😊 ELASTIC CLOUD COMPUTE (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service provided by Amazon Web Services (AWS) that allows users to rent virtual servers in the cloud. EC2 instances provide scalable computing capacity, allowing users to quickly scale up or down based on their application requirements.

Key features of Amazon EC2 include:

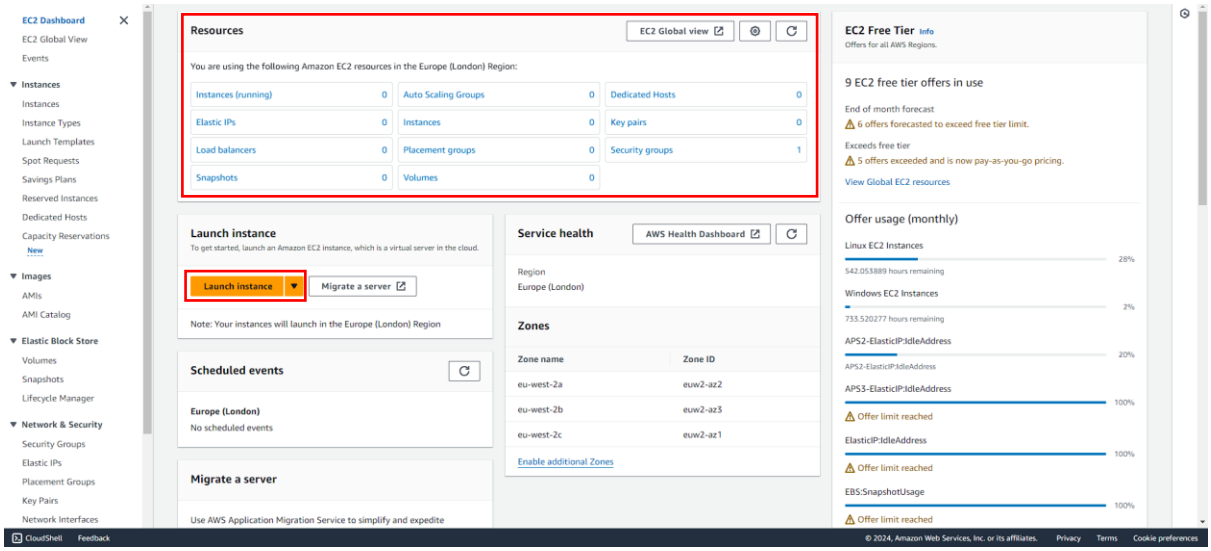
1. **Scalability:** Users can easily scale their compute capacity by launching additional instances or stopping unused instances. This flexibility is particularly useful for applications with varying workloads.
2. **Variety of Instance Types:** EC2 offers a wide range of instance types optimized for different use cases, such as compute-optimized, memory-optimized, storage-optimized, and GPU instances.
3. **Customization:** Users can choose the operating system, instance type, storage, and networking configurations for their EC2 instances, providing a high degree of customization to meet specific application requirements.
4. **Security:** EC2 instances can be launched within a Virtual Private Cloud (VPC), allowing users to control network settings, security groups, and access control lists. Amazon EC2 also supports key pairs for secure access to instances.
5. **Pay-as-You-Go Pricing:** Users pay only for the compute capacity they consume, with different pricing options based on instance type, region, and usage duration. This pay-as-you-go model is cost-effective and allows for efficient resource utilization.
6. **Elastic Load Balancing:** EC2 instances can be easily integrated with Elastic Load Balancers to distribute incoming traffic across multiple instances, improving fault tolerance and availability.
7. **AMI (Amazon Machine Image):** Users can create custom machine images containing their applications, configurations, and operating systems, making it easy to launch instances with pre-configured settings.

😁 TO BEGIN WITH LAB:

