AWS Server Migration Service

User Guide

AWS Server Migration Service: User Guide

The AWS Documentation website is getting a new look!

Try it now and let us know what you think. Switch to the new look >>

You can return to the original look by selecting English in the language selector above.

Table of Contents

What Is AWS SMS?

AWS Server Migration Service automates the migration of your on-premises VMware vSphere, Microsoft Hyper-V/SCVMM, and Azure virtual machines to the AWS Cloud. AWS SMS incrementally replicates your server VMs as cloud-hosted Amazon Machine Images (AMIs) ready for deployment on Amazon EC2. Working with AMIs, you can easily test and update your cloud-based images before deploying them in production.

By using AWS SMS to manage your server migrations, you can:

- **Simplify the cloud migration process.** You can begin migrating a group of servers with just a few clicks in the AWS Management Console. After the migration has initiated, AWS SMS manages all the complexities of the migration process, including automatically replicating volumes of live servers to AWS and creating new AMIs periodically. You can quickly launch EC2 instances from AMIs in the console.
- **Orchestrate multi-server migrations.** AWS SMS orchestrates server migrations by allowing you to schedule replications and track progress of a group of servers that constitutes an *application*. You can schedule initial replications, configure replication intervals, and track progress for each server using the console. When you launch a migrated application, you can apply customized configuration scripts that run during startup.
- **Test server migrations incrementally:** With support for incremental replication, AWS SMS allows fast, scalable testing of migrated servers. Because AWS SMS replicates incremental changes to your on-premises servers and transfers only the delta to the cloud, you can test small changes iteratively and save on network bandwidth.
- **Support the most widely used operating systems.** AWS SMS supports the replication of operating system images containing Windows, as well as several major Linux distributions.
- **Minimize downtime.** Incremental AWS SMS replication minimizes the business impact associated with application downtime during the final cutover.

Use of AWS SMS is limited as follows:

- 50 concurrent VM migrations per account, unless a customer requests a limit increase.
- 90 days of service usage per VM (not per account), beginning with the initial replication of a VM. We terminate an ongoing replication after 90 days unless a customer requests a limit increase.
- 50 concurrent application migrations per account, with a limit of 10 groups and 50 servers in each application.

Pricing

There is no additional fee to use Server Migration Service. You pay the standard fees for the S3 buckets, EBS volumes, and data transfer used during the migration process, and for the EC2 instances that you run.

Server Migration Service (SMS) Requirements

Your VMware vSphere, Microsoft Hyper-V/SCVMM, or Microsoft Azure environment must meet the following requirements for you to use the Server Migration Service to migrate your on-premises virtualized servers to Amazon EC2.

Requirements

General Requirements

Before setting up AWS SMS, take action as needed to meet all of the following requirements.

All VMs

- Disable any antivirus or intrusion detection software on the VM you are migrating. These services can be re-enabled after the migration process is complete.
- Disconnect any CD-ROM drives (virtual or physical) connected to the VM.

Windows VMs

- Enable Remote Desktop (RDP) for remote access.
- Install the appropriate version of .NET Framework on the VM. Note that .NET Framework 4.5 or later will be installed automatically on your VM if required.

Windows Version	.NET Framework Version
Windows Server 2008 or earlier	3.5 or later
Windows Server 2008 R2 or later	4.5 or later
Windows 8 or earlier	3.5 or later
Windows 8.1 or later	4.5 or later

- When preparing a Microsoft Windows VM for migration, configure a fixed pagefile size and ensure that at least 6 GiB of free space is available on the root volume. This is necessary for successful installation of the drivers.
- Make sure that your host firewall (such as Windows firewall) allows access to RDP. Otherwise, you will not be able to access your instance after the migration is complete.

- Apply the following hotfixes:
  - You cannot change system time if `RealTimeIsUniversal` registry entry is enabled in Windows
  - High CPU usage during DST changeover in Windows Server 2008, Windows 7, or Windows Server 2008 R2

Linux VMs

- Enable Secure Shell (SSH) for remote access.
- Make sure that your host firewall (such as iptables) allows access to SSH. Otherwise, you will not be able to access your instance after the migration is complete.
- Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
- Make sure that the root volume of your Linux VM uses one of the following file systems:
  - EXT2
  - EXT3
  - EXT4
  - Btrfs
  - JFS
  - XFS

Programmatic Modifications to VMs

When importing a VM, AWS modifies the file system to make the imported VM accessible to the customer. The following actions may occur:

- [Linux] Installing Citrix PV drivers either directly in OS or modify initrd/initramfs to contain them.
- [Linux] Modifying network scripts to replace static IP addresses with dynamic IP addresses.
- [Linux] Modifying `/etc/fstab`, commenting out invalid entries and replacing device names with UUIDs. If no matching UUID can be found for a device, the `nofail` option is added to the device description. You will need to correct the device naming and remove `nofail` after import. As a best practice when preparing your VMs for import, we recommend that you specify your VM disk devices by UUID rather than device name.

  Entries in `/etc/fstab` that contain distributed file system types (nfs, cifs, smbfs, vboxsf, sshfs, etc.) will be disabled.
- [Linux] Modifying grub bootloader settings such as the default entry and timeout.
- [Windows] Modifying registry settings to make the VM bootable.

When writing a modified file, AWS retains the original file at the same location under a new name.

Operating Systems Supported by AWS SMS

The following operating systems can be migrated to EC2 using SMS:

Windows (32- and 64-bit)

- Microsoft Windows Server 2003 (Standard, Datacenter, Enterprise) with Service Pack 1 (SP1) or later (32- and 64-bit)
- Microsoft Windows Server 2003 R2 (Standard, Datacenter, Enterprise) (32- and 64-bit)
- Microsoft Windows Server 2008 (Standard, Datacenter, Enterprise) (32- and 64-bit)

- Microsoft Windows Server 2008 R2 (Standard, Web Server, Datacenter, Enterprise) (64-bit only)
- Microsoft Windows Server 2012 (Standard, Datacenter) (64-bit only)
- Microsoft Windows Server 2012 R2 (Standard, Datacenter) (64-bit only) (Nano Server installation not supported)
- Microsoft Windows Server 2016 (Standard, Datacenter) (64-bit only)
- Microsoft Windows Server 1709 (Standard, Datacenter) (64-bit only)
- Microsoft Windows Server 1803 (Standard, Datacenter) (64-bit only)
- Microsoft Windows Server 2019 (Standard, Datacenter) (64-bit only)
- Microsoft Windows 7 (Home, Professional, Enterprise, Ultimate) (US English) (32- and 64-bit)
- Microsoft Windows 8 (Home, Professional, Enterprise) (US English) (32- and 64-bit)
- Microsoft Windows 8.1 (Professional, Enterprise) (US English) (64-bit only)
- Microsoft Windows 10 (Home, Professional, Enterprise, Education) (US English) (64-bit only)

Linux/Unix (64-bit)

- Ubuntu 12.04, 12.10, 13.04, 13.10, 14.04, 14.10, 15.04, 16.04, 16.10, 17.04, 18.04
- Red Hat Enterprise Linux (RHEL) 5.1-5.11, 6.1-6.9, 7.0-7.6 (6.0 lacks required drivers)
- SUSE Linux Enterprise Server 11 with Service Pack 1 and kernel 2.6.32.12-0.7
- SUSE Linux Enterprise Server 11 with Service Pack 2 and kernel 3.0.13-0.27
- SUSE Linux Enterprise Server 11 with Service Pack 3 and kernel 3.0.76-0.11, 3.0.101-0.8, or 3.0.101-0.15
- SUSE Linux Enterprise Server 11 with Service Pack 4 and kernel 3.0.101-63
- SUSE Linux Enterprise Server 12 with kernel 3.12.28-4
- SUSE Linux Enterprise Server 12 with Service Pack 1 and kernel 3.12.49-11
- SUSE Linux Enterprise Server 12 with Service Pack 2 and kernel 4.4
- SUSE Linux Enterprise Server 12 with Service Pack 3 and kernel 4.4
- CentOS 5.1-5.11, 6.1-6.6, 7.0-7.6 (6.0 lacks required drivers)
- Debian 6.0.0-6.0.8, 7.0.0-7.8.0, 8.0.0
- Oracle Linux 5.10-5.11 with el5uek kernel suffix
- Oracle Linux 6.1-6.10 using RHEL-compatible kernel 2.6.32 or UEK kernels 3.8.13, 4.1.12
- Oracle Linux 7.0-7.6 using RHEL compatible kernel 3.10.0 or UEK kernels 3.8.13, 4.1.12, 4.14.35
- Fedora Server 19-21

Volume Types and File Systems Supported by AWS SMS

AWS Server Migration Service supports migrating Windows and Linux instances with the following file systems:

Operating System	File System	Architecture	Partition Table	Data Volumes Supported	Boot Volumes Supported
Windows	NTFS	32-bit	MBR	✓	✓
			GPT	✓	

Operating System	File System	Architecture	Partition Table	Data Volumes Supported	Boot Volumes Supported
		64-bit	MBR	✓	✓
			GPT	✓	✓ (VHDX only)
	ReFS	32-bit	MBR		
			GPT		
		64-bit	MBR	✓	
			GPT	✓	
Linux/Unix	ext2, ext3, ext4, Btrfs, JFS, XFS	64-bit	MBR	✓	✓
			GPT	✓	

AMIs with volumes using EBS encryption are not supported. When migrating servers using AWS SMS, do not turn on encryption by default. If encryption by default is already on and you are experiencing delta replication failures, turn off this feature.

Licensing Options

When you create a new replication job, the AWS Server Migration Service console provides a **License type** option. The possible values include:

- **Auto** (default)

  Detects the source-system operating system (OS) and applies the appropriate license to the migrated virtual machine (VM).
- **AWS**

  Replaces the source-system license with an AWS license, if appropriate, on the migrated VM.
- **BYOL**

  Retains the source-system license, if appropriate, on the migrated VM.

> **Note**
> If you choose a license type that is incompatible with your VM, the replication job fails with an error message. For more information, see the OS-specific information below.

The same licensing options are available through the AWS SMS API and CLI. For example:

```
aws sms create-replication-job --license-type <value>
```

The value of the `--license-type` parameter can be AWS or BYOL. Leaving it unset is the same as choosing **Auto** in the console.

Licensing for Linux

Linux operating systems support only BYOL licenses. Choosing **Auto** (the default) means that AWS SMS uses a BYOL license.

Migrated Red Hat Enterprise Linux (RHEL) VMs must use Cloud Access (BYOL) licenses. For more information, see Red Hat Cloud Access on the Red Hat website.

Migrated SUSE Linux Enterprise Server VMs must use SUSE Public Cloud Program (BYOS) licenses. For more information, see SUSE Public Cloud Program—Bring Your Own Subscription.

Licensing for Windows

Windows server operating systems support either BYOL or AWS licenses. Windows client operating systems (such as Windows 10) support only BYOL licenses.

If you choose **Auto** (the default), AWS SMS uses the AWS license if the VM has a server OS. Otherwise, the BYOL license is used.

The following rules apply when you use your BYOL Microsoft license, either through MSDN or Windows Software Assurance Per User:

- Your BYOL instances are priced at the prevailing Amazon EC2 Linux instance pricing, provided that you meet the following conditions:
  - Run on a Dedicated Host (Dedicated Hosts)
  - Launch from VMs sourced from software binaries provided by you using AWS SMS, which are subject to the current terms and abilities of AWS SMS
  - Designate the instances as BYOL instances
  - Run the instances within your designated AWS regions, and where AWS offers the BYOL model
  - Activate using Microsoft keys that you provide or which are used in your key management system
- You must account for the fact that when you start an Amazon EC2 instance, it can run on any one of many servers within an Availability Zone. This means that each time you start an Amazon EC2 instance (including a stop/start), it may run on a different server within an Availability Zone. You must account for this fact in light of the limitations on license reassignment, as described in the Microsoft Volume Licensing Product Terms available at Licensing Terms, or consult your specific use rights to determine if your rights are consistent with this usage.
- You must be eligible to use the BYOL program for the applicable Microsoft software under your agreements with Microsoft, for example, under your MSDN user rights or under your Windows Software Assurance Per User Rights. You are solely responsible for obtaining all required licenses and for complying with all applicable Microsoft licensing requirements, including the PUR/PT. Further, you must have accepted Microsoft's End User License Agreement (Microsoft EULA), and by using the Microsoft Software under the BYOL program, you agree to the Microsoft EULA.
- AWS recommends that you consult with your own legal and other advisers to understand and comply with the applicable Microsoft licensing requirements. Usage of the Services (including usage of the **licenseType** parameter and **BYOL** flag) in violation of your agreements with Microsoft is not authorized or permitted.

