

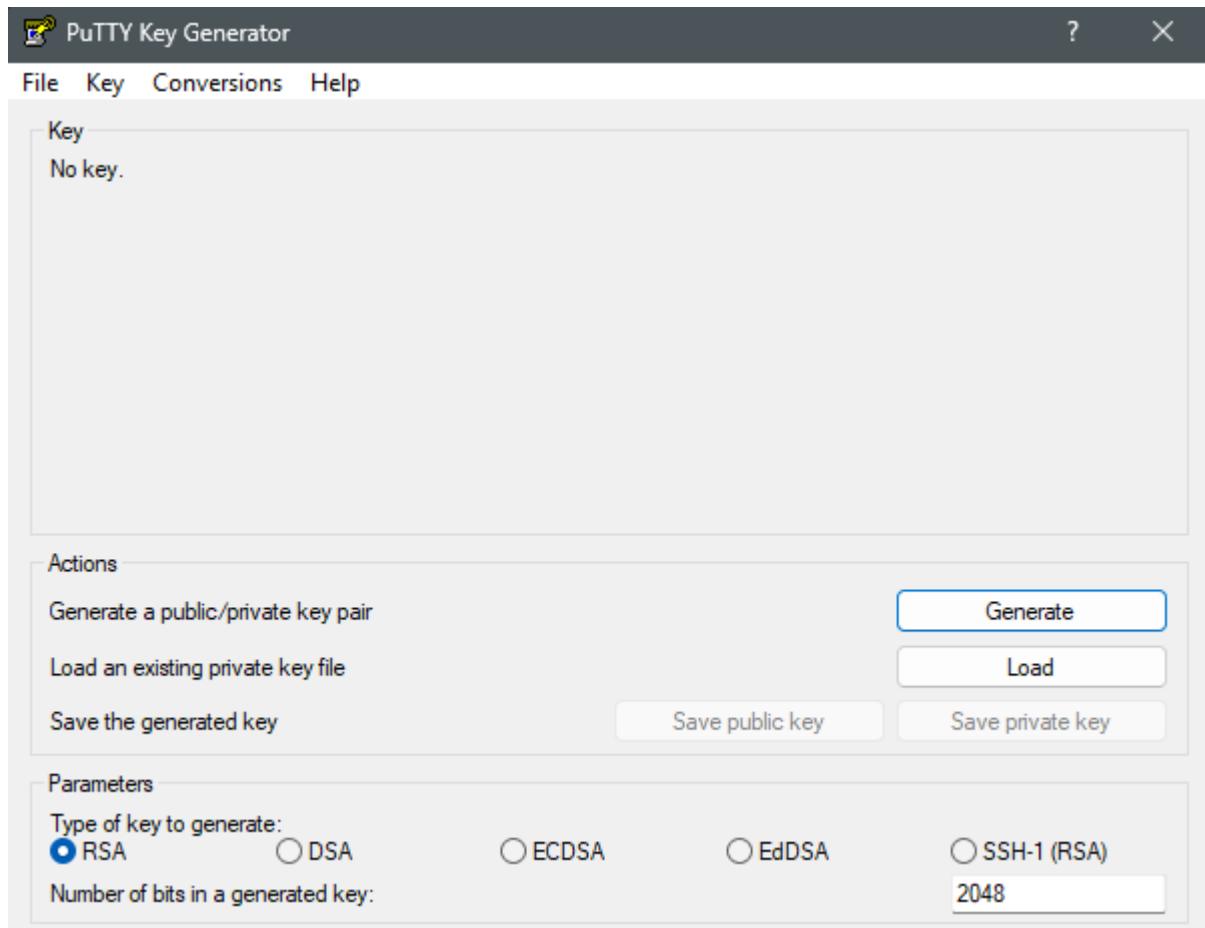
Task AWS restart/program

Name: Nandagopal K

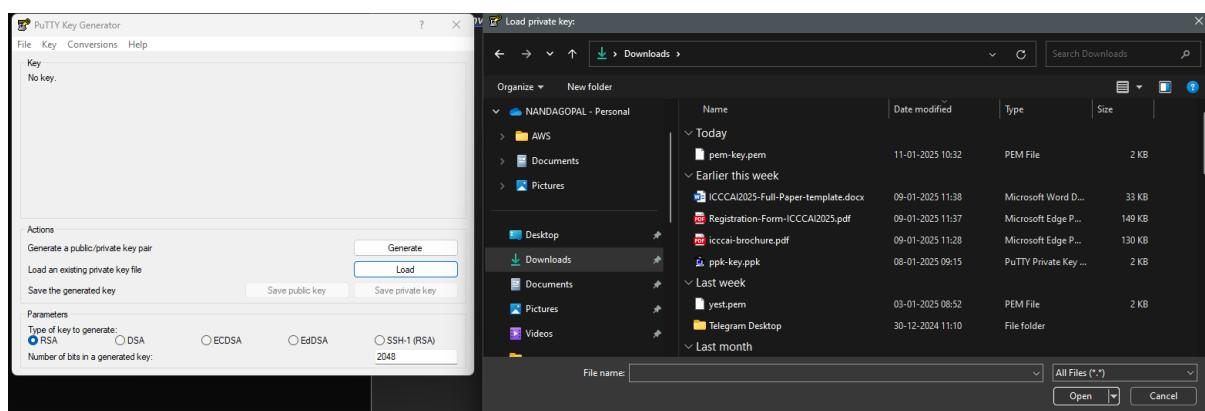
Task 1.Convert .pem keypair to .ppk

Step 1: Download Putty and puTTYgen

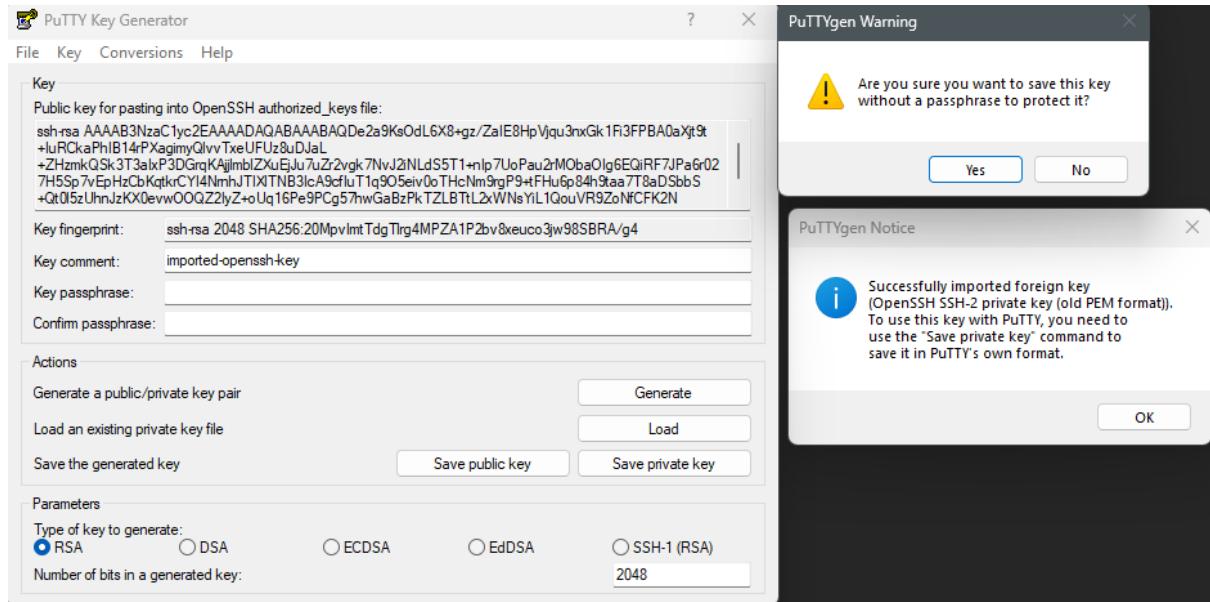
Step 2: Open the putTYgen and navigate to Load button



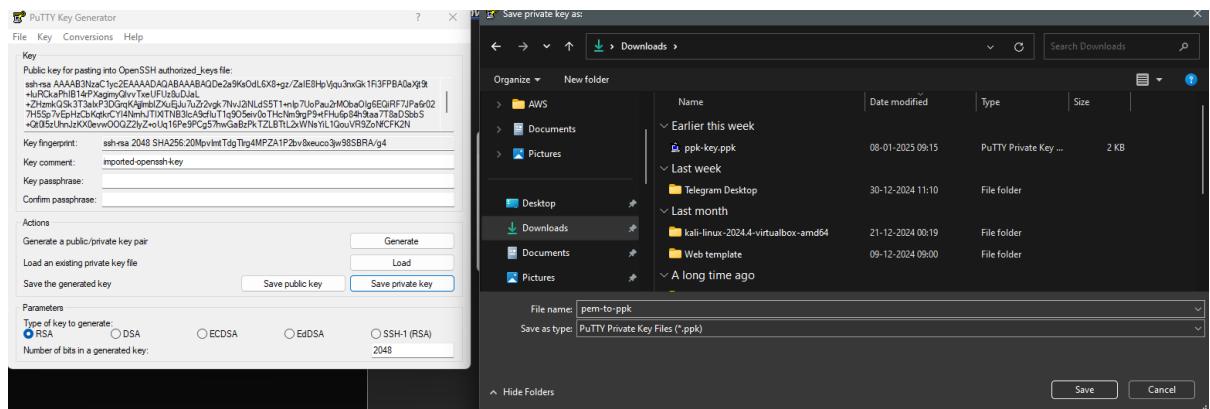
Step3: Select the pem file which you had already created



Step 4: once the .pem file is loaded a successful pop up message will be given and that shows to save the pem private file as putty private file.

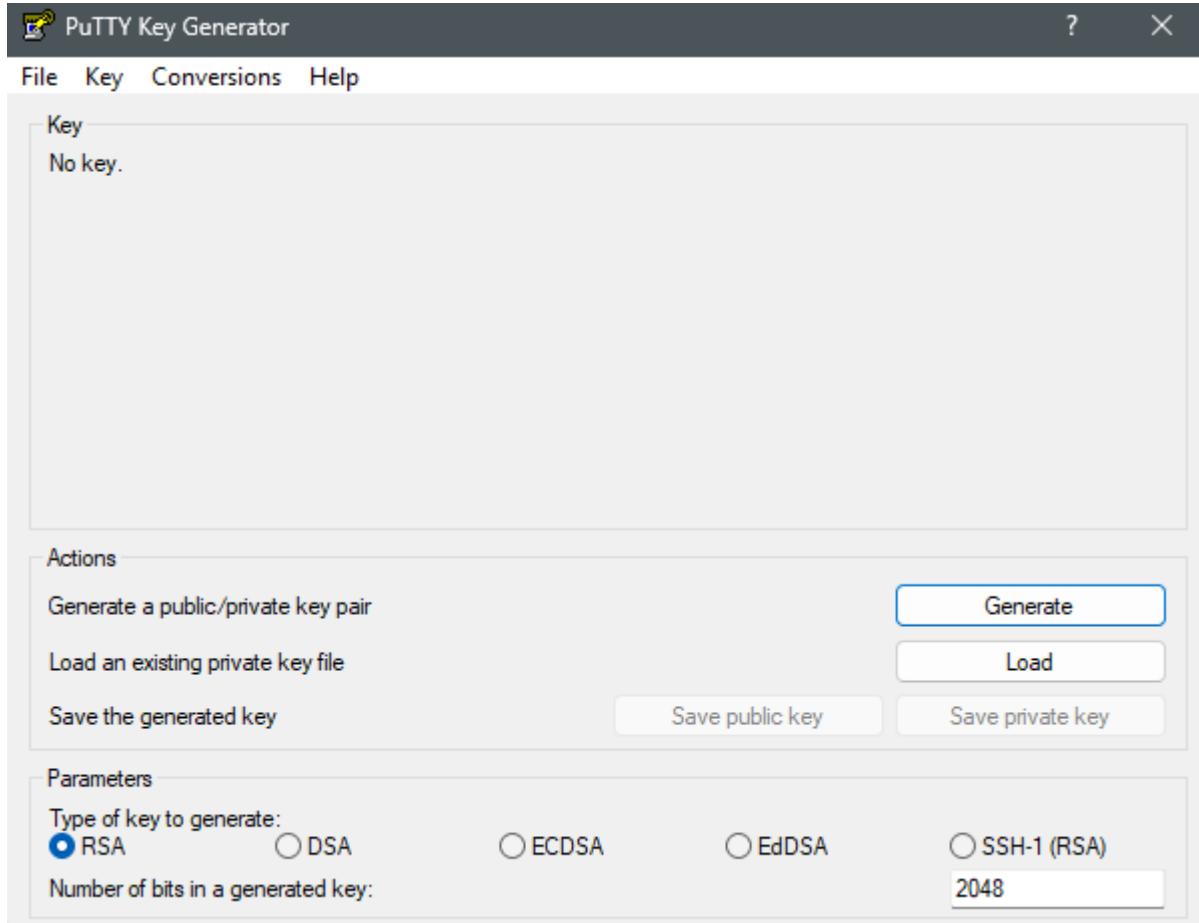


Step 5: Give the naming conversion for the new converted file and then you had successfully converted a pem file into a ppk file.

