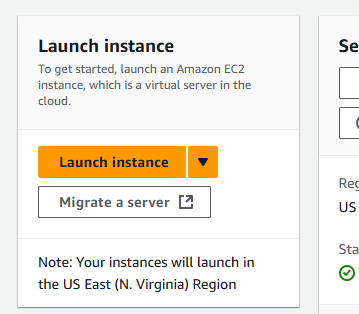
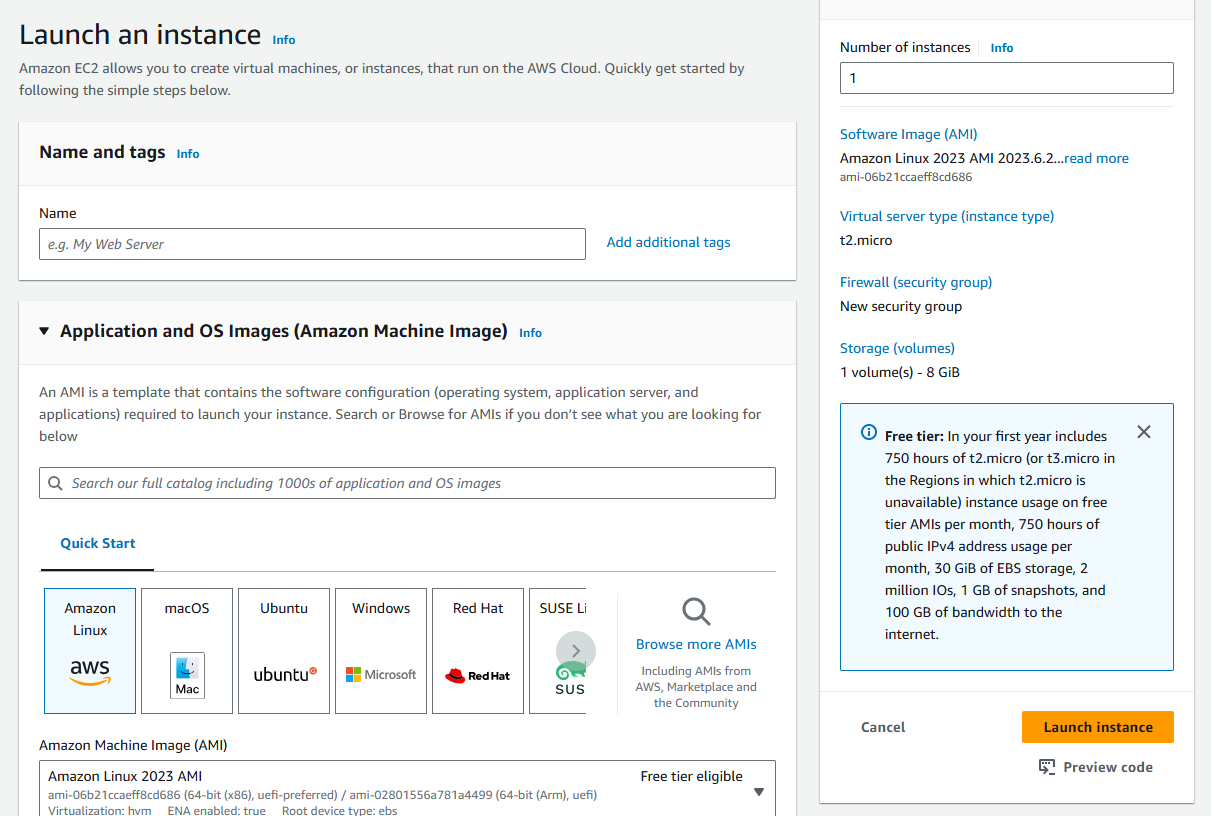
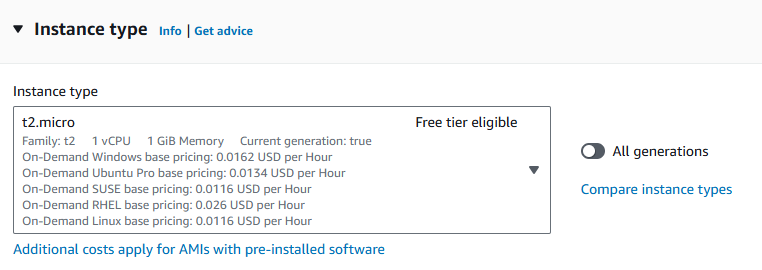
To create an instance in EC2



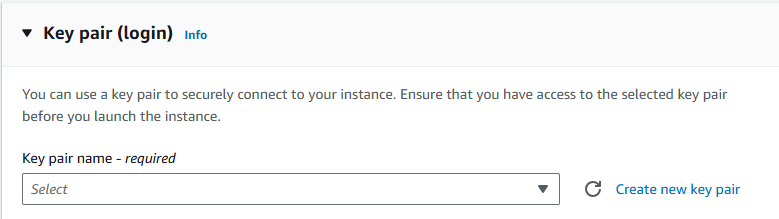


On this

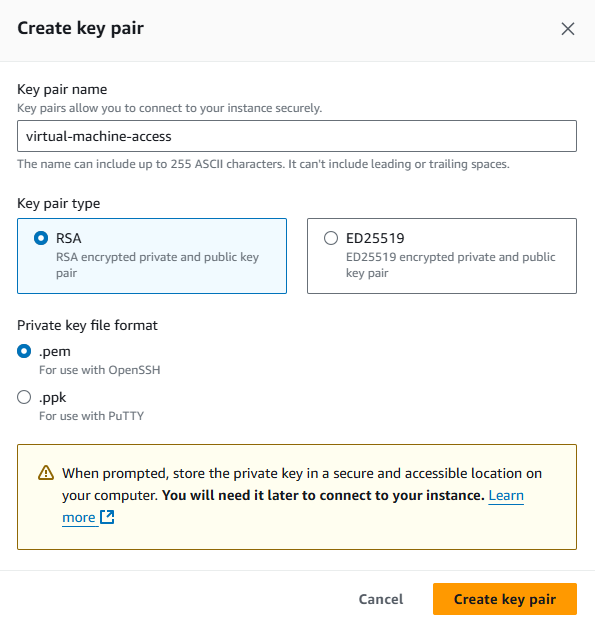
1. We need to provide a meaningful name
2. AMI (Application machine image) – the OS we want to use in our application



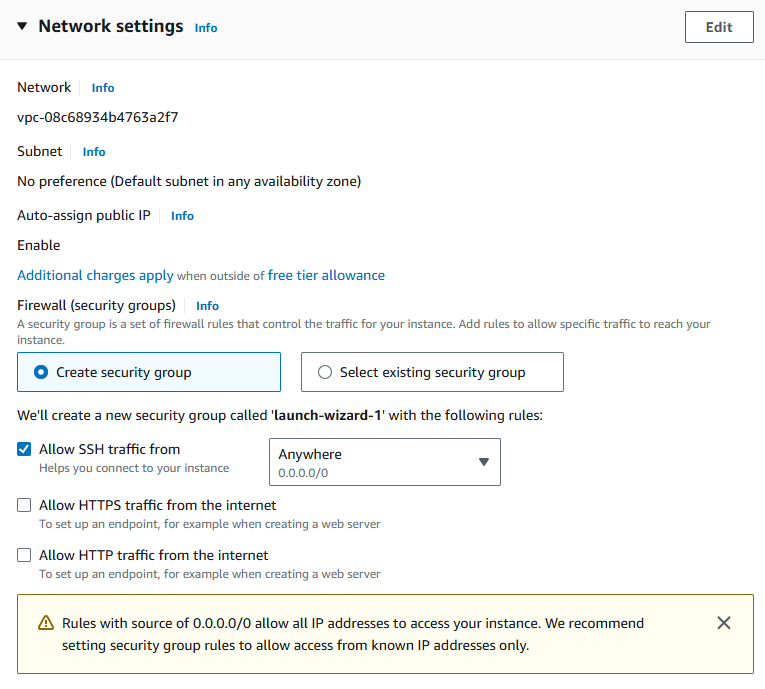
Instance type: Its configuration required



