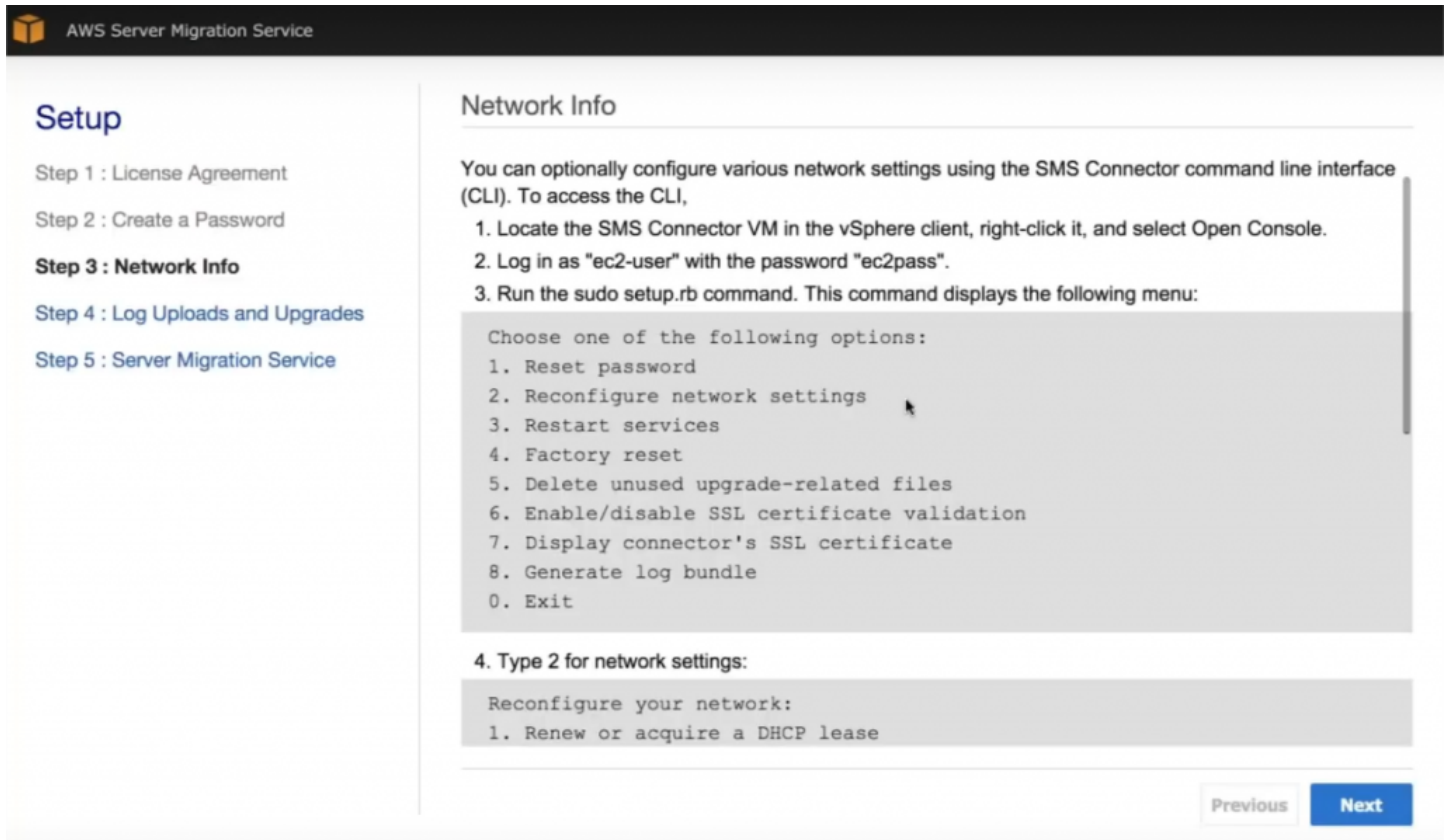
For example, VMware is a common scenario. First, you must set up Server Migration Connector and import server catalog:

1. Login to AWS Server Migration Service Console, and click to **Connectors**.
2. Download .ova file and deploy it to your VMWare environment using vSphere client from **AWS Server Migration Setup** page.

