aws

User Guide

VM Import/Export

VM Import/Export: User Guide

This documentation is a draft for private preview for regions in the AWS European Sovereign Cloud. Documentation content will continue to evolve. Published: January 14, 2026.

Table of Contents

What is VM Import/Export?

VM Import/Export enables you to import virtual machine (VM) images from your existing virtualization environment to Amazon EC2, and then export them back. This enables you to migrate applications and workloads to Amazon EC2, copy your VM image catalog to Amazon EC2, or create a repository of VM images for backup and disaster recovery. For more information, see VM Import/Export.

For more information about how to use VM Import/Export, see How to get started with VM Import/Export.

Topics

Benefits of VM Import/Export

You can use VM Import/Export to migrate applications and workloads, copy your VM image catalog, or create a disaster recovery repository for VM images.

Migrate existing applications and workloads to Amazon EC2

When you migrate your VM-based applications and workloads to Amazon EC2, you preserve their software and configuration settings. When you create an AMI from your VM, you can run multiple instances based on the same imported VM. You can also use the AMI to replicate your applications and workloads around the world using AMI copy. For more information, see Copying an AMI in the *Amazon EC2 User Guide*.

Import your VM image catalog to Amazon EC2

If you maintain a catalog of approved VM images, you can copy your image catalog to Amazon EC2 and create AMIs from the imported images. You can import your existing software, including products that you have installed such as anti-virus software, intrusion detection systems, and so on, along with your VM images. You can use the AMIs you create as your Amazon EC2 image catalog.

Create a disaster recovery repository for VM images

You can import your local VM images into Amazon EC2 for backup and disaster recovery purposes. You can import your VMs and store them as AMIs. The AMIs you create will be ready to launch in Amazon EC2 when you need them. If your local environment suffers an event, you can quickly launch your instances to preserve business continuity while simultaneously exporting them to rebuild your local infrastructure.

Features of VM Import/Export

VM Import provides the following features:

- The ability to import a VM from your virtualization environment to Amazon EC2 as an Amazon Machine Image (AMI). You can launch EC2 instances from your AMI any time.
- The ability to import a VM from your virtualization environment to Amazon EC2 as an EC2 instance. The instance is initially in a `stopped` state. You can create an AMI from the instance.
- The ability to export a VM that was previously imported from your virtualization environment.
- The ability to import disks as Amazon EBS snapshots.
- VM import supports ENA drivers for Linux. ENA support will be enabled only if the original VM has ENA and/or NVMe drivers installed. We recommend installing the latest drivers.

Pricing for VM Import/Export

With Amazon Web Services, you pay only for what you use. There is no additional fee to use VM Import/Export. You pay the standard fees for the Amazon Simple Storage Service (Amazon S3) bucket and EBS volumes used during the import and export processes, and for the EC2 instances that you run.

Related services

Consider the following services as you plan your migration to AWS:

- **AWS Application Discovery Service** – You can use the Application Discovery Service to gather information about your data center, such as server utilization data and dependency mappings, so that you can view information about your workloads. For more information, see the Application Discovery Service User Guide.

- **AWS Application Migration Service** – If you use VMware vSphere, Microsoft Hyper-V, or Microsoft Azure, you can use Application Migration Service to automate the migration of your virtual machines to AWS. For more information, see the Application Migration Service User Guide.

How to get started with VM Import/Export

First, you must decide whether you will import your VMs as AMIs or instances. To get started, read about how image import and instance import work. You can also read through the prerequisites and limitations of each method. For more information, see the following resources:

- How VM Import/Export works
- VM Import/Export Requirements
- Accessing VM Import/Export
- Import a VM to Amazon EC2 as an image using VM Import/Export
- Import a disk as an EBS snapshot using VM Import/Export

Accessing VM Import/Export

You can access VM Import/Export using the following interfaces.

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon EC2, see ec2 in the *AWS CLI Command Reference*.

AWS Tools for PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the AWS Tools for PowerShell User Guide. For more information about the Cmdlets for Amazon EC2, see the AWS Tools for PowerShell Cmdlet Reference.

Amazon EC2 API

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see Actions in the *Amazon EC2 API Reference*.

AWS SDKs and Tools

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources

for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see AWS SDKs and Tools.

Tip

In supported AWS Regions, you can also use AWS CloudShell for a browser-based, pre-authenticated shell that launches directly from the AWS Management Console.

How VM Import/Export works

To use your VM in Amazon EC2, you must first export it from the virtualization environment, and then import it into Amazon EC2 as either an Amazon Machine Image (AMI) or an instance. You must decide whether you will import your VMs as AMIs or instances.

Topics

Compare image import and instance import processes in VM Import/Export

The following table summarizes the key differences between image import and instance import.

Characteristic	Image import (Recommended)	Instance import
CLI support	AWS CLI	Amazon EC2 CLI
Supported formats for import	OVA, VHD, VHDX, VMDK, raw	VHD, VMDK, raw
Multi-disk support	✓	
Windows BYOL support	✓	

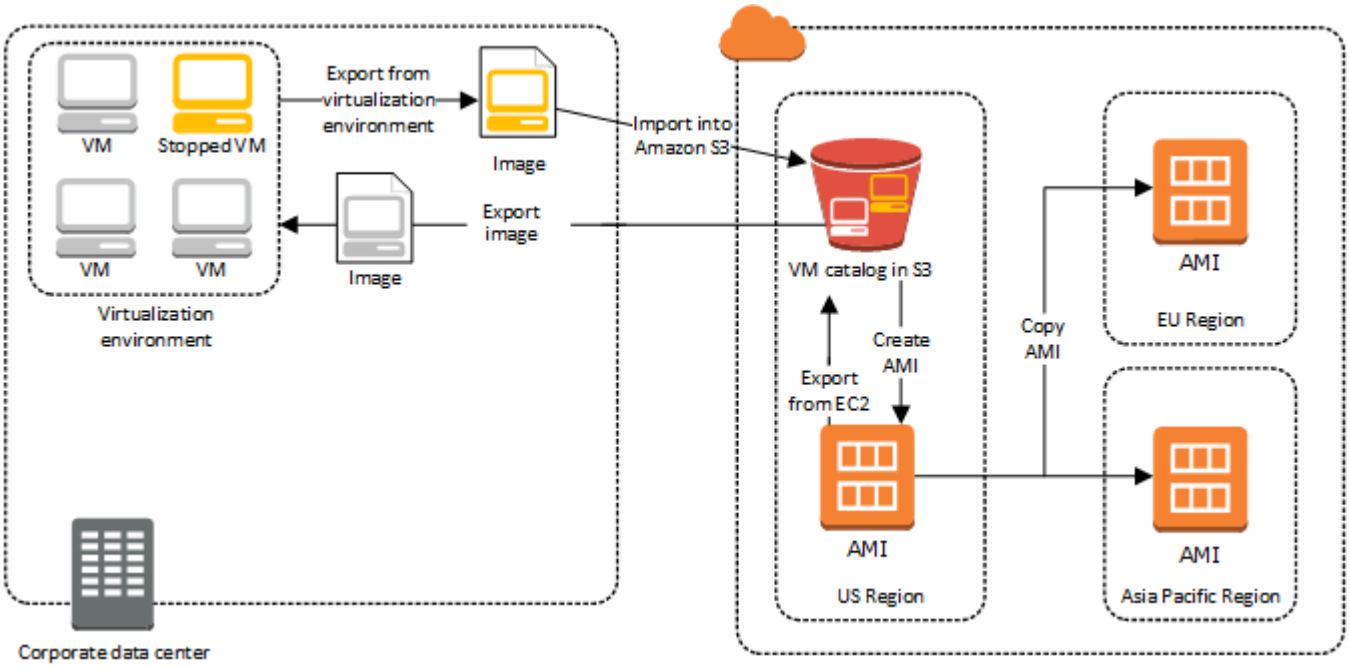
For additional information on these import processes, see Image import overview and Instance import overview.

Image import overview

First, you'll need to prepare your virtual machine for export, and then export it using one of the supported formats. Next, you'll need to upload the VM image to Amazon S3, and then start the

image import task. After the import task is complete, you can launch instances from the AMI. If you want, you can copy the AMI to other Regions so that you can launch instances in those Regions. You can also export an AMI to a VM.

The following diagram shows the process of exporting a VM from your virtualization environment to Amazon EC2 as an AMI.

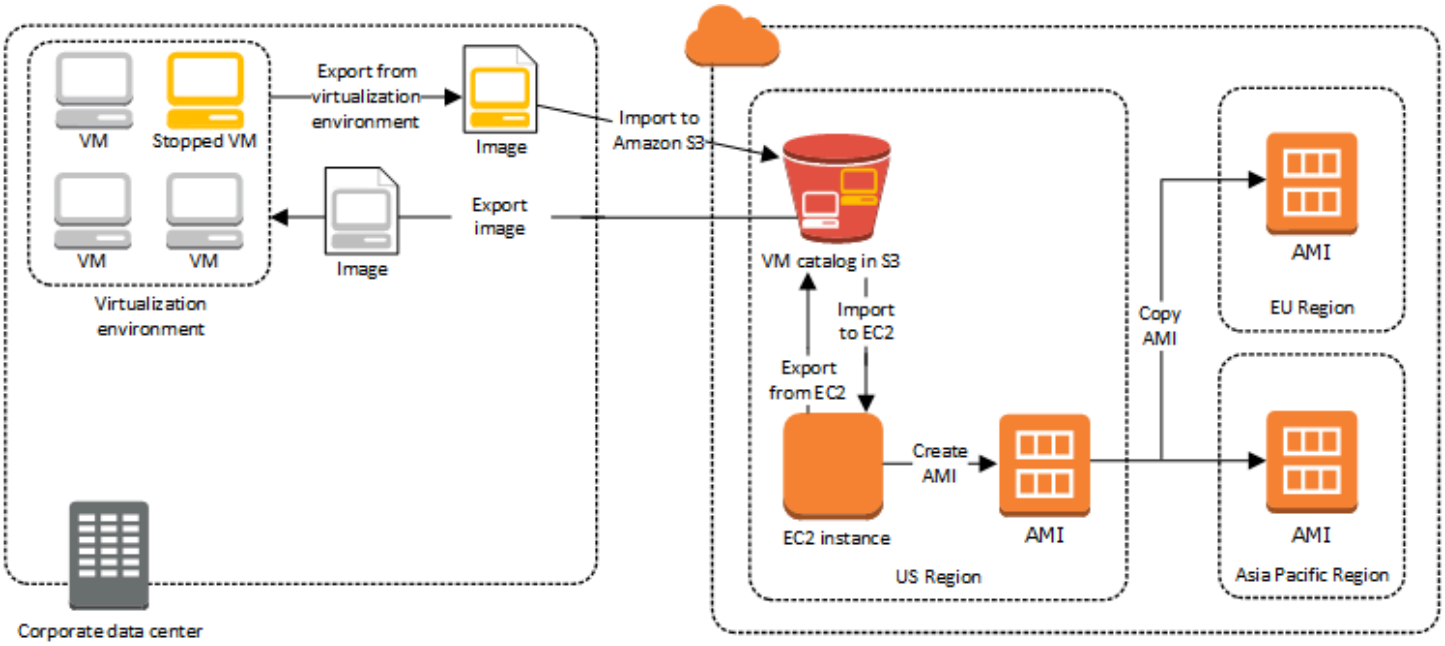


Before you proceed with this process, see VM Import/Export Requirements.

Instance import overview

First, you'll need to prepare your virtual machine for export, and then export it using one of the supported formats. Next, you'll need to upload the VM image to Amazon S3, and then start the instance import task. After the import task is complete, you can create an AMI from the stopped instance. If you want, you can copy the AMI to other Regions so that you can launch instances in those Regions. You can also export a previously imported instance to your virtualization environment.

The following diagram shows the process of exporting a VM from your virtualization environment to Amazon EC2 as an instance.



Before you proceed with this process, see VM Import/Export Requirements.

VM Import/Export Requirements

Before attempting to import a VM, you might need to perform tasks such as preparing your AWS environment by creating a service account with appropriate permissions. You might also need to prepare your locally hosted VM so that is accessible once it is imported into AWS. Review each of these requirements to ensure that your resources are supported for import and take action as needed.

Topics

Requirements for resources that you import with VM Import/Export

Before you begin, you must be aware of the operating systems and image formats that VM Import/Export supports, and understand the limitations on importing instances and volumes.

Topics

Image formats supported by VM Import/Export

VM Import/Export supports the following image formats for importing both disks and VMs:

- Open Virtual Appliance (OVA) image format, which supports importing images with multiple hard disks.
- Stream-optimized ESX Virtual Machine Disk (VMDK) image format, which is compatible with VMware ESX and VMware vSphere virtualization products.

- Fixed and Dynamic Virtual Hard Disk (VHD/VHDX) image formats, which are compatible with Microsoft Hyper-V, Microsoft Azure, and Citrix Xen virtualization products.
- Raw format for importing disks and VMs.

> **Important**
>
> VMs that are created as the result of a physical-to-virtual (P2V) conversion are not supported. For more information, see Limitations for resources being imported with VM Import/Export.

Operating systems supported by VM Import/Export

The following operating systems (OS) can be imported to and exported from Amazon EC2. VMs using ARM64 architecture are not currently supported.

> **Important**
>
> We strongly recommend that you avoid using OS versions that have reached End-of-Life (EOL). OS vendors typically don't provide security patches or other updates for versions that have reached EOL. Continuing to use an EOL system greatly increases the risk of not being able to apply upgrades, including security fixes, and other operational problems. VM Import Export functionalities are not tested on OS versions that have reached EOL.

