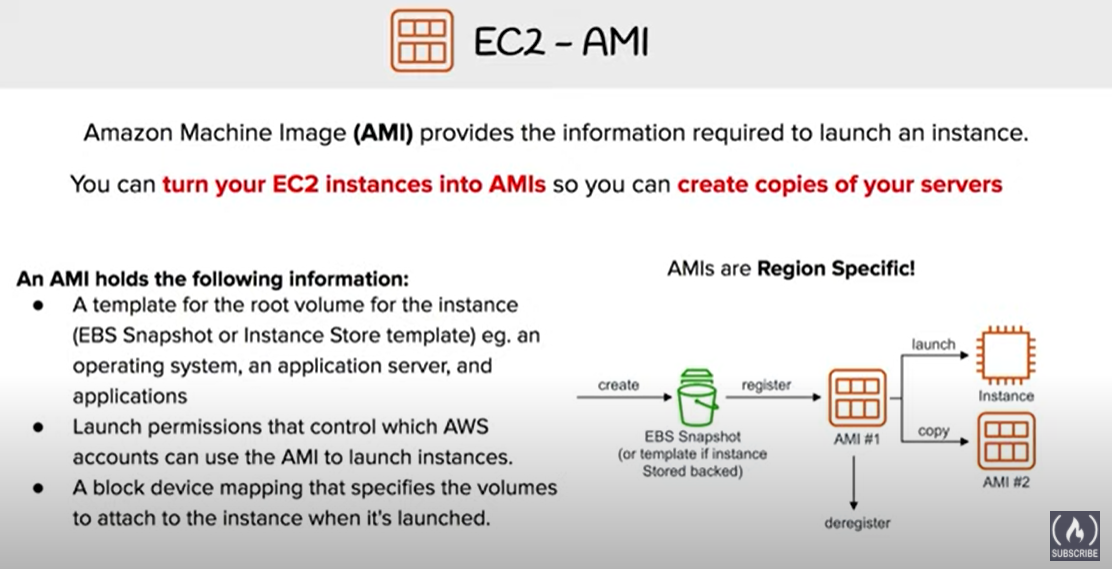
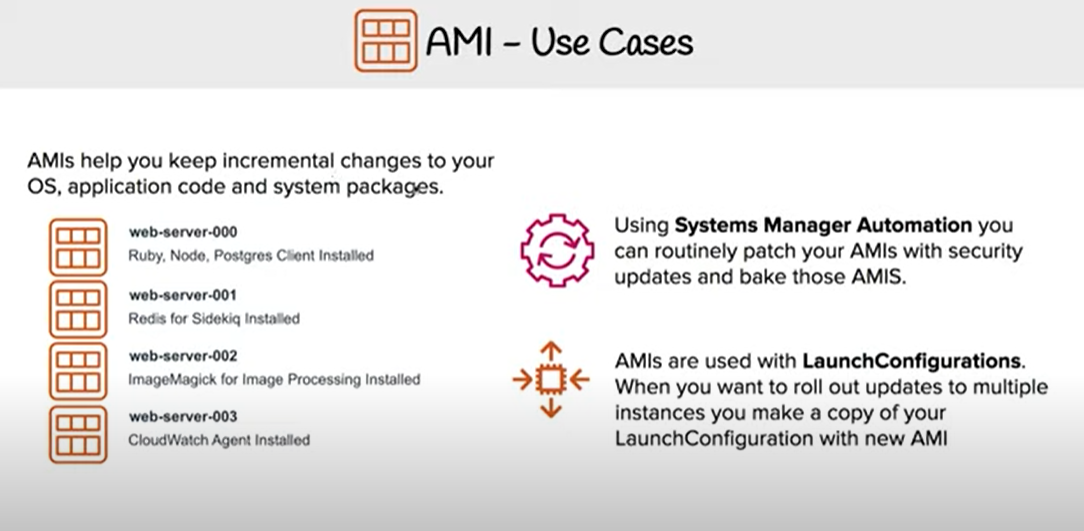
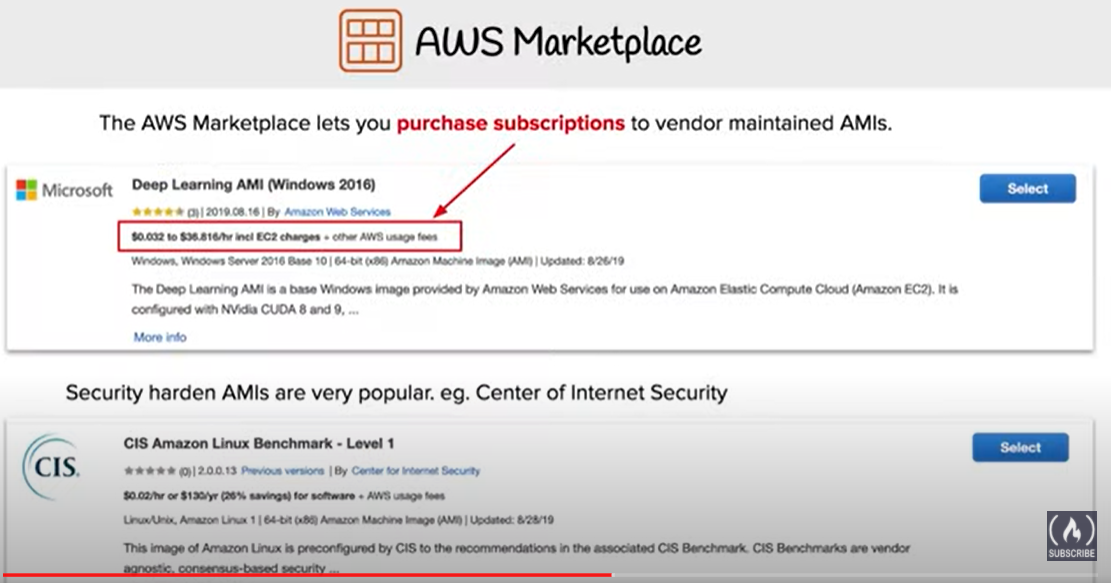
