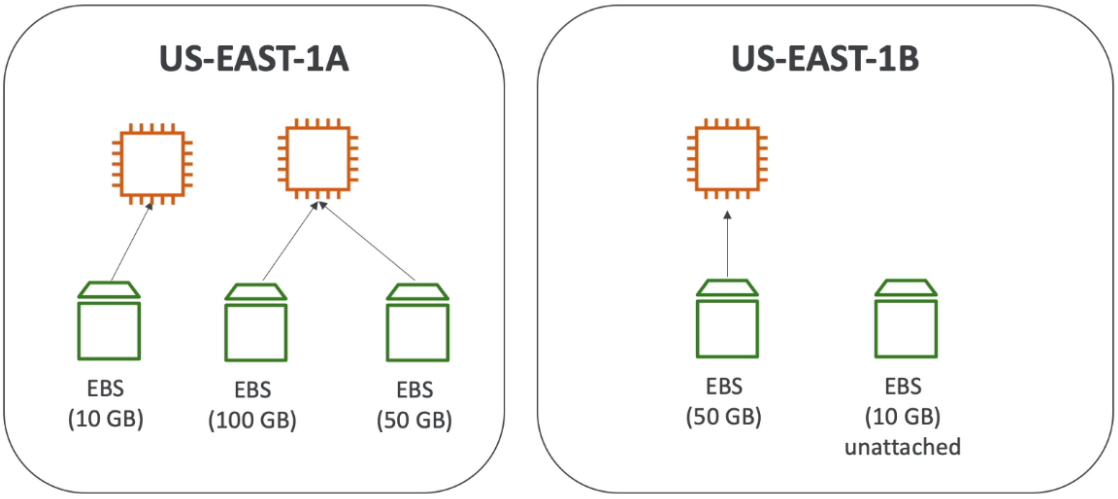
EC2 Storage and stuff

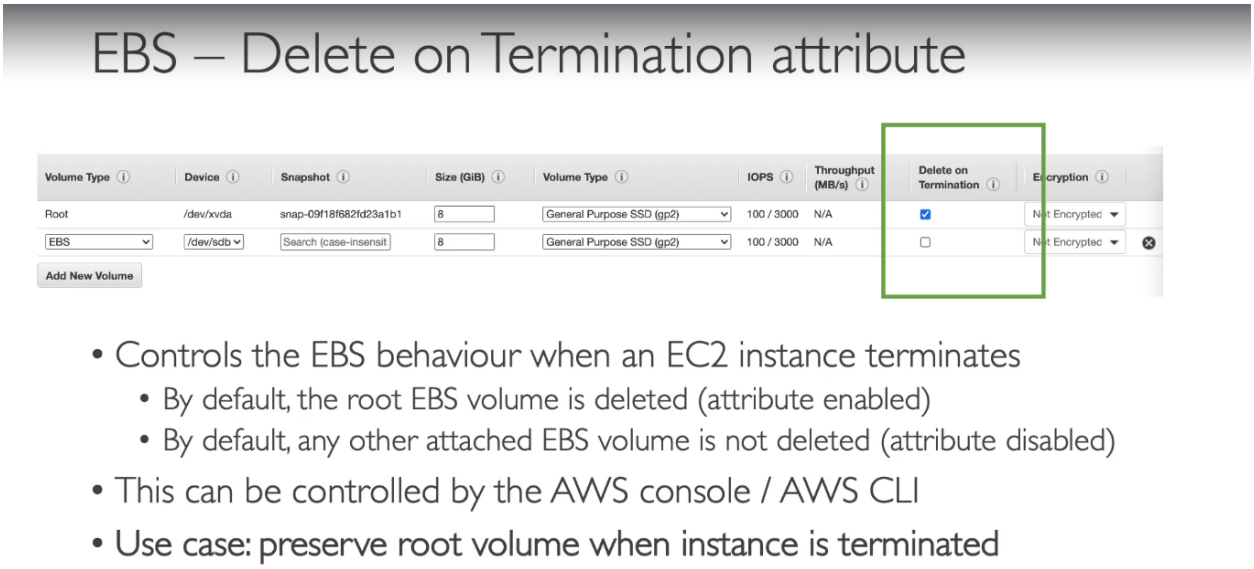
EBS Volumes

- EBS is short for elastic block storage
- It is basically a network drive (not a physical one) you can attach to your instances while they run.
- EBS allows your instance to persist the data even after termination. So then later, you can attach the same block of storage to another instance and get your data back.
- **An EBS volume can be attached to only one instance at a time. But one instance can have multiple volumes attached to it.** Also, you don't necessarily have to attach the block of volume to an instance, you can attach it on-demand, which makes it powerful.



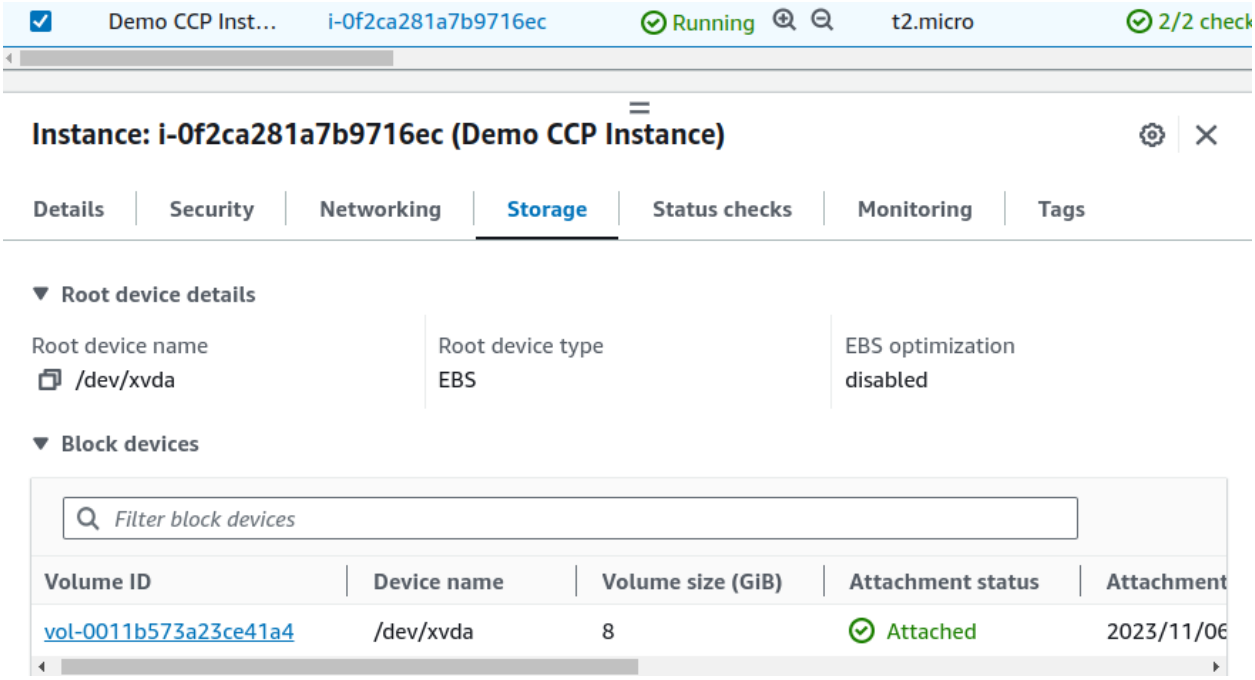
- Also, they are **locked to availability zones**. You cannot attach an EBS volume created within ap-south-1 to an instance from ap-south-3. (though can you create a snapshot to move the volume around)
- An EBS volume can communicate to its instance through the internet and hence, latency is possible.

- To create an EBS, you have **provision certain capacity (GBs or IOPS)**. You get billed for all the provisioned capacity, whether you use it or not. You can increase the capacity over time as per needs.