> **Important**
>
> Starting from February 1, 2026, VM Import Export will begin deprecating support for i386 architecture and End-of-Life OS versions. This deprecation will start with Windows Server 2003 (all versions), Windows Server 2003 R2 (all versions), Windows Server 2008 (all versions), Windows 7 (all versions), Windows 8 (all versions), Windows 8.1 (all versions), CentOS 5 (all versions), CentOS 6 (all versions), CentOS 7 (all versions), CentOS 8 (all versions), Debian 6 (all versions), Debian 7 (all versions), Debian 10 (all versions), Fedora 18 (all versions), Fedora 19 (all versions), Fedora 20 (all versions), Fedora 37 (all versions), Fedora 38 (all versions), Fedora 39 (all versions), Oracle Linux 5 (all versions), Oracle Linux 6 (all versions), Red Hat Enterprise Linux 5 (all versions), Red Hat Enterprise Linux 6 (all versions), SUSE Linux Enterprise Server 11 (all versions), Ubuntu 12.04 (all versions),

Ubuntu 12.10(all versions), Ubuntu 13.04 (all versions), Ubuntu 13.10 (all versions), Ubuntu 14.04 (all versions), Ubuntu 14.10 (all versions), and Ubuntu 15.04 (all versions).

Linux/Unix

The following Linux/Unix operating systems are support by VM Import/Export.

Operating system	Version	Kernel	Service pack
Amazon Linux 2	-	4.14, 4.19, 5.4, 5.10	-
CentOS	5.1–5.11	2.6.18	-
	6.1–6.8	2.6.32	-
	7.0–7.9	3.10.0	-
	8.0–8.2	4.18.0	-
	9	5.14.0	-
Debian	6.0.0–6.0.8	2.6.32	-
	7.0.0–7.8.0	3.2.0	-
	10	4.19.0	-
	11	5.10.0	-
	12.2	6.1.0	-
	12.4	6.1.0	-
Fedora	18	3.2.5	-
	19	3.9.5	-
	20	3.11.10	-
	37	6.0.7	-

Operating system	Version	Kernel	Service pack
	38	6.2.9	-
	39	6.5.6	-
Oracle Linux	5.10–5.11	Unbreakable Enterprise Kernel (UEK) el5uek kernel suffixes	-
	6.1–6.10	Red Hat Compatible Kernel (RHCK) 2.6.32, 2.6.39 Unbreakable Enterprise Kernel (UEK) 3.8.13, 4.1.12	-
	7.0–7.6	Red Hat Compatible Kernel (RHCK) 3.10.0 Unbreakable Enterprise Kernel (UEK) 3.8.13, 4.1.12, 4.14.35, 5.4.17	-
	8.0–8.9	Red Hat Compatible Kernel (RHCK) 4.18.0 Unbreakable Enterprise Kernel (UEK) 5.15.0 (el8uek)	-

Operating system	Version	Kernel	Service pack
	9.0–9.5	Red Hat Compatible Kernel (RHCK) 5.14.0, 5.15.0 Unbreakable Enterprise Kernel (UEK) 5.15.0 (el9uek)	-
	9.6	Red Hat Compatible Kernel (RHCK) 6.12.0 Unbreakable Enterprise Kernel (UEK) 6.12.0 (el9uek)	-
	10.0	Red Hat Compatible Kernel (RHCK) 6.12.0 Unbreakable Enterprise Kernel (UEK) 6.12.0 (el10uek)	-
Red Hat Enterprise Linux (RHEL)	5	2.6.18	-
	6	2.6.32 (except 2.6.32-71)	-
	7	3.10.0	-
	8.0–8.9	4.18.0	-
	9.0–9.6	5.14.0	-
	10	6.12.0	-
Rocky Linux	9.0–9.6	5.14.0	-

Operating system	Version	Kernel	Service pack
	10	6.12.0	-
SUSE Linux Enterprise Server (SLES)	11	2.6.32.12	1
		3.0.13	2
		3.0.76, 3.0.101	3
		3.0.101	4
	12	3.12.28	None
		3.12.49	1
		4.4	2, 3
		4.12	4, 5
	15	4.12	None, 1
		5.3	2, 3
		5.14.21	4, 5
		6.4	6
Ubuntu	12.04	3.2.0	-
	12.10	3.5.0	-
	13.04	3.8.0	-
	13.10	3.11	-
	14.04	3.13.0, 3.16.0, 3.19.0	-
	14.10	3.16	-
	15.04	3.19.0	-

Operating system	Version	Kernel	Service pack
	16.04	4.2.0, 4.4.0, 4.8.0, 4.10.0, 4.15.0	-
	16.10	4.8.0	-
	17.04	4.10.0	-
	18.04	4.15.0, 5.4.0	-
	20.04	5.4.0	-
	22.04	5.15.0	-
	23.04	5.15.0	-

Windows

The following Windows operating systems are supported by VM Import/Export.

Operating system	Edition	Bit version	Available with non-default Regions
Windows Server 2003 (Service Pack 1 or later)	Standard, Datacenter, Enterprise	32, 64	No
Windows Server 2003 R2	Standard, Datacenter, Enterprise	32, 64	No
Windows Server 2008	Standard, Datacenter, Enterprise	32, 64	No
Windows Server 2008 R2	Standard, Web Server, Datacenter, Enterprise	64	Yes [5]
Windows Server 2012	Standard, Datacenter	64	Yes [5]

Operating system	Edition	Bit version	Available with non-default Regions
Windows Server 2012 R2	Standard, Datacenter	64	Yes [5]
Windows Server 2016	Standard, Datacenter [3]	64	Yes [5]
Windows Server 1709	Standard, Datacenter	64	Yes [5]
Windows Server 1803	Standard, Datacenter	64	Yes [5]
Windows Server 2019	Standard, Datacenter	64	Yes [5]
Windows Server 2022	Standard, Datacenter	64	Yes [5,6]
Windows 7 [1]	Home, Professional, Enterprise, Ultimate	32, 64 [4]	Yes [5]
Windows 8 [1]	Home, Professional, Enterprise	32, 64 [4]	Yes [5]
Windows 8.1 [1]	Professional, Enterprise	64	Yes [5]
Windows 10 [1]	Home, Professional, Enterprise, Education	64	Yes [5]
Windows 11 [1,2]	Home, Professional, Enterprise, Education	64	Yes [5,7]

[1] The operating system must have its language set as US `English` during import.

[2] Windows 11 requires the Unified Extensible Firmware Interface (UEFI) boot mode to function. To help ensure a successful import of your VM, we recommend that you specify the optional `--boot-mode` parameter as `uefi`. For more information, see Boot modes supported by VM Import/Export.

[3] Nano Server installations are not supported.

[4] Only the 64-bit version of the OS is supported when launching instances within non-default AWS Regions. For more information, see Available Regions in the *Amazon EC2 User Guide*.

[5] You must first enable the Region before you can use the operating system there. For more information, see Enable or disable AWS Regions in your account in the *AWS Account Management Reference Guide*.

[6] Windows Server 2022 and Windows Server 2025 are not supported in the China (Beijing) and China (Ningxia) Regions.

[7] Windows 11 isn't supported in the Asia Pacific (Hyderabad), Asia Pacific (Jakarta), Asia Pacific (Melbourne), China (Beijing), China (Ningxia), Europe (Spain), Europe (Zurich), and Middle East (UAE) Regions.

Boot modes supported by VM Import/Export

When a computer boots, the first software that it runs is responsible for initializing the platform and providing an interface for the operating system to perform platform-specific operations. VM Import/Export supports two variants of the boot mode: Unified Extensible Firmware Interface (UEFI) and Legacy BIOS. You can choose whether to specify the optional `--boot-mode` parameter as `legacy-bios` or `uefi` when importing your VM.

Refer to the Boot Modes section of the *Amazon Elastic Compute Cloud User Guide* for more information about specifying a boot mode, and UEFI variables.

Volume types and file systems supported by VM Import/Export

VM Import/Export supports importing Windows and Linux VMs with the following file systems.

Linux/Unix

MBR partitioned volumes and GUID Partition Table (GPT) partitioned volumes that are formatted using the ext2, ext3, ext4, Btrfs, JFS, or XFS file system are supported.

> **Important**
>
> Btrfs subvolumes are not supported.

Windows

GUID Partition Table (GPT) and Master Boot Record (MBR) partitioned volumes that are formatted using the NTFS file system are supported. If no boot parameter is specified, and the VM is compatible in both boot modes, the GPT volumes will be converted to MBR partitioned volumes.

VM Import/Export will automatically detect the boot modes your Windows VM is compatible with. If the Windows VM is only compatible in a single boot mode, you don't need to specify a specific `--boot-mode` parameter.

If your Windows VM is compatible with both boot modes, and the following criteria is met for the imported disk, VM Import/Export will select Legacy BIOS by default. You can specify `uefi` for the `--boot-mode` parameter to override this behavior.

- The disk is smaller than 2 terabytes
- The disk does not contain more than 4 primary partitions
- The disk is not a Windows dynamic disk
- The file format is VHDX

Limitations for resources being imported with VM Import/Export

Review the following limitations that apply when you import a VM into Amazon EC2.

Topics

General limitations for your resources

The following limitations apply to any operating system that you can import.

- VMs that are created as the result of a physical-to-virtual (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.

- Importing VMs with dual-boot configurations isn't supported.
- Importing VMs with encrypted volumes isn't supported.
- VM Import/Export doesn't support VMs that use Raw Device Mapping (RDM). Only VMDK disk images are supported.
- VM Import/Export doesn't support VMware SEsparse delta-file format.
- If you import a VM that's compatible with UEFI using the `import-image` command while specifying an EBS snapshot, you must specify a value for the `platform` parameter. For more information, see import-snapshot in the Amazon EC2 API Reference.
- An imported VM may fail to boot if the root partition is not on the same virtual hard drive as the MBR.
- A VM import task fails for VMs with more than 21 volumes attached. Additional disks can be individually imported using the `ImportSnapshot` API.
- VM Import/Export assigns only private IPv4 addresses to your instances, regardless of the auto-assign public IP setting for the subnet. To use a public IPv4 address, you can allocate an Elastic IP address to your account and associate it with your instance. You can also add IPv6 addresses. For more information, see IP addressing for your VPCs and subnets in the *Amazon Virtual Private Cloud User Guide.*
- Multiple network interfaces are not currently supported. After import, your VM has a single virtual network interface that uses DHCP to assign addresses.
- Disk images must be less than 16 TiB. For disk images that are larger than 8 TiB, you must use a manifest file.
  - You can use the `ImportInstance` operation to import VMs with disks up to the maximum supported size.
  - You can use the `ImportImage` operation to import VMs with disks less than 8 TiB in size.

Limitations for Linux/Unix resources

The following limitations apply to Linux operating systems that you can import.

- Imported Linux VMs must use 64-bit images. Migrating 32-bit Linux images isn't supported.
- Imported Linux VMs should use default kernels for best results. VMs that use custom Linux kernels might not migrate successfully.
- When preparing Linux VMs for import, make sure that there is sufficient disk space available on the root volume for installing drivers and other software.

- To help ensure your Linux VM can import successfully and run on Amazon EC2 using the AWS Nitro System, you can install the AWS NVMe and AWS Elastic Network Adapter (ENA) drivers before exporting your VM from its virtualization environment. For more information, see Amazon EBS and NVMe on Linux instances and Enable enhanced networking with the Elastic Network Adapter (ENA) on Linux instances in the *Amazon EC2 User Guide*.
- If you import a Linux VM compatible with UEFI, you must have a fallback EFI binary, BOOTX64.EFI, located on the EFI System Partition.
- Predictable network interface names are not supported for virtual machine imports.

Limitations for Windows resources

The following limitations apply to Windows operating systems that you can import.

- When preparing Windows VMs for import, make sure that there is sufficient disk space available on the root volume for installing drivers and other software. For Microsoft Windows VMs, configure a fixed page file size and ensure that there is at least 6 GiB of free space available on the root volume. If Windows is configured to use the "Automatically manage paging file size for all drives" setting, it might create 16 GB `pagefile.sys` files on the C drive of the instance.
- If you import a Windows VM compatible with UEFI, we convert GPT boot volumes to MBR if the following are true: the image format is VHDX, the uncompressed size is 2 TiB or smaller, there are no more than three primary partitions, and the volume is not a dynamic disk.
- If you import a Windows Server 2012 R2 VM, VM Import/Export installs the single root I/O virtualization (SR-IOV) drivers. These drivers are not required unless you plan to use enhanced networking, which provides higher performance (packets per second), lower latency, and lower jitter.
- VM Import/Export does not support Emergency Management Services (EMS). If EMS is enabled for a source Windows VM, we disable it in the imported image.
- 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 VMs.
- Windows Server VMs with the Hyper-V server role installed are not supported.

Configurations to export VMs from your virtualization environment

Before you can import your VM to Amazon EC2, you need to export it from your virtualization environment. Use the following guidelines to configure your VM before exporting it.

Topics

General configurations

The following configurations should be made in your VM before you export it from your virtualization environment. You should also review the section specific to your operating system for additional required configurations.

- Disable any antivirus or intrusion detection software on your VM. These services can be re-enabled after the import process is complete.
- Uninstall the VMware Tools from your VMware VM.
- Disconnect any CD-ROM drives (virtual or physical).
- Your source VM must have a functional DHCP client service. Ensure that the service can start and is not disabled administratively. All static IP addresses currently assigned to the source VM are removed during import. When your imported instance is launched in an Amazon VPC, it receives a primary private IP address from the IPv4 address range of the subnet. If you don't specify a primary private IP address when you launch the instance, we select an available IP address in the subnet's IPv4 range for you. For more information, see VPC and Subnet Sizing.

Linux/Unix configurations

The following configurations should be made in your Linux VM before you export it from your virtualization environment. This section assumes you have already reviewed General configurations.