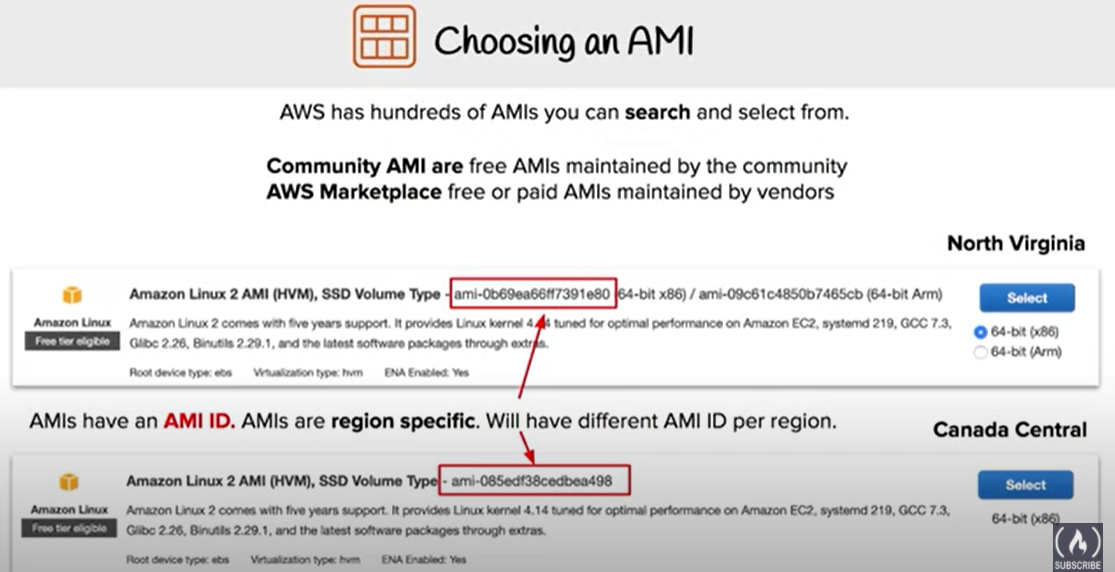
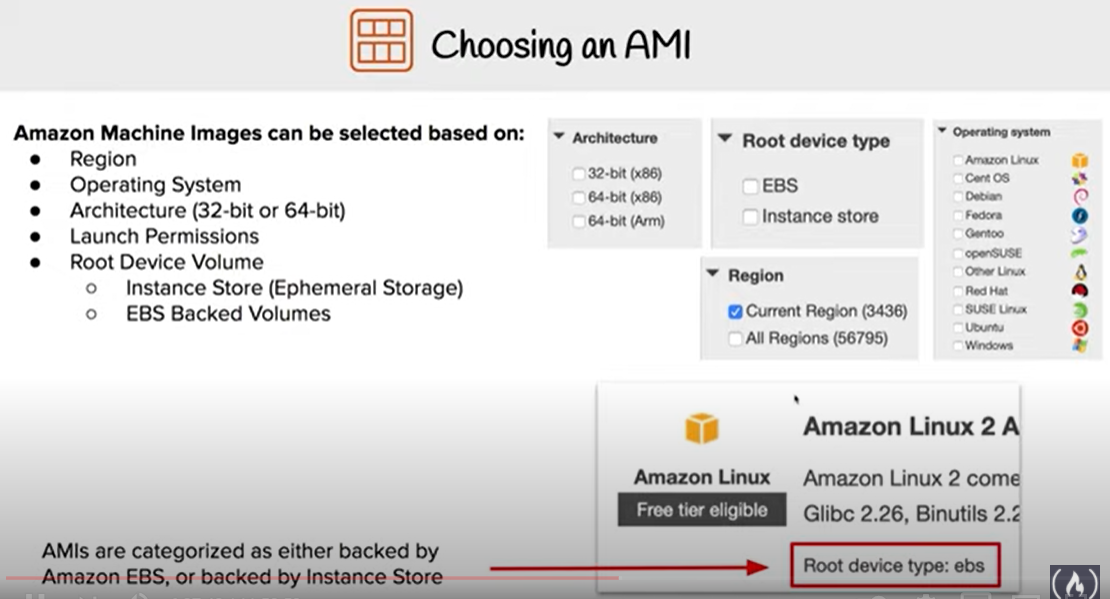
Amazon Machine Image – AMI