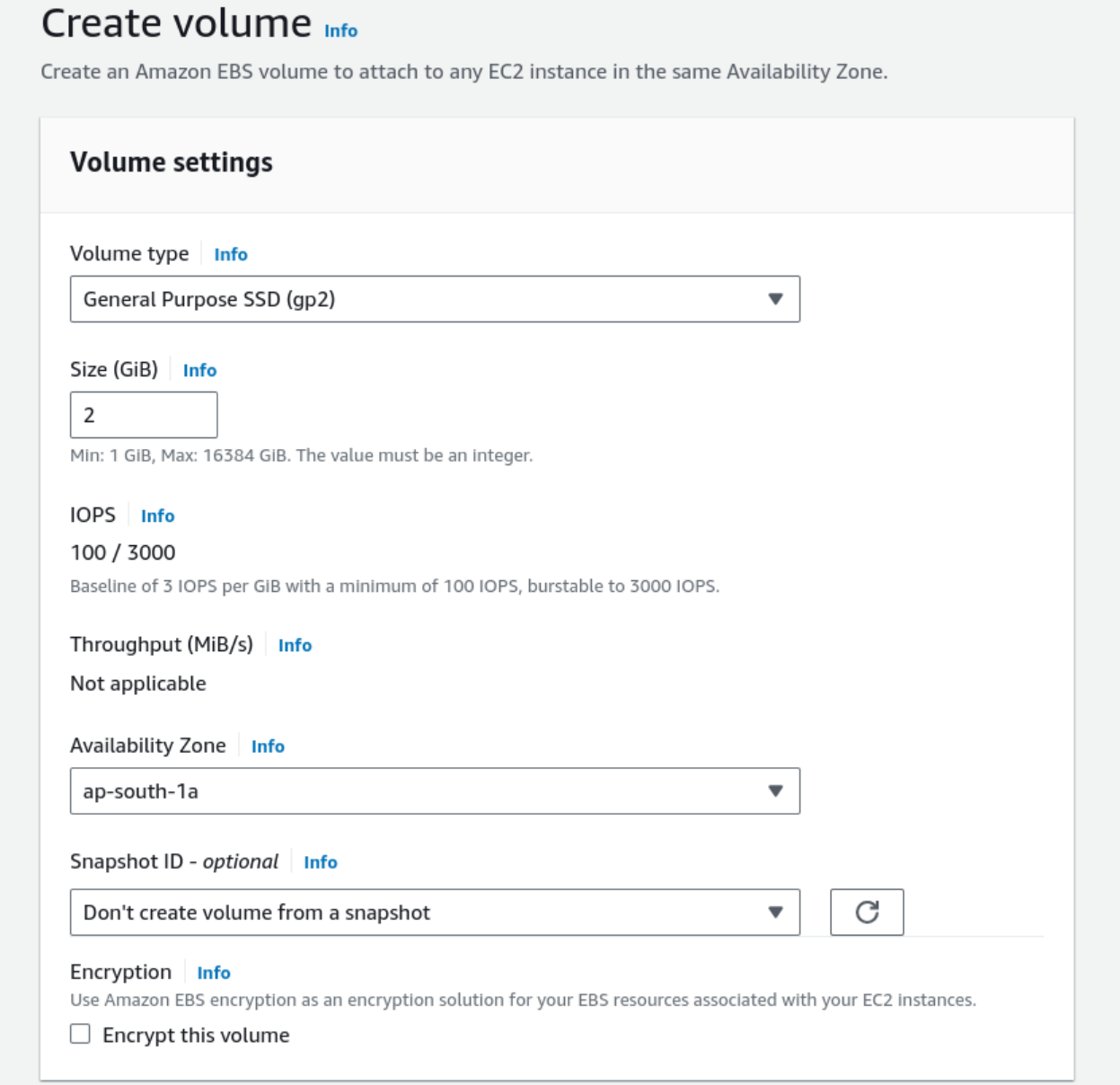


EBS Hands-on

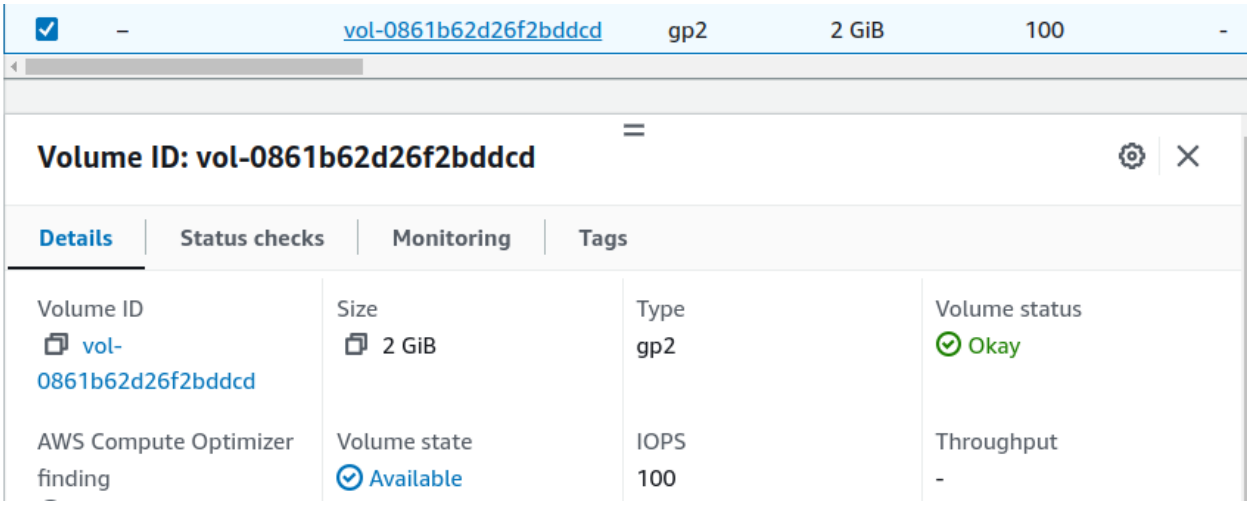
- We have unknowingly been using EBS since we created our first EC2 instance.



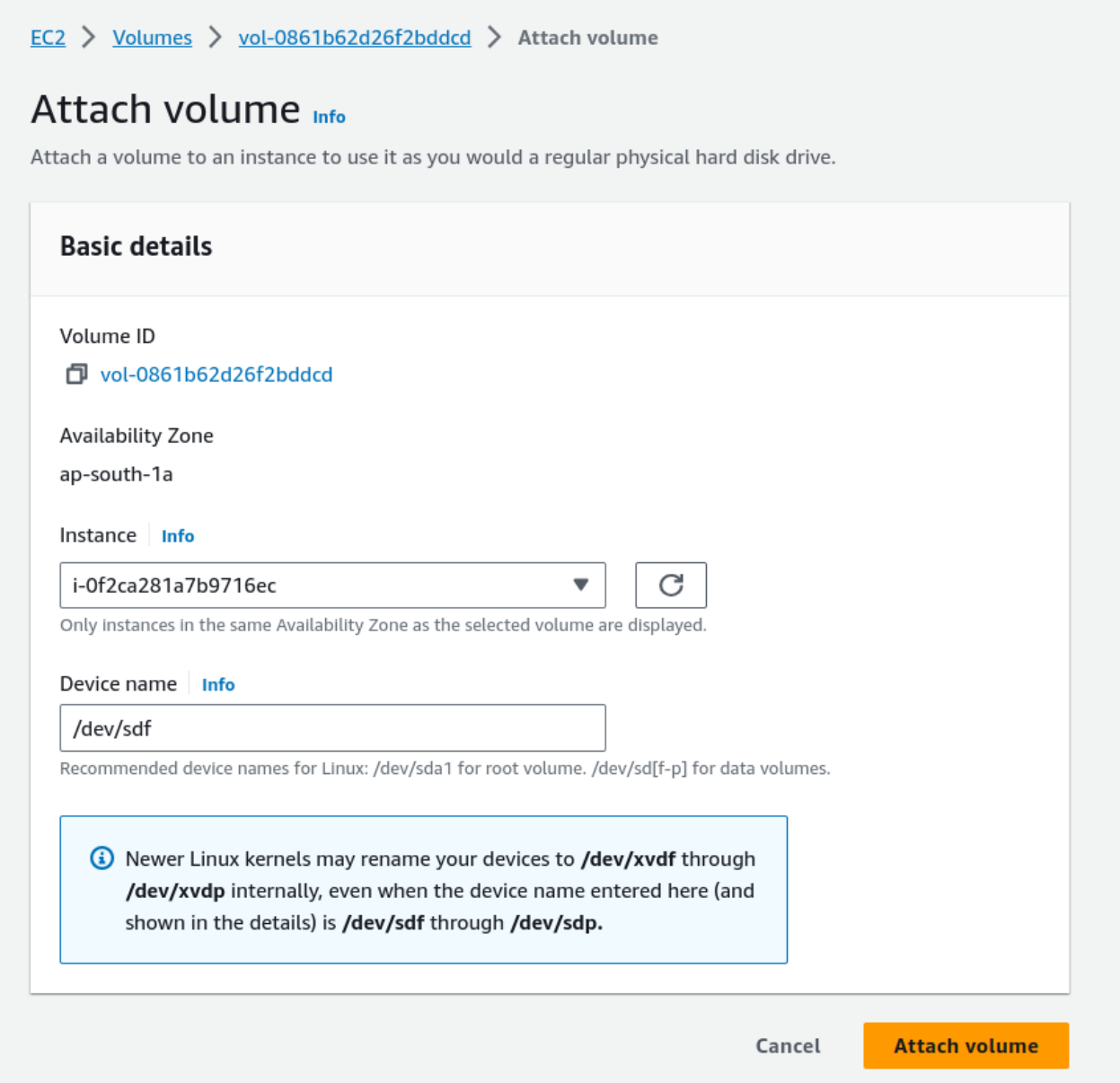
- Create EBS volume
- Choose the AZ same as your EC2 instance



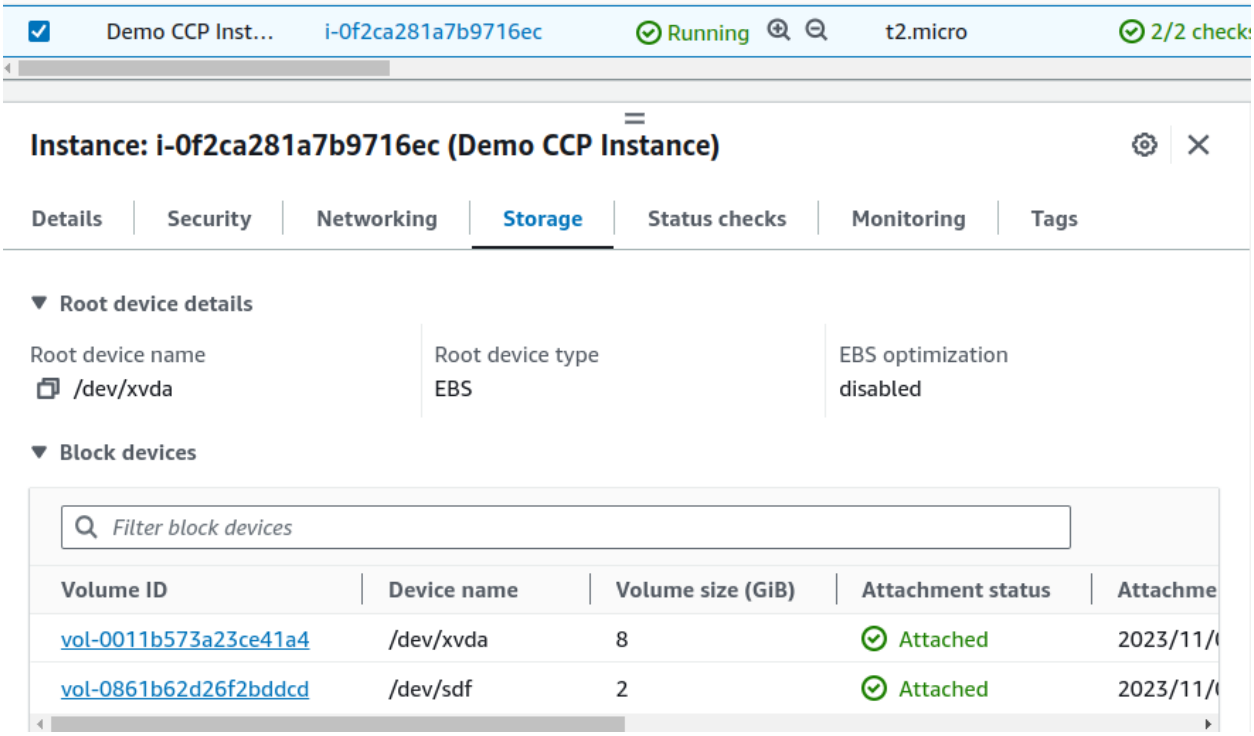
- Wait for the volume state to change to available



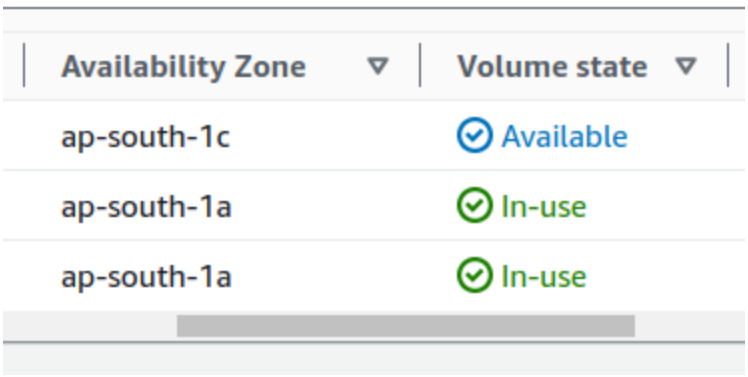
- Actions >> attach volume



- Cross check if attached, by checking instance's storage details



- We can create another EBS volume in an AZ where no instances exist and you won't be able to attach it to any instance.