- Enable Secure Shell (SSH) for remote access.

- Make sure that your host firewall (such as Linux **iptables**) allows access to SSH. Otherwise, you won't be able to access your instance after the import is complete.
- Make sure that you have configured a non-root user to use public key-based SSH to access your instance after it is imported. The use of password-based SSH and root login over SSH are both possible, but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import does not configure an `ec2-user` account as part of the import process.
- Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
- Make sure that your Linux VM uses one of the following for the root file system: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.
- Make sure that your Linux VM is not using predictable network interface device names.
- Shut down your VM before exporting it from your virtualization environment.

Windows configurations

The following configurations should be made in your Windows VM before you export it from your virtualization environment. This section assumes you have already reviewed General configurations.

- Enable Remote Desktop (RDP) for remote access.
- Make sure that your host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, you cannot access your instance after the import is complete.
- Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the import process might fail.
- Install .NET Framework 4.5 or later on the VM. We install the .NET framework on your VM as needed.
- Disable Autologon on your Windows VM.
- Open **Control Panel** > **System and Security** > **Windows Update**. In the left pane, choose **Change settings**. Choose the desired setting. Be aware that if you choose **Download updates but let me choose whether to install them** (the default value) the update check can temporarily consume between 50% and 99% of CPU resources on the instance. The check usually occurs several minutes after the instance starts. Make sure that there are no pending Microsoft updates, and that the computer is not set to install software when it reboots.
- Apply the following hot fixes as needed:

  - 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
- Set the RealTimeIsUniversal registry key. For more information, see Set the time for your Amazon EC2 instance in the *Amazon EC2 User Guide*.
- Run System Preparation (Sysprep) on your Windows Server VM images, either before or after importing your VM.
  - If you run Sysprep before importing your VM, the import process adds an answer file (`unattend.xml`) to the VM that automatically accepts the End User License Agreement (EULA) and sets the locale to EN-US.
  - If you run Sysprep after importing your VM, we recommend that you use EC2Launch (Windows Server 2016 and later) or EC2Config (through Windows Server 2012 R2) to run Sysprep.

To include your own answer file instead of the default (`unattend.xml`)

1. Copy the following sample file below and set the **processorArchitecture** parameter to **x86** or **amd64**, depending on your operating system architecture:

```
<?xml version='1.0' encoding='UTF-8'?>
<unattend xmlns:wcm='https://schemas.microsoft.com/WMIConfig/2002/State'
 xmlns='urn:schemas-microsoft-com:unattend'>
 <settings pass='oobeSystem'>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
 name='Microsoft-Windows-International-Core' publicKeyToken='31bf3856ad364e35'
 language='neutral'>
   <InputLocale>en-US</InputLocale>
   <SystemLocale>en-US</SystemLocale>
   <UILanguage>en-US</UILanguage>
   <UserLocale>en-US</UserLocale>
  </component>
  <component versionScope='nonSxS' processorArchitecture='x86 or amd64'
 name='Microsoft-Windows-Shell-Setup' publicKeyToken='31bf3856ad364e35'
 language='neutral'>
   <OOBE>
    <HideEULAPage>true</HideEULAPage>
    <SkipMachineOOBE>true</SkipMachineOOBE>
    <SkipUserOOBE>true</SkipUserOOBE>
   </OOBE>
  </component>
 </settings>
```

```
</unattend>
```

2. Save the file in the `C:\Windows\Panther` directory with the name `unattend.xml`.
3. Run Sysprep with the **/oobe** and **/generalize** options. These options strip all unique system information from the Windows installation and prompt you to reset the administrator password.
4. Shut down the VM and export it from your virtualization environment.

Required permissions for VM Import/Export

VM Import/Export requires certain permissions for your users, groups, and roles. Additionally, a service role is required to perform certain operations on your behalf.

Topics

Required permissions

Your users, groups, and roles need the following permissions in their IAM policy to use VM Import/Export:

> **Note**
>
> Some actions require the use of an Amazon Simple Storage Service (Amazon S3) bucket. This example policy does not grant permission to create S3 buckets. The user or role that you use will need to specify an existing bucket, or have permissions to create a new bucket with the `s3:CreateBucket` action.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-import-bucket",
        "arn:aws:s3:::amzn-s3-demo-import-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-export-bucket",
        "arn:aws:s3:::amzn-s3-demo-export-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CancelConversionTask",
        "ec2:CancelExportTask",
        "ec2:CreateImage",
        "ec2:CreateInstanceExportTask",
        "ec2:CreateTags",
        "ec2:DescribeConversionTasks",
        "ec2:DescribeExportTasks",
        "ec2:DescribeExportImageTasks",
        "ec2:DescribeImages",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeSnapshots",
        "ec2:DescribeTags",
        "ec2:ExportImage",
        "ec2:ImportInstance",
        "ec2:ImportVolume",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances",
        "ec2:ImportImage",
        "ec2:ImportSnapshot",
        "ec2:DescribeImportImageTasks",
        "ec2:DescribeImportSnapshotTasks",
        "ec2:CancelImportTask"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

Required service role

VM Import/Export requires a role to perform certain operations on your behalf. You must create a service role named `vmimport` with a trust relationship policy document that allows VM Import/Export to assume the role, and you must attach an IAM policy to the role. For more information, see IAM Roles in the *IAM User Guide*.

Prerequisite

You must enable AWS Security Token Service (AWS STS) in any Region where you plan to use VM Import/Export. For more information, see Activating and deactivating AWS STS in an AWS Region.

To create the service role

1. Create a file named `trust-policy.json` on your computer. Add the following policy to the file:

   JSON

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

2. Use the create-role command to create a role named `vmimport` and grant VM Import/Export access to it. Ensure that you specify the full path to the location of the `trust-policy.json`

file that you created in the previous step, and that you include the `file://` prefix as shown the following example:

```
aws iam create-role --role-name vmimport --assume-role-policy-document "file://C:\import\trust-policy.json"
```

3. Create a file named `role-policy.json` with the following policy, where *amzn-s3-demo-import-bucket* is the bucket for imported disk images and *amzn-s3-demo-export-bucket* is the bucket for exported disk images:

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-import-bucket",
            "arn:aws:s3:::amzn-s3-demo-import-bucket/*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject",
            "s3:GetBucketAcl"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-export-bucket",
            "arn:aws:s3:::amzn-s3-demo-export-bucket/*"
         ]
      },
      {
```

```
            "Effect": "Allow",
            "Action": [
               "ec2:ModifySnapshotAttribute",
               "ec2:CopySnapshot",
               "ec2:RegisterImage",
               "ec2:Describe*"
            ],
            "Resource": "*"
         }
      ]
}
```

4. (Optional) To import resources encrypted using an AWS KMS key from AWS Key Management Service, add the following permissions to the `role-policy.json` file.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant",
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:Encrypt",
    "kms:GenerateDataKey*",
    "kms:ReEncrypt*"
  ],
  "Resource": "*"
}
```

   If you use a KMS key other than the default provided by Amazon EBS, you must grant VM Import/Export permission to the KMS key if you enable Amazon EBS encryption by default or enable encryption on an import operation. You can specify the Amazon Resource Name (ARN) of the KMS key as the resource instead of *.

5. (Optional) To attach license configurations to an AMI, add the following License Manager permissions to the `role-policy.json` file.

```
{
  "Effect": "Allow",
  "Action": [
    "license-manager:GetLicenseConfiguration",
    "license-manager:UpdateLicenseSpecificationsForResource",
    "license-manager:ListLicenseSpecificationsForResource"
```

```
  ],
  "Resource": "*"
}
```

6. Use the following put-role-policy command to attach the policy to the role created above. Ensure that you specify the full path to the location of the `role-policy.json` file.

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-
document "file://C:\import\role-policy.json"
```

7. For additional security controls, context keys such as `aws:SourceAccount` and `aws:SourceArn` can be added to the trust policy for this newly created role. VM Import/Export will publish the `SourceAccount` and `SourceArn` keys as specified in the example below to assume this role:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "vmie.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:Externalid": "vmimport",
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:vmie:*:111122223333:*"
                }
            }
        }
    ]
}
```

Licensing for your imported VMs

When you create a new VM Import task, you have two options for how to specify the license type for the operating system. You can specify a value for either the `--license-type` or the `--usage-operation` parameter. Specifying a value for both parameters will return an error. You can use `--usage-operation` to blend your operating system and SQL Server licenses.

> **Important**
>
> AWS VM Import/Export strongly recommends specifying a value for either the `--license-type` or `--usage-operation` parameter when you create a new VM Import task. This ensures your operating system is licensed appropriately and your billing is optimized. If you choose a license type that is incompatible with your VM, the VM Import task fails with an error message. For more information, see Specify a licensing option for your import.

Topics

Licensing considerations

We recommend that you review the following licensing considerations appropriate for the operating system that you wish to import.

Topics

Licensing considerations for Linux/Unix

Linux operating systems support only the BYOL license type for a VM import task.

Migrated Red Hat Enterprise Linux (RHEL) VMs must use Cloud Access (BYOS) 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 considerations for Windows

Windows Server operating systems support either the `BYOL` or `AWS` license type. Windows client operating systems (such as Windows 10) support only BYOL licenses.

By default, an AWS license is used when you create a VM import task if the VM has a Windows Server OS. Otherwise, a 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 VM Import/Export, which are subject to the current terms and abilities of AWS VM Import/Export.
  - 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 Microsoft's document Volume Licensing Product 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.

  For more information, see *Generating Windows Server and SQL Server on Amazon EC2 estimates in the AWS Pricing Calculator User Guide*.

Specify a licensing option for your import

You can specify either a license type or a usage operation for the VMs that you migrate. Specifying a license option ensures your operating system is licensed appropriately and your billing is optimized. If you choose a license type that is incompatible with your VM, the VM Import task fails with an error message. For more information on troubleshooting errors, see Troubleshooting VM Import/Export.

Topics

Specify a license type

Specify license type

You can specify the following values for the `--license-type` parameter:

- `AWS` (license included) – Replaces the source-system license with an AWS license on the migrated VM.
- `BYOL` – Retains the source-system license on the migrated VM.

> **Note**
>
> Leaving the `--license-type` parameter undefined while importing a Windows Server OS is the same as choosing `AWS` and the same as choosing `BYOL` when importing a Windows client OS (such as Windows 10) or a Linux OS.

For example, to specify the license type as an AWS license, run the following command:

```
aws ec2 import-image \
    --license-type aws \
    --disk-containers Format=OVA,Url=S3://bucket_name/sql_std_image.ova
```

Specify a usage operation

> **Important**
>
> AWS stamps the software edition with the information that you provide. You are responsible for entering the correct software edition information for any licenses that you bring to AWS.

You can specify the following values for the `--usage-operation` parameter:

Platform details	Usage operation *
Windows Server License Included without SQL Server	RunInstances:0002
Windows Server License Included with SQL Server (any edition) BYOL	RunInstances:0002
Windows Server License Included with SQL Server Standard License Included	RunInstances:0006
Windows Server License Included with SQL Server Enterprise License Included	RunInstances:0102
Windows Server License Included with SQL Server Web License Included	RunInstances:0202
Windows Server BYOL without SQL Server	RunInstances:0800
Windows Server BYOL with SQL (any edition) BYOL	RunInstances:0800

Platform details	Usage operation *
Linux/UNIX without SQL Server	RunInstances
Linux/UNIX with SQL Server (any edition) BYOL	RunInstances
Linux/UNIX with SQL Server Enterprise License Included	RunInstances:0100
Linux/UNIX with SQL Server Standard License Included	RunInstances:0004
Linux/UNIX with SQL Server Web License Included	RunInstances:0200
Red Hat Enterprise Linux	RunInstances:0010
SUSE Linux	RunInstances:000g

* If you are running Spot Instances, the `lineup/Operation` on your AWS Cost and Usage Report might be different from the **Usage operation** value that is listed here.

For example, to specify the usage operation for Windows with SQL Server Standard, run the following command:

```
aws ec2 import-image \
    --usage-operation RunInstances:0006 \
    --disk-containers Format=OVA,Url=S3://bucket_name/sql_std_image.ova
```

For more information about billing codes, see AMI billing information fields.

VM Import/Export processes

VM Import/Export has processes for eligible resources that you can use to import into and export out of the AWS Cloud. You can import individual disks, or whole VMs that meet the respective requirements of the import process.

You can also export an Amazon EC2 instance or an AMI in a supported file format. For more information on resources that are eligible for export, see Considerations for instance export and Considerations for image export.

Processes

- Import a VM to Amazon EC2 as an image using VM Import/Export
- Import a disk as an EBS snapshot using VM Import/Export
- Export an EC2 instance as a VM using VM Import/Export
- Export a VM from an Amazon Machine Image (AMI) using VM Import/Export

Import a VM to Amazon EC2 as an image using VM Import/Export

> **Tip**
>
> To import your virtual machines (VMs) with a console-based experience, you can use the *Import virtual machine images to AWS* template in the Migration Hub Orchestrator console. For more information, see the *AWS Migration Hub Orchestrator User Guide*.

You can use VM Import/Export to import virtual machine (VM) images from your virtualization environment to Amazon EC2 as Amazon Machine Images (AMI), which you can use to launch instances. Subsequently, you can export the VM images from an instance back to your virtualization environment. This enables you to leverage your investments in the VMs that you have built to meet your IT security, configuration management, and compliance requirements by bringing them into Amazon EC2.

Contents

- Programmatic modifications made to VMs by VM Import/Export
- Import your VM as an image
- Monitor an import image task
- Cancel an import image task
- Create an EC2 instance from an imported image

Export your VM from its virtualization environment

After you have prepared your VM for export, you can export it from your virtualization environment. When importing a VM as an image, you can import disks in the following formats: Open Virtualization Archive (OVA), Virtual Machine Disk (VMDK), Virtual Hard Disk (VHD/VHDX), and raw. With some virtualization environments, you would export to Open Virtualization Format (OVF), which typically includes one or more VMDK, VHD, or VHDX files, and then package the files into an OVA file.