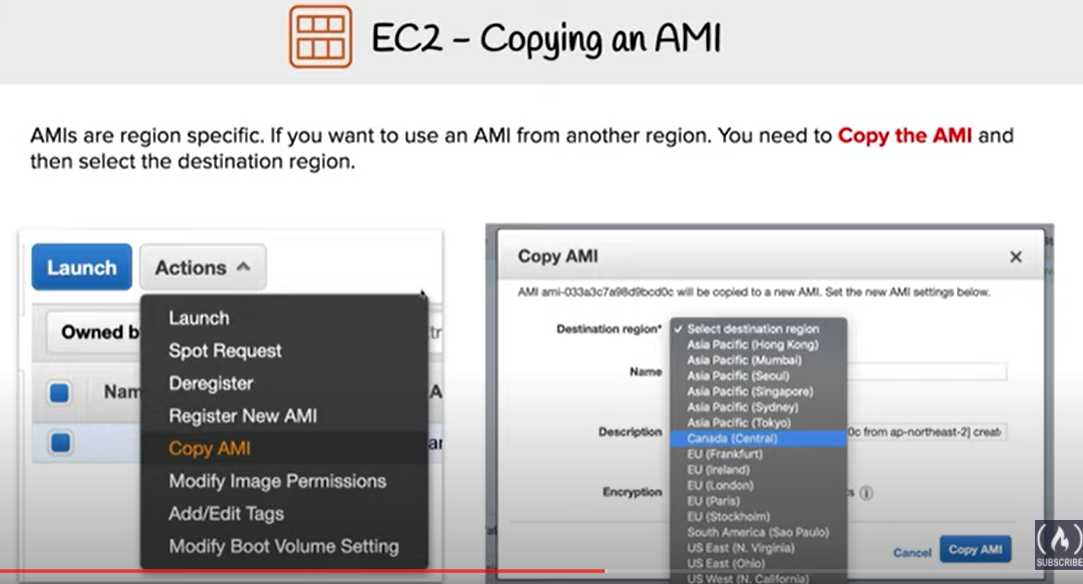












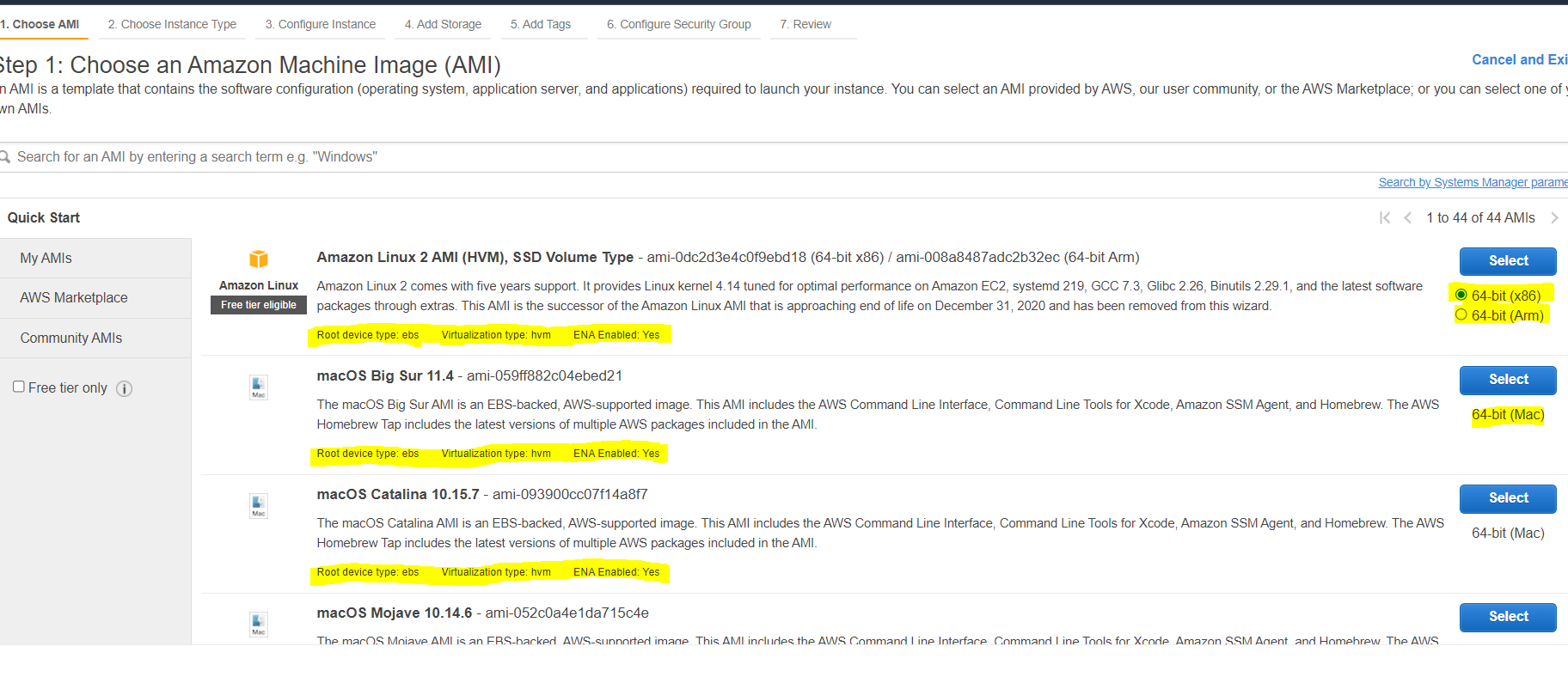
* An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud.
* An AMI is basically an template and can be used to launch as many instances as needed
* Within an VPC, instances can be launched from as many different AMIs
* An AMI includes the following:
  + One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance for e.g, an operating system, an application server, and applications
  + Launch permissions that control which AWS accounts can use the AMI to launch instances for e.g. AWS account ids whom the AMI is shared
  + A block device mapping that specifies the volumes to attach to the instance when it’s launched
* AMIs can be either
  + AWS managed, provided and published AMIs
  + Third party or Community provided public custom AMIs
  + Private AMIs created by other AWS accounts and shared with you
  + Private and Custom AMIs created by you