Key Pair: For authentication

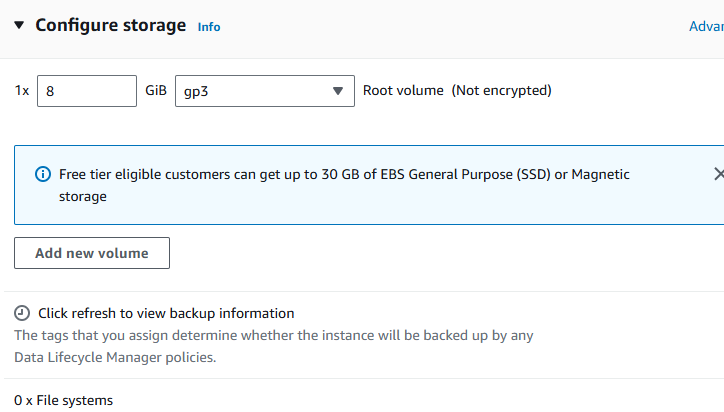


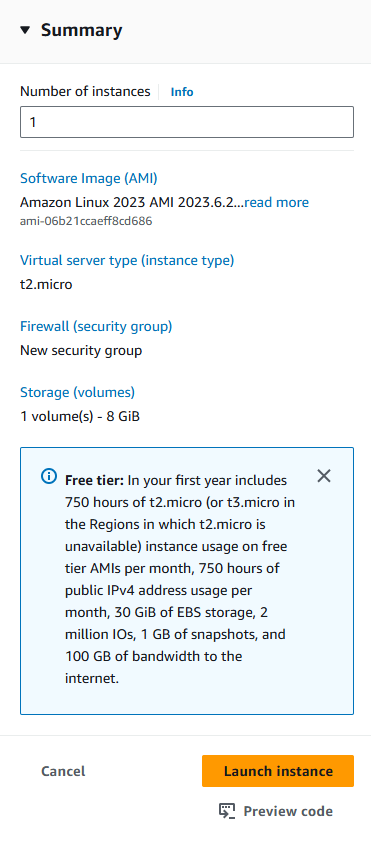
As soon as we create, a file will be downloaded



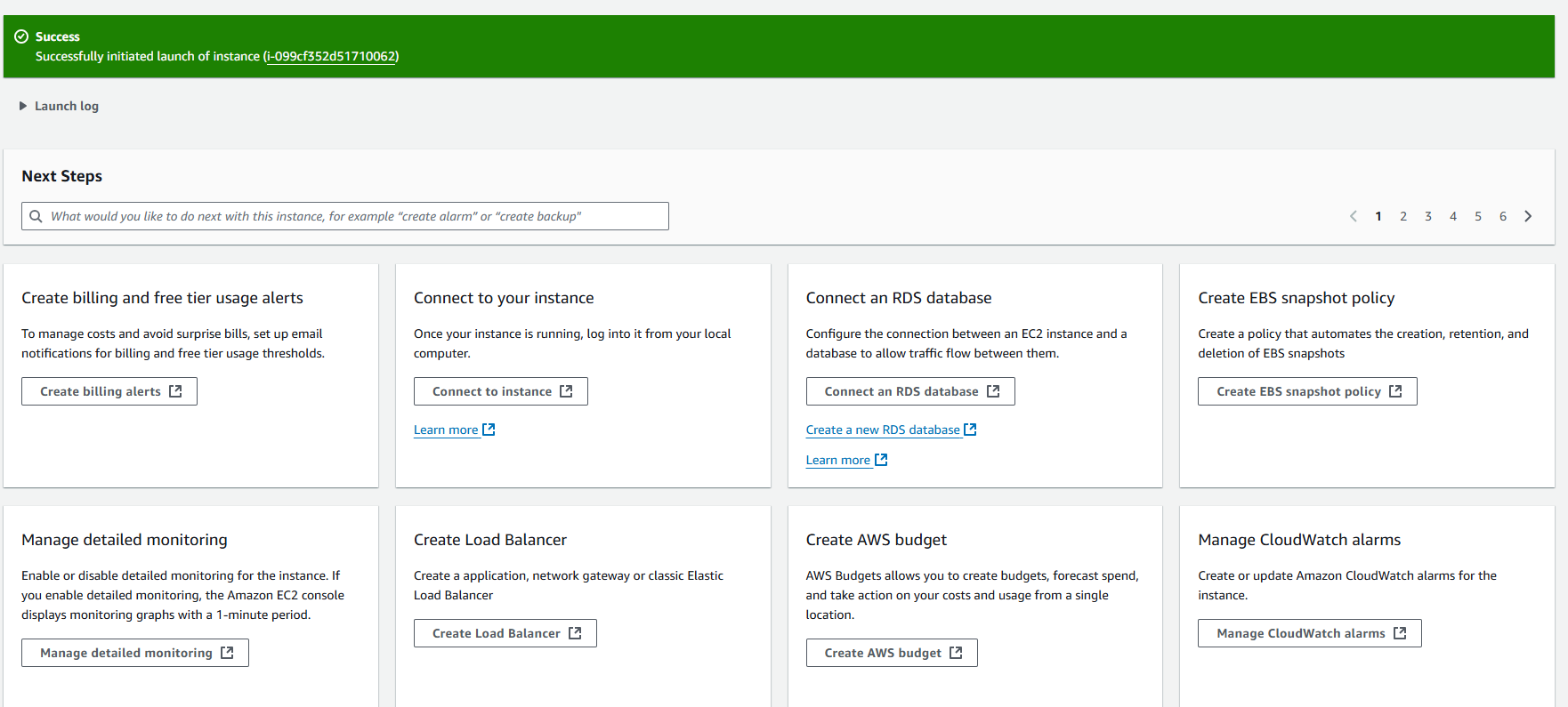
For more security

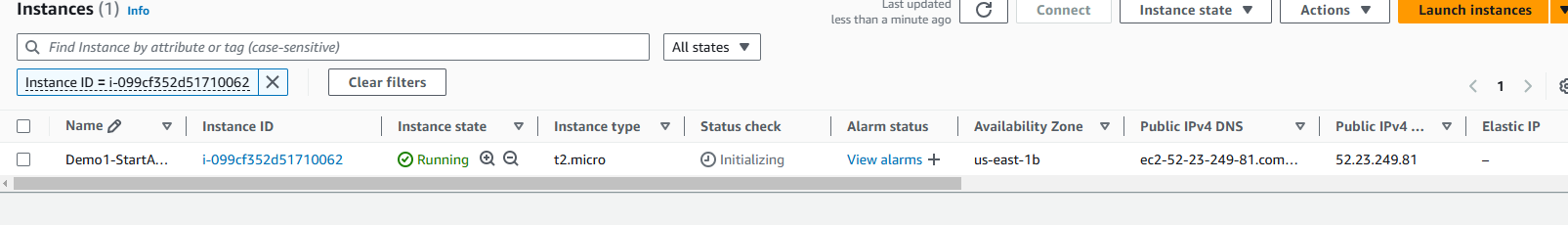
SSH stands for Secure Shell. It is a cryptographic network protocol used to securely access and manage devices over an unsecured network. SSH provides a secure channel for sending commands and transferring data, ensuring confidentiality and integrity through encryption and authentication mechanisms.