The screenshot shows the AWS Server Migration Connector setup page. At the top, there's a navigation bar with 'Services', 'Resource Groups', and a user profile 'Admin/arudra-Isengard @ ashw...'. The main heading is 'AWS Server Migration Connector setup'. Below it, the section 'Download the Server Migration Connector' provides instructions for different environments. For vCenter, it offers a 'Download OVA' button and a URL: 'https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector.ova'. It also provides integrity checksums (MD5 and SHA256) and links to download the files. For SCVMM/Hyper-V, it offers a 'Download VHD ZIP' button and a URL: 'https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector-for-SCVMM-HyperV.zip'. It also provides integrity checksums (MD5 and SHA256) and links to download the files. For Azure, it offers a 'Download PowerShell setup script' button and a link to 'or manually install following these instructions.' It also provides integrity checksums (MD5 and SHA256) and links to download the files.

3. Install OVA file on your VMWare environment. This is your AWS SMS Connector.

4. Use Connector Host IP address access Connector page and start five step registration process. Refer AWS documentation, [Install the Server Migration Connector on VMWare](#).
5. Follow the five-step process of registration. Here, you set up the password and network configuration between the connector virtual machine and AWS accounts.



**AWS Server Migration Service**

## Setup

- Step 1 : License Agreement
- Step 2 : Create a Password
- Step 3 : Network Info**
- Step 4 : Log Uploads and Upgrades
- Step 5 : Server Migration Service

### Network Info

You can optionally configure various network settings using the SMS Connector command line interface (CLI). To access the CLI,

1. Locate the SMS Connector VM in the vSphere client, right-click it, and select Open Console.
2. Log in as "ec2-user" with the password "ec2pass".
3. Run the `sudo setup.rb` command. This command displays the following menu:

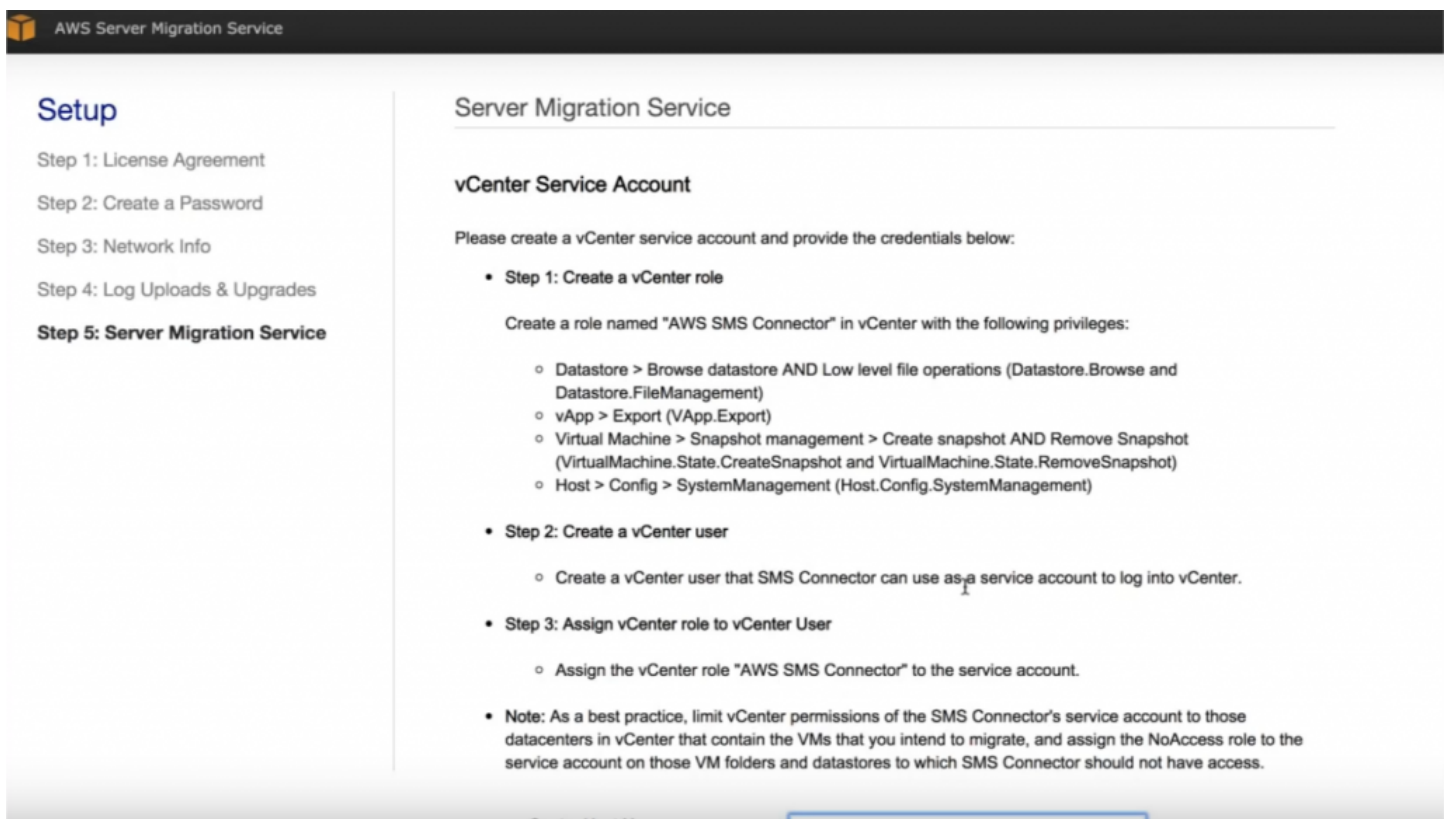
```
Choose one of the following options:
1. Reset password
2. Reconfigure network settings
3. Restart services
4. Factory reset
5. Delete unused upgrade-related files
6. Enable/disable SSL certificate validation
7. Display connector's SSL certificate
8. Generate log bundle
0. Exit
```

4. Type 2 for network settings:

```
Reconfigure your network:
1. Renew or acquire a DHCP lease
```

[Previous](#) [Next](#)

### Setup network info



**AWS Server Migration Service**

## Setup

- Step 1: License Agreement
- Step 2: Create a Password
- Step 3: Network Info
- Step 4: Log Uploads & Upgrades
- Step 5: Server Migration Service**

### Server Migration Service

#### vCenter Service Account

Please create a vCenter service account and provide the credentials below:

- **Step 1: Create a vCenter role**

Create a role named "AWS SMS Connector" in vCenter with the following privileges:

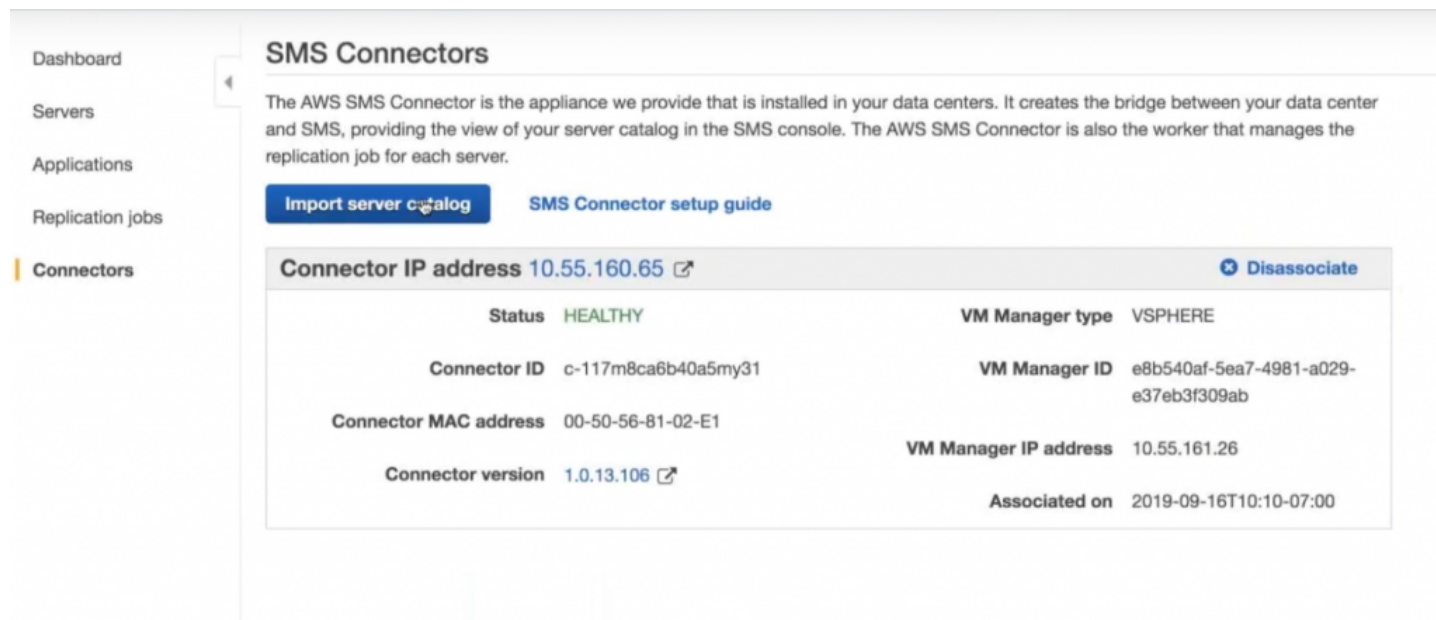
  - Datastore > Browse datastore AND Low level file operations (Datastore.Browse and Datastore.FileManagement)
  - vApp > Export (VApp.Export)
  - Virtual Machine > Snapshot management > Create snapshot AND Remove Snapshot (VirtualMachine.State.CreateSnapshot and VirtualMachine.State.RemoveSnapshot)
  - Host > Config > SystemManagement (Host.Config.SystemManagement)
- **Step 2: Create a vCenter user**
  - Create a vCenter user that SMS Connector can use as a service account to log into vCenter.
- **Step 3: Assign vCenter role to vCenter User**
  - Assign the vCenter role "AWS SMS Connector" to the service account.
- **Note:** As a best practice, limit vCenter permissions of the SMS Connector's service account to those datacenters in vCenter that contain the VMs that you intend to migrate, and assign the NoAccess role to the service account on those VM folders and datastores to which SMS Connector should not have access.

vCenter Host Name:

vCenter credentials

At the end, provide your vCenter host name and credentials.

In setup, provide all details of the VMWare and AWS environment to the connector. After establishing the connection, the connector setup looks like this in the AWS Management Console:



The screenshot shows the AWS Management Console interface for SMS Connectors. On the left is a navigation menu with options: Dashboard, Servers, Applications, Replication jobs, and Connectors (highlighted). The main content area is titled 'SMS Connectors' and includes a description: 'The AWS SMS Connector is the appliance we provide that is installed in your data centers. It creates the bridge between your data center and SMS, providing the view of your server catalog in the SMS console. The AWS SMS Connector is also the worker that manages the replication job for each server.' Below the description are two buttons: 'Import server catalog' and 'SMS Connector setup guide'. A summary card for a specific connector shows the following details:

Connector IP address	10.55.160.65	Disassociate	
Status	HEALTHY	VM Manager type	VSPHERE
Connector ID	c-117m8ca6b40a5my31	VM Manager ID	e8b540af-5ea7-4981-a029-e37eb3f309ab
Connector MAC address	00-50-56-81-02-E1	VM Manager IP address	10.55.161.26
Connector version	1.0.13.106	Associated on	2019-09-16T10:10-07:00

SMS Connectors

## Create an application from SMS server catalog

After configuring the connector and selecting “Import Server Catalog”, you are able to view the server catalog in AWS Server Migration Service console. To migrate your SharePoint application, select the application server, SQL Server, and Active Directory server from the server catalog.

Depending on the application architecture, you can group these servers to apply server-specific configuration settings and select appropriate instance types. Here are the steps:

1. Navigate to the application feature of SMS and create a “new application” for the SharePoint farm. Provide the application name, description, and IAM role.

## Create new application

**Step 1: Application settings**  
Step 2: Select servers  
Step 3: Add servers to groups  
Step 4: Add tags  
Step 5: Review

### Application settings

**Application name\***  ⓘ  
Up to 255 characters

**Application description**  ⓘ  
Up to 1024 characters

**Role name\***  ⓘ

Create new application – Application settings

- Select servers to migrate from the catalog.
- Create different groups for these servers in your console. For this, select servers and choose **Add servers to group**. This helps in defining different instance types and run user-defined PowerShell scripts for all servers in a group during application launch. You may create different groups for the application, web frontend, database, and Active Directory servers. In the below example, there are two groups – one for application servers, and the other for database servers. This process assumes that you have the authentication services servers already in place and operational in AWS. For more information on SharePoint authentication services, Active Directory and Domain Services, Refer Active Directory Domain Services on [AWS Deployment Scenario and Architecture](#).

## Create new application

Step 1: Application settings  
Step 2: Select servers  
**Step 3: Add servers to groups**  
Step 4: Add tags  
Step 5: Review

### Add servers to groups

You can optionally define custom groups for your servers. You can add up to 10 custom groups in an application. Organizing servers into groups (for example, AppTier, DBTier) can help in managing your migration. If no groups are specified, your servers will be grouped into 'Default'.

⊕ Add servers to group | ⊖ Delete group

	VM name ▾	Server ID ▾	VM ID ▾	VM manager ID ▾	Group ▾
<input type="checkbox"/>	sharepoint-appserver	s-dcbc59b5	vm-93	e8b540af-5ea7-4981-a...	App
<input type="checkbox"/>	sharepoint-dbserver	s-ddbc59b4	vm-92	e8b540af-5ea7-4981-a...	DB

Create new application – Add servers to groups

- Add tags, per your organization tagging strategy or policy.
- Review your application and click “Next” when it is ready for replication settings configurations.

## Configure replication settings



1. Define “replication job type”, “when to start replication job”, and “automatic AMI deletion” based on your requirements. Choose **Next**.
2. On the **Configure server-specific settings** page, in the **License type** column, select the license type for AMIs created from the replication job. Windows Servers can use either an AWS-provided license or Bring Your Own License (BYOL). Check [Microsoft Licensing](#) to review the licensing options. You can also choose Auto to allow AWS SMS to detect the source-system operating system (OS) and apply the appropriate license to the migrated virtual machine. Choose **Next**.

Configure replication settings - sharepoint-demo-app

Step 1: Replication settings

**Step 2: Server-specific settings**

Step 3: Review

Server-specific settings

Provide replication configuration for servers. You can use 'Edit Multiple Servers' to apply the same settings to all servers in a group. For example, apply the same 'Quiesce' to all servers.