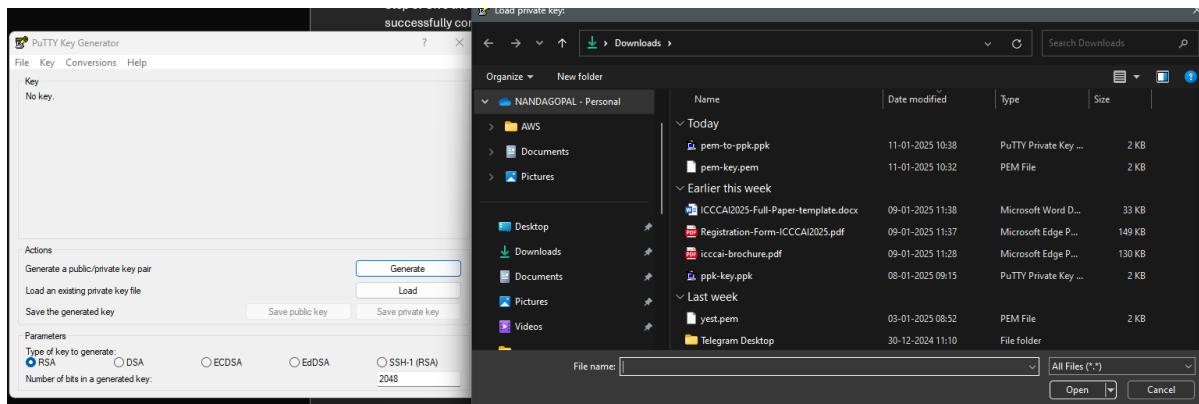


Task 2 : Convert .ppk file to .pem file

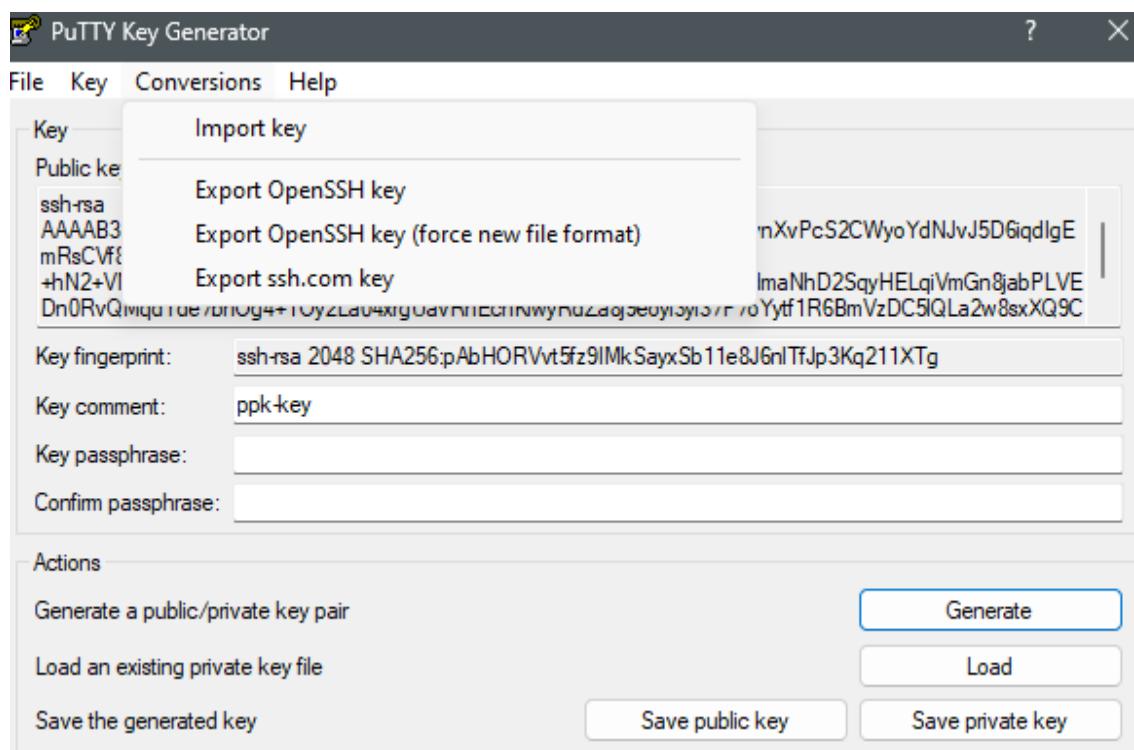
Step1: Install puTTYgen and navigate into the software



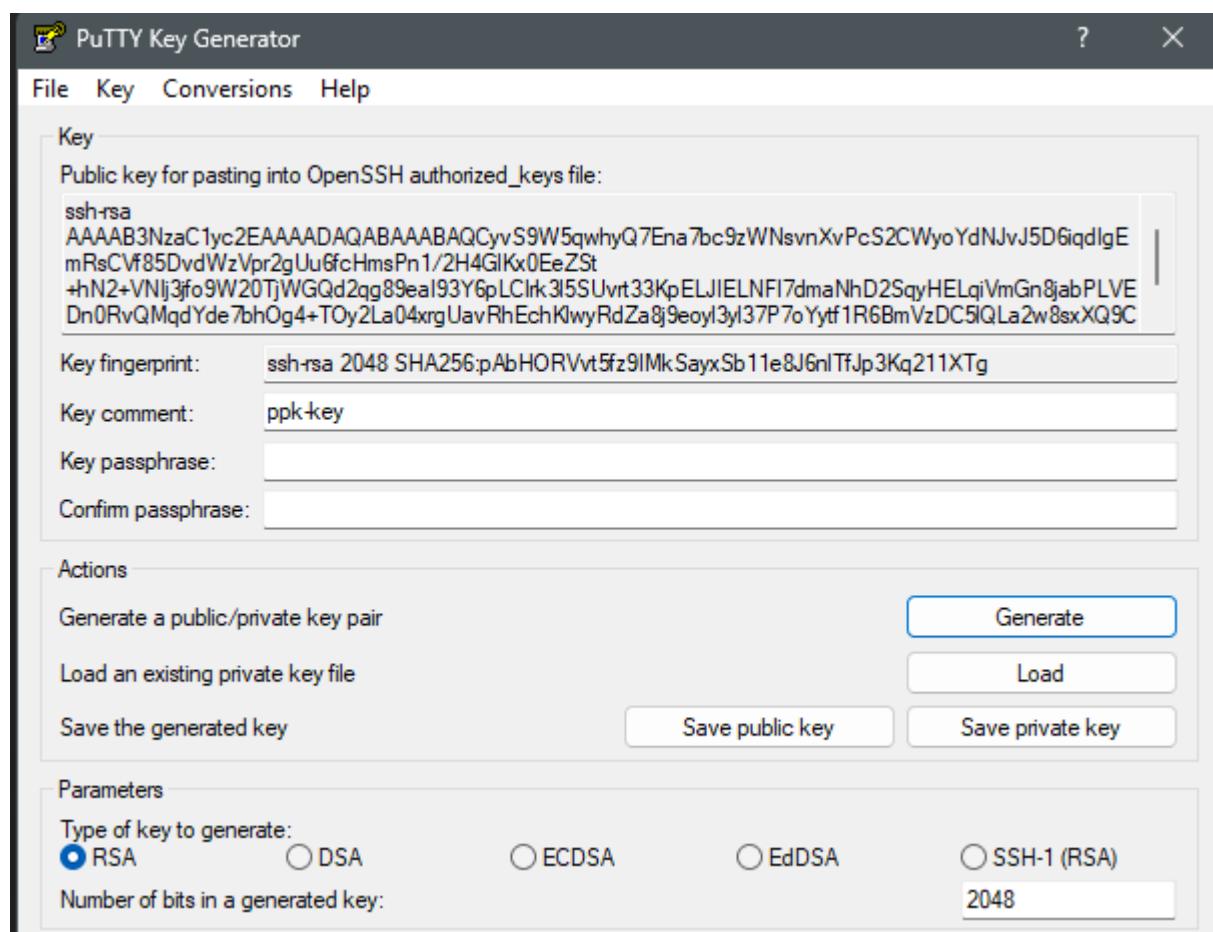
Step 2: Click on conversions and there will be a load option and click on load



Step 3 : one the .ppk file is loaded if you need a key passpharase you can apply it or else leave it as blank it is nothing but an password for creating a new .pem file



Step 4: Go to conversions and click Export OpenSSH Key and then export it as .pem file

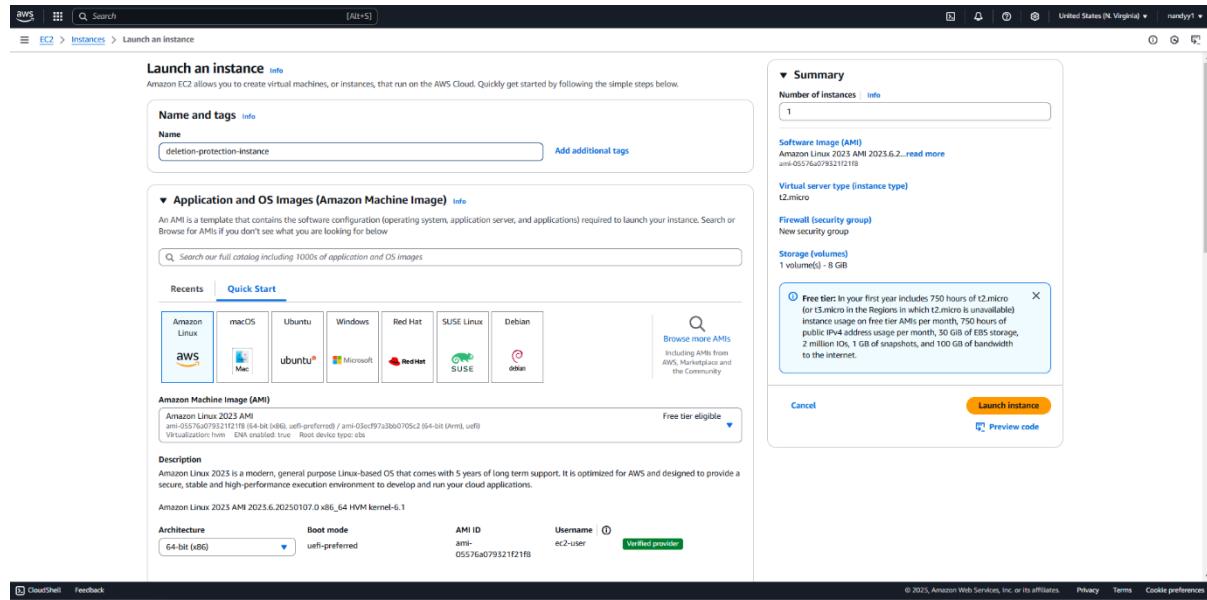


Step 5: The output of the .ppk to pem resembles like this and it is important to note that the file has begin with *RSA PRIVATE KEY* and *END RSA PRIVATE KEY*