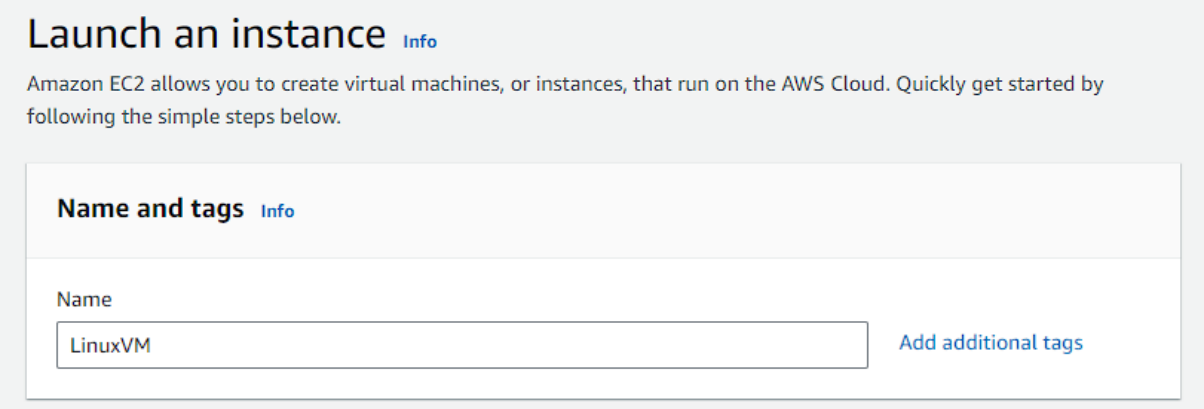
😎 STEP 1: LAUNCH A LINUX EC2 INSTANCE

1. Log in to AWS Console.
2. There you need to search EC2. Choose this service accordingly.

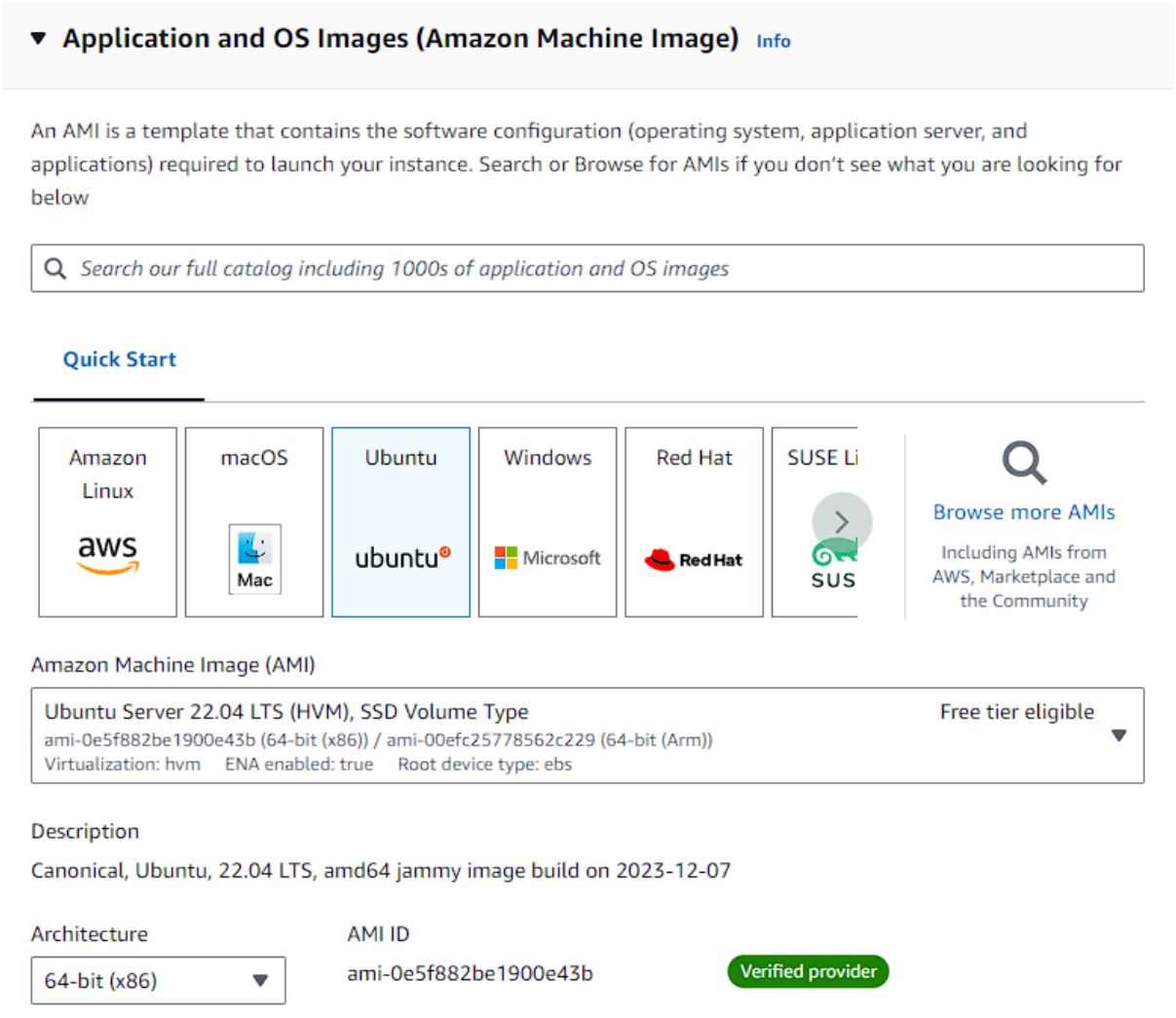
3. This the dashboard for EC2. You can find different resources here.
4. But, to create an instance you need to click on **Launch Instance.**



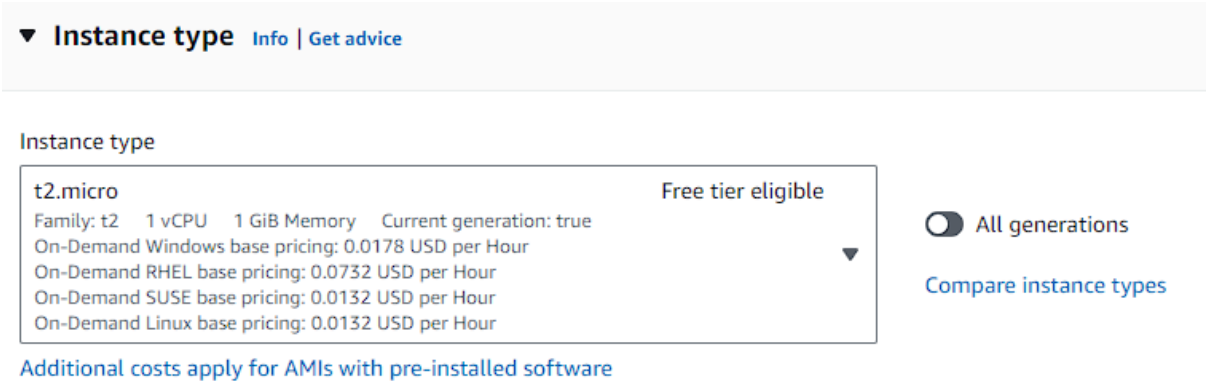
5. So, first and foremost, you must name your instance.



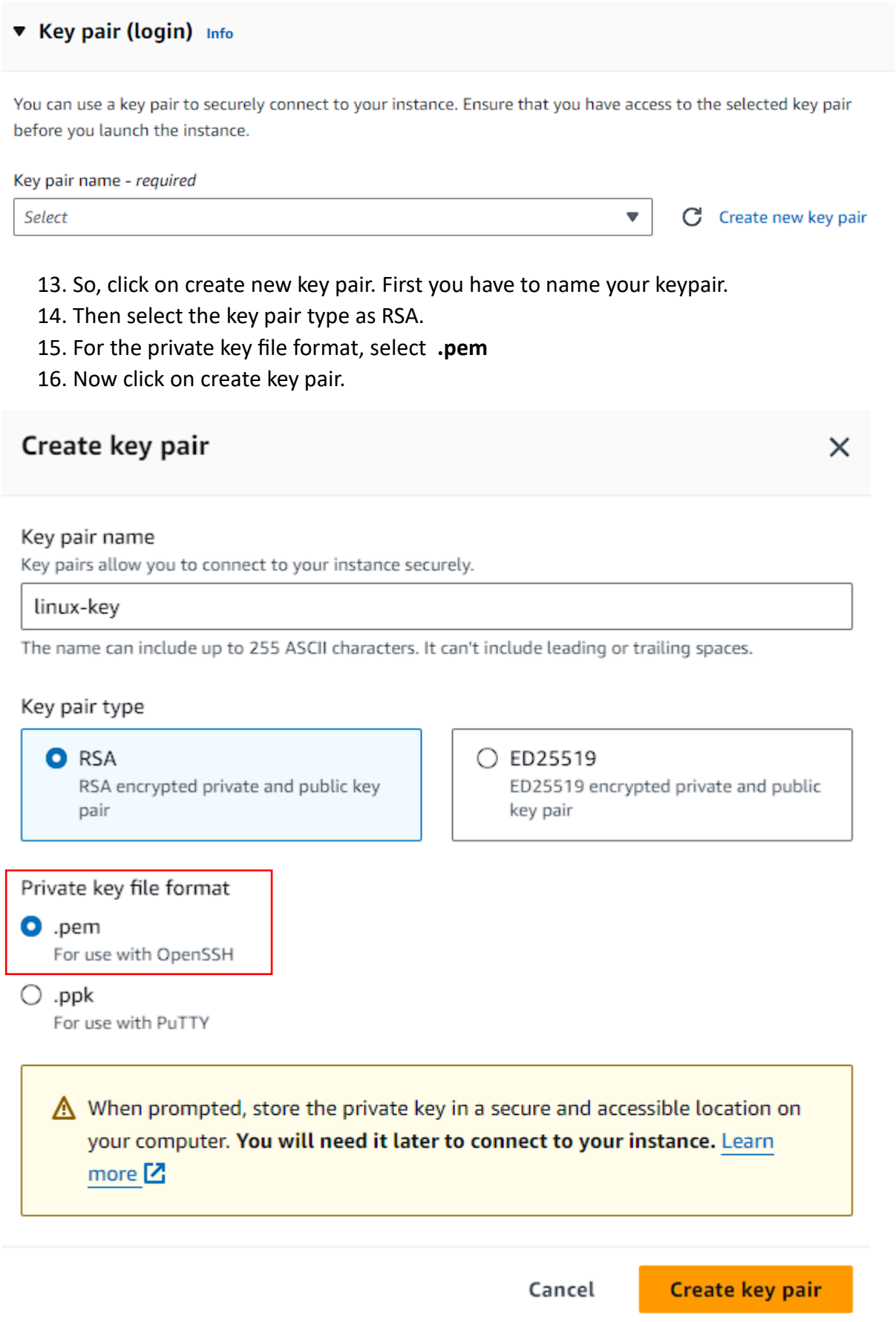
6. Then in the **Application and OS Images**, you have to select an Operating system.
7. Here, you are going to select **Ubuntu** as your Operating system.
8. For **Amazon Machine Image (AMI)**, select the latest AMI that is available. In this case it is ubuntu server 22.02 LTS.
9. Keep architecture to default as 64-bit (x86)