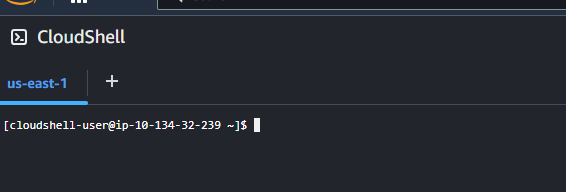


By default, no off instance is 1, But it can be increased or decreased according to the usage

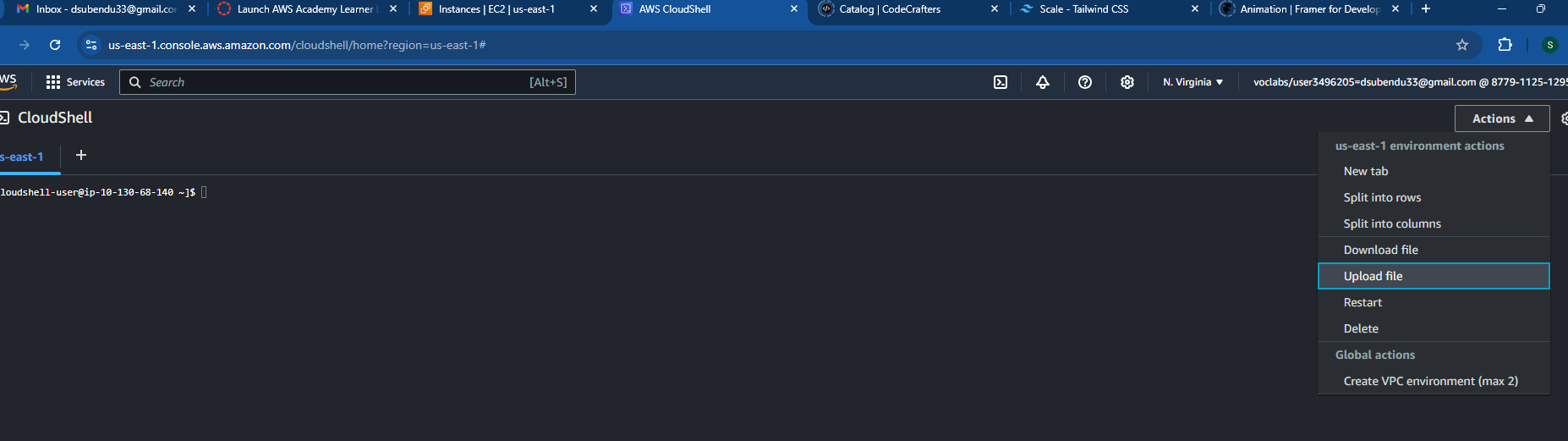




Now to connect with this



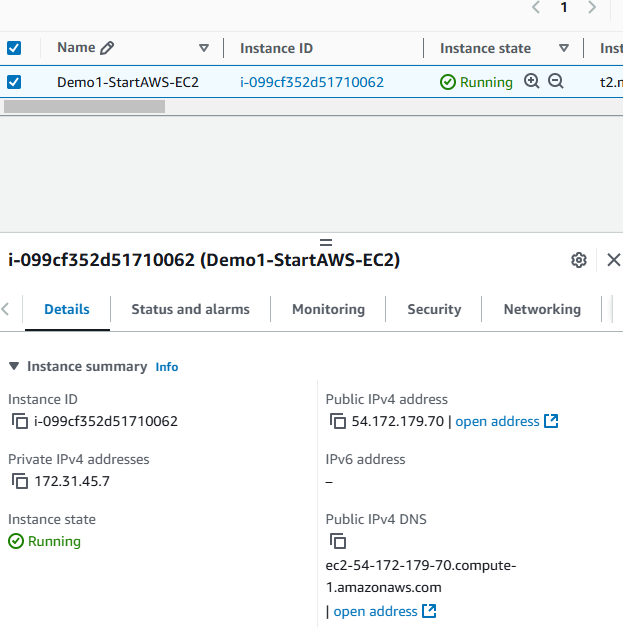
Open cloud shell to connect to the instance



Upload your key pair file



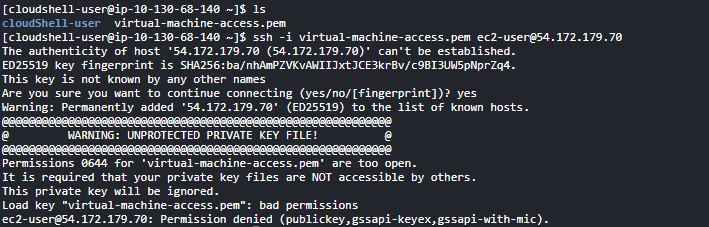
The file will be uploaded in present working directory



Get public address



ec2-user is default user name



The key file shouldn’t be in executable or write state and it mustn’t be read by any other user

In Linux, there are 3 users

1. **Current user**
2. **Group user**
3. **Others**

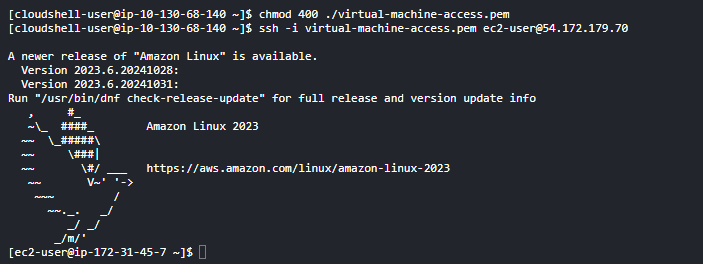
**For file, we have Read (4), Write (2) & Execute (1) permissions**

**7: 4 2 1**

If for some file, 777: all users have all access to the file

If for some file, 444: all users have read access to the file

If for some file, 400: current user has read permission while other users don’t have any



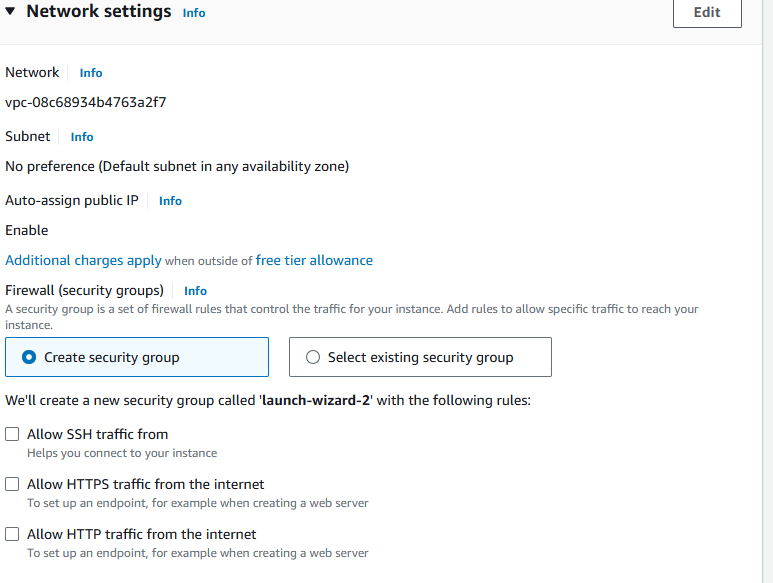
The Ip address has been attached with the terminal line now.

And now we landed to home directory of ec2-user.

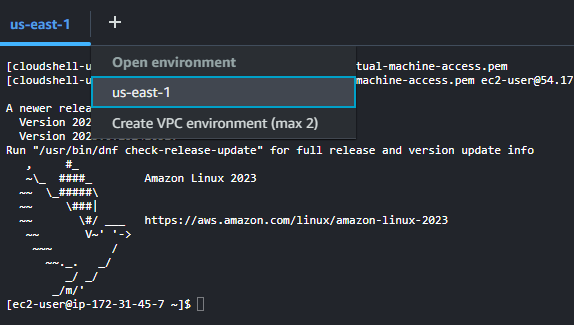
After these we can do anything on this instance.

You can create now as many instances as you want by this process.