For more information, see the documentation for your virtualization environment. For example:

- **VMware** — Search for "Export an OVF Template" on the VMware Docs site. Follow the instructions to export an OVA.
- **Citrix** — Importing and Exporting VMs on the Citrix website.
- **Microsoft Hyper-V** — Overview of exporting and importing a virtual machine on the Microsoft website.
- **Microsoft Azure** — Download a Windows VHD from Azure or Download a Linux VHD from Azure on the Microsoft website. From the Azure Portal, choose the VM to migrate, and then choose **Disks**. Select each disk (either OS or data) and choose **Create Snapshot**. On the completed snapshot resource, choose **Export**. This creates a URL that you can use to download the virtual image.

Programmatic modifications made to VMs by VM Import/Export

When importing a VM using the `ImportImage` API, AWS modifies the file system and adds drivers to make the imported VM bootable. When writing a modified file, AWS retains the original file at the same location under a new name. The following actions may occur:

General

- For parity with images provided by AWS, the AWS Systems Manager client is installed on the VM.

Windows

- Modifying registry settings to make the VM bootable.

Linux

- Installing Citrix PV drivers either directly in OS or modify initrd/initramfs to contain them.
- Modifying network scripts to replace static IPs with dynamic IPs.
- 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 must 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 non-standard file system types (cifs, smbfs, vboxsf, sshfs, etc.) are disabled.
- Modifying grub bootloader settings such as the default entry and timeout.

Import VM without modifications

If you need to import a VM without programmatic modifications, we recommend that you follow these steps instead of using `ImportImage`.

> **Important**
>
> If you use this process, AWS does not do any post-import validations to ensure that the image is bootable. It is your responsibility to ensure that you properly prepare your VM for exporting.

To import a VM without modifications

1. Prepare your VM for export. For more information, see Configurations to export VMs from your virtualization environment.
2. Export the boot disk for your VM in one of the following file formats: VHD/VHDX, VMDK, or raw. For more information, refer to the documentation for your virtualization environment.
3. Use the put-object command to upload the exported boot disk file to an Amazon S3 bucket in the Region where you want to create the image.
4. Use the import-snapshot command to import the boot disk as a snapshot. For more information about importing a snapshot, see Import a disk as an EBS snapshot using VM Import/Export.

   > **Note**
   >
   > You can monitor the progress of the import snapshot task using the describe-import-snapshot-tasks command.

   Make a note of the snapshot ID returned by the command. You'll need it for the next step.
5. Use the register-image command to register a new AMI, and specify the snapshot from the previous step as the root device volume.

   Make a note of the image ID returned by the command. You'll need it for the next step.
6. After the AMI reaches the `available` state, you can use it to launch instances.

Import your VM as an image

After exporting your VM from your virtualization environment, you can import it to Amazon EC2 using VM Import/Export. The import process is the same regardless of the origin of the VM.

Tasks

- Prerequisites for importing a VM into Amazon EC2
- Upload the image to Amazon S3
- Import the VM

Prerequisites for importing a VM into Amazon EC2

- Create an Amazon Simple Storage Service (Amazon S3) bucket for storing the exported images or choose an existing bucket. The bucket must be in the Region where you want to import your VMs. For more information about S3 buckets, see the Amazon Simple Storage Service User Guide.
- Create an IAM role named `vmimport`. For more information, see Required service role.
- If you have not already installed the AWS CLI on the computer you'll use to run the import commands, see the AWS Command Line Interface User Guide.

> **Tip**
>
> In supported AWS Regions, you can also use AWS CloudShell for a browser-based, pre-authenticated shell that launches directly from the AWS Management Console.

Upload the image to Amazon S3

Upload your VM image file to your S3 bucket using the upload tool of your choice. For information about uploading objects through the Amazon S3 console, see Uploading Objects.

Import the VM

After you upload your VM image file to Amazon S3, you can use the AWS CLI to import the image. These tools accept either the S3 bucket and path to the file or a URL for a public Amazon S3 file. Private Amazon S3 files require a presigned URL.

You can also use the *Import virtual machine images to AWS template* in the Migration Hub Orchestrator console to import your on-premises virtual machine images to AWS. For more information, see the section called "Example 4: Import an image using Migration Hub Orchestrator".

> **Important**
>
> - AWS VM Import/Export strongly recommends specifying a value for either the `--license-type` or `--usage-operation` parameter when you create a new VM Import task. This ensures your operating system is licensed appropriately and your billing is optimized. For more information, see Licensing for your imported VMs.

- AWS VM Import/Export only supports images that were natively installed inside the source VM and not those created using a physical-to-virtual (P2V) conversion process. For more information, see the VM Import/Export Requirements.

Examples

- Example 1: Import an image using an OVA file
- Example 2: Import an image with multiple disks
- Example 3: Import with the encrypted option enabled
- Example 4: Import an image using Migration Hub Orchestrator

Example 1: Import an image using an OVA file

AWS CLI

Use the following import-image command.

```
aws ec2 import-image \
    --description "$(date '+%b %d %H:%M') My server VM" \
    --license-type "AWS" \
    --disk-containers '[{
    "Format": "OVA",
    "UserBucket": {
      "S3Bucket": "amzn-s3-demo-import-bucket",
      "S3Key": "vms/my-server-vm.ova"
    }
  }]'
```

PowerShell

Use the Import-EC2Image cmdlet as follows.

```
Import-EC2Image `
    -Description ((Get-Date -Format "MMM dd HH:mm ") + "My server OVA") `
    -LicenseType "AWS" `
    -DiskContainer @(
        @{
            Format = "OVA"
            UserBucket = @{
```

```
                S3Bucket = "amzn-s3-demo-import-bucket"
                S3Key = "vms/my-server-vm.ova"
            }
        }
    )
```

Example 2: Import an image with multiple disks

AWS CLI

Use the import-image command.

```
aws ec2 import-image \
    --description "$(date '+%b %d %H:%M') My server disks" \
    --license-type "AWS" \
    --disk-containers '[
    {
      "Description": "First disk",
      "Format": "vmdk",
      "UserBucket": {
        "S3Bucket": "amzn-s3-demo-import-bucket",
        "S3Key": "disks/my-server-vm-disk2.vmdk"
      }
    },
    {
      "Description": "Second disk",
      "Format": "vmdk",
      "UserBucket": {
        "S3Bucket": "amzn-s3-demo-import-bucket",
        "S3Key": "disks/my-server-vm-disk2.vmdk"
      }
    }
  ]'
```

PowerShell

Use the Import-EC2Image cmdlet as follows.

```
Import-EC2Image `
    -Description ((Get-Date -Format "MMM dd HH:mm ") + "My server disks") `
    -LicenseType "AWS" `
    -DiskContainer @(
```

```
        @{
            Description = "First disk"
            Format = "vmdk"
            UserBucket = @{
                S3Bucket = "amzn-s3-demo-import-bucket"
                S3Key = "disks/my-server-vm-disk1.vmdk"
            }
        },
        @{
            Description = "Second disk"
            Format = "vmdk"
            UserBucket = @{
                S3Bucket = "amzn-s3-demo-import-bucket"
                S3Key = "disks/my-server-vm-disk2.vmdk"
            }
        }
    )
```

Example 3: Import with the encrypted option enabled

The CMK provided for encryption must not be disabled during the entire import process. For more information, see Amazon EBS encryption in the *Amazon EBS User Guide*.

AWS CLI

Use the following import-image command.

```
aws ec2 import-image \
    --description "$(date '+%b %d %H:%M') My server OVA" \
    --encrypted \
    --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
    --disk-containers '[{
        "Format": "OVA",
        "UserBucket": {
          "S3Bucket": "amzn-s3-demo-import-bucket",
          "S3Key": "vms/my-server-vm.ova"
        }
    }]'
```

PowerShell

Use the Import-EC2Image cmdlet as follows.

```
Import-EC2Image `
    -Description ((Get-Date -Format "MMM dd HH:mm ") + "My server disks") `
    -LicenseType "AWS" `
    -DiskContainer @(
        @{
            Format = "OVA"
            UserBucket = @{
                S3Bucket = "amzn-s3-demo-import-bucket"
                S3Key = "vms/my-server-vm.ova"
            }0
        }
    ) `
    -Encrypted $true `
    -KmsKeyId "alias/aws/ebs"
```

Example 4: Import an image using Migration Hub Orchestrator

Console

To import an image using a template

1. Open the Migration Hub Orchestrator console.
2. In the navigation pane, choose **Create migration workflow**.
3. On the **Choose a workflow template** page, choose the **Import virtual images to AWS** template.
4. Configure and submit your workflow to begin the VM import. For more information, see the *AWS Migration Hub Orchestrator User Guide*.

Monitor an import image task

You can monitor the progress of an import image task for VM Import/Export. The following are the status values for an import image task:

- `active` — The import task is in progress.
- `deleting` — The import task is being canceled.
- `deleted` — The import task is canceled.
- `updating` — Import status is updating.

- `validating` — The imported image is being validated.
- `validated` — The imported image was validated.
- `converting` — The imported image is being converted into an AMI.
- `completed` — The import task is completed and the AMI is ready to use.

AWS CLI

To get the status of an import image task

Use the following describe-import-image-tasks command.

```
aws ec2 describe-import-image-tasks \
    --import-task-ids import-ami-1234567890abcdef0
```

The following is example output. When the import task is completed, the ID of the AMI is provided in `ImageId`.

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-01234567890abcdef",
            "ImageId": "ami-1234567890EXAMPLE",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "SnapshotId": "snap-111222333444aaabb",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "amzn-s3-demo-import-bucket",
                        "S3Key": "vms/my-server-vm.ova"
                    }
                }
            ],
            "Status": "completed"
        }
    ]
}
```

To get the status of all import image tasks

Use the following describe-import-image-tasks command. The **sed** command truncates the status message. If the task fails and the status message is long, it makes the table harder to read.

```
aws ec2 describe-import-image-tasks \
  --query "ImportImageTasks[*].{Description:Description, Progress:Progress,
 Status:Status, ImportTaskId:ImportTaskId, StatusMessage:StatusMessage}" \
  --output table | \
    sed 's/\(.\{120\}\).*/\1|/'
```

The following is example output. You can display any additional fields that you need.

```
+---------------------+-------------------------------+-----------+----------
+-----------------
|    Description      |         ImportTaskId          | Progress  | Status   |
 StatusMessage |
+-----------------------------------+------------------+-----------+----------
+----------------+
|  My server disks    |  import-ami-01234567890abaaaa |  62       |  active  |
 booting       |
|  My server OVA      |  import-ami-01234567890abbbbb |  62       |  active  |
 booting       |
|  My server disks    |  import-ami-01234567890accccc |  62       |  active  |
 booting       |
+-----------------------------------+------------------+-----------+----------
+----------------+
```

PowerShell

To get the status of an import image task

Use the Get-EC2ImportImageTask cmdlet as follows.

```
Get-EC2ImportImageTask `
    -ImportTaskId import-ami-01234567890abcdef |
        Format-List ImportTaskId, Status, Progress, ImageId,
           @{Name='SnapshotDetails';Expression={ $_.SnapshotDetails | Out-
String }},
           @{Name='UserBucket';Expression={ $_.SnapshotDetails.UserBucket | Out-
String }},
```

The following is example output. When the import task is completed, the ID of the AMI is provided in `ImageId`.

```
ImportTaskId          : import-ami-01234567890abcdef
Status                : completed
Progress              :
ImageId               : ami-1234567890EXAMPLE
SnapshotDetails       :
                        Description   :
                        DeviceName    : /dev/sda1
                        DiskImageSize : 549272064
                        Format        : VMDK
                        Progress      :
                        SnapshotId    : snap-111222333444aaabb
                        Status        : completed
                        StatusMessage :
                        Url           :
                        UserBucket    : Amazon.EC2.Model.UserBucketDetails
UserBucket            :
                        S3Bucket : amzn-s3-demo-import-bucket
                        S3Key    : vms/my-server-vm.ova
```

To get the status of all import image tasks

Use the Get-EC2ImportImageTask cmdlet as follows.

```
Get-EC2ImportImageTask |
    Format-Table Description, ImportTaskId, Progress, Status, StatusMessage -
AutoSize
```

The following is example output. You can display any additional fields that you need.

```
Description       ImportTaskId                Progress Status     StatusMessage
-----------------  ------------                -------- ------     -------------
My server disks   import-ami-01234567890abaaaa 62       active     booting
My server OVA     import-ami-01234567890abbbbb 62       active     booting
My server disks   import-ami-01234567890accccc          completed
```

Cancel an import image task

After you start an image import task using VM Import/Export, you can cancel the import operation if needed.

To describe your import image tasks, see Monitor an import image task.

AWS CLI

To cancel an import image task

Use the cancel-import-task command.

```
aws ec2 cancel-import-task \
    --import-task-id import-ami-1234567890abcdef0
```

PowerShell

To cancel an import image task

Use the Stop-EC2ImportTask cmdlet.