Limitations

Image Format

- When migrating VMs managed by Hyper-V/SCVMM, SMS supports both Generation 1 VMs (using either VHD or VHDX disk format) and Generation 2 VMs (VHDX only).
- AWS SMS does not support VMs on Hyper-V running any version of RHEL 5 if backed by a VHDX disk. We recommend converting disks in this format to VHD for migration.
- AWS SMS does not support VMs that have a mix of VHD and VHDX disk files.
- On VMware, AWS SMS does not support VMs that use Raw Device Mapping (RDM). Only VMDK disk images are supported.

File System

- Migrated Linux VMs must use 64-bit images. Migrating 32-bit Linux images is not supported.
- Migrated Linux VMs should use default kernels for best results. VMs that use custom Linux kernels might not migrate successfully.
- When preparing Amazon EC2 Linux VMs for migration, make sure that at least 250 MiB of disk space is available on the root volume for installing drivers and other software. For Microsoft Windows VMs, configure a fixed pagefile size and ensure that at least 6 GiB of free space is available on the root volume.

Booting

- UEFI/EFI boot partitions are supported only for Windows boot volumes with VHDX as the image format. Otherwise, a VM's boot volume must use Master Boot Record (MBR) partitions. In either case, boot volume cannot exceed 2 TiB (uncompressed) due to MBR limitations.

  > **Note**
  > When AWS detects a Windows GPT boot volume with an UEFI boot partition, it converts it on-the-fly to an MBR boot volume with a BIOS boot partition. This is because EC2 does not directly support GPT boot volumes.

- An imported VM might fail to boot if the root partition is not on the same virtual hard drive as the MBR.
- A migrated VM might fail to boot if the root partition is not on the same virtual hard disk as the MBR.
- Migrating VMs with dual-boot configurations is not supported.

Networking

- Multiple network interfaces are not currently supported. After migration, your VM will have a single virtual network interface that uses DHCP to assign addresses. Your instance receives a private IP address.
- A VM migrated into a VPC does not receive a public IP address, regardless of the auto-assign public IP setting for the subnet. Instead, you can allocate an Elastic IP address to your account and associate it with your instance.
- Internet Protocol version 6 (IPv6) IP addresses are not supported.

Application Import from Migration Hub

- SMS imports application-related servers from AWS Migration Hub only if they exist in the SMS Server Catalog. As a result, some applications may only be partially migrated.
- If none of the servers in a Migration Hub application exist in the SMS Server Catalog, the import will fail silently and the application will not be visible in SMS.
- Imported applications can be migrated but cannot be edited in SMS. They can, however, be edited in Migration Hub.

Miscellaneous

- An SMS replication job will fail for VMs with more than 22 volumes attached.

- AMIs with volumes using EBS encryption are not supported. When migrating servers using AWS SMS, do not turn on encryption by default. If encryption by default is already on and you are experiencing delta replication failures, turn off this feature.
- AWS SMS creates AMIs that use Hardware Virtual Machine (HVM) virtualization. It can't create AMIs that use Paravirtual (PV) virtualization. Linux PVHVM drivers are supported within migrated VMs.
- VMs that are created as the result of a P2V conversion are not supported. A P2V conversion occurs when a disk image is created by performing a Linux or Windows installation process on a physical machine and then importing a copy of that Linux or Windows installation to a VM.
- AWS SMS does not install the single root I/O virtualization (SR-IOV) drivers except with imports of Microsoft Windows Server 2012 R2 VMs. These drivers are not required unless you plan to use enhanced networking, which provides higher performance (packets per second), lower latency, and lower jitter. For Microsoft Windows Server 2012 R2 VMs, SR-IOV drivers are automatically installed as a part of the migration process.
- Because independent disks are unaffected by snapshots, AWS SMS does not support interval replication for VMDKs in independent mode.
- Windows language packs that use UTF-16 (or non-ASCII) characters are not supported for import. We recommend using the English language pack when importing Windows Server 2003, Windows Server 2008, and Windows Server 2012 R1 VMs.
- For Windows Server 2003, disable Windows driver-signing checks before migrating.

Other Requirements

Support for VMware vMotion

AWS Server Migration Service partially supports vMotion, Storage vMotion, and other features based on virtual machine migration (such as DRS and Storage DRS) subject to the following limitations:

- Migrating a virtual machine to a new ESXi host or datastore after one replication run ends, and before the next replication run begins, is supported as long as the Server Migration Connector's vCenter service account has sufficient permissions on the destination ESXi host, datastores, and datacenter, and on the virtual machine itself at the new location.
- Migrating a virtual machine to a new ESXi host, datastore, and/or datacenter while a replication run is active—that is, while a virtual machine upload is in progress—is not supported.
- Cross vCenter vMotion is not supported for use with the AWS Server Migration Service.

Support for VMware vSAN

VMs on vSAN datastores are only supported when **Replication job type** on the **Configure replication jobs settings** page is set to **One-time migration**.

Support for VMware Virtual Volumes (VVol)

AWS does not provide support for migrating VMware Virtual Volumes. Some implementations may work, however.

VMs with Snapshots

AWS SMS supports only one-time migration on VMs where snapshot-based backup software is used. Also, avoid creating snapshots on VMs replicated through AWS SMS.

Permissions for IAM Users

The following permission and role prerequisites are required by AWS SMS.

Configure User Permissions for AWS SMS

If your IAM user account, group, or role is assigned administrator permissions, then you already have access to AWS SMS. To call the AWS SMS API with the credentials of an IAM user that does not have administrative access to your AWS account:

- Create a custom inline policy defined by the following JSON code.
- Apply it to the IAM user.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "sms:*"
         ],
         "Effect":"Allow",
         "Resource":"*"
      },
      {
         "Action":[
            "cloudformation:ListStacks",
            "cloudformation:DescribeStacks",
            "cloudformation:DescribeStackResources"
         ],
         "Effect":"Allow",
         "Resource":"*"
      },
      {
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:GetObject"
         ],
         "Effect":"Allow",
         "Resource":"*"
      },
      {
         "Action":[
            "ec2:DescribeKeyPairs",
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Effect":"Allow",
         "Resource":"*"
      },
      {
         "Action":"iam:PassRole",
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "iam:AssociatedResourceArn":"arn:aws:cloudformation:*:*:stack/sms-app-*/*"
            }
         }
      },
      {
         "Action":[
             "iam:ListRoles",
             "iam:CreateServiceLinkedRole"
         ],
         "Effect":"Allow",
         "Resource":"*"
      }
   ]
}
```

Note
If you are using multiple connectors, we recommend that you create a unique IAM role for each connector to avoid having a single point of failure.

Configure an IAM User for Server Migration Connector

To create an IAM user for Server Migration Connector in your AWS account

1. Create a new IAM user for your connector to communicate with AWS. Save the generated access key and secret key for use during the initial connector setup. For information about managing IAM users and permissions, see Creating an IAM User in Your AWS Account.
2. Attach the managed IAM policy `ServerMigrationConnector` to the IAM user. For more information, see Managed Policies and Inline Policies.

Install the Server Migration Connector

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

Contents

Install the Server Migration Connector on VMware

Use the following information to install the Server Migration Connector, so that you can use AWS SMS to migrate VMs from a VMware environment to Amazon EC2.

This information applies only to VMs in an on-premises VMware environment. For information about installing the connector on other environments, see Install the Server Migration Connector (p. 11).

Requirements for VMware connector

- vCenter version 5.1 or higher (validated up to 6.7)
- ESXi 5.1 or higher (validated up to 6.7)
- Minimum 8 GiB RAM
- Minimum available disk storage of 20 GiB (thin-provisioned) or 250 GiB (thick-provisioned)
- Support for the following network services. Note that you might need to reconfigure your firewall to permit stateful outbound connections from the connector to these services.
  - DNS—Allow the connector to initiate connections to port 53 for name resolution.
  - HTTPS on vCenter—Allow the connector to initiate secure web connections to port 443 of vCenter. You can also configure a non-default port at your discretion. If your vCenter Server is configured to use a non-default port, specify both the vCenter's hostname and port, separated by a colon (for example, `HOSTNAME:PORT` or `IP:PORT`) in the vCenter Service Account page in **Connector setup**.
  - HTTPS on ESXi—Allow the connector to initiate secure web connections to port 443 of the ESXi hosts containing the VMs you intend to migrate.
  - NTP—Optionally allow the connector outbound access to port 123 for time synchronization. If the connector synchronizes its clock with the ESXi host, this is unnecessary.
- Allow outbound connections from the connector to the following URL ranges:
  - *.amazonaws.com
  - *.aws.amazon.com

To set up the connector for a VMware environment

1. Open the **AWS Server Migration Service** console and choose **Connectors**, **SMS Connector setup guide**.
2. On the **AWS Server Migration Connector setup** page, choose **Download OVA** to download the connector for VMware environments. You can also download the connector using the URL provided.

The connector is a preconfigured FreeBSD VM in OVA format that is ready for deployment in your vCenter.

3. Set up your vCenter service account. Create a vCenter user with permissions necessary to create and delete snapshots on VMs that need be migrated to AWS and download their delta disks.

   Note
   As a best practice, we recommend that you limit vCenter permissions for the connector service account to only those vCenter data centers that contain the VMs that you intend to migrate. We also recommend that you lock down your vCenter service account permissions by assigning this user the NoAccess role in vCenter on the hosts, folders, and datastores that do not have any VMs for migration.

4. Create a role in vCenter with the following privileges:

   - **Datastore** > **Browse datastore and Low level file operations** (Datastore.Browse and Datastore.FileManagement)
   - **Host** > **Configuration** > **System Management** (Host.Config.SystemManagement)
   - **vApp** > **Export** (VApp.Export)
   - **Virtual Machine** > **Snapshot management** > **Create snapshot and Remove Snapshot** (VirtualMachine.State.CreateSnapshot and VirtualMachine.State.RemoveSnapshot)

5. Assign the role as follows:

   a. Assign this vCenter role to the service account for the connector to use to log in to vCenter.

   b. Assign this role with propagating permissions to the data centers that contain the VMs to migrate.

6. To manually verify your vCenter service account’s permissions, verify that you can log in to vSphere Client with your connector service account credentials. Then, export your VMs as OVF templates, use the datastore browser to download files off the datastores that contain your VMs, and view the properties on the **Summary** tab of the ESXi hosts of your VMs.

To configure the connector

1. Deploy the connector OVA downloaded in the previous procedure to your VMware environment using vSphere Client.
2. Open the connector's virtual machine console and log in as `ec2-user` with the password `ec2pass`. Supply a new password if prompted.
3. Obtain the IP address of the connector as follows:

   a. Run the command **sudo setup.rb**. This displays a configuration 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
Please enter your option [1-9]:
```

   b. Enter option 2. This displays current network information and a submenu for making changes to the network settings. The output should resemble the following:

```
Current network configuration: DHCP
IP: 192.0.2.100
```

```
Netmask: 255.255.254.0
Gateway: 192.0.2.1
DNS server 1: 192.0.2.200
DNS server 2: 192.0.2.201
DNS suffix search list: subdomain.example.com
Web proxy: not configured

Reconfigure your network:
      1. Renew or acquire a DHCP lease
      2. Set up a static IP
      3. Set up a web proxy for AWS communication
      4. Set up a DNS suffix search list
      5. Exit
Please enter your option [1-5]:
```

You need to enter this IP address in later procedures.

4. [Optional] Configure a static IP address for the connector. This prevents you from having to reconfigure the trusted hosts lists on your LAN each time DHCP assigns a new address to the connector.

   In the **Reconfigure your network** menu, enter option **2**. This displays a form to supply network settings:

   For each field, provide an appropriate value and press Enter. You should see output similar to the following:

```
Setting up static IP:
      1. Enter IP address: 192.0.2.50
      2. Enter netmask: 255.255.254.0
      3. Enter gateway: 192.0.2.1
      4. Enter DNS 1: 192.0.2.200
      5. Enter DNS 2: 192.0.2.201

Static IP address configured.
```

5. In the connector's network configuration menu, configure domain suffix values for the DNS suffix search list.
6. If your environment uses a web proxy to reach the internet, configure that now.
7. Before leaving the connector console, use **ping** to verify network access to the following targets inside and outside your LAN:

   - Inside your LAN, to your ESXi hosts and vCenter by hostname, FQDN, and IP address
   - Outside your LAN, to AWS
8. In a web browser, access the connector VM at its IP address (https://*ip-address-of-connector*/) to open the setup wizard, and choose **Get started now**.
9. Review the license agreement, select the check box, and choose **Next**.
10. Create a password for the connector.
11. Choose **Upload logs automatically** and **Server Migration Connector auto-upgrade**.
12. For **AWS Region**, choose your Region from the list. For **AWS Credentials**, enter the IAM credentials that you created in Permissions for IAM Users (p. 8). Choose **Next**.
13. For **vCenter Service Account**, enter the vCenter hostname, user name, and password from step 3. Choose **Next**.
14. After accepting the vCenter certificate, complete registration and then view the connector configuration dashboard.
15. Verify that the connector you registered shows up on the **Connectors** page.