When



From network setting, if nothing is chosen



  
  
Now if we try to connect it won’t happen, as there is no ssh permission

It will throw after sometime

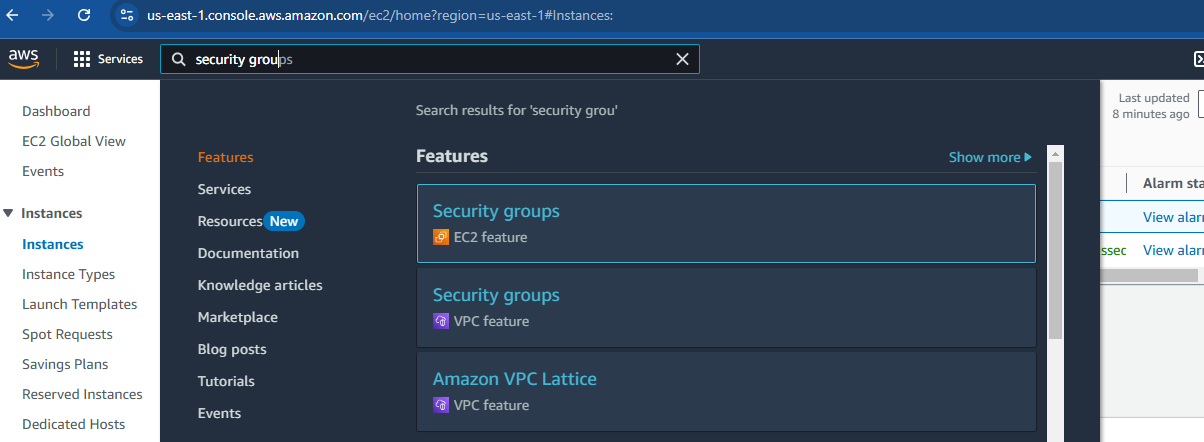


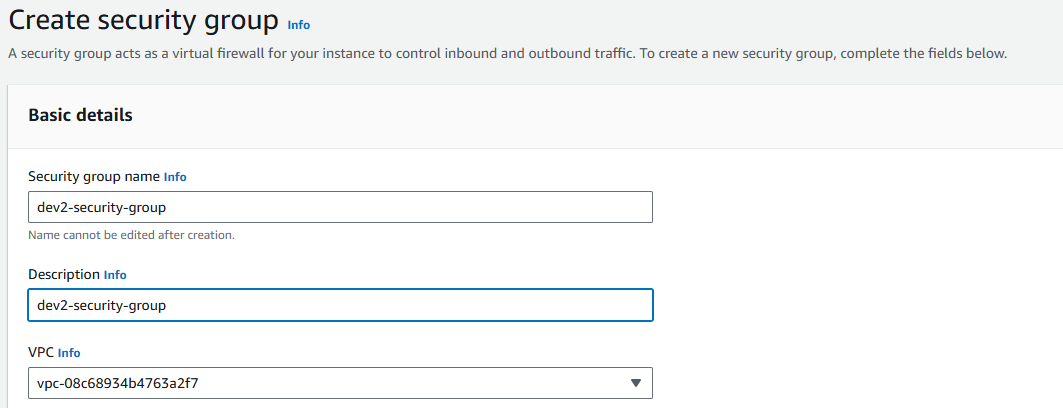
**Inbound connection:** Client tries to connect with EC2 instance

**Outbound connection:** Instance tries to connect with outer Api’s

We need mechanism where we can perform both.

These can be done through Security Groups.