```
Stop-EC2ImportTask `
    -ImportTaskId import-ami-1234567890abcdef0
```

Create an EC2 instance from an imported image

After the import image task is complete, you can launch an instance using the resulting AMI or copy the AMI to another Region. For more information, see the following documentation in the *Amazon EC2 User Guide*:

- Launch an instance
- Copy an AMI

For some operating systems, the device drivers for enhanced networking and NVMe block devices that are required by instances built on the Nitro system are not installed automatically during import. To install these drivers manually, use the directions in the following documentation in the *Amazon EC2 User Guide*.

- (Windows instances) Install the latest version of one of the following: EC2LaunchV2, EC2Launch, or EC2Config.
- (Windows instances) Install or upgrade AWS NVMe drivers using PowerShell
- (Linux instances) Install or upgrade the NVMe driver
- Enable enhanced networking

After you finish customizing your instance, create you can create a new image from the customized instance. For more information, see Create an AMI in the *Amazon EC2 User Guide*.

Import a disk as an EBS snapshot using VM Import/Export

VM Import/Export enables you to import your disks as Amazon EBS snapshots. After the snapshot is created, you can create an EBS volume from the snapshot, and then attach the volume to an EC2 instance.

An imported snapshot has an arbitrary volume ID that should not be used for any purpose.

Prerequisites for importing a snapshot

- The following disk formats are supported: Virtual Hard Disk (VHD/VHDX), ESX Virtual Machine Disk (VMDK), and raw.
- You must first upload your disks to Amazon S3.
- If you have not already installed the AWS CLI on the computer you'll use to run the import commands, see the AWS Command Line Interface User Guide.

> **Tip**
>
> In supported AWS Regions, you can also use AWS CloudShell for a browser-based, pre-authenticated shell that launches directly from the AWS Management Console.

Start an import snapshot task

You can specify the URL of the S3 bucket that contains the disk image or provide the S3 bucket name and key.

AWS CLI

To import a snapshot

Use the import-snapshot command.

```
aws ec2 import-snapshot \
    --description "My server VM" \
    --disk-container "file://C:\import\containers.json"
```

The file `containers.json` is a JSON document that contains the required information.

```
{
    "Description": "My server VM",
    "Format": "VMDK",
    "UserBucket": {
        "S3Bucket": "amzn-s3-demo-import-bucket",
        "S3Key": "vms/my-server-vm.vmdk"
    }
}
```

The following is example output.

```
{
    "Description": "My server VM",
    "ImportTaskId": "import-snap-1234567890abcdef0",
    "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "0.0",
        "Format": "VMDK",
        "Progress": "3",
        "Status": "active",
        "StatusMessage": "pending",
        "UserBucket": {
            "S3Bucket": "amzn-s3-demo-import-bucket",
            "S3Key": "vms/my-server-vm.vmdk"
        }
    }
}
```

PowerShell

To import a snapshot

Use the Import-EC2Snapshot cmdlet.

```
Import-EC2Snapshot `
    -DiskContainer_Description "My server VM" `
    -DiskContainer_Format "VMDK" `
    -DiskContainer_S3Bucket "amzn-s3-demo-import-bucket" `
    -DiskContainer_S3Key "vms/my-server-vm.vmdk"
```

The following is example output.

```
Description  ImportTaskId                 SnapshotTaskDetail                 Tags
-----------  ------------                 ------------------                 ----
My server VM import-snap-1234567890abcdef0 Amazon.EC2.Model.SnapshotTaskDetail
```

Monitor an import snapshot task

After you start an import snapshot task using VM Import/Export, you can monitor the import operation. If the task status is `active`, it means that the import task is in progress. The snapshot is ready to use when the status is `completed`.

AWS CLI

To get the status of an import snapshot task

Use the following describe-import-snapshot-tasks command.

```
aws ec2 describe-import-snapshot-tasks \
    --import-task-ids import-snap-1234567890abcdef0
```

The following is example output.

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VM",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
                "Description": "My server VMDK",
                "DiskImageSize": "3.115815424E9",
                "Format": "VMDK",
                "Progress": "22",
```

```
                "Status": "active",
                "StatusMessage": "downloading/converting",
                "UserBucket": {
                    "S3Bucket": "amzn-s3-demo-import-bucket",
                    "S3Key": "vms/my-server-vm.vmdk"
                },
            }
        }
    ]
}
```

To get the status of all import snapshot tasks

Use the following describe-import-snapshot-tasks command.

```
aws ec2 describe-import-snapshot-tasks \
    --query "ImportSnapshotTasks[*].{Description:Description,
 ImportTaskId:ImportTaskId, Status:SnapshotTaskDetail.Status, Progress:
 SnapshotTaskDetail.Progress, SnapshotID: SnapshotTaskDetail.SnapshotId,  S3Key:
 SnapshotTaskDetail.UserBucket.S3Key}" \
    --output table
```

The following is example output. You can display any additional fields that you need.

```
-----------------------------------------------------------------------------------------------------
|                                                    DescribeImportSnapshotTasks
                                       |
+--------------+--------------------------------+-------------+-----------
+--------------------+------------------------+
|  Description |         ImportTaskId           |   Status    | Progress  |
 S3Key          |       SnapshotID        |
+--------------+--------------------------------+-------------+-----------
+--------------------+------------------------+
|  My server VM|  import-snap-1234567890abcdef0 |  active     |  19       |  my-
server-vm.vmdk   |                          |
|  My server VM|  import-snap-1234567890abcdef1 |  completed  |  None     |  my-
server-vm1.vmdk  |  snap-0bd3ea32600000000 |
|  My server VM|  import-snap-1234567890abcdef2 |  completed  |  None     |  my-
server-vm2.vmdk  |  snap-090ec0d0eb1111111 |
|  My server VM|  import-snap-1234567890abcdef3 |  deleted    |  None     |  my-
server-vm3.vmdk  |                          |
+--------------+--------------------------------+-------------+-----------
+--------------------+------------------------+
```

PowerShell

To get the status of an import snapshot task

Use the Get-EC2ImportSnapshotTask cmdlet as follows.

```
Get-EC2ImportSnapshotTask `
    -ImportTaskId import-snap-1234567890abcdef0 |
        Format-List *,
           @{Name='SnapshotTaskDetail';Expression={ $_.SnapshotTaskDetail | Out-
String }},
           @{Name='UserBucket';Expression={ $_.SnapshotTaskDetail.UserBucket | Out-
String }}
```

The following is example output.

```
Description        : My server VM
ImportTaskId       : import-snap-1234567890abcdef0
SnapshotTaskDetail : Amazon.EC2.Model.SnapshotTaskDetail
Tags               :
SnapshotTaskDetail :
                     Description   :
                     DiskImageSize : 2495933952
                     Encrypted     :
                     Format        : VMDK
                     KmsKeyId      :
                     Progress      :
                     SnapshotId    : snap-111222333444aaabb
                     Status        : completed
                     StatusMessage :
                     Url           :
                     UserBucket    : Amazon.EC2.Model.UserBucketDetails
UserBucket         :
                     S3Bucket : amzn-s3-demo-import-bucket
                     S3Key    : my-server-vm.vmdk
```

To get the status of all import snapshot tasks

Use the Get-EC2ImportSnapshotTask cmdlet as follows.

```
Get-EC2ImportSnapshotTask |
    Format-Table Description, ImportTaskId,
```

```
        @{Name='Status';Expression={$_.SnapshotTaskDetail.Status}},
        @{Name='Progress';Expression={$_.SnapshotTaskDetail.Progress}},
        @{Name='SnapshotID';Expression={$_.SnapshotTaskDetail.SnapshotID}},
        @{Name='S3Key Source';Expression={$_.SnapshotTaskDetail.UserBucket.S3Key}}
```

The following is example output. You can display any additional fields that you need.

```
Description  ImportTaskId                 Status    Progress SnapshotID
 S3Key Source
-----------  ------------                 ------    -------- ----------
 ------------
My server VM import-snap-1234567890abcdef0 active    19
 my-server-vm.vmdk
My server VM import-snap-1234567890abcdef1 completed          snap-0450e071240000000
 my-server-vm1.vmdk
My server VM import-snap-1234567890abcdef2 completed          snap-0bd3ea32601111111
 my-server-vm2.vmdk
My server VM import-snap-1234567890abcdef3 deleted
 my-server-vm3.vmdk
```

Cancel an import snapshot task

After you start an import snapshot task using VM Import/Export, you can cancel the import operation if needed.

To describe your snapshot import tasks, see Monitor an import snapshot task.

AWS CLI

To cancel an import snapshot task

Use the cancel-import-task command.

```
aws ec2 cancel-import-task \
    --import-task-id import-snap-1234567890abcdef0
```

PowerShell

To cancel an import snapshot task

Use the Stop-EC2ImportTask cmdlet.

```
Stop-EC2ImportTask `
    -ImportTaskId import-snap-1234567890abcdef0
```

Create an EBS volume from an imported snapshot

You can create EBS volumes from an EBS snapshot. You can attach an EBS volume to an EC2 instance.

AWS CLI

To create a volume and attach it to an EC2 instance

1. Use the describe-import-snapshot-tasks command to determine the ID of the snapshot that was created by the import task.
2. Use the following create-volume command to create a volume from the snapshot. You must select the Availability Zone of the instance to which you'll attach the volume.

   ```
   aws ec2 create-volume \
       --availability-zone us-east-1a \
       --snapshot-id snap-1234567890abcdef0
   ```

   The following is example output.

   ```
   {
       "AvailabilityZone": "us-east-1a",
       "VolumeId": "vol-1234567890abcdef0",
       "State": "creating",
       "SnapshotId": "snap-1234567890abcdef0"
   }
   ```

3. Use the following attach-volume command to attach the EBS volume that you created in the previous step to one of your existing instances.

   ```
   aws ec2 attach-volume \
       --volume-id vol-1234567890abcdef0 \
       --instance-id i-1234567890abcdef0 \
       --device /dev/sdf
   ```

   The following is example output.

```
{
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "attaching",
    "Device": "/dev/sdf"
}
```

4. Mount the attached volume. For more information, see the documentation for the operating system for your instance.

PowerShell

To create a volume and attach it to an EC2 instance

1. Use the Get-EC2ImportSnapshotTask cmdlet to determine the ID of the snapshot that was created by the import task.
2. Use the New-EC2Volume cmdlet to create a volume from the snapshot. You must select the Availability Zone of the instance to which you'll attach the volume.

```
New-EC2Volume `
    -AvailabilityZone us-east-1a `
    -SnapshotId snap-1234567890abcdef0
```

The following is example output.

```
Attachments        : {}
AvailabilityZone   : us-east-1a
CreateTime         : 7/15/2025 3:37:56 PM
Encrypted          : False
FastRestored       : False
Iops               : 3000
KmsKeyId           :
MultiAttachEnabled : False
Operator           :
OutpostArn         :
Size               : 41
SnapshotId         : snap-1234567890abcdef0
SseType            :
State              : creating
```

```
Tags                : {}
Throughput          : 125
VolumeId            : vol-1234567890abcdef0
VolumeType          : gp3
```

3. Use the Add-EC2Volume cmdlet

```
Add-EC2Volume `
    -VolumeId vol-1234567890abcdef0 `
    -InstanceId i-1234567890abcdef0 `
    -Device xvdb
```

The following is example output.

```
AssociatedResource    :
AttachTime            : 7/15/2025 3:47:20 PM
DeleteOnTermination   : False
Device                : xvdb
InstanceId            : i-1234567890abcdef0
InstanceOwningService :
State                 : attaching
VolumeId              : vol-1234567890abcdef0
```

4. Mount the attached volume. For more information, see the documentation for the operating system for your instance.

Export an EC2 instance as a VM using VM Import/Export

Exporting as a VM is useful when you want to deploy a copy of an Amazon EC2 instance in your virtualization environment. You can export most EC2 instances to Citrix Xen, Microsoft Hyper-V, or VMware vSphere.

When you export an instance, you are charged the standard Amazon S3 rates for the bucket where the exported VM is stored. In addition, there might be a small charge for the temporary use of an Amazon EBS snapshot. For more information about Amazon S3 pricing, see Amazon Simple Storage Service Pricing.

Contents

- Start an instance export task
- Monitor an instance export task
- Cancel an instance export task

Prerequisites for exporting an instance from Amazon EC2

To export a VM from Amazon EC2, first meet the following prerequisites:

- Create an Amazon S3 bucket for storing the exported instances or choose an existing bucket. The bucket must be in the Region where you want export your VMs. Additionally, the bucket must belong to the AWS account where you are performing the export operation. For more information, see the Amazon Simple Storage Service User Guide.
- You can't export a VM to an S3 bucket that uses the bucket owner enforced setting for S3 Object Ownership because ACLs are disabled. For more information, see Configuring ACLs in the *Amazon Simple Storage Service User Guide*.
- Prepare your S3 bucket by attaching an access control list (ACL) containing the following grants. For more information, see Managing access with ACLs in the *Amazon Simple Storage Service User Guide*.
  - For each `Grantee`, provide the following permissions:
    - `READ_ACP` (In the Amazon S3 console, **Bucket ACL** should have the **Read** permission)
    - `WRITE` (In the Amazon S3 console, **Objects** should have the **Write** permission)

Configure your S3 bucket

Console

To configure the S3 bucket

1. Open the Amazon S3 console at https://eusc-de-east-1.console.amazonaws-eusc.eu/s3/.
2. Select the bucket in which to store the exported instances.
3. On the **Permissions** tab, change the object ownership to **Bucket owner preferred**.
4. Attach the following bucket policy. For `CanonicalUser`, enter the canonical account ID for the bucket Region. For `Resource`, enter the name of your bucket in the bucket ARNs.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "GrantReadAclAndWrite",
            "Effect": "Allow",
            "Principal": {
                "CanonicalUser":
 "c4d8eabf8db69dbe46bfe0e517100c554f01200b104d59cd408e777ba442a322"
            },
            "Action": [
                "s3:GetBucketAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-export-bucket",
                "arn:aws:s3:::amzn-s3-demo-export-bucket/*"
            ]
        }
    ]
}
```

AWS CLI

To configure the S3 bucket

Use the put-bucket-ownership-controls command to change the object ownership.