## AMI Types

* Region & Availability Zone
  + AMIs are specific to a region, but can be copied over to other regions
* Operating system
  + AMIS are available in a variety of OS flavors for e.g. linux, windows, ubuntu etc
* Architecture (32-bit or 64-bit)
* Launch Permissions
  + Launch permissions define who has access to the AMI
    - Public – Accessible to all AWS accounts
    - Explicit – Shared with specific AWS accounts
    - Private/Implicit – Owned and available for AMI creator account only
* Root device storage
  + AMIs can have EBS or Instance store as the root device storage
  + EBS backed
    - EBS volume are independent of the EC2 instance lifecycle and can persist independently
    - EBS backed instances can be stopped without losing the volumes
    - EBS instance can also be persist without losing the volumes on instance termination, if the Delete On Termination flag is disabled
    - EBS backed instances boot up much faster than the Instance store backed instances as only the parts required to boot the instance needs to be retrieved from the snapshot before the instance is made available
    - AMI creation is much easier for AMIs backed by Amazon EBS. The CreateImage API action creates your Amazon EBS-backed AMI and registers it
  + Instance Store backed
    - Instance store is an ephemeral storage and is dependent on the lifecycle of the Instance
    - Instance store is deleted if the instance is terminated or if the EBS backed instance, with attached instance store volumes, is stopped
    - Instance store volumes cannot be stopped
    - Instance store volumes have their AMI in S3 and have higher boot times compared to EBS backed instances, as all the parts have to be retrieved from Amazon S3 before the instance is made available
    - To create Linux AMIs backed by instance store, you must create an AMI from your instance on the instance itself using the Amazon EC2 AMI tools.

More detailed @ [EBS vs Instance Store](https://jayendrapatil.com/aws-ebs-vs-instance-store/)

## Linux Virtualization Types



* Linux Amazon Machine Images use one of two types of virtualization**: paravirtual (PV) or hardware virtual machine (HVM).**
* Main difference between PV and HVM AMIs is the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.
* For the best performance, AWS recommends use of current generation instance types and HVM AMIs when launching instances.
* 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 of the image.
  + HVM virtualization type provides the ability to run an operating system directly on top of a virtual machine without any modification, as if it were run on the bare-metal hardware.
  + EC2 host system emulates some or all of the underlying hardware that is presented to the guest.
  + HVM guests, unlike PV guest, can take advantage of hardware extensions that provide fast access to the underlying hardware on the host system.
  + HVM AMIs are required **to take advantage of enhanced networking and GPU processing**. In order to pass through instructions to specialized network and GPU devices, the OS needs to be able to have access to the native hardware platform; HVM virtualization provides this access.
  + All current generation instance types support HVM AMIs.The CC2, CR1, HI1, and HS1 previous generation instance types support HVM AMIs.
* 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.lst file on your image.
  + 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**
  + C3 and M3 current generation instance types support PV AMIs. The C1, HI1, HS1, M1, M2, and T1 previous generation instance types support PV AMIs.

