Create Custom AMIs

**Contents**

[Scope 2](#_Toc115190395)

[Problem Statement 2](#_Toc115190396)

[Introduction 2](#_Toc115190397)

[Overview 2](#_Toc115190398)

[AMI Lifecycle 3](#_Toc115190399)

[How the creation of a custom AMI works 4](#_Toc115190400)

[Create a Windows AMI from a running instance 4](#_Toc115190401)

[Implementation 5](#_Toc115190402)

[Creating Custom AMI from an instance - Console: 5](#_Toc115190403)

[Creating Custom AMIs from EC2 Image Builder 11](#_Toc115190404)

[EC2 Image Builder 11](#_Toc115190405)

[Step1: create image pipeline 12](#_Toc115190406)

[Step2: Recipe 14](#_Toc115190407)

[Step 3: Define Infrastructure configuration (optional) 19](#_Toc115190408)

[Step 4: Define distribution settings (optional) 20](#_Toc115190409)

[Step 5: Review 22](#_Toc115190410)

## Step 6: Clean up …………………………………………………………………………….26

[Creating Custom AMI using AWS - CLI 28](#_Toc115190404)

[Use Case 31](#_Toc115190411)

[Python installed Windows AMI](#_Toc115190412) 31

[OpenLDAP Server configured Ubuntu AMI](#_Toc115190412) 34

[Advantages 37](#_Toc115190413)

[Limitations 37](#_Toc115190413)

Create Custom AMIs

Document Name: Create Custom AMIs

Owner: Hema Paul

# Scope

A custom AMI can improve provisioning times when instances are launched in your environment if you need to install a lot of software that isn't included in the standard AMIs.

# Problem Statement

An Amazon Machine Image (AMI) is a supported and maintained image provided by AWS that provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you require multiple instances with the same configuration.

# Introduction

## Overview

An *Amazon Machine Image (AMI)* is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI, as shown in the following figure.


     Launch multiple instances from an AMI
    

## AMI Lifecycle

The following diagram summarizes the AMI lifecycle. After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.) You can copy an AMI within the same AWS Region or to different AWS Regions. When you no longer require an AMI, you can deregister it.


    The AMI lifecycle (create, register, launch, copy, deregister).
   

***Note:*** *Your instances keep running until you stop, hibernate, or terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI*.

## How the creation of a custom AMI works

First, launch an instance from an AMI that's similar to the AMI that you'd like to create. You can connect to your instance and customize it. When the instance is set up the way you want it, ensure data integrity by stopping the instance before you create an AMI and then create the image. We automatically register the AMI for you.

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance. You're charged for the snapshots until you Deregister the AMI and Delete the snapshots. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instance types that support Amazon EBS encryption.

Depending on the size of the volumes, it can take several minutes for the AMI-creation process to complete (sometimes up to 24 hours). You may find it more efficient to create snapshots of your volumes prior to creating your AMI. This way, only small, incremental snapshots need to be created when the AMI is created, and the process completes more quickly (the total time for snapshot creation remains the same)

If you add instance store volumes or Amazon Elastic Block Store (Amazon EBS) volumes to your instance in addition to the root device volume, the block device mapping for the new AMI contains information for these volumes, and the block device mappings for instances that you launch from the new AMI automatically contain information for these volumes. The instance store volumes specified in the block device mapping for the new instance are new and don't contain any data from the instance store volumes of the instance you used to create the AMI. The data on EBS volumes persists

## Create a Windows AMI from a running instance

You can create an AMI using the AWS Management Console or the command line. The following diagram summarizes the process for creating an AMI from a running EC2 instance. Start with an existing AMI, launch an instance, customize it, create a new AMI from it, and finally launch an instance of your new AMI. The steps in the following diagram match the steps in the procedure below.

**Note:** If you already have a running Windows instance, you can go directly to step 5.

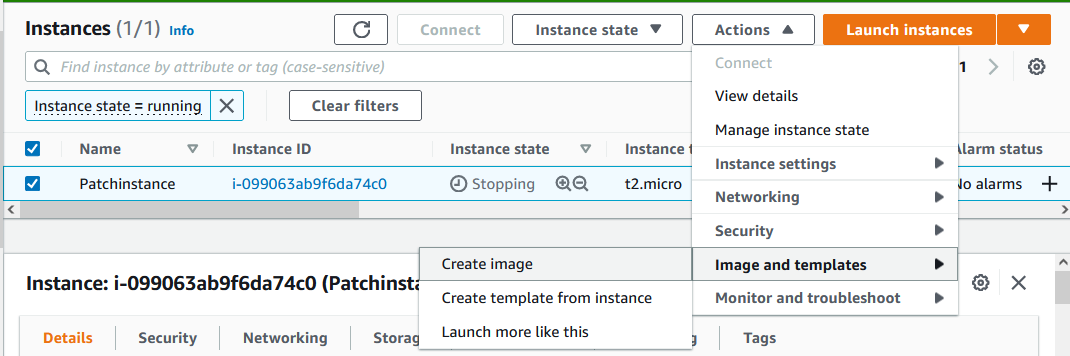

     Workflow for creating an AMI from an instance
    

# Implementation

Use the following procedure to **create your own AMI** using the AWS Management Console:

## Creating an AMI from an instance:

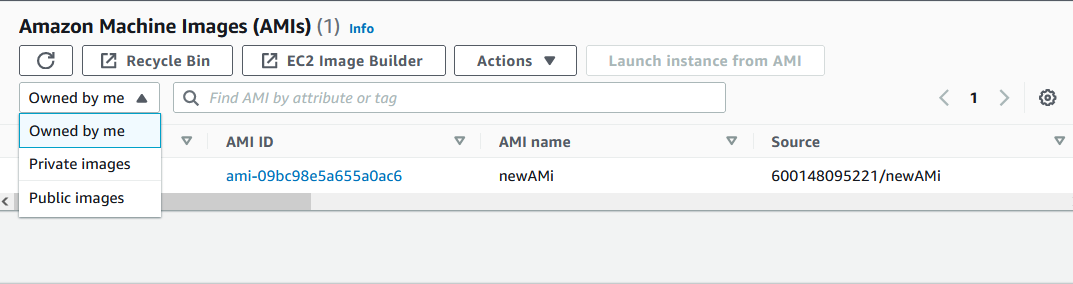
1. Open the Amazon EC2 console.
2. Choose the instances tab from the left pane and select an instance; from there click in Action button and select Image and templates -> Create image



3. In the navigation pane, under **Images**, choose **AMIs**.

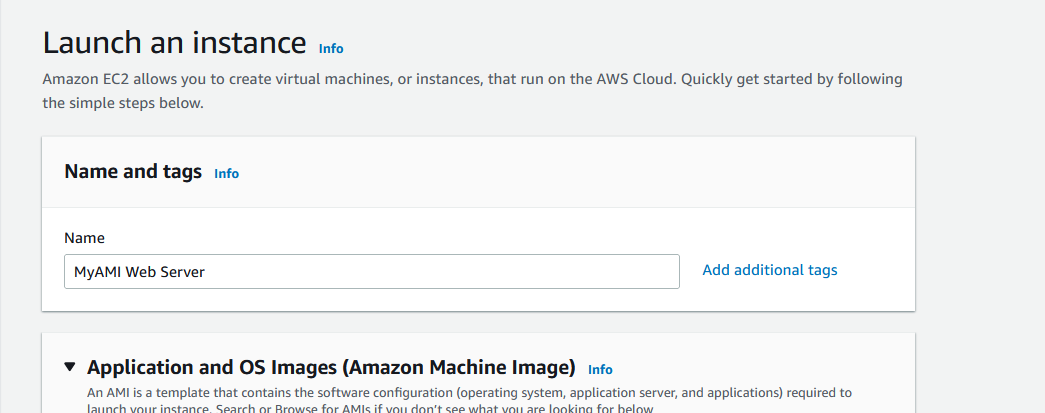
4. Use the **Filter** options to scope the list of AMIs to the Windows AMIs that meet your needs. For example, to view the Windows AMIs provided by AWS, choose **Public images** from the drop-down list. Choose the Search bar. Choose **Owner** from the menu and choose **amazon**. Choose **Source** from the menu and enter one of the following, depending on the version of Windows Server that you need.

