AMI

- Introduction to Amazon Machine Image (AMI)
- Using AMI
- AMI Types
- Storage for the root device
- Creation and charges
- Finding Linux AMI
- Shared AMI
- Paid AMI
- Linux AMI Virtualization Types
- AMI Lab

Amazon Machine Images (AMI)

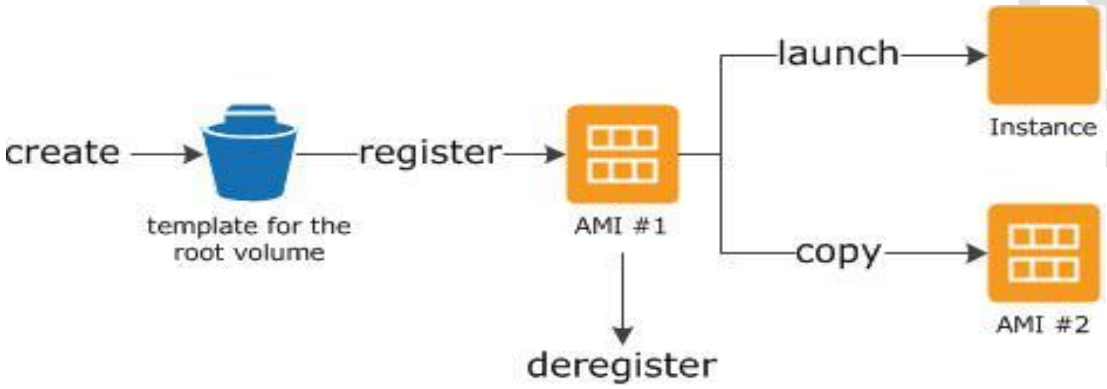
- An Amazon Machine Image (AMI) provides template kind of the information required to launch an instance, which is a virtual server in the cloud.
- We specify an AMI when we launch an instance, and we can launch as many instances as we need from the AMI.

An AMI includes the following:

- A **template** for the root volume for the instance (for example, an operating system, an application server, or applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mappings to attach the volumes to the instance.

Using an AMI

- After we create and register an AMI, we can use it to launch new instances.
- We can also launch instances from an AMI if the AMI owner grants launch permissions.
- We can copy an AMI to the same region or to different regions.
- We can search for an AMI that meets the criteria for your instance provided by AWS or community



Using AMI

Creating Our Own AMI

- We can customize the instance that we launch from a public AMI and then save that configuration as a **custom AMI** for our own use.
- Instances that we launch from your AMI use all the customizations that you've made.
- The root storage volume of an instance is either an **Amazon EBS volume** or an instance store volume determines the process we follow to create an AMI.

Buying, Sharing, and Selling AMIs

- After we create an AMI, we can keep it private so that only we can use it, or we can share it with a specified list of AWS accounts or we can also make our custom AMI public so that the community can use it.

AMI Types

We can select an AMI based on the following characteristics:

- Region
- Operating system
- Architecture (32-bit or 64-bit)
- Launch Permissions
- Storage for the Root Device

Launch Permissions

- The owner of an AMI determines launch permissions.

Launch permissions fall into the following categories

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

Storage for the Root Device

- All AMIs are categorized as either ***backed by Amazon EBS*** or ***backed by instance store***.
- AMI ***backed by EBS*** means that the root device for an instance launched from the AMI, which is created from an **Amazon EBS snapshot**.
- AMI ***backed by instance store*** means that the root device for an instance launched from the AMI, which is created from a **template stored in Amazon S3**
- By default, Amazon EBS-backed instance root volumes have the **DeleteOnTerminationflag** set to true

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates.* Data on any other Amazon EBS volumes persists after instance termination by default.	Data on any instance store volumes persists only during the life of the instance.
Upgrading	The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.	Instance attributes are fixed for the life of an instance.
Charges	You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot.	You're charged for instance usage and storing your AMI in Amazon S3.
Stopped state	Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS	Cannot be in stopped state; instances are running or terminated

To determine the root device type of an AMI using the console

- Open the Amazon EC2 console.
- In the navigation pane, click **AMIs**, and select the AMI.
- Check the value of **Root Device Type** in the **Details** tab as follows:
- If the value is EBS, this is an Amazon EBS-backed AMI.
- If the value is instance store, this is an instance store-backed AMI

Boot Times

- Amazon EBS-backed AMIs launch faster than Amazon EC2 instance store-backed AMIs

AMI Creation

- To create Linux AMIs backed by instance store, we must create an AMI from your instance on the instance itself.
- AMI creation is much easier for AMIs backed by Amazon EBS. The Create Image API action creates your Amazon EBS-backed AMI and registers it.
- There's also a button in the AWS Management Console that lets we create an AMI from a running instance

Finding a Linux AMI Using the Amazon EC2 Console

1. We can find Linux AMIs using the Amazon EC2 console. We can search through all available AMIs using the Images page, or select from commonly used AMIs on the Quick Launch tab when we use the console to launch an instance.
2. To find a Linux AMI using the Images page
3. Open the **Amazon EC2 console**.
4. From the navigation bar, select a region. we can select any region that's available to you, regardless of your location. This is the region in which you'll launch your instance.
5. In the navigation pane, click AMIs.
6. (Optional) Use the Filter options to scope the list of displayed AMIs to see only the AMIs that interest you. For example, to list all Linux AMIs provided by AWS, select Public images. Click the Search bar and select Owner from the menu, then select Amazon images. Click the Search bar again to select Platform and then the operating system from the list provided.
7. (Optional) Click the Show/Hide Columns icon to select which image attributes to display, such as the root device type. Alternatively, we can select an AMI from the list and view its properties in the Details tab.
8. Before we select an AMI, it's important that we check whether it's backed by instance store or by Amazon EBS and that we are aware of the effects of this difference.

To launch an instance from this AMI, select it and then click Launch.

Shared AMIs

- *A shared AMI* is an AMI that a developer created and made available for other developers to use.

- Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users.
- It is recommended that we get an AMI from a trusted source

Making an AMI Public

- Amazon EC2 enables us to share our AMIs with other AWS accounts.
- We are not billed when our AMI is launched by other AWS accounts; only the accounts launching the AMI are billed

Sharing a Public AMI Using the Console

- Open the Amazon EC2 console.
- In the navigation pane, click **AMIs**.
- Select your AMI in the list, and then select **Modify Image Permissions** from the **Actions** list.
- Select the **Public** radio button, and then click **Save**.

Note

- If an AMI has a product code, we can't make it public. we must share the AMI with only specific AWS accounts.

Paid AMIs

- A *paid AMI* is an AMI that we can purchase from a developer.
- The AWS Marketplace is an online store where we can buy software that runs on AWS.
- Launching an instance from a paid AMI is the same as launching an instance from any other AMI. No additional parameters are required.
- The instance is charged according to the rates set by the owner of the AMI, as well as the standard usage fees for the related web services.

Linux Amazon Machine Images use one of two types of virtualization:

HVM AMIs

- HVM AMIs are presented with a fully virtualized set of hardware and boot by executing the master boot record of the root block device. This virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification.

PV AMIs

- PV AMIs boot with a special boot loader called **PV-GRUB**, which starts the boot cycle and then chain loads the kernel specified in the menu.

- Paravirtual guests can run on host hardware that **does not have explicit support for virtualization**, but they **cannot** take advantage of special hardware extensions such as enhanced networking or GPU processing. Historically, PV guests had better performance than HVM guests in many cases.