## Shared AMIs

* Shared AMI is an AMI that can be created and shared with others for use
* A Shared AMI with all the components needed can be used to get started and then add custom components as and when needed
* Shared AMI can be risky as Amazon does not perform detailed checks and vouch for the integrity and security of these AMIs
* Before using a Shared AMI, check for any pre-installed credentials that would allow unwanted access to the instance by a third party and no pre-configured remote logging that could transmit sensitive data to a third party
* Amazon allows you to share an AMI, by defining launch permissions, to all (making it public) or only to a specific AWS accounts
* Launch permissions work at the AWS account level only; and can’t be used to restrict specific users within an AWS account.
* Guidelines for Shared Linux AMIs
  + Update the AMI Tools at Boot Time
    - Update the AMI tools or any software during startup.
    - Take into account the software updates not not break any software and consider the WAN traffic as the downloads will be charged to the AMI user
  + Disable Password-Based Remote Logins for Root
    - Fixed root passwords can be a security risk and needs to be disabled
  + Disable Local Root Access
    - disable direct root logins
  + Remove SSH Host Key Pairs
    - Remove the existing SSH host key pairs located in /etc/ssh, which forces SSH to generate new unique SSH key pairs when someone launches an instance using your AMI, improving security and reducing the likelihood of “man-in-the-middle” attacks
  + Install Public Key Credentials
    - EC2 allows users to specify a public-private key pair name when launching an instance.
    - A valid key pair name needs to be provided when launching an instance, the public key, the portion of the key pair that EC2 maintains on the server, is made available to the instance through an HTTP query against the instance metadata and appended to the authorized keys
    - Users can launch instances of the AMI with a key pair and log in without requiring a root password
  + Disabling sshd DNS Checks (Optional)
    - Disabling sshd DNS checks slightly weakens your sshd security. However, if DNS resolution fails, SSH logins still work. If you do not disable sshd checks, DNS resolution failures prevent all logins.
  + Identify Yourself
    - AMI is only represented by an account ID without any further information, so it is better to provide more information to help describe the AMI
  + Protect Yourself
    - Don’t store any sensitive data or software on the AMI
    - Exclude & Skip any directories holding sensitive data or secret information and delete the shell history before creating an AMI

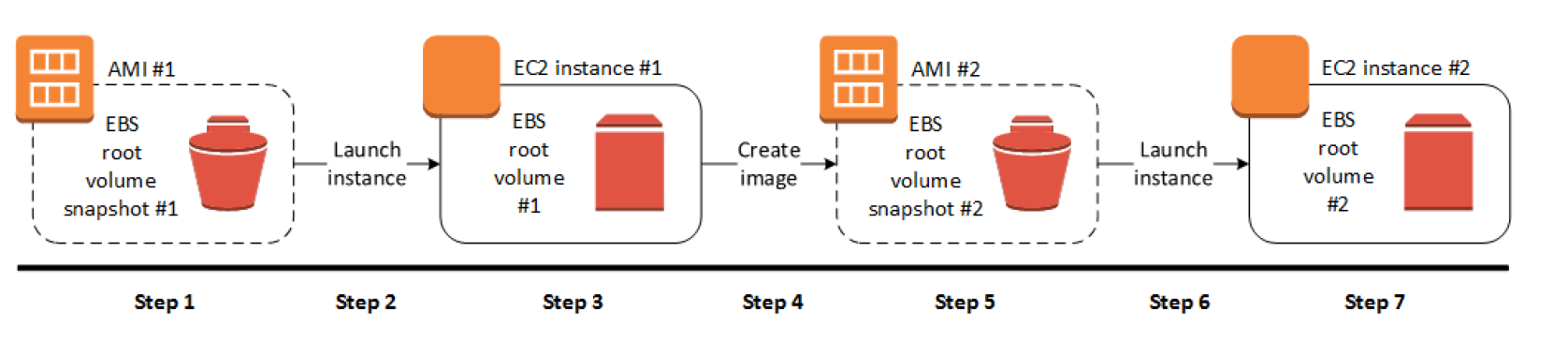
## AMI lifecycle

* + Create and register an AMI
  + launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions)
  + Copy an AMI to the same region or to different regions.
  + Deregister the AMI, when finished launching instance from an AMI