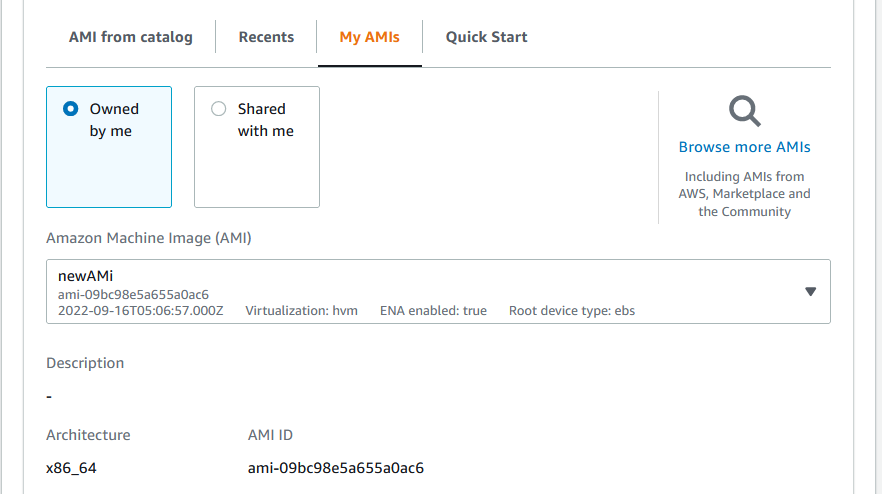
5. Specify a unique name for the image and an optional description

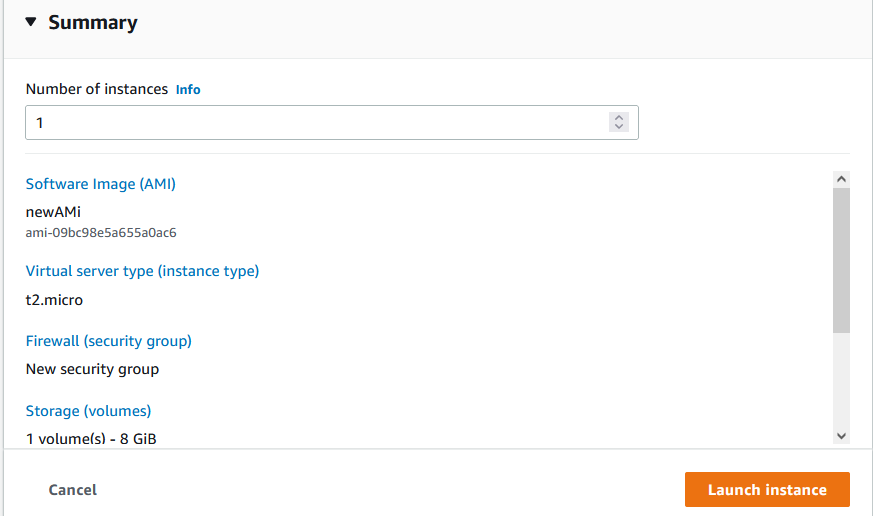


Add any other filters that you need. When you have chosen an AMI, select its check box.

6. Choose **Launch instance from AMI**. Accept the default values as you step through the wizard.







7. Once you connect to the instance, you can perform any of the following actions to customize it for your needs:

* Install software and applications
* Copy data
* Reduce start time by deleting temporary files, defragmenting your hard drive, and zeroing out free space
* Attach additional EBS volumes
* Create a new user account and add it to the Administrators group

By default, when Amazon EC2 creates the new AMI, it reboots the instance so that it can take snapshots of the attached volumes while data is at rest, in order to ensure a consistent state. For the **No reboot** setting, you can select the **Enable** check box to prevent Amazon EC2 from shutting down and rebooting the instance.

1. Warning:

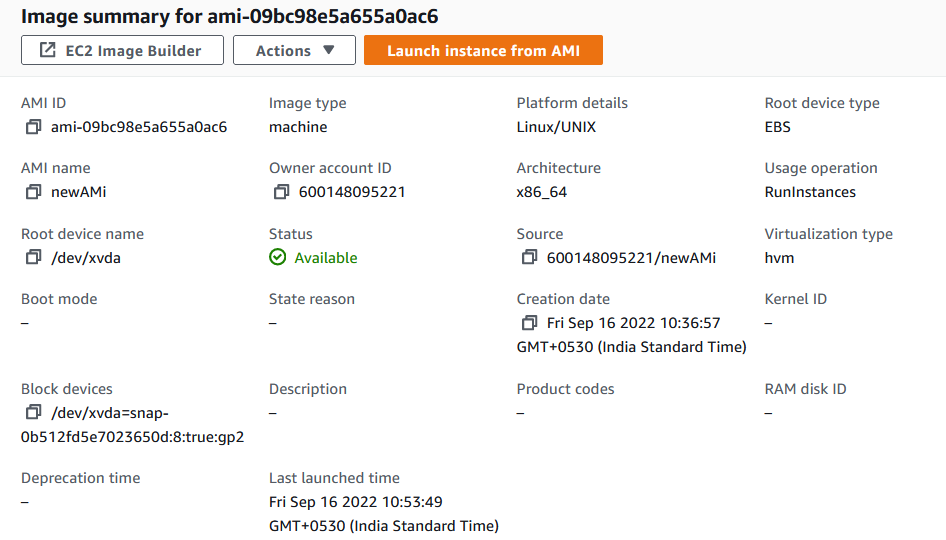
If you choose to enable **No reboot**, we can't guarantee the file system integrity of the created image.

(Optional) Modify the root volume, EBS volumes, and instance store volumes as needed. For example:

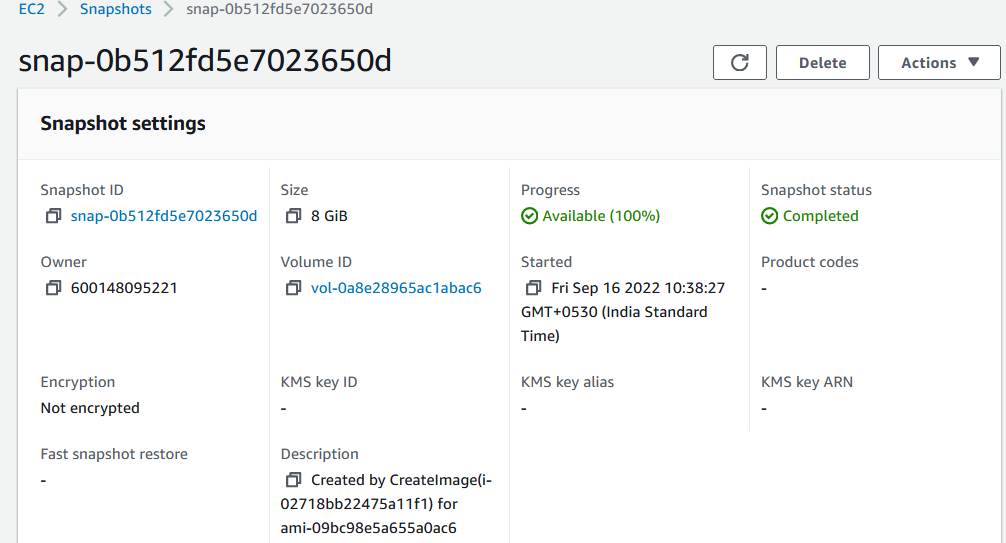
* + To change the size of the root volume, locate the **Root** volume in the **Type** column, and fill in the **Size** field.
  + To suppress an EBS volume specified by the block device mapping of the AMI used to launch the instance, locate the EBS volume in the list and choose **Delete**.
  + To add an EBS volume, choose **Add New Volume**, **Type**, and **EBS**, and fill in the fields. When you then launch an instance from your new AMI, these additional volumes are automatically attached to the instance. Empty volumes must be formatted and mounted. Volumes based on a snapshot must be mounted.
  + To suppress an instance store volume specified by the block device mapping of the AMI used to launch the instance, locate the volume in the list and choose **Delete**.
  + To add an instance store volume, choose **Add New Volume**, **Type**, and **Instance Store**, and select a device name from the **Device** list. When you launch an instance from your new AMI, these additional volumes are automatically initialized and mounted. These volumes don't contain data from the instance store volumes of the running instance from which you based your AMI.

When you are finished, choose **Create Image**.

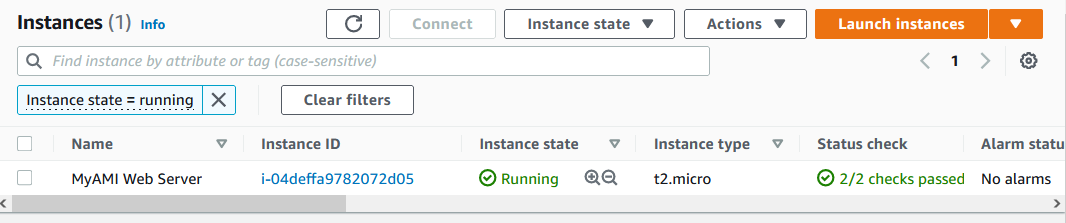
1. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Clear your previous filters, and choose **Owned by me** from the drop-down list. Initially, the status is pending. After a few minutes, the status should change to available.



(Optional) Choose **Snapshots** in the navigation pane to view the snapshot that was created for the new AMI. When you launch an instance from this AMI, we use this snapshot to create its root device volume.



1. Launch an instance from your new AMI. The new running instance contains all of the customizations you applied in previous steps, and any additional customization you add when launching the instance, such as user data (scripts that run when the instance starts).