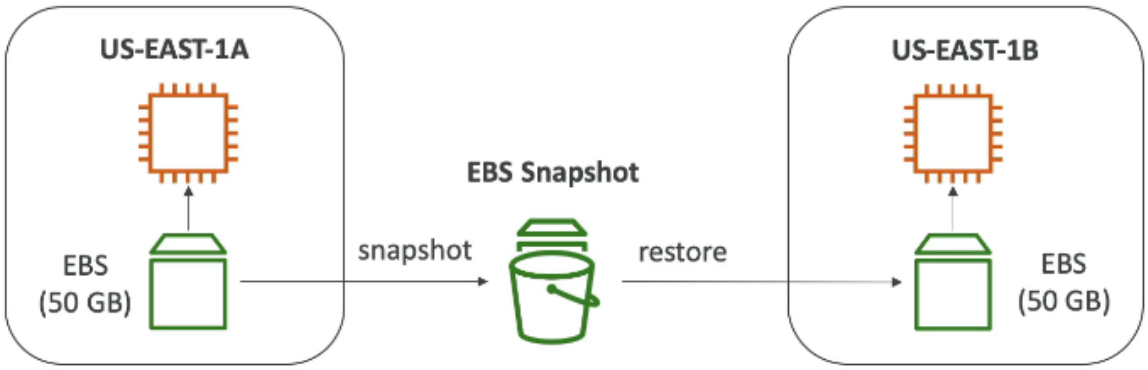


- Also, if you terminate your EC2 instance, then refresh the EBS volumes page, you will notice the already existing volume block (8 gib) is deleted automatically as it set to **yes** for `delete on termination` setting. This setting is available in the advanced storage settings while creating an instance and you can modify it from there.

Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS key ID	Delete on termination
8	Attached	2023/11/06 20:14 GMT+5:30	No	–	Yes
2	Attached	2023/11/08 19:11 GMT+5:30	No	–	No

EBS Snapshots

- EBS snapshots basically act as a backup for your EBS volumes. You can restore your volume to the exact state as when you took the snapshot, even if the volume has been terminated. (similar to versioning in a sense)
- It is recommended to detach the volume before taking a snapshot. This ensures consistency throughout the data.
- EBS volumes are locked to specific region or AZ but you can use snapshots to copy your data across different regions or AZs. This way you can leverage the global infrastructure.

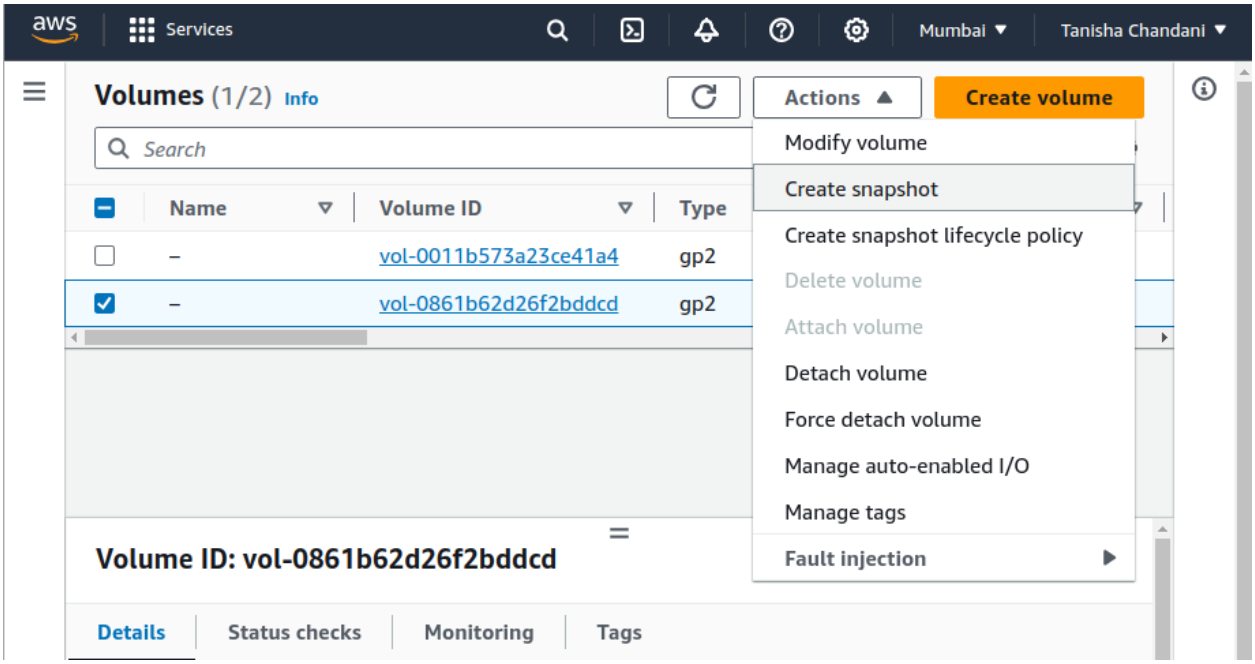


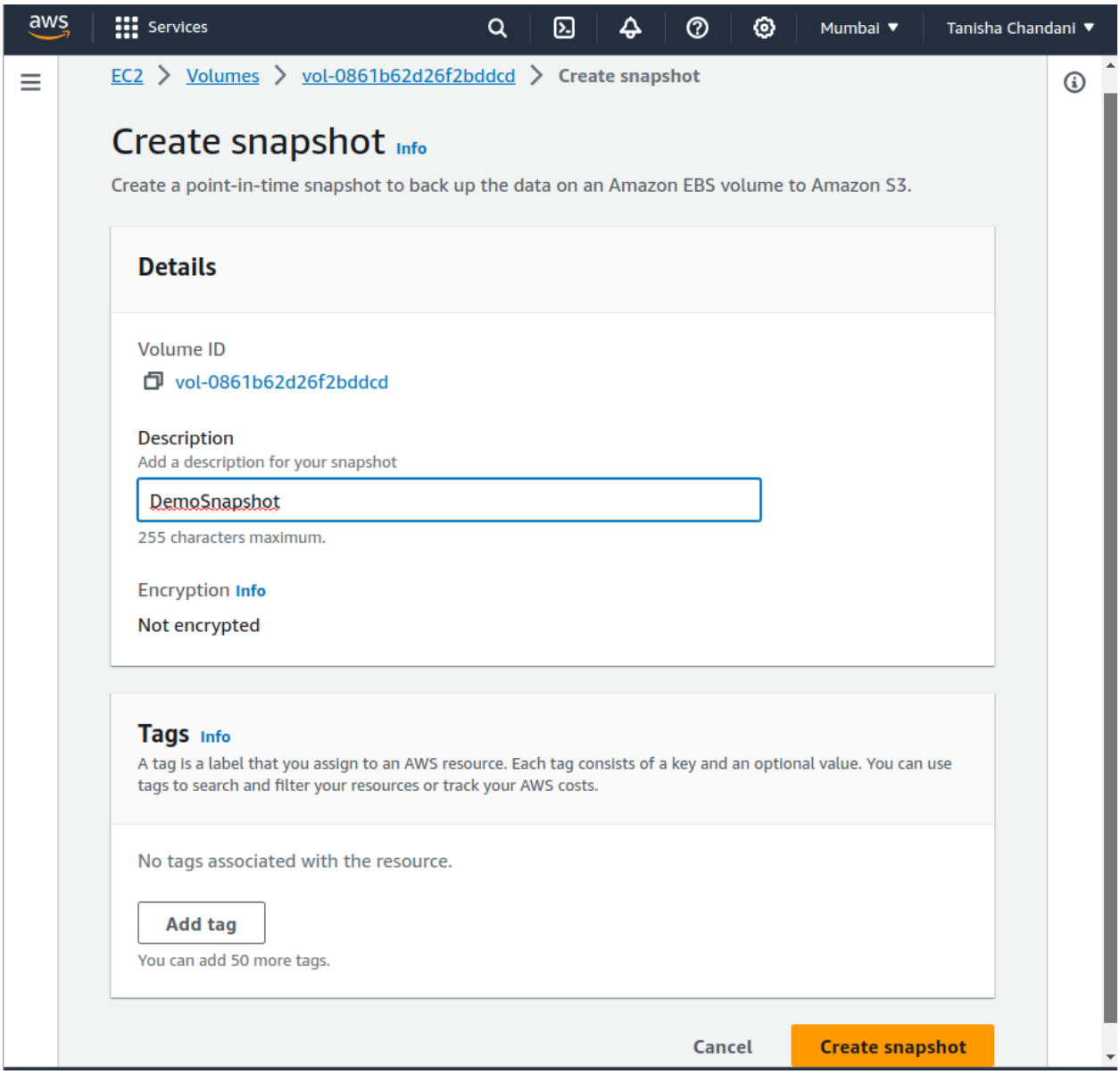
- **EBS Snapshots features**

1. **EBS snapshot archive** - Moves your snapshot to the archive tier which is 75% cheaper, hence, saving you a lot of money. It takes around 24-72h to restore the archive data.
2. **Recycle Bin for deleted snapshots** - for a specific retention period (you can set this), your deleted snapshots are stored in the bin. This can help with