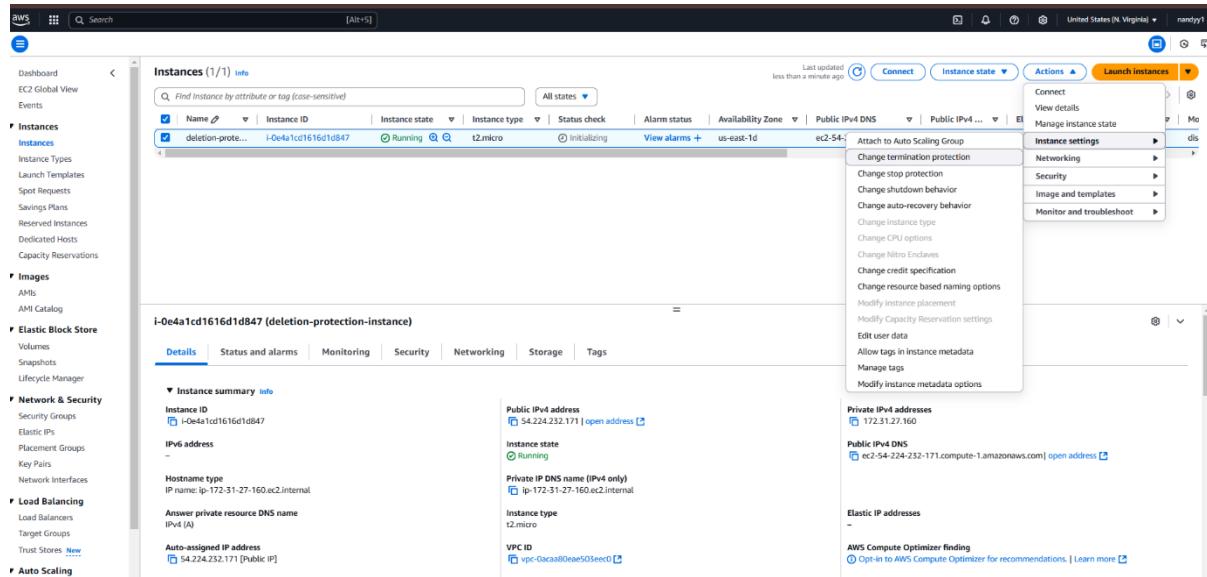
```
|---- BEGIN RSA PRIVATE KEY----  
MIIEowIBAAKCAQEAsr0vVuasIckOxJ2u23Pc1jbL517z3EtglsgHTSbyeQ+oqns  
IBJkbA1X/OQ73Vs1aa9oFLun3B5rD59f9h+BpSsdBHmUrfoTdv1TSI9436PVttE4  
1hkHdqoPPXmiPd20qSwiK5N5eU1L67d9yqRCySBCzRSO3ZmjYQ9kqshxC6o1Zh/  
I2mzy1RA59Eb0DKnWHXu24ToOPkzsti2tOMa4FGr0YRHISpcMkXWVvI/XqMpd8pd  
+z+6GMrX9UegZlcwwuZUC2tsPLMV0PQqRFkgbQwpJBYL+7xmPlHSNXAoq+1QNGnX  
QNmplj9YvHNx1Niv5BHvudTYpw3FpK0BN8StNwIDAQABAOIBAHueyzIQYncPhu5H  
nNtdHBzwU0CX9J01co14Tx+99g3BleyNBa526guqvrdC54Cmh5+r2JGWZHeidiuc  
8BLbIaCp2+zvZfyahIJJ20czi8cpmjNSKzwwnW/eGKJNEQA2z2eEBbIb0oUnQ6Ip  
C5Bu4V4XuPv8Pv9XNp93tQ1FjqIgh8frFSzEy0GhZmg9mkWGNSfyHwy8VayQjpRO  
fephltooXziRpZjcV5qpo521w2n0xFU8ToJmB2KD/edxJirk4YPJZDe8j5nFZLMh  
7neGoUMNQ8U5a4I2sHfZG6M1mdVuprElPwzxOw0WqIWf+YcRAv6NPfSY185PjSOK  
U+SjXwECgYEAE6HhIB8AmNJ3uNHqOz1i3IqnCZDANC81CxKrmaDIIKcv3vJsrki60  
K/cLAT+pEW8hc8X81UjiAdMzVtxmnxFaI2eMgKuV7vTA89KK6I2Xy7egQ3kyI6  
ykZjhntw70DBWe27tsLzUcedDJ8M/Aiu9q751Toibh5h1jD2Gw4ircCgYEAxNSj  
n2GJHK0tby5y/0hTboITfMKMfYfPiZ8zfLveDtnJmf4D/90QXKyb5oLjSTGUb7ZI  
3QIQw3Ij0SoNwVqf9chRgLD7zCtoDKZ4T8tFGS/tP+90YPYYNf5ZZEqQgLqzItec  
YJ9R+xDCzUHEO/XJyFMYe8xGMQ16T07Co3pUcYECgYEAYQ907TUpVhTJ8yWU4oy0  
SDKe6FwOzITTdJmDEddD7DJ3H+7s7k/JswjFwdy3c1a0jhdm8CjLIFz0aEnTj/1K  
2TSy//oFo+fVBuk0Ej1yKviQhWwkU2M1idD7P6S42awmw8QCGe/l8zNxiXoophsp  
T2RNY1dIj+S3ZpTbPhHLjV0CgYAFjNlwTRfIrzlwwSaM/EUeAJz8D6BWidXYU5zo  
D4QbdxiwNvF17Y3gxj12SoqOLc+5trInST+fw0AvTbhFk4CfNZ7WuibfgvnUIX2T  
AnBYXXmbK+1kLoxQOXPb2B5rAti6SsomD1gIdFiJ4SjfZ/bgpzcxhnkU0OgyBE6I  
vZJUAQKBgGUDrhaOmF3Rnk4dUPQaf0x9okMPNfqQiFR5qnfp9Ay5SkoX7FiAoAXE  
5bRHkgQhno4pAKLd9gnqrFhGIyoy1g6rmQS7dnwVx8NIFMo8+6/1iwwhKjLqHmK7  
JHO6fWSyq/luMICzaf6Oc6WujB5k4D1d9vSgV9rb0n0qcYyMwZH  
----- END RSA PRIVATE KEY-----
```

Task 3: Enable Deletion Protection for instances

Step 1: Create an instance with a name , ami, keypair, volume, security group, and leave all the other settings as default and click the launch button.



Step 2: click on Actions and instance settings and in the dropdown menu click change termination protection



Step 3: Check the Enable button this buttons shows that if any accidental deletion occurs on the instance means the instance will not get deleted.

The screenshot shows the AWS EC2 Instances page with a single instance listed. A modal dialog box is open over the instance details, titled "Change termination (deletion) protection". Inside the dialog, there is a checkbox labeled "Enable" which is checked. At the bottom right of the dialog, there is a yellow "Save" button. The main EC2 page shows the instance status as "Running" and its type as "t2.micro".

Step 4: If you try to terminate the instance the instance won't get terminated because in the previous step we have enabled the prevent instance from being accidentally deleted.

The screenshot shows the AWS EC2 Instances page with a red error message at the top: "Failed to terminate (delete) an instance: The instance i-0e4a1cd1616d1d847 may not be terminated. Modify its DisableApiTermination instance attribute and try again." Below the error message, the instance details are shown, including its ID, state (Running), and type (t2.micro). The instance summary section also displays its public and private IP addresses and DNS names.

This makes our instance to prevent from accidentally deletion process

Step 5: To delete the instance navigate to the same setting where you enabled the button go and disable it for delete the instance.

Step 6: Once it is disabled and save this settings now go to the instances and click on instance state and click terminate instance this will now get terminated.

The output is to enable accidental deletion instance and terminate the insatance

Task 4 : Login the instance created by customized AMI

Step-1: go to the aws instance settings and create an instance for custom AMI.

Click Launch instance

After launching the instance will be available on the instances dashboard

Step 2: Open Putty and load the PPK file and the CLI will open

The screenshot shows the Putty Configuration dialog box and the AWS Lambda console interface.

Putty Configuration Dialog:

- Category:** SSH > Auth
- Credentials to authenticate with:** Public-key authentication
- Private key file for authentication:** C:\Users\Nanda\Downloads\ppk-key.ppk (Browse...)
- Certificate to use with the private key (optional):** (Browse...)
- Plugin to provide authentication responses:** (Plugin command to run)

AWS Lambda Instance Details:

- Instance:** ip-172-31-87-179
- Public IPv4 address:** 34.201.67.40 | [open address](#)
- Instance state:** Running

CLI Session Output:

```
root@ip-172-31-87-179:/home/ec2-user
└─ login as: ec2-user
└─ Authenticating with public key "ppk-key"
   ,#
   ~\###~      Amazon Linux 2023
   ~~\#####\
   ~~ \###|
   ~~  \#/   https://aws.amazon.com/linux/amazon-linux-2023
   ~~   V~'-'>
   ~~~
   ~~~.-
   ~~~/ \
   /m/
[ec2-user@ip-172-31-87-179 ~]$ sudo su
[root@ip-172-31-87-179 ec2-user]#
```

Step 3: Enter the commands to run the nginx server the nginx is the middleware server which is used to connect the physical system and the hardware.

This will be installed on the instances which we create.

Paste the Following commands

Sudo su

Yum update -y

Yum install nginx -y

systemctl start nginx

systemctl enable nginx

systemctl status nginx

Step-4: copy the instance IP and paste it on the web browser