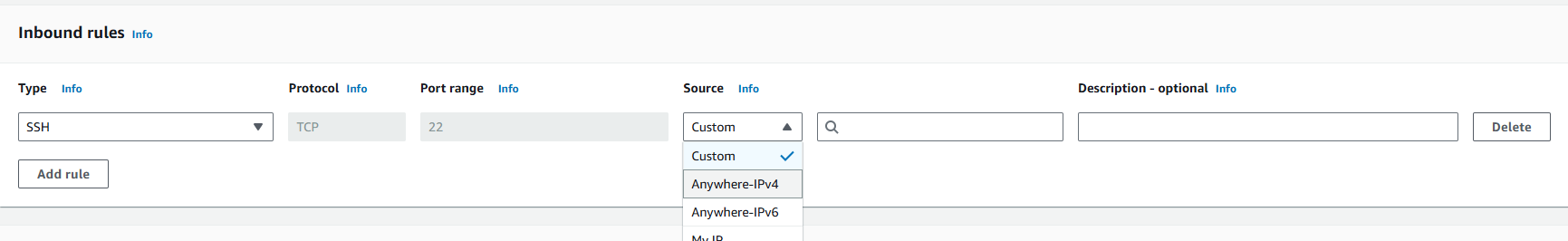
For http: default port is 80

For ssh: default port is 22

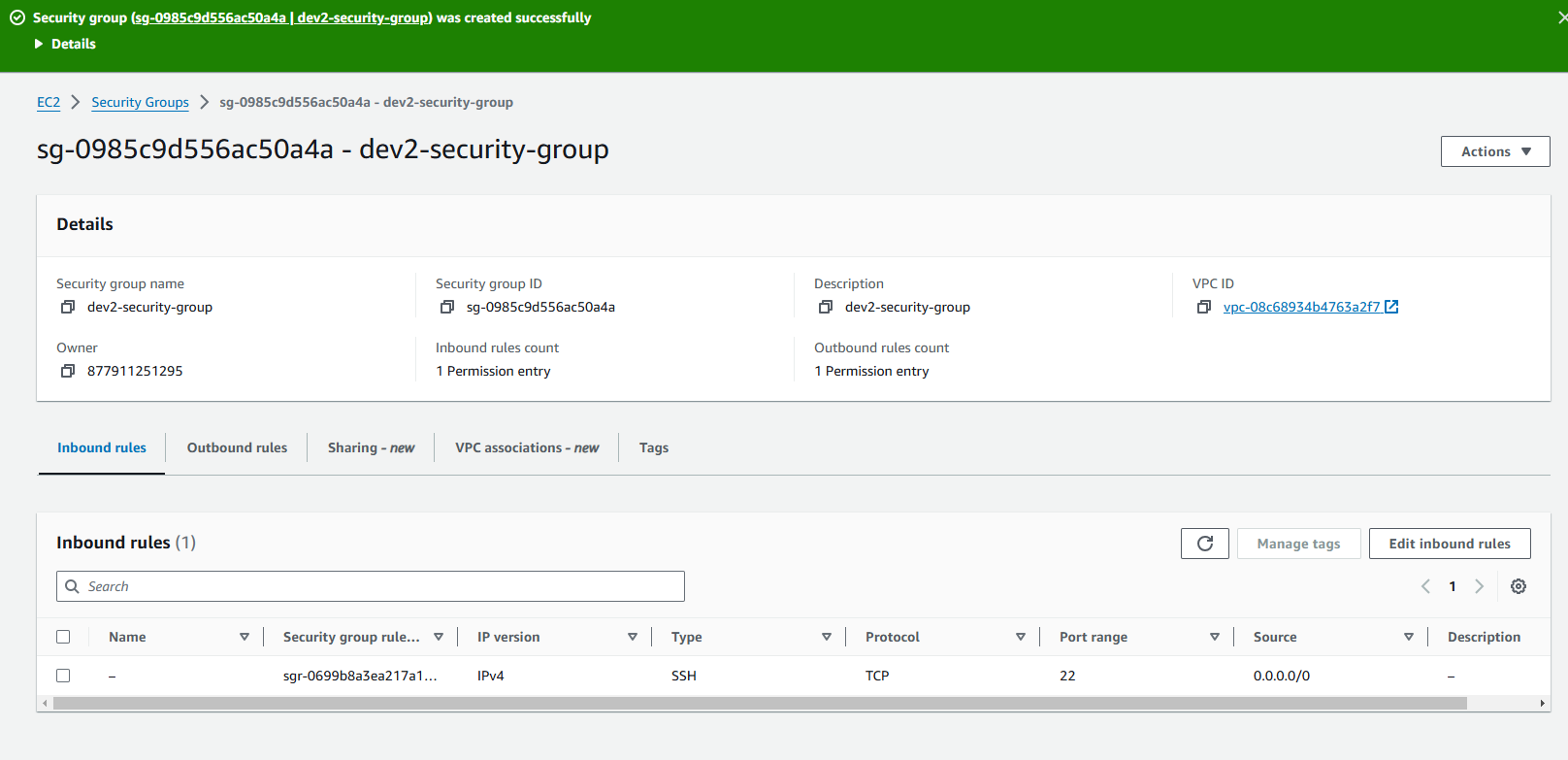
For https: default port is 443

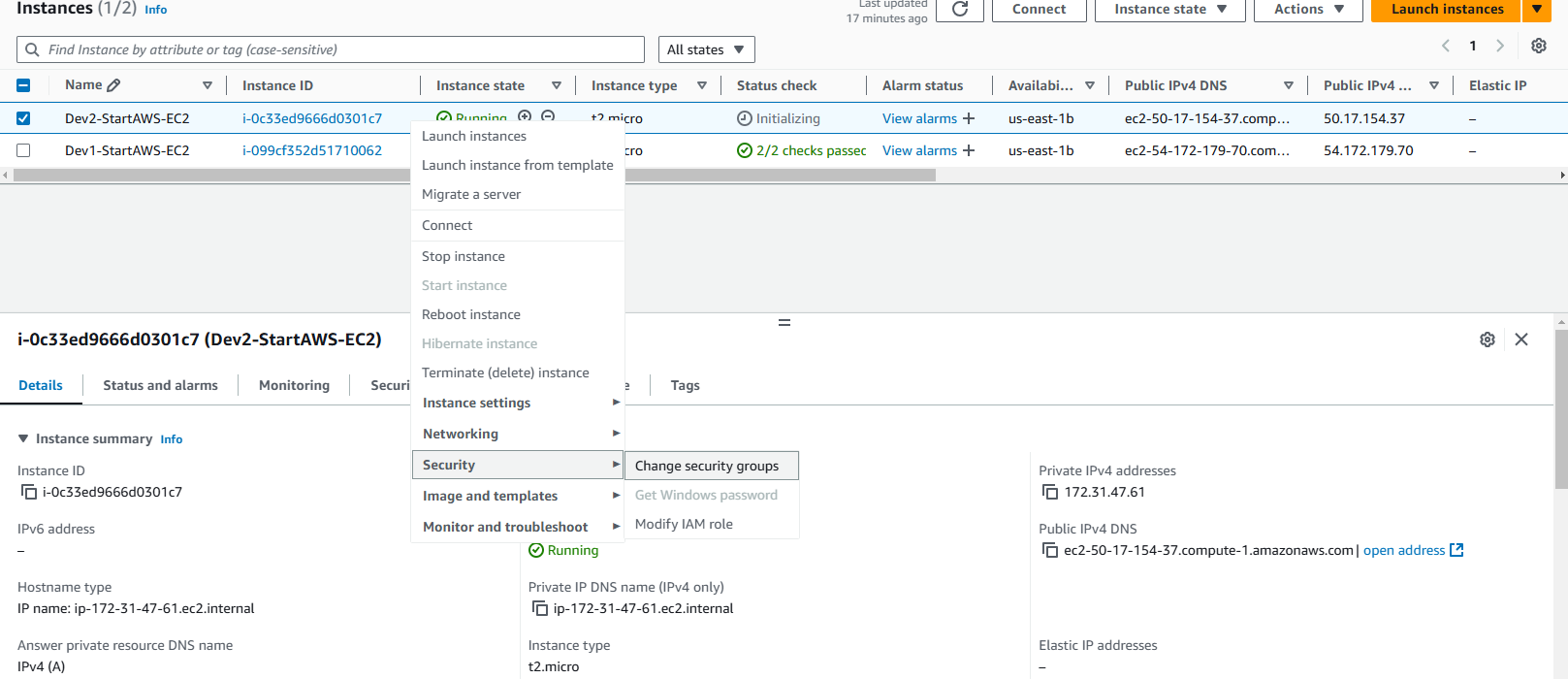
For MySQL: default port is 3306

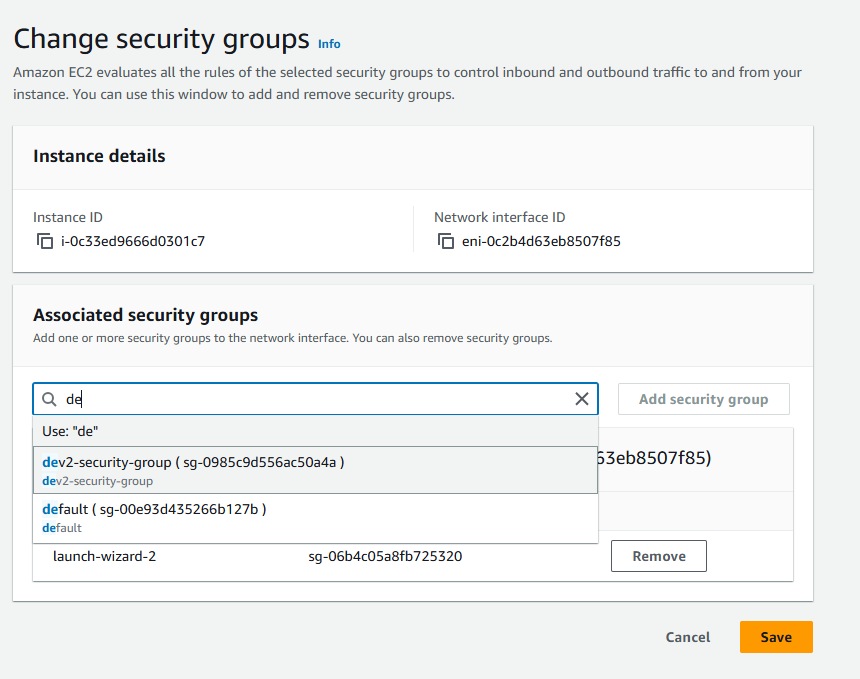
For MongoDB: default port is 27017

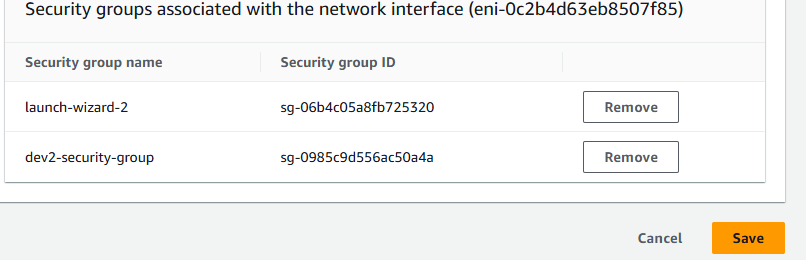