[Edit multiple servers](#) [Collapse all](#)

▼ DB Group total: 1 server

VM name ▼	License type	Quiesce	Server ID ▼
sharepoint-dbserver	AUTO ▼	<input type="checkbox"/>	s-ddbc59b4

▼ App Group total: 1 server

VM name ▼	License type	Quiesce	Server ID ▼
sharepoint-appserver	AUTO ▼	<input type="checkbox"/>	s-dcbc59b5

[Cancel](#) [Previous](#) [Next](#)

### Server-specific settings

3. Review your application replication setting and choose **Configure Launch Settings**.

## Configure launch settings

An important aspect of migration is how this application should launch on EC2. This is configured on this page of SMS:

1. On the **Configure launch settings** page, for the **IAM CloudFormation role**, provide an IAM role for launch settings. Refer AWS Documentation on [IAM Roles for AWS SMS](#).

## Configure launch settings - sharepoint-demo-app

## Step 1: Configure launch settings

Step 2: Configure target instance

Step 3: Configure target network

Step 4: Review

## Configure launch settings

IAM CloudFormation role sms-launch ⓘ

## Specify launch order

Launch order allows servers in a group to be launched prior to servers in other groups. Groups with a lower launch order will be launched first.

Group name	Launch order
DB	1 ▼
App	1 ▼

## Configure launch settings

2. Under **Specify launch order**, configure a launch order for your groups. For this SharePoint application, you may prefer Active Directory first, followed by the SQL database, and then the application servers.

3. Under **Configure launch settings** for the application, edit the server settings individually:

## Configure launch settings - sharepoint-demo-app

Step 1: Configure launch settings

## Step 2: Configure target instance

Step 3: Configure target network

Step 4: Review

## Configure target instance details

Provide instance configuration for servers. You can use 'Edit Multiple Servers' to apply the same settings to all servers in a group. For example, apply the same 'Instance Type' to all servers.

Edit multiple servers

Collapse all

▼ DB

VM name ▼	Logical ID*	Instance type*	Key pair*
sharepoint-dbserver	server1	m5.2xlarge × ▼	vivekcNewKey × ▼

▼ App

VM name ▼	Logical ID*	Instance type*	Key pair*
sharepoint-appserver	server2	m5.2xlarge × ▼	vivekcNewKey × ▼

\*Required

Cancel

Previous

Next

## Target instance details

- Logical ID: AWS CloudFormation resource ID. This is the logical ID of the CloudFormation template that AWS SMS generates for the application. A value is created automatically when you use the console, but you must supply it manually when using the API, CLI, or SDKs. For more information, see [Resources](#) in the AWS CloudFormation User Guide.
- Instance type: specifies the EC2 instance type on which to launch the server.
- Key pair: specifies the SSH key pair for access to the server.
- Configuration script: a script to run configuration commands at the startup of EC2 instances launched as part of an application. This is important for your SharePoint migration, as you can provide registry settings and configuration settings using PowerShell script for your SharePoint servers and SQL database servers.