```
Total download size: 1.1 M
Total installed size: 3.6 M
Downloaded: 1 file(s), 1.1 MB, 5.9 kB/s, 0:00:00
Preparing: 1/1
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing: 1/1
Running scriptlet: nginx-fs-1.11.26.2-1.amzn2023.0.1.noarch
Installing : nginx-fs-1.11.26.2-1.amzn2023.0.1.noarch
Unpacking: libunwind-1.4.0-5.amzn2023.0.1.x86_64.rpm
Installing : libunwind-1.4.0-5.amzn2023.0.1.x86_64
Unpacking: gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Installing : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Unpacking: gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Installing : generic-logs-nginx-1.11.26.2-1.amzn2023.5.3.noarch
Unpacking: generic-logs-nginx-1.11.26.2-1.amzn2023.5.3.noarch
Preparing: 1/1
Running scriptlet: nginx-1.11.26.2-1.amzn2023.0.1.x86_64
Installing : nginx-1.11.26.2-1.amzn2023.0.1.x86_64
Unpacking: libunwind-1.4.0-5.amzn2023.0.3.noarch
Verifying : libunwind-1.4.0-5.amzn2023.0.3.noarch
Verifying : libunwind-1.4.0-5.amzn2023.0.3.x86_64
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying : generic-logs-nginx-1.11.26.2-1.amzn2023.5.3.noarch
Verifying : generic-logs-nginx-1.11.26.2-1.amzn2023.5.3.noarch
Complete!
[root@ip-172-31-87-179 ec2-user]# systemctl start nginx
[root@ip-172-31-87-179 ec2-user]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[root@ip-172-31-87-179 ec2-user]# systemctl status nginx
* nginx.service - The nginx HTTP and reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
  Active: active (running) since Sat 2025-01-11 06:08:16 UTC; 248 ago
    Main PID: 23346 (nginx)
      Tasks: 2 (limit: 1111)
     Memory: 2.4M
        CPU: 0.000 CPU(s)
       CGroup: /system.slice/nginx.service
           ├─23346 *nginx master process /usr/sbin/nginx*
           └─23343 nginx worker process*
Jan 11 06:08:16 ip-172-31-87-179.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy ...
Jan 11 06:08:16 ip-172-31-87-179.ec2.internal nginx[23262]: nginx: the configuration file /etc/nginx/nginx.conf syntax is valid
Jan 11 06:08:16 ip-172-31-87-179.ec2.internal nginx[23264]: nginx: configuration file /etc/nginx/nginx.conf test is
success
Jan 11 06:08:16 ip-172-31-87-179.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy ...
lines 1-15/15 (END)
```

Step – 5 click the instance and click on actions -> image and templates -> create image for creating a custom ami.

The screenshot shows the AWS EC2 Instances page. In the center, there is a table of instances. The first instance, 'custom-ami-in...', has its context menu open. The 'Actions' dropdown is expanded, and the 'Image and templates' option is highlighted with a yellow background. Other options like 'Create image' and 'Create template from instance' are also visible in the menu.

Create a custom AMI by selection following settings

The screenshot shows the 'Create image' configuration dialog. It includes fields for 'Image name' (set to 'ami-for-instance'), 'Image description - optional' (left empty), and a 'Reboot instance' checkbox (unchecked). Below these are 'Instance volumes' settings for an EBS volume (size 8, IOPS 3000, Throughput 125, Delete on termination checked, Encrypted unchecked). A note states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' At the bottom, there are sections for 'Tags - optional' with two options: 'Tag image and snapshots together' (checked) and 'Tag image and snapshots separately' (unchecked).

The screenshot shows the 'Amazon Machine Images (AMIs)' page. A table lists the created AMI, 'ami-for-instance', with details: AMI ID 'ami-05cc9f7406d51162e8', Source '255490764458/ami-for-instance', Owner '255490764458', Visibility 'Private', Status 'Pending', Creation date '2025/01/11 11:40 GMT+5:30', and Platform 'Linux/UNIX'. The 'Launch instance from AMI' button is visible at the top right of the table header.

Step 6: Go to Ami and click Launch instance AMI and this is the Ami instance which you had created with the following configurations

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The steps are as follows:

- Name and tags:** The instance is named 'instance-ami'. There is a link to 'Add additional tags'.
- Application and OS Images (Amazon Machine Image):** A search bar is available to find AMIs. The selected AMI is 'ami-05cc0f7406d3162e0' (ami-for-instance).
- AMI from catalog:** Shows the AMI details: Name (ami-for-instance), Description (This is an ami created by me), Image ID (ami-05cc0f7406d3162e0), Username (root), Published (2025-01-11T06:10:13.000Z), Architecture (x86_64), Virtualization (hvm), Root device type (ebs), ENA Enabled (Yes).
- Browse more AMIs:** A link to view other AMIs from AWS, Marketplace, and the Community.
- Virtual server type (instance type):** The selected instance type is 't2.micro'.
- Summary:** Shows 'Number of instances' (1) and a summary of the instance configuration.
- Free tier:** Information about the free tier usage.
- Launch instance:** The primary button to start the instance.
- Preview code:** A link to preview the launch configuration.

Now the instance and the instance with ami is running

The screenshot shows the 'Instances (1/4)' page in the AWS EC2 console. The instance 'instance-ami' is listed with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Mo
custom-ami-...	i-05eefc113242edadd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-34-201-67-40.com...	34.201.67.40	-	-	dis
deletion-prote...	i-0e4a1cd1616d1d847	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-	-	-	dis
instace-from-...	i-03ae45727dabdc3ba	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-54-167-52-89.com...	54.167.52.89	-	-	dis
custom-ami	i-099473d4bfaf5a542e	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	dis

The instance 'instace-from-' is currently running. Below the table, the instance details for 'instace-from-' are shown:

- Details:** Shows the instance ID (i-03ae45727dabdc3ba), status (Running), and instance type (t2.micro).
- Instance summary:** Shows the public IP address (54.167.52.89), private IP address (172.31.85.120), and instance type (t2.micro).
- Networking:** Shows the VPC ID (vpc-0aca80eae503eed0), subnet ID (subnet-0a11111111111111), and elastic IP (ec2-54-167-52-89.compute-1.amazonaws.com).
- Security:** Shows the IAM role assigned to the instance.

Step -7: Output for Login the instance created by customized AMI

The screenshot shows the AWS Lambda function configuration page. At the top, there's a search bar and tabs for 'Recents', 'My AMIs' (which is selected), and 'Quick Start'. Below these are two filter buttons: 'Owned by me' (selected) and 'Shared with me'. To the right is a 'Browse more AMIs' section with a search icon and text: 'Including AMIs from AWS, Marketplace and the Community'. The main content area is titled 'Amazon Machine Image (AMI)' and shows a single item: 'ami-for-instance' with ID 'ami-05cc9f7406d3162e8'. Below it, details are listed: '2025-01-11T06:10:13.000Z', 'Virtualization: hvm', 'ENA enabled: true', 'Root device type: ebs', and 'Boot mode: uefi-preferred'. A 'Description' section follows, stating 'This is an ami created by me'. Under 'Architecture', 'x86_64' is listed, and under 'AMI ID', 'ami-05cc9f7406d3162e8' is shown. On the far right of the main content area is a small dropdown arrow.

Task-5 Transer volume to another region

First creating the snapshot and creating the volume for the different availability zone

Step-1 : Create an instance

The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', has a 'Name' field containing 'crate-ss-instance'. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a list of AMIs. It highlights 'Amazon Linux 2023 AMI' with ID 'ami-05576a079321f21f8'. The third step, 'Description', contains a detailed description of Amazon Linux 2023. The fourth step, 'Architecture', shows '64-bit (x86)' selected. The fifth step, 'Boot mode', shows 'uefi-preferred'. The sixth step, 'AMI ID', shows 'ami-05576a079321f21f8'. The seventh step, 'Username', shows 'ec2-user'. The eighth step, 'Storage (volumes)', is partially visible on the right. At the bottom right are 'Cancel', 'Launch instance', and 'Preview code' buttons. A tooltip for the storage step provides information about free tier usage.

The screenshot shows the AWS EC2 Volumes page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays a table titled 'Volumes (1) Info' with one row. The columns are: Name (vol-01a993ca3f08764b7), Volume ID (vol-01a993ca3f08764b7), Type (gp3), Size (8 GiB), IOPS (5000), Throughput (125), Snapshot ID (snap-055e2a2...), Created (2025/01/11 12:28 GMT+5:...), Availability Zone (us-east-1c), Volume state (In-use), Alarm status (No alarms), and Attached resource (i-08b7e4a1d1c40). Action buttons for Actions, Create volume, and a copy icon are at the top right.

Step-2 : Go to volume and right click the volume and from the volume create a snapshot

This screenshot is similar to the previous one, showing the EC2 Volumes page. However, the volume 'vol-01a993ca3f08764b7' is selected, and a context menu is open over it. The menu options include: Create volume, Modify volume, Create snapshot, Create snapshot lifecycle policy, Delete volume, Attach volume, Detach volume, Force detach volume, Manage auto-enabled I/O, Manage tags, and Fault injection. The main content area shows the same volume details table as before.

Snapshot is created

The screenshot shows the AWS EC2 Snapshots page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays a table titled 'Snapshots (1/1) Info' with one row. The columns are: Name (snap-0128ccbeffedca5), Snapshot ID (snap-0128ccbeffedca5), Volume size (8 GiB), Description (this is my snapshot from N...), Storage tier (Standard), Snapshot status (Pending), Started (2025/01/11 12:30 GMT+5:...), Progress (0%), Encryption (Not encrypted), KMS key ID (None), and KMS key al... (None). Action buttons for Recycle Bin, Actions, Create snapshot, and a copy icon are at the top right.

Step-3 : Click the snapshot and create the volume from the snapshot

Snapshots (1/1) info

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	KMS key ID	KMS key al...
<input checked="" type="checkbox"/> snap-0128ccbeffe8ce8a5	8 GiB	this is my snapshot from N...	Standard		Create snapshot	2025/01/11 12:30 GMT+5:30	100%	Not encrypted	-	-

Actions: Recycle Bin, Create snapshot, < 1 >, Snapshot settings, Archiving, Recycle bin.

Snapshot ID: snap-0128ccbeffe8ce8a5

Details	Snapshot settings	Storage tier	Tags
Snapshot ID <input checked="" type="checkbox"/> snap-0128ccbeffe8ce8a5	Progress 100%	Snapshot status Completed	Owner <input checked="" type="checkbox"/> Z549076458
Started <input checked="" type="checkbox"/> Sat Jan 11 2025 12:30:18 GMT+0530 (India Standard Time)	Product codes -	Fast snapshot restore -	Description this is my snapshot from N.virginia from an instance volume
Source volume <input checked="" type="checkbox"/> Volume ID vol-01a995ca908764b7	Volume size <input checked="" type="checkbox"/> 8 GiB		
Encryption <input checked="" type="checkbox"/> Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -

This volume can be encrypted and select the different available zone from the SAME region

Create volume info

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Snapshot ID
 snap-0128ccbeffe8ce8a5

Volume type: General Purpose SSD (gp3)

Size (GiB): 8

IOPS: 3000

Throughput (MiB/s): 125

Availability Zone: us-east-1a

Fast snapshot restore: Not enabled for selected snapshot

Encryption: Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.
 Encrypt this volume

Tags - optional: A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.

The first volume was us-east-1c and the volume created from the snapshot is us-east-1a

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created	Availability Zone	Volume state	Alarm status	Attached resource
-	vol-01a993ca3f08764b7	gp3	8 GiB	3000	125	snap-055e2a2...	2025/01/11 12:28 GMT+5...	us-east-1c	In-use	No alarms	i-0887e4a1d1c40
diff-az-volume	vol-0d9f9185361e4b283	gp3	8 GiB	3000	125	snap-0128ccb...	2025/01/11 12:35 GMT+5...	us-east-1a	Available	No alarms	-

Step-4 : right click the snapshot -> copy -snapshot

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	KMS key ID	KMS key alias
-	snap-0128ccbfe8ce8a5	8 GiB	this is my snapshot from N...	Standard	Completed	2025/01/11 12:30 GMT+5...	100%	Not encrypted	-	-

Copy snapshot to different region

Step-5: We can also proceed with default settings and can also modify the setting for copying the snapshot for another region

- ➔ Encrypted
- ➔ Destination Region
- ➔ Tags

Then the new snapshot copy will be created for the new AZ's

Copy snapshot Info

Copy a snapshot from one AWS Region to another, or within the same Region.

Source snapshot
The original snapshot that is to be copied.

Snapshot ID

Region

Snapshot copy details

Description
A description for the snapshot copy.
 255 characters maximum.

Destination Region
The Region in which to create the snapshot copy.

Time-based copy - new Info
Specify a completion duration for the snapshot copy operation. Additional costs apply. [Learn more](#) Enable time-based copy

Encryption Info
Use Amazon EBS encryption as an encryption solution for your EBS resources.
 Encrypt this snapshot
KMS key Info (default)

KMS key description

KMS key owner account

KMS key ID

Step-6 : This is the Singapore region snapshot WHICH IS BIENG created

Snapshots (1) Info

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	KMS key ID	KMS key al...
snap-0442aacabb8b16c64	8 GiB	Copied snap-0128ccbe...	Standard	Completed	2025/01/11 12:43 GMT+5...	100%	Encrypted	ad533f9e-2af...	aws/ebs	

Step-7: create a volume for the snapshot

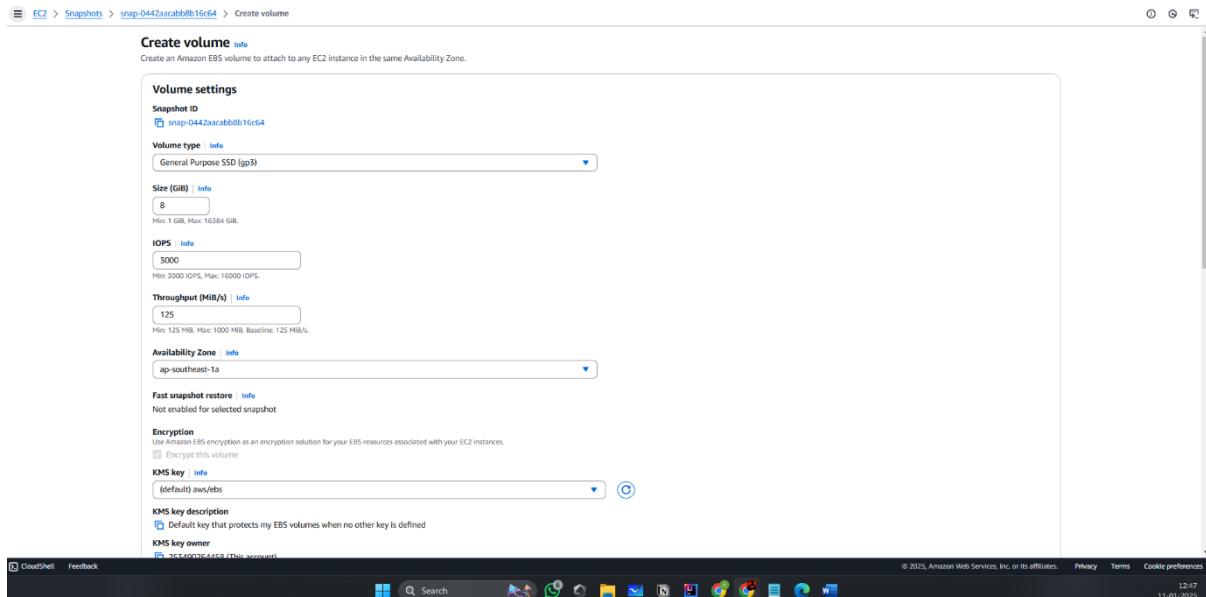
Snapshots (1/1) Info

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress	Encryption	KMS key ID	KMS key al...
snap-0442aacabb8b16c64	8 GiB	Copied snap-0128ccbe...	Standard	Completed	2025/01/11 12:43 GMT+5...	100%	Encrypted	ad533f9e-2af...	aws/ebs	

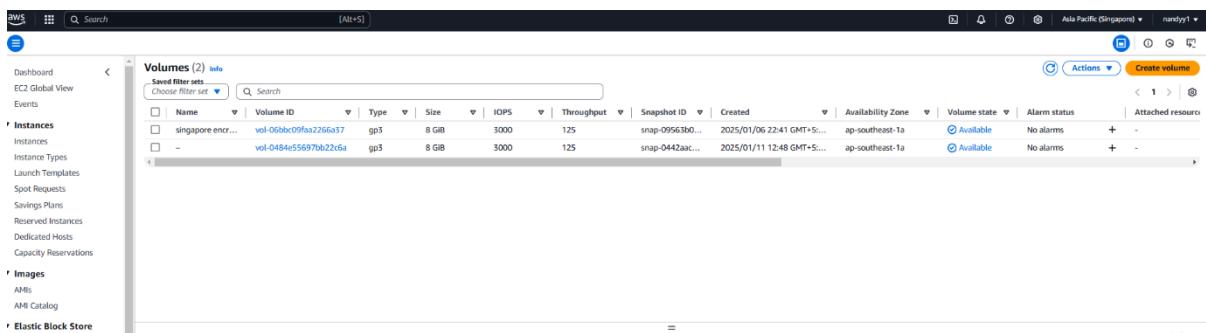
Create snapshot

- Create volume from snapshot**
- Create image from snapshot
- Copy snapshot
- Launch copy duration calculator
- Delete snapshot
- Manage tags
- Snapshot settings**
- Archiving
- Recycle Bin

Snapshot ID: snap-0442aacabb8b16c64



The output for task 5 is creating a snapshot to another region and availability zone and creating a volume from the snapshot

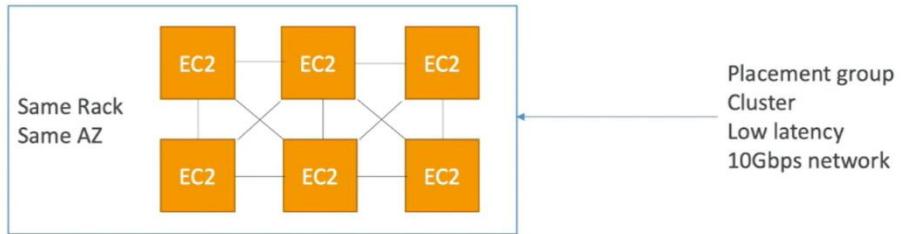


Task-6 What is Placement group

Placement Groups

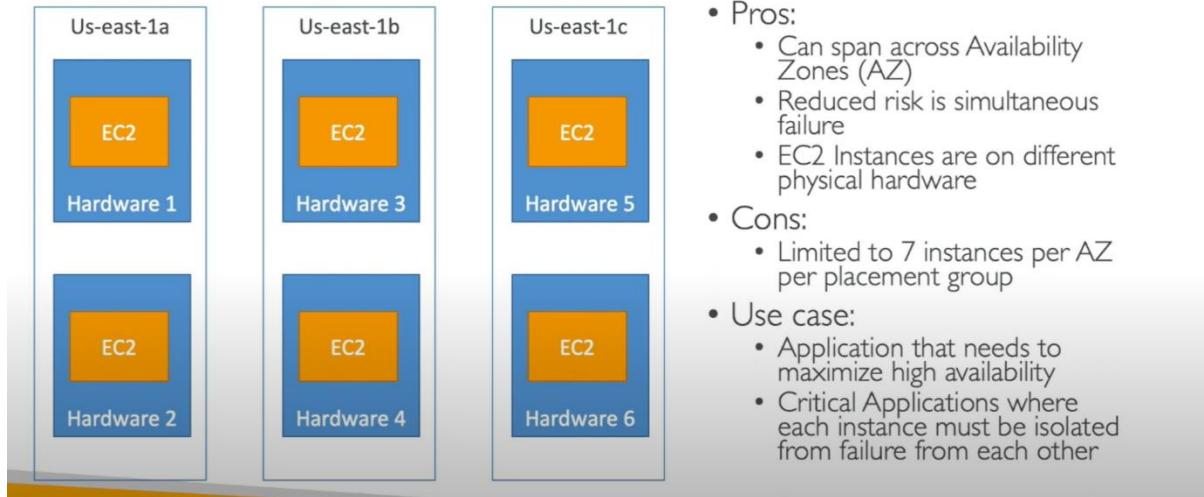
- Sometimes you want control over the EC2 Instance placement strategy
- That strategy can be defined using placement groups
- When you create a placement group, you specify one of the following strategies for the group:
 - *Cluster*—clusters instances into a low-latency group in a single Availability Zone
 - *Spread*—spreads instances across underlying hardware (max 7 instances per group per AZ) – critical applications
 - *Partition*—spreads instances across many different partitions (which rely on different sets of racks) within an AZ. Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)

Placement Groups Cluster

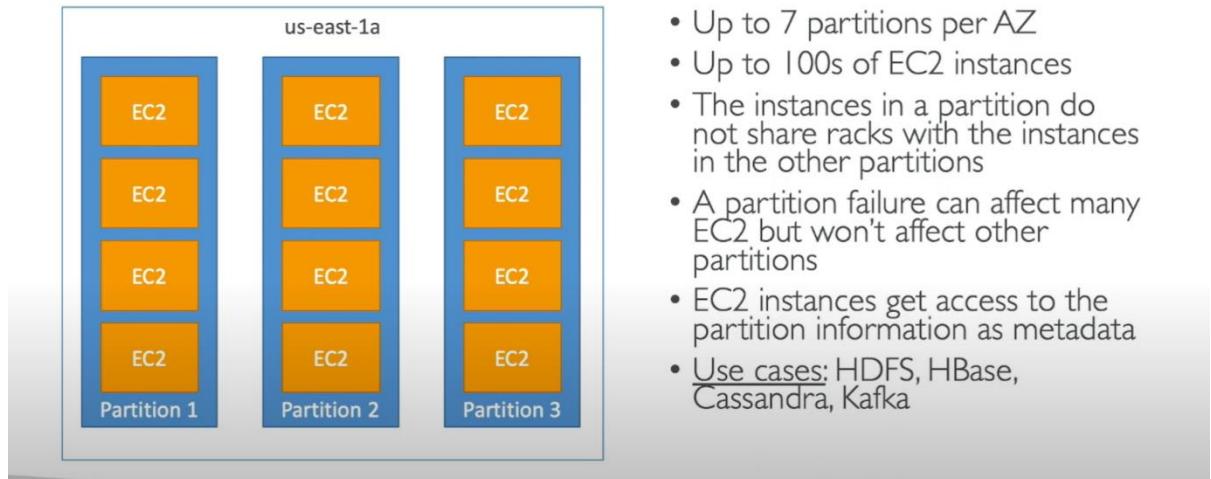


- Pros: Great network (10 Gbps bandwidth between instances)
- Cons: If the rack fails, all instances fail at the same time
- Use case:
 - Big Data job that needs to complete fast
 - Application that needs extremely low latency and high network throughput

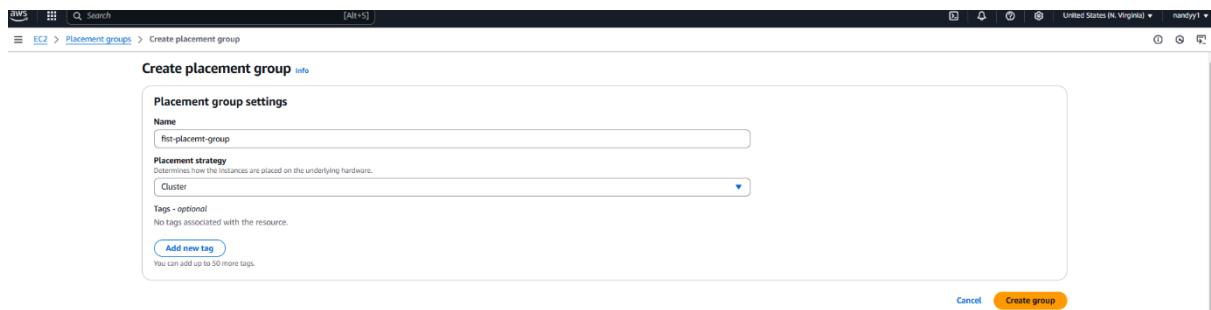
Placement Groups Spread



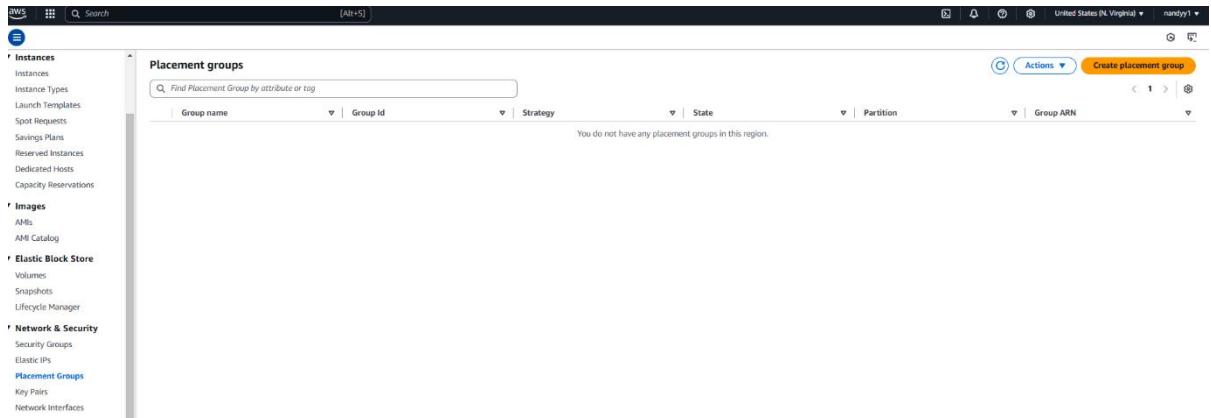
Placements Groups Partition



Step: 1 -> Go to the Aws console and go to Network & security under Placement Groups



Step-2: Click create placement group and click cluster and name



Create placement group Info

Placement group settings

Name
my-third-placement-groups

Placement strategy
Determines how the instances are placed on the underlying hardware.

Partition

Number of partitions
Choose the number of partitions to create in this placement group.
3

Tags - optional