10. Now you must choose an instance type. There are more instance types with higher specifications available, but for the time being, you must select **t2.micro**. This instance type is qualified for the free tier.

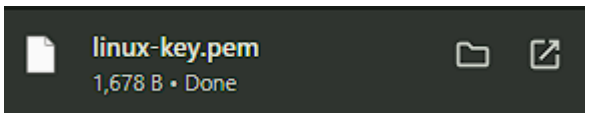


11. In the key pair, you have to create a key pair, so that you can log in to your virtual machine. These key pairs are used as keys to log into your VMs.
12. As of now, you might not have any key pairs, so you have to create a key pair.

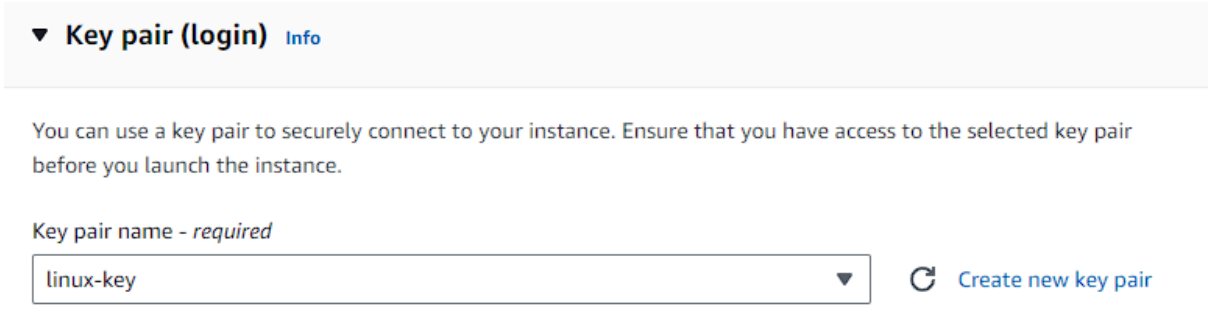


13. So, click on create new key pair. First you have to name your keypair.
14. Then select the key pair type as RSA.
15. For the private key file format, select **.pem**
16. Now click on create key pair.

17. After clicking on the Create Key Pair button, a file should have been downloaded; this file will be utilized throughout this experiment.

