

**Name: Prasad Deshpande**

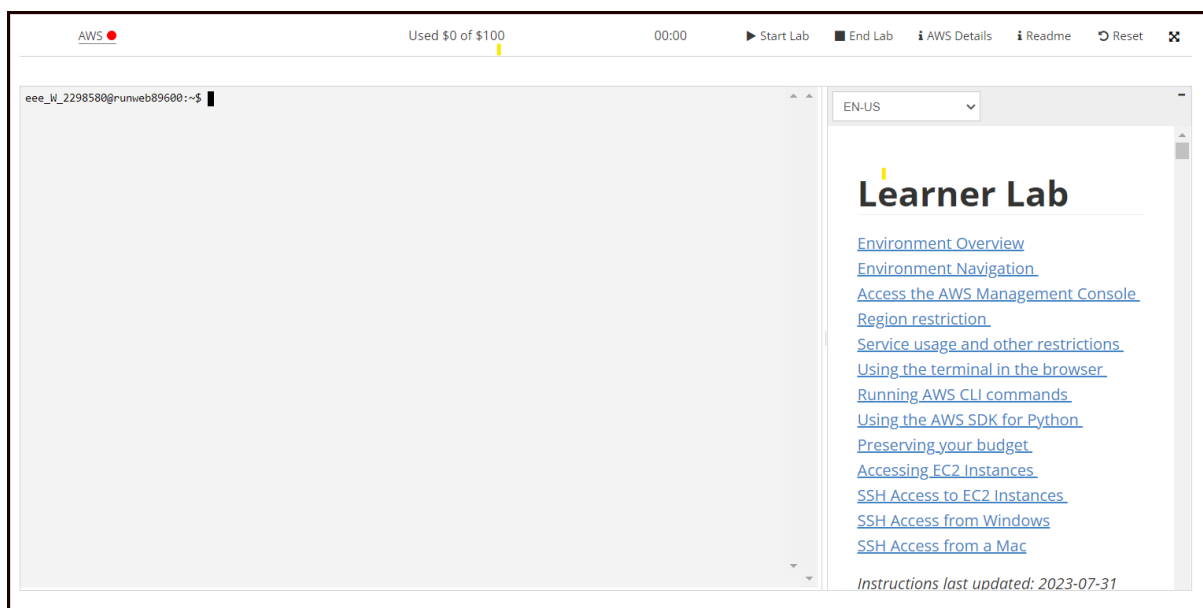
**Enrollment Number: 243341024**

**MSC(CS) Part I**

## **Cloud Computing Practical Assignment No 3**

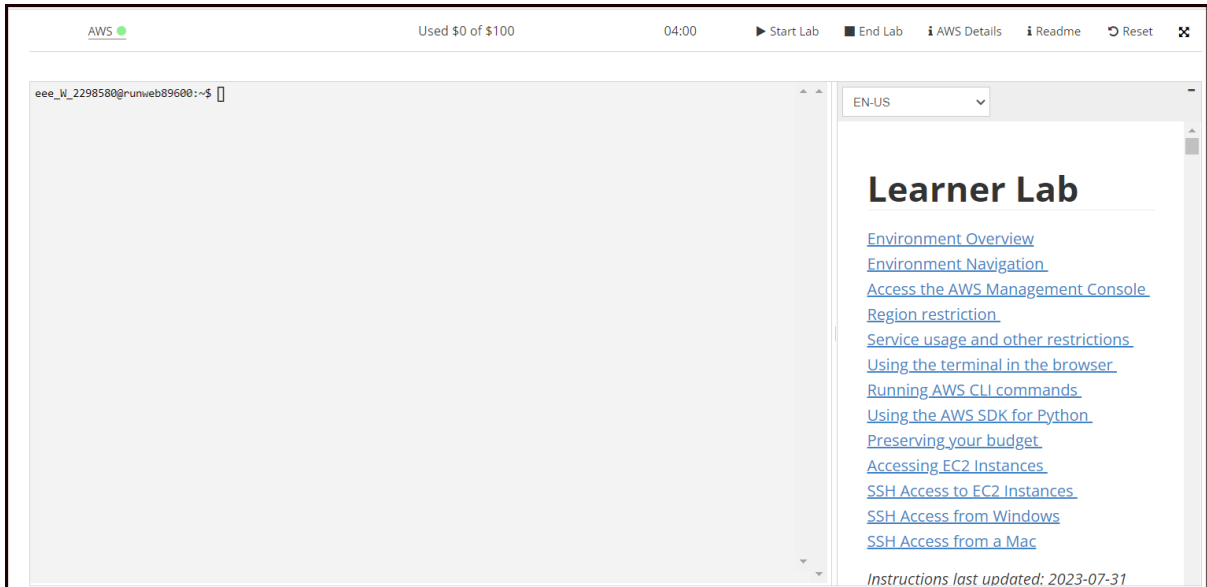
Working and Implementation of Infrastructure as a service. Create an EC2 Linux Instance, Install an Apache Web Server and run the Hello World page by typing commands on a virtual machine.

First of all open Virtual Lab. After opening the lab, you will get an interface like Fig 1.