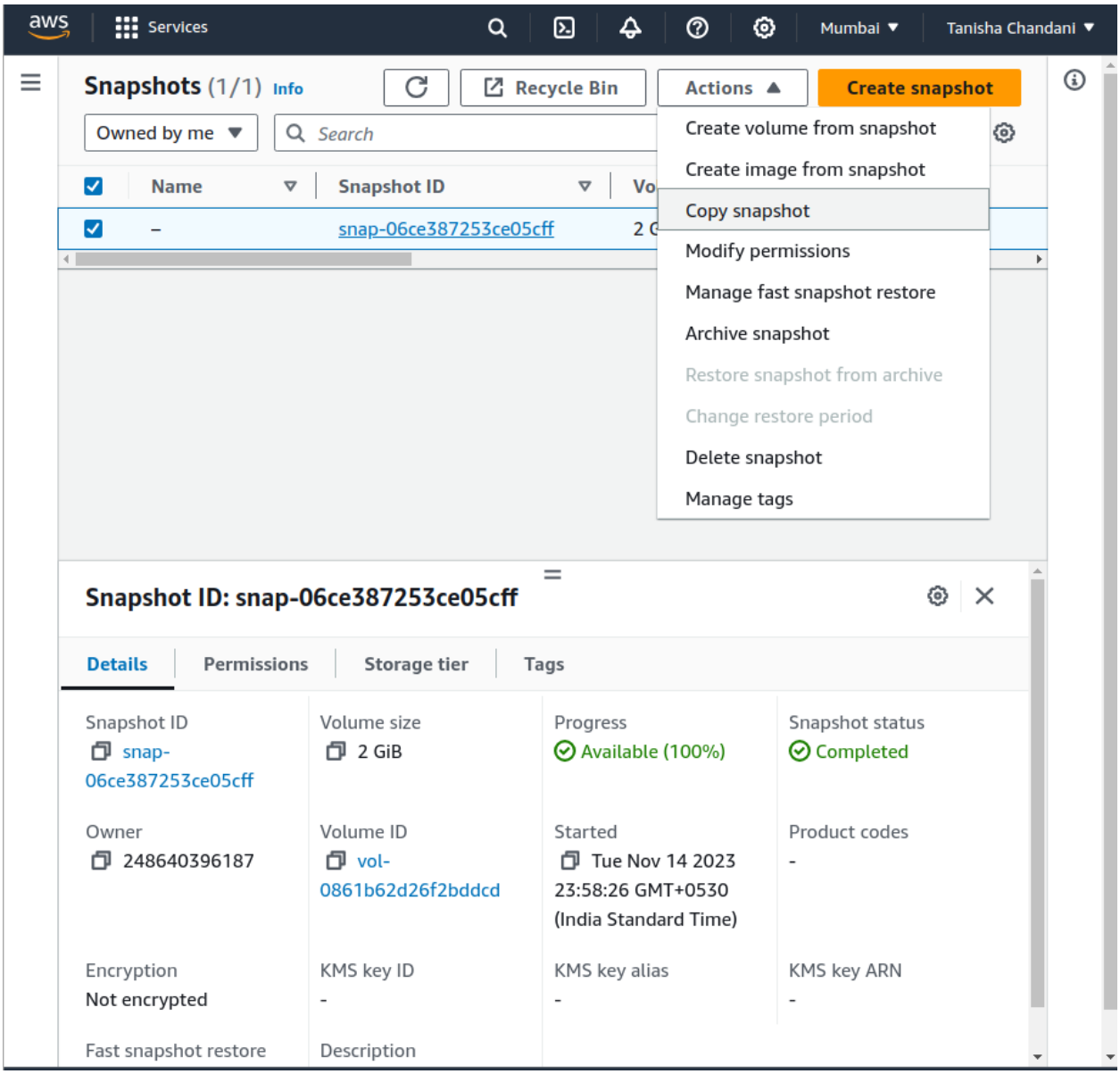
accidental deletes.

EBS Snapshots Hands-on

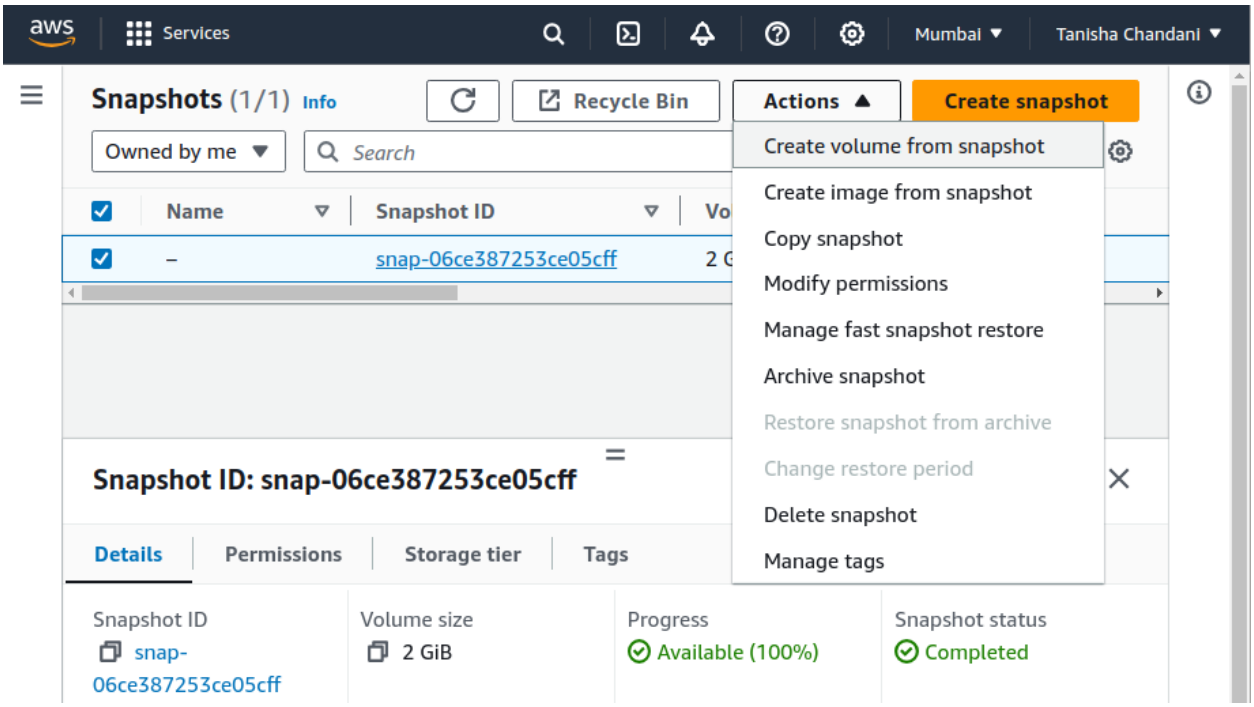




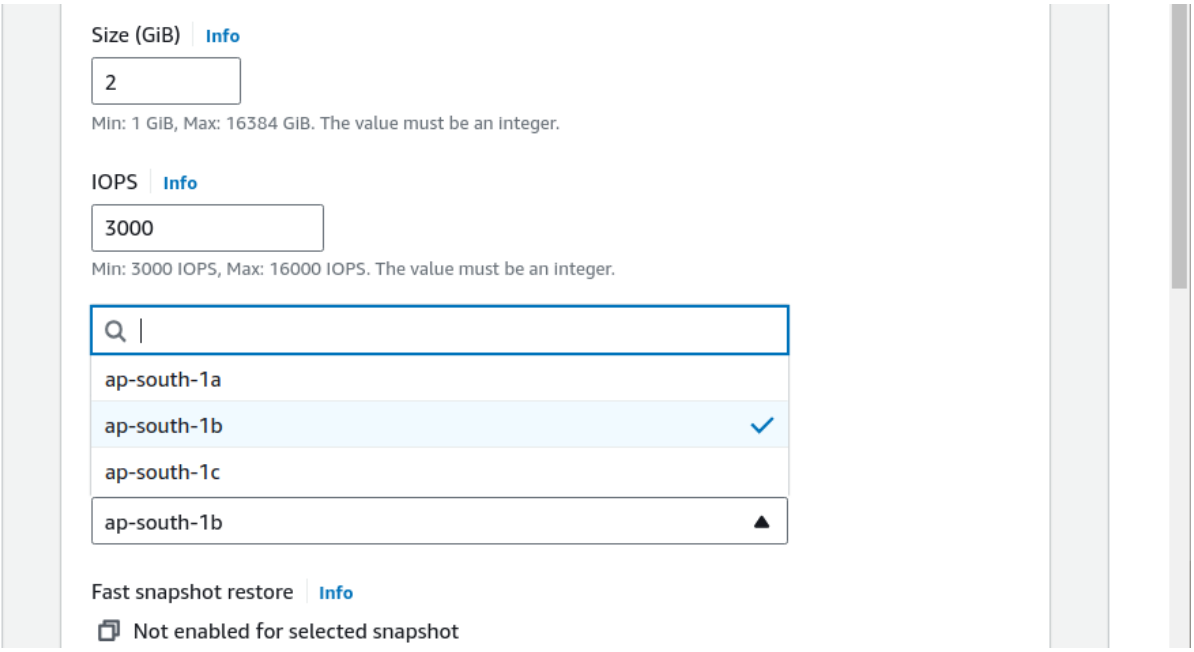
- Copying the created snapshot into another region