# Creating Custom AMIs from EC2 Image Builder

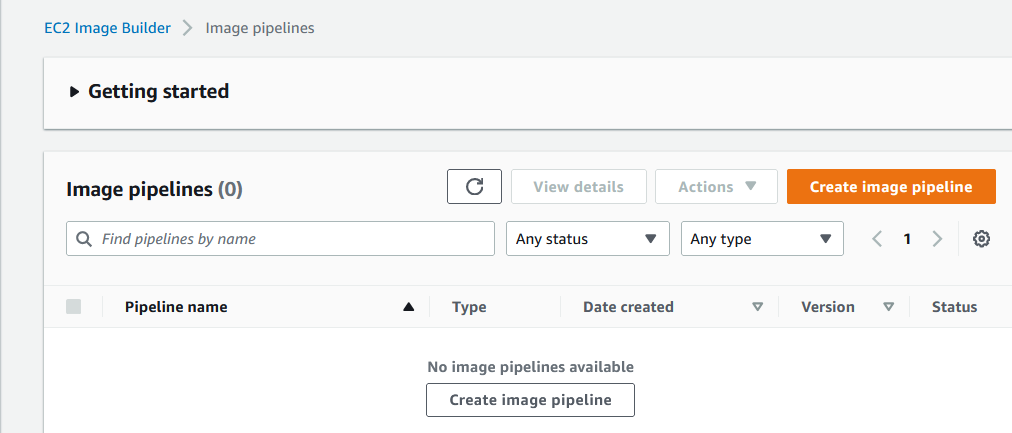
## EC2 Image Builder

https://docs.aws.amazon.com/imagebuilder/latest/userguide/start-build-image-pipeline.html

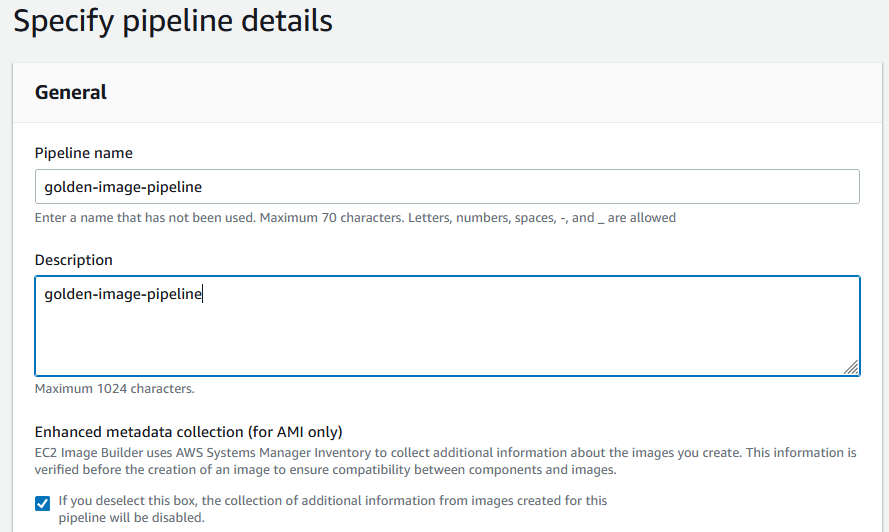
EC2 Image Builder is a fully managed AWS service that makes it easier to automate the creation, management, and deployment of customized, secure, and up-to-date server images that are pre-installed and pre-configured with software and settings to meet specific IT standards.

### Step1: create image pipeline

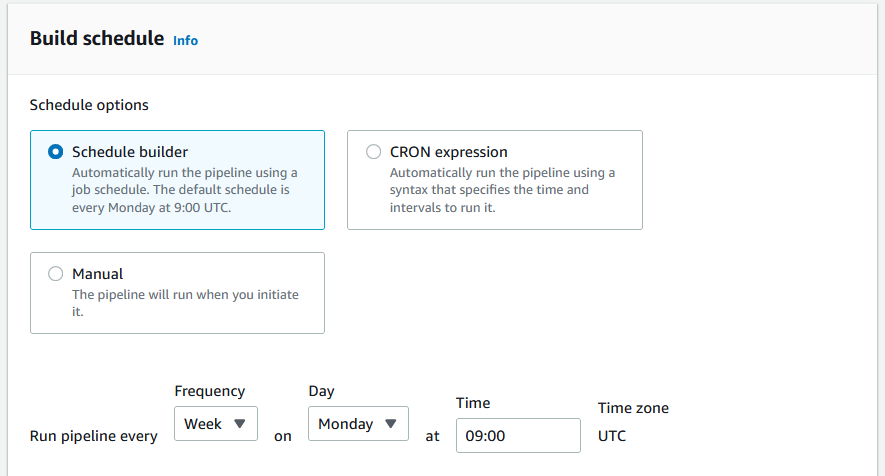
1. Open the EC2 Image Builder console.



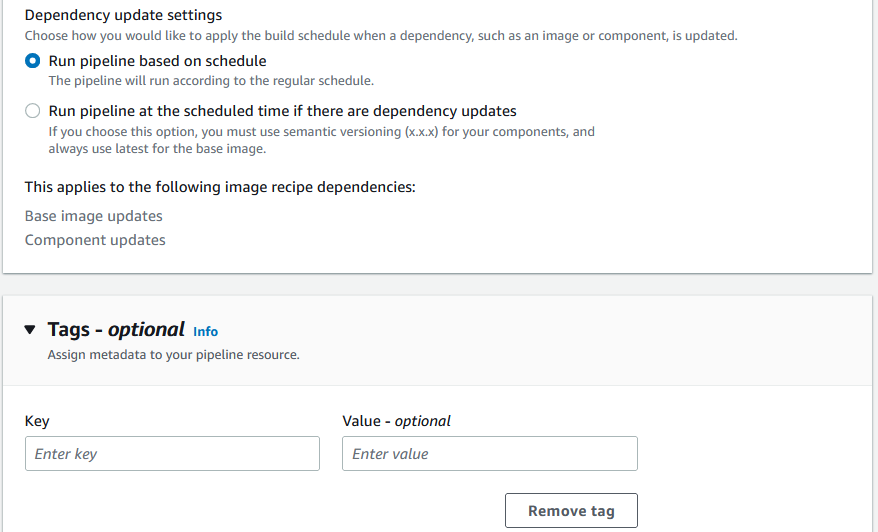
1. To begin creating your pipeline, choose **Create image pipeline**.
2. In the **General** section, enter your **Pipeline name** (required).



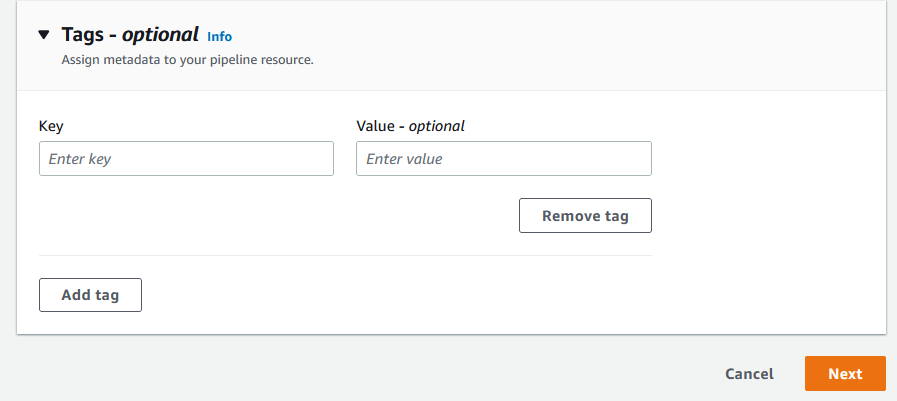
In the **Build schedule** section, you can keep the defaults for the **Schedule options**. Note that the **Time zone** shown for the default schedule is Universal Coordinated Time (UTC)



For **Dependency update settings**, choose the **Run pipeline at the scheduled time if there are dependency updates** option. This setting causes your pipeline to check for updates before starting the build. If there are no updates, it skips the scheduled pipeline build.

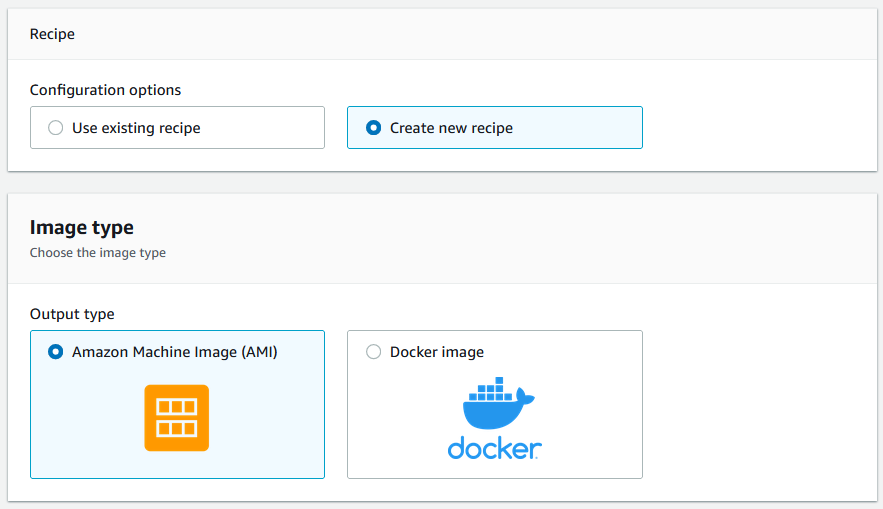


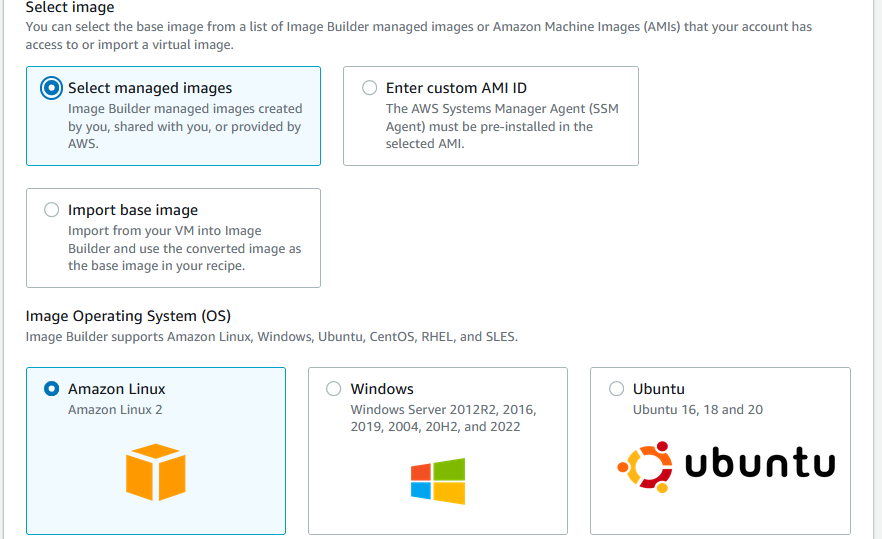
1. Choose **Next** to proceed to the next step.