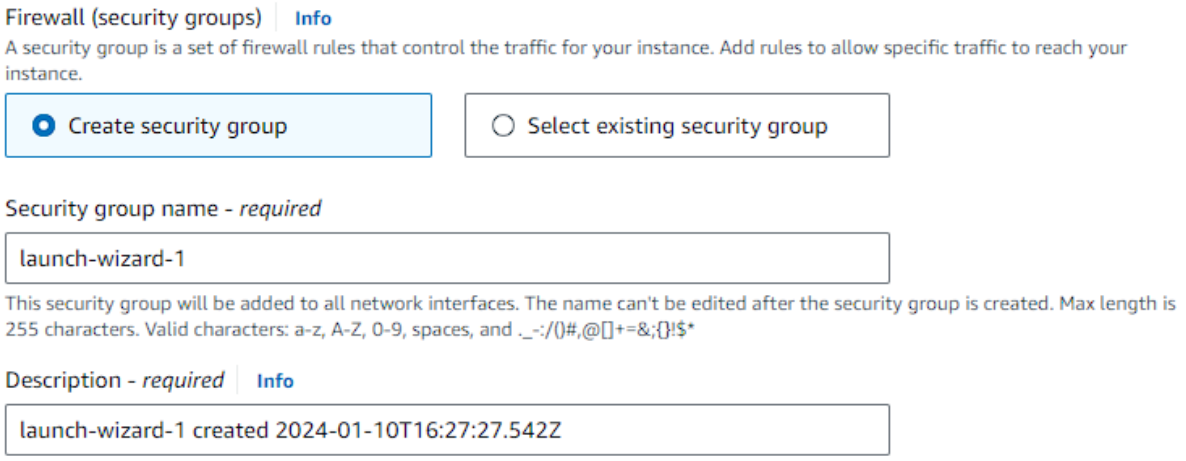


18. Here you can see your key pair.

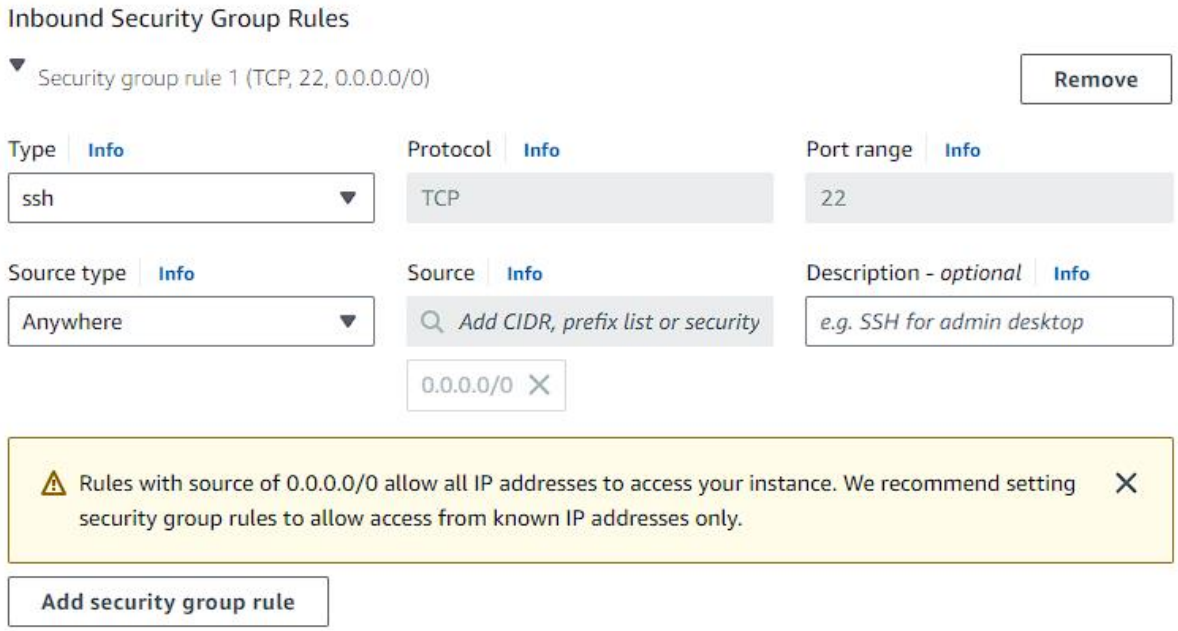


19. If you will take a look at network settings, you will see a VPC which is required and it is a default VPC.
20. The subnet has no preference because you haven't given it any.
21. The **auto assign public IP** feature is enabled.

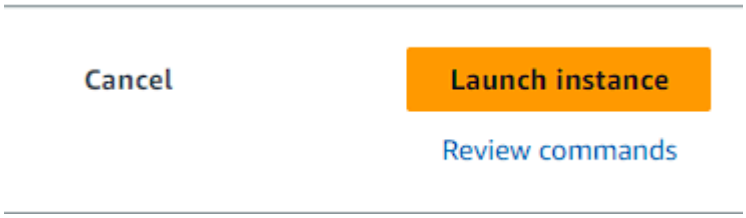
22. In the firewall option, you will see that it is creating a new security group, you can change its name and description, but for the time being keep it to default.



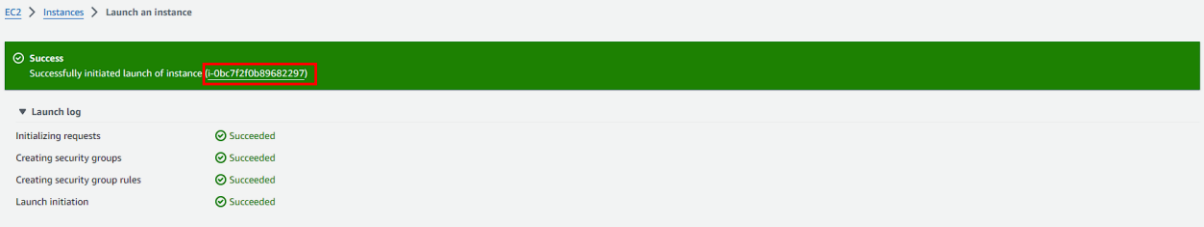
23. And the security group you will see this Inbound security group rules. These Inbound rules are used for virtual machine to log in to different areas of interest. Let it be default too.



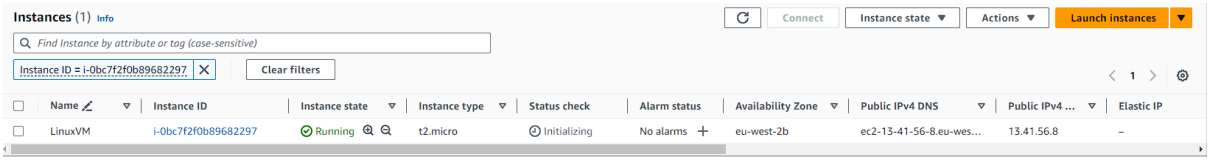
24. Now the setup for EC2 instance is completed, you can click on launch instance. This will launch your instance and you will be able to see your instance.



25. You can see that your instance is launched successfully.
26. Now if you will click on this instance id, you will directed towards your instance.