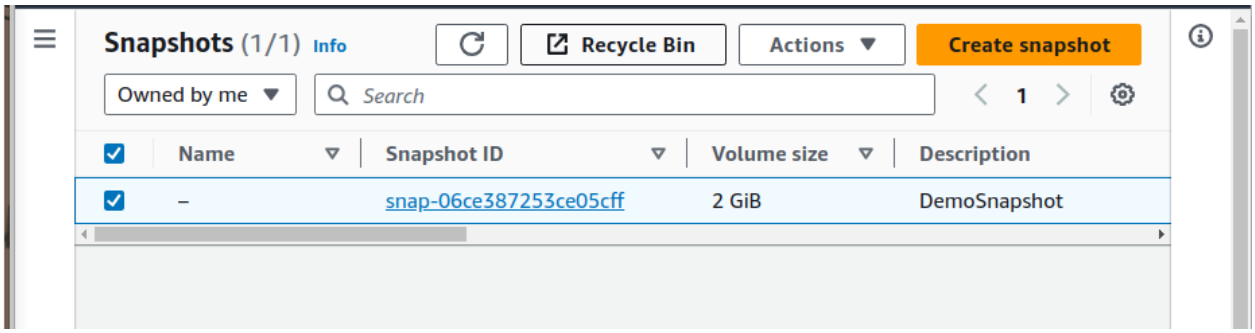
- Create a new volume from the snapshot

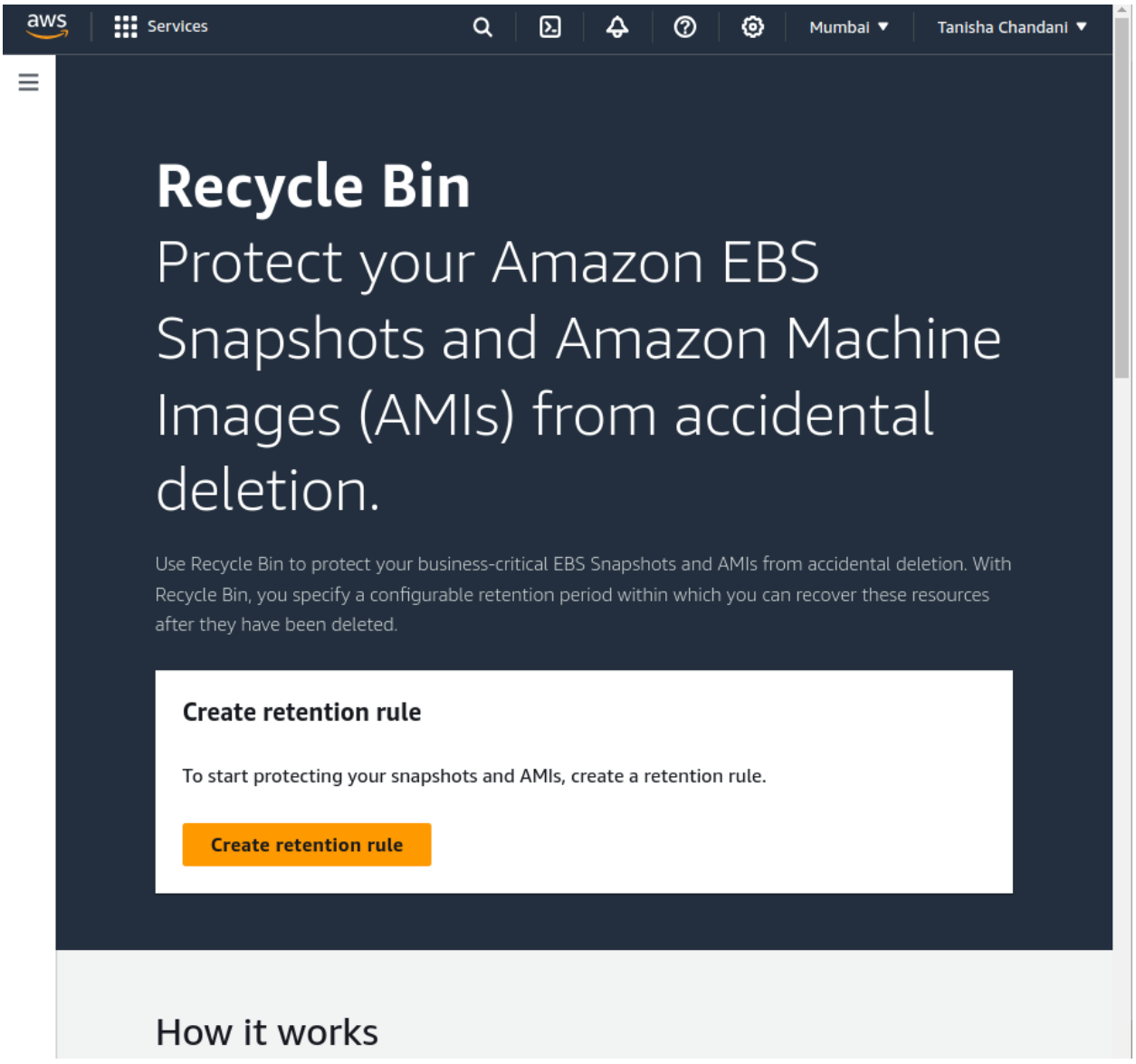


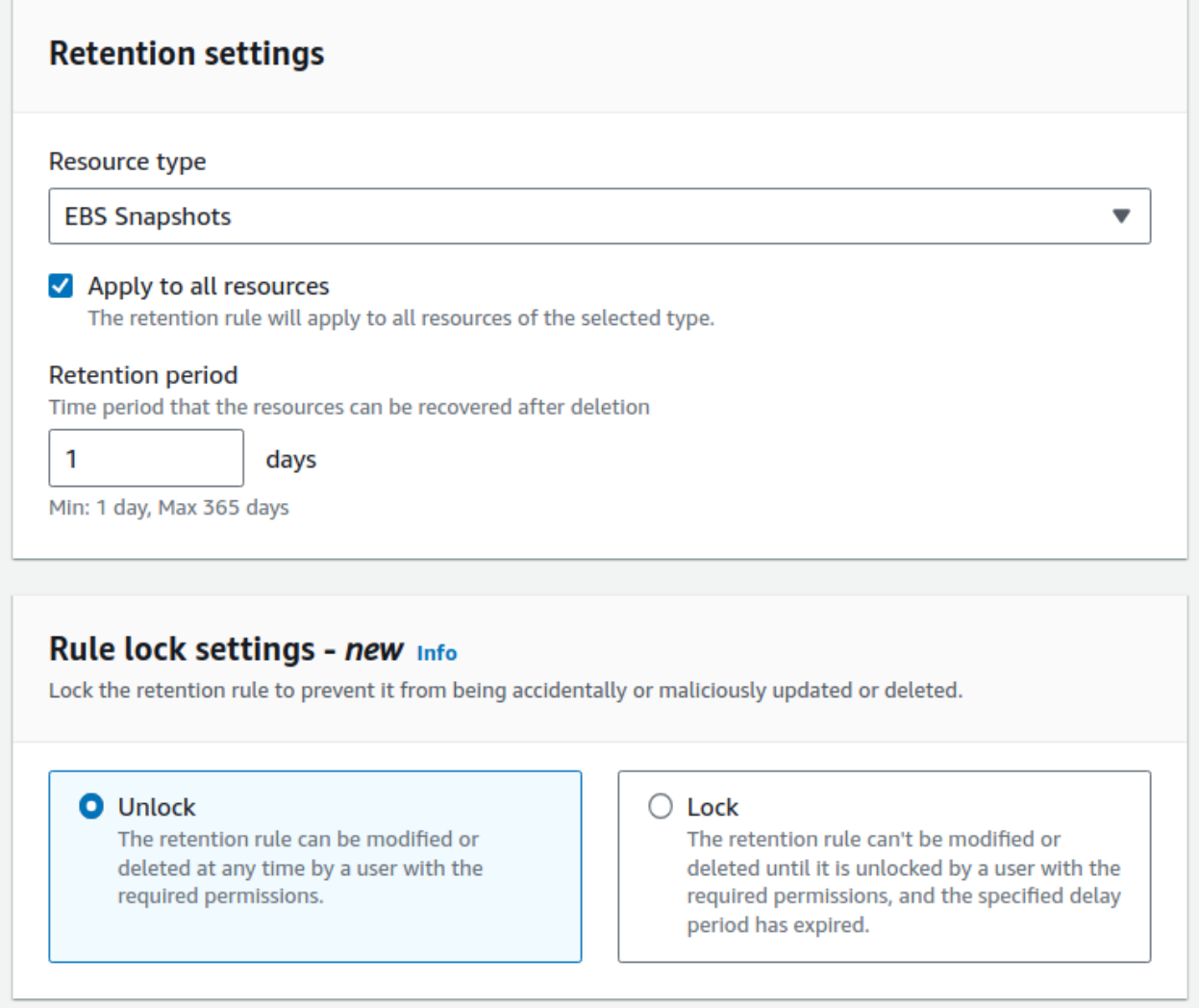
- As you create another volume through the snapshot, you can select a different AZ within the same region. (**Availability Zones are distinct locations within an AWS Region that are engineered to be isolated from failures in other Availability Zones)**



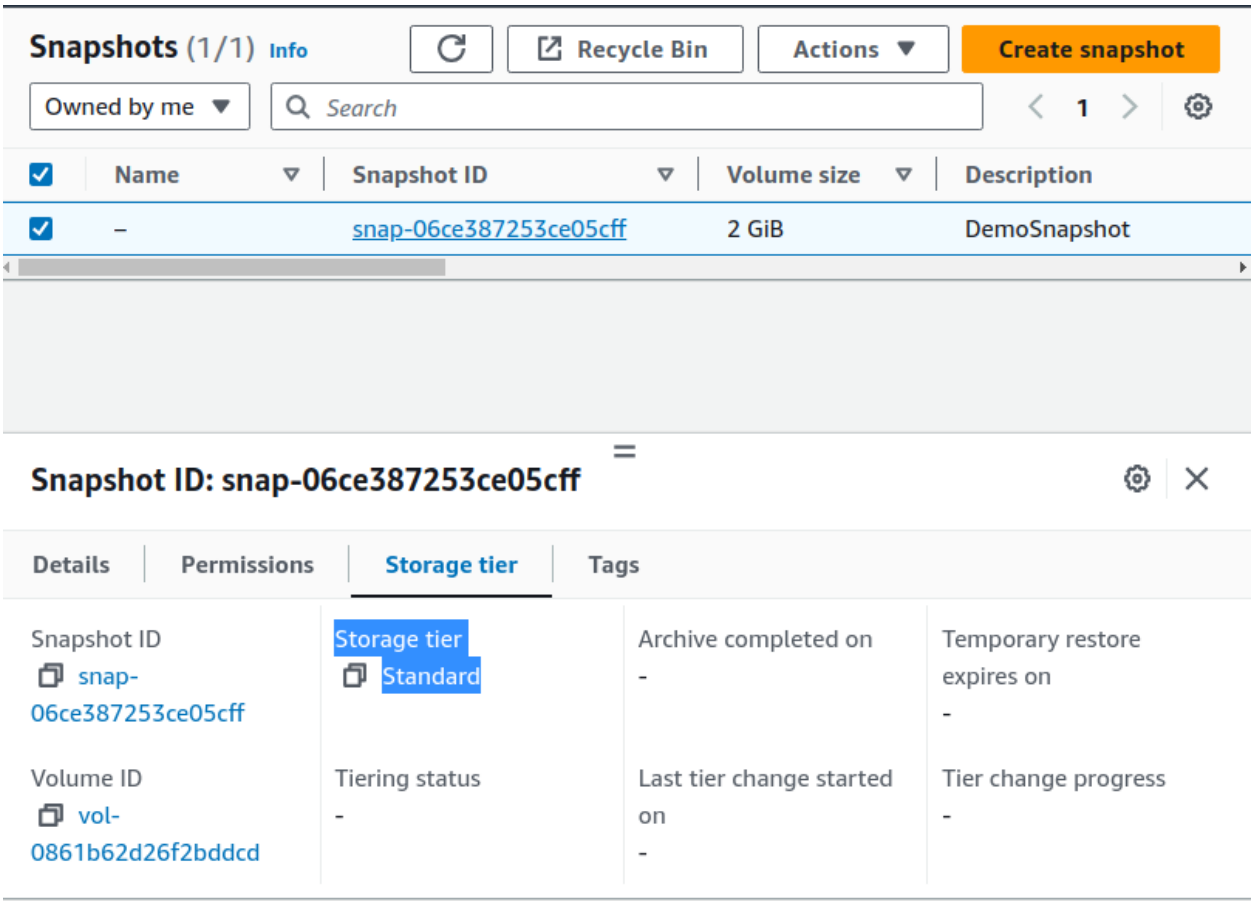
- Let us look at recycle bin for a snapshot