Install the Server Migration Connector on Hyper-V

AWS SMS supports migration in either of two modes: from standalone Hyper-V servers, or from Hyper-V servers managed by System Center Virtual Machine Manager (SCVMM). Use the following information to install the Server Migration Connector on Hyper-V so that you can use AWS SMS to migrate VMs from Hyper-V to Amazon EC2.

This information applies only to VMs in an on-premises Hyper-V environment. For information about installing the connector on other environments, see Install the Server Migration Connector (p. 11).

Considerations for Migration Scenarios

- The installation procedures for standalone Hyper-V and for SCVMM environments are not interchangeable.
- When configured in SCVMM mode, one Server Migration Connector appliance supports migration from one SCVMM (which may manage multiple Hyper-V servers).
- When configured in standalone Hyper-V mode, one Server Migration Connector appliance supports migration from multiple Hyper-V servers.
- AWS SMS supports deploying any number of connector appliances to support migration from multiple SCVMMs and multiple standalone Hyper-V servers in parallel.

Requirements for Hyper-V connector

- Hyper-V role on Windows Server 2012 R2 or Windows Server 2016
- Active Directory 2012 or above
- [Optional] SCVMM 2012 SP1 or SCVMM 2016
- Minimum 8 GiB RAM
- Minimum available disk storage of 300 GiB
- Support for the following network services. Note that you might need to reconfigure your firewall to permit stateful outbound connections from the connector to these services.
  - DNS—Allow the connector to initiate connections to port 53 for name resolution.
  - HTTPS on WinRM port 5986 on your SCVMM or standalone Hyper-V host
  - Inbound HTTPS on port 443 of the connector—Allow the connector to receive secure web connections on port 443 from Hyper-V hosts containing the VMs you intend to migrate.
  - NTP—Optionally allow the connector outbound access to port 123 for time synchronization. If the connector synchronizes its clock with the Hyper-V host, this is unnecessary.
- Allow outbound connections from the connector to the following URL ranges:
  - *.amazonaws.com
  - *.aws.amazon.com

Contents

About the Server Migration Connector Installation Script

The AWS SMS configuration script automates creation of appropriate permissions and network connections that allow AWS SMS to execute tasks on your Hyper-V environment. You must run the script as administrator on each Hyper-V and SCVMM host that you plan to use in migrating VMs. When you run the script, it performs the following actions:

1. **[All systems]** Checks whether the Windows Remote Management (WinRM) service is enabled on SCVMM and all Hyper-V hosts, enables it if necessary, and sets it to start automatically on boot.
2. **[All systems]** Enables PowerShell remoting, which allows the connector to execute PowerShell commands on that host over a WinRM connection.
3. **[All systems]** Creates a self-signed X.509 certificate, creates a WinRM HTTPS listener, and binds the certificate to the listener.
4. **[All systems]** Creates a firewall rule to accept incoming connections to the WinRM listener.
5. **[All systems]** Adds the IP address or domain name of the connector to the list of trusted hosts in the WinRM configuration. You must have this IP address or domain name configured before running the script so that you can provide it interactively.
6. **[All systems]** Enables Credential Security Support Provider (CredSSP) authentication with WinRM.
7. **[All systems]** Grants read and execute permissions to a pre-configured Active Directory user on WinRM configSDDL. This user is the same as the service account described below in Step 1: Create a Service Account for Server Migration Connector in Active Directory (p. 15).
8. **[Standalone Hyper-V only]** Adds the Active Directory user to the groups Hyper-V Administrators and Remote Management Users on your Hyper-V host.
9. **[Standalone Hyper-V only]** Gives read-only permissions to all VM data folders manged by this Hyper-V.
10. **[SCVMM only]** Grants "Execute Methods," "Enable Account," and "Remote Enable" permissions to the Active Directory user on two WMI objects, CIMV2 and SCVMM.
11. **[SCVMM only]** Creates a Delegated Administrator role in SCVMM with permissions to access all Hyper-V hosts. It assigns the role to the Active Directory user. You can selectively remove access to hosts by editing this role in SCVMM.
12. **[SCVMM only]** Checks whether a secure (HTTPS) network path exists between SCVMM and the Hyper-V hosts. If the script does not detect a secure channel, it returns an error and generates instructions for the administrator to secure the channel.
13. **[SCVMM only]** Iterates through all the Hyper-V hosts managed by SCVMM and grants the Active Directory user read-only permissions on all VM folders on each Hyper-V host.

Step 1: Create a Service Account for Server Migration Connector in Active Directory

The Server Migration Connector requires a service account in Active Directory. As the connector configuration script is run on each SCVMM and Hyper-V host, it grants permissions on those hosts to this account.

> **Note**
> When configured in SCVMM mode, the SCVMM host and all the Hyper-V hosts that it manages must be located in a single Active Directory domain. If you have multiple Active Directory domains, configure a connector for each.

To create the Active Directory user

1. Using the Active Directory Administrative Center on the Windows computer where your Active Directory forest is installed, create a new user and assign a password to it.
2. Add the new user to the **Remote Management Users** group.

Step 2: Download and Deploy the Server Migration Connector

Download the Server Migration Connector for Hyper-V and SCVMM to your on-premises environment and install it on a Hyper-V host.

To set up the connector for a Hyper-V environment

1. Open the **AWS Server Migration Service** console and choose **Connectors**,**SMS Connector setup guide**.
2. On the **AWS Server Migration Connector setup** page, choose **Download VHD ZIP** to download the connector for Hyper-V.
3. Transfer the downloaded connector file to your Hyper-V host, unzip it, and import the connector as a VM.
4. Open the connector's virtual machine console and log in as `ec2-user` with the password `ec2pass`. Supply a new password if prompted.
5. Obtain the IP address of the connector as follows:

   a. Run the command **sudo setup.rb**. This displays a configuration 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
Please enter your option [1-9]:
```

   b. Enter option 2. This displays current network information and a submenu for making changes to the network settings. The output should resemble the following:

```
Current network configuration: DHCP
IP: 192.0.2.100
Netmask: 255.255.254.0
Gateway: 192.0.2.1
DNS server 1: 192.0.2.200
DNS server 2: 192.0.2.201
DNS suffix search list: subdomain.example.com
Web proxy: not configured

Reconfigure your network:
      1. Renew or acquire a DHCP lease
      2. Set up a static IP
      3. Set up a web proxy for AWS communication
      4. Set up a DNS suffix search list
      5. Exit
```

```
Please enter your option [1-5]:
```

You need to enter this IP address in later procedures.

6. [Optional] Configure a static IP address for the connector. This prevents you from having to reconfigure the trusted hosts lists on your LAN each time DHCP assigns a new address to the connector.

   In the **Reconfigure your network** menu, enter option **2**. This displays a form to supply network settings:

   For each field, provide an appropriate value and press Enter. You should see output similar to the following:

```
Setting up static IP:
      1. Enter IP address: 192.0.2.50
      2. Enter netmask: 255.255.254.0
      3. Enter gateway: 192.0.2.1
      4. Enter DNS 1: 192.0.2.200
      5. Enter DNS 2: 192.0.2.201

Static IP address configured.
```

7. In the connector's network configuration menu, configure domain suffix values for the DNS suffix search list.
8. If your environment uses a web proxy to reach the internet, configure that now.
9. Before leaving the connector console, use **ping** to verify network access to the following targets inside and outside your LAN:

   - Inside your LAN, to your Hyper-V hosts and SCVMM by hostname, FQDN, and IP address
   - Outside your LAN, to AWS

Step 3: Download and Install the Hyper-V/SCVMM Configuration Script

AWS SMS provides a downloadable PowerShell script to configure the Windows environment to support communications with the Server Migration Connector. The same script is used for configuring either standalone Hyper-V or SCVMM. The script is cryptographically signed by AWS.

Download the script and hash files from the following URLs:

File	URL
Installation script	https://s3.amazonaws.com/sms-connector/aws-sms-hyperv-setup.ps1
MD5 hash	https://s3.amazonaws.com/sms-connector/aws-sms-hyperv-setup.ps1.md5
SHA256 hash	https://s3.amazonaws.com/sms-connector/aws-sms-hyperv-setup.ps1.sha256

After download, transfer the downloaded files to the computer or computers where you plan to run the script.

Step 4: Validate the Integrity and Cryptographic Signature of the Script File

Before running the script, we recommend that you validate its integrity and signature. These procedures assume that you have downloaded the script and the hash files, that they are installed on the desktop of the computer where you plan to run the script, and that you are signed in as the administrator. You may need to modify the procedures to match your setup.

To validate script integrity using cryptographic hashes (PowerShell)

1. Use one or both of the downloaded hash files to validate the integrity of the script file, ensuring that it has not changed in transit to your computer.

    a. To validate with the MD5 hash, run the following command in a PowerShell window:

    ```
    PS C:\Users\Administrator> Get-FileHash aws-sms-hyperv-setup.ps1 -Algorithm MD5
    ```

    This should return information similar to the following:

    ```
    Algorithm         Hash
    ---------         ----
    MD5               1AABAC6D068EEF6EXAMPLEDF50A05CC8
    ```

    b. To validate with the SHA256 hash, run the following command in a PowerShell window:

    ```
    PS C:\Users\Administrator> Get-FileHash aws-sms-hyperv-setup.ps1 -Algorithm SHA256
    ```

    This should return information similar to the following:

    ```
    Algorithm         Hash
    ---------         ----
    SHA256            6B86B273FF34FCE19D6B804EFF5A3F574EXAMPLE22F1D49C01E52DDB7875B4B
    ```

2. Compare the returned hash values with the values provided in the downloaded files, `aws-sms-hyperv-setup.ps1.md5` and `aws-sms-hyperv-setup.ps1.sha256`.

Next, use either the Windows user interface or PowerShell to check that the script file includes a valid signature from AWS.

To check the script file for a valid cryptographic signature (Windows GUI)

1. In Windows Explorer, open the context (right-click) menu on the script file and choose **Properties**, **Digital Signatures**, **Amazon Web Services**, and **Details**.
2. Verify that the displayed information contains "This digital signature is OK" and that "Amazon Web Services, Inc." is the signer.

To check the script file for a valid cryptographic signature (PowerShell)

- In a PowerShell window, run the following command:

```
PS C:\Users\Administrator> Get-AuthenticodeSignature aws-sms-hyperv-setup.ps1 | Select
 *
```

A correctly signed script file should return information similar to the following:

```
SignerCertificate           : [Subject]
                             CN="Amazon Web Services, Inc." ...
                          [Issuer]
                             CN=DigiCert EV Code Signing CA (SHA2), OU=www.digicert.com,
 O=DigiCert Inc, C=US
...