For source: choose ipv4 => so that anyone can connect with the key file

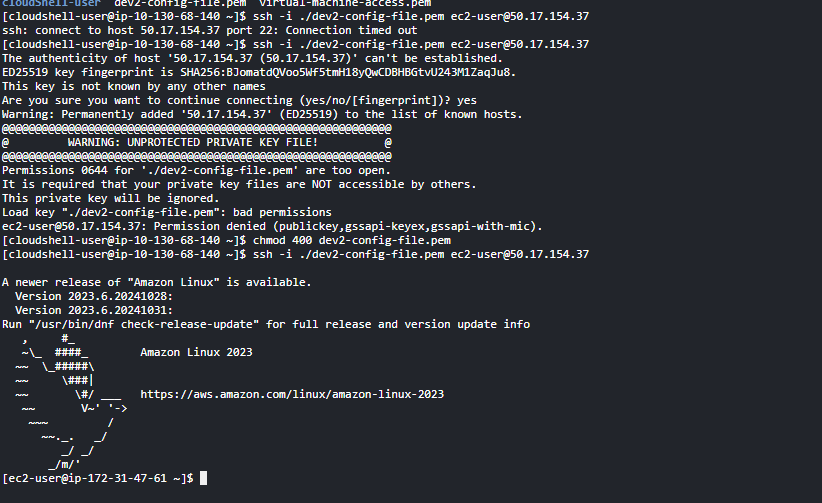
  
Now we need to attach these to our instance required



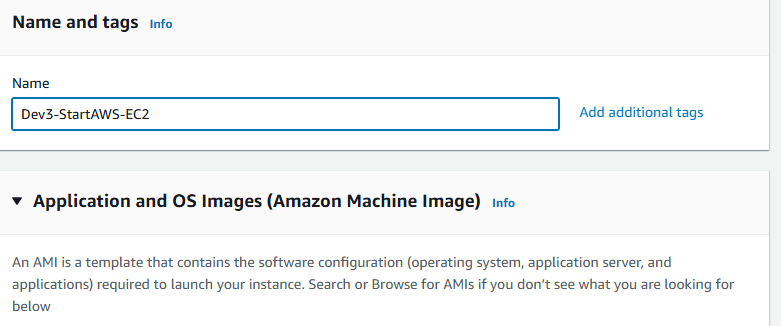




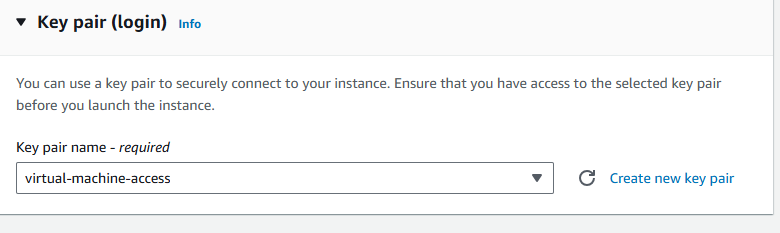
Then save

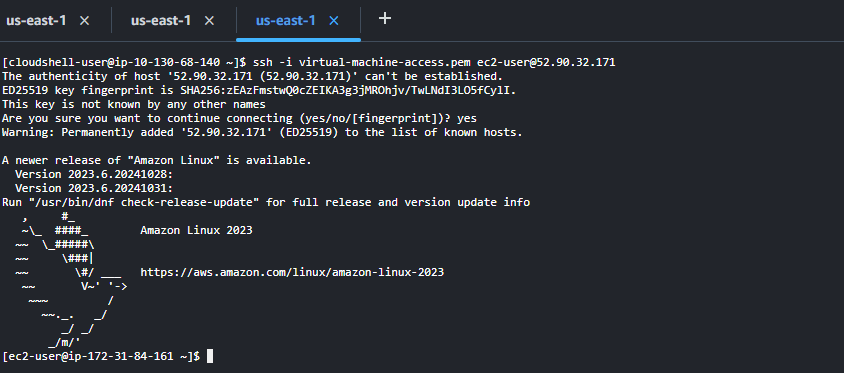


Now, when we want to use same key pair



Using key pair used by 1st dev

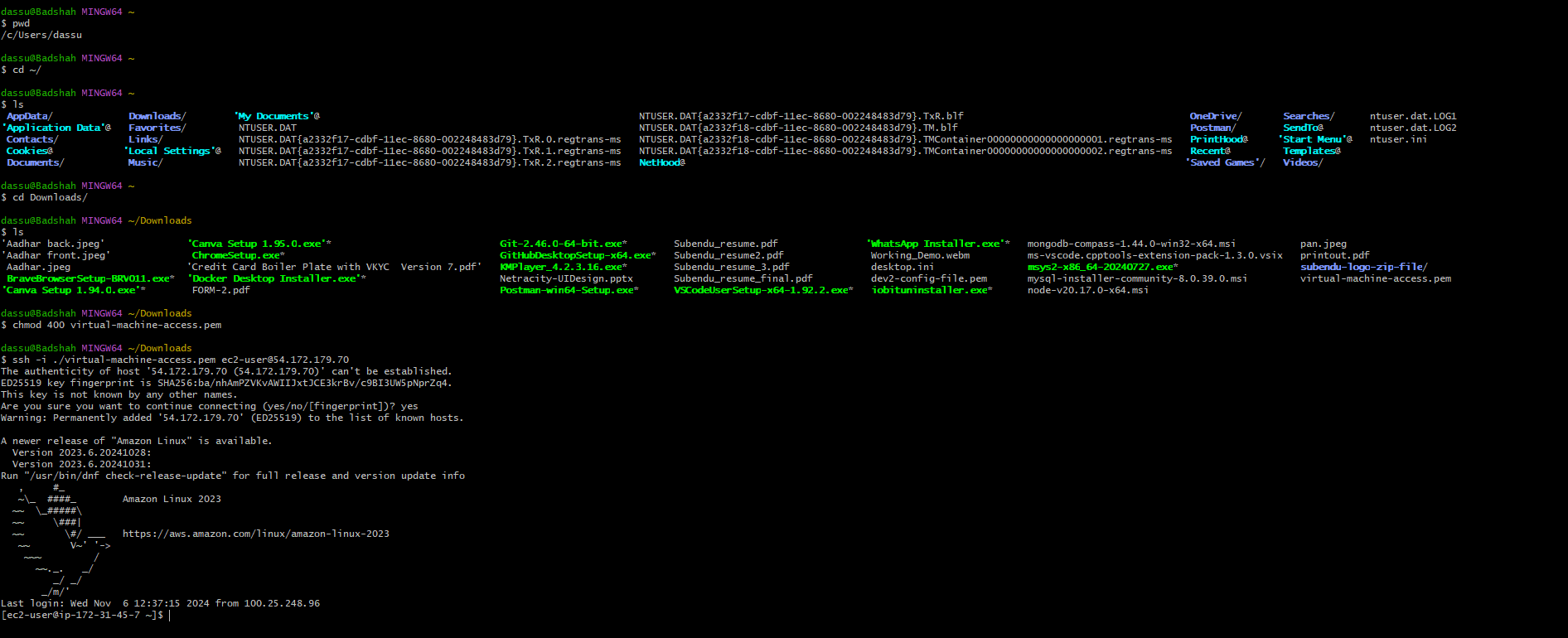