Snapshots (1/1) Info
Recycle Bin
Actions
Create snapshot
Owned by me
Search
1
Name
Snapshot ID
Volume size
Description
–
snap-06ce387253ce05cff
2 GiB
DemoSnapshot


aws
Services
Mumbai
Tanisha Chandani
Recycle Bin
Protect your Amazon EBS Snapshots and Amazon Machine Images (AMIs) from accidental deletion.
Use Recycle Bin to protect your business-critical EBS Snapshots and AMIs from accidental deletion. With Recycle Bin, you specify a configurable retention period within which you can recover these resources after they have been deleted.
Create retention rule
To start protecting your snapshots and AMIs, create a retention rule.
Create retention rule
How it works

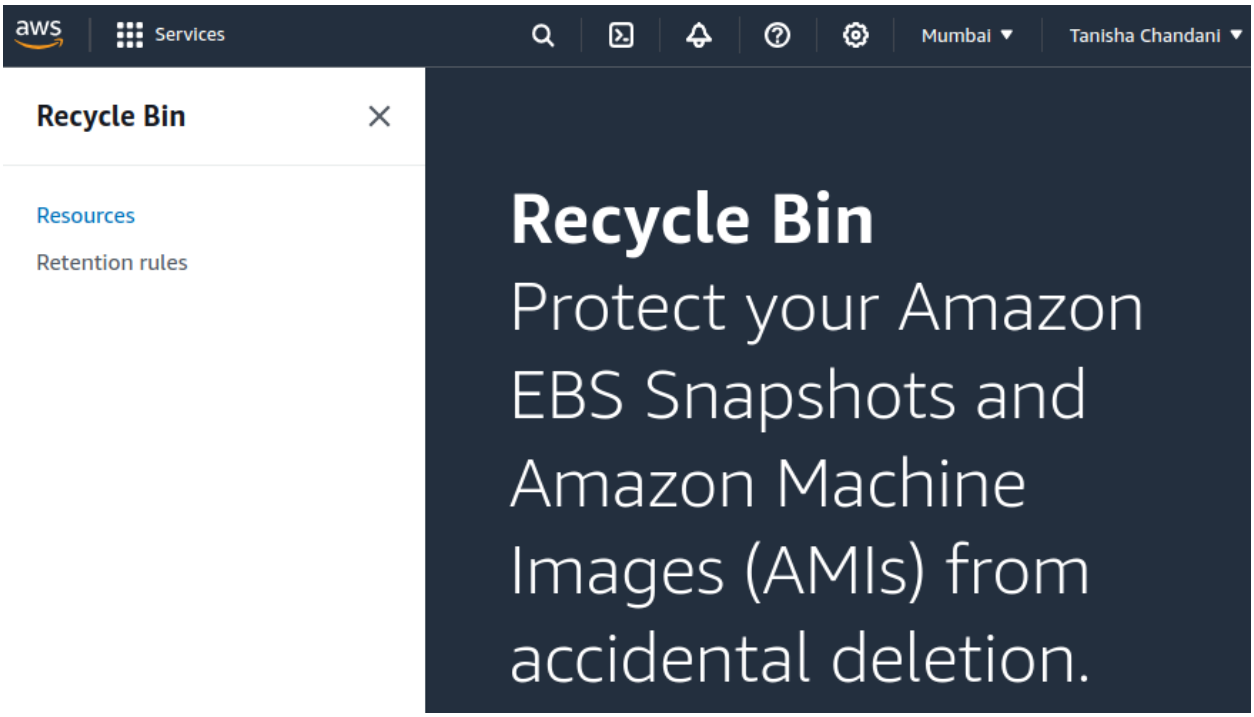


- We will now see how the recycle bin works by deleting a snapshot.

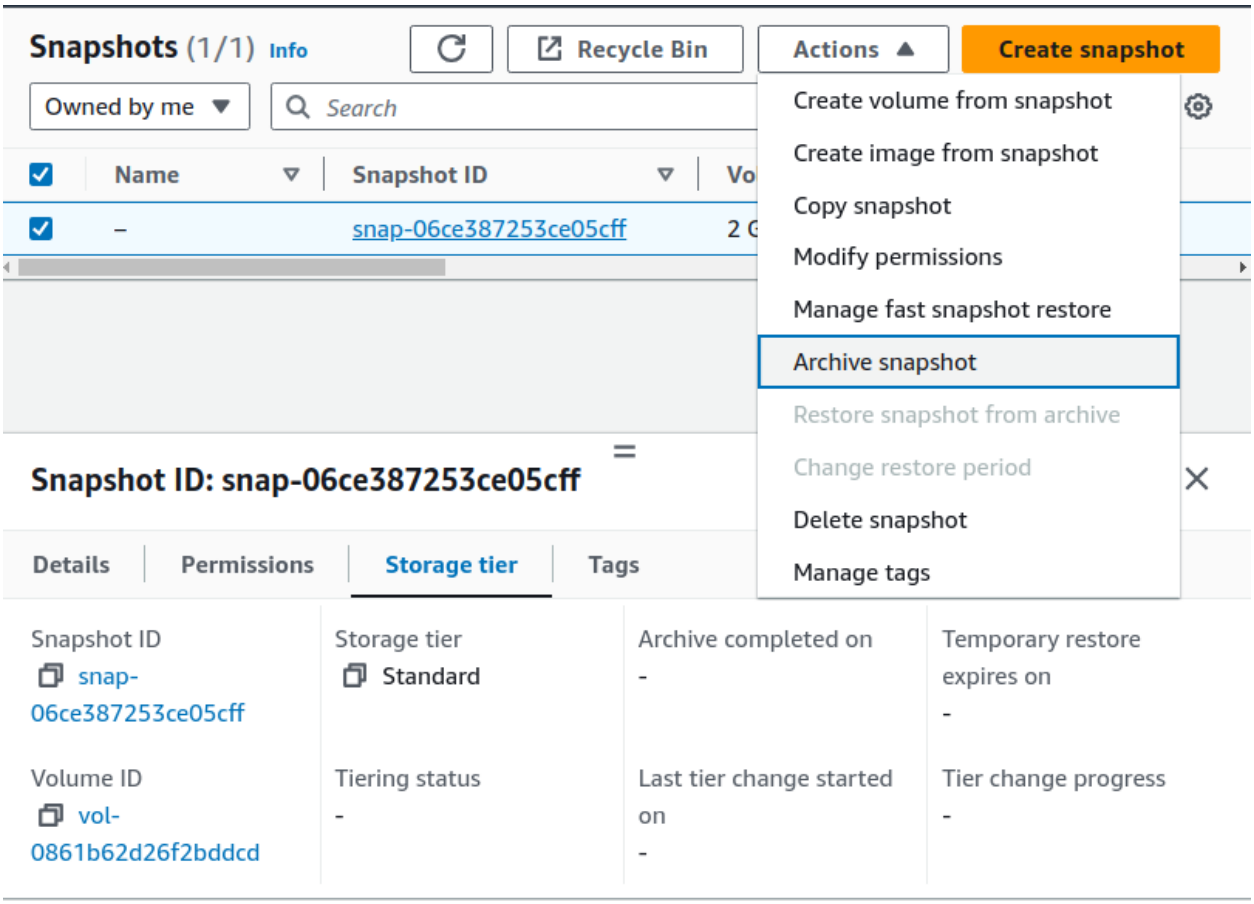


(Currently it is in the standard storage tier).

Go to recycle bin then resources where you will find your deleted ebs snapshot and you can recover it.



- Off topic but we can also archive a snapshot.



EC2 AMI (Amazon Machine Image)

An AMI is a pre-configured virtual machine image used for creating EC2 instances in Amazon Web Services (AWS). It is essentially a snapshot of an entire operating system, including the root file system, application server, and any installed applications. AMIs are used in traditional virtual machine (VM) scenarios, where each instance is a separate virtual server with its own operating system.

- An AMI is essentially a template for a virtual machine (VM). It contains the necessary information to launch an instance

- An AMI is a pre-configured virtual machine image (sort of an architecture, overview, layout) which includes an operating system, application server, and applications.
- AWS provides a variety of pre-configured, publicly available AMIs that cover a range of operating systems and software (ex: amazon linux 2 AMI which we used to create our demo instance). You can also create private AMIs for your organization's specific needs.

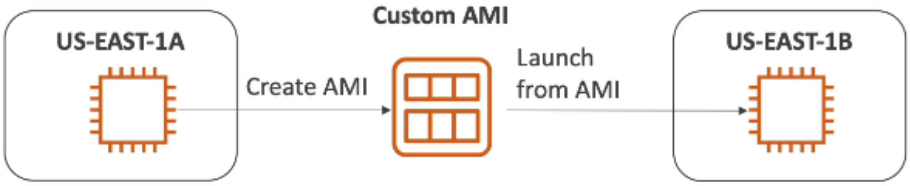
- AMI are built for a **specific region** (and can be copied across regions)
- You can launch EC2 instances from:
  - A Public AMI: AWS provided
  - Your own AMI: you make and maintain them yourself
  - An AWS Marketplace AMI: an AMI someone else made (and potentially sells)

- A customized AMI is especially useful if you want to make copies of your existing ec2 instance. **AMIs are for ec2 instances what snapshots are to ebs.**
- Whenever you build an AMI, it subsequently creates snapshots of the ebs attached to that instance which allows the whole instance to be copied.