TimeStamperCertificate      :
Status                      : Valid
StatusMessage               : Signature verified.
Path                        : C:\Users\Administrator\Desktop\aws-sms-hyperv-setup.ps1

...
```

Step 5: Run the Script

This procedure assumes that you have downloaded the script onto the desktop of the computer where you plan to run the script, and that you are signed in as the administrator. You may need to modify the procedure shown to match your setup.

> **Note**
> If you are using SCVMM, you must first run this script on each Hyper-V host you plan to migrate from, and then run it on SCVMM.

To run the script on each host

1. Using RDP, log in to your SCVMM system or standalone Hyper-V host as administrator.
2. Run the script using the following PowerShell command:

```
PS C:\Users\Administrator> .\aws-sms-hyperv-setup.ps1
```

> **Note**
> If your PowerShell execution policy is set to verify signed scripts, you are prompted for an authorization when you run the connector configuration script. Verify that the script is published by "Amazon Web Services, Inc." and choose "R" to run one time. You can view this setting using **Get-ExecutionPolicy** and modify it using **Set-ExecutionPolicy**.

3. As the script runs, it prompts you for several pieces of information. Be prepared to respond to the following prompts:

Script action	Customer prompt	Customer action
Prompts for an option based on the connector's mode of operation (migrate from standalone Hyper-V vs. migrate using SCVMM), which determines what changes must be made to your Windows environment.	**0. Exit** **1. Reconfigure standalone Hyper-V...** **2. Reconfigure Hyper-V managed by SCVMM...** **3. Reconfigure SCVMM...** **4. Help/Support**	Choose **0** to exit the script. Choose **1** to reconfigure a standalone Hyper-V host to allow migration of its guest VMs. Choose **2** to reconfigure a Hyper-V host allow SCVMM to manage migration its guest VMs.

Script action	Customer prompt	Customer action
		Choose **3** to reconfigure SCVMM to allow migration of guest VMs on all Hyper-V hosts it manages. Option **4** links to this document and to information about AWS Support.
Prompts for the Active Directory user that the connector uses when communicating with SCVMM and Hyper-V.	**Enter the AD user that the connector will use (DOMAIN \user)**	Provide the Active Directory user that you previously configured. For more information, see Step 1: Create a Service Account for Server Migration Connector in Active Directory (p. 15).
Prompts for the IP address or hostname of the connector.	**Enter the IP Address or Hostname of the connector appliance**	Provide the IP address or host name that you configured on the connector.
Prompts for confirmation before modifying your Windows environment.	**Make changes to Windows system configuration? (Enter "yes" or "no")**	Enter "yes" and press Enter to start the reconfiguration. Entering "no" causes the script to exit.

Step 6: Configure the Connector

When the connector configuration has been successfully run, browse to the connector's web interface:

```
https://ip-address-of-connector/
```

Complete the following steps to set up the new connector.

To configure the connector

1. On the connector landing page, choose **Get started now**.
2. Review the license agreement, select the check box, and choose **Next**.
3. Create a password for the connector. The password must meet the displayed criteria. Choose **Next**.
4. On the **Network Info** page, you can (among other tasks) assign a static IP address to the connector if you have not already done so. Choose **Next**.
5. On the **Log Uploads and Upgrades** page, select **Upload logs automatically** and **Server Migration Connector auto-upgrade**, and choose **Next**.
6. On the **Server Migration Service** page, provide the following information:
   - For **AWS Region**, choose your Region from the list.
   - For **AWS Credentials**, enter the IAM credentials that you created in Permissions for IAM Users (p. 8). Choose **Next**.
7. On the **Choose your VM manager type** page, choose either **Microsoft® System Center Virtual Manager (SCVMM)** or **Microsoft® Hyper-V** depending on your environment. Selecting **VMware® vCenter** results in an error if you have installed the Hyper-V connector. Choose **Next**.

8. On the **Hyper-V: Host and Service Account Setup** or **SCVMM: Host and Service Account Setup** page, provide the account information for the Active Directory user that you created in Step 1: Create a Service Account for Server Migration Connector in Active Directory (p. 15), including **Username** and **Password**.
9. - [SCVMM only] Provide the SCVMM hostname to be served by this connector and choose **Next**. Inspect the certificate for the host and choose **Trust** if the certificate is valid.
   - [Stand-alone Hyper-V only] Provide the Hyper-V hostname for each host to be served by this connector. To add additional hosts, use the plus symbol. To inspect the certificate for each host, choose **Verify Certificate** and choose **Trust** if the certificate is valid. Choose **Next**.

   Alternatively, you can select the host-specific option to **Ignore hostname mismatch and expiration errors...** for either SCVMM or Hyper-V host certificates. We do not recommend overriding security in production, but it may be useful during testing.

   > **Note**
   > If you have Hyper-V hosts located in multiple Active Directory domains, we recommend configuring a separate connector for each domain.
10. If you successfully authenticated with the connector, you should see the **Congratulations** page. To view the connector's health status, choose **Go to connector dashboard**.
11. To verify that the connector that you registered is now listed, open the **Connectors** page on the AWS Server Migration Service console.

Install the Server Migration Connector on Azure

Use the following information to install the Server Migration Connector on Azure so that you can use AWS SMS to migrate VMs from Azure to Amazon EC2.

This information applies only to VMs hosted by Azure. For information about installing the connector on other environments, see Install the Server Migration Connector (p. 11).

Considerations for Migration Scenarios

- A single Server Migration Connector appliance can only migrate VMs under one subscription and one Azure Region.
- After a Server Migration Connector appliance is deployed, you cannot change its subscription or Region unless you deploy another connector in the new subscription/Region.
- AWS SMS supports deploying any number of Server Migration Connector appliance VMs to support migration from multiple Azure subscriptions and Regions in parallel.

Requirements for Azure connector

- The recommended VM size of Azure connector is F4s – 4 vCPUs and 8 GB RAM. Ensure that you have a sufficient Azure CPU quota in the region where you are deploying the connector.
- A Standard Storage Account (cannot be Premium) under which the connector can be deployed.
- A virtual network where the connector can be deployed.
- Inbound access on port 443 (HTTPS), either from within the connector’s virtual network (recommended) or open to the public (not recommended), for connector registration and viewing the connector dashboard.
- Outbound Internet access to access AWS services, Azure services, to perform connector OS updates, and so on.

Contents

Step 1: Download the Connector Installation Script

AWS SMS provides a downloadable PowerShell script to deploy the connector in your Azure environment. The script is cryptographically signed by AWS. Complete this procedure to run the PowerShell script and install the connector automatically in your Azure environment. The script requires PowerShell 5.1 or later.

> **Note**
> AWS recommends using the scripted installation, but you can alternatively install the connector manually. For more information, see (Alternative Procedure) Deploy the Server Migration Connector Manually (p. 25).

To download the script and hash files

1. Download the PowerShell script and hash files from the following URLs:

File	URL
Installation script	https://s3.amazonaws.com/sms-connector/aws-sms-azure-setup.ps1
MD5 hash	https://s3.amazonaws.com/sms-connector/aws-sms-azure-setup.ps1.md5
SHA256 hash	https://s3.amazonaws.com/sms-connector/aws-sms-azure-setup.ps1.sha256

2. After download, transfer the files to the computer or computers where you plan to run the script.

Step 2: Validate the Integrity and Cryptographic Signature of the Script File

Before running the script, we recommend that you validate its integrity and signature, ensuring that it has not changed in transit to your computer. These procedures assume that you have downloaded the script and the hash files, that they are installed on the desktop of the computer where you plan to run the script, and that you are signed in as administrator. You may need to modify the procedures to match your setup.

To validate script integrity using cryptographic hashes (PowerShell)

1. Use one or both of the downloaded hash files to validate the integrity of the script file.

   a. To validate with the MD5 hash, run the following command in a PowerShell window:

   ```
   PS C:\Users\Administrator> Get-FileHash aws-sms-azure-setup.ps1 -Algorithm MD5
   ```

   This should return information similar to the following:

   ```
   Algorithm         Hash
   ```

```
---------         ----
MD5               1AABAC6D068EEF6EXAMPLEDF50A05CC8
```

b. To validate with the SHA256 hash, run the following command in a PowerShell window:

```
PS C:\Users\Administrator> Get-FileHash aws-sms-azure-setup.ps1 -Algorithm SHA256
```

This should return information similar to the following:

```
Algorithm         Hash
---------         ----
SHA256            6B86B273FF34FCE19D6B804EFF5A3F574EXAMPLE22F1D49C01E52DDB7875B4B
```

2. Compare the returned hash values with the values provided in the downloaded files, `aws-sms-azure-setup.ps1.md5` and `aws-sms-azure-setup.ps1.sha256`.

Next, use either PowerShell or the Windows user interface to check that the script file includes a valid signature from AWS.

To check the script file for a valid cryptographic signature (PowerShell)

- In a PowerShell window, run the following command:

```
PS C:\Users\Administrator> Get-AuthenticodeSignature aws-sms-azure-setup.ps1 | Select *
```

A correctly signed script file should return information similar to the following:

```
SignerCertificate          : [Subject]
                            CN="Amazon Web Services, Inc." ...
                         [Issuer]
                            CN=DigiCert EV Code Signing CA (SHA2), OU=www.digicert.com,
 O=DigiCert Inc, C=US
...

TimeStamperCertificate     :
Status                     : Valid
StatusMessage              : Signature verified.
Path                       : C:\Users\Administrator\Desktop\aws-sms-azure-setup.ps1

...
```

To check the script file for a valid cryptographic signature (Windows GUI)

1. In Windows Explorer, open the context (right-click) menu on the script file and choose **Properties**, **Digital Signatures**, **Amazon Web Services**, and **Details**.
2. Verify that the displayed information contains "This digital signature is OK" and that "Amazon Web Services, Inc." is the signer.

Step 3: Run the Script

Run this script from any computer with PowerShell 5.1 or later installed.

> **Note**
> If your PowerShell execution policy is set to verify signed scripts, you are prompted for an authorization when you run the connector configuration script. Verify that the script is

published by "Amazon Web Services, Inc." and choose "R" to run one time. You can view this setting using **Get-ExecutionPolicy** and modify it using **Set-ExecutionPolicy**.

```
PS C:\Users\Administrator> .\aws-sms-azure-setup.ps1 -StorageAccountName name -
ExistingVNetName name -SubscriptionId id -SubnetName name
```

`StorageAccountName`

The name of the storage account where you want to deploy the connector.

`ExistingVNetName`

The name of the virtual network where you want to deploy the connector.

`SubscriptionId`

(Optional) The ID of the subscription to use. If you do not specify this parameter, the default subscription for the account is used.

`SubnetName`

(Optional) The name of the subnet in the virtual network. If you do not specify this parameter, the subnet named "default" is used.

When the script prompts for an Azure login, use a login that has administrator permissions for the subscription under which you are deploying the connector.

When the script completes, the connector is deployed in your account. The script prints out the connector's private IP address and the Object ID of the System Assigned Identity of the connector VM. You need both of these to complete the next step.

Step 4: Configure the Connector

From another VM on the same virtual network where you deployed the connector, browse to the connector's web interface using the following URL, which includes the private IP address of the connector that you obtained in the previous step:

```
https://ip-address-of-connector
```

To configure the connector

1. On the connector landing page, choose **Get started now**.
2. Review the license agreement, select the check box, and choose **Next**.
3. Create a password for the connector. The password must meet the displayed criteria. Choose **Next**.
4. On the **Network Info** page, you can find instructions to perform network-related tasks, such as setting up AWS proxy for the connector. Choose **Next**.
5. On the **Log Uploads** page, select **Upload logs automatically** and choose **Next**.
6. On the **Server Migration Service** page, provide the following information:
   - For **AWS Region**, choose your Region from the list.
   - For **AWS Credentials**, enter the IAM credentials that you created in Permissions for IAM Users (p. 8). Choose **Next**.
7. On the **Azure Account Verification** page, verify that your Azure subscription ID and location are correct. This connector can migrate VMs under this subscription and location. Provide the object ID of the System Assigned Identity of the connector VM, which was provided as output from the deployment script.

8. If you successfully set up the connector, the **Congratulations** page is displayed. To view the health status of the connector, choose **Go to connector dashboard**.
9. To verify that the connector that you registered is listed, open the **Connectors** page on the Systems Manager console.

(Alternative Procedure) Deploy the Server Migration Connector Manually

Complete this procedure to install the connector manually in your Azure environment.

To install the connector manually

1. Log into the Azure Portal as a user with administrator permissions for the subscription under which you are deploying this connector.
2. Make sure that you are ready to supply a Storage Account, its Resource Group, a Virtual Network, and the Azure Region as described in Requirements for Azure connector (p. 21).
3. Download the connector VHD and associated files from the URLs in the following table.

File	URL
Connector VHD	https://awssmsconnector.blob.core.windows.net/release/AWS-SMS-Connector-for-Azure.vhd
MD5 hash	https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector-for-Azure.vhd.md5
SHA256 hash	https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector-for-Azure.vhd.sha256

4. Verify the cryptographic integrity of the connector VHD using procedures similar to those described in Step 2: Validate the Integrity and Cryptographic Signature of the Script File (p. 22).
5. Upload the connector VHD and associated files to your Storage Account.
6. Create a new managed disk with the following parameter values:
   - **Resource Group**: Select your resource group
   - **Name**: Any name - for example, sms-connector-disk-westus
   - **Region**: Select your Azure Region
   - **Availability Zone**: None
   - **Source Type**: Storage Blob (Choose the VHD blob you uploaded from step 3.c.)
   - **OSType**: Linux
   - **Size**: 60 GB/Standard HDD
7. Choose **Create VM** to create a new virtual machine from the managed disk that you created. Assign the following parameter values.

   Under the **Basics** tab:

   - **Resource Group**: Enter in your resource group
   - **Virtual Machine Name**: Any name, for example sms-connector-vm-westus
   - **Region**: Select your Azure Region
   - **Size**: F4s
   - **Public Inbound Ports**: None

Under the **Disks** tab:

- **OS Disk Type**: Standard HDD

Under the **Networking** tab:

- **Virtual Network**: Enter in your Virtual Network name
- **Subnet**: Leave as default or choose a particular subnet
- **Public IP**: Leave as new
- **NIC Network Security Group**: Basic
- **Public Inbound Ports**: None
- Accept defaults for the remaining fields.

Under the **Management** tab:

- **Boot Diagnostics**: On
- **OS Guest Diagnostics**: Off
- **Diagnostics Storage account**: Storage Account
- **System Assigned Managed Identity**: On
- **Enable auto-shutdown**: Off

8. Review and create the VM. This will be your connector VM.
9. Download the two role documents:

   - https://s3.amazonaws.com/sms-connector/SMSConnectorRole.json
   - https://s3.amazonaws.com/sms-connector/SMSConnectorRoleSA.json

10. **(Important)** Customize the role documents.

    Edit `SMSConnectorRole.json`. Change the `name` field to `sms-connector-role-`*`subscription_id`*. Then change the `AssignableScopes` field to match your subscription ID.

    Edit `SMSConnectorRoleSA.json`. Change the `name` field to `sms-connector-role-`*`storage_account`*. For example, if your account is *testStorage*, then the name field must be `sms-connector-role-testStorage`. Then change the `AssignableScopes` field to match your Subscription, Resource Group, and Storage Account values.
11. Create a role definition. Currently, there is no way to create a role definition from the Azure Portal. You must use Az CLI or Az PowerShell for this step. Use the New-AzRoleDefinition (Az PowerShell) or az role definition create (Az CLI) command to create these custom roles in your subscription, using the JSON files that you created in the previous step.
12. Assign roles to the connector VM. In Azure Portal, choose **Storage Account**, **Access Control**, **Roles**, **Add**, **Add Role Assignment**. Choose the role `sms-connector-role`, assign access to *Virtual Machine*, and select the connector VM's System Assigned Identity from the list. Repeat this for the role `sms-connector-role-`*`storage_account`*.
13. Restart the connector VM to activate the role assignments.
14. Continue to Step 4: Configure the Connector (p. 24).

Replicate VMs Using the AWS SMS Console

Use the AWS SMS console to import your server catalog and migrate your on-premises servers to Amazon EC2.

If you have enabled integration between AWS SMS and AWS Migration Hub, your SMS server catalog will be also visible on Migration Hub. For more information, see Importing Applications from Migration Hub (p. 37).

Tasks

Replicate a Server

AWS SMS automatically replicates live server volumes to AWS and creates an Amazon Machine Image (AMI) as needed.

To replicate a server

1. Install the Server Migration Connector as described in Install the Server Migration Connector (p. 11).
2. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.

   Tip
   If this link takes you to the AWS SMS page, trim the "gettingStarted" from the end of the URL and press return.
3. In the navigation menu, choose **Connectors**. Verify that the connector that you deployed in your VMware environment is shown with a status of healthy.
4. If you have not yet imported a catalog, choose **Servers**, **Import server catalog**. To reflect new servers added in your VMware environment after your previous import operation, choose **Re-import server catalog**. This process can take up to a minute.
5. Select a server to replicate and choose **Create replication job**.
6. On the **Configure server-specific settings** page, in the **License type** column, select the license type for AMIs to be created from the replication job. Linux servers can only use Bring Your Own License (BYOL). Windows servers can use either an AWS-provided license or BYOL. You can also choose **Auto** to allow AWS SMS to select the appropriate license. Choose **Next**.
7. On the **Configure replication job settings** page, the following settings are available:

   - For **Replication job type**, choose a value. The **replicate server every *interval*** option creates a repeating replication process that creates new AMIs at the interval you provide from the menu. The **One-time migration** option triggers a single replication of your server without scheduling repeating replications.

- For **Start replication run**, configure your replication run to start either immediately or at a later date and time up to 30 days in the future. The date and time settings refer to your browser's local time.
- For **IAM service role**, select **Allow automation role creation** to have AWS SMS create a service-linked role on your behalf. For more information, see Service-Linked Roles for AWS SMS (p. 43). Select **Use my own role** to specify an existing IAM role to use.
- (Optional) For **Description**, provide a description of the replication run.
- For **Enable automatic AMI deletion**, configure AWS SMS to delete older replication AMIs in excess of a number that you provide in the field.
- For **Enable AMI Encryption**, choose a value. If you choose **Yes**, AWS SMS encrypts the generated AMIs. Your default CMK is used unless you specify a non-default CMK. For more information, see Amazon EBS Encryption.
- For **Enable notifications**, choose a value. If you choose **Yes**, you can configure Amazon Simple Notification Service (Amazon SNS) to notify a list of recipients when the replication job has completed, failed, or been deleted. For more information, see What is Amazon Simple Notification Service?.
- For **Pause replication job on consecutive failures**, choose a value. The default is set to **Yes**. If the job encounters consecutive failures, it will be moved to the `PausedOnFailure` state and not marked `Failed` immediately.

  Note
  This option is not available for one-time replication jobs.

Choose **Next**.

8. On the **Review** page, review your settings. If the settings are correct, choose **Create**. To change the settings, choose **Previous**. After a replication job is set up, replication starts automatically at the specified time and interval.

In addition to your scheduled replication runs, you may also start up to two on-demand replication runs per 24-hour period. On the **Replication jobs** page, select a job and choose **Actions**, **Start replication run**. This starts a replication run that does not affect your scheduled replication runs, except in the case that the on-demand run is still ongoing at the time of your scheduled run. In this case, the scheduled run is skipped and rescheduled at the next interval. The same thing happens if a scheduled run is due while a previous scheduled run is still in progress.

Resume a Replication Job

AWS SMS can pause a replication job after the maximum number of consecutive scheduled replication jobs have failed. Before attempting to resume a job that is in the `PausedOnFailure` state, try to identify and fix the root cause of the replication run failure. For more information, see Replication Run Fails During the Preparing Stage (p. 53).

To resume a replication job that is paused

1. In the AWS SMS console, choose **Replication jobs**.
2. In the search bar, filter the jobs by `PausedOnFailure` to identify all paused jobs.
3. To resume a paused job, select the job and choose **Actions**, **Resume replication job**.

Monitor a Server Replication Job

You can manage and track the progress of each migration.

To monitor and modify server replication jobs

1. In the AWS SMS console, choose **Replication jobs**. You can view all replication jobs by scrolling through the table. In the search bar, you can filter the table contents on specific values.
2. Select a single replication job to view details about it in the lower pane. The **Job details** tab displays information about the current replication run, including the ID of the latest AMI created by the replication job. The **Run history** tab shows details about all of the replication runs for the selected replication job.
3. To change any job parameters, select a job on the **Replication jobs** page and choose **Actions**, **Edit replication job**. After entering new information in the **Edit configuration job** form, choose **Save** to commit your changes.

   Note
   You may need to refresh the page for the changes to become visible.

Delete a Replication Job

After you have finished replicating a server, you can delete the replication job. This stops the replication job and cleans up any artifacts created by the service (for example, the job's S3 bucket). This does not delete any AMIs created by runs of the stopped job. When you are done using a connector and no longer need it for any replication jobs, you can disassociate it from AWS SMS.

To shut down replication

1. Choose **Replication jobs**, select the desired job, choose **Actions**, and then choose **Delete replication jobs**. In the confirmation window, choose **Delete**.

   Note
   You may need to refresh the page for the changes to become visible.
2. To clear your server catalog after you no longer need it, choose **Servers**, **Clear server catalog**.
3. To disassociate a connector after you no longer need it, choose **Connectors** and select the connector. Choose **Disassociate** at the top-right corner of its information section and choose **Disassociate** again in the confirmation window.

Replicate VMs Using AWS SMS on the Command Line

You can use the AWS Command Line Interface (AWS CLI) to inventory and migrate your on-premises servers to Amazon EC2.

Prerequisites

- Install the Server Migration Connector as described in Install the Server Migration Connector (p. 11).
- If you have not used the AWS SMS console to start a replication job, you must use the following create-service-linked-role command to create the required service-linked role.

  ```
  aws iam create-service-linked-role --aws-service-name sms.amazonaws.com
  ```

  For more information, see Service-Linked Roles for AWS SMS (p. 43).

To replicate a server using the CLI

1. Use the get-connectors command to obtain a list of connectors that are registered to you.

   ```
   aws sms get-connectors
   ```

2. After a connector has been installed and registered through the console, use the import-server-catalog command to create an inventory of your servers. This process can take up to a minute.

   ```
   aws sms import-server-catalog
   ```

3. Use the get-servers command to display a list of servers available for import to Amazon EC2.

   ```
   aws sms get-servers
   ```

   The output should be similar to the following:

   ```
   {
       "serverList": [
           {
               "serverId": "s-12345678",
               "serverType": "VIRTUAL_MACHINE",
               "vmServer": {
                   "vmManagerName": "vcenter.yourcompany.com",
                   "vmServerAddress": {
                       "vmManagerId": "your-vcenter-instance-uuid",
                       "vmId": "vm-123"
                   },
                   "vmName": "your-linux-vm",
                   "vmPath": "/Datacenters/DC1/vm/VM Folder Path/your-linux-vm",
                   "vmManagerType": "vSphere"
               }
           },
           {
               "replicationJobTerminated": false,
               "serverId": "s-23456789",
               "serverType": "VIRTUAL_MACHINE",
               "replicationJobId": "sms-job-12345678",
   ```

```
                "vmServer": {
                    "vmManagerName": "vcenter.yourcompany.com",
                    "vmServerAddress": {
                        "vmManagerId": "your-vcenter-instance-uuid",
                        "vmId": "vm-234"
                    },
                    "vmName": "Your Windows VM",
                    "vmPath": "/Datacenters/DC1/vm/VM Folder Path/Your Windows VM",
                    "vmManagerType": "vSphere"
                }
            }
        ]
}
```

If you have not yet imported a server catalog, you see output similar to the following:

```
{
    "lastModifiedOn": 1477006131.856,
    "serverCatalogStatus": "NOT IMPORTED",
    "serverList": []
}
```

A catalog status of DELETED or EXPIRED also shows that no servers exist in the catalog.

4. Select a server to replicate, note the server ID, and specify the ID in the create-replication-job command.

```
aws sms create-replication-job --server-id s-12345678 --frequency 12 --seed-
replication-time 2016-10-24T15:30:00-07:00
```

   After the replication job is set up, it starts replicating automatically at the time specified with the `--seed-replication-time` parameter, expressed in seconds of the Unix epoch or according to ISO 8601. For more information, see Specifying Parameter Values for the AWS Command Line Interface. Thereafter, the replication repeats with an interval specified by the `--frequency` parameter, expressed in hours.

5. You can view details of all running replication jobs using the get-replication-jobs command. If you do not specify any parameters, the command lists all your replication jobs.

   This command returns output similar to the following:

```
{
    "replicationJobList": [
        {
            "vmServer": {
                "vmManagerName": "vcenter.yourcompany.com",
                "vmServerAddress": {
                    "vmManagerId": "your-vcenter-instance-uuid",
                    "vmId": "vm-1234"
                },
                "vmName": "VM name in vCenter",
                "vmPath": "/Datacenters/DC1/vm/VM Folder Path/VM name in vCenter"
            },
            "replicationRunList": [
                {
                    "scheduledStartTime": 1487007010.0,
                    "state": "Deleted",
                    "type": "Automatic",
                    "statusMessage": "Uploading",
                    "replicationRunId": "sms-run-12345678"
                }
            ],
            "replicationJobId": "sms-job-98765432",
            "state": "Deleted",
            "frequency": 12,
            "seedReplicationTime": 1477007049.0,
            "roleName": "sms"
        },
        {
            "vmServer": {
                "vmManagerName": "vcenter.yourcompany.com",
                "vmServerAddress": {
                    "vmManagerId": "your-vcenter-instance-uuid",
                    "vmId": "vm-2345"
                },
                "vmName": "win2k12",
                "vmPath": "/Datacenters/DC1/vm/VM Folder Path/win2k12"
            },
            "replicationRunList": [
                {
                    "scheduledStartTime": 1477008789.0,
                    "state": "Active",
                    "type": "Automatic",
                    "statusMessage": "Converting",
                    "replicationRunId": "sms-run-12345679"
                }
            ],
            "replicationJobId": "sms-job-23456789",
            "state": "Active",
            "frequency": 24,
            "seedReplicationTime": 1477008789.0,
            "roleName": "sms"
        }
    ]
}
```

6. You can also use the get-replication-runs command to retrieve information about all replication runs for a specific replication job. To do this, specify a replication job ID as follows:

```
aws sms get-replication-runs --replication-job-id sms-job-12345678
```

This command returns a list of all replication runs for the specified replication job, as well as details for that replication job, similar to the following:

```
{
    "replicationRunList": [
        {
            "scheduledStartTime": 1477310423.0,
            "state": "Active",
            "type": "Automatic",
            "statusMessage": "Converting",
            "replicationRunId": "sms-run-23456789"
        },
        {
            "amiId": "ami-abcdefab",
            "state": "Completed",
            "completedTime": 1477227683.652,
            "scheduledStartTime": 1477224023.0,
            "replicationRunId": "sms-run-34567890",
            "type": "Automatic",
            "statusMessage": "Completed"
        },
        {
            "amiId": "ami-efababcd",
            "state": "Completed",
            "completedTime": 1477144823.486,
            "scheduledStartTime": 1477137623.0,
            "replicationRunId": "sms-run-45678903",
            "type": "Automatic",
            "statusMessage": "Completed"
        }
    ]
}
```

7. To change any of the parameters of a replication job after you have created it, use the update-replication-job command, by providing the replication job ID and any parameters to change.

   ```
   aws sms update-replication-job --replication-job-id sms-job-12345678 --frequency 24 --
   next-replication-run-start-time 2016-10-24T15:30:00-07:00
   ```

8. In addition to your scheduled replication runs, you may also start up to two on-demand replication runs per 24-hour period. To do this, use the start-on-demand-replication-run command, which starts a replication run immediately. If another replication run is currently active, an on-demand replication run cannot be started.

   ```
   aws sms start-on-demand-replication-run --replication-job-id sms-job-12345678
   ```

   If a scheduled replication run is expected to start while an on-demand replication run is ongoing, then the scheduled run is skipped and rescheduled for the next interval.

9. After you are finished replicating a server, you may stop the replication job using the delete-replication-job command. This stops the replication job and cleans up any artifacts created by the service (for example, the job's S3 bucket). This does not delete any AMIs created by runs of the stopped job.

   ```
   aws sms delete-replication-job --replication-job-id sms-job-12345678
   ```

10. When you no longer need to maintain your catalog of servers, use the delete-server-catalog command to clear the catalog of servers maintained by the service.

    ```
    aws sms delete-server-catalog
    ```

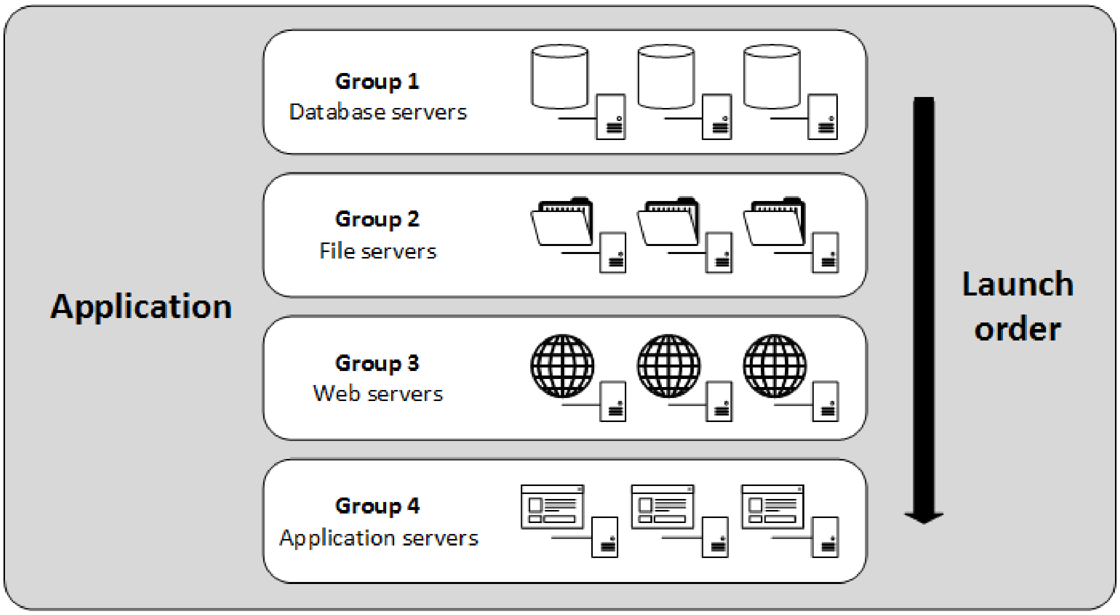
11. When you are done using a connector, use the disassociate-connector command to deregister the connector from AWS SMS. Call this command only after all replications using that connector are complete.

    ```
    aws sms disassociate-connector --connector-id c-12345678901234567
    ```

Migrate Applications Using AWS SMS

AWS Server Migration Service supports the automated migration of multi-server application stacks from your on-premises data center to Amazon EC2. Where server migration is accomplished by replicating a single server as an Amazon Machine Image (AMI), application migration replicates all of the servers in an application as AMIs and generates an AWS CloudFormation template to launch them in a coordinated fashion.

Applications can be further subdivided into groups that allow you to launch tiers of servers in a defined order. The following diagram provides a sample case of a database-backed web application:



In this example, the application is divided into four groups, each with three servers. The AWS CloudFormation template starts the servers in the following order: databases, file servers, web servers, and application servers.

After your servers are organized into applications and launch groups, you can specify a replication frequency, provide configuration scripts, and configure a target VPC in which to launch them. When you launch an application, AWS SMS configures it based on the generated template.

Application migration relies on the procedures for discovering on-premises resources described in Install the Server Migration Connector (p. 11). After you have imported a server catalog into AWS SMS using the Server Migration Connector, you can configure settings for applications, replication, and launch, as well as monitor migration status, in the **Applications** section of the AWS SMS console. You can also perform these tasks using the resources for AWS SMS in the AWS SMS API, AWS CLI, or AWS SDKs.

You can replicate your on-premises servers to AWS for up to 90 days per server. Usage time is calculated from the time a server replication begins until you terminate the replication job. After 90 days, your replication job is automatically terminated. You can request an extension from AWS Support.

Note
Application migration from Microsoft Azure environments is supported, but the Server Migration Connector for Azure does not currently guarantee the closeness of the server snapshots in the application.

Using Application Migration

This section provides step-by-step procedures for creating, configuring, replicating, and launching applications.

To create an application

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view your existing applications (if any).
3. Choose **Create new application**.
4. On the **Create new application** page, under **Application settings**, supply the following information and then choose **Next**:
   - **Application name** — Specify a name for the application.
   - **Application description** — Optionally, specify a description for the application.
   - **Role name** — Select **Allow automation role creation** to have AWS SMS create a service-linked role on your behalf. For more information, see Service-Linked Roles for AWS SMS (p. 43). Select **Use my own role** to specify an existing IAM role to use.
5. Under **Select servers**, select the available servers to include in the application. In the search bar, you can filter the table contents on specific values. Choose **Next: Add servers to groups**.

   Note
   Ungrouped servers are added to a default group.
6. Under **Add servers to groups**, you can create groups, delete groups, add selected servers from your application to groups, and remove servers from groups.

   Complete the following steps to add one or more servers to a new group:

   a. Select the servers to be added to the new group.
   b. Choose **Add servers to group**.
   c. Under **Add servers to group**, choose **Add to new group** and provide a name for the group.
   d. Choose **Add**. The list of servers now displays the associated group for each server that you selected.
7. After creating one or more groups, you can delete a group by completing the following steps:

   a. Choose **Delete group**.
   b. For **Group to delete**, choose a group.
   c. Choose **Delete**.

   Deleting a group has no effect on servers that belong to it.
8. Under **Add tags**, tag your applications with key/value pairs that propagate to all of the servers created when the application is launched. Choose **Next**.
9. Under **Review**, you can review and edit your application and group settings. When you are satisfied that the settings are correct, choose **Create**. From the status page, you can proceed directly to **Configure replication settings**.

To configure replication settings for an application

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view the available applications.
3. Select the name of the application to configure.
4. Choose **Actions**, **Configure replication settings**.

5. On the **Replication settings** page, provide the following information and then choose **Next**:

   - **Replication job type** — Specify the replication period (1-24 hours) or choose **One-time replication**.
   - **Start replication run** — Choose to start a replication run immediately, or choose **At a later time and date** and enter the information.
   - **Enable automatic AMI deletion** — Choose **Yes** or **No**.
6. The **Server-specific settings** page displays the application servers and their group memberships. You can edit the following server settings individually or together:

   - **License type** — Choose **Auto**, **AWS**, or **BYOL**.
   - **Quiesce** — Before taking a snapshot of the VM, halt data input/output and store the system memory state (for VMware).
7. Choose **Next**.
8. Review the replication settings and choose **Save**. From the status page, you can proceed directly to **Configure launch settings**.

To configure launch settings for an application

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view the available applications.
3. Select the name of the application to configure.
4. Choose **Actions**, **Configure launch settings**.
5. On the **Configure launch settings** page, for **IAM CloudFormation role**, specify a non-default value. Under **Specify launch order**, configure a launch order for your groups. Choose **Next**.
6. Under **Configure launch settings** for the application, you can edit the following server settings individually or multiply:

   - **Logical ID** — AWS CloudFormation resource ID. This parameter is used as 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. This field is required.
   - **Key pair** — Specifies the SSH key pair that gives access to the server. This field is required.

     Note
     If the logged-in user does not have IAM permissions to list key pairs, this list will be empty.
   - **Configuration script** — A script to run configuration commands at startup of EC2 instances launched as part of an application.

   Choose **Next**.
7. Under **Configure target network and security** settings for the application, you can edit the following server settings individually or multiply. Network settings require prior setup as described in RunInstances.

   Note
   If the logged-in user does not have IAM permissions to list VPCs, subnets, security groups, these lists will be empty.

   - **VPC** — VPC in which to deploy the application. This field is required.
   - **Subnet** — Subnet in which to deploy the application. This field is required.

- **Security Group** — Security group membership for the application. This field is required.
- **Publicly Accessible** — Whether the application should be accessible from the internet.

Choose **Next**.

8. Review the launch configuration settings and choose **Save**.

To start replicating an application

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view the available applications.
3. Choose the name of the application to replicate.
4. On the application details page, choose **Actions**, **Start replication**.
5. In the **Start replication** window, choose **Start**. Replication can take anywhere from 30 minutes to several days depending on the disk size. On the application details page, you can observe the status of the replication in the **Replication status** field. If the replication fails, you may be able to find the reason in the **Replication status message** field.

To launch an application

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view the available applications.
3. Choose the name of the application to launch.
4. On the application details page, choose **Actions**, **Launch application**. A replication job must complete before you perform this action.
5. 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 may be able to find the reason in the **Launch status message** field.

To generate an AWS CloudFormation template for the application

Use the following procedure if you want to examine the AWS CloudFormation template that is auto-generated when you launch the application.

1. Open the AWS SMS console at https://console.aws.amazon.com/servermigration/.
2. Choose **Applications**. On the **Applications** page, you can view the available applications.
3. Choose the name of the application for which to create a template.
4. On the application details page, choose **Actions**, **Generate template**. A replication job must complete before you perform this action.
5. In the **Generate template** window, choose **Generate**.

Importing Applications from Migration Hub

Application Migration supports the import and migration of applications discovered by AWS Migration Hub.

To import applications from Migration Hub

1. To enable application catalog import, complete the AWS Server Migration Service (SMS) instructions in the Migration Hub user guide.

> **Note**
> Taking this action exports the SMS server catalog and makes it visible on Migration Hub.

2. In the SMS console, on the **Applications** page, choose **Import applications**.
3. In the **Import applications** window, you can optionally provide a value in the **Role name** field. If no role name is specified, the default role name *sms* is used. Choose **Import**.

   > **Note**
   > SMS imports application-related servers from Migration Hub only if they exist in the SMS Server Catalog and are not part of an existing SMS application. As a result, some applications may be only partially imported.
4. After import completes, the applications imported from Migration Hub appear in the **Applications** table. Imported applications can be migrated but cannot be edited in SMS. They can, however, be edited in Migration Hub. After editing, re-import.

   > **Note**
   > An application cannot be re-imported if it is being actively replicated or launched by SMS. If this conflict occurs, stop the replication or launch and re-import.

Using Amazon CloudWatch Events and AWS Lambda with AWS SMS

You can use Amazon CloudWatch Events with AWS Server Migration Service to automate actions based on your migration workflow. This requires you to create an IAM policy for Lambda to assume, a Lambda function to handle the event, and a CloudWatch Events rule that matches incoming events and routes them to the Lambda function.

Handling CloudWatch Events Rules for AWS SMS

The following procedure uses an AWS Lambda function to monitor AWS SMS job state changes and launches an Amazon EC2 instance whenever an AMI ID has been created.

To create a Lambda function that monitors job state changes

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. Create an IAM policy to provide permissions to execute an action (called by Lambda) and to write to the CloudWatch log when invoked by CloudWatch Events. The following example provides permissions to execute a `RunInstances` action. Assign the policy to the IAM role of the user that will handle the CloudWatch event.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:PutLogEvents"
         ],
         "Resource":"arn:aws:logs:*:*:*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:RunInstances"
         ],
         "Resource":"*"
      }
   ]
}
```

3. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
4. Choose **Create function**.
5. To ensure that your Lambda function is available from the CloudWatch console, create it in the region where the CloudWatch event will occur. For more information, see the AWS Lambda Developer Guide. Name the function `LaunchInstanceFromAMI` and select **Python 2.7** as the runtime.
6. For **Role**, select **Choose an existing role**. Under **Existing role**, in the list of available roles, choose the role to which you added your policy.

