

Grandstream Networks, Inc.

SoftwareUCM Installation Guide - AWS



SoftwareUCM is a powerful communication management system that provides businesses with control over their telephony infrastructure. By deploying SoftwareUCM on AWS, organizations can take advantage of cloud scalability, flexibility, and reliability. This setup eliminates the need for on-premises hardware, allowing for easier management, automated backups, and increased resilience.

This guide provides a step-by-step walk-through for setting up a virtual machine on AWS, importing the SoftwareUCM image, and configuring it for optimal performance.

Prerequisites

Before proceeding with the instructions in this guide, users should ensure the following criteria are met:

- An **AWS account** with appropriate permissions to create and manage EC2 instances, S3 storage, and IAM roles.
- Ensure that you have a **SoftwareUCM virtual machine** set up and ready to be exported.

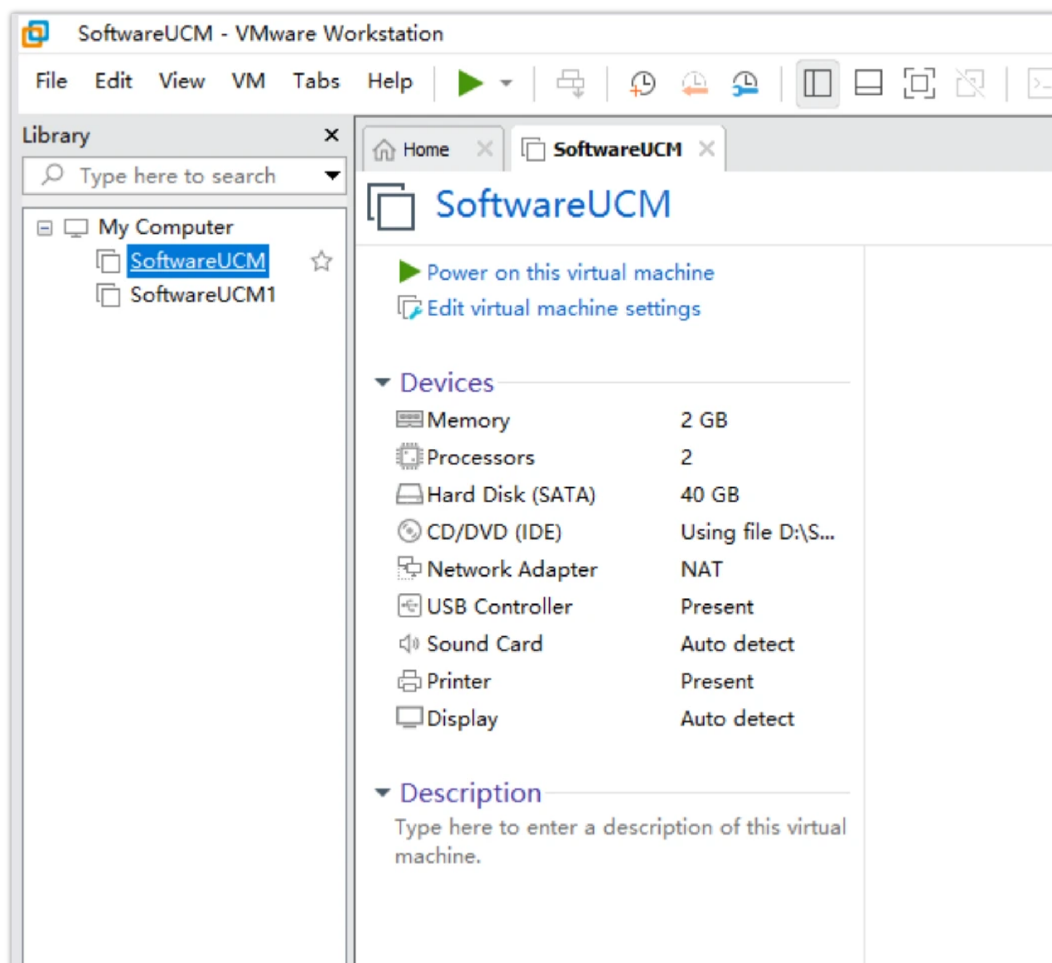
Export the SoftwareUCM Virtual Machine from VMware

Before deploying SoftwareUCM on AWS, start by exporting your virtual machine from your hypervisor.