```
aws s3api put-bucket-ownership-controls \
    --bucket amzn-s3-demo-export-bucket \
    --ownership-controls='{"Rules":[{"ObjectOwnership":"BucketOwnerPreferred"}]}'
```

Use the put-bucket-policy command to attach the bucket policy. For `CanonicalUser`, enter the canonical account ID for the bucket Region. For `Resource`, enter the name of your bucket in the bucket ARNs.

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-export-bucket \
```

```
    --policy \
'{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GrantReadAcpAndWrite",
            "Effect": "Allow",
            "Principal": {
                "CanonicalUser":
 "c4d8eabf8db69dbe46bfe0e517100c554f01200b104d59cd408e777ba442a322"
            },
            "Action": [
                "s3:GetBucketAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-export-bucket",
                "arn:aws:s3:::amzn-s3-demo-export-bucket/*"
            ]
        }
    ]
}'
```

PowerShell

To configure the S3 bucket

Use the Write-S3BucketOwnershipControl cmdlet to change the object ownership.

```
Write-S3BucketOwnershipControl `
    -BucketName "amzn-s3-demo-export-bucket" `
    -OwnershipControls_Rule @{ObjectOwnership="BucketOwnerPreferred"}
```

Use the Write-S3BucketPolicy cmdlet to attach the bucket policy. For `CanonicalUser`, enter the canonical account ID for the bucket Region. For `Resource`, enter the name of your bucket in the bucket ARNs.

```
Write-S3BucketPolicy `
    -BucketName "amzn-s3-demo-export-bucket" `
    -Policy `
'{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "GrantReadAcpAndWrite",
            "Effect": "Allow",
            "Principal": {
                "CanonicalUser":
 "c4d8eabf8db69dbe46bfe0e517100c554f01200b104d59cd408e777ba442a322"
            },
            "Action": [
                "s3:GetBucketAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-export-bucket",
                "arn:aws:s3:::amzn-s3-demo-export-bucket/*"
            ]
        }
    ]
}'
```

Considerations for instance export

Exporting instances and volumes is subject to the following limitations:

- You must export your instances and volumes to one of the following image formats that your virtualization environment supports:
  - Open Virtual Appliance (OVA), which is compatible with VMware vSphere versions 4, 5, and 6.
  - Virtual Hard Disk (VHD), which is compatible with Citrix Xen and Microsoft Hyper-V virtualization products.
  - Stream-optimized ESX Virtual Machine Disk (VMDK), which is compatible with VMware ESX and VMware vSphere versions 4, 5, and 6.
- You can't export an instance if it contains third-party software provided by AWS. For example, VM Export cannot export Windows or SQL Server instances, or any instance created from an image in the AWS Marketplace.
- You can't export an instance with encrypted EBS snapshots in the block device mapping.
- You can't export an instance with instance store volumes in the block device mapping.
- You can only export EBS volumes that are specified in the block device mapping, not EBS volumes attached after instance launch.

- You can't export an instance launched from an imported image if you deleted the AMI or the EBS snapshot for the AMI. To work around the issue, create an AMI from the instance and export the AMI.
- You can't export an instance that has more than one virtual disk.
- You can't export an instance that has more than one network interface.
- You can't export an instance from Amazon EC2 if you've shared it from another AWS account.
- By default, you can't have more than 5 conversion tasks per Region in progress at the same time. This limit is adjustable up to 20.
- VMs with volumes larger than 1 TiB are not supported.
- You can export a volume to either an unencrypted S3 bucket or to a bucket encrypted using SSE-S3. You cannot export to an S3 bucket encrypted using SSE-KMS.
- VM Import/Export only supports exporting VMs to an S3 bucket in the same AWS account that you export them from.
- Export operations do not support hybrid configurations. GRUB2 must be enabled for either BIOS or UEFI, but it can't be enabled for both.

Start an instance export task

When you export your instance using VM Import/Export, the exported file is written to the specified S3 bucket using the following S3 key:

```
prefixexport-i-xxxxxxxxxxxxxxxxx.format
```

For example, if the bucket name is `amzn-s3-demo-export-bucket`, the prefix is `vms/`, and the format is OVA, the exported file is written to `amzn-s3-demo-export-bucket/vms/export-i-1234567890abcdef0.ova`.

For more information about the supported formats, see the section called “Considerations for image export”.

> **Important**
>
> Your instance might reboot during the export process. Ensure that you are performing this action when some downtime is acceptable.

AWS CLI

To export an instance

Use the create-instance-export-task command.

```
aws ec2 create-instance-export-task \
    --description "$(date '+%b %d %H:%M') My instance export" \
    --instance-id i-1234567890abcdef0 \
    --target-environment vmware \
    --export-to-s3-task '{
        "ContainerFormat": "ova",
        "DiskImageFormat": "VMDK",
        "S3Bucket": "amzn-s3-demo-export-bucket",
        "S3Prefix": "vms/"
    }'
```

The following is an example response. The status shown is `active`, which means that the export task is in progress. The instance export is finished when the status is `completed`.

```
{
    "ExportTask": {
        "Description": "Jul 15 14:55 My instance export",
        "ExportTaskId": "export-i-021345abcdef6789",
        "ExportToS3Task": {
            "ContainerFormat": "ova",
            "DiskImageFormat": "vmdk",
            "S3Bucket": "amzn-s3-demo-export-bucket",
            "S3Key": "vms/export-i-021345abcdef6789.ova"
        },
        "InstanceExportDetails": {
            "InstanceId": "i-1234567890abcdef0",
            "TargetEnvironment": "vmware"
        },
        "State": "active"
    }
}
```

PowerShell

To export an instance

Use the New-EC2InstanceExportTask cmdlet.

```
New-EC2InstanceExportTask `
    -Description ((Get-Date -Format "MMM dd HH:mm ") + "My instance export") `
    -InstanceId "i-1234567890abcdef0" `
    -TargetEnvironment "vmware" `
    -ExportToS3Task_ContainerFormat "ova" `
    -ExportToS3Task_DiskImageFormat "VMDK" `
    -ExportToS3Task_S3Bucket "amzn-s3-demo-export-bucket" `
    -ExportToS3Task_S3Prefix "vms/"
```

The following is an example response. The status shown is `active`, which means that the export task is in progress. The instance export is finished when the status is `completed`.

```
Description           : Jul 15 14:53 My instance export
ExportTaskId          : export-i-021345abcdef6789
ExportToS3Task        : Amazon.EC2.Model.ExportToS3Task
InstanceExportDetails : Amazon.EC2.Model.InstanceExportDetails
State                 : active
StatusMessage         :
Tags                  : {}
```

Monitor an instance export task

After you start an instance export task using VM Import/Export, you can monitor the export operation.

AWS CLI

To monitor an instance export task

Use the following describe-export-tasks command.

```
aws ec2 describe-export-tasks \
    --export-task-ids export-i-1234567890abcdef0
```

The following is example output. The status shown is `active`. The VM is ready to use when the status is `completed`.

```
{
    "ExportTasks": [
        {
```

```
            "ExportTaskId": "export-i-1234567890abcdef0",
            "ExportToS3Task": {
                "ContainerFormat": "ova",
                "DiskImageFormat": "VMDK",
                "S3Bucket": "amzn-s3-demo-export-bucket",
                "S3Key": "vms/export-i-1234567890abcdef0.ova"
            },
            "InstanceExportDetails": {
                "InstanceId": "i-1234567890abcdef0",
                "TargetEnvironment": "vmware"
            },
            "State": "active"
        }
    ]
}
```

To monitor all instance export tasks

Use the following describe-export-tasks command.

```
aws ec2 describe-export-tasks \
    --query "ExportTasks[*].
{Description:Description,ExportTaskId:ExportTaskId,State:State,S3Bucket:ExportToS3Task.S3Buc
 \
    --output table
```

The following is example output. You can display any additional fields that you need.

```
-------------------------------------------------------------------------------------------
|                                                    DescribeExportTasks
                                                 |
+----------------------------------+-----------------------------
+----------------------+-----------------------------+------------+
|            Description           |        ExportTaskId         |      InstanceId
    |     S3Bucket                |    State   |
+----------------------------------+-----------------------------
+----------------------+-----------------------------+------------+
|  Jul 15 01:18 My instance export |  export-i-01234567890abaaaa |  None
    |  amzn-s3-demo-export-bucket |  active    |
|  Jul 15 11:01 My instance export |  export-i-01234567890abbbbb |  None
    |  amzn-s3-demo-export-bucket |  active    |
|  Jul 13 11:00 My instance export |  export-i-01234567890accccc |
 i-0abcdef1234567890 |  amzn-s3-demo-export-bucket |  completed |
```

```
+---------------------------------+----------------------------
+---------------------+----------------------------+------------+
```

PowerShell

To monitor an instance export task

Use the Get-EC2ExportTask cmdlet as follows.

```
Get-EC2ExportTask `
    -ExportTaskId export-i-1234567890abcdef0 |
        Format-List *,
           @{Name='ExportToS3Task';Expression={$_.ExportToS3Task | Out-string}},
           @{Name='InstanceExportDetails';Expression={$_.InstanceExportDetails |
 Out-string}}
```

The following is example output. The status shown is `active`. The VM is ready to use when the status is `completed`.

```
Description           : Jul 15 14:55 My instance export
ExportTaskId          : export-i-1234567890abcdef0
ExportToS3Task        : Amazon.EC2.Model.ExportToS3Task
InstanceExportDetails : Amazon.EC2.Model.InstanceExportDetails
State                 : completed
StatusMessage         :
Tags                  : {}
ExportToS3Task        :
                        ContainerFormat : ova
                        DiskImageFormat : VMDK
                        S3Bucket        : amzn-s3-demo-export-bucket
                        S3Key           : vms/export-i-1234567890abcdef0.ova
InstanceExportDetails :
                        InstanceId        : i-1234567890abcdef0
                        TargetEnvironment : vmware
```

To monitor all instance export tasks

Use the Get-EC2ExportTask cmdlet as follows.

```
Get-EC2ExportTask |
    Format-Table Description, ExportTaskId, State,
        @{Name='S3Bucket';Expression={$_.ExportToS3Task.S3Bucket}},
```

```
        @{Name='InstanceId';Expression={$_.InstanceExportDetails.InstanceId}}
```

The following is example output. You can display any additional fields that you need.

```
Description                    ExportTaskId               State     S3Bucket
              InstanceId
-----------                    ------------               -----     --------
              ----------
Jul 15 01:18 My instance export export-i-01234567890abaaaa active    amzn-s3-demo-
export-bucket
Jul 15 11:01 My instance export export-i-01234567890abbbbb active    amzn-s3-demo-
export-bucket
Jul 13 11:00 My instance export export-i-01234567890accccc completed amzn-s3-demo-
export-bucket   i-0abcdef1234567890
```

Cancel an instance export task

After you start an instance export task using VM Import/Export, you can cancel the export operation if needed. The cancel operation removes all artifacts of the export, including any partially created Amazon S3 objects. If the export task is complete or is in the process of transferring the final disk image, the cancel operation fails and returns an error.

To describe your instance export tasks, see Monitor an instance export task.

AWS CLI

To cancel an instance export task

Use the cancel-export-task command.

```
aws ec2 cancel-export-task \
    --export-task-id export-i-1234567890abcdef0
```

PowerShell

To cancel an instance export task

Use the Stop-EC2ExportTask cmdlet.

```
Stop-EC2ExportTask `
```

```
    -ExportTaskId export-i-1234567890abcdef0
```

Export a VM from an Amazon Machine Image (AMI) using VM Import/Export

Exporting a VM file based on an Amazon Machine Image (AMI) is useful when you want to deploy a new, standardized instance in your virtualization environment. You can export most AMIs to Citrix Xen, Microsoft Hyper-V, or VMware vSphere.

When you export an image, you are charged the standard Amazon S3 rates for the bucket where the exported VM is stored. In addition, there might be a small charge for the temporary use of an Amazon EBS snapshot. For more information about Amazon S3 pricing, see Amazon Simple Storage Service Pricing.

Contents

Prerequisites for exporting an image from Amazon EC2

To export a VM from Amazon EC2, first meet the following prerequisites.

- Install the AWS CLI. For more information, see the AWS Command Line Interface User Guide.

> **Tip**
>
> In supported AWS Regions, you can also use AWS CloudShell for a browser-based, pre-authenticated shell that launches directly from the AWS Management Console.

- Create an Amazon Simple Storage Service (Amazon S3) bucket for storing the exported images or choose an existing bucket. The bucket must be in the Region where you want to export your VMs. Additionally, the bucket must belong to the AWS account where you are performing the

export operation. For more information about S3 buckets, see the Amazon Simple Storage Service User Guide.

- Create an IAM role named `vmimport`. For more information, see Required service role.

Considerations for image export

Exporting images and volumes is subject to the following limitations:

- You must export to one of the following image formats that your virtualization environment supports:
  - Virtual Hard Disk (VHD), which is compatible with Citrix Xen and Microsoft Hyper-V virtualization products.
  - Stream-optimized ESX Virtual Machine Disk (VMDK), which is compatible with VMware ESX and VMware vSphere versions 4, 5, and 6.
  - Raw format.
- The base AMI used to launch an instance must exist when you attempt to export the instance. If you have deleted the AMI, the export fails.
- VM Import/Export only supports exporting VMs to an S3 bucket in the same AWS account that you export them from.
- Export operations do not support hybrid configurations. GRUB2 must be enabled for either BIOS or UEFI, but it can't be enabled for both.
- You can't export an image if it contains third-party software provided by AWS. For example, VM Export cannot export Windows or SQL Server images, or any image created from an image in the AWS Marketplace.
- You can't export an image with encrypted EBS snapshots in the block device mapping.
- You can only export EBS data volumes that are specified in the block device mapping, not EBS volumes attached after instance launch.
- You can't export an image from Amazon EC2 if you've shared it from another AWS account.
- You can't have multiple export image tasks in progress for the same AMI at the same time.
- By default, you can't have more than 5 conversion tasks per Region in progress at the same time. This limit is adjustable up to 20.
- VMs with volumes larger than 1 TiB are not supported.
- You can export a volume to either an unencrypted S3 bucket or to a bucket encrypted using SSE-S3 encryption. You cannot export to an S3 bucket encrypted using SSE-KMS encryption.