### Step2: Recipe

1. Image Builder defaults to **Use existing recipe** in the **Recipe** section. For your first time through, choose the **Create new recipe** option.
2. In the **Image type** section, choose the **Amazon Machine Image (AMI)** option to create an image pipeline that will produce and distribute an AMI.





1. In the **General** section, enter the following required boxes:

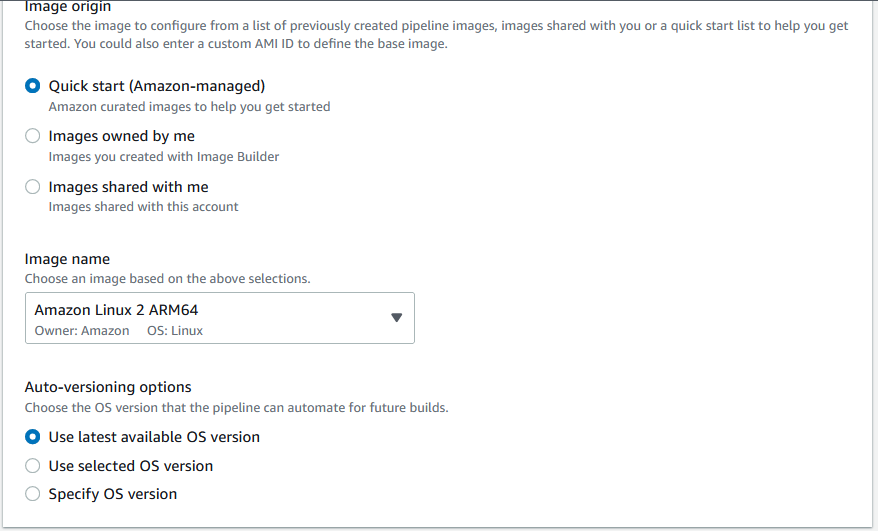
**Name** – your recipe name

**Version** – your recipe version (use the format <major>.<minor>.<patch>, where major, minor, and patch are integer values). New recipes generally start with 1.0.0.

1. In the **Source image** section, keep the default values for **Select image**, **Image Operating System (OS)**, and **Image origin**. This results in a list of Amazon Linux 2 AMIs, managed by Amazon, for you to choose from for your base image.

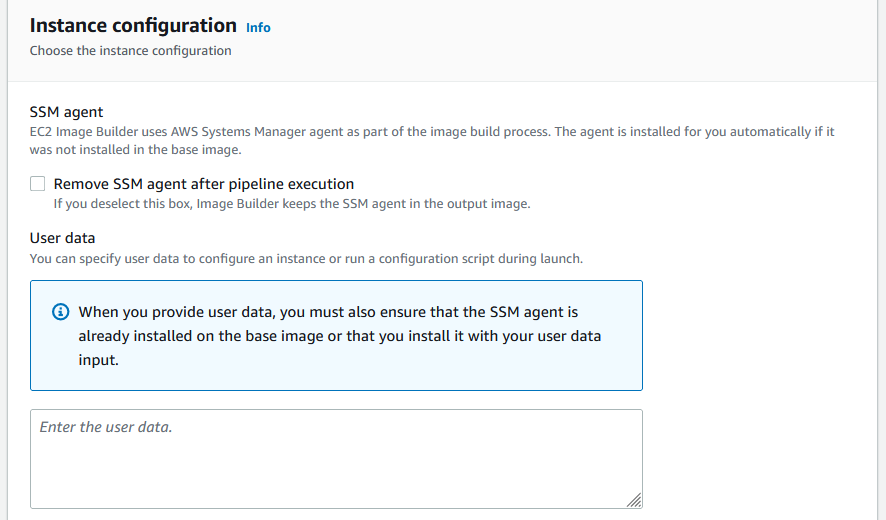
From the **Image name** dropdown, choose an image.

Keep the default for **Auto-versioning options** (**Use latest available OS version**).



1. In the **Instance configuration** section, keep the default values for the **Systems Manager agent**. This results in Image Builder keeping the Systems Manager agent after the build and tests are complete, to include the Systems Manager agent in your new image.

Keep **User data** blank. You can use this area at other times to provide commands, or a command script to run when you launch your build instance. However, it replaces any commands that Image Builder might have added to ensure that Systems Manager is installed. When you do use it, make sure that the Systems Manager agent is preinstalled on your base image, or that you include the install in your user data.



1. In the **Components** section, you must choose at least one build component.

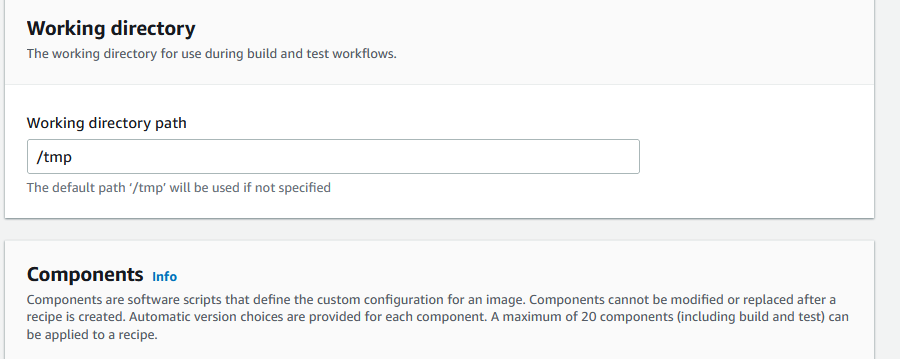
In the **Build components – Amazon Linux** panel, you can browse through the components listed on the page. Use the pagination control in the upper right corner to navigate through additional components that are available for your base image OS. You can also search for specific components, or create your own build component using the Component manager.

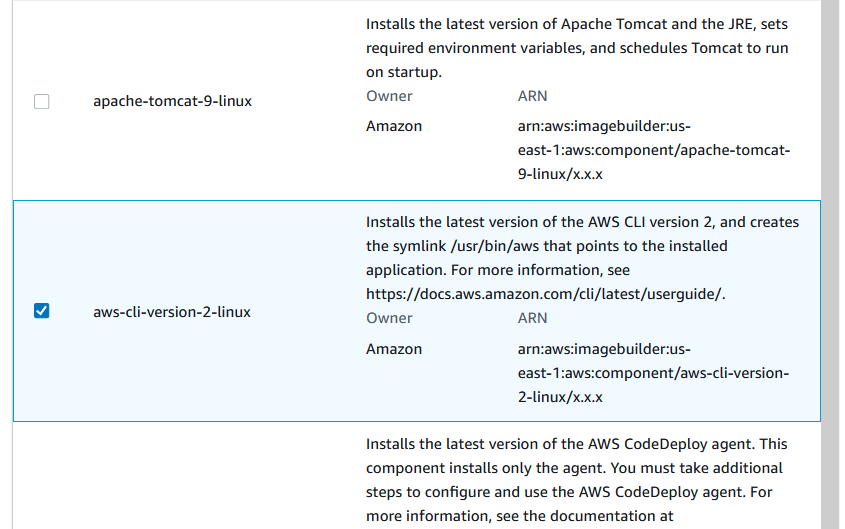
For this tutorial, choose a component that updates Linux with the latest security updates, as follows:

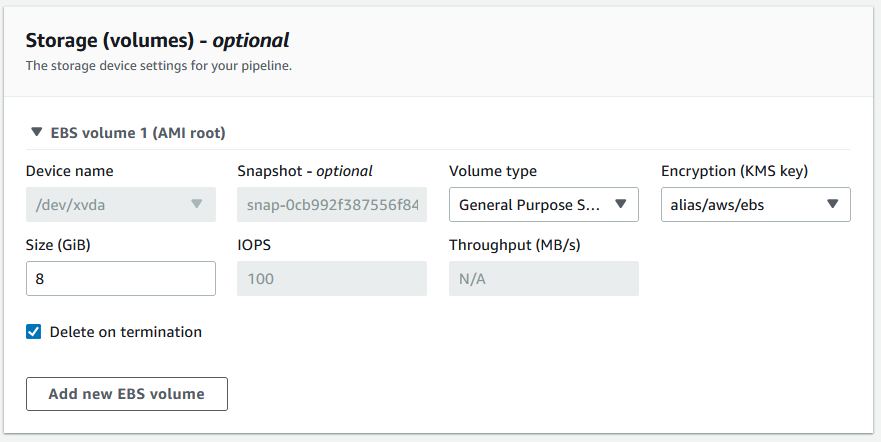
1. Filter the results by entering the word update in the search bar that's located at the top of the panel.
2. Select the check box for the update-linux build component.
3. Scroll down, and in the upper right corner of the **Selected components** list, choose **Expand all** .
4. Keep the default for **Versioning options** (**Use latest available component version**)

**Reorder components (optional)**

If you have chosen more than one component to include in your image, you can use the drag-and-drop action to rearrange them into the order in which they should run during the build process.