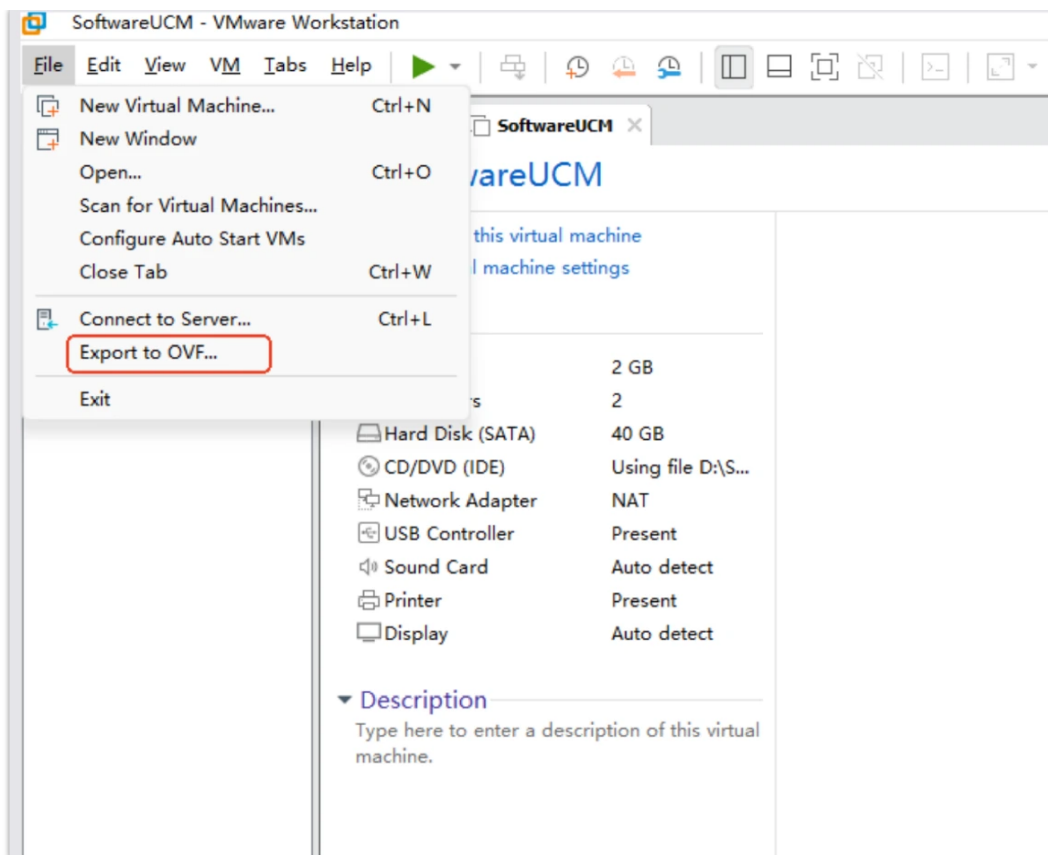
Note:

This guide uses VMware Workstation as an example, but any supported hypervisor can be used.

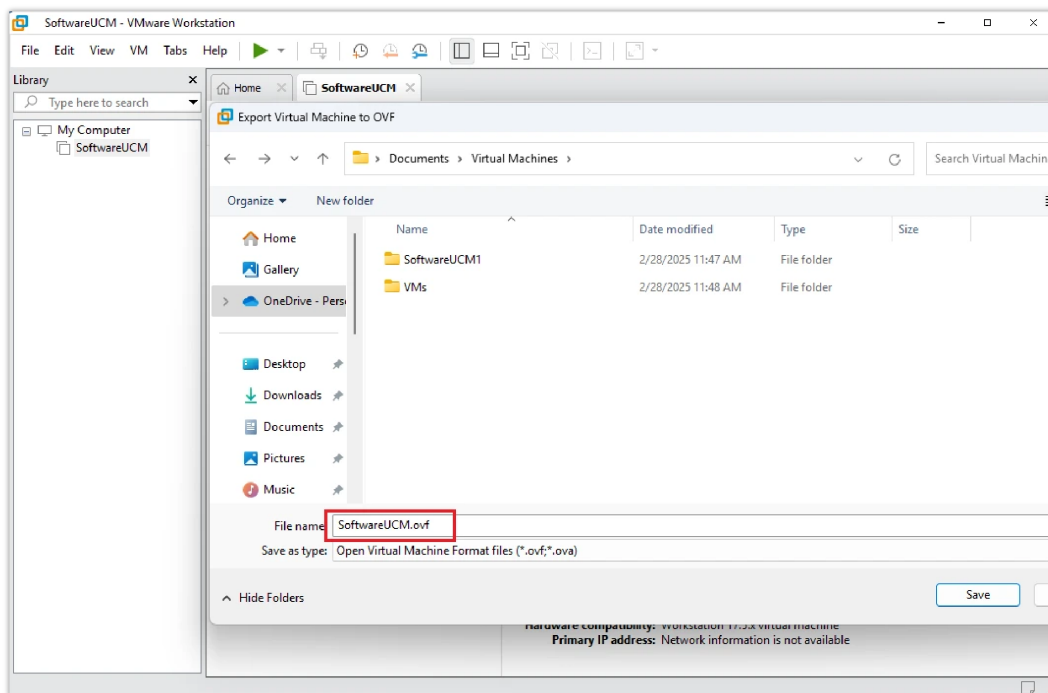
1. Open the VMware Workstation and locate the SoftwareUCM virtual machine you want to export.