Start an export image task

When you export your image using VM Import/Export, the exported file is written to the specified S3 bucket using the following S3 key:

```
prefixexport-ami-xxxxxxxxxxxxxxxxx.format
```

For example, if the bucket name is `amzn-s3-demo-export-bucket`, the prefix is `exports/`, and the format is VMDK, the exported image is written to `amzn-s3-demo-export-bucket/exports/export-ami-1234567890abcdef0.vmdk`.

For information about the supported formats, see the section called “Considerations for image export”.

AWS CLI

To export an image

Use the export-image command.

```
aws ec2 export-image \
    --description "$(date '+%b %d %H:%M') My image export" \
    --image-id ami-1234567890abcdef0 \
    --disk-image-format VMDK \
    --s3-export-location S3Bucket=amzn-s3-demo-export-bucket,S3Prefix=exports/
```

The following is example output.

```
{
    "Description": "Jul 15 16:31 My image export",
    "DiskImageFormat": "VMDK",
    "ExportImageTaskId": "export-ami-36a041c1000000000",
    "ImageId": "ami-1234567890abcdef0",
    "Progress": "0",
    "S3ExportLocation": {
        "S3Bucket": "amzn-s3-demo-export-bucket",
        "S3Prefix": "exports/"
    },
    "Status": "active",
    "StatusMessage": "validating"
```

```
}
```

PowerShell

To export an image

Use the Export-EC2Image cmdlet.

```
Export-EC2Image `
    -Description ((Get-Date -Format "MMM dd HH:mm ") + "My image export") `
    -ImageId ami-1234567890abcdef0 `
    -DiskImageFormat VMDK `
    -S3ExportLocation_S3Bucket amzn-s3-demo-export-bucket `
    -S3ExportLocation_S3Prefix exports/
```

The following is example output.

```
Description       : Jul 15 16:35 My image export
DiskImageFormat   : VMDK
ExportImageTaskId : export-ami-36a041c1000000000
ImageId           : ami-1234567890abcdef0
Progress          : 0
RoleName          :
S3ExportLocation  : Amazon.EC2.Model.ExportTaskS3Location
Status            : active
StatusMessage     : validating
Tags              : {}
```

Monitor an export image task

After you start an image export using VM Import/Export, you can monitor the export operation.

AWS CLI

To monitor an export image task

Use the following describe-export-image-tasks command.

```
aws ec2 describe-export-image-tasks \
    --export-image-task-ids export-ami-1234567890abcdef0
```

The following is example output. The status shown is `active`, which means that the export task is in progress. The image is ready to use when the status is `completed`.

```
{
  "ExportImageTasks": [
      {
          "Description": "Jul 15 16:31 My image export",
          "ExportImageTaskId": "export-ami-1234567890abcdef0",
          "Progress": "21",
          "S3ExportLocation": {
              "S3Bucket": "amzn-s3-demo-export-bucket",
              "S3Prefix": "exports/"
          },
          "Status": "active",
          "StatusMessage": "updating"
      }
  ]
}
```

To monitor all export image tasks

Use the following describe-export-image-tasks command.

```
aws ec2 describe-export-image-tasks \
  --query "ExportImageTasks[*].{\
    Description:Description,\
    ExportImageTaskId:ExportImageTaskId,\
    ImageId:ImageId,\
    Status:Status,\
    Progress:Progress,\
    S3Bucket:S3ExportLocation.S3Bucket}" \
  --output table
```

The following is example output.

```
-------------------------------------------------------------------------------------------
|                                                    DescribeExportImageTasks
                                                          |
+-----------------------------+-------------------------------
+------------------------+-----------+-------------------------------+-------------+
|          Description         |       ExportImageTaskId       |        ImageId
    | Progress  |          S3Bucket            |   Status    |
```

```
+-----------------------------+-----------------------------
+-------------------------+-----------+-----------------------------+-------------+
|  Jul 15 16:35 My image export|  export-ami-1234567890abcdef0 |
    |  80       |  amzn-s3-demo-export-bucket  |  active     |
|  Jul 15 16:31 My image export|  export-ami-1234567890abcdef1 |  ami-
ab34567890abcdef0 |  None     |  amzn-s3-demo-export-bucket  |  completed  |
+-----------------------------+-----------------------------
+-------------------------+-----------+-----------------------------+-------------+
```

PowerShell

To monitor an export image task

Use the Get-EC2ExportImageTask cmdlet as follows.

```
Get-EC2ExportImageTask `
   -ExportImageTaskId export-ami-1234567890abcdef0 |
      Format-List *,
         @{Name='S3ExportLocation';Expression={$_.S3ExportLocation | Format-List |
 Out-String}}
```

The following is example output. The status shown is `active`, which means that the export task is in progress. The image is ready to use when the status is `completed`.

```
Description       : Jul 15 16:35 My image export
ExportImageTaskId : export-ami-1234567890abcdef0
ImageId           : ami-ab34567890abcdeff
Progress          : 80
S3ExportLocation  : Amazon.EC2.Model.ExportTaskS3Location
Status            : active
StatusMessage     : converting
Tags              : {}
S3ExportLocation  :
                    S3Bucket : amzn-s3-demo-export-bucket
                    S3Prefix : exports/
```

To monitor all export image tasks

Use the Get-EC2ExportImageTask cmdlet as follows.

```
Get-EC2ExportImageTask |
   Format-Table Description, ExportImageTaskId, ImageId, Status, Progress,
```

```
    @{Name='S3Bucket';Expression={$_.S3ExportLocation.S3Bucket}}
```

The following is example output.

```
Description                  ExportImageTaskId            ImageId
 Status    Progress S3Bucket
-----------                  -----------------            -------
 ------    -------- --------
Jul 15 16:35 My image export export-ami-1234567890abcdef0
 active    80       amzn-s3-demo-export-bucket
Jul 15 16:31 My image export export-ami-1234567890abcdef1 ami-ab34567890abcdef0
 completed          amzn-s3-demo-export-bucket
```

Cancel an export image task

After you start an image export using VM Import/Export, you can cancel the export operation if needed. If you attempt to cancel the export task after it is complete or is in the process of transferring the final disk image, the cancel operation fails and returns an error.

To describe your export image tasks, see Monitor an export image task.

AWS CLI

To cancel an export image task

Use the cancel-export-task command. If the command succeeds, no output is returned.

```
aws ec2 cancel-export-task \
    --export-task-id export-ami-1234567890abcdef0
```

PowerShell

To cancel an export image task

Use the Stop-EC2ExportTask cmdlet.

```
Stop-EC2ExportTask `
    -ExportTaskId export-ami-1234567890abcdef0
```

Security in VM Import/Export

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the .
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using VM Import/Export. It shows you how to configure VM Import/Export to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your VM Import/Export resources.

Topics

For more information about security and EC2 instances, Amazon Machine Images (AMI), and EBS volumes, see Security in Amazon EC2 in the *Amazon EC2 User Guide*.

Data protection in VM Import/Export

The AWS applies to data protection in VM Import/Export. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are

also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with VM Import/Export or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

VM Import/Export does not store your data at rest.

Encryption in transit

VM Import/Export encrypts your data while performing import tasks. To ensure that the destination AMI or snapshot is encrypted, specify the `--encrypted` parameter when you call the import-image or import-snapshot command.

When performing an import task, VM Import/Export stores data temporarily in an intermediate EBS volume. Each task gets a separate EBS volume. When an import task is completed, VM Import/Export deletes its intermediate EBS volume.

Compliance validation for VM Import/Export

Third-party auditors assess the security and compliance of VM Import/Export as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see . For general information, see .

Your compliance responsibility when using VM Import/Export is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.

Resilience in VM Import/Export

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in VM Import/Export

As a managed service, VM Import/Export is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access VM Import/Export through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Troubleshooting VM Import/Export

When you import or export a virtual machine (VM), most errors occur because of an attempt to do something that isn't supported. To avoid these errors, be sure to check the requirements and limitations carefully.

An import task might stop before it completes, and then fail. You can gather details about the import task that appears to have stopped due to a failure before it changes to the `completed` status. To gather such details, use the appropriate command for the import operation you used to describe details of the conversion task that's in progress:

- **ImportInstance** and **ImportVolume** – Use the DescribeConversionTasks operation.
- **ImportImage** – Use the DescribeImportImageTasks operation.
- **ImportSnapshot** – Use the DescribeImportSnapshotTasks operation.

Errors

- Import image errors
- Import instance errors
- VM export errors
- Windows VM errors
- Linux VM errors

Import image errors

Error Code: InvalidParameter, Error Message: Message: Parameter disk-image-size=0 has an invalid format

The specified image format is not supported. Retry the operation using one of the following supported image formats: VHD, VHDX, VMDK, or raw.

A client error (MalformedPolicyDocument) occurred when calling the CreateRole operation: Syntax errors in policy

You must include the `file://` prefix before the policy document name.

ClientError: Disk validation failed [OVF file parsing error: OVA with chunked disk files is not supported]

VM Import/Export does not support importing disks separated into multiple files. Check the disk format and retry the operation with the VM disk as a single file.

ClientError: Disk validation failed [Unsupported VMDK File Format]

The VMDK file must be stream-optimized. For more information, see Image formats supported by VM Import/Export.

ClientError: Multiple different grub/menu.lst files found

VM Import/Export found duplicate files during the import task for at least one of the following: `grub.cfg`, `grub.conf`, or `menu.lst`. VMs with dual-boot configurations are not supported. For more information, see Limitations for resources being imported with VM Import/Export.

The service role `vmimport` does not exist or does not have sufficient permissions for the service to continue

The VM import service role is missing or incorrect. You may also receive this error if the user, group, or role trying to start the import does not have sufficient access privileges on Amazon EC2 resources.

This error can also occur if the user calling `ImportImage` has `Decrypt` permission but the `vmimport` role does not. If you use Server-Side Encryption with AWS KMS–Managed Keys (SSE-KMS) to secure your at-rest data in Amazon S3, you need to assign additional `Decrypt` permission to your service role as shown in the following JSON code:

```
{
   "Sid":"Allow vmimport to decrypt SSE-KMS key",
   "Effect":"Allow",
   "Principal":{
      "AWS":[
         "arn:aws:iam::accountid:role/vmimport"
      ]
   },
   "Action":[
      "kms:Decrypt"
   ],
   "Resource":"*"
}
```

Import instance errors

Error Code: InvalidParameter, Error Message: Message: Parameter disk-image-size=0 has an invalid format

The specified image format is not supported. Retry the operation using one of the following supported image formats: OVA, VHD, VMDK, or raw.

Client.Unsupported: No bootable partition found. (Service: AmazonEC2; Status Code: 400; Error Code: Unsupported; Request ID: <RequestID>)

The root volume is GUID Partition Table (GPT) partitioned. GPT partitioned volumes are not supported. Convert the root volume to an MBR partition and try again.

ClientError: Footers not identical

You attempted to import a differencing VHD, or there was an error in creating the VHD. Export your VM again and retry importing it into Amazon EC2.

ClientError: Uncompressed data has an invalid length

The VMDK file is corrupted. You can try repairing or recreating the VMDK file, or use a different file.

ERROR: Bucket <MyBucketName> is not in the <RegionName> Region, it's in <RegionName>

The Amazon Simple Storage Service (Amazon S3) bucket is not in the same AWS Region as the instance you want to import. Try adding the `--ignore-region-affinity` option, which ignores whether the bucket's Region matches the Region where the import task is created. You can also create an S3 bucket using the Amazon Simple Storage Service console and set the Region to the Region where you want to import the VM. Run the command again and specify the new bucket you just created.

ERROR: File uses unsupported compression algorithm 0

The VMDK was created using OVA format instead of OVF format. Create the VMDK in OVF format.

Invalid S3 source location

The command syntax or S3 bucket name is incorrect. Create an S3 bucket in the appropriate Region solely for VM Import and upload the VM files to the root of the bucket.

The given S3 bucket is not local to the Region

The S3 bucket used for VM Import must reside in the same AWS Region where you want to import the VM.

ClientError: Unknown OS / Missing OS files

The operating system is not recognized. Verify that your OS is listed as support in the VM Import/Export Requirements for resources that you import with VM Import/Export.

VM export errors

Client.UnsupportedOperation: This instance has multiple volumes attached. Please remove additional volumes.

Detach volumes other than the root volume and try again. If you need the data from the volumes, you can copy it to the root volume or import the volumes to Amazon EBS.

Client.NotExportable: This instance cannot be exported. (Service: AmazonEC2; Status Code: 400; Error Code: NotExportable; Request ID: <RequestID>)

You can only export certain instances. For more information, see Considerations for instance export.

Error starting instances: Invalid value <instance ID> for instanceId. Instance does not have a volume attached at root (/dev/sda1).

You attempted to start the instance before the VM import process and all conversion tasks were complete. Wait for the VM import process and all conversion tasks to completely finish, and then start the instance.

An error occurred (InvalidParameter) when calling the CreateInstanceExportTask operation: The given S3 object is not local to the region.

The EC2 instance and S3 bucket must be in the same AWS Region. You must also ensure the `create-instance-export-task` command is being run in the same Region as your resources being exported. You can specify the Region by using `--region` parameter. For more information, see AWS CLI supported global command line options in the *AWS Command Line Interface User Guide*.