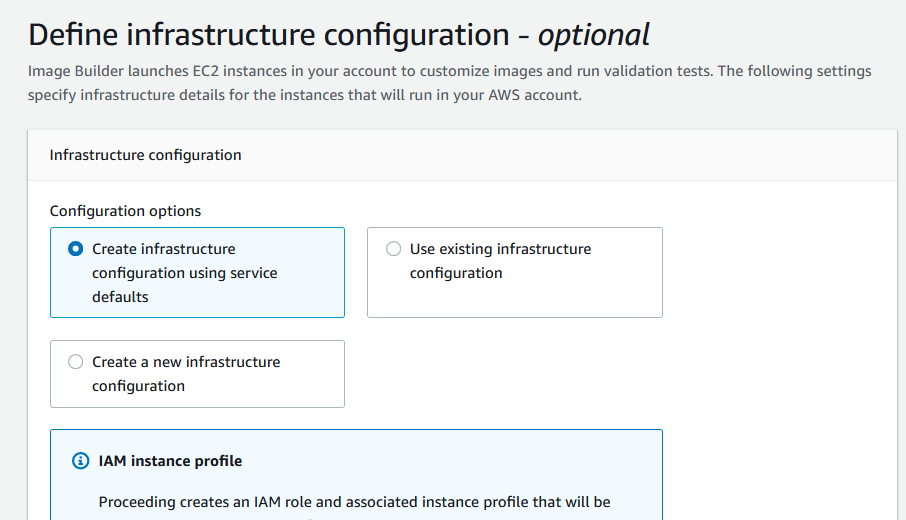


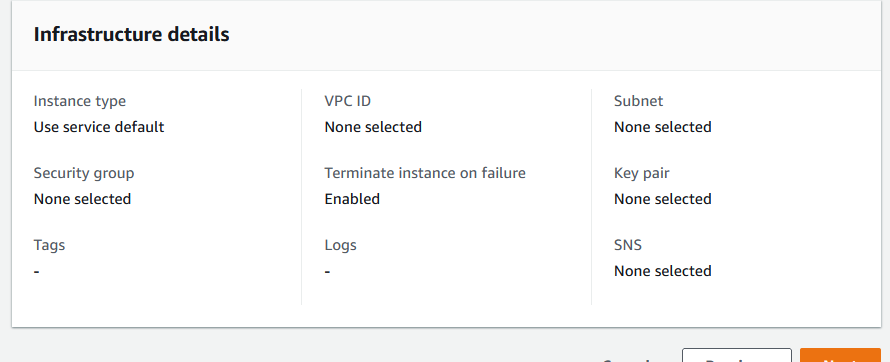
1. Choose **Next** to proceed to the next step.

### Step 3: Define Infrastructure configuration (optional)

Image Builder launches EC2 instances in your account to customize images and run validation tests. The Infrastructure configuration settings specify infrastructure details for the instances that will run in your AWS account during the build process.

In the **Infrastructure configuration** section, the **Configuration options** default to Create infrastructure configuration using service defaults. This creates an IAM role and associated instance profile for the EC2 build and test instances that are used to configure your image.



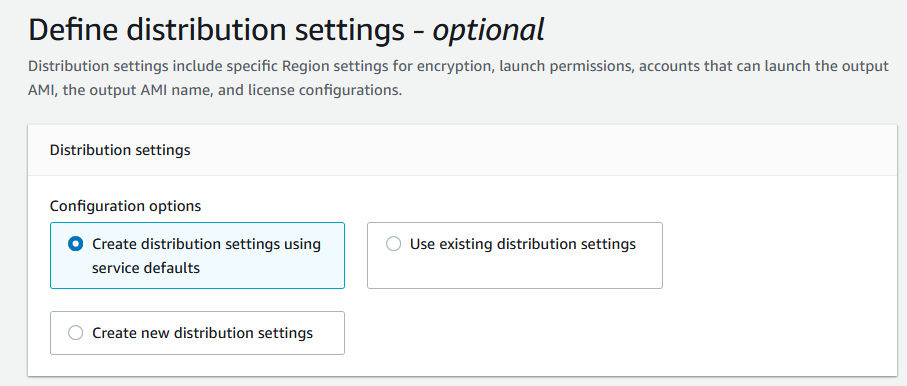


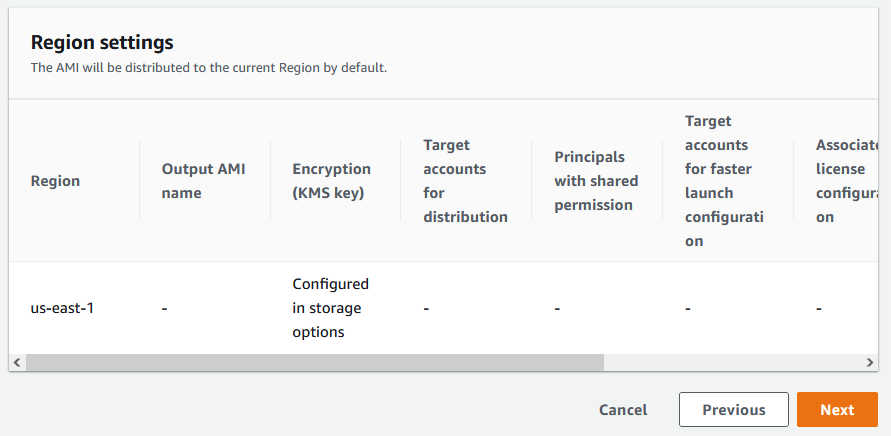
* Choose **Next** to proceed to the next step.

### Step 4: Define distribution settings (optional)

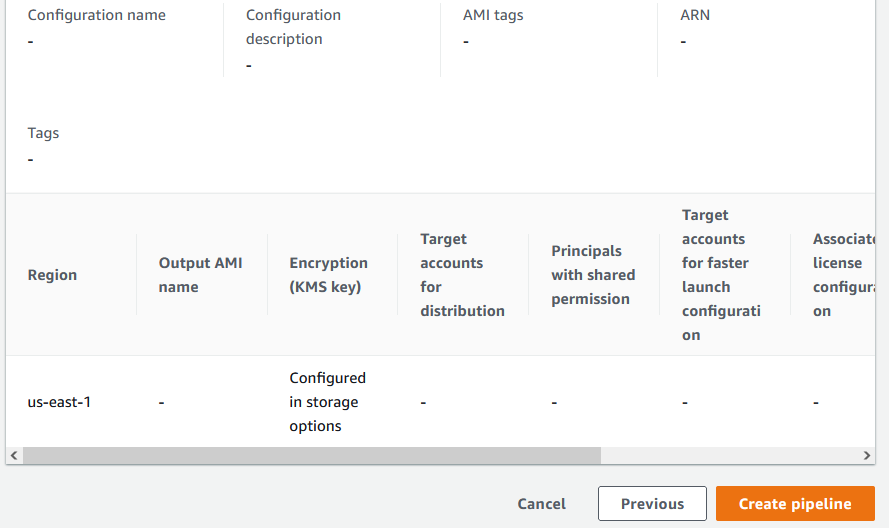
Distribution configurations include the output AMI name, specific Region settings for encryption, launch permissions, and AWS accounts, organizations, and organizational units (OUs) that can launch the output AMI, and license configurations.

In the **Distribution settings** section, the **Configuration options** default to Create distribution settings using service defaults. This option will distribute the output AMI to the current Region..





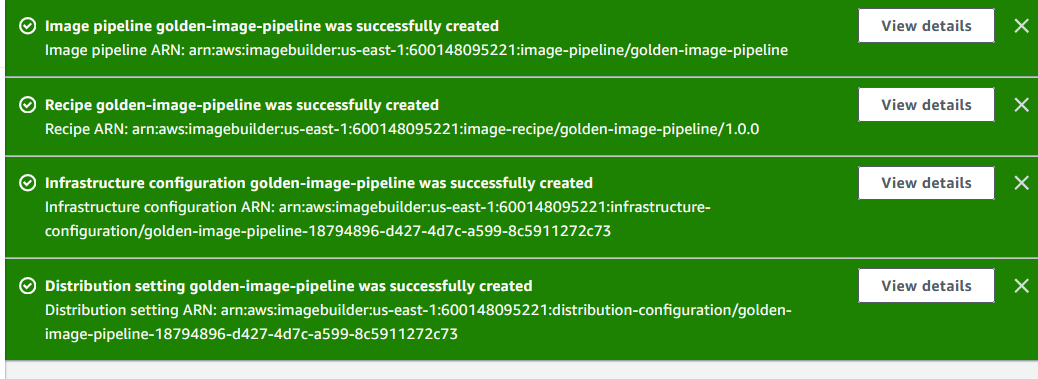
* Choose **Next** to proceed to the next step.