## AMI Lifecycle

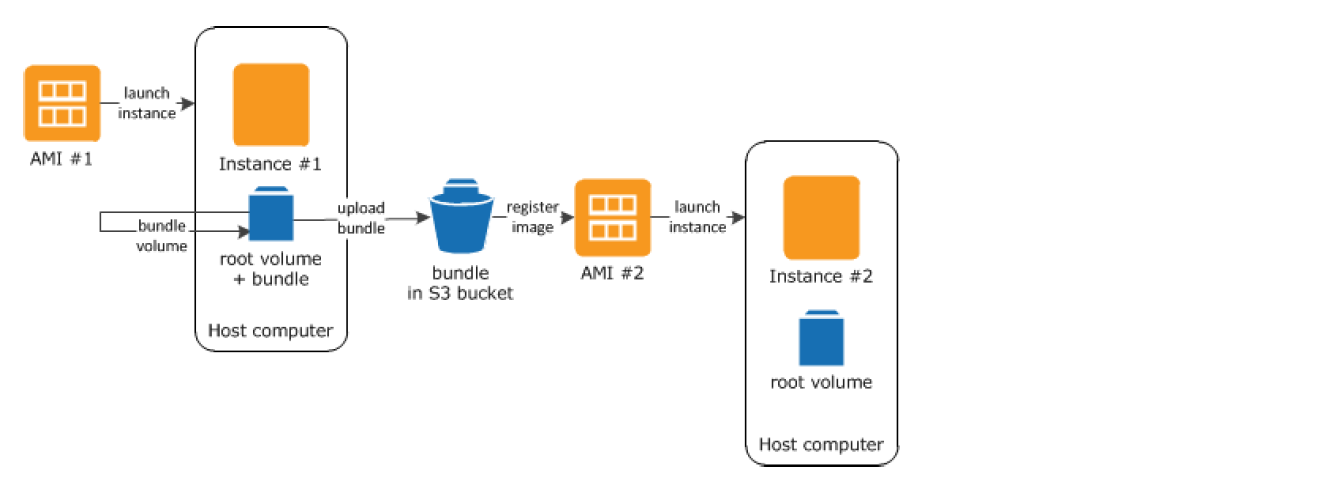
### AMI Creation

#### **EBS-Backed Linux AMI**



* EBS-Backed Linux AMI can be created from the instance directly or from the EBS snapshot
* EBS-backed linux AMI creation process :-
  1. Select an AMI #1 similar to what you want to have your new AMI #2
  2. Launch an Instance from AMI #1 and configure it accordingly
  3. Stop the instance to ensure data integrity
  4. Create AMI #2 OR create an EBS snapshot and then create an AMI #2 from the snapshot
  5. Amazon automatically register the EBS-backed AMI
  6. AMI #2 can be now used to launch new instances
* By default, EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance.
* Shut down & Reboot of the Instance can be prevented using the **No reboot** option, however the file system integrity of the created image can’t be guaranteed
* EC2 creates snapshots of the instance’s root volume and any other EBS volumes attached to the instance. If any volumes attached to the instance are encrypted, the new AMI only launches successfully on instances that support Amazon EBS encryption
* For any additional instance-store volumes or EBS volumes, 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.
* While data on EBS volumes persists, 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.
* Its more efficient to create an EBS-backed AMI with EBS snapshots already taken as the snapshot created during AMI creation is just an incremental one
* You are charged for the storage of both the AMI and the snapshots

#### **Instance Store-Backed Linux AMI**

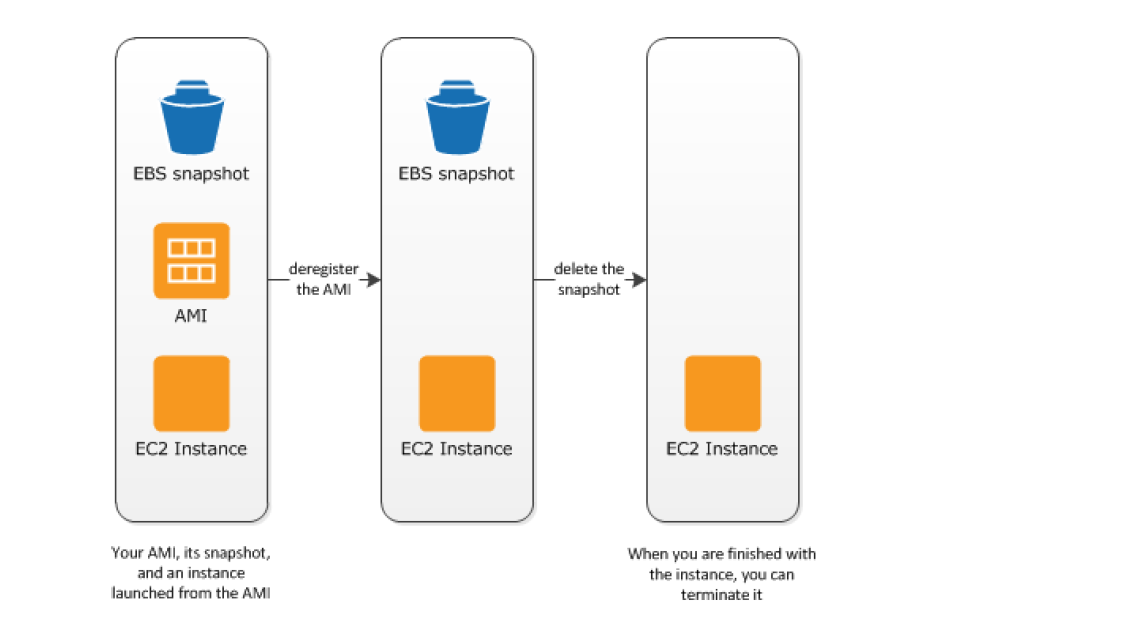


* Instance Store-Backed Linux AMI creation process
  1. Select an AMI #1 similar to what you want to have your new AMI #2
  2. Launch an Instance from AMI #1 and configure the instance accordingly
  3. Bundle the Instance. It takes several minutes for the bundling process to complete.
  4. After the process completes, you have a bundle, which consists of an image manifest (image.manifest.xml) and files (image.part.xx) that contain a template for the root volume.
  5. Upload the bundle to the S3 bucket
  6. Register the Instance Store-backed AMI.
  7. Launching an instance using the new AMI #2, the root volume for the instance is created using the bundle that you uploaded to S3.
* Charges are incurred for the storage space used by the bundle in S3 until deleted
* For additional instance store not root volumes, 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.