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Create group

Step-3: after creating a cluster create a instance and attach the placement group under the advance settings

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name
placement-group-instance

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recent **Quick Start**

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-05576a079321f21f8 (64-bit (x86), uefi-preferred) / ami-05ecf97a33b0705c2 (64-bit (Arm), uefi)
Virtualization: hvm | ENI enabled: true | Root device type: ebs

Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250107.0 x86_64 HVM kernel-6.1

Architecture 64-bit (x86) **Boot mode** uefi-preferred **AMI ID** ami-05576a079321f21f8 **Username** ec2-user **Verified provider**

Summary
Number of instances: Info 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6... [read more](#)
ami-05576a079321f21f8

Virtual server type (instance type)
t2.micro
launch-wizard-1

Firewall (security group)

Storage (volumes)
1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) included in the Free Tier. See the [AWS Free Tier Terms and Conditions](#). 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

Launch Instance Cancel Preview code

This is the output of the following task 6. Placement groups

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Placement group' section is highlighted, displaying a dropdown menu with three options: 'first-placemt-group', 'my-second-placemt-group', and 'my-third-placemt-groups'. The 'first-placemt-group' option is selected and has a checkmark next to it. Below the dropdown, there are two radio button options: 'Capacity Blocks' and 'Spot instances', both of which are unselected. At the bottom of the placement group section, there is a link to 'Create new placement group'.

Task-7: What are the different types of Instance types

<https://aws.amazon.com/ec2/instance-types/> (Ref)

PAGE CONTENT

Compute Optimized

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this category are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

[General Purpose](#)

[Compute Optimized](#)

[Memory Optimized](#)

[Accelerated Computing](#)

[Storage Optimized](#)

[HPC Optimized](#)

[Instance Features](#)

[Measuring Instance Performance](#)

C8g	C7g	C7gn	C7i	C7i-flex	C7a	C6g	C6gn	C6i	C6in	C6a	C5	C5n																		
C5a	C4																													
<p>Amazon EC2 C8g instances are powered by AWS Graviton4 processors. They deliver the best price performance in Amazon EC2 for compute-intensive workloads.</p>																														
<p>Features:</p> <ul style="list-style-type: none">• Powered by custom-built AWS Graviton4 processors• Larger instance sizes with up to 3x more vCPUs and memory than C7g instances• Features the latest DDR5-5600 memory• Optimized for Amazon EBS by default• Supports Elastic Fabric Adapter (EFA) on c8g.24xlarge, c8g.48xlarge, c8g.metal-24xl, and c8g.metal-48xl• Powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor																														
<table border="1"><thead><tr><th>Instance size</th><th>vCPU</th><th>Memory (GiB)</th><th>Instance storage (GB)</th><th>Network bandwidth (Gbps)</th><th>Amazon EBS bandwidth (Gbps)</th></tr></thead><tbody><tr><td>c8g.medium</td><td>1</td><td>2</td><td>EBS-only</td><td>Up to 12.5</td><td>Up to 10</td></tr><tr><td>c8g.large</td><td>2</td><td>4</td><td>EBS-only</td><td>Up to 12.5</td><td>Up to 10</td></tr></tbody></table>													Instance size	vCPU	Memory (GiB)	Instance storage (GB)	Network bandwidth (Gbps)	Amazon EBS bandwidth (Gbps)	c8g.medium	1	2	EBS-only	Up to 12.5	Up to 10	c8g.large	2	4	EBS-only	Up to 12.5	Up to 10
Instance size	vCPU	Memory (GiB)	Instance storage (GB)	Network bandwidth (Gbps)	Amazon EBS bandwidth (Gbps)																									
c8g.medium	1	2	EBS-only	Up to 12.5	Up to 10																									
c8g.large	2	4	EBS-only	Up to 12.5	Up to 10																									



Amazon EC2 Instance types

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

PAGE CONTENT

General Purpose

[General Purpose](#)

[Compute Optimized](#)

[Memory Optimized](#)

[Accelerated Computing](#)

[Storage Optimized](#)

[HPC Optimized](#)

[Instance Features](#)

[Measuring Instance Performance](#)

Amazon EC2 M8g instances are powered by AWS Graviton4 processors. They deliver the best price performance in Amazon EC2 for general purpose workloads.

Features:

- Powered by custom-built AWS Graviton4 processors
- Larger instance sizes with up to 3x more vCPUs and memory than M7g instances
- Features the latest DDR5-5600 memory
- Optimized for Amazon EBS by default
- Supports [Elastic Fabric Adapter \(EFA\)](#) on m8g.24xlarge, m8g.48xlarge, m8g.metal-24xl, and m8g.metal-48xl



Memory Optimized

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

General Purpose

Compute Optimized

Memory Optimized

Accelerated Computing

Storage Optimized

HPC Optimized

Instance Features

Measuring Instance Performance

R8g	R7g	R7i	R7iz	R7a	R6g	R6i	R6in	R6a	R5	R5n	R5b	R5a	R4
U7i	High Memory (U-1)	X8g	X2gd	X2idn	X2iedn	X2iezn	X1	X1e					z1d

[Amazon EC2 R8g instances](#) are powered by AWS Graviton4 processors. They deliver the best price performance in Amazon EC2 for memory-intensive workloads.

Features:

- Powered by custom-built AWS Graviton4 processors
- Larger instance sizes with up to 3x more vCPUs and memory than R7g instances
- Features the latest DDR5-5600 memory
- Optimized for Amazon EBS by default
- Supports [Elastic Fabric Adapter \(EFA\)](#) on r8g.24xlarge, r8g.48xlarge, r8g.metal-24x, and r8g.metal-48x
- Powered by the [AWS Nitro System](#), a combination of dedicated hardware and lightweight hypervisor

Instance size	vCPU	Memory (GiB)	Instance storage (GB)	Network bandwidth (Gbps)	EBS bandwidth (Gbps)
r8g.medium	1	8	EBS-only	Up to 12.5	Up to 10
r8g.large	2	16	EBS-only	Up to 12.5	Up to 10
r8g.xlarge	4	32	EBS-only	Up to 12.5	Up to 10

Accelerated Computing

Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

General Purpose

Compute Optimized

Memory Optimized

Accelerated Computing

Storage Optimized

HPC Optimized

Instance Features

Measuring Instance Performance

P5	P4	G6e	G6	G5g	G5	G4dn	G4ad	Trn2	Trn1	Inf2	Inf1	DL1
DL2q	F2	VT1										

[Amazon EC2 P5 instances](#) are the latest generation of GPU-based instances and provide the highest performance in Amazon EC2 for deep learning and high performance computing (HPC).

Features:

- Intel Sapphire Rapids CPU and PCIe Gen5 between the CPU and GPU in P5en instances; 3rd Gen AMD EPYC processors (AMD EPYC 7R13) and PCIe Gen4 between the CPU and GPU in P5 and P5e instances.
- Up to 8 NVIDIA H100 (in P5) or H200 (in P5e and P5en) Tensor Core GPUs
- Up to 3,200 Gbps network bandwidth with support for Elastic Fabric Adapter (EFA) and NVIDIA GPUDirect RDMA (remote direct memory access)
- 900 GB/s peer-to-peer GPU communication with NVIDIA NVSwitch

Instance	GPUs	vCPUs	Instance Memory (TiB)	GPU Memory	Network Bandwidth	GPUDirect RDMA	GPU Peer to Peer	Instance Storage (TB)	EBS Bandwidth (Gbps)
p5.48xlarge	8 H100	192	2	640 GB HBM3	3200 Gbps EFAv2	Yes	900 GB/s NVSwitch	8 x 3.84 NVMe SSD	80
p5e.48xlarge	8 H200	192	2	1128 GB HBM3	3200 Gbps EFAv2	Yes	900 GB/s NVSwitch	8 x 3.84 NVMe SSD	80

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

I8g	I7ie	I4g	Im4gn	Is4gen	I4i	I3	I3en	D3	D3en	D2	H1																														
<p>Amazon EC2 I8g instances are powered by AWS Graviton4 processors and 3rd generation AWS Nitro SSDs. They deliver the best compute and storage performance among storage-optimized Amazon EC2 instances.</p> <p>Features:</p> <ul style="list-style-type: none"> Powered by custom-built AWS Graviton4 processors Featuring up to 22.5 TB of local NVMe SSD instance storage with 3rd generation AWS Nitro SSDs. Features the latest DDR5-5600 memory Up to 56.25 Gbps of network bandwidth Up to 30 Gbps of bandwidth to Amazon Elastic Block Store (EBS) Powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor 																																									
<table border="1"> <thead> <tr> <th>Instance Size</th><th>vCPU</th><th>Memory (GiB)</th><th>Instance Storage (GB)</th><th>Network Bandwidth (Gbps)</th><th>EBS Bandwidth (Gbps)</th></tr> </thead> <tbody> <tr> <td>I8g.large</td><td>2</td><td>16</td><td>1 x 468 GB = 468 GB</td><td>Up to 10</td><td>Up to 10</td></tr> <tr> <td>I8g.xlarge</td><td>4</td><td>32</td><td>1 x 937 GB = 937 GB</td><td>Up to 10</td><td>Up to 10</td></tr> <tr> <td>I8g.2xlarge</td><td>8</td><td>64</td><td>1 x 1,875 GB = 1,875 GB</td><td>Up to 12</td><td>Up to 10</td></tr> <tr> <td>I8g.4xlarge</td><td>16</td><td>128</td><td>1 x 3,750 GB = 3,750 GB</td><td>Up to 25</td><td>Up to 10</td></tr> </tbody> </table>												Instance Size	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)	EBS Bandwidth (Gbps)	I8g.large	2	16	1 x 468 GB = 468 GB	Up to 10	Up to 10	I8g.xlarge	4	32	1 x 937 GB = 937 GB	Up to 10	Up to 10	I8g.2xlarge	8	64	1 x 1,875 GB = 1,875 GB	Up to 12	Up to 10	I8g.4xlarge	16	128	1 x 3,750 GB = 3,750 GB	Up to 25	Up to 10
Instance Size	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)	EBS Bandwidth (Gbps)																																				
I8g.large	2	16	1 x 468 GB = 468 GB	Up to 10	Up to 10																																				
I8g.xlarge	4	32	1 x 937 GB = 937 GB	Up to 10	Up to 10																																				
I8g.2xlarge	8	64	1 x 1,875 GB = 1,875 GB	Up to 12	Up to 10																																				
I8g.4xlarge	16	128	1 x 3,750 GB = 3,750 GB	Up to 25	Up to 10																																				

pes/#General_Purpose

Looking for previous generation instances that were not listed here? Please see the [Previous Generation Instances](#) page.

HPC Optimized

High performance computing (HPC) instances are purpose built to offer the best price performance for running HPC workloads at scale on AWS. HPC instances are ideal for applications that benefit from high-performance processors such as large, complex simulations and deep learning workloads.

Hpc7g	Hpc7a	Hpc6id	Hpc6a																								
<p>Amazon EC2 Hpc7g instances are designed for compute-intensive high performance computing (HPC) workloads, such as computational fluid dynamics (CFD), weather forecasting, and molecular dynamics.</p> <p>Features:</p> <ul style="list-style-type: none"> Up to 64 cores of Graviton3E processors with 128 GiB of memory Elastic Fabric Adapter (EFA) is enabled for internode network bandwidth speeds of up to 200 Gbps, delivering increased performance for network-intensive applications Powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor 																											
<table border="1"> <thead> <tr> <th>Instance Size</th><th>Physical Cores</th><th>Memory (GiB)</th><th>Instance Storage</th><th>EFA Network Bandwidth (Gbps)</th><th>Network Bandwidth (Gbps*)</th></tr> </thead> <tbody> <tr> <td>hpc7g.4xlarge</td><td>16</td><td>128</td><td>EBS-Only</td><td>200</td><td>25</td></tr> <tr> <td>hpc7g.8xlarge</td><td>32</td><td>128</td><td>EBS-Only</td><td>200</td><td>25</td></tr> <tr> <td>hpc7g.16xlarge</td><td>64</td><td>128</td><td>EBS-Only</td><td>200</td><td>25</td></tr> </tbody> </table>				Instance Size	Physical Cores	Memory (GiB)	Instance Storage	EFA Network Bandwidth (Gbps)	Network Bandwidth (Gbps*)	hpc7g.4xlarge	16	128	EBS-Only	200	25	hpc7g.8xlarge	32	128	EBS-Only	200	25	hpc7g.16xlarge	64	128	EBS-Only	200	25
Instance Size	Physical Cores	Memory (GiB)	Instance Storage	EFA Network Bandwidth (Gbps)	Network Bandwidth (Gbps*)																						
hpc7g.4xlarge	16	128	EBS-Only	200	25																						
hpc7g.8xlarge	32	128	EBS-Only	200	25																						
hpc7g.16xlarge	64	128	EBS-Only	200	25																						

Task-8 What is launch Template? Create one instance through Launch template

It is a template which we can create and use like an ami, The AMI is created for the instances and the launch template is created for the whole Ec2 instance configuration which comes under every settings.

Step1: Click launch template and select the particulars which needed to be configured

The screenshot shows the 'Create launch template' wizard on the AWS EC2 console. The current step is 'Launch template name and description'. The 'Launch template name - required' field contains 'MyTemplate'. The 'Template version description' field contains 'A prod webserver for MyApp'. Under 'Auto Scaling guidance', there is a checkbox for 'Select this if you intend to use this template with EC2 Auto Scaling'. Below this, there are sections for 'Template tags' and 'Source template'. On the right side, a summary panel displays the selected AMI ('Amazon Linux 2023.6.2'), instance type ('t2.micro'), security group ('launch-wizard-1'), and storage ('1 volume(s) - 8 GB'). A note about the free tier is also present.

Once it is filled a summary will be shown on the right

The screenshot shows the 'Create launch template' wizard on the AWS EC2 console. The current step is 'Advanced details'. It includes sections for 'Firewall (security groups)', 'Storage (volumes)', 'Resource tags', and 'Advanced details'. The 'Firewall (security groups)' section shows a selected security group 'launch-wizard-1'. The 'Storage (volumes)' section shows an EBS volume 'Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3), 3000 IOPS)'. The 'Resource tags' section shows an option to 'Add new tag'. The 'Advanced details' section is currently collapsed. On the right side, a summary panel displays the selected AMI ('Amazon Linux 2023.6.2'), instance type ('t2.micro'), security group ('launch-wizard-1'), storage ('1 volume(s) - 8 GB'), and a note about the free tier.

Launch Templates (1/1) Info

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Managed	Operator
lt-0db2e9e39fb5c65ad	my-basic-configure-web	1	1	2025-01-11T08:18:33.000Z	arn:aws:iam::253490764458:root	False	-

my-basic-configure-web (lt-0db2e9e39fb5c65ad)

Launch template details

Launch template ID lt-0db2e9e39fb5c65ad	Launch template name my-basic-configure-web	Default version 1	Owner arn:aws:iam::253490764458:root
--	--	----------------------	---

Launch template version details

Version 1 (Default)	Description v1	Date created 2025-01-11T08:18:33.000Z	Created by arn:aws:iam::253490764458:root
Instance details		Network interfaces	
AMI ID ami-05576a079321f21f8	Instance type t2.micro	Availability Zone	Key pair name ppk-key
Security groups		Security group IDs sg-0b7aaf9c0ee3f2392	

Step 2 : click launch instance form the template

Launch Templates (1/1) Info

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-0db2e9e39fb5c65ad	my-basic-configure-web	1	1	2025-01-11T08:18:33.000Z	arn:aws:iam::253490764458:root

my-basic-configure-web (lt-0db2e9e39fb5c65ad)

Launch template details

Launch template ID lt-0db2e9e39fb5c65ad	Launch template name my-basic-configure-web	Default version 1	Owner arn:aws:iam::253490764458:root
--	--	----------------------	---

Actions ▾

- Launch instance from template
- Modify template (Create new version)
- Delete template
- Delete template version
- Set default version
- Manage tags
- Create Spot Fleet
- Create Auto Scaling group
- View details

Select the instance name and check all the configurations are correct

EC2 > Launch templates > Launch instance from template

Launch instance from template

Launching from a template allows you to launch from an instance configuration that you would have saved in the past. These saved configurations can be reused and shared with other users to standardize launches across an organization.

Choose a launch template

Source template
my-basic-configure-web
ID: lt-0db2e9e39fb5c65ad

1 (Default)
v1

Instance details

Your instance details are listed below. Any fields that are not specified as part of the configuration below will use the template or default values for those fields. Ensure that you have permissions to override these parameters or your instance launch will fail.

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

AMI from catalog **Recents** **Quick Start**

Name: Amazon Linux 2023 AMI