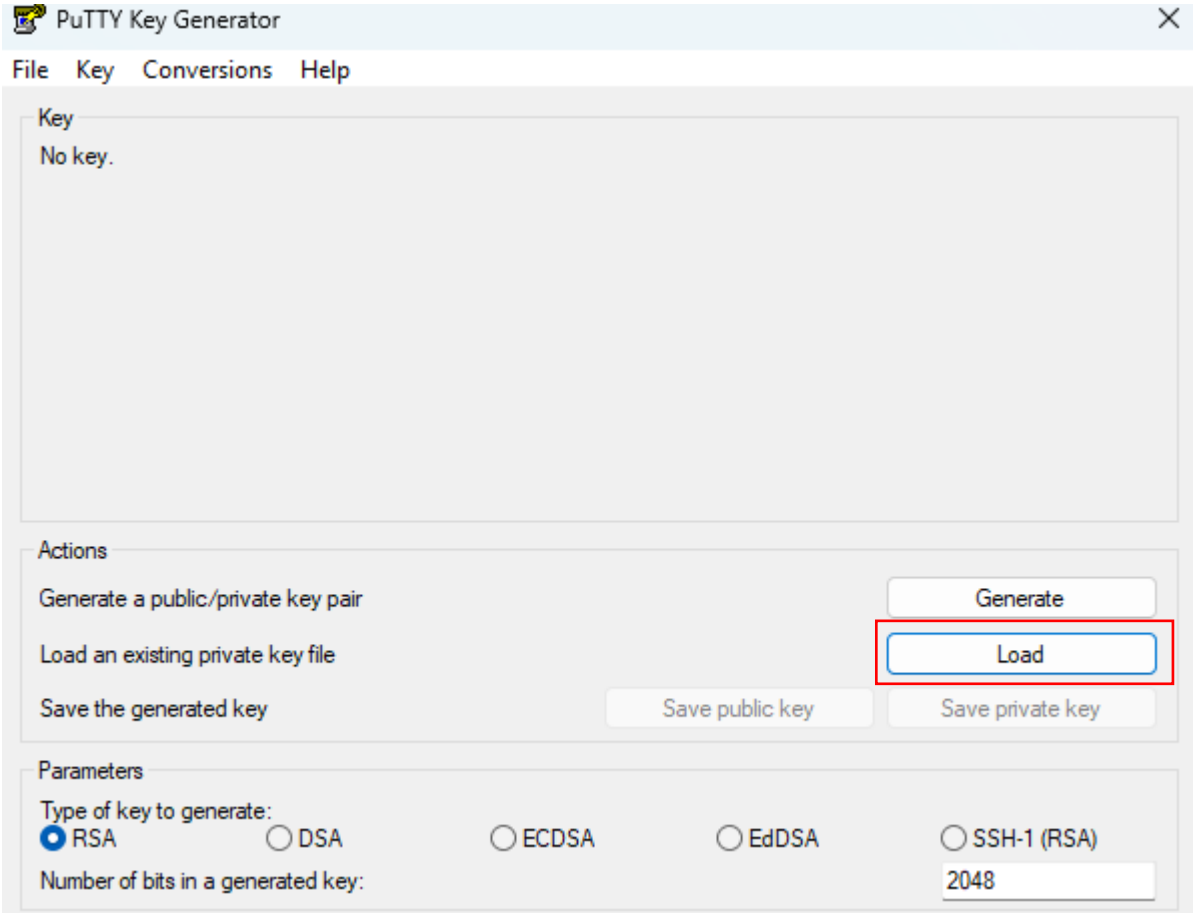


27. Here is your instance, it is up and running.

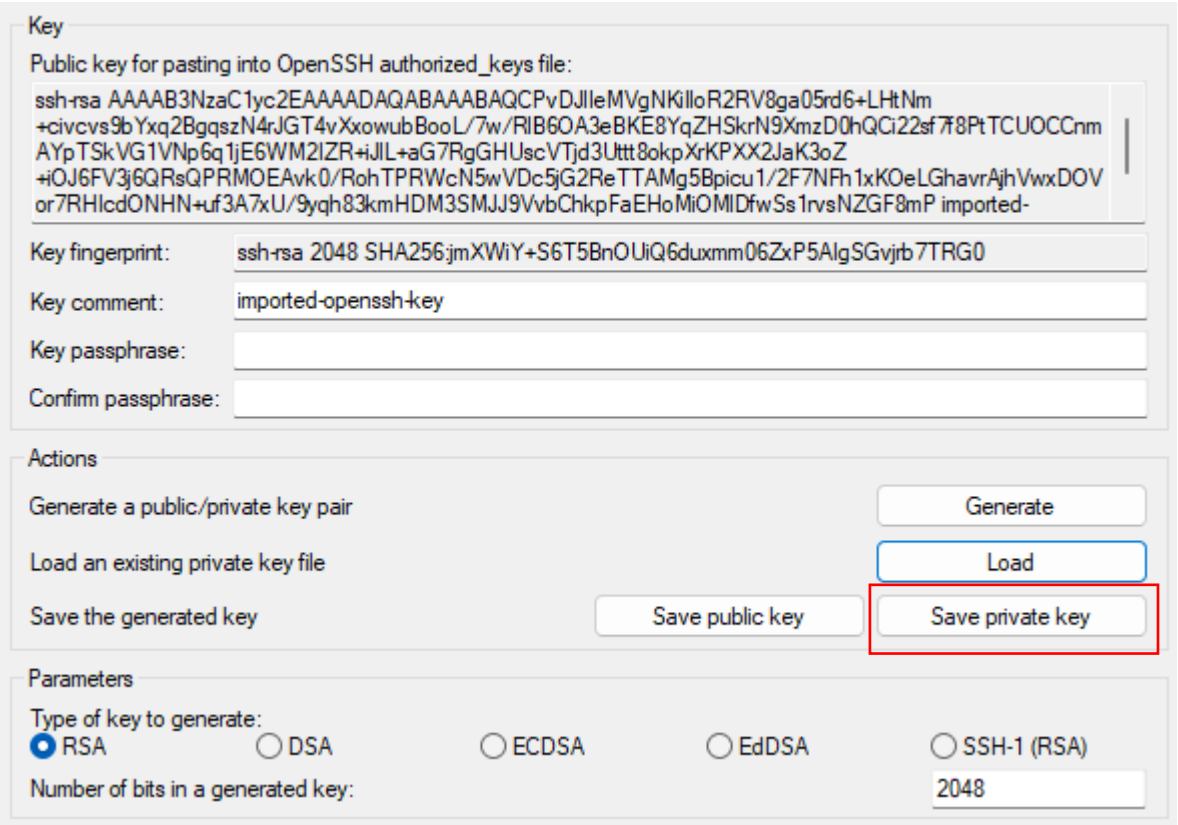


STEP 2: CONNECTING THE INSTANCE

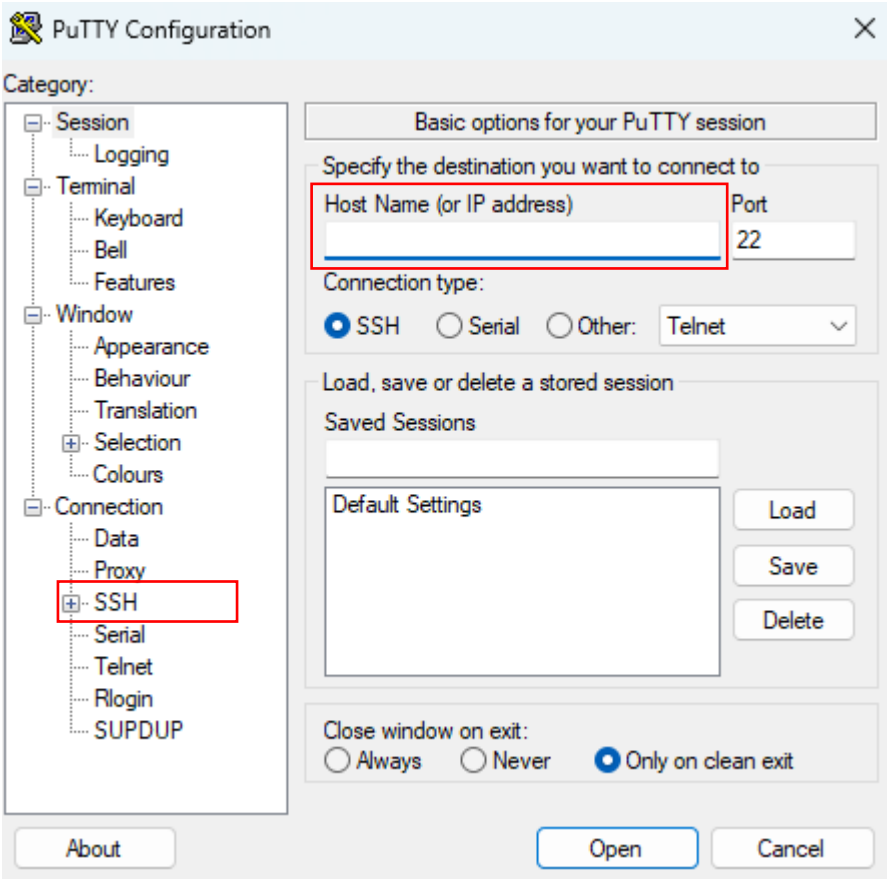
1. So, if you want to connect your instance, you have to download **Putty**.
2. **Putty** is a tool which helps users to connect their instance and work on it.
3. To download **Putty**, you can use this website. https://www.putty.org/
4. With **Putty** you also need to download a tool called **PuttyGen.**
5. **PuttyGen** can also be downloaded from this website too.
6. Now once you have downloaded Putty and puttygen tools. First you have to open PuttyGen because this tool will help you to convert your .pem file to .ppk because Putty does not support .pem files it only supports .ppk files.