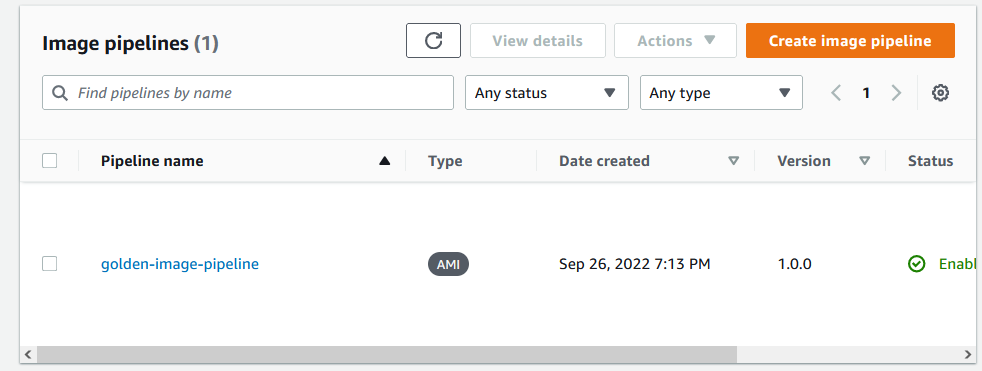


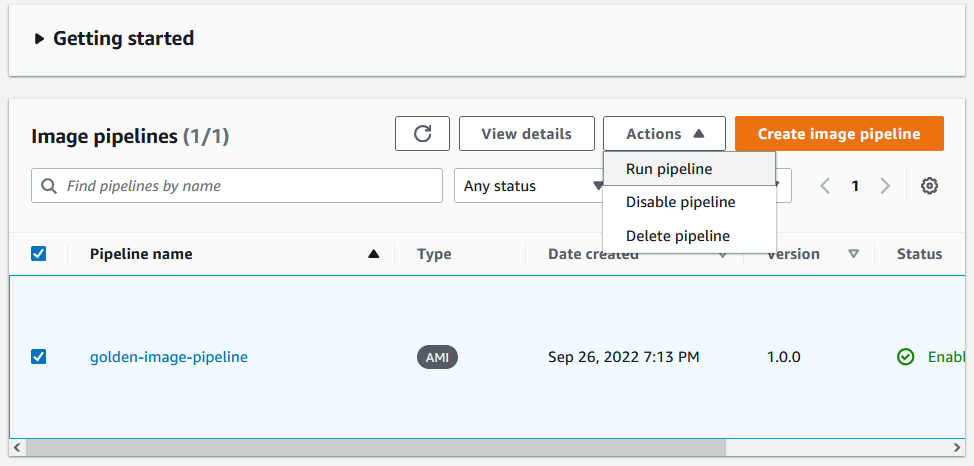
### Step 5: Review

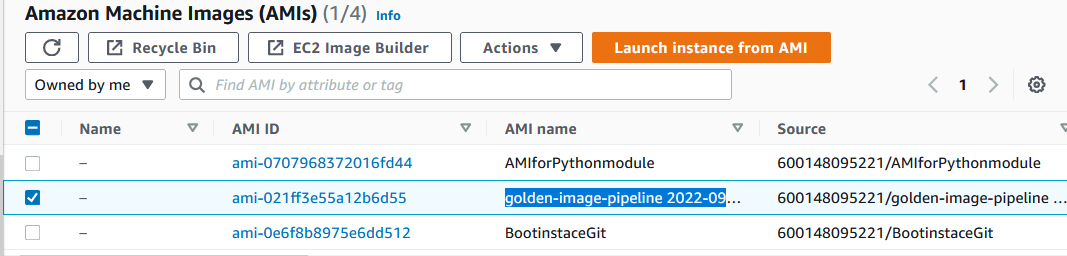
The **Review** section displays all of the settings you have configured. To edit information in any given section, choose the **Edit** button located in the top right corner of the step section. For example, if you want to change your pipeline name, choose the **Edit** button in the top right corner of the **Step 1: Pipeline details** section.

1. When you have reviewed your settings, choose **Create pipeline** to create your pipeline.
2. You can see success or failure messages at the top of the page, as your resources are created for distribution settings, infrastructure configuration, your new recipe, and the pipeline. To see details for a resource, including the resource identifier, choose **View details**.
3. After you have viewed the details for a resource, you can view details about other resources by choosing the resource type from the navigation pane. For example, to see details for your new pipeline, choose **Image pipelines** from the navigation pane. If your build was successful, your new pipeline is displayed in the **Image pipelines** list.

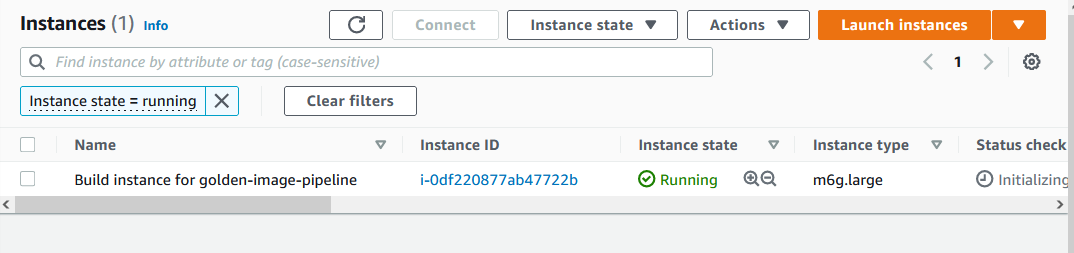








When the AMI is created, an instance will also get created in the background which is visible in EC2 instances



**Connect the instance**

The details can be viewed .

## Step 6: Clean up

Your Image Builder environment, just like your home, needs regular maintenance to help you find what you need, and complete your tasks without wading through clutter. Make sure to regularly clean up temporary resources that you created for testing. Otherwise, you might forget about those resources, and then later, not remember what they were used for. By then, it might not be clear if you can safely get rid of them.

**Tip:**

To prevent dependency errors when you delete resources, make sure to delete your resources in the following order:

1. Image pipeline
2. Image recipe
3. All remaining resources

To clean up the resources that you created for this tutorial, follow these steps:

**Delete the pipeline**

1. To see a list of the build pipelines created under your account, choose **Image pipelines** from the navigation pane.
2. Select the check box next to **Pipeline name** to select the pipeline that you want to delete.
3. At the top of the **Image pipelines** panel, on the **Actions** menu, choose **Delete**.
4. To confirm the deletion, enter Delete in the box, and choose **Delete**.

**Delete the recipe**

1. To see a list of the recipes created under your account, choose **Image recipes** from the navigation pane.
2. Select the check box next to **Recipe name** to select the recipe that you want to delete.
3. At the top of the **Image recipes** panel, on the **Actions** menu, choose **Delete recipe**.
4. To confirm the deletion, enter Delete in the box, and choose **Delete**.

**Delete infrastructure configuration**

1. To see a list of the infrastructure configurations created under your account, choose **Infrastructure configuration** from the navigation pane.
2. Select the check box next to **Configuration name** to select the infrastructure configuration that you want to delete.
3. At the top of the **Infrastructure configurations** panel, choose **Delete**.
4. To confirm the deletion, enter Delete in the box, and choose **Delete**.

**Delete distribution settings**

1. To see a list of the distribution settings created under your account, choose **Distribution settings** from the navigation pane.
2. Select the check box next to **Configuration name** to select the distribution settings that you created for this tutorial.
3. At the top of the **Distribution settings** panel, choose **Delete**.
4. To confirm the deletion, enter Delete in the box, and choose **Delete**.

**Delete the image**

Follow these steps to verify that you have deleted any image that was created from the tutorial pipeline. This tutorial is not likely to create an image unless enough time has elapsed since you created your pipeline that it runs, according to the build schedule.

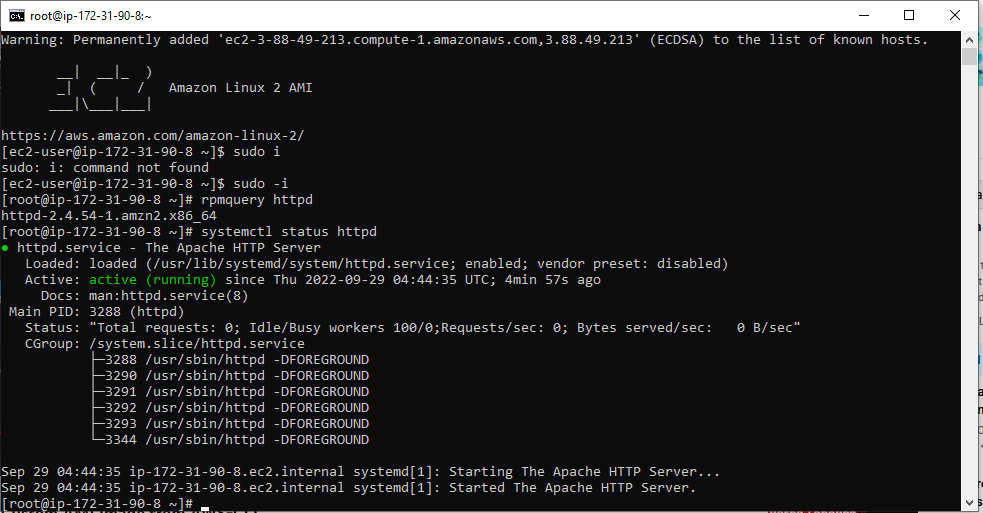
1. To see a list of the images created under your account, choose **Images** from the navigation pane.
2. Choose the image **Version** for the image that you want to remove. This opens the **Image build versions** page.
3. Select the check box next to the **Version** for any image that you want to delete. You can select more than one image version at a time.
4. At the top of the **Image build versions** panel, choose **Delete version**.
5. To confirm the deletion, enter Delete in the box, and choose **Delete**.

**Creating Custom AMI through AWS CLI**

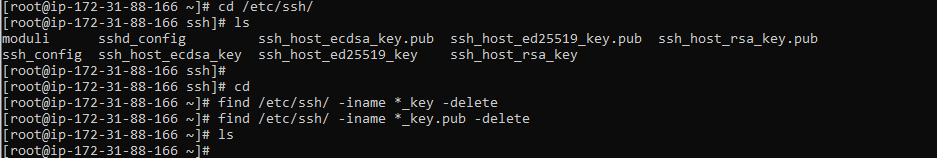
Launch instance from the console.