**If we remove security group, the effect will take immediately.**

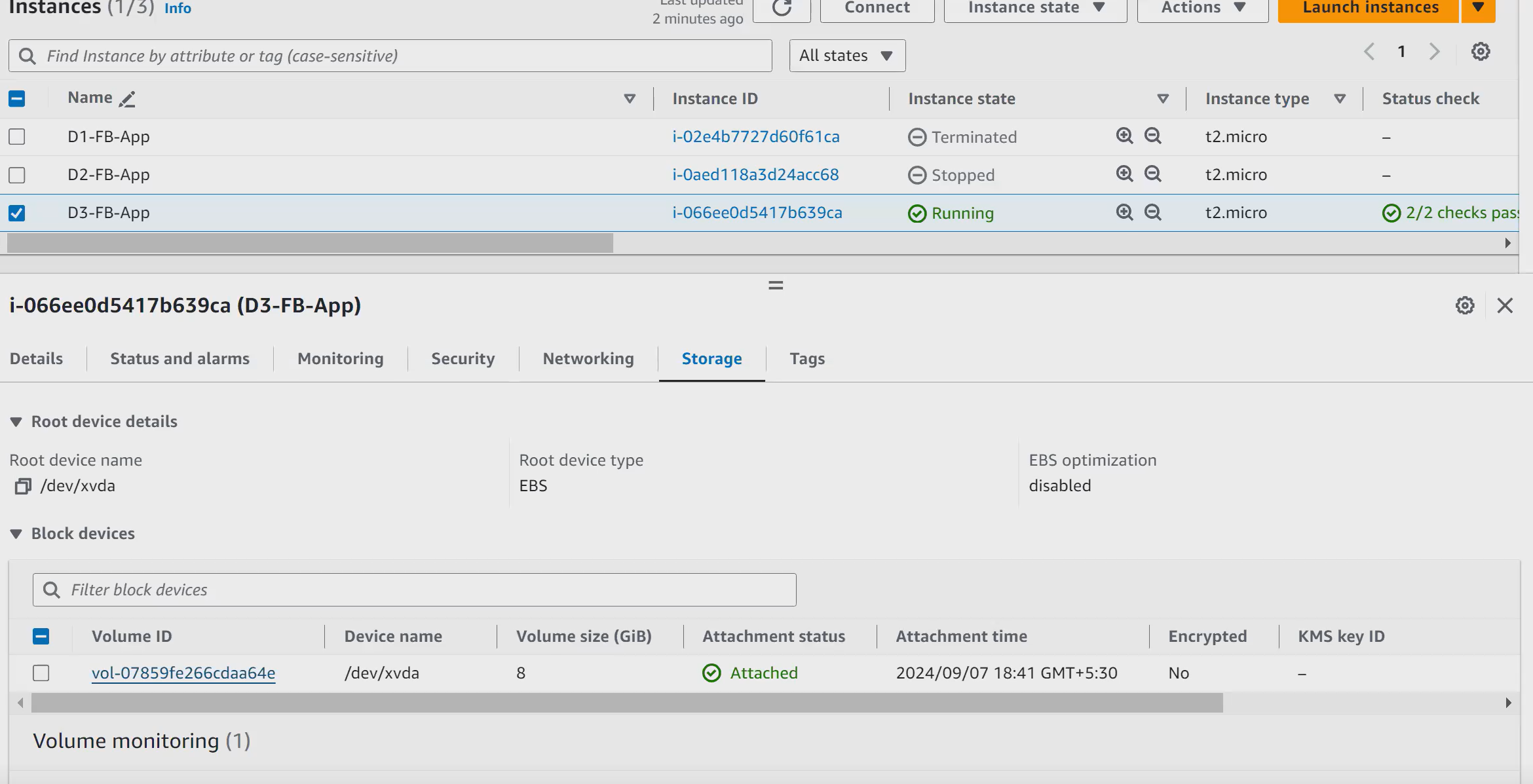
**The user working on the instance will get immediate changes.**

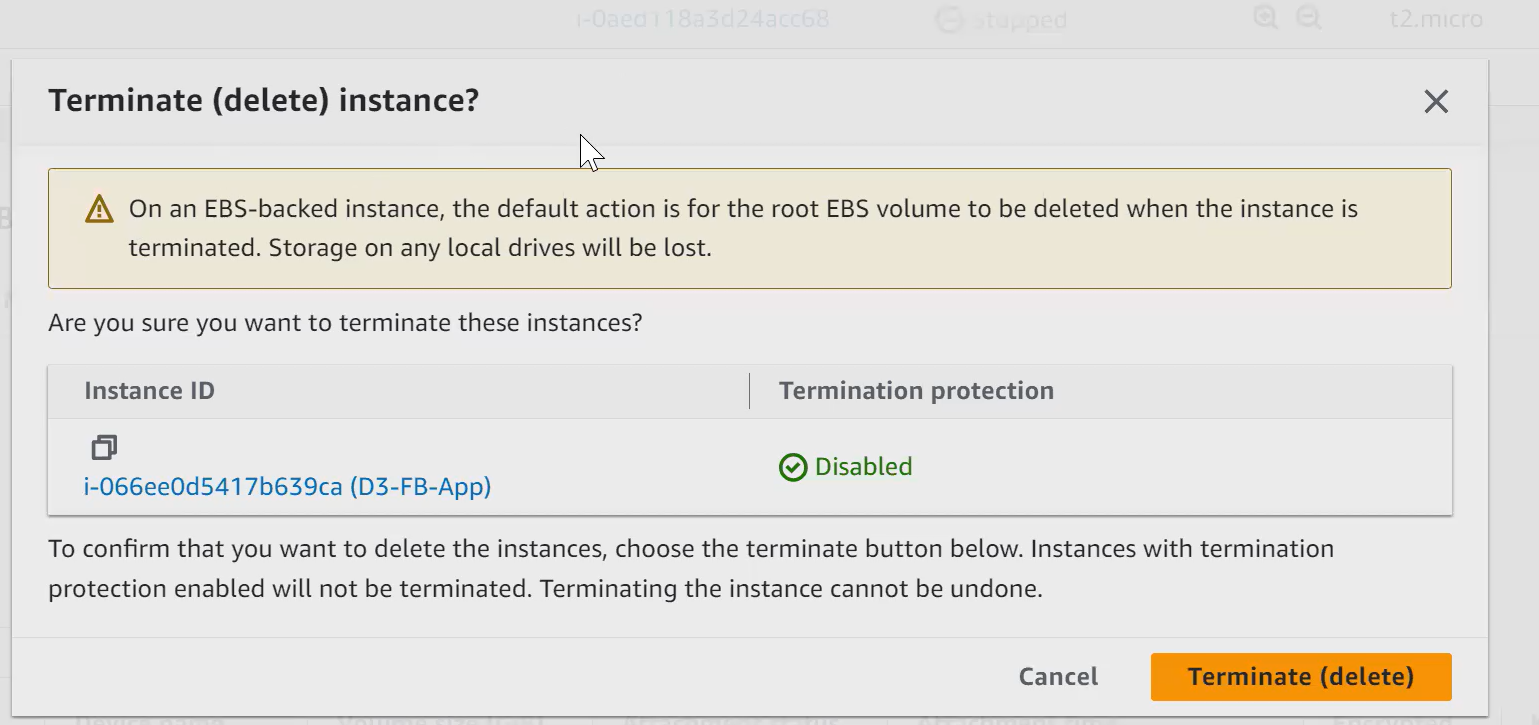
With windows – We can perform the same operations from gitbash



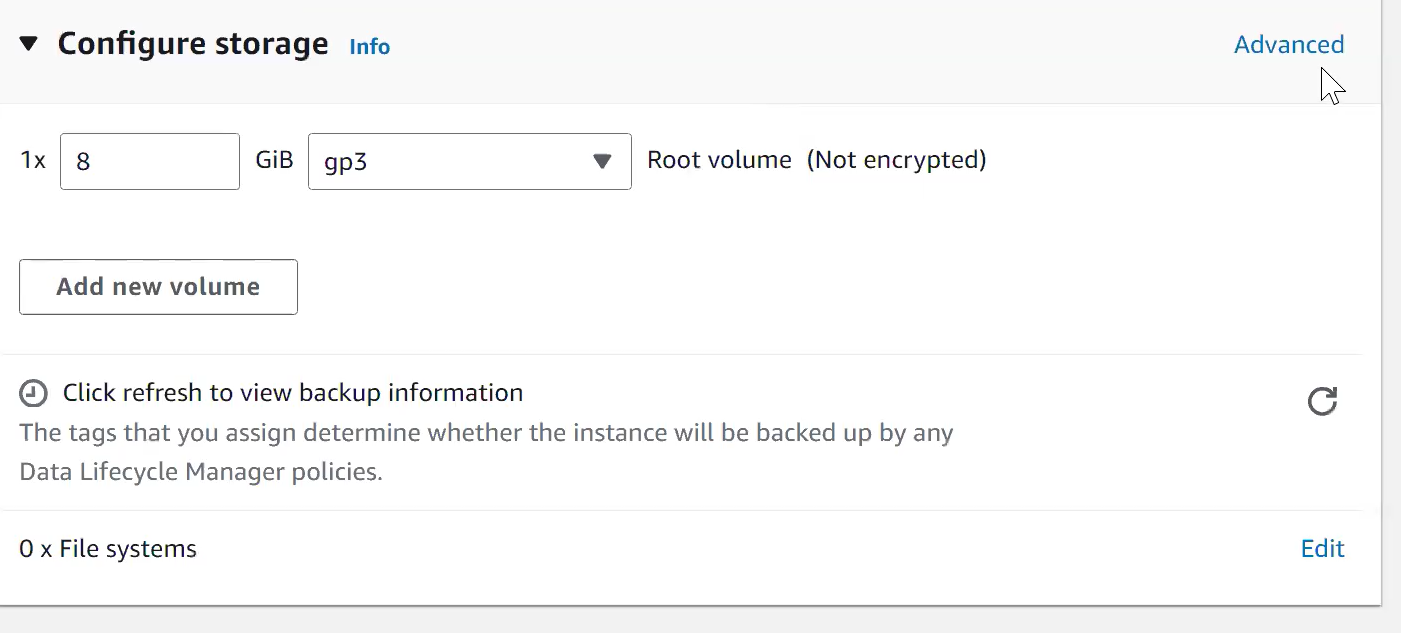
**When we start and stop an instance, the public IP may change.**