Description: Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Image ID: ami-05576a079321f21f8
Username: ec2-user

Summary

Number of instances: 1

Software image (AMI): Amazon Linux 2023 AMI 2023.6.2... [read more](#)
ami-05576a079321f21f8

Virtual server type (instance type): t2.micro

Firewall (security group): launch-wizard-1

Storage (volumes): 1 volume(s) - 8 GB

Free tier in your first year includes 750 hours of t2.micro (or t3.micro in the regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 50 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch Instance **Preview code**

The output is Task 8 -> from the template an instance is being created

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Mac
placement-gro...	i-04d6bc7b5ce77cac	Terminated	t2.micro	-	View alarms +	us-east-1c	-	-	-	-	dis
	i-05e308c170c6ed62	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-3-80-198-204.com...	3.80.198.204	-	-	dis

Task-9 Describe about different volume types

1. General Purpose SSD (gp3 and gp2)

gp3 (Recommended)

- **Use Case:** Most applications requiring high performance at low cost, including boot volumes, small databases, and development/test environments.
- **Performance:**
 - Baseline IOPS: 3,000 IOPS (default, scalable up to 16,000 IOPS).
 - Baseline Throughput: 125 MiB/s (scalable up to 1,000 MiB/s).
- **Cost Advantage:** Offers a lower price and more consistent performance compared to gp2.
- **IOPS Scaling:** Independent of volume size.

gp2

- **Use Case:** General-purpose workloads.
- **Performance:**
 - Baseline IOPS: 3 IOPS per GiB (minimum 100 IOPS, maximum 16,000 IOPS).
 - Burstable IOPS: Allows temporary performance bursts for smaller volumes.
- **IOPS Scaling:** Directly proportional to volume size.

2. Provisioned IOPS SSD (io1 and io2)

io2 (Recommended for High Durability)

- **Use Case:** Critical applications requiring sustained high performance, low latency, and high durability (e.g., large databases like Oracle, Microsoft SQL Server).
- **Performance:**
 - Up to 64,000 IOPS.
 - Durability: 99.999% (higher than io1).
 - Supports Multi-Attach (attach to multiple EC2 instances).
- **Cost:** Higher than gp3/gp2.

io1

- **Use Case:** High-performance workloads with lower durability needs than io2.
 - **Performance:**
 - Up to 64,000 IOPS.
 - Durability: 99.8%-99.9%.
 - **Cost:** Slightly less than io2 but less durable.
-

3. Throughput Optimized HDD (st1)

- **Use Case:** Streaming workloads requiring high throughput, such as big data, data warehouses, and log processing.
 - **Performance:**
 - Sequential Throughput: Up to 500 MiB/s.
 - Baseline Performance: 40 MB/s per TiB.
 - IOPS: Not suitable for small or random I/O operations.
 - **Cost:** Lower than SSD volumes.
-

4. Cold HDD (sc1)

- **Use Case:** Infrequently accessed data, such as cold storage and archival backups.
- **Performance:**

- Sequential Throughput: Up to 250 MiB/s.
 - Baseline Performance: 12 MB/s per TiB.
 - IOPS: Not suitable for small or random I/O operations.
 - **Cost:** Lowest among all EBS volume types.
-

5. Magnetic (Standard) [Deprecated]

- **Use Case:** Legacy workloads with low performance requirements.
 - **Performance:** Lower IOPS and throughput compared to modern volumes.
 - **Cost:** Cheaper than SSDs but rarely used due to the availability of better options.
-

Comparison Table

Volume Type	Max IOPS	Max Throughput	Use Case	Cost
gp3	16,000	1,000 MiB/s	General workloads, boot volumes	\$\$
gp2	16,000	250 MiB/s	General workloads	\$\$
io2	64,000	1,000 MiB/s	Critical apps, large databases	\$\$\$\$
io1	64,000	1,000 MiB/s	High-performance workloads	\$\$\$
st1	N/A	500 MiB/s	Streaming, big data, log processing	\$
sc1	N/A	250 MiB/s	Archival, infrequently accessed workloads	\$

Choosing the Right Volume Type (Industry Examples)

- **Web Server Hosting (gp3):** A general-purpose SSD provides low latency and high performance for serving website traffic.
- **E-commerce Databases (io2):** Critical, low-latency applications need io2 for consistent performance and durability.
- **Big Data Analytics (st1):** For high throughput during log analysis or data streaming.
- **Backup Storage (sc1):** Cold HDDs for storing archival data at minimal cost.

Let me know if you need more clarity or assistance setting up any of these volume types!

Task 10: Give region based access to IAM user

Step1: Create an Ec2 instance

The screenshot shows the AWS EC2 Instances page. It lists two instances: 'local-machine1' (running, t2.micro, Public IP 44.203.132.4, Private IP 172.31.84.80) and 'lambda-machine...' (terminated, t2.micro). The left sidebar includes sections for Instances, Images, Elastic Block Store, Network & Security, and more.

The screenshot shows the AWS EC2 Instances page with a single instance named 'local-machine...' in a Pending state (t2.micro, Public IP ec2-3-110-191-176.ap-south-1.compute-1.amazonaws.com, Private IP 3.110.191.176). The left sidebar includes sections for Instances, Images, Elastic Block Store, Network & Security, and more.

Step2: create an iam user

The screenshot shows the 'Specify user details' step of the IAM User creation wizard. It includes fields for 'User name' (demo-user), 'Console password' (radio button for Autogenerated password), 'Custom password' (radio button for Custom password), and 'Show password' (checkbox). Other options like 'Provide user access to the AWS Management Console - optional' and 'I want to create an IAM user' are also present.

Step 4: go to the iam browser and see if any instance running user api fetch error

The screenshot shows the AWS IAM browser interface. On the left, a sidebar lists various AWS services: Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area is titled "Resources" and displays a table of Amazon EC2 resources in the Asia Pacific (Mumbai) Region. The table includes columns for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, Volumes, and API Error status. Most rows show an "API Error" status. A "Launch instance" section contains buttons for "Launch instance" and "Migrate a server". Below it, a note says "Your instances will launch in the Asia Pacific (Mumbai) Region". A "Service health" section shows an "AWS Health Dashboard" button and a red-bordered box indicating "An error occurred" with the message "An error occurred retrieving service health information". Another red-bordered box in the "Zones" section shows "An error occurred" with the message "An error occurred retrieving service health information". A "CloudWatch Metrics" section is partially visible. On the right, sections for "EC2 Free Tier" (info), "EC2 free tier offers in use" (0), "End of month forecast" (with a detailed error message about a user not authorized to perform the "freetier:GetFreeTierUsage" action), "Exceeds free tier" (with a similar error message), and "Account attributes" (with a red-bordered box showing "An error occurred" and "An error occurred checking for a default VPC"). The bottom of the page includes standard copyright and footer links.

This screenshot shows the AWS IAM browser interface, specifically the "AWS Health" dashboard. The top navigation bar includes "Console Home" (Info), "Reset to default layout", and "+ Add widgets". The main content area is titled "AWS Health" (Info). It features several cards: "Recently visited" (Info) which says "No recently visited services" and lists EC2, S3, RDS, and Lambda; "Welcome to AWS" (Info) which includes "Getting started with AWS" and "Training and certification"; "Applications (0)" (Info) which says "Region: US East (N. Virginia)" and has a "Create application" button; "Cost and usage" (Info) which shows "Current month costs" (Access denied), "Forecasted month end costs" (Access denied), and "Savings opportunities" (Access denied); and "AWS Health" (Info) which says "No health data" and notes "You don't have permissions to access AWS Health". A red-bordered box highlights an "Access denied" message in the Applications card. The bottom of the page includes standard copyright and footer links.

Step 5 : Create a policy for the user

The screenshot shows the AWS IAM Policies page. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Access management' (User groups, Users, Roles, Policies), 'Access reports' (Access Analyzer, External access, Unused access, Analyzer settings), 'Credential report', 'Organization activity', 'Service control policies', and 'Resource control policies'. The main content area displays a table titled 'Policies (1317)'. The table has columns for 'Policy name', 'Type' (AWS managed or AWS managed - job function), 'Used as' (Permissions policy, None), and 'Description'. The table lists numerous policies, such as 'AccessAnalyzerServiceRolePolicy', 'AdministratorAccess', 'AIOPsAssistantPolicy', and various AlexaForBusiness and Amazon API Gateway policies.

Step 6: In the request condition section click add another condition

Search ec2:Region

The screenshot shows the 'Create policy' wizard. On the left, there's a sidebar with 'VPC' and 'AWS Lambda' sections. The main area shows a list of actions: 'vpc-endpoint', 'vpc-endpoint-connection', 'vpc-endpoint-service', 'vpc-endpoint-service-permission', 'vpc-flow-log', 'vpc-peering-connection', 'vpn-connection', 'vpn-connection-device-type', and 'vpn-gateway'. Below this is a section for 'Request conditions - optional' with checkboxes for 'User is MFA Authenticated', 'Requested from IP', and 'Requested from IP address'. At the bottom, there are buttons for '+ Add another condition' and '+ Add more permissions'. A modal window titled 'Add request condition' is open, showing fields for 'Condition key' (set to 'ec2:Region'), 'Qualifier' (set to 'Default'), and 'Operator' (set to 'Select operator'). To the right of the modal, there are several checkboxes for 'Any in this account'.

Step 1
Specify permissions
Step 2
Review and create

Review and create Info
Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
restrict-user-access
Maximum 128 characters. Use alphanumeric and '+'-'_-' characters.

Description - optional
Add a short explanation for this policy.
Maximum 1,000 characters. Use alphanumeric and '+'-'_-' characters.

Permissions defined in this policy Info
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 438 services)			
Service	Access level	Resource	Request condition
EC2	Full access	All resources	ec2:Region = ap-south-1

Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Create policy

Policy had been created

demo-USER Info

Summary

ARN: arn:aws:iam::253490764458:user/demo-user
Console access: Enabled without MFA
Created: January 11, 2025, 14:37 (UTC+05:30)
Last console sign-in: Never
Access key 1: Create access key

Permissions **Groups** **Tags** **Security credentials** **Last Accessed**

Permissions policies (2)
Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
IAMUserChangePassword	AWS managed	Directly
restrict-user-access	Customer managed	Directly

Permissions boundary (not set)

Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

Generate policy
No requests to generate a policy in the past 7 days.

Step7: Once the policy had been created go to the iam user and see how many instance running in the particular region so the action had been seen in the region by adding the permissions

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images, Elastic Block Store, Network & Security, Load Balancing, and CloudWatch Feedback. The main content area displays a table of instances. One instance is listed: 'local-machine...' with ID 'i-0b24fbe6897752f4'. The instance is shown as 'Running' with status code '2/2 checks passed'. It is associated with the user 'arn:aws:ec2:ap-south-1:123456789012:instance/local-machine-...'. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, and Elastic IP.

The screenshot shows the AWS EC2 Home page. The left sidebar has the same navigation options as the previous screenshot. The main area contains several cards: 'Resources' (listing Instances (running) 1, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 1, Key pairs 2, Load balancers 0, Placement groups 0, Security groups 6, Snapshots 0, and Volumes 1), 'Launch instance' (button to 'Launch instance'), 'Service health' (button to 'AWS Health Dashboard', showing an error message: 'An error occurred: An error occurred retrieving service health information'), 'Zones' (listing zones ap-south-1a, ap-south-1b, and ap-south-1c with their respective Zone IDs aps1-az1, aps1-az3, and aps1-az2), 'Instance alarms' (listing an alarm for 'cloudwatch:DescribeAlarms' action), and 'Account attributes' (listing Default VPC (vpc-0a1e7acb0967c4d9d), Settings (Data protection and security, Allowed AMIs, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences), and End of month forecast (warning about unauthorized API calls)).

In the different region it can't be seen

The screenshot shows the AWS EC2 Instances page in the us-east-1 region. The URL is https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances. The left sidebar shows navigation links for Dashboard, EC2 Global View, Events, Instances (selected), Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'Instances Info' and contains a search bar and a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, Elastic IP, IPv6 IP, and Mon. A red error message at the top states: 'You are not authorized to perform this operation. User: arnawsiam:253490764458:user/demo-user is not authorized to perform: ec2:DescribeInstances because no identity-based policy allows the ec2:DescribeInstances action'. Below the message, there is a table with one row showing an instance named 'local-machine...' with an ID of 'i-0b24fb6e8977752f4', which is 'Running' and has an 't2.micro' instance type. It also shows '2/2 checks passed' and 'User: arnawsiam'.

The output for the task 10 had been done

The screenshot shows the AWS EC2 Instances page in the ap-south-1 region. The URL is https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#instances. The left sidebar shows the same navigation links as the previous screenshot. The main content area is titled 'Instances (1) Info' and shows the same instance details as the previous screenshot: 'local-machine...' (i-0b24fb6e8977752f4), 'Running', 't2.micro', '2/2 checks passed', and 'User: arnawsiam'. The table header includes additional columns: IPv6 IP and Mon. The status bar at the bottom indicates 'Last updated less than a minute ago'.

Task 11: Mount 2gb of volume in linux instance

Step1 : Create an ec2-instance and the volume will be created along with the instance

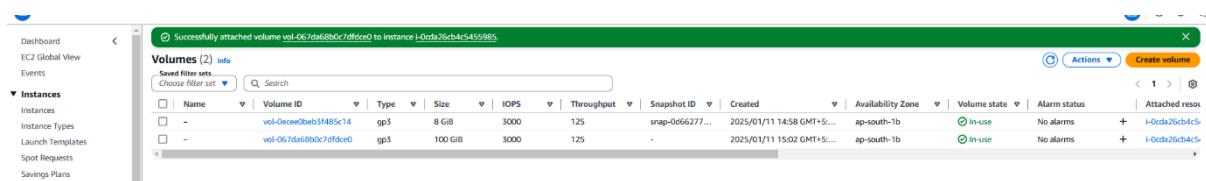
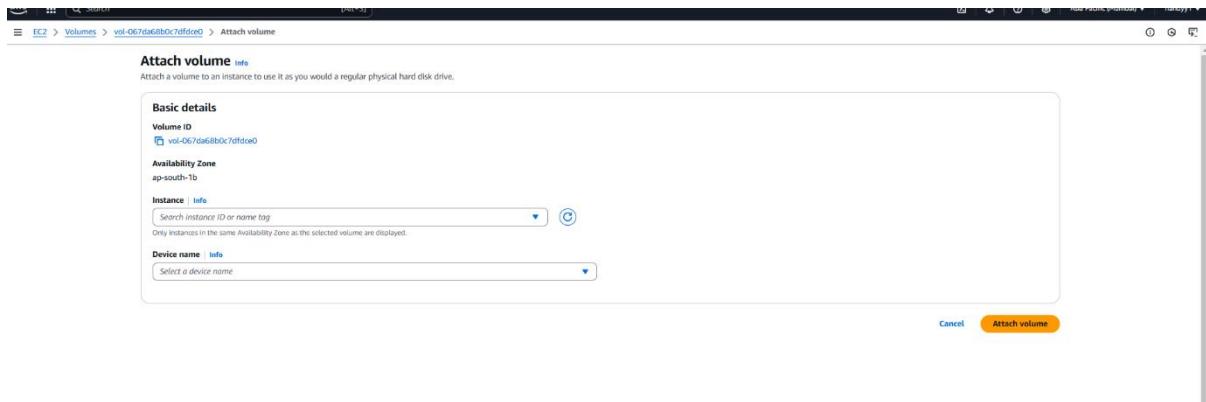
The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, a tag 'mount-volume' is added. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Amazon Linux 2023 AMI' is selected. The 'Description' section notes it has 5 years of support and is a secure, stable, and high-performance environment. The 'AMI ID' is ami-07b69f62c1d38b012. The 'Architecture' is 64-bit (x86), 'Boot mode' is uefi-preferred, and 'Username' is ec2-user. The 'Storage (volumes)' section shows one volume of 8 GB. On the right, a summary panel shows the instance type as t2.micro, a free tier warning, and a 'Launch instance' button.

Step 2: Create a volume and attach the instance

The screenshot shows the 'Volumes' page with two volumes listed: 'vol-060e627daf24c1367' (gp3, 20 GB) and 'vol-deceeb3f485c14' (gp3, 8 GB). Both volumes are in the 'Available' state. A success message at the top indicates 'Successfully created volume vol-060e627daf24c1367.'

Step 3: attach the volume to the ec2 instance

The screenshot shows the 'Volumes' page with the same two volumes. The 'vol-060e627daf24c1367' volume is selected. A context menu is open over this volume, with 'Attach volume' highlighted. The 'Volume ID: vol-060e627daf24c1367' details page is shown below, displaying volume information like Volume ID, Size (20 GB), Type (gp3), and Status (Available).