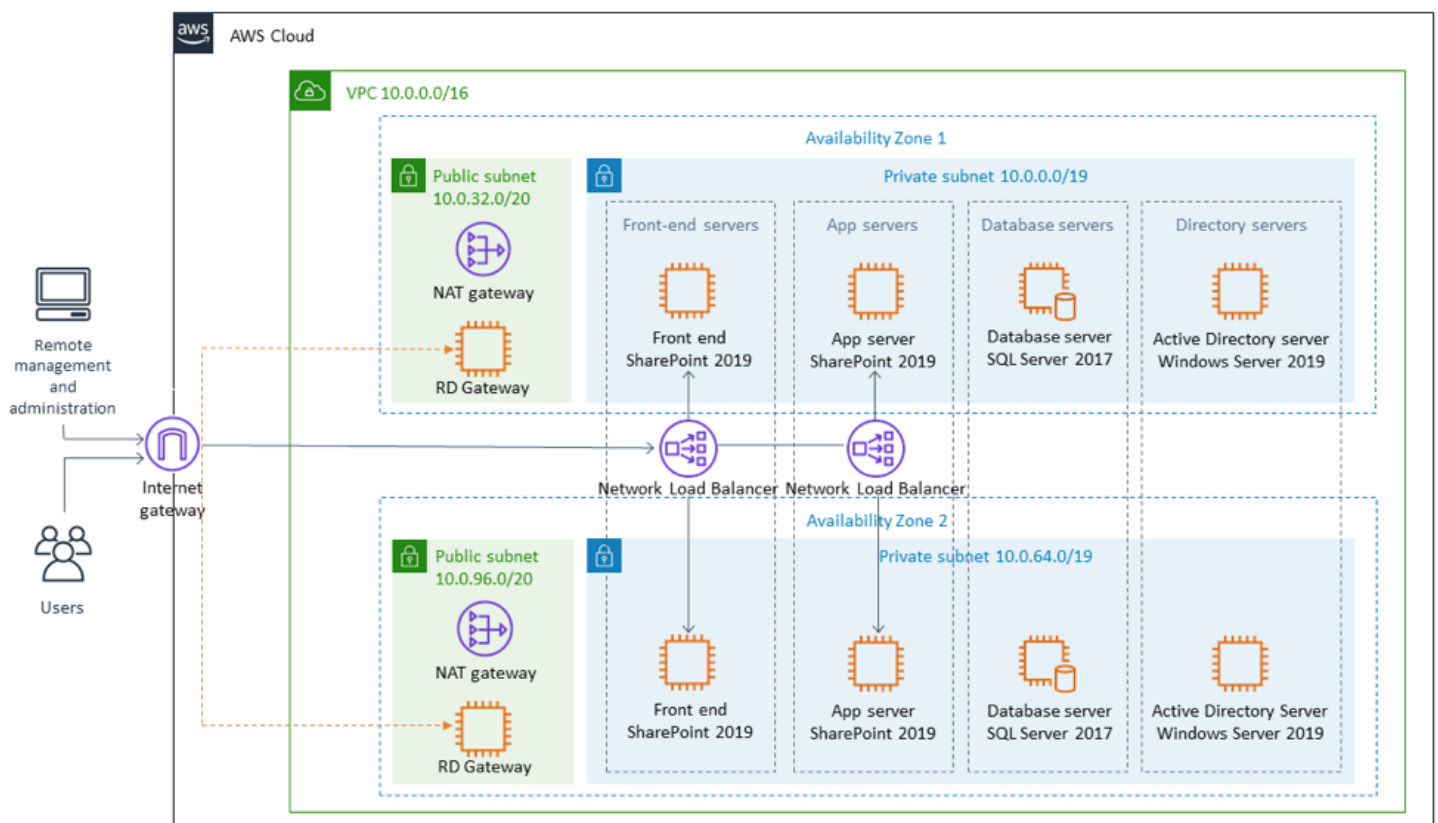


For example, the PowerShell script below retrieves the IP address of the current SQL Server and replaces the old SQL Server IP in the SharePoint configuration database and connection strings. You can automate many SharePoint configuration tasks using PowerShell scripts.

#### PowerShell

```
Start-Transcript -Path "C:\UserData.log" -Append
$oldIP = <<Your old IP goes here>>
$newIP = ([System.Net.Dns]::GetHostAddresses("sp-ip-sql-server.aws.local")).IPAddressToString
$registryPath = "HKLM:\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\16.0\Secure\
$Name = "dsn"
$regValue = (Get-ItemProperty -Path $registryPath -Name $Name).dsn
$updatedRegValue = $regValue.Replace($oldIP, $newIP)
New-ItemProperty -Path $registryPath -Name $Name -Value $updatedRegValue -PropertyType String
```

4. Configure the target network: VPC, subnets and security groups. Refer to the SharePoint on AWS documentation ([Reference Deployment](#)) for more guidance. The network topology varies based on platform requirements. Here is a reference architecture for SharePoint on AWS:



Architecture topology

## Start application replication

To start replication, select **Start Replication** under the **Actions** menu on the Applications page. The replication time depends on the amount of data replicated and available network bandwidth. On the application details page, you

can observe the status of the replication in the **Replication status** field. If the replication fails, the status message field shows the reason.