- To create a copy of an instance:

AMI Process (from an EC2 instance)

- Start an EC2 instance and customize it
- Stop the instance (for data integrity)
- Build an AMI – this will also create EBS snapshots
- Launch instances from other AMIs

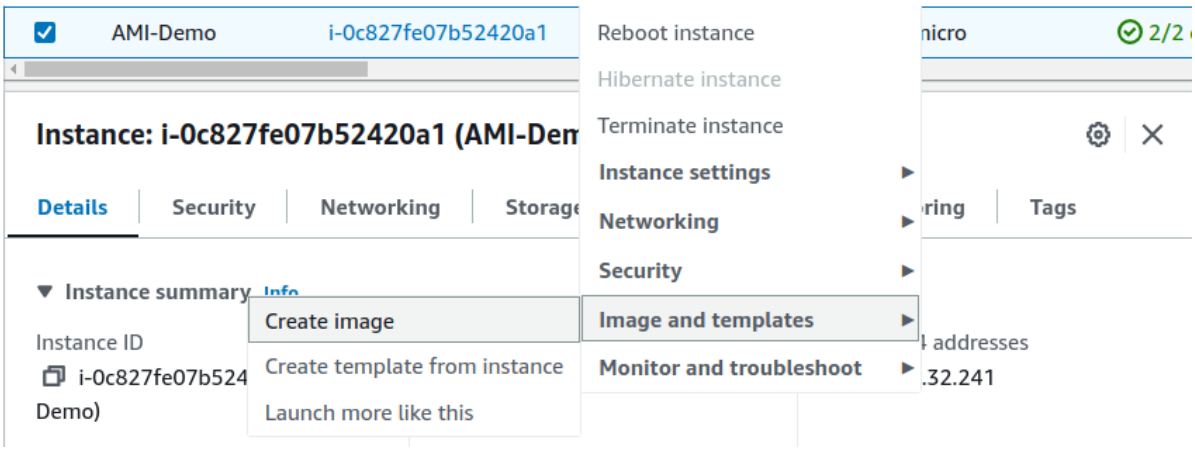


AMI Hands-on

- Launch a new instance following the same steps as before [amazon linux ami 2, key pair = ccp demo, select existing security group = launch-wizard-7], except the user data script. Modify the user data script to:

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
```

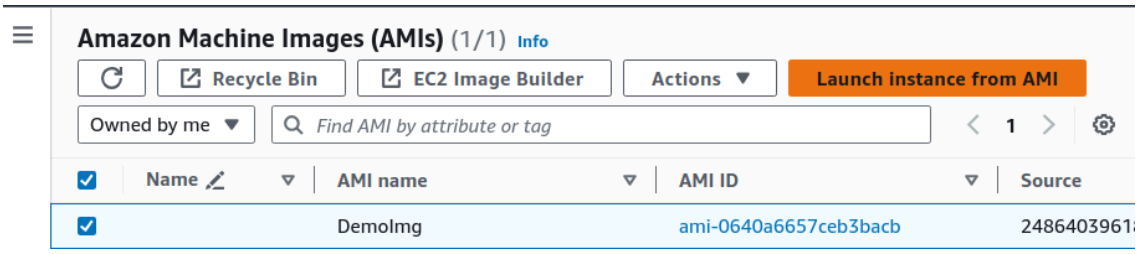
- Once the instance is launched, we will create an AMI.



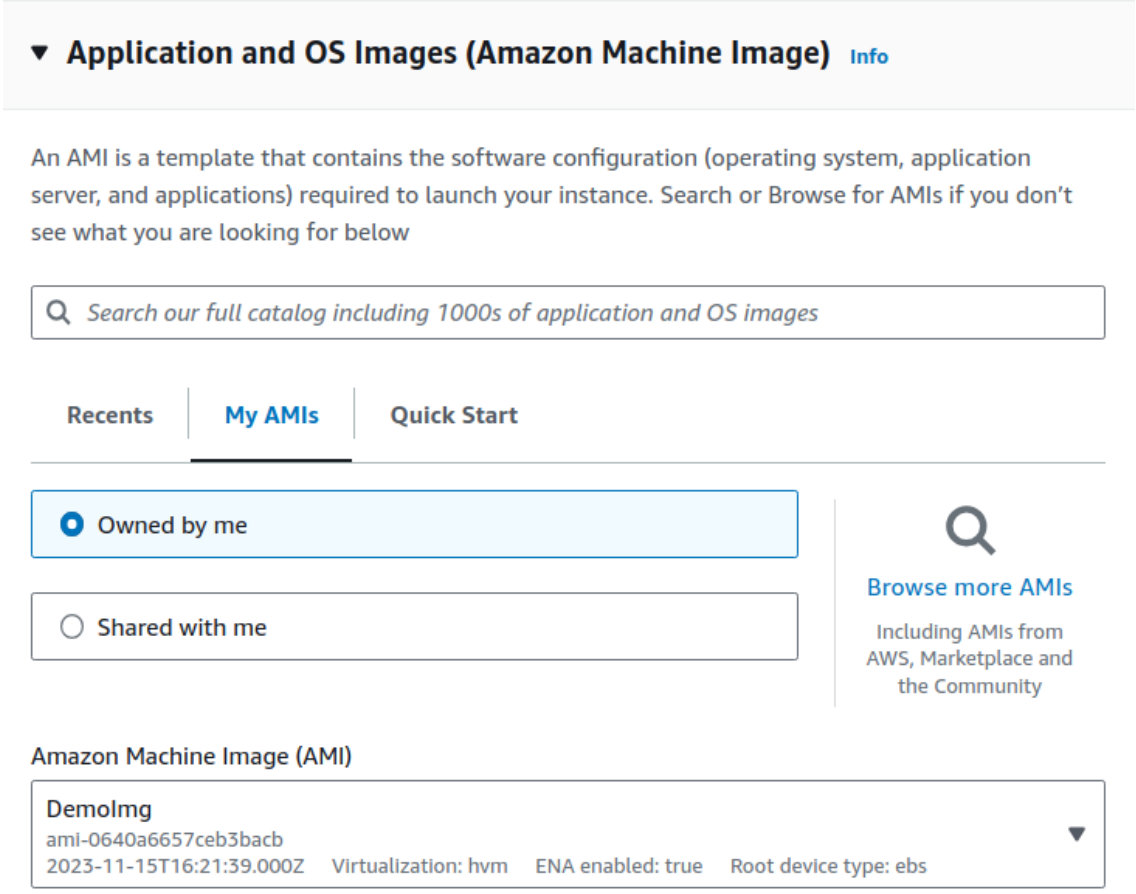
- After creating an image, go to AMIs and wait for your AMI to go from pending state to available state.
- Now, using this AMI you will be able to create copies of the AMI-Demo EC2 instance.

  You have two options to do so:

  1. Directly through the AMI dashboard



  2. While launching a new EC2 instance, you can choose from the `My AMIs` option.



- **NOTE:**

  Creating an AMI can help speed up the boot time. It does so by having pre-configured packages we require in the new instance.

  For example, now that we are creating this new instance using the blueprint (AMI) of our previous instance, we don't need to install httpd all over again. So, we can skip the code for that and get to the point directly (purpose of the application / instance).

  Hence, in this new instance, we will only write this code:

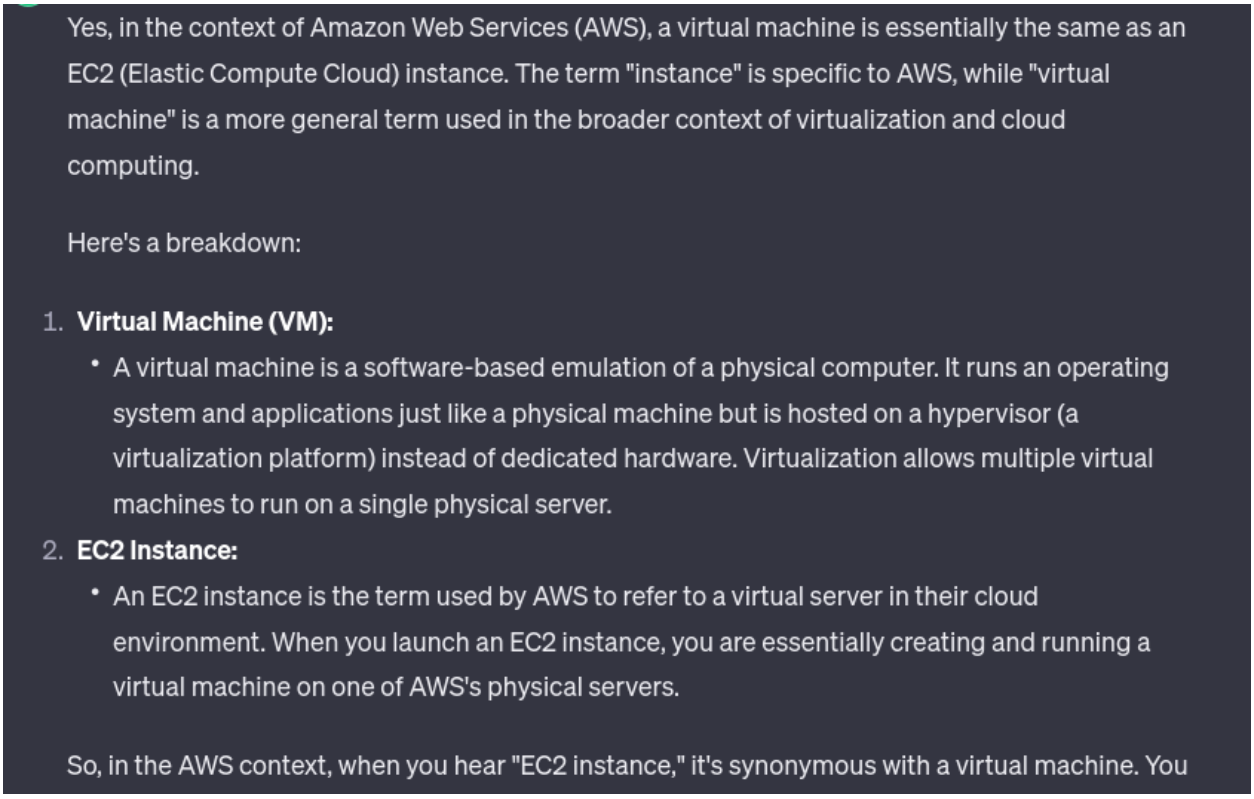
```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
```