7. Now open puttygen. Click on load. This option opens your system files to browse from them.



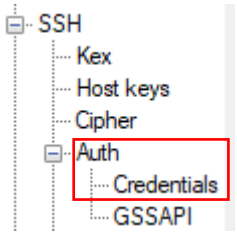
8. Once you have loaded the .pem file here, now you have to click on **Save Private key.** This tool will save your file as a private key.



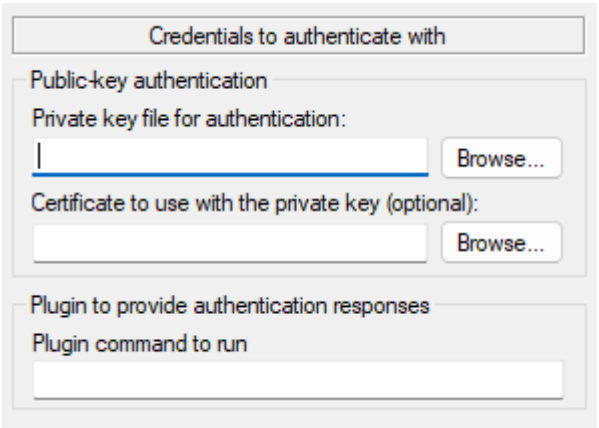
9. Now open **Putty** and in the host name you have to paste the public IP address of the instance. So, copy that and paste it in the host name.
10. Once you've pasted the IP, now you have to click on SSH.



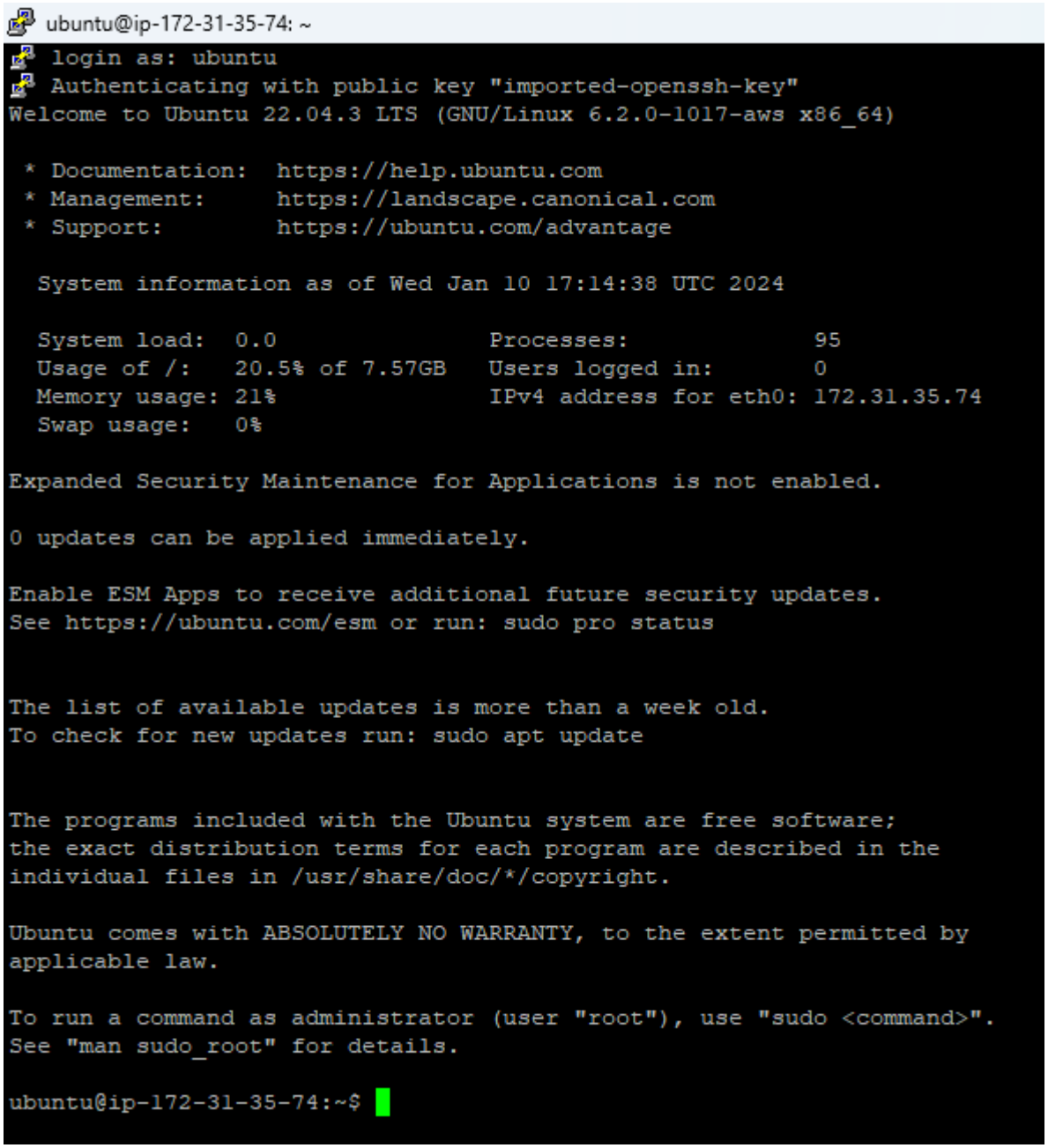
11. After clicking on SSH, you will some more options, so you have to choose Auth, then after click on credentials.