```
ec2-user@ip-172-31-3-6:~$ login as: ec2-user
[ec2-user@ip-172-31-3-6 ~]$ Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-3-6 ~]$ 
[ec2-user@ip-172-31-3-6 ~]$ 
[ec2-user@ip-172-31-3-6 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M    0   4.0M   0% /dev
tmpfs          475M    0  475M   0% /dev/shm
tmpfs          190M  456K  190M   1% /run
/dev/xvda1      8.0G  1.6G  6.4G  20% /
tmpfs          475M    0  475M   0% /tmp
/dev/xvda1s8     10M  1.3M  8.7M  13% /boot/efi
tmpfs           95M    0   95M   0% /run/user/1000
[ec2-user@ip-172-31-3-6 ~]$
```

Step4 : Enter the following commands on the CLI

```
[ec2-user@ip-172-31-9-130 ~]$ lsblk
NAME      MAJ:MIN   RM  SIZE RO TYPE MOUNTPOINTS
xvda      202:0     0   8G  0 disk
└─xvdal    202:1     0   8G  0 part /
└─xvdal27 259:0     0   1M  0 part
└─xvdal28 259:1     0  10M  0 part /boot/efi
xvdbf     202:14592  0  20G  0 disk
[ec2-user@ip-172-31-9-130 ~]$ sudo fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 457F738E-0372-43C3-9B6E-08BACC847AAC