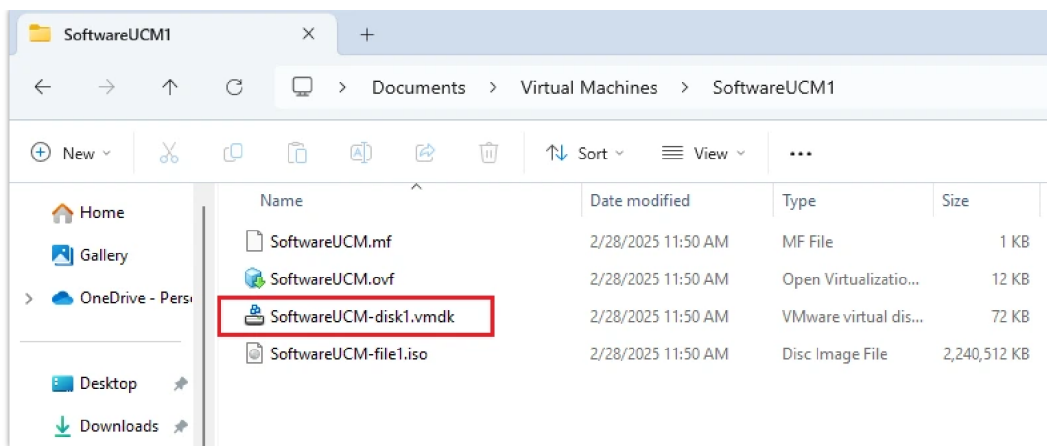
2. Right-click on the VM and choose **Export OVF Template**.



3. Select the **OVA** format and specify a location to save the file.



4. Locate the virtual machine in **.vmdk** format. This file will be used to create an AMI (Amazon Machine Image) in AWS.



Create an AWS S3 Bucket

1. Log in to your AWS Management Console.
2. Navigate to the **Amazon S3** service and click on **Create bucket**.
3. Provide a unique name for your bucket and configure the settings as shown below.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

Amazon Web Services Region
China (Beijing) cn-north-1

Bucket name [Info](#)
softucm-shhwang-test

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)
Format: s3://bucket/prefix

Object Ownership [Info](#)
Control ownership of objects written to this bucket from other Amazon Web Services accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**
Objects in this bucket can be owned by other Amazon Web Services accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. Amazon Web Services recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☒ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☒ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☒ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☒ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

- ☒ **Server-side encryption with Amazon S3 managed keys (SSE-S3)**
- ☐ **Server-side encryption with Amazon Key Management Service keys (SSE-KMS)**
- ☐ **Dual-layer server-side encryption with Amazon Key Management Service keys (DSSE-KMS)**
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to Amazon KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ **Disable**