12. In the credentials you have to browse to the place where you saved your .ppk file after converting it.
13. Then once the location is saved here. You can click on Open.

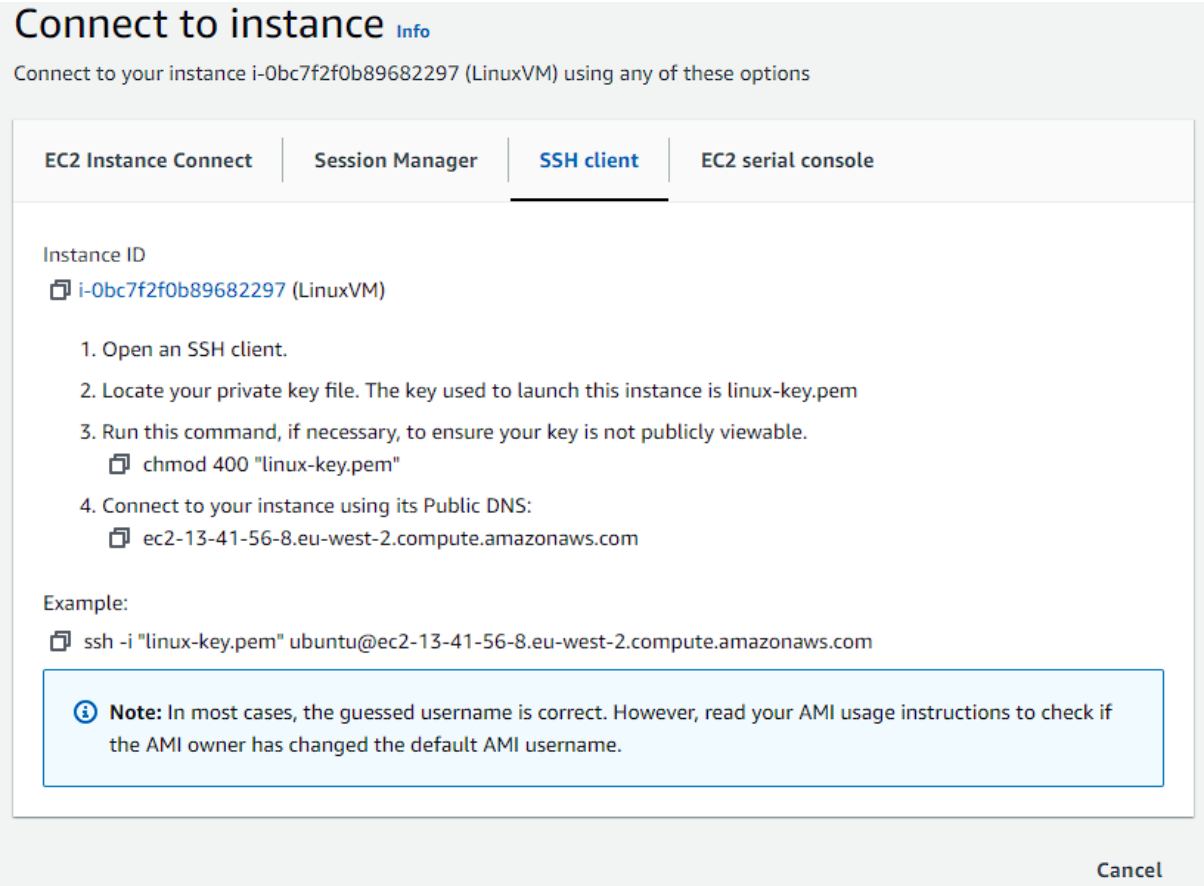


14. You will see that you have logged into the server.
15. Here in login as you have to **Write ubuntu,** because this will be your username to login.

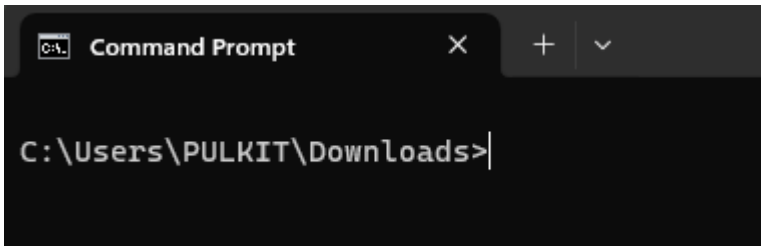


🤓 STEP 3: CONNECTING THE INSTANCE THOUGH SSH CLIENT

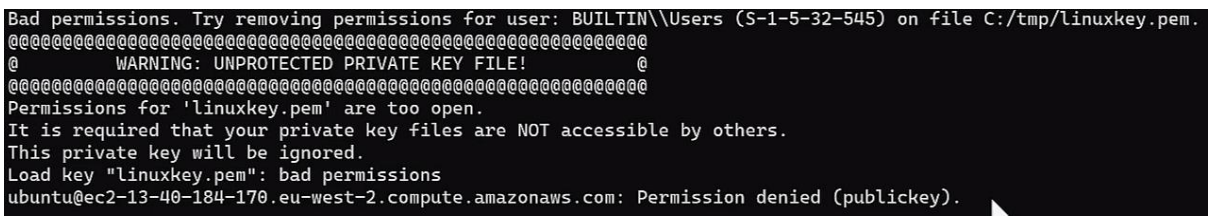
1. To connect you instance via SSH client you have click on connect on the console.
2. Then you have to navigate to SSH client.
3. Now open command prompt on your local machine.