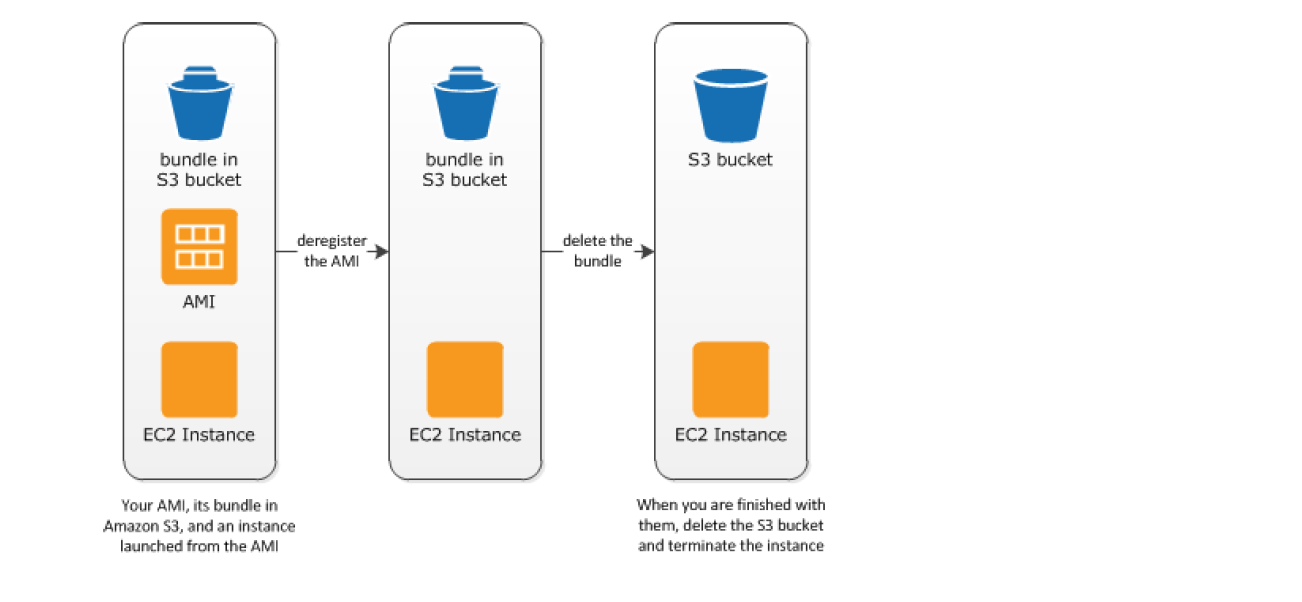
### Amazon EBS-backed AMI vs Instance Store-backed AMI

### Deregistering AMI

* Charges are incurred on the AMI created and they can be deregistered, if not needed.
* Deregistering an AMI does not delete the EBS snapshots or the bundles in the S3 buckets and have to be removed separately
* Once deregistered, new instances cannot be launched from the AMI. However, it does not impact already created instances from the AMI
* Clean up EBS-Backed AMI



* + Deregister the EBS-Backed AMI
  + Delete the EBS Snapshot, as deregistering the AMI doesn’t impact the snapshot
    - Clean up Instance Store-backed AMI



* + Deregister the EBS-Backed AMI
  + Delete the bundle from the S3 bucket, as deregistering the AMI doesn’t effect the bundles stored in the S3 bucket

## AMIs with Encrypted Snapshots

* AMIs, with EBS backed snapshots, can be attached with both an encrypted root and data volume
* AMIs copy image can be used to create AMIs with encrypted snapshots from AMIs with unencrypted snapshots. By default, copy image preserves the encryption status of the snapshots
* Snapshots can be encrypted with either default AWS Key Management Service customer master key (CMK), or with a custom key that you specify

## AMI Copying

* EBS-backed AMIs and instance store-backed AMIs can be copied.
* Copying an AMI
  + An identical target AMI is created, but with its own unique identifier
  + For EBS backed AMI, an identical but distinct root and data snapshots are created
  + Encryption status of the snapshots are preserved
  + However, Launch permissions, user-defined tags, or S3 bucket permissions are not copied from the source AMI to the new AMI. After the copy operation is complete, different launch permissions, user-defined tags, and S3 bucket permissions to the new AMI
* Source AMI can be deregistered without any impact to the Target AMI
* AMIs owned or shared with proper permissions can be copied
* AMIs are created specific to a region and can be copied within or across regions which can help to aid in consistent global deployment and build highly scalable and available applications
* AMI copy image can be used to encrypt an AMI from an unencrypted AMI
* **AMIs with encrypted snapshots can be copied and also encryption status changed during the copy process**
* AWS Marketplace AMI cannot be copied, regardless of whether obtained directly or shared. Instead, launch an EC2 instance using the AWS Marketplace AMI and then create an AMI from the instance.