The screenshot shows the AWS Server Migration Service console for an application named 'sharepoint-demo-app'. The left sidebar contains navigation links: Dashboard, Servers, Applications (selected), Replication jobs, and Connectors. The main content area displays application details under the 'Replication' tab. The details include:

- Application status:** Active
- Status message:** -
- # of groups:** 2
- # of servers:** 2
- IAM role:** sms
- Creation time:** 2019-09-16T10:16-07:00
- Last modified:** 2019-09-16T10:16-07:00
- Description:** -
- Latest replication time:** -
- Replication status:** Ready for replication
- Replication status message:** -
- Latest launch time:** -
- Launch status:** Ready for launch
- Launch status message:** -
- CloudFormation stack:** -

An 'Actions' dropdown menu is open, showing options: Configure replication settings, Configure launch settings, Start replication (highlighted), Stop replication, Launch application, Terminate application, Generate template, and Generate changeset.

Start replication

## Launch SharePoint in AWS

1. On the **Application** page, choose **Actions, Launch application**. A replication job must complete before you perform this action.
2. In the *Launch application window*, choose **Launch**. On the application details page, you can observe the status of the launch in the Launch status field. If the launch fails, you are able to find the reason in the status message field. You can also generate a CloudFormation template and download this template to use in different AWS accounts.

The screenshot shows the AWS Server Migration Service console for an application named 'demo-sharepoint-sp'. The left sidebar contains navigation links: Dashboard, Servers, Applications (selected), Replication jobs, and Connectors. The main content area displays application details under the 'Launch' tab. The details include:

- Application status:** Active
- Status message:** -
- # of groups:** 2
- # of servers:** 2
- IAM role:** sms
- Creation time:** 2019-09-09T18:40-07:00
- Last modified:** 2019-09-09T18:40-07:00
- Description:** -
- Latest replication time:** 2019-09-09T20:44-07:00
- Replication status:** Replicated
- Replication status message:** -
- Latest launch time:** -
- Launch status:** Terminated
- Launch status message:** -
- CloudFormation stack:** -

An 'Actions' dropdown menu is open, showing options: Configure replication settings, Configure launch settings, Start replication, Stop replication, Launch application (highlighted), Terminate application, Generate template, and Generate changeset.

Below the application details, there is a section for 'Last Updated Sep 16th, 10:21am' with a refresh icon. It shows two groups: 'DB' and 'App'. The 'DB' group contains one server with the following details:

VM name	Replication job ID	Latest AMI ID	Job state	Progress	Status message	Server ID	VM ID
vivekc-sp-ip-sql-server	sms-job-78f11411	ami-0c028fb7859...	Active	Preparing		s-dca742b5	vm-102

The 'App' group also contains one server, but its details are partially obscured.

Launch application

## Test your migration

When the SharePoint application is launched, you can connect to Amazon EC2 instances based servers via Remote Desktop Protocol (RDP). You can access the application based on your Internet Information Services (IIS) Server settings runs on SharePoint Web Front End (WFE) application server (on Amazon EC2). It is also recommended to investigate optimizations using the AWS Compute Optimizer. For this blog, we have not verified the migration steps with SharePoint 2007 and 2010.

## Conclusion

AWS Server Migration Service simplifies SharePoint application migration. Using AWS SMS, you can easily migrate a SharePoint farm and reduce your migration timeline using the launch setting and launch order features.

To learn more, watch [Application migration Using AWS Server Migration Service \(SMS\)](#) or view a demo on the [AWS Online Tech Talks Channel](#). If you have feedback, let us know in the comments section below.

TAGS: [Amazon EC2 VM Import/Export](#), [AWS Server Migration Service](#), [AWSMigration Hub](#), [SharePoint](#)