```
echo "<h1>Hello world from ${hostname -f}</h1>"> /var/www/
html/index.html
```

This, above, is a use case on a minute scale. Imagine the dec in boot time if you consider pre-installed softwares or applications or scripts.

EC2 Image Builder

- It is a service used to automate the creation of virtual machines or container images. This essentially means that you will be able to automate the creation, maintenance, validation and testing of AMIs for EC2 instances.



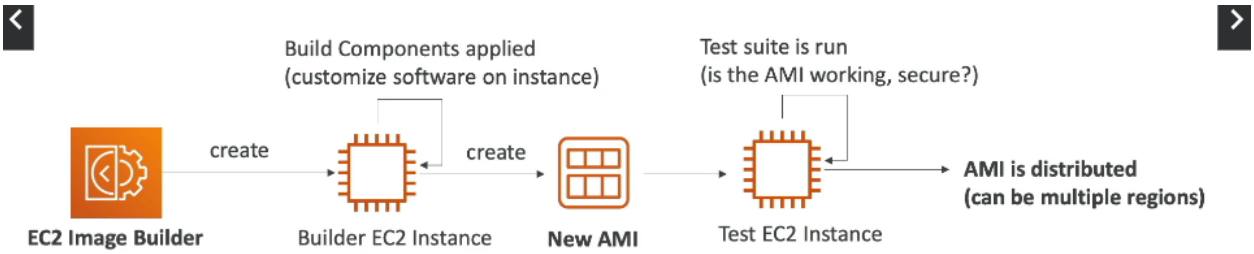
Yes, in the context of Amazon Web Services (AWS), a virtual machine is essentially the same as an EC2 (Elastic Compute Cloud) instance. The term "instance" is specific to AWS, while "virtual machine" is a more general term used in the broader context of virtualization and cloud computing.

Here's a breakdown:

1. **Virtual Machine (VM):**
   - A virtual machine is a software-based emulation of a physical computer. It runs an operating system and applications just like a physical machine but is hosted on a hypervisor (a virtualization platform) instead of dedicated hardware. Virtualization allows multiple virtual machines to run on a single physical server.
2. **EC2 Instance:**
   - An EC2 instance is the term used by AWS to refer to a virtual server in their cloud environment. When you launch an EC2 instance, you are essentially creating and running a virtual machine on one of AWS's physical servers.

So, in the AWS context, when you hear "EC2 instance," it's synonymous with a virtual machine. You

- **How does this service work?**
  - **Note:** all of these steps are automated.



1. We set up the service.
2. When it is run, it automatically creates a builder ec2 instance.
3. Now this builder instance is going to build the defined (by you) components (for example, install java or update the cli or update software or install firewalls, etc.)
4. After that, an AMI is created out of the builder instance.
5. Now, if you want to validate this AMI, ec2 image builder will automatically create a test ec2 instance which will run a bunch of tests (defined by you in advance).
6. Once the tests pass, the AMI will be distributed.
    - **Note:** EC2 image builder in itself is a regional service, it is still possible for you to take the AMI and distribute it across multiple regions, which allows you to go global.

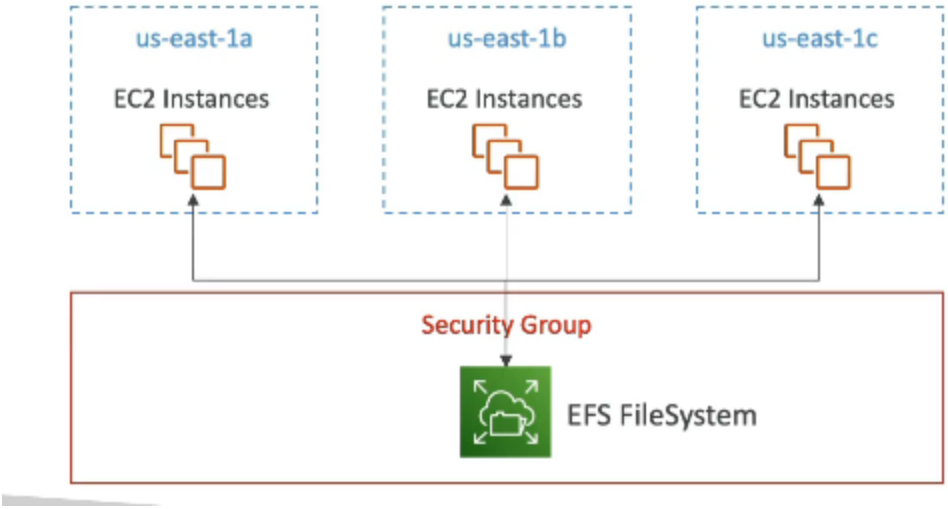
- Another feature is, EC2 image builder can be run on a schedule. For example, you can set it to run weekly or monthly or even when a condition is met like whenever packages are updated, etc.
- **It is a free service and you will only be paying for the underlying resources** created during this whole process (the instances and amis created and the storage required and stuff).

EC2 Instance Store (hardware)

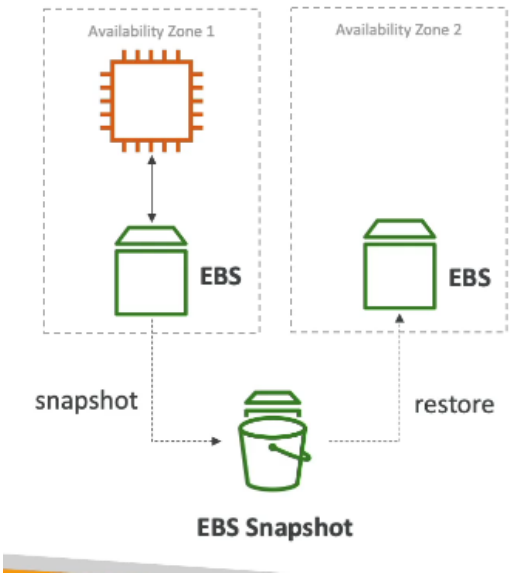
- If you need a high-performance hardware disk, you can use ec2 instance storage.
- Now, ec2 instances are virtual machines but they are obviously connected to a physical server and often times those physical servers have hard disk storage on them. If not, you can physically connect one to it. This is basically named as the ec2 instance store.
- Compared to EBS network drives which perform good but have limitations like latency, ec2 instance store offers better I/O performance.
- One of the cons is that ec2 instance store lose their storage if they are stopped. (ephemeral) which makes it ideal for short term use cases like buffer / cache / temporary content.
- Risk of data loss if hardware fails. So you decide to use this, it is your responsibility to backup and ensure fault tolerance.

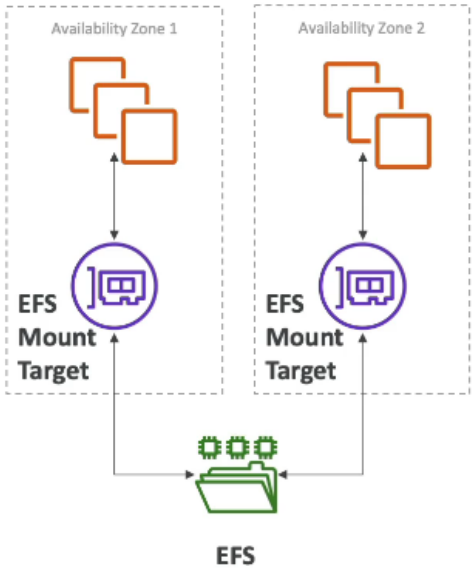
EFS - Elastic File System

- EFS is basically a managed NFS (network file system).
- A Network File System (NFS) is a distributed file system protocol that allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed. NFS enables file sharing and access among multiple computers over a network, creating a shared file system environment.
- EFS can be mounted to hundreds of ec2 instances at a time. This is a major advantage compared to EBS.
- EFS works only with linux ec2 instances and it works across multiple AZs within a region.
- It is highly available, scalable, expensive but you pay per use and don’t have to plan for capacity as in EBS.



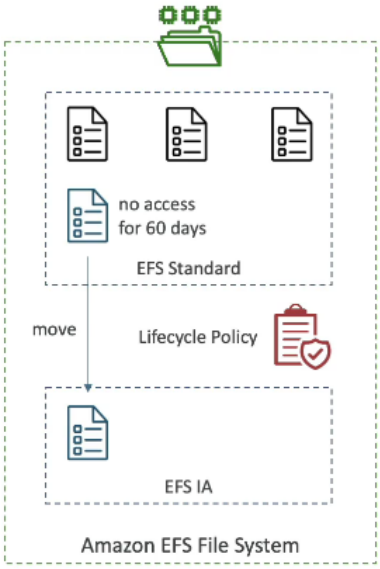
EBS vs EFS





EFS Infrequent Access storage class (EFS-IA)

- Storage class that is cost-optimized for files not accessed every day
- Up to 92% lower cost compared to EFS Standard
- EFS will automatically move your files to EFS-IA based on the last time they were accessed
- Enable EFS-IA with a Lifecycle Policy
- Example: move files that are not accessed for 60 days to EFS-IA
- Transparent to the applications accessing EFS



Amazon FSx (3rd party)

** refer to section 6 FSx overview video.

- A managed service to get third-party high-performance file systems on AWS.
- It is a fully managed service.
- You have mainly 3 varieties
  - FSx for lustre - storage for HPC (high performance computing)
  - FSx for windows file server - storage for windows server instances
  - FSx for NetApp ONTAP

Shared Responsibility Model for EC2 Storage

- Infrastructure
- Replication for data for EBS volumes & EFS drives
- Replacing faulty hardware
- Ensuring their employees cannot access your data

- Setting up backup / snapshot procedures
- Setting up data encryption
- Responsibility of any data on the drives
- Understanding the risk of using EC2 Instance Store

SUMMARY

- EBS volumes:
  - network drives attached to one EC2 instance at a time
  - Mapped to an Availability Zones
  - Can use EBS Snapshots for backups / transferring EBS volumes across AZ
- AMI: create ready-to-use EC2 instances with our customizations
- EC2 Image Builder: automatically build, test and distribute AMIs
- EC2 Instance Store:
  - High performance hardware disk attached to our EC2 instance
  - Lost if our instance is stopped / terminated
- EFS: network file system, can be attached to 100s of instances in a region
- EFS-IA: cost-optimized storage class for infrequent accessed files
- FSx for Windows: Network File System for Windows servers
- FSx for Lustre: High Performance Computing Linux file system

- **Note: EBS is restricted to a particular AZ while EFS is regional (can target multiple AZs within a region).**