4. After opening command prompt navigate cmd to the location where your .ppk file is installed.



5. Once you are on the location where you have stored your .ppk file now you have to paste that example link that you copied earlier.
6. If you encounter this error then you can run these two commands which will give it permission to log in your instance.
   icacls .\linuxkey.pem /inheritance:r
   icacls .\linuxkey.pem /grant:r "Administrator:(r)"



7. Once you have those two commands. Now try again to log in.

8. You will see that you have successfully logged into your Linux instance successfully.

```
C:\Users\PULKIT\Downloads>ssh -i "linux-key.pem" ubuntu@ec2-13-41-56-8.eu-west-2.compute.amazonaws.com
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan 10 17:42:23 UTC 2024

  System load:  0.0               Processes:             97
  Usage of /:   20.8% of 7.57GB   Users logged in:       1
  Memory usage: 21%               IPv4 address for eth0: 172.31.35.74
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Jan 10 17:14:39 2024 from 192.140.153.216
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-35-74:~$
```

STEP 4: DEPLOYING A WEB SERVER ON THE INSTANCE

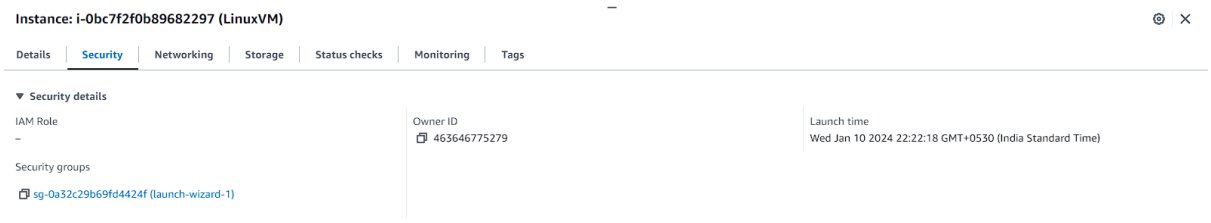
1. Now you are going to install NGINX web server on your Linux instance.
2. For that you have to run a set of commands on your instance. You can choose to run these either in Putty or on Command Prompt.
3. The first command is for updating your Linux instance server.
   sudo apt-get update

```
ubuntu@ip-172-31-35-74:~$ sudo apt-get update
```

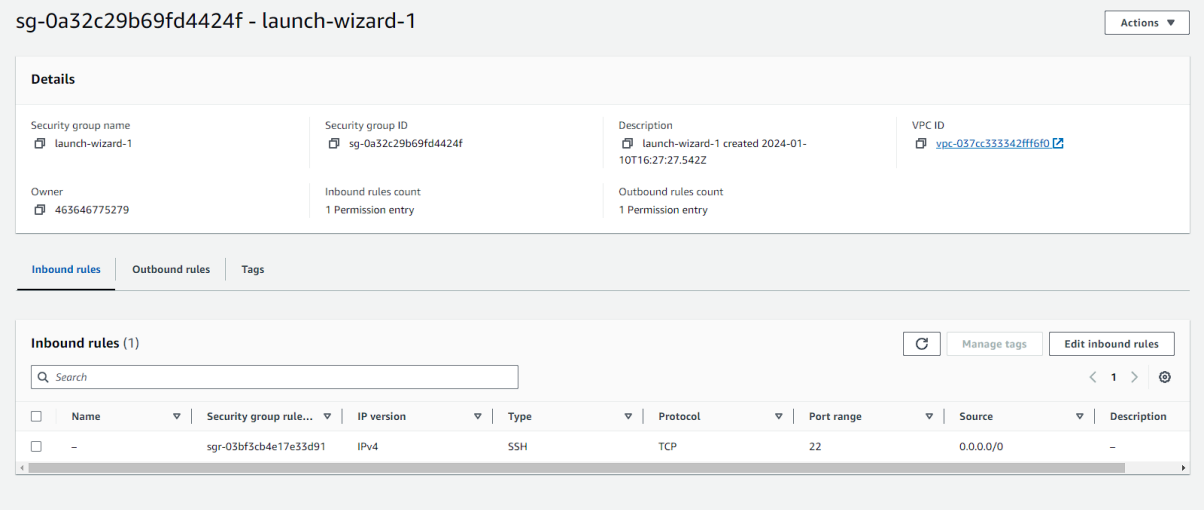
4. The second command is for installing your nginx server.
   sudo apt-get update install nginx

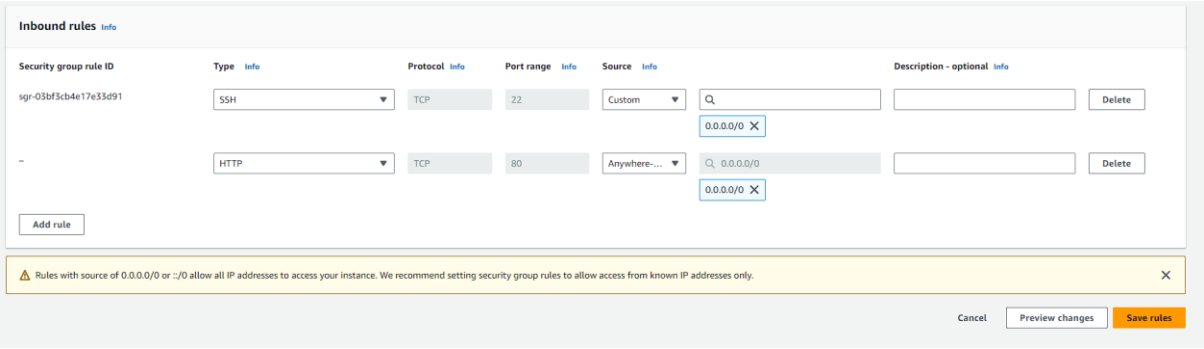
```
ubuntu@ip-172-31-35-74:~$ sudo apt-get install nginx
```

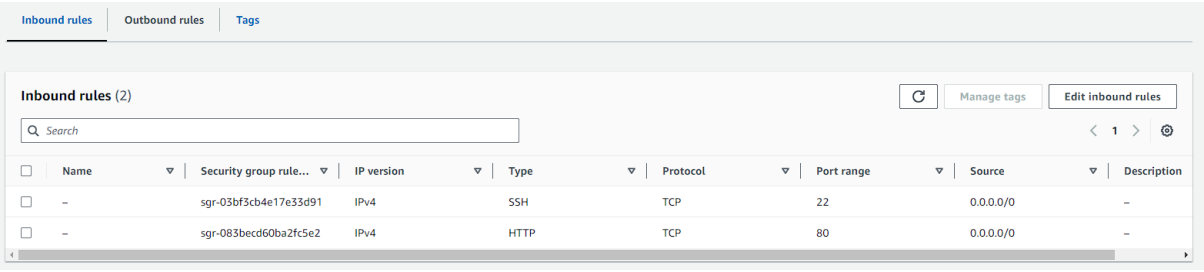
5. So, now you have to go back to your console and add port 80 to your security group.
6. Now select your instance and go to security, there you will a link to navigate you to the security group. Click on it.



7. In the security group you will see some options present over there.
8. If you will just click on edit inbound rules under the inbound rules and add port 80 to your instance.







9. Now go back to your instance and copy your public IP address and paste it in a new tab.
10. You will be able to see the nginx server up and running.