When we create an instance and add a storage

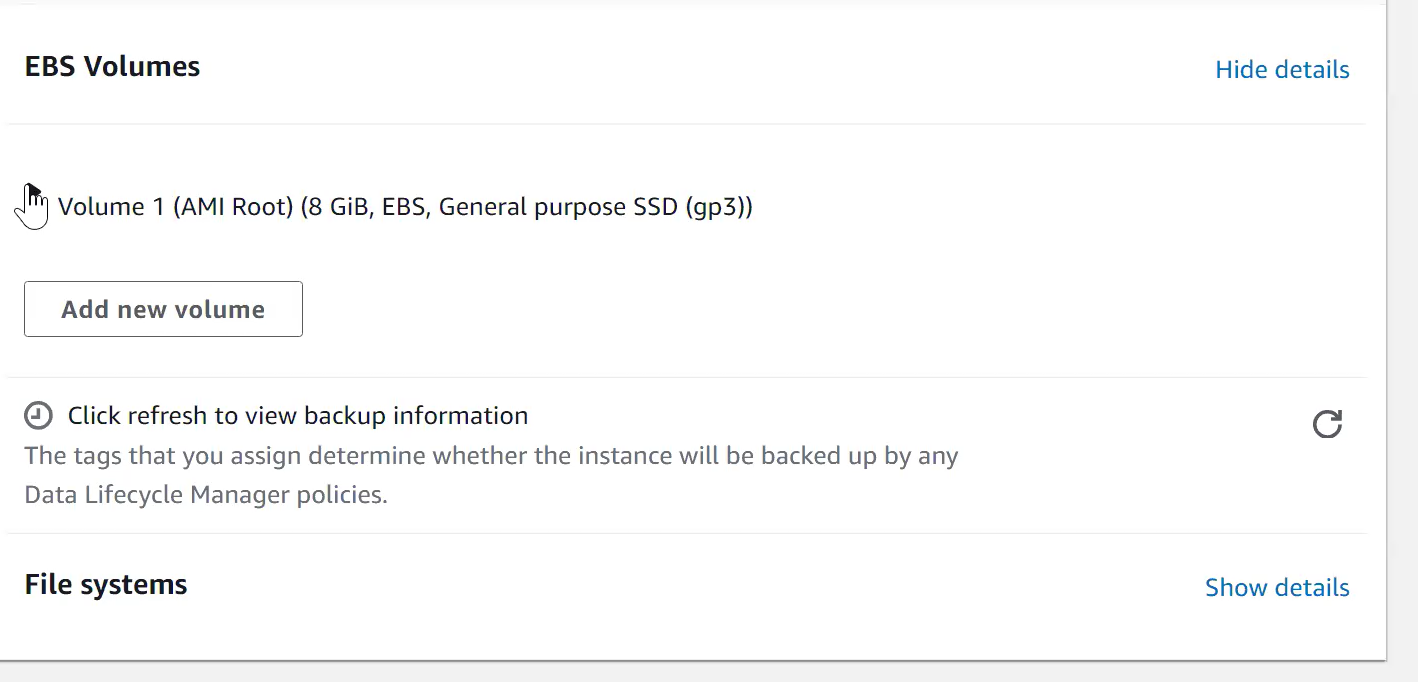


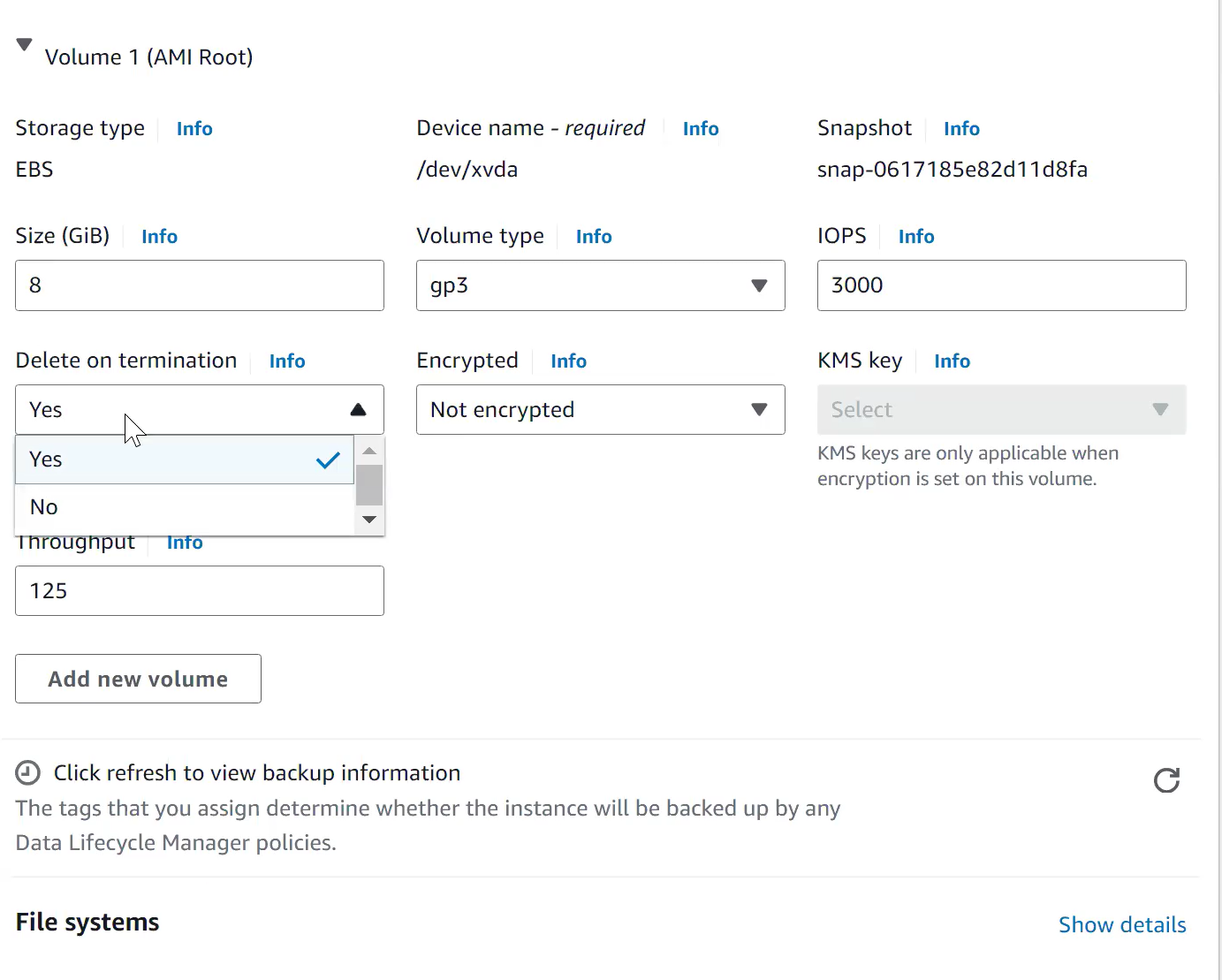
  
  
So, when we terminate the instance, the storage will be terminated with it.

These is default behaviour.



We can select advanced option





We can select no on termination, so volume won’t get deleted.