Windows VM errors

ClientError: Booter Networking failure/instance not reachable. Please retry after installation of .Net framework 3.5 SP1 or greater.

The EC2 Config Service requires the Microsoft .NET Framework 3.5 Service Pack 1 or later. Install Microsoft .NET Framework 3.5 Service Pack 1 or later on your Windows VM and try again.

FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.

When you receive the `FirstBootFailure` error message, it means that your virtual disk image was unable to perform one of the following steps:

- Boot up and start Windows.
- Install Amazon EC2 networking and disk drivers.
- Use a DHCP-configured network interface to retrieve an IP address.
- Activate Windows using the Amazon EC2 Windows volume license.

The following best practices can help you to avoid Windows first boot failures:

- **Disable anti-virus and anti-spyware software and firewalls** — These types of software can prevent installing new Windows services or drivers or prevent unknown binaries from running. Software and firewalls can be re-enabled after importing.
- **Do not harden your operating system** — Security configurations, sometimes called hardening, can prevent unattended installation of Amazon EC2 drivers. There are numerous Windows configuration settings that can prevent import. These settings can be reapplied once imported.
- **Disable or delete multiple bootable partitions** — If your virtual machine boots and requires you to choose which boot partition to use, the import may fail.

This inability of the virtual disk image to boot up and establish network connectivity could be due to any of the following causes:

TCP/IP networking and DHCP are not enabled

Cause: TCP/IP networking and DHCP must be enabled.

Resolution: Ensure that TCP/IP networking is enabled. For more information, see Change TCP/IP settings at the Microsoft Support website. Ensure that DHCP is enabled. For more information, see Dynamic Host Configuration Protocol (DHCP) at the Microsoft website.

The Hyper-V server role is installed

Cause: Importing a virtual machine with the Hyper-V role installed is not supported.

Resolution: Remove the Hyper-V role from the virtual machine and try the import again.

A volume that Windows requires is missing from the virtual machine

Cause: Importing a VM into Amazon EC2 only imports the boot disk, all other disks must be detached and Windows must able to boot before importing the virtual machine. For example, Active Directory often stores the Active Directory database on the `D:\` drive. A domain controller cannot boot if the Active Directory database is missing or inaccessible.

Resolution: Detach any secondary and network disks attached to the Windows VM before exporting. Move any Active Directory databases from secondary drives or partitions onto the primary Windows partition. For more information, see "Directory Services cannot start" error message when you start your Windows-based or SBS-based domain controller at the Microsoft Support website.

Windows always boots into System Recovery Options

Cause: Windows can boot into System Recovery Options for a variety of reasons, including when Windows is pulled into a virtualized environment from a physical machine, also known as a physical-to-virtual (P2V) conversion process.

Resolution: Ensure that Windows boots to a login prompt before exporting and preparing for import. Do not import virtualized Windows instances that have come from a physical machine.

The virtual machine was created using a physical-to-virtual (P2V) conversion process

Cause: A P2V conversion occurs when a disk image is created by performing the Windows installation process on a physical machine and then importing a copy of that Windows installation into a VM. VMs that are created as the result of a P2V conversion are not supported by VM Import/Export. VM Import/Export only supports Windows images that were natively installed inside the source VM.

Resolution: Install Windows in a virtualized environment and migrate your installed software to that new VM.

Windows activation fails

Cause: During boot, Windows will detect a change of hardware and attempt activation. During the import process we attempt to switch the licensing mechanism in Windows to a volume license provided by Amazon Web Services. However, if the Windows activation process does not succeed, then the import fails.

Resolution: Ensure that the version of Windows that you are importing supports volume licensing. Beta or preview versions of Windows might not.

No bootable partition found

Cause: During the import process of a virtual machine, we could not find the boot partition.

Resolution: Ensure that the disk you are importing has a boot partition.

Linux VM errors

ClientError: Invalid configuration - Could not read fstab

Linux VMs with dual-boot volumes or multiple `/etc` directories are not supported.

ClientError: BLSC-style GRUB found, but unable to detect default kernel

VM Import/Export can't detect the default kernel. This can occur when it has been moved out of the main `grub.cfg` file. You can set the configuration to `$saved_entry` and ensure the `grubenv` contains the `bootloader` entry as the default.

ClientError: We were unable to read your import's initramfs/initrd to determine what drivers your import requires to run in EC2

We were unable to read required files while importing your Linux VM to prepare it to run as an instance in Amazon EC2. You can run the `lsinitramfs` command to verify the integrity of the file. For example, you might use the following command:

```
lsinitramfs /boot/initrd.img-5.4.0-77-generic 2>&1 | less
```

If errors are returned in the output, you can try rebuilding the `initramfs` file to resolve the issue and import the VM again.

ClientError: Unsupported configuration - Logical volume group activation failed

A logical volume on your virtual disk image failed to activate. This may indicate file or disk corruption. Verify the uploaded disk image files.

ClientError: Unsupported configuration - Multiple directories found

Linux VMs with multi-boot volumes or multiple `/etc` directories are not supported.

ClientError: Unsupported kernel version

The kernel version used by the operating system is not supported. Confirm that your import meets the requirements listed for the operating system. For more information, see Operating systems supported by VM Import/Export.

Linux is not supported on the requested instance

Linux VMs can be imported to specific instance types. Try again using one of the following supported instance types.

- General purpose: `t2.micro` | `t2.small` | `t2.medium` | `m3.medium` | `m3.large` | `m3.xlarge` | `m3.2xlarge`
- Compute optimized: `c3.large` | `c3.xlarge` | `c3.2xlarge` | `c3.4xlarge` | `c3.8xlarge` | `cc1.4xlarge` | `cc2.8xlarge`
- Memory optimized: `r3.large` | `r3.xlarge` | `r3.2xlarge` | `r3.4xlarge` | `r3.8xlarge` | `cr1.8xlarge`
- Storage optimized: `i2.xlarge` | `i2.2xlarge` | `i2.4xlarge` | `i2.8xlarge` | `hi1.4xlarge` | `hi1.8xlarge`

Document history for VM Import/Export

The following table describes important additions to the VM Import/Export documentation after August 2019. For notification about updates to this documentation, you can subscribe to the RSS feed.

Change	Description	Date
VM Import/Export supports more Red Hat Enterprise Linux (RHEL), Rocky Linux, and Oracle Linux operating systems.	VM Import/Export added support for Red Hat Enterprise Linux (RHEL) 9.6 with kernel 5.14.0, Rocky Linux 9.6 with kernel 5.14.0, and Oracle Linux 9.6 with Red Hat Compatible Kernel (RHCK) 6.12.0 and Unbreakable Enterprise Kernel (UEK) 6.12.0. For more information, see Operating systems.	July 17, 2025
VM Import/Export supports more Red Hat Enterprise Linux (RHEL), Rocky Linux, and Oracle Linux operating systems.	VM Import/Export added support for Red Hat Enterprise Linux (RHEL) 9.5 with kernel 5.15.0, Rocky Linux 9.5 with kernel 5.15.0, and Oracle Linux 9.5 with kernel 5.15.0. For more information, see Operating systems.	June 11, 2025
VM Import/Export is available in the Asia Pacific (Malaysia) Region	VM Import/Export is now available in the Asia Pacific (Malaysia) Region.	August 21, 2024
VM Import/Export supports more Oracle Linux, Red Hat Enterprise Linux (RHEL),	VM Import/Export added support for Oracle Linux 8.9 with the Red Hat Compatible Kernel (RHCK) 4.18.0 and	June 26, 2024

and Rocky Linux operating systems.	Unbreakable Enterprise Kernel (UEK) 5.15.0 (el8uek) kernels, Oracle Linux 9.3–9.4 with the Red Hat Compatible Kernel (RHCK) 5.14.0 and Unbreakable Enterprise Kernel (UEK) 5.15.0 (el9uek) kernels, RHEL 8.9 with the 4.18.0 kernel, RHEL 9.3–9.4 with the 5.14.0 kernel, and Rocky Linux 9.1–9.4 with the 5.14.0 kernel. For more information, see Operating Systems.	
VM Import/Export supports UEFI boot mode in more AWS Regions	VM Import/Export supports UEFI boot in all of the commercial AWS Regions. For more information, see Boot modes and Region in the *AWS Glossary*.	April 18, 2024
VM Import/Export supports more Debian and Fedora Linux operating systems	VM Import/Export added support for Debian 12.2 and Debian 12.4 with kernel 6.1.0 operating systems. VM Import/Export added support for Fedora Linux 37 with kernel 6.0.7, Fedora Linux 38 with kernel 6.2.9, and Fedora Linux 39 with kernel 6.5.6 operating systems. For more information, see Operating Systems.	January 25, 2024

VM Import/Export is available in the Canada West (Calgary) Region	VM Import/Export is now available in the Canada West (Calgary) Region.	December 20, 2023
VM Import/Export supports more Oracle Linux operating systems	VM Import/Export added support for Oracle Linux 8.0–8.8 with kernel 4.18.0 and Oracle Linux 9.0–9.2 with kernel 5.14.0 operating systems. For more informati on, see Operating Systems.	December 18, 2023
VM Import/Export supports more SLES kernels	VM Import/Export added support for the SLES 5.14.21 kernel with service packs 4 and 5. For more information, see Operating Systems.	December 1, 2023
VM Import/Export supports more Windows operating systems	VM Import/Export added support for the Windows Server 2022 operating system. For more information, see Operating Systems.	September 26, 2023
VM Import/Export supports more RHEL operating systems	VM Import/Export added support for Red Hat Enterpris e Linux (RHEL) 8.7 and 8.8 operating systems with kernel 4.18.0. For more information, see Operating Systems.	September 1, 2023
VM Import/Export added support for the Rocky Linux operating system	VM Import/Export added support for the Rocky Linux 9 operating system. For more information, see Operating Systems.	September 1, 2023

VM Import/Export is available in the Israel (Tel Aviv) Region	VM Import/Export is now available in the Israel (Tel Aviv) Region.	August 1, 2023
VM Import/Export supports more Ubuntu operating systems	VM Import/Export added support for the Ubuntu 23.04 operating system with kernel 5.15.0. For more information, see Operating Systems.	May 30, 2023
VM Import/Export is available in the Asia Pacific (Melbourne) Region	VM Import/Export is now available in the Asia Pacific (Melbourne) Region.	January 24, 2023
VM Import/Export supports more SLES operating systems	VM Import/Export added support for the SUSE Linux Enterprise Server (SLES) 15 operating system with service pack 3 and kernel 5.3. For more information, see Operating Systems.	December 15, 2022
VM Import/Export is available in the Asia Pacific (Hyderabad) Region	VM Import/Export is now available in the Asia Pacific (Hyderabad) Region.	November 22, 2022
VM Import/Export supports more Ubuntu operating systems	VM Import/Export added support for the Ubuntu 22.04 operating system with kernel 5.15.0. For more information, see Operating Systems.	November 18, 2022
VM Import/Export is available in the Europe (Spain) Region	VM Import/Export is now available in the Europe (Spain) Region.	November 16, 2022

VM Import/Export is available in the Europe (Zurich) Region	VM Import/Export is now available in the Europe (Zurich) Region.	November 9, 2022
VM Import/Export supports more RHEL operating systems	VM Import/Export added support for Red Hat Enterprise Linux (RHEL) 8.3, 8.4, 8.5, and 8.6 operating systems with kernel 4.18.0. For more information, see Operating Systems.	October 19, 2022
VM Import/Export supports more Windows operating systems	VM Import/Export added support for the Windows 11 operating system. For more information, see Operating Systems.	August 2, 2022
VM Import/Export supports more SLES operating systems	VM Import/Export added support for more SUSE Linux Enterprise Server (SLES) 12 and 15 operating systems. SLES 12 with service pack 4 and kernel 4.12, SLES 12 with service pack 5 and kernel 4.12, SLES 15 without any service pack and kernel 4.12, SLES 15 with service pack 1 and kernel 4.12, and SLES 15 with service pack 2 and kernel 5.3 are now supported. For more information, see Operating Systems.	February 28, 2022
VM Import/Export is available in the Middle East (UAE) Region	VM Import/Export is now available in the Middle East (UAE) Region.	December 13, 2021

VM Import/Export is available in the Asia Pacific (Jakarta) Region	VM Import/Export is now available in the Asia Pacific (Jakarta) Region.	December 13, 2021
VM Import/Export supports more Red Hat Enterprise Linux (RHEL) and CentOS operating systems	VM Import/Export added support for RHEL and CentOS 8.0, 8.1, and 8.2 operating systems. For more informati on, see Operating Systems.	July 17, 2020
VM Import/Export is available in the Europe (Milan) Region	VM Import/Export is now available in the Europe (Milan) Region.	April 28, 2020

Earlier updates

The following table describes important additions to the VM Import/Export documentation in 2019 and earlier years.

Change	Description	Date
Export a VM from an AMI	Added support for exporting a VM file based on an Amazon Machine Image (AMI).	August 23, 2019
Import VMs with multiple volumes as images	Added support for importing VMs as an Amazon Machine Image (AMI) using the ImportImage API. ImportIns tance also supports importing VMs with multiple volumes. The new API improves performance and flexibility.	April 23, 2015
Import Linux virtual machines	Added support for importing Linux instances.	December 16, 2013

Change	Description	Date
Export a VM from an instance	Added support for exporting Windows Server instances that you originally imported into Amazon EC2. Added support for exporting Linux instances to Citrix Xen, Microsoft Hyper-V, and VMware vSphere.	May 25, 2012
Import in VHD file format	Added support for importing virtual machine image files in VHD format. With this release, VM Import now supports RAW, VHD, and VMDK (VMware ESX-compa tible) image formats.	August 24, 2011