Insert script in the user data field for the needed module installations while launching the instance

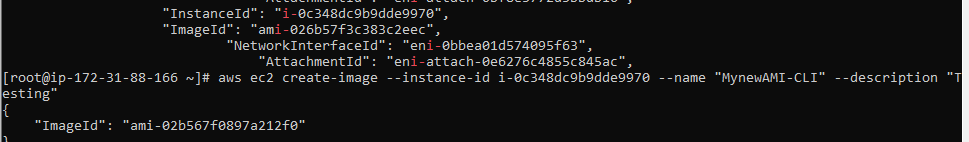
Connect the instance and check for the installations work



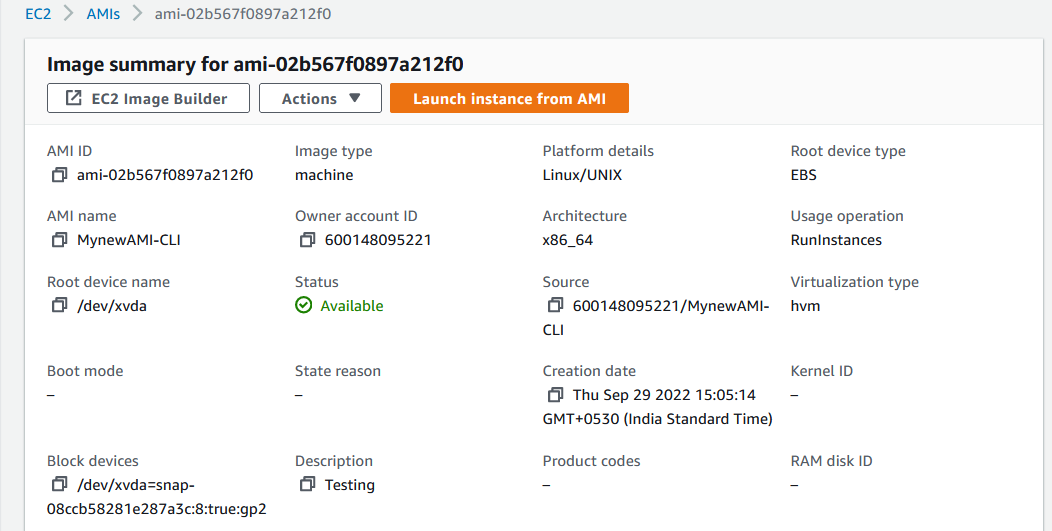
Now remove all the existing key files



Create image of the running instance



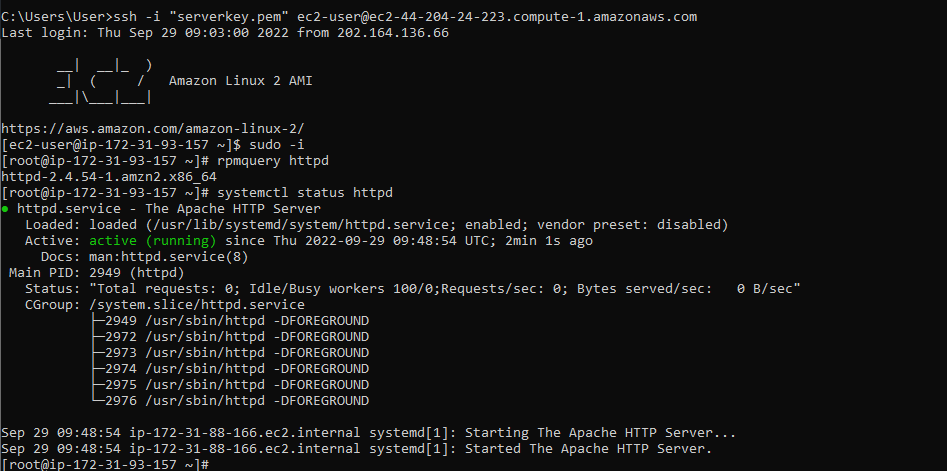
Check in console for the created image in the EC2 AMIs section



Launch new instance from the created AMI

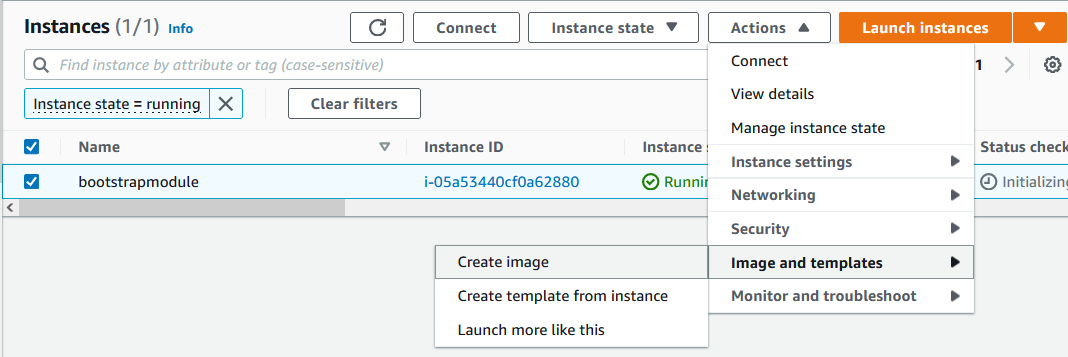
Connect the new instance

Check for the httpd module installed

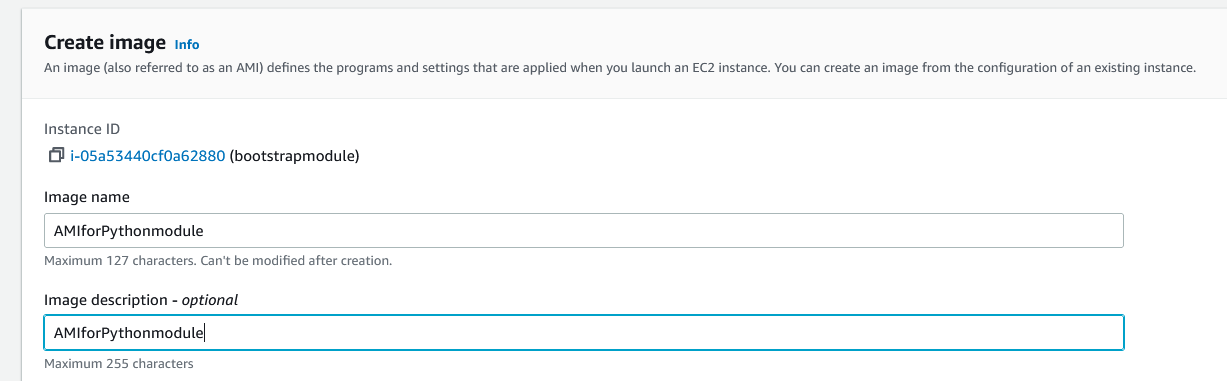


# Use Case

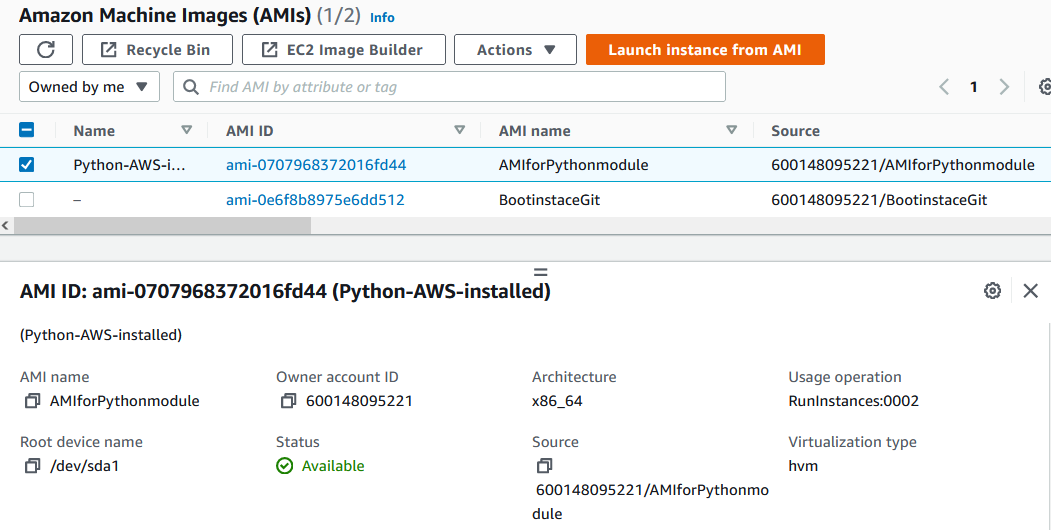
## Python installed Windows AMI



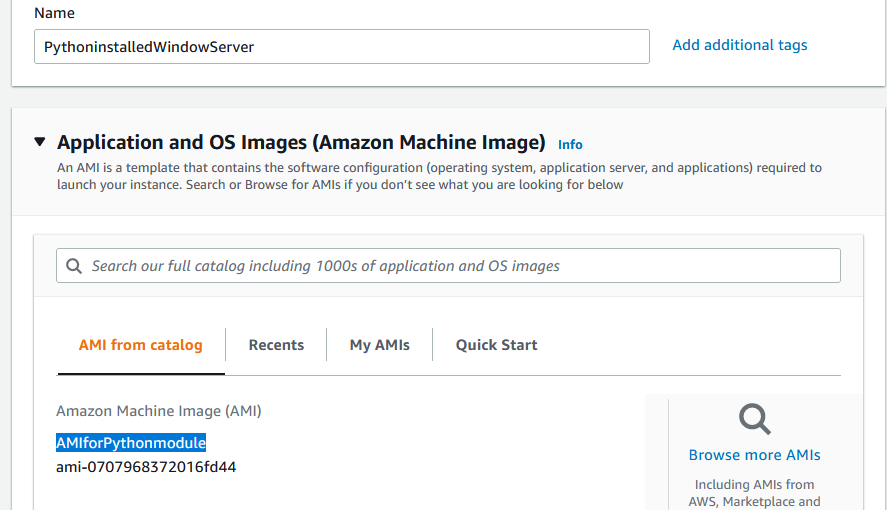
Creating image from a running instance with python module installed.