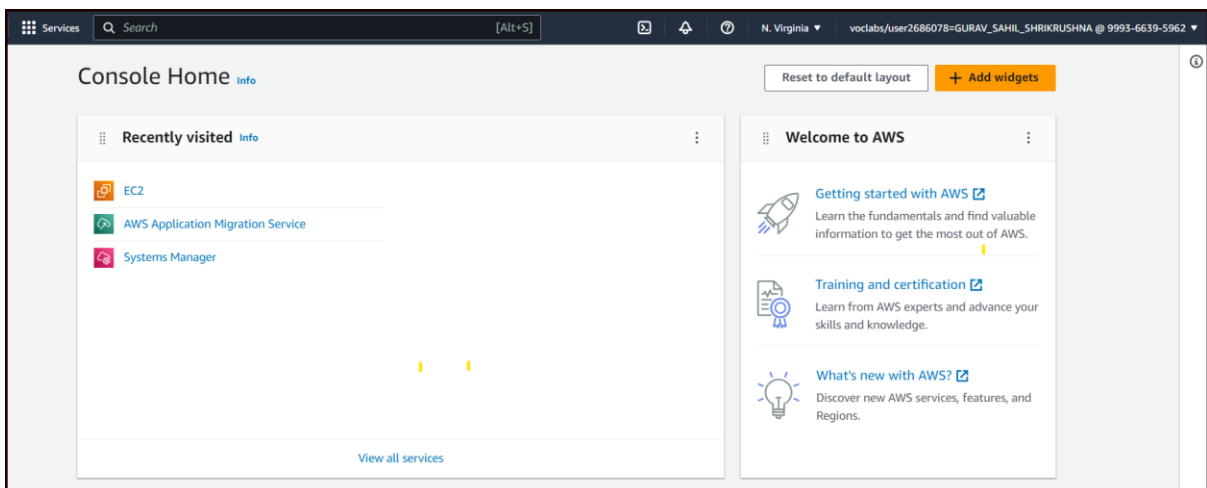
**Fig 1**

Then click on the Start Lab button. When the circle icon to the right of the AWS link in the upper-left corner turns green, it indicates that the lab environment is ready to use this we can see in Fig 2. To launch the AWS Management Console in a new tab, select the AWS link



**Fig 2**

After selecting AWS link new console is open on new tab which we can see in Fig 3. In that we select the EC2 (Elastic Cloud Computing) service. You can see that service in Fig 3. If you have used it before then you can see that service in recently visited service. If you don't see EC2 service then follow the path Services => Compute => EC2.



**Fig 3**

After selecting the EC2 service the new interface will be shown like in Fig 4. In that click on Launch Instance.

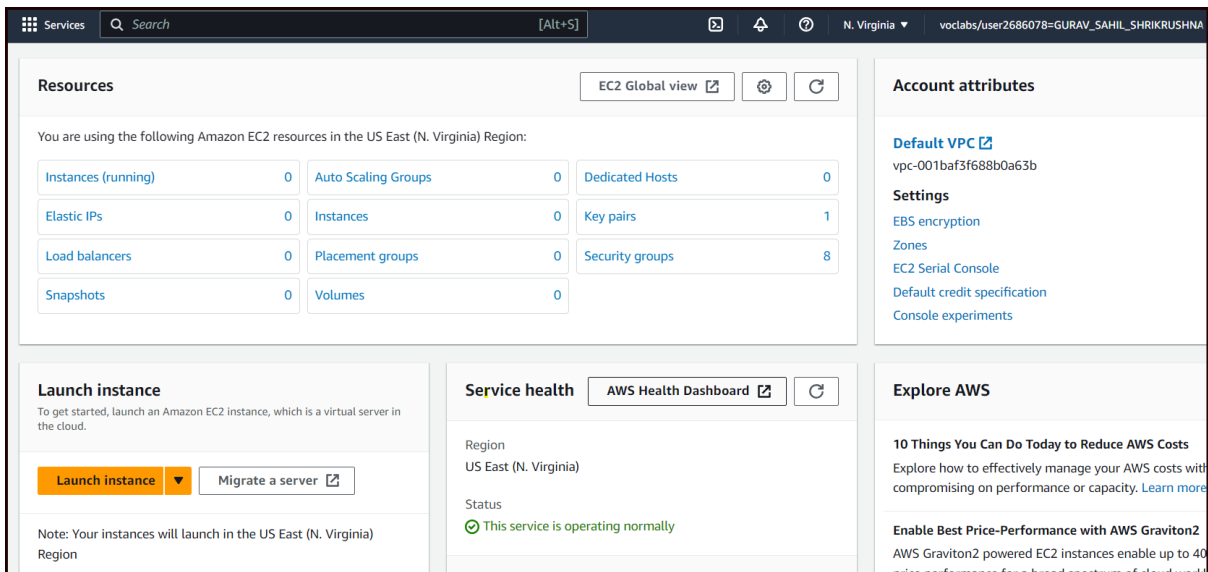


Fig 4

After clicking on Launch Instance some information regarding that instance will appear which we need to fill. That we can see in Fig 5.