## Amazon Linux 2 and Amazon Linux AMI

Amazon Linux 2 and Linux AMI is a supported and maintained Linux image provided by AWS with the following features

* A stable, secure, and high-performance execution environment for applications running on EC2.
  + does not allow remote root SSH by default.
  + Password authentication is disabled to prevent brute-force password attacks.
  + Instances launched using Amazon Linux AMI must be provided with a key pair at launch to enable SSH logins
  + Inbound security group must allow SSH access
  + By default, the only account that can log in remotely using SSH is ec2-user ; this account also has sudo privileges.
  + are configured to download and install security updates at launch time.
* Provided at no additional charge to Amazon EC2 users.
* Repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, Tomcat, and many more common packages.
* Updated on a regular basis to include the latest components, and these updates are also made available in the yum repositories for installation on running instances.
* Includes pre-installed packages to enable easy integration with AWS services, such as the AWS CLI, Amazon EC2 API and AMI tools, the Boto library for Python, and the Elastic Load Balancing tools.

## 

## AWS Certification Exam Practice Questions

* Questions are collected from Internet and the answers are marked as per my knowledge and understanding (which might differ with yours).
* AWS services are updated everyday and both the answers and questions might be outdated soon, so research accordingly.
* AWS exam questions are not updated to keep up the pace with AWS updates, so even if the underlying feature has changed the question might not be updated
* Open to further feedback, discussion and correction.

1. A user has launched an EC2 instance from an instance store backed AMI. The infrastructure team wants to create an AMI from the running instance. Which of the below mentioned credentials is not required while creating the AMI?
   1. AWS account ID
   2. 509 certificate and private key
   3. **AWS login ID to login to the console**
   4. Access key and secret access key
2. A user has launched an EC2 Windows instance from an instance store backed AMI. The user wants to convert the AMI to an EBS backed AMI. How can the user convert it?
   1. Attach an EBS volume to the instance and unbundle all the AMI bundled data inside the EBS
   2. **A Windows based instance store backed AMI cannot be converted to an EBS backed AMI**
   3. It is not possible to convert an instance store backed AMI to an EBS backed AMI
   4. Attach an EBS volume and use the copy command to copy all the ephemeral content to the EBS Volume
3. A user has launched two EBS backed EC2 instances in the US-East-1a region. The user wants to change the zone of one of the instances. How can the user change it?
   1. Stop one of the instances and change the availability zone
   2. The zone can only be modified using the AWS CLI
   3. From the AWS EC2 console, select the Actions – > Change zones and specify new zone
   4. **Create an AMI of the running instance and launch the instance in a separate AZ**
4. A user has launched a large EBS backed EC2 instance in the US-East-1a region. The user wants to achieve Disaster Recovery (DR) for that instance by creating another small instance in Europe. How can the user achieve DR?
   1. Copy the running instance using the “Instance Copy” command to the EU region
   2. **Create an AMI of the instance and copy the AMI to the EU region. Then launch the instance from the EU AMI**
   3. Copy the instance from the US East region to the EU region
   4. Use the “Launch more like this” option to copy the instance from one region to another
5. A user has launched an EC2 instance store backed instance in the US-East-1a zone. The user created AMI #1 and copied it to the Europe region. After that, the user made a few updates to the application running in the US-East-1a zone. The user makes an AMI#2 after the changes. If the user launches a new instance in Europe from the AMI #1 copy, which of the below mentioned statements is true?
   1. The new instance will have the changes made after the AMI copy as AWS just copies the reference of the original AMI during the copying. Thus, the copied AMI will have all the updated data
   2. The new instance will have the changes made after the AMI copy since AWS keeps updating the AMI
   3. It is not possible to copy the instance store backed AMI from one region to another
   4. **The new instance in the EU region will not have the changes made after the AMI copy**
6. George has shared an EC2 AMI created in the US East region from his AWS account with Stefano. George copies the same AMI to the US West region. Can Stefano access the copied AMI of George’s account from the US West region?
   1. **No, copy AMI does not copy the permission**
   2. It is not possible to share the AMI with a specific account
   3. Yes, since copy AMI copies all private account sharing permissions
   4. Yes, since copy AMI copies all the permissions attached with the AMI
7. EC2 instances are launched from Amazon Machine images (AMIS). A given public AMI can:
   1. be used to launch EC2 Instances in any AWS region.
   2. only be used to launch EC2 instances in the same country as the AMI is stored.
   3. **only be used to launch EC2 instances in the same AWS region as the AMI is stored.**(An AMI is tied to the region where its files are located within Amazon S3)
   4. only be used to launch EC2 instances in the same AWS availability zone as the AMI is stored.