☒ **Enable**

Advanced settings

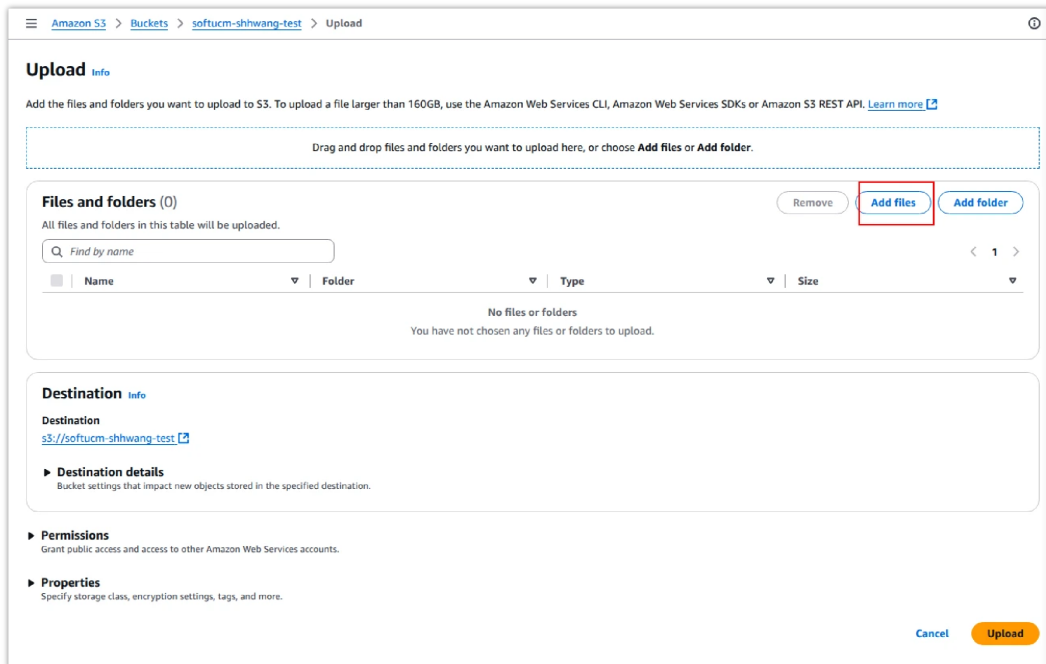
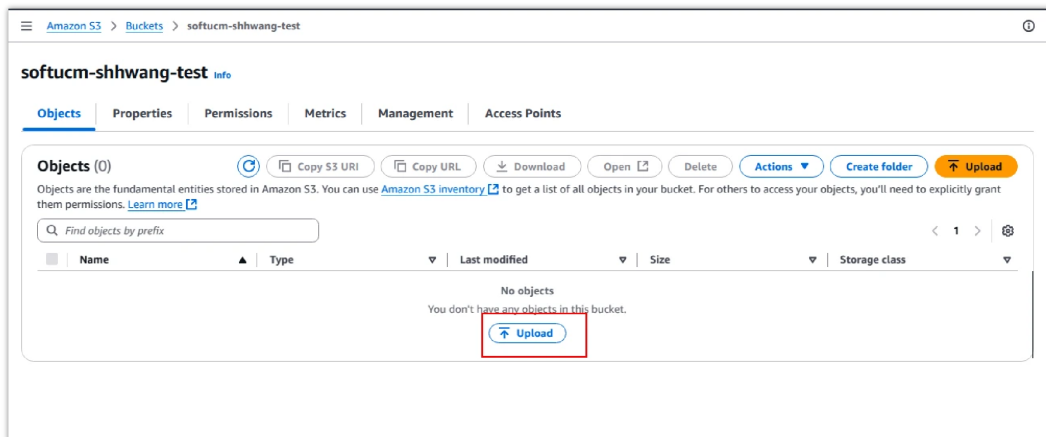
[Info](#) After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

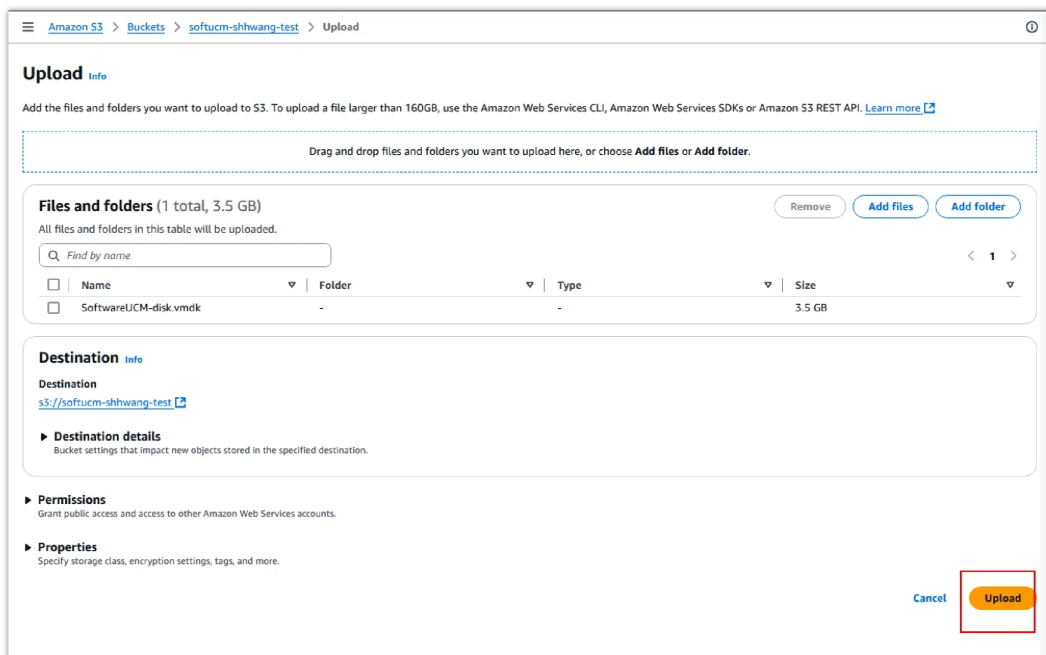
4. Configure the bucket settings as needed and click **Create bucket**.