7. Choose **Create function** and define a Lambda function similar to the one below. This sample function, written in Python 2.7, is invoked by CloudWatch Events when an AWS SMS job completion sends an event with an AMI ID. When invoked, it launches a `t2.micro` instance in the region of the event.

```
# Sample Lambda function to launch an EC2 instance from all AMI ID's created from a
# Server Migration Service replication job

import boto3

# main function
def lambda_handler(event, context):

    # create an ec2 client
    ec2 = boto3.client('ec2', region_name=event['region'])

    # match any event that returns an ami-id
    if 'ami-id' in event['detail']:
        imageId = event['detail']['ami-id']

        # launch instance from the AMI ID
        ec2.run_instances(
            ImageId=imageId,
            MaxCount=123,
            MinCount=1,
            InstanceType='t2.micro'
        )
        print 'launched instance with ami id: ' + imageId
    else:
        print 'did not launch instance'
```

8. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
9. Choose **Events**, **Create rule**. For **Service Name**, choose **Server Migration Service (SMS)**. For **Event Type**, choose **Server Migration Job State Change**.
10. Choose **Target**, **Add Target**.
11. For **Lambda function**, select the Lambda function that you previously created and choose **Configure details**.
12. On the **Configure rule details** page, type values for **Name** and **Description**. Select the **State** check box to activate the function (setting it to **Enabled**).
13. Choose **Create rule**.

Your rule should now appear on the **Rules** tab. In the example shown, the configured event should launch an EC2 instance each time that you receive an AMI ID.

Logging AWS Server Migration Service API Calls with AWS CloudTrail

AWS Server Migration Service is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS SMS. CloudTrail captures all API calls for AWS SMS as events. The calls captured include calls from the AWS SMS console and code calls to the AWS SMS API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS SMS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS SMS, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information, see the AWS CloudTrail User Guide.

AWS SMS Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS SMS, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS SMS, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All AWS SMS actions are logged by CloudTrail and are documented in the AWS SMS API Reference. For example, calls to the `CreateReplicationJob` , `GetConnectors`, and `ImportServerCatalog` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding AWS SMS Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateReplicationJob` action.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "0123456789abcdef01234",
        "arn": "arn:aws:iam::0123456789ab:user/sms-user",
        "accountId": "0123456789ab",
        "accessKeyId": "0123456789abcdef0123",
        "userName": "sms-user"
    },
    "eventTime": "2018-09-04T16:34:49Z",
    "eventSource": "sms.amazonaws.com",
    "eventName": "CreateReplicationJob",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "1.2.3.4",
    "userAgent": "aws-sdk-java/example-sdk-version Linux/example-kernel-version …",
    "requestParameters": {
        "roleName": "sms",
        "serverId": "s-01234567",
        "runOnce": true,
        "seedReplicationTime": "Sep 4, 2018 4:36:48 PM"
    },
    "responseElements": {
        "replicationJobId": "sms-job-012345677"
    },
    "requestID": "00000000-1111-2222-3333-444444444444",
    "eventID": "55555555-6666-7777-8888-999999999999",
    "eventType": "AwsApiCall",
    "recipientAccountId": "0123456789ab"
}
```

Service-Linked Roles for AWS SMS

AWS SMS uses a service-linked role for the permissions that it requires to call other AWS services on your behalf. For more information, see Using Service-Linked Roles in the *IAM User Guide*.

Before the introduction of a service-linked role for AWS SMS, you were required to create two IAM roles to grant AWS SMS the permissions that it needs. These roles are no longer required to use AWS SMS. However, they are documented here for completeness. For more information, see Legacy IAM Roles for AWS SMS (p. 46).

Permissions Granted by the Service-Linked Role

AWS SMS uses the service-linked role named **AWSServiceRoleForSMS** to enable AWS SMS to manage your replication jobs.

AWSServiceRoleForSMS trusts the `sms.amazonaws.com` service principal to assume the role.

The role permissions policy allows AWS SMS to complete actions on resources as follows:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudformation:CreateChangeSet",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:ExecuteChangeSet"
            ],
            "Resource": "arn:aws:cloudformation:*:*:stack/sms-app-*/*",
            "Effect": "Allow",
            "Condition": {
                "ForAllValues:StringLikeIfExists": {
                    "cloudformation:ResourceTypes": [
                        "AWS::EC2::*"
                    ]
                }
            }
        },
        {
            "Action": [
                "cloudformation:DeleteChangeSet",
                "cloudformation:DescribeChangeSet",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStackResources",
                "cloudformation:GetTemplate"
            ],
            "Resource": "arn:aws:cloudformation:*:*:stack/sms-app-*/*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:ValidateTemplate",
                "cloudformation:DescribeStackResource",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutLifecycleConfiguration"
            ],
            "Resource": "arn:aws:s3:::sms-app-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "sms:CreateReplicationJob",
                "sms:DeleteReplicationJob",
                "sms:GetReplicationJobs",
                "sms:GetReplicationRuns",
                "sms:GetServers",
                "sms:ImportServerCatalog",
                "sms:StartOnDemandReplicationRun",
                "sms:UpdateReplicationJob"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "ec2:ModifySnapshotAttribute",
                "ec2:CopySnapshot",
                "ec2:CopyImage",
                "ec2:Describe*",
                "ec2:DeleteSnapshot",
                "ec2:DeregisterImage",
                "ec2:RunInstances"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*"
        },
        {
            "Action": "iam:GetRole",
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": "iam:PassRole",
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringLike": {
                    "iam:AssociatedResourceArn": "arn:aws:cloudformation:*:*:stack/sms-app-
*/*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyInstanceAttribute",
                "ec2:StopInstances",
                "ec2:StartInstances",
                "ec2:TerminateInstances"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-id":
 "arn:aws:cloudformation:*:*:stack/sms-app-*/*"
                }
            }
        }
    ]
}
```

Create the Service-Linked Role

You don't need to manually create the **AWSServiceRoleSMS** role. AWS SMS creates this role for you when you select the **Allow automatic role creation** option when creating or updating a replication job, application, or launch configuration using the AWS Management Console.

For AWS SMS to create a service-linked role on your behalf, you must have the required permissions. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

If you do not use the console, you can create this service-linked role manually. For example, use the following AWS CLI create-service-linked-role command to create **AWSServiceRoleSMS**.

```
aws iam create-service-linked-role --aws-service-name sms.amazonaws.com
```

Edit the Service-Linked Role

You can edit the description of **AWSServiceRoleForSMS** using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Delete the Service-Linked Role

If you no longer need to use AWS SMS, we recommend that you delete the **AWSServiceRoleForSMS** role. The service-linked role can only be deleted in the following conditions:

- The service-linked role is not being used by an active replication job
- The service-linked role is not being used by an application that has an associated active replication job
- The service-linked role is not being used by an application that has an associated AWS CloudFormation stack

You can use the IAM console, the IAM CLI, or the IAM API to delete service-linked roles. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

After you delete the **AWSServiceRoleForSMS** role, AWS SMS creates the role again if you start a replication job.

Legacy IAM Roles for AWS SMS

Before the introduction of **AWSServiceRoleForSMS**, you would have been required to create a service role and a launch role to grant AWS SMS the permissions that it needs. It is no longer necessary for you to create these roles.

Configure a Service Role for AWS SMS

Use one of the following procedures to create an IAM role that grants permissions to AWS SMS to place migrated resources into your Amazon EC2 account. In AWS Regions that make an IAM role template available, Option 1 works. If you find that no template for **AWS Server Migration Service** exists in your AWS Region, proceed to Option 2.

Option 1: Create an AWS SMS IAM role with a template

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, **Create role**.
3. Under **Choose the service that will use this role**, choose **SMS**, **Next: Permissions**.
4. Under **Attached permissions policies**, confirm that the policy **ServerMigrationServiceRole** is visible and choose **Next: Review**.
5. Under **Review**, for **Role name**, enter **`sms`**.

   Note
   Alternatively, you can apply a different name. However, you must then specify the role name explicitly each time that you create a replication job or an application.
6. Choose **Create role**. You should now see the **sms** role in the list of available roles.

Use the following option in AWS Regions that do not make an IAM role template available. This option also works as a manual alternative to Option 1 in all Regions.

Option 2: Create an AWS SMS IAM role manually

1. Create a local file named `trust-policy.json` with the following content:

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "sms.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "sms"
                    }
                }
            }
        ]
    }
```

2. Create a local file named `role-policy.json` with the following content:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "cloudformation:CreateChangeSet",
            "cloudformation:CreateStack",
            "cloudformation:DeleteStack",
            "cloudformation:ExecuteChangeSet"
         ],
         "Resource":"arn:aws:cloudformation:*:*:stack/sms-app-*/*",
         "Effect":"Allow",
         "Condition":{
            "ForAllValues:StringLikeIfExists":{
               "cloudformation:ResourceTypes":[
                  "AWS::EC2::*"
               ]
            }
         }
      },
      {
         "Action":[
            "cloudformation:DeleteChangeSet",
            "cloudformation:DescribeChangeSet",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStackResources",
            "cloudformation:GetTemplate"
         ],
         "Resource":"arn:aws:cloudformation:*:*:stack/sms-app-*/*",
         "Effect":"Allow"
      },
      {
         "Action":[
            "cloudformation:DescribeStacks",
            "cloudformation:ValidateTemplate",
            "cloudformation:DescribeStackResource",
            "s3:ListAllMyBuckets"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Action":[
            "s3:CreateBucket",
            "s3:DeleteBucket",
            "s3:DeleteObject",
            "s3:GetBucketAcl",
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutLifecycleConfiguration",
            "s3:ListAllMyBuckets"
         ],
         "Resource":"arn:aws:s3:::sms-app-*",
         "Effect":"Allow"
      },
      {
         "Action":[
            "sms:CreateReplicationJob",
            "sms:DeleteReplicationJob",
            "sms:GetReplicationJobs",
            "sms:GetReplicationRuns",
            "sms:GetServers",
            "sms:ImportServerCatalog",
            "sms:StartOnDemandReplicationRun",
            "sms:UpdateReplicationJob"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Action":[
            "ec2:ModifySnapshotAttribute",
            "ec2:CopySnapshot",
            "ec2:CopyImage",
            "ec2:Describe*",
            "ec2:DeleteSnapshot",
            "ec2:DeregisterImage",
            "ec2:CreateTags",
            "ec2:DeleteTags"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Action":"iam:GetRole",
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Action":"iam:PassRole",
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "iam:AssociatedResourceArn":"arn:aws:cloudformation:*:*:stack/sms-app-*/
*"
            }
         }
      }
   ]
}
```

3. At a command prompt, go to the directory where you stored the two JSON policy files, and run the following commands to create the AWS SMS service role:

```
aws iam create-role --role-name sms --assume-role-policy-document file://trust-
policy.json
aws iam put-role-policy --role-name sms --policy-name sms --policy-document
 file://role-policy.json
```

Note
Your AWS CLI user must have permissions on IAM. You can grant these by attaching the `IAMFullAccess` managed policy to your AWS CLI user. For information about managing IAM users and permissions, see Creating an IAM User in Your AWS Account.

Configure a Launch Role for AWS SMS

If you plan to launch applications, you need an AWS SMS launch role. You assign this role using the `PutAppLaunchConfiguration` API. When the `LaunchApp` API is called, the role is used by AWS CloudFormation.

Use one of the following procedures to configure this role. Use Option 2 in AWS Regions that do not make an AWS SMS launch role template available, or as a manual alternative to Option 1 in all Regions.

Option 1: Create an AWS SMS launch role with a template

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, **Create role**.
3. Under **Choose the service that will use this role**, choose **CloudFormation**, **Next: Permissions**.
4. Under **Attached permissions policies**, confirm that the policy **ServerMigrationServiceLaunchRole** is visible and choose **Next: Review**.
5. Under **Review**, for **Role name**, enter **`sms-launch`**.

   Note
   Alternatively, you can apply a different name. However, you must then specify the role name explicitly each time that you create a launch configuration for an application.
6. Choose **Create role**. You should now see the **sms-launch** role in the list of available roles.