Device        Start      End  Sectors Size Type
/dev/xvdal    24576 16777182 16752607   8G Linux filesystem
/dev/xvdal27  22528    24575     2048   1M BIOS boot
/dev/xvdal28   2048    22527    20480  10M EFI System

Partition table entries are not in disk order.

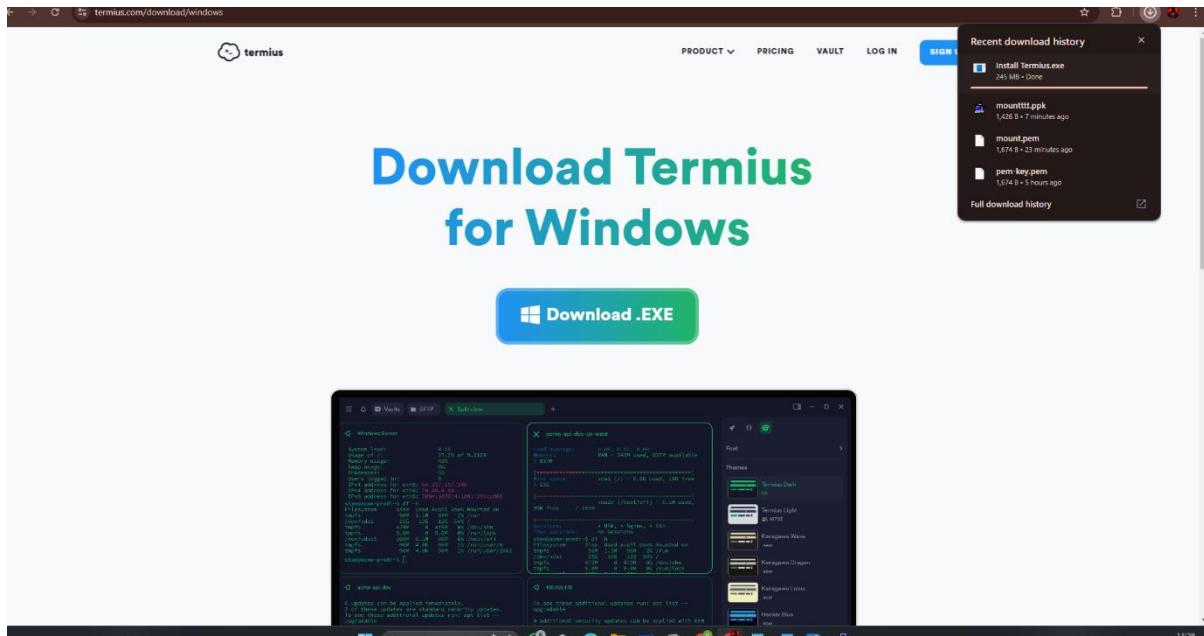
Disk /dev/xvdbf: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[ec2-user@ip-172-31-9-130 ~]$ sudo mkfs.ext4 /dev/xvdbf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5242880 4k blocks and 1310720 inodes
Filesystem UUID: 0cfcc77c-17e5-4dle-a35f-b58956fb5e16
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
      4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

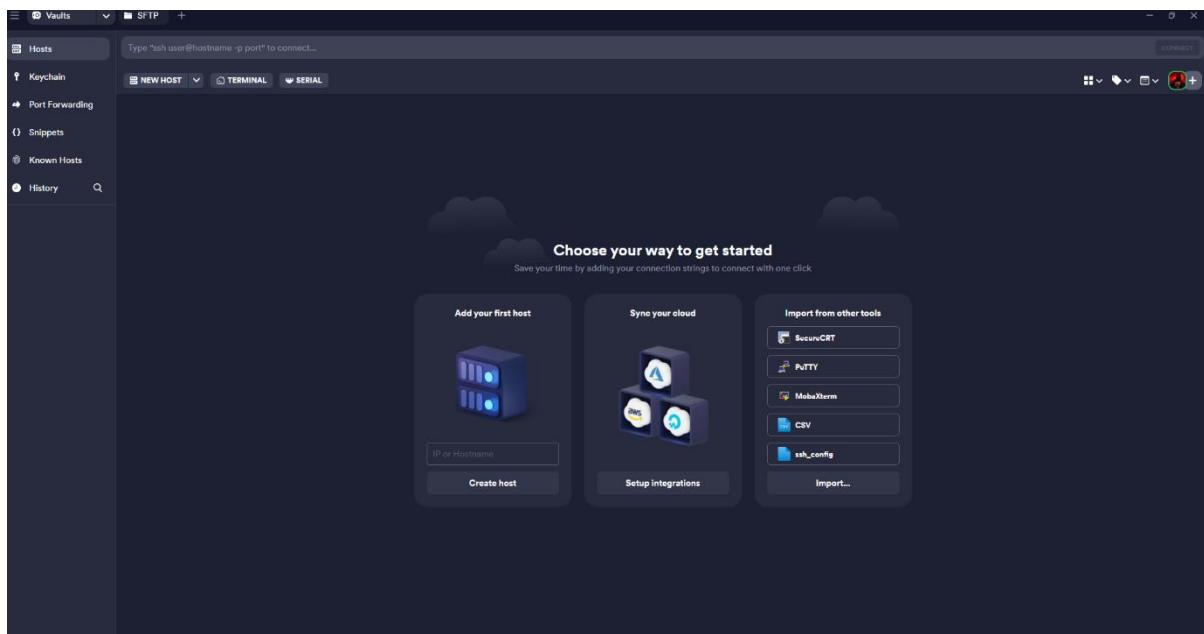
[ec2-user@ip-172-31-9-130 ~]$ sudo mkdir /mnt/data
[ec2-user@ip-172-31-9-130 ~]$ sudo mount /mnt/data /dev/xvdbf
mount: /mnt/data is not a block device.
[ec2-user@ip-172-31-9-130 ~]$ df -h
-bash: df-h: command not found
[ec2-user@ip-172-31-9-130 ~]$ sudo mount /dev/xvdbf /mnt/data
[ec2-user@ip-172-31-9-130 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs       4.0M    0  4.0M  0% /dev
tmpfs          475M    0  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvdal      8.0G  1.6G  6.4G 20% /
tmpfs          475M    0  475M  0% /tmp
/dev/xvdal28    10M   1.3M  8.7M 13% /boot/efi
tmpfs          95M    0   95M  0% /run/user/1000
/dev/xvdbf      20G   24K  19G  1% /mnt/data
[ec2-user@ip-172-31-9-130 ~]$ █
```

Task 12. Use any alternative tool than putty to connect linux instance

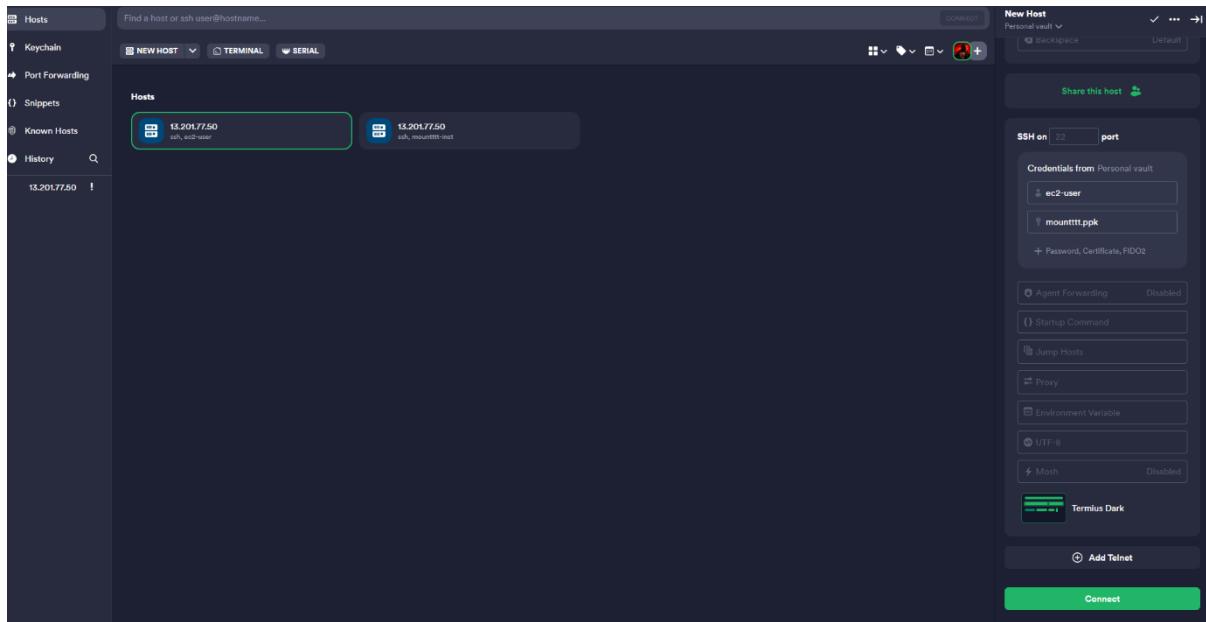
Step 1 – download Termius



Step -2 go into the termius



Step-3 On the right side of the termius there will be a set of options enter the user name
-> as ec2-user and password as .ppk file



Terminal is opened so this the alternative for putty



Task 13 . what are different purchase model of EC2 instance

13) what are different purchase model of EC2 instance

The purchase model in Amazon EC2 defines how you pay for and manage the compute resources you use. AWS offers different pricing and commitment options for EC2 instances to cater to various workload types, budget constraints, and operational flexibility.

These purchase models allow you to optimize costs while ensuring your workloads have

the required compute capacity.

Different types of purchase model of EC2 instance

1. On-Demand Instances

- Pay for compute capacity per second or hour without long-term commitments.
- Best for applications with short-term or unpredictable workloads.

Key Features:

- Highly flexible: Scale up/down as needed.
- No upfront payment required.
- Charged at a fixed rate per second or hour.

Use Cases:

- Development and testing environments.
- Applications with unpredictable demand.
- Temporary workloads.

2. Reserved Instances (RIs)

Commit to a 1-year or 3-year term in exchange for significant cost savings (up to 72% compared to On-Demand pricing).

Types of RIs:

- Standard RIs: Offers maximum discount but with limited flexibility to change instance types.

- Convertible RIs: Allows you to exchange the instance type, OS, or tenancy with
- slightly lower discounts.

Key Features:

- Requires upfront payment, partial upfront payment, or no upfront payment.
- Ideal for steady-state workloads.

Use Cases:

- Long-term, predictable workloads.
- Applications running 24/7, like web servers or databases.

3. Spot Instances

- Utilize unused EC2 capacity at up to 90% lower cost than On-Demand instances.
- AWS can terminate these instances with a 2-minute notice if the capacity is reclaimed.

Key Features:

- Highly cost-effective.
- Instances can be interrupted, so workloads must be fault-tolerant.

Use Cases:

- Bigdata analytics.
- Batch processing.
- Continuous Integration/Continuous Deployment (CI/CD) pipelines.

4. Savings Plans

Flexible pricing model that provides cost savings (up to 66%) in exchange for a commitment to a specific usage amount (\$/hour) over 1 or 3 years.

Key Features:

- Flexibility to change instance types and regions with the Compute Savings Plan.
- Predictable billing.

Use Cases:

- Long-running workloads with some flexibility in instance type or region.

5. Dedicated Hosts

Rent a physical server dedicated to your use, giving you control over instance placement.

Key Features:

- Ideal for compliance or regulatory needs (e.g., HIPAA, PCI DSS).
- Bring Your Own License (BYOL) support.

Use Cases:

- Regulatory or compliance-driven applications.
- Applications requiring isolation from other AWS customers.

6. Dedicated Instances

EC2 instances that run on single-tenant hardware, isolating your workloads from other customers.

Key Features:

- No upfront cost or long-term commitment.
- Provides a level of hardware isolation similar to Dedicated Hosts.

Use Cases:

- Applications needing some level of isolation but not full control of hardware placement.

Task 14 -> Tabulate 50 protocols with port number

HTTP	80	HyperText Transfer Protocol
HTTPS	443	HTTP Secure
FTP	21	File Transfer Protocol
FTPS	990	FTP Secure
SFTP	22	SSH File Transfer Protocol
SSH	22	Secure Shell
Telnet	23	Remote Login Protocol
SMTP	25	Simple Mail Transfer Protocol
SMTPS	465	Secure SMTP
POP3	110	Post Office Protocol v3
POP3S	995	Secure POP3
IMAP	143	Internet Message Access Protocol
IMAPS	993	Secure IMAP
DNS	53	Domain Name System
DHCP	67, 68	Dynamic Host Configuration Protocol
TFTP	69	Trivial File Transfer Protocol
SNMP	161, 162	Simple Network Management Protocol
RDP	3389	Remote Desktop Protocol
VNC	5900	Virtual Network Computing
NTP	123	Network Time Protocol
LDAP	389	Lightweight Directory Access Protocol
LDAPS	636	Secure LDAP
SIP	5060	Session Initiation Protocol
SIPS	5061	Secure SIP

RTP	5004, 5005	Real-time Transport Protocol
RTSP	554	Real-Time Streaming Protocol
SMB	445	Server Message Block Protocol
NetBIOS	137-139	NetBIOS Name Service, Datagram, and Session
ICMP	N/A	Internet Control Message Protocol
IGMP	N/A	Internet Group Management Protocol
BGP	179	Border Gateway Protocol
OSPF	89	Open Shortest Path First
IPsec	500	Internet Protocol Security (IKE)
GRE	47	Generic Routing Encapsulation
L2TP	1701	Layer 2 Tunneling Protocol
PPTP	1723	Point-to-Point Tunneling Protocol
MySQL	3306	MySQL Database Service
PostgreSQL	5432	PostgreSQL Database Service
MS SQL Server	1433	Microsoft SQL Server
MongoDB	27017	MongoDB Database Service