Upload the .vmdk File to the S3 Bucket

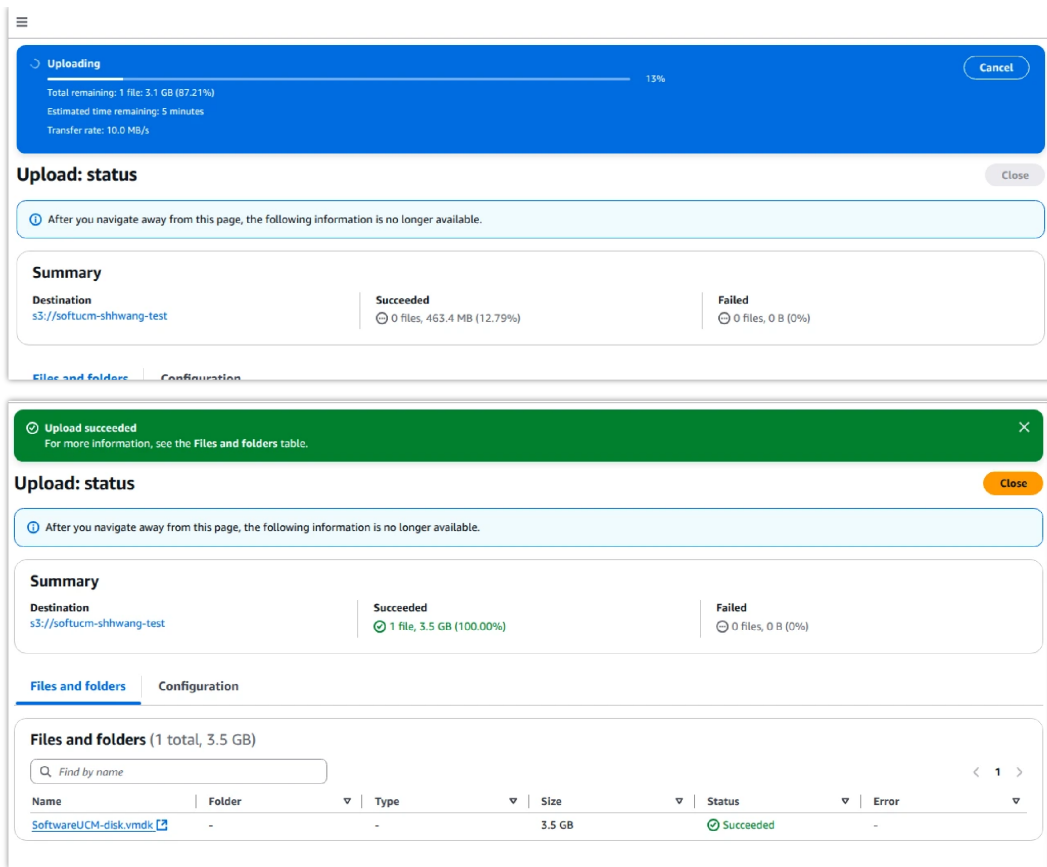
1. In the S3 console, locate the bucket you just created.
2. Click on the bucket name, select **Upload** then click **Add files**.



3. Select the .vmdk file you exported from VMware and click **Upload**.

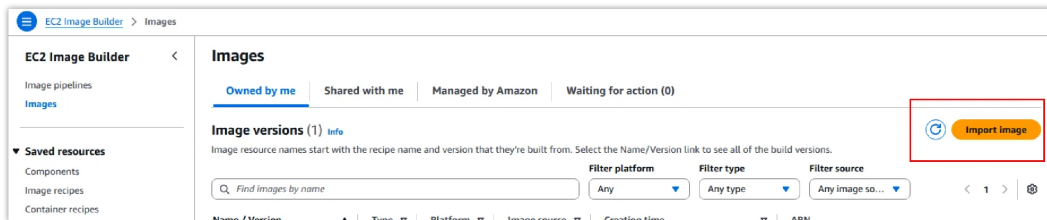


4. Wait for the upload to complete. This may take some time depending on the size of the file.



Import the Image Using AWS EC2 Image Builder

1. Navigate to the **EC2 Image Builder** service in the AWS Management Console.
2. Click on **Import Image**.



2. Select the **.vmdk** file from the S3 bucket you uploaded earlier.

EC2 Image Builder > Images > Import image

Import image

General
Enter a name and description for the base image that is being imported.

Name: Version:
Maximum of 120 characters. Letters, numbers, spaces, -, and _ are allowed. Format: major.minor.patch

Description - optional

Maximum 1024 characters.

Base image operating system
Image Operating System (OS)
Image Builder supports Amazon Linux, Windows, Ubuntu, CentOS, RHEL, and SLES.

OS version

VM import configuration
Specify the location of your import source, and optionally configure security, encryption, licensing, and other settings to transform your VM source into your base image.

Import source | [Info](#)
Import disk container files (created when you export your VM from its virtual environment) as the source for your Image Builder image.

▼ **Disk container 1** Remove

Source
☒ S3 bucket ☐ Snapshot ID

S3 URI

Enter the location in Amazon S3 where your disk images are stored.