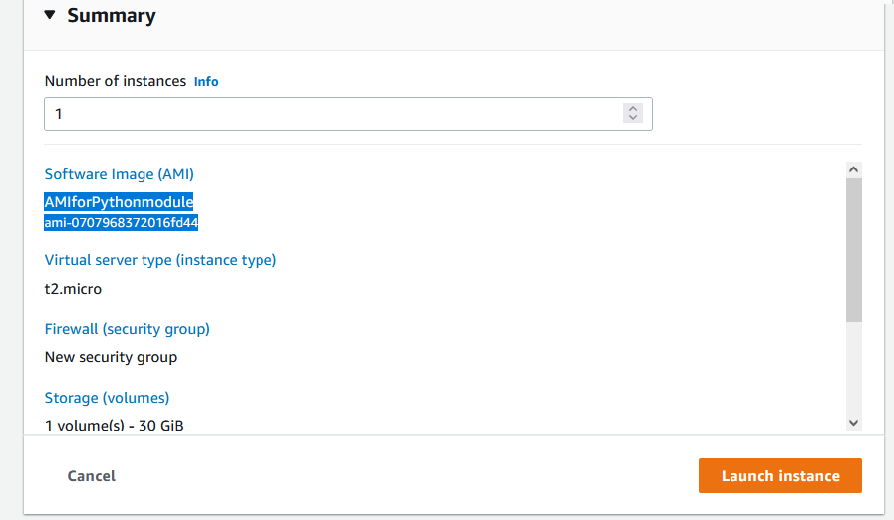
Launch instance from the created AMI when its status becomes **Available**



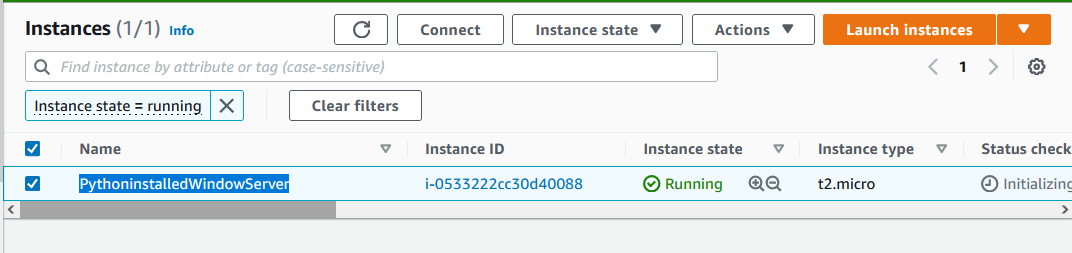
Choose created AMI



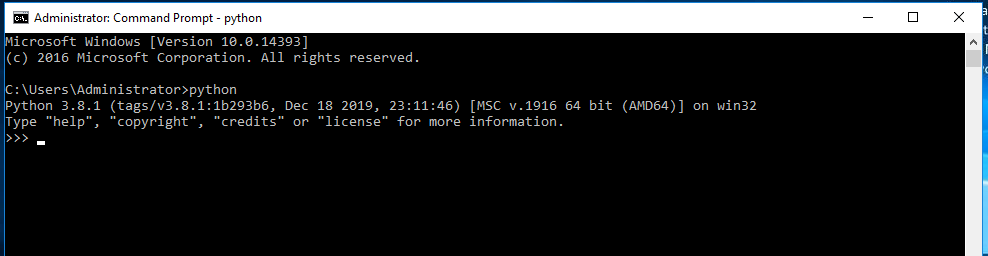
Launch instance



Successfully launched the new instance from custom AMI



Connect the instance and check whether the python module is installed in command prompt of the remote desktop



Check whether AWSCLI is installed in command prompt



# Configure OpenLDAP Server on Ubuntu

# Launch instance with ubuntu server.

# Install the needed modules (ldap, python, textfile…).

# Create AMI from the running instance.

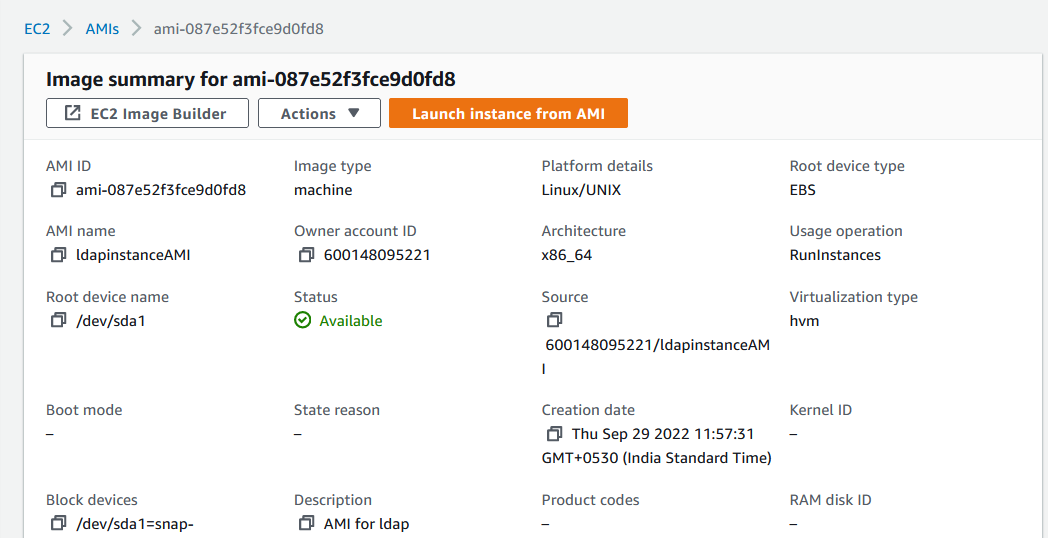
# Launch instance from created AMI.

# Stop the EC2 instance previously created.

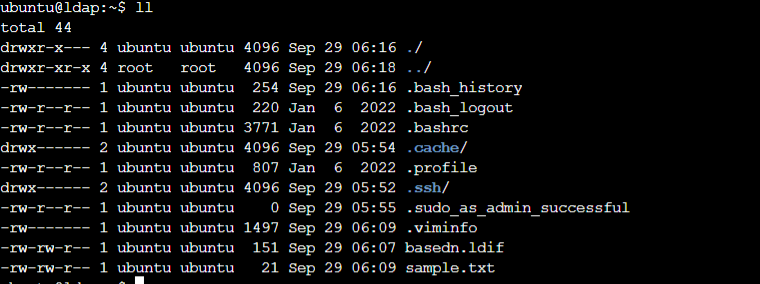
# Connect the new instance created from AMI.

# Check for the installed modules.

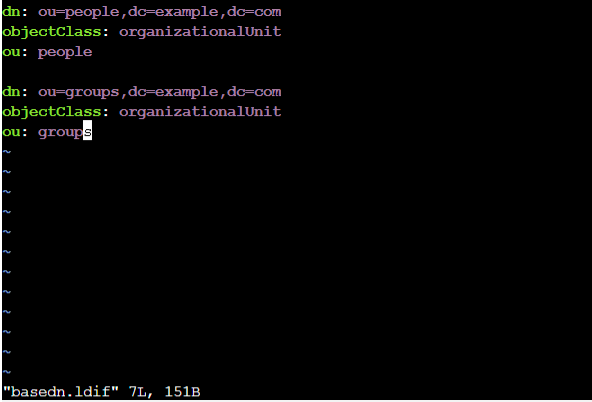
# Created AMI



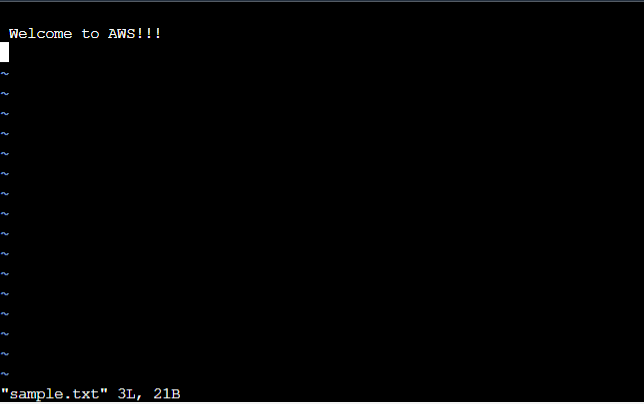
Connect and Check the installed modules



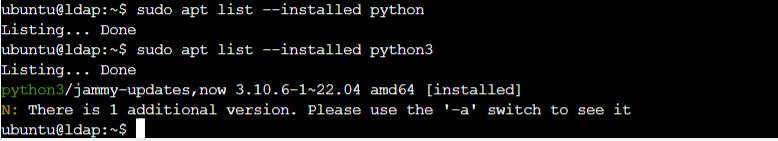
View the content of **basedn.ldif**



View the content of **sample.txt**



Check for the **python** module



# Advantages

You can launch multiple instances from a single AMI when you require multiple instances with the same configuration.

# Limitations

A custom AMI must exist in the same AWS Region where you create the cluster.

## 