Option 2: Create an AWS SMS launch role manually

1. Create a local file named `trust-policy.json` with the following content:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"cloudformation.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

2. Create a local file named `role-policy.json` with the following content:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:ModifyInstanceAttribute",
            "ec2:StopInstances",
            "ec2:StartInstances",
            "ec2:TerminateInstances"
         ],
         "Resource":"*",
         "Condition":{
            "ForAllValues:StringLike":{
               "ec2:ResourceTag/aws:cloudformation:stack-
id":"arn:aws:cloudformation:*:*:stack/sms-app-*/*"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:*:*:instance/*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:RunInstances",
            "ec2:Describe*"
         ],
         "Resource":"*"
      }
   ]
}
```

3. At a command prompt, go to the directory where you stored the two JSON policy files, and run the following commands to create the AWS SMS service role:

```
aws iam create-role --role-name sms-launch --assume-role-policy-document file://trust-policy.json
aws iam put-role-policy --role-name sms-launch --policy-name sms --policy-document file://role-policy.json
```

Troubleshooting AWS SMS

The following information can help you troubleshoot issues with errors that you might encounter when using AWS SMS. Before using these procedures, confirm that your SMS setup and the server you are trying to migrate meet the requirements in Server Migration Service (SMS) Requirements (p. 2).

Issues

Certificate Error When Uploading a VM to Amazon S3

The connector may fail to replicate your VM because the VM is on an ESXi host with an SSL certificate problem. If this occurs, you see the following error message displayed in the **Latest run's status message** section: "ServerError: Failed to upload base disk(s) to S3. Please try again. If this problem persists, please contact AWS Support: vSphere certificate hostname mismatch: Certificate for <*somehost.somedomain.com*> doesn't match any of the subject alternative names: [*localhost.localdomain*]."

You can override this ESXi host certificate problem by completing the following tasks:

Tasks

Upgrade Your Connector

This section is for customers who are manually upgrading the connector. If you have previously configured automatic upgrades, skip these steps and continue to Re-Register Your Connector (p. 51).

To upgrade your connector

1. Open the connector console.
2. Log in to the connector.
3. Choose **Upgrade**.
4. Wait for the connector to finish upgrading to version 1.0.11.13 or later.

Re-Register Your Connector

This section applies to all customers encountering the certificate mismatch problem.

To re-register your connector

1. Open the connector console.
2. Log in to the connector.
3. In the **General Health** section, check that the connector version is 1.0.11.13 or later.
4. Choose **Edit AWS Server Migration Service Settings**.
5. On the **Setup** page, for **AWS Region**, select the desired region from the list. For **AWS Credentials**, enter the IAM access key and secret key that you created in Step 2 of the setup guide (p. 11). Choose **Next**.
6. On the **vCenter Service Account** page, enter the vCenter hostname, user name, and password that you created in Step 3 of the setup guide (p. 11).
7. Select the **Ignore hostname mismatch and expiration errors for vCenter and ESXi certificates** check box. Choose **Next**.
8. Complete registration and view the connector configuration dashboard.
9. In the AWS SMS console, delete and restart your stuck replication jobs.

Server Migration Connector Fails To Connect To AWS with Error "PKIX path building failed"

In some customer environments, secure network traffic is proxied through a certificate re-signing mechanism for auditing and management purposes. This can cause your AWS credentials to fail when the connector attempts to contact AWS SMS. The error message contains "PKIX path building failed," indicating that an invalid certificate was presented.

For the connector to work in such an environment, the re-signing certificate (a user certificate that your organization trusts and uses to sign outbound packets) must be added to the connector's trust store, as described in the following steps.

To add the re-signing certificate to the connector trust store

1. On your connector system, disable the FreeBSD packet filter and enable SSH with the following commands:

   ```
   sudo service pf stop
   sudo service sshd onestart
   ```
2. Copy your user certificate to the connector by a method such as the following:

   ```
   scp userCertFile ec2-user@10.0.0.100:/tmp/
   ```
3. Add the user certificate to the trust store:

   ```
   keytool -importcert -keystore /usr/local/amazon/connector/config/jetty/trustStore -
   storepass AwScOnNeCtOr -file /tmp/userCertFileName -alias userCertName
   ```
4. Restart services using the following command (part of AWS Management Portal for vCenter):

   ```
   sudo setup.rb
   ```

   Select option **3** and type "yes".
5. Re-enable the packet filter:

```
sudo service pf start
```

This CA Root certificate is not trusted

When you access the IP address of a virtual machine that you installed on-premises, you may receive the following message:

```
This CA Root certificate is not trusted. To enable trust,
install this certificate in the Trusted Root Certifications
Authorities store.
```

You can safely ignore this message.

Replication Run Fails During the Preparing Stage

In some cases, AWS SMS allows a replication job to continue scheduling incremental replication runs even when the latest replication run has failed. When the maximum allowed number of consecutive failures is reached, the default behavior for a replication job is to be paused. The job can be resumed within four days, after which it is deleted. In such cases, the Amazon EBS snapshots from the latest replication run are shared with the customer account, and a status message for the failed replication run is sent. The message contains the snapshot IDs and states the reason for the failure. A typical status message resembles the following:

```
EBS snapshot(s) created with snapshot ID(s): snap-12345678abcdefgh. Another run
has been scheduled after the last run failed due to an import failure. 2 re-try run(s)
remaining before the job will be failed.
```

The reason for replication-run failures (including first-boot failures) often correlates closely with failures observed when Amazon EC2 VM Import/Export is used for VM migrations. For more information, see Troubleshooting VM Import/Export.

If you need further help with resolving a problem, contact AWS Support. EBS snapshots generated during a failed migration are shared with your account, and the snapshot IDs are included in the status message for the replication job. Be sure to have these details available when you contact AWS Support.

Replicated AMI Doesn't Support Some Instance Types for Launch

Some instances require ENA support. If the migration does not enable ENA support, then the replicated AMI does not allow you to launch instances that require ENA support.

Verify that ENA is enabled. For more information, see Enabling Enhanced Networking on Windows or Enabling Enhanced Networking on Linux in the Amazon EC2 documentation.

Failure to upload to Amazon S3

The following error message indicates that the vCenter cannot export the VMDK.

```
Error message: ServerError: Failed to upload base disk(s) to S3. Please try again.
If this problem persists, please contact AWS support: java.io.IOException: Server
returned HTTP response code: 500 for URL: https://<url>
```

Verify that the VMDK is snapshottable and that the VM does not have any mounted ISOs.

Incremental Replication Delta Exceeds 1 TB

The connector is designed to handle frequent replication with small deltas. The connector does not support deltas larger than 1 TB. If you do not replicate on a regular basis, the delta can exceed this limit and the replication run fails.

To prevent this issue, set up frequent incremental replication runs. If you cannot replicate frequently, you can increase the delta upload limit. For example, run the following commands on the connector to increase the part size of S3 uploads from 25 MB to 100 MB. When prompted, select option 3.

```
sudo sms-connector-config -set slotSizeMB 100
sudo setup.rb
```

Increasing the upload limit impacts the performance and memory usages of the connector. Do not increase the upload limit while the connector is uploading multiple deltas.

Release Notes for Server Migration Connector

The following tables describe the release history of the Server Migration Connector.

Releases

Releases for vCenter Environments

To download the latest connector for vCenter environments, open https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector.ova.

Release date	Version	Comment
May 29, 2019	1.0.13.106	• Fixed a bug that was blocking registration of the connector appliance due to connectivity errors with AWS
May 3, 2019	1.0.13.90	• Fixed a bug that was blocking migrations in the AWS GovCloud (US-East) Region
December 12, 2018	1.0.13.15	• Added support for the EU (Stockholm) Region
December 5, 2018	1.0.13.1	• Connector optimized for Application Migration feature
October 19, 2018	1.0.12.109	• Fix for "Premature EOF" caused by VM disk upload resumption after on-premises infrastructure or network disruptions
September 18, 2018	1.0.12.88	• Fixes to resume VM disk transfers interrupted by on-premises network outages
June 11, 2018	1.0.12.3	• Added support for VMs with disk-size larger than 4 TB using the S3 Manifest functionality • Minor bug fixes

Release date	Version	Comment
April 26, 2018	1.0.11.34	• Added support for the AWS Region South America (São Paulo) • Minor bug fixes and performance improvements
January 29, 2018	1.0.10.x	• Added support for the following AWS Regions: EU (London), EU (Paris), US West (N. California), and China (Beijing) • Minor bug fixes and performance improvements
November 08, 2017	1.0.9.x	• Improved resilience in disk uploads • Bug fixes and other performance improvements
August 29, 2017	1.0.8.x	• Added French, Chinese, Korean and Japanese language support • Improved VM disk upload speeds • Minor bug fixes
June 02, 2017	1.0.7.12	• Added support for the AWS GovCloud (US-West) Region
May 5, 2017	1.0.5.2	• Added support for vCenter 5.1 • Added support for one-time migration • Improved error messages and security-related bug fixes
Nov 3, 2016	1.0.0.84	• Server Migration Connector virtual appliance for VMware environments • AWS Server Migration Service console to manage VM migrations and SMS replication tasks using a graphical interface • AWS Server Migration Service CLI to manage VM migrations and SMS replication tasks using the command line

Releases for Hyper-V/SCVMM Environments

To download the latest connector for Hyper-V/SCVMM environments, open https://s3.amazonaws.com/sms-connector/AWS-SMS-Connector-for-SCVMM-HyperV.zip.

Release date	Version	Comment
December 12, 2018	1.1.0.378	• Added support for the EU (Stockholm) Region
December 5, 2018	1.1.0.364	• Connector optimized for Application Migration feature
October 9, 2018	1.1.0.357	• Windows Hyper-V Generation 2 VM migration • Minor bug fixes
June 11, 2018	1.1.0.304	• Added support for VMs with disk-size larger than 4 TB using the S3 Manifest functionality • Minor bug fixes
April 25, 2018	1.1.0.287	• Added support for migrating VMs from multiple Hyper-V servers using a single connector • Added support for the AWS Region South America (São Paulo) • Minor bug fixes
February 28, 2018	1.1.0.x	• Added support for the following AWS Regions: EU (London), EU (Paris), US West (N. California), and China (Beijing) • Minor bug fixes
December 14, 2017	1.1.0.76	• Added support for Microsoft’s Hyper-V environment

Releases for Azure Environments

To download the latest connector for Azure environments, open https://s3.amazonaws.com/sms-connector/aws-sms-azure-setup.ps1.

Release date	Version	Comment
May 31st, 2019	1.2.0.286	• Deployment script supports nondefault subscriptions, includes minor bug fixes • Connector bug fixes and performance improvements
April 18, 2019	1.2.0.269	• Added support for Microsoft's Azure environment

Document History for AWS SMS

The following table describes the releases of AWS SMS.

update-history-change	update-history-description	update-history-date
Application Migration (VMware, Hyper-V/SCVMM) (p. 58)	Added support for migrating groups of servers organized as applications, and or automated application launching using CloudFormation.	December 5, 2018
Larger disk size support (VMware, Hyper-V/SCVMM) (p. 58)	Added support for VMs with disk-size larger than 4 TB using the S3 Manifest functionality.	June 11, 2018
New region support (VMware) (p. 58)	Support for AWS region South America (Sao Paulo).	April 26, 2018
Migrate multiple servers with single connector; new region support (Hyper-V/SCVMM) ((p. 58)	Support for migrating VMs from multiple Hyper-V servers using a single connector. Support for AWS region South America (Sao Paulo).	April 25, 2018
New region support (Hyper-V/SCVMM) (p. 58)	Support for AWS regions EU (London), EU (Paris), US West (N. California), and China (Beijing).	February 28, 2018
New region support (VMware) (p. 58)	Support for AWS regions EU (London), EU (Paris), US West (N. California), and China (Beijing).	January 29, 2018
Hyper-V support (p. 58)	Support for Microsoft's Hyper-V environment.	December 14, 2017
Upload resilience (p. 58)	Improved resilience in disk uploads.	November 8, 2017
New languages; enhanced upload speeds (p. 58)	Added French, Chinese, Korean and Japanese language support. Improved VM disk upload speeds.	August 29, 2017
New region support (p. 58)	Support for AWS GovCloud (US) region.	June 2, 2017
vCenter 5.1; one-time migration; error messages; security (p. 58)	Support for vCenter 5.1. Support for one-time migration. Improved error messages and security-related bug fixes.	May 5, 2017
Initial release (p. 58)	Server Migration Connector virtual appliance for VMware environments. AWS Server Migration Service console to	November 3, 2016

	manage VM migrations and SMS replication tasks using a graphical interface. AWS Server Migration Service CLI to manage VM migrations and SMS replication tasks using the command line.	