Description - optional

Maximum of 255 characters.

Choose an object in S3 ×

S3 buckets > softucm-shhwang-test

Objects (1/1)

Key	Last modified	Size
☒ SoftwareUCM-disk.vmdk	February 24, 2025, 01:50 (UTC+00:00)	3.54 GB

- In the **IAM Role** section, select **vmimport**. This is the default IAM role for importing images. Ensure that this role has the necessary permissions to access the S3 bucket where the **.vmdk** file is stored.

VM import configuration
Specify the location of your import source, and optionally configure security, encryption, licensing, and other settings to transform your VM source into your base image.

Import source | [Info](#)
Import disk container files (created when you export your VM from its virtual environment) as the source for your Image Builder image.

▼ **Disk container 1** Remove

Source
☒ S3 bucket ☐ Snapshot ID

S3 URI

Enter the location in Amazon S3 where your disk images are stored.

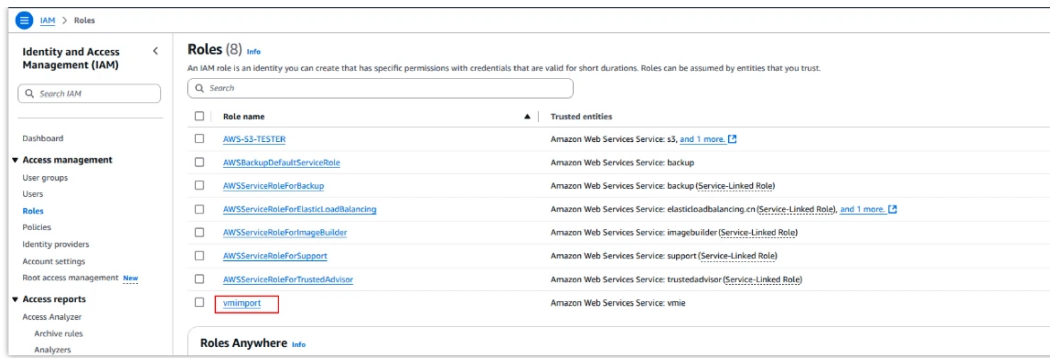
Description - optional

Maximum of 255 characters.

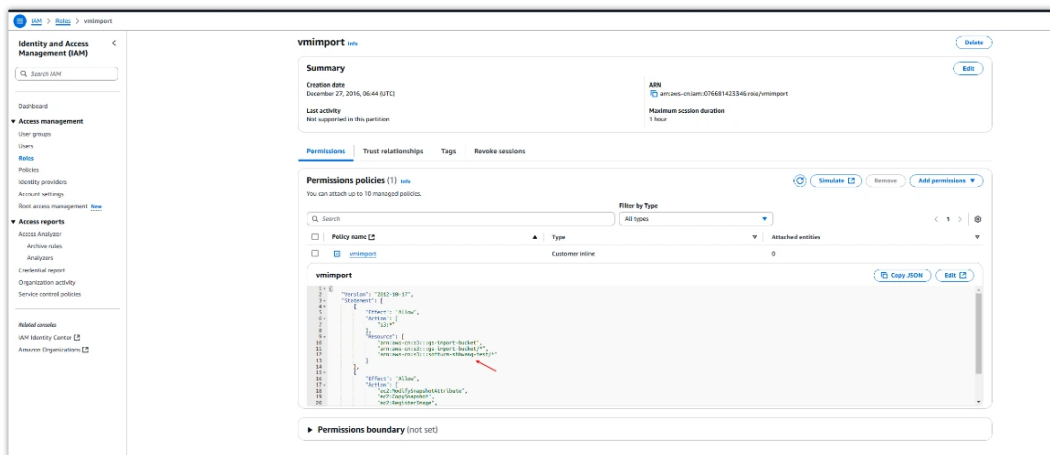
IAM role | [Info](#)
Specify an IAM role to use during the import process, or choose Create new role to create a new one.

► **Advanced settings - optional**
You can define the Base Image encryption, License configuration settings and custom name for this image.

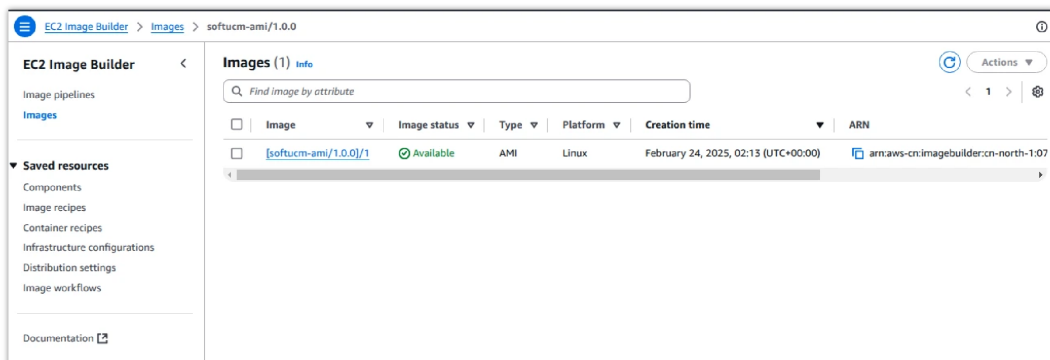
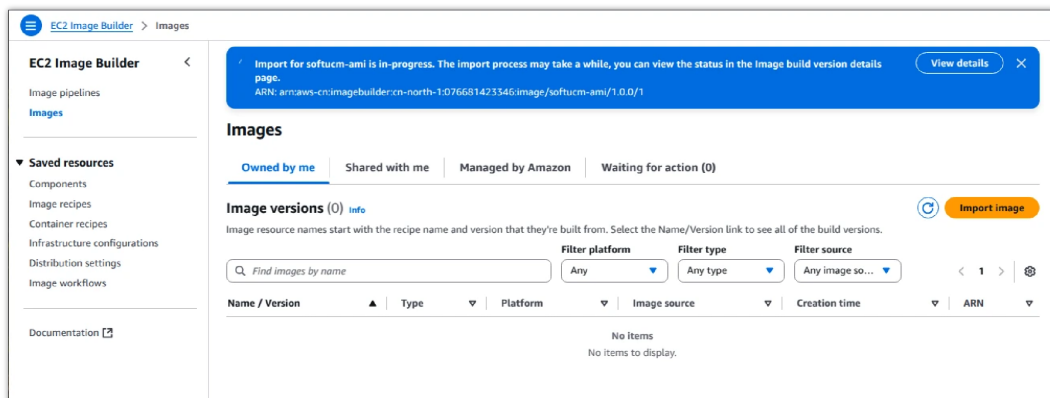
- To verify that the role has the necessary permissions, search for **IAM** in the AWS Management Console and access it.
- In the IAM dashboard, click on **Roles** in the left-hand menu. Search for the **vmimport** role in the list of roles.



- Click on the **vmimport** role to view its details. Under the **Permissions** tab, you will see the policies attached to the role. Ensure that the role has the following permissions

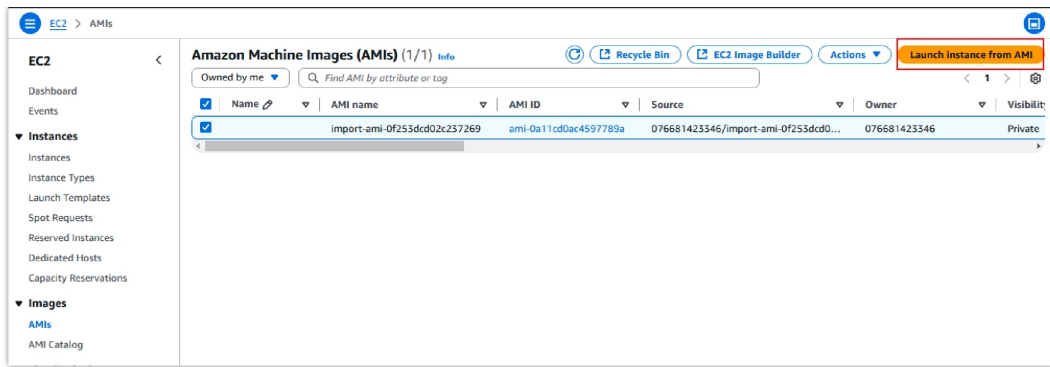


- Click **Import** and wait for the image creation process to complete. This may take some time.

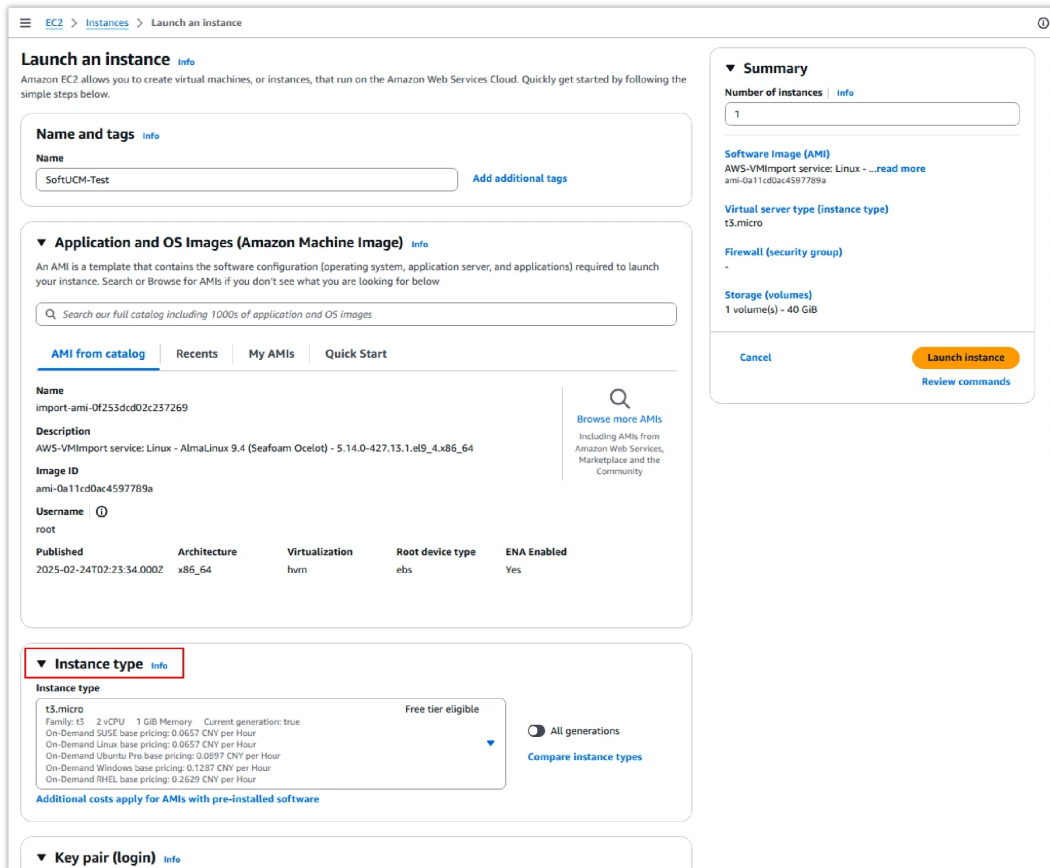


Launch an EC2 Instance from the AMI

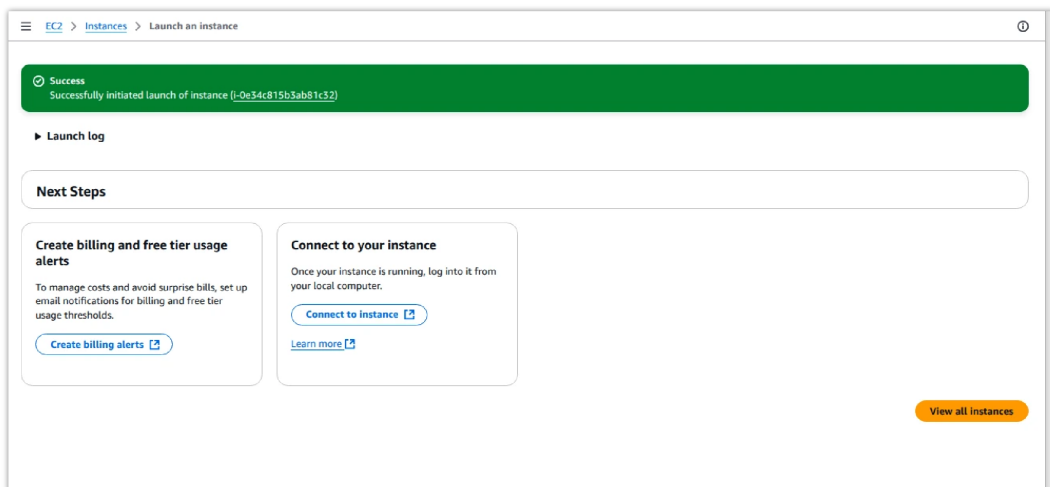
- Once the image import is complete, navigate to the **EC2** service in the AWS Management Console.
- Click on **Launch Instance** and select the AMI you just created.



3. Choose an instance type that meets your performance requirements and configure the instance settings as needed.



4. Launch the instance and wait for it to initialize.



Access SoftwareUCM on AWS

Once the EC2 instance is running, you can access the SoftwareUCM system through the AWS Management Console.

EC2 > Instances > i-0e34c815b3ab81c32 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0e34c815b3ab81c32 (SoftUCM-Test) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID

i-0e34c815b3ab81c32 (SoftUCM-Test)

Serial port

ttyS0

Cancel

Connect

```
SoftwareUCM login: support
Password:
Last failed login: Mon Feb 24 06:43:05 UTC 2025 on ttyS0
There were 5 failed login attempts since the last successful login.
Last login: Thu Feb 20 17:11:50 on tty1

Please enter select:
[1] SoftwareUCM Management.
[2] System Status.
[3] System Settings.
[4] Maintenance.
[5] API Configuration.
[0] Exit.
|
```

Note:

Due to AWS network limitations, the Multi-Tenant mode is not supported when running SoftwareUCM on AWS.