AWS EC2:

EC2 VM:

Amazon Machine Image (AMI)

Instance type

Keypair

Virtual Private Cloud (VPI)

Security Groups

Elastic Block Storage (EBS)

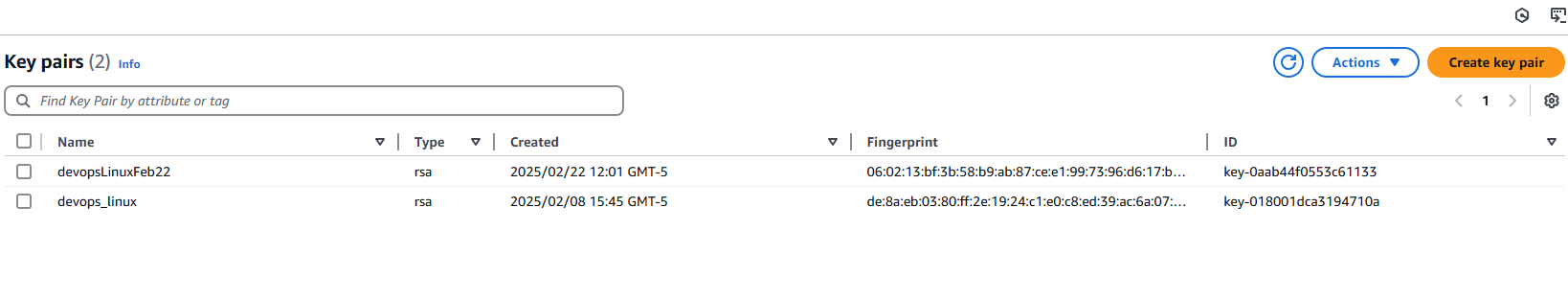
EBS (Volumes & Snapshots) -> Load-balancer, Types of Load-balancer, Auto-scaling

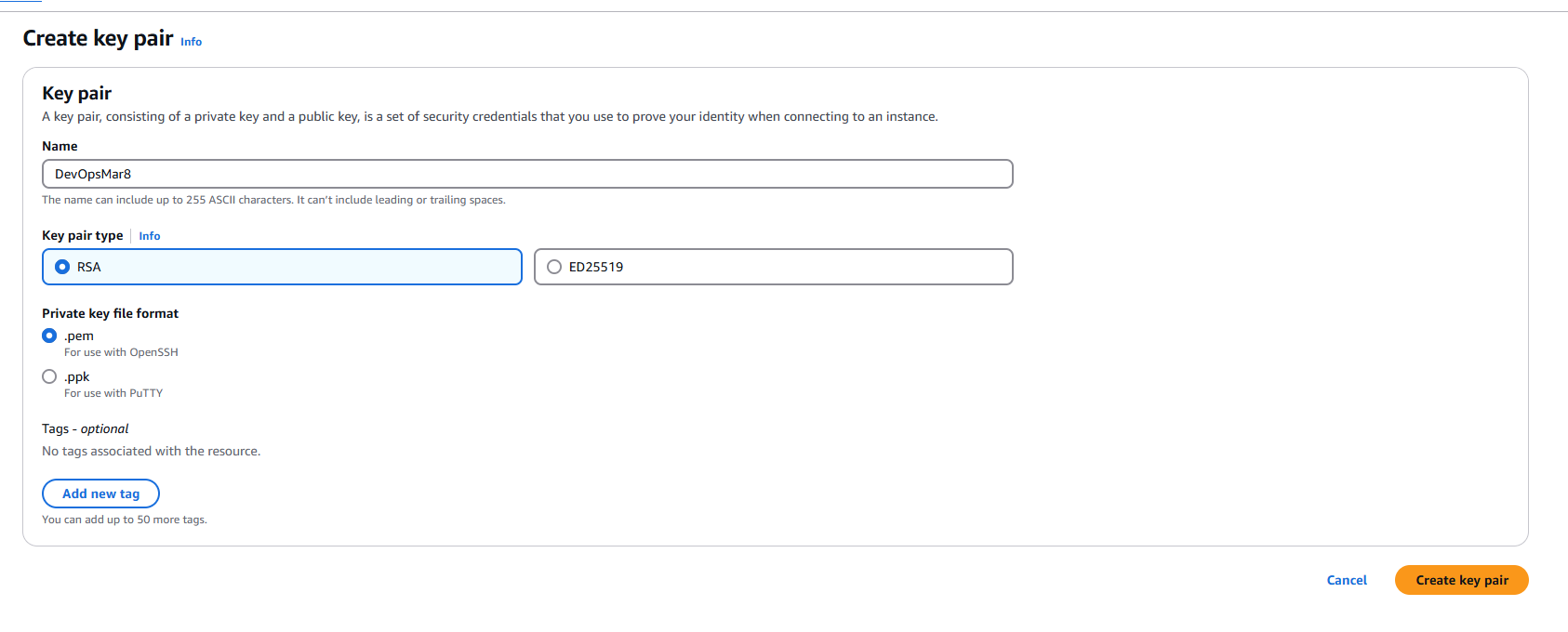
IPs

Static Website Hosting

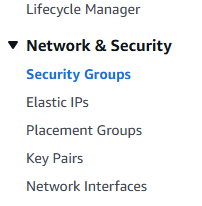
Practicals on EC2 VM:

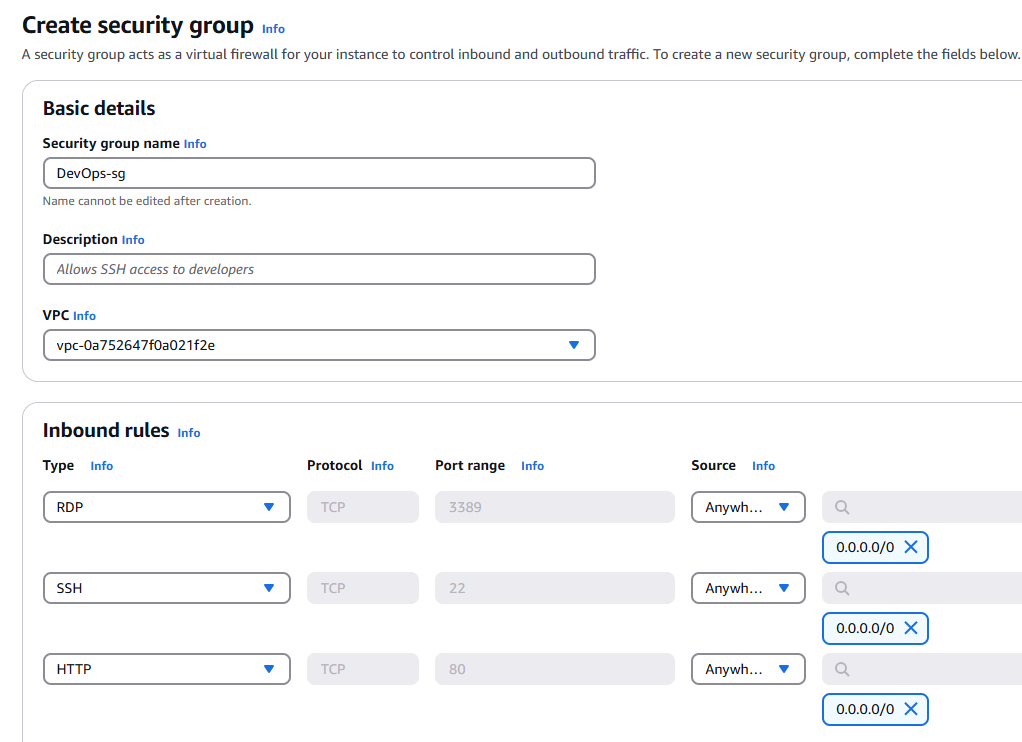
1. Create Key pair (.pem file)
   1. Public key (AWS will keep it)
   2. Private key (We will get)

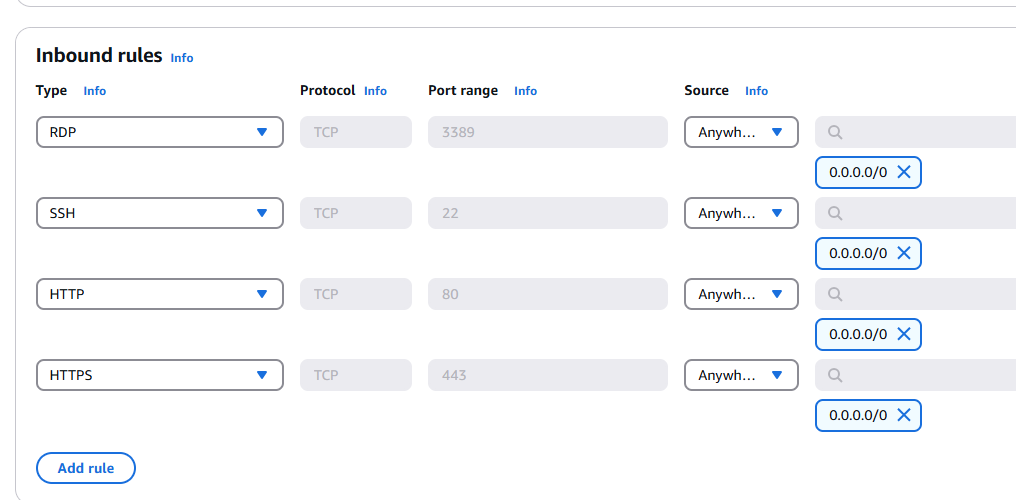




1. Create Security groups & update Inbound rules to allow traffic

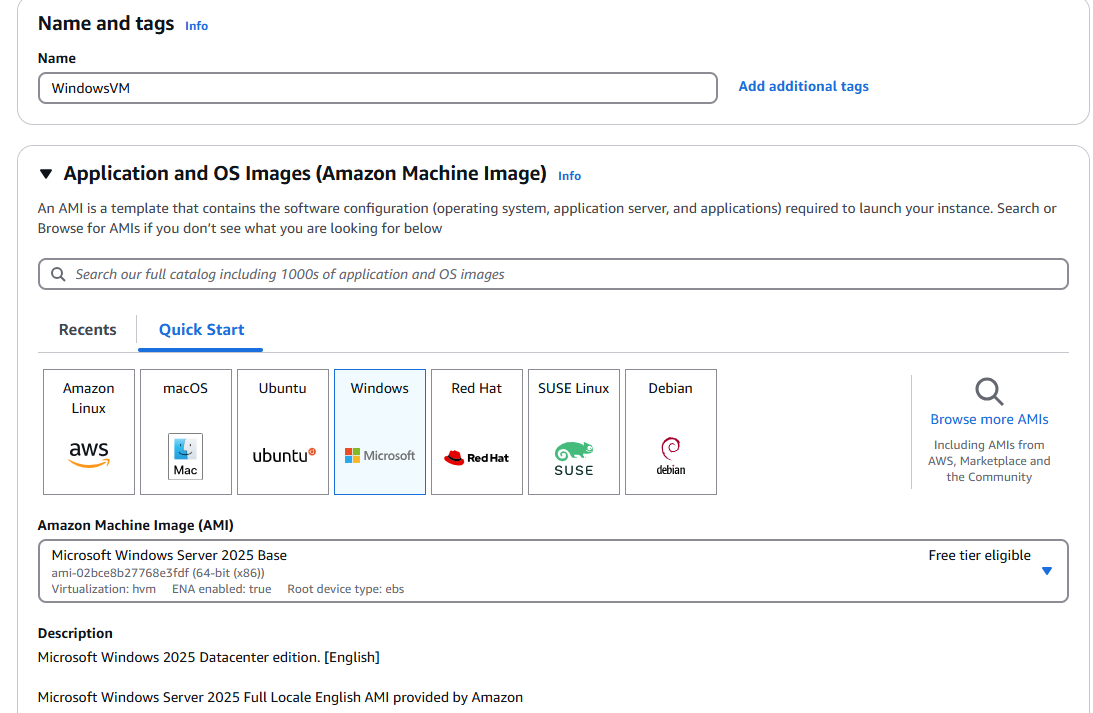


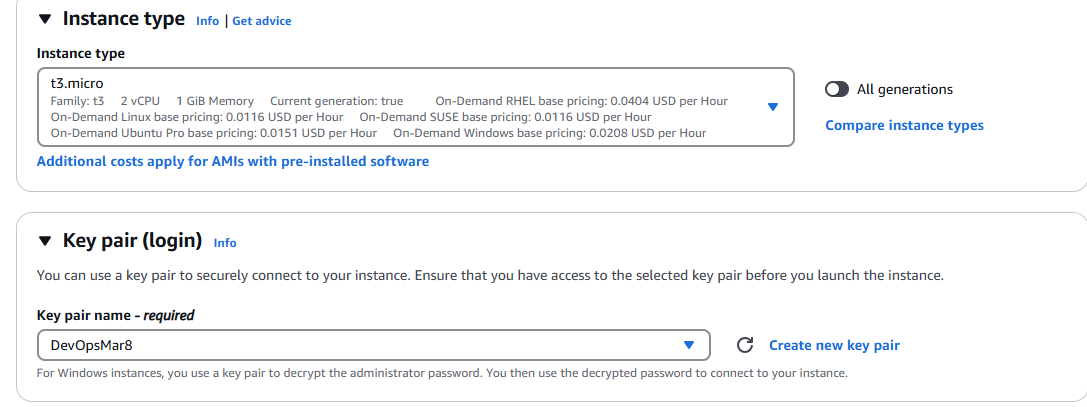


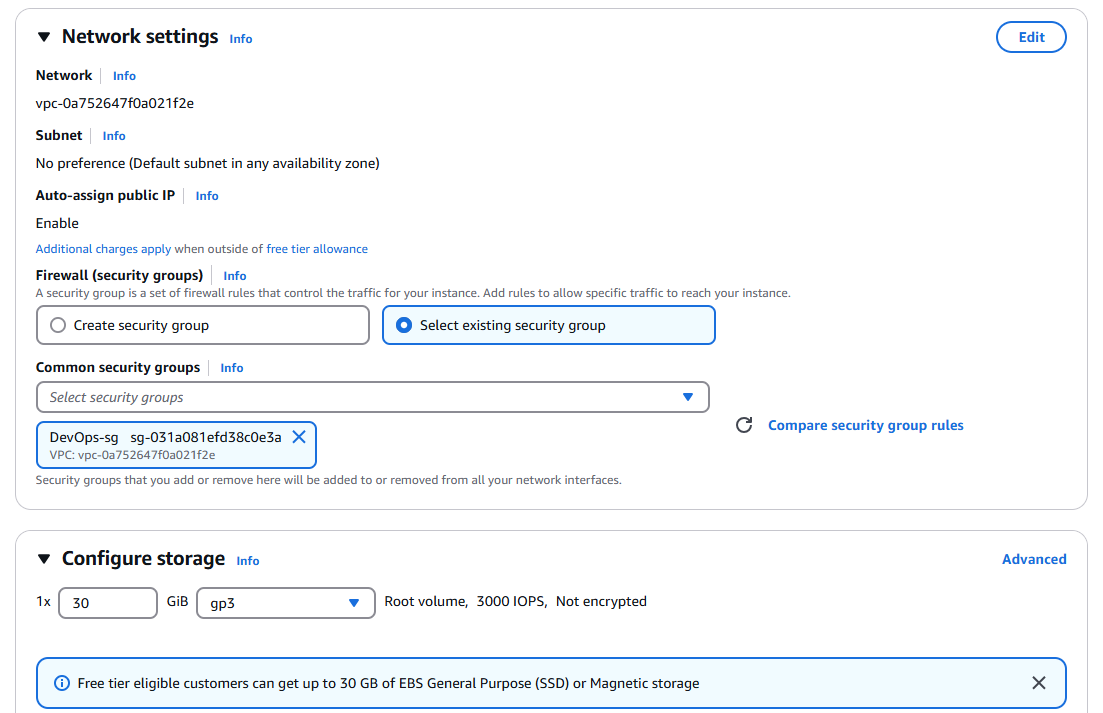


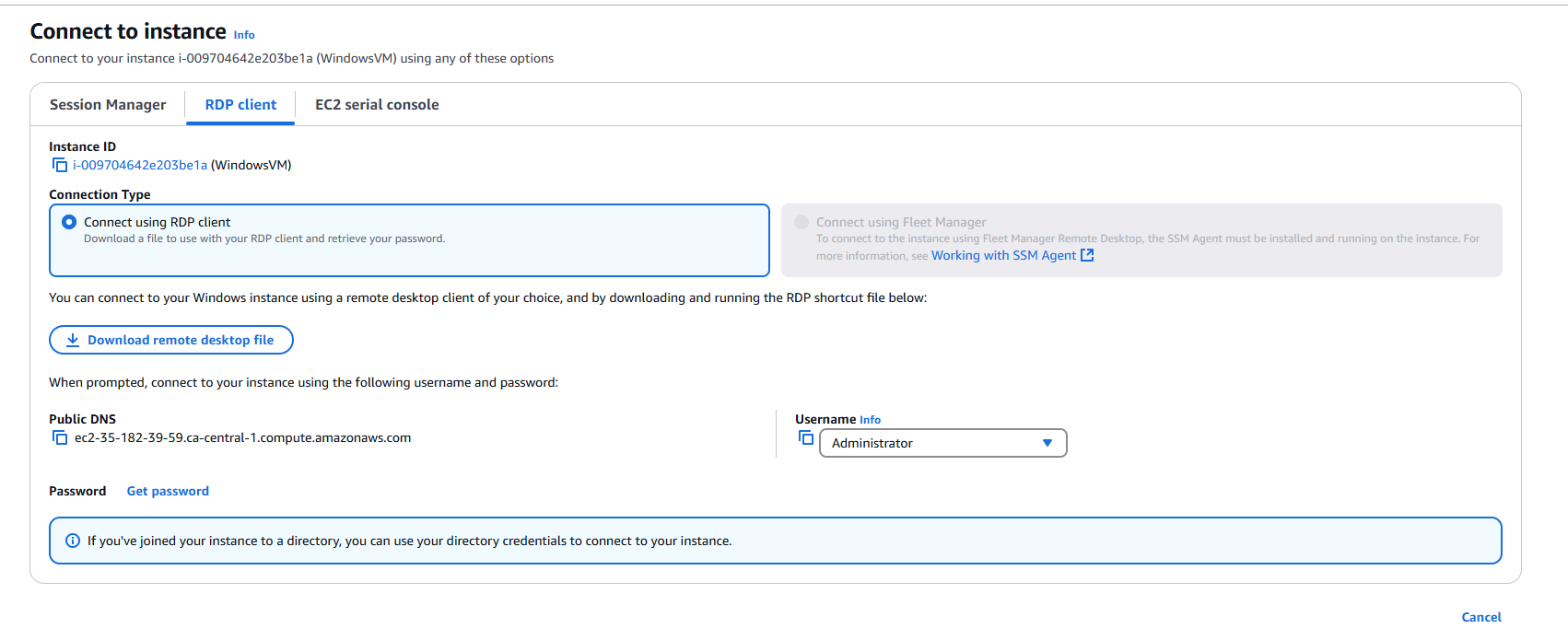


1. Create Windows EC2:

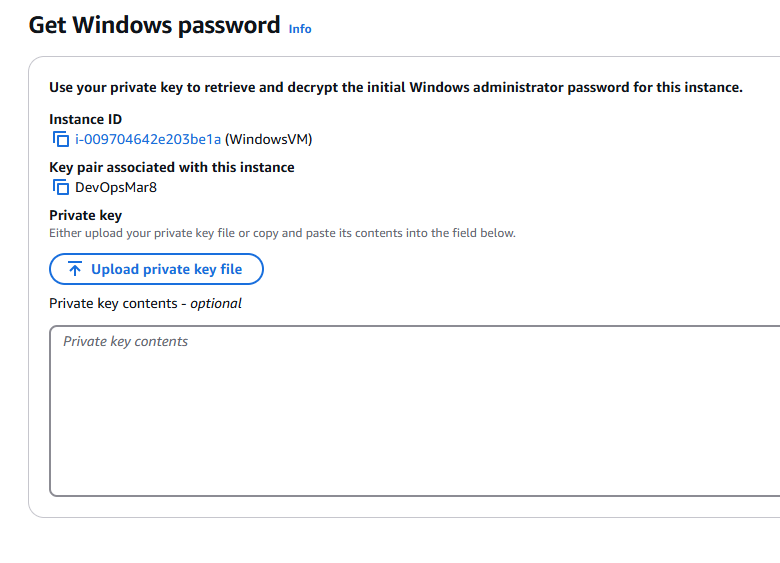




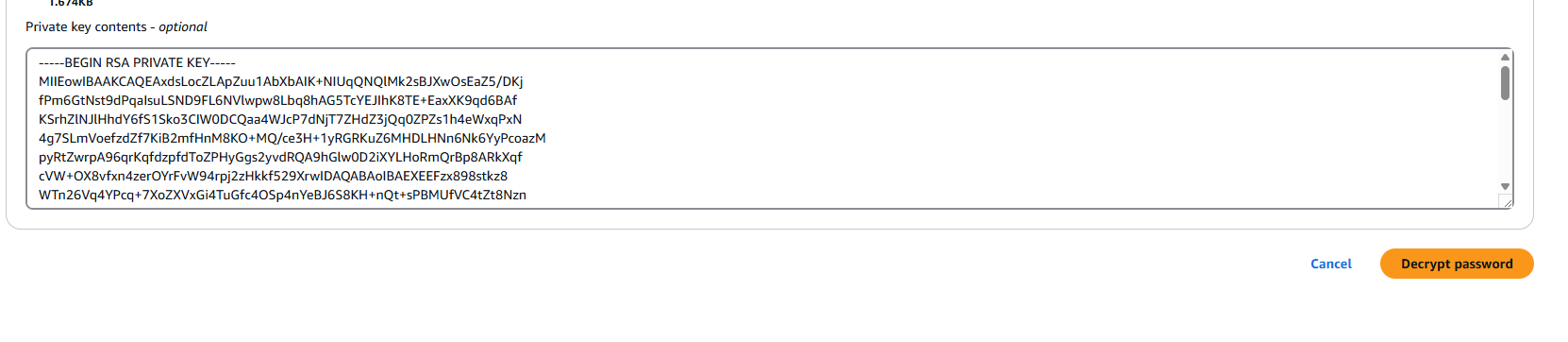




Click on Get Password



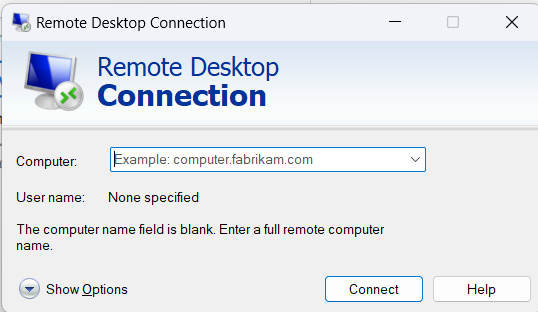
Click on Upload Private Key File

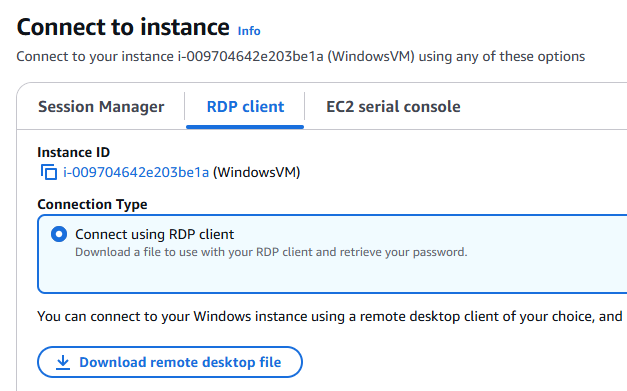


When you click Decrypt password, we can get the password

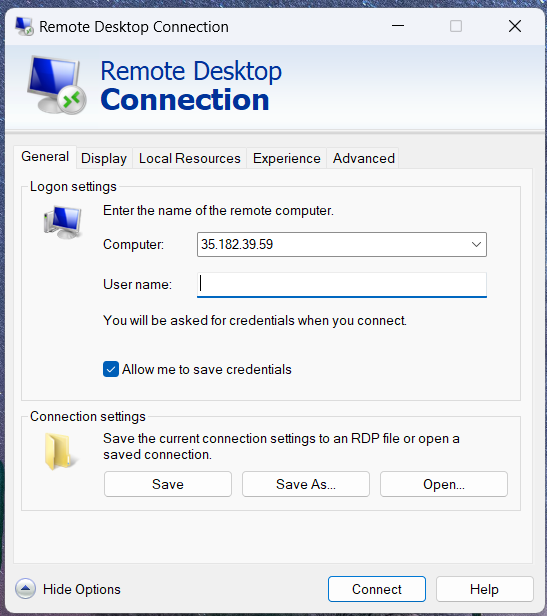
That’s the password for Windows

Open Remote Desktop Connection

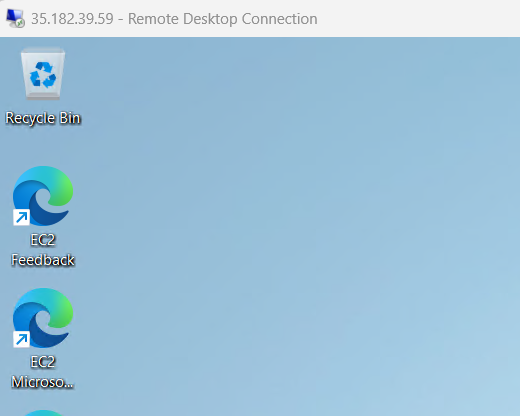




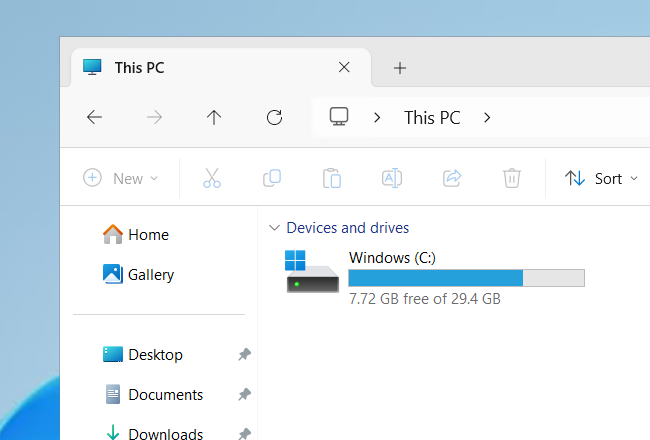
Computer is Public IP from EC2



Click Connect



Totally 30GB memory is available for free - Default EBS storage



Windows RDP: 3389 port#

Linux SSH: 22

HTTP: 80

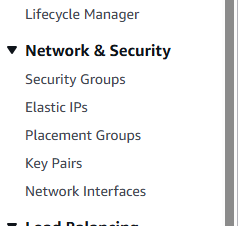
HTTPS: 443

MySQL: 3306

Types of IPs:

Private IP -> it is a fixed IP . this IP will not change. Used for internal communication (within VPC)

Public IP -> Dynamic IP in AWS -> used to connect with EC2 VM from outside world -> whenever we restart our VM then Public IP gets changed. We could have fixed public IPs, we should use Elastic IP address -> Allocate Elastic IP address -> it is paid service



Action -> Associate Elastic IP address -> Instance (where you want to associate) -> select an instance -> Paste Private IP address -> click Associate

Go back to Instances -> we should see Elastic IP option as well now

Now if you stop instance and restart instance, the Elastic IP wont change

Paid service: first Disassociate Elastic IP -> Actions -> Release Elastic IP address -> Release. No Elastic IP found in the region

Elastic IP -> If we want fixed IP -> Elastic IPs are commercial (bills will be generated)

Lab Practicals on Elastic IP:

1. Allocate Elastic IP from AWS
2. Associate Elastic IP with EC2 VM
3. Restart and check EC2 public IP and it will remain the same and wont change
4. Deassociate Elastic IP
5. Release Elastic IP to AWS -> to avoid billing

EBS -> Elastic Block Storage

It is a block level storage device (HDD/SSD)

Whenever we create an EC2 by default, EBS will be attached to EC2 instance

In EBS, we have two types of volumes (storage) -> root volume and additional volume

EBS provides both Primary and Secondary storages for EC2 instances

Whenever we launch EC2 instances, by default we will be getting a Root volume

For Windows virtual machines, 30GB

For Linux virtual machines, we get 8GB of default value

Note: EBS volume can be upto 16TB

Root volume is mandatory to launch EC2 instance and if we try to remove root volume from EC2 then we cannot start or use that instance

We can have additional EBS volumes and they are optional (we can add or remove based on need)

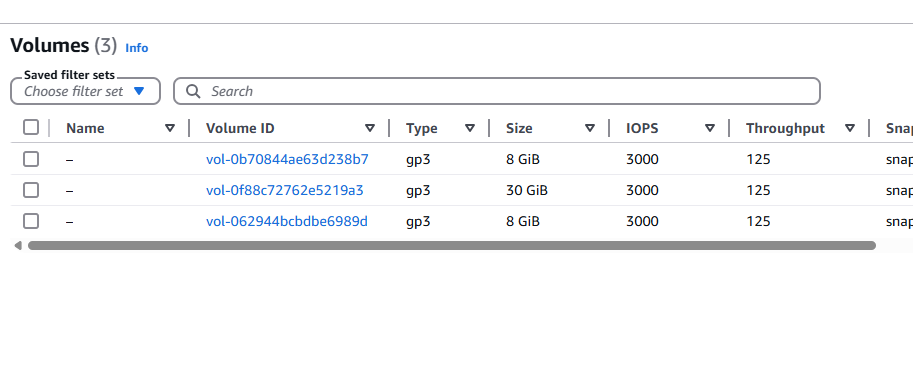
One EC2 VM can have multiple EBS volumes. However, one EBS volume can be attached to one EC2 VM at a time.

Can we create additional EBS volumes and attach to EC2? answer is Yes

Note: EC2 VM availability zone and EBS volume zone must be the same. Availability zone must be the same to attach EBS volume to EC2

EBS volumes are Availability zone specific that’s if an EC2 VM is created in AP-South-1A zone then our EBS volume must also be there in AP-South-1a only then we can attach it

Default volumes for EC2



EBS volume types:

1. General purpose -> minimum 1 GiB and maximum 16384 GiB
2. Provisioned IOPS -> min 4 GiB to 16384 GiB
3. Cold HDD -> Min 125 GiB to 16384 GiB
4. Throughout optimized -> Min 125 GiB to 16384 GiB
5. Magnetic -> Min 1 GiB to Max 1024 GiB

EBS & Snapshot -> VM1 and VM2. Lets attach an additional volume to VM1 take that and attach to VM2.

“lsblk” command to check volumes in EC2

[ec2-user@ip-172-31-13-80 ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

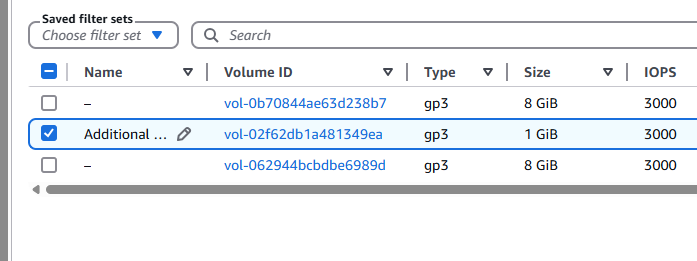
xvda 202:0 0 8G 0 disk

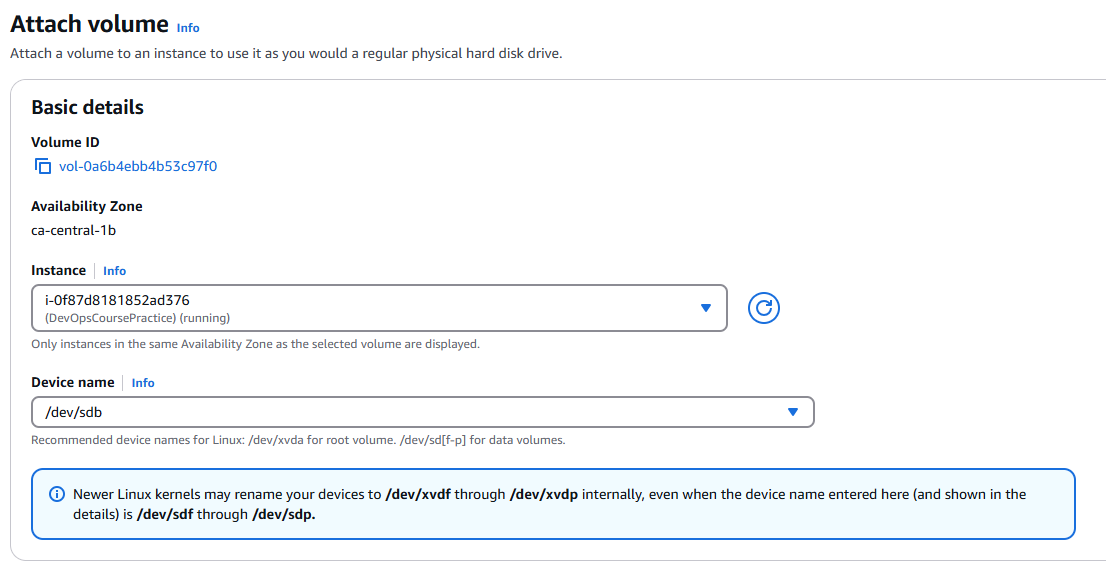
├─xvda1 202:1 0 8G 0 part /

├─xvda127 259:0 0 1M 0 part

└─xvda128 259:1 0 10M 0 part /boot/efi

Lets add additional 1GiB volume





We could create an EBS volume of 10GB attach to a VM1 then download data, detach this additional volume from VM1 then attach to another VM2. copy the data into VM2. it is like attaching and transferring data via pen drive

Mounting: attaching additional volume with VM1

[ec2-user@ip-172-31-13-80 ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

xvda 202:0 0 8G 0 disk

├─xvda1 202:1 0 8G 0 part /

├─xvda127 259:0 0 1M 0 part

└─xvda128 259:1 0 10M 0 part /boot/efi

[ec2-user@ip-172-31-13-80 ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

xvda 202:0 0 8G 0 disk

├─xvda1 202:1 0 8G 0 part /

├─xvda127 259:0 0 1M 0 part

└─xvda128 259:1 0 10M 0 part /boot/efi

xvdb 202:16 0 1G 0 disk

“sudo mkfs -t ext4 /dev/xvdb”

[ec2-user@ip-172-31-13-80 ~]$ sudo mkfs -t ext4 /dev/xvdb

mke2fs 1.46.5 (30-Dec-2021)

Creating filesystem with 262144 4k blocks and 65536 inodes

Filesystem UUID: 524d9f3d-018f-4e72-8c32-73ada2009eda

Superblock backups stored on blocks:

32768, 98304, 163840, 229376

Allocating group tables: done

Writing inode tables: done

Creating journal (8192 blocks): done

Writing superblocks and filesystem accounting information: done

Create directory after mounting

mkdir dir1

sudo mount /dev/xvdb dir1

Mounting

[ec2-user@ip-172-31-13-80 ~]$ sudo mount /dev/xvdb dir1

[ec2-user@ip-172-31-13-80 ~]$

ls -l

[ec2-user@ip-172-31-13-80 ~]$ ls -l

total 128

-r--------. 1 ec2-user ec2-user 1674 Feb 22 21:59 devops\_linux.pem

drwxr-xr-x. 3 root root 4096 Mar 8 21:40 dir1

[ec2-user@ip-172-31-13-80 ~]$ cd dir1

[ec2-user@ip-172-31-13-80 dir1]$

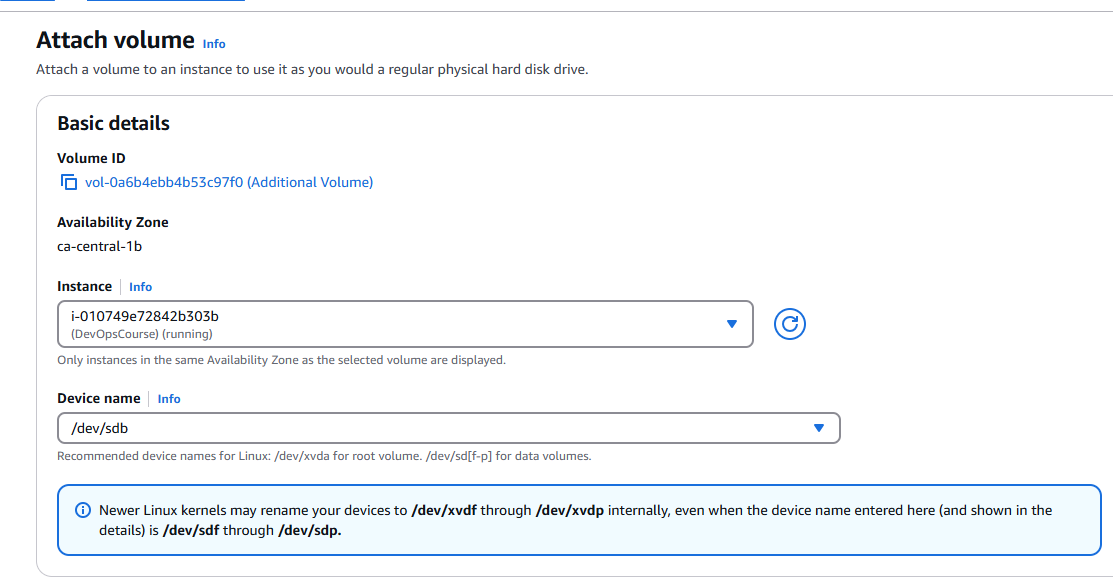
[ec2-user@ip-172-31-13-80 dir1]$

[ec2-user@ip-172-31-13-80 dir1]$ sudo touch f1.txt f2.txt

I detach the Additional Volume from VM1

Now I log into VM2

Attach the same Additional Volume to VM2



We can see xvdb in VM2

[ec2-user@ip-172-31-6-153 ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS

xvda 202:0 0 8G 0 disk

├─xvda1 202:1 0 8G 0 part /

├─xvda127 259:0 0 1M 0 part

└─xvda128 259:1 0 10M 0 part /boot/efi

xvdb 202:16 0 1G 0 disk

[ec2-user@ip-172-31-6-153 ~]$ sudo mkfs -t ext4 /dev/xvdb

mke2fs 1.46.5 (30-Dec-2021)

/dev/xvdb contains a ext4 file system

last mounted on /home/ec2-user/dir1 on Sat Mar 8 21:40:54 2025

Proceed anyway? (y,N) y

Creating filesystem with 262144 4k blocks and 65536 inodes

Filesystem UUID: d9d2e351-d492-4761-97d7-dea4833745cb

Superblock backups stored on blocks:

32768, 98304, 163840, 229376

Allocating group tables: done

Writing inode tables: done

Creating journal (8192 blocks): done

Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-6-153 ~]$ ls -l

total 8

-rw-r--r--. 1 ec2-user ec2-user 52 Feb 22 22:13 alien.txt

-rw-r--r--. 1 ec2-user ec2-user 25 Feb 23 02:54 first-script.sh

[ec2-user@ip-172-31-6-153 ~]$ mkdir dir1

[ec2-user@ip-172-31-6-153 ~]$ cd dir1

Mounting

sudo mount /dev/xvdb dir1

We see f1.txt and f2.txt in VM2. those files were originally created in VM1

How to copy files from VM zone ‘a’ to VM in zone ‘b’?

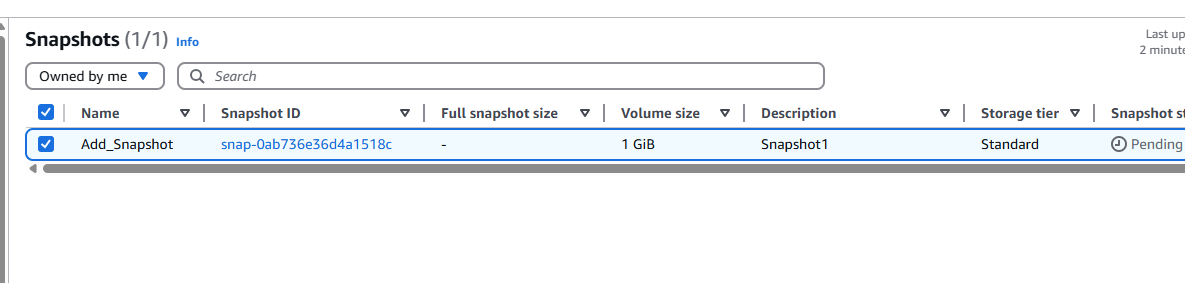
Concept of Snapshots come into picture. Snapshots are backups for volumes and they are regional specific (Volumes are zone specific).

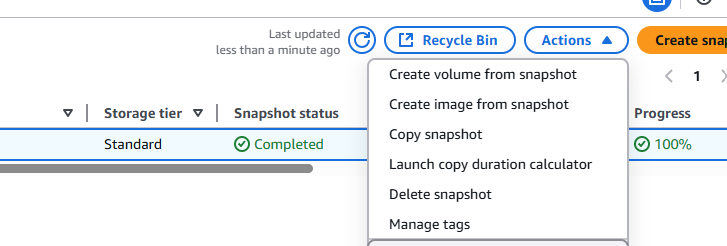
From a Volume, we can create a Snapshot (backup of volume), and from Snapshot again we can create a Volume

Volume -> Snapshot -> Volume

1a Volume -> Snapshot -> 1b Volume -> Attach to VM in zone b

Snapshots cannot be attached to EC2 Instance directly (as seen before Volumes can be attached but not Snapshots)





How to know how many Volumes are there?

Created EC2 VM -> EBS root volume 8GB

Created an Additional Volume with 10GB (Check AZ)

VM in 1a AZ then Volume also in 1a AZ

Attach additional volumes to existing VM as shown in live class

Connect to VM from MobaXTerm:

lsblk -> Check volumes attached

Store files into EBS additional volume and follow commands

sudo mkfs -t ext4/dev/xvdb

mkdir dir1

Mounting is establishing relation between additional volume and directory

sudo mount /dev/xvdb dir1 -> dev/xvdb is the name we have selected at the time of volume creation

cd dir1 -> change dir to our created directory dir1

sudo touch f1.txt f2.txt and add into dir1

Detached the additional volume from EC2 VM1

Create new EC2 VM and attach EBS additional volume to EC2 new VM

Check files by running commands

lsblk

mkdir dir1

sudo mount /dev/xvdb dir1

ls -l dir1

NOTE: make sure after practice detach additional volumes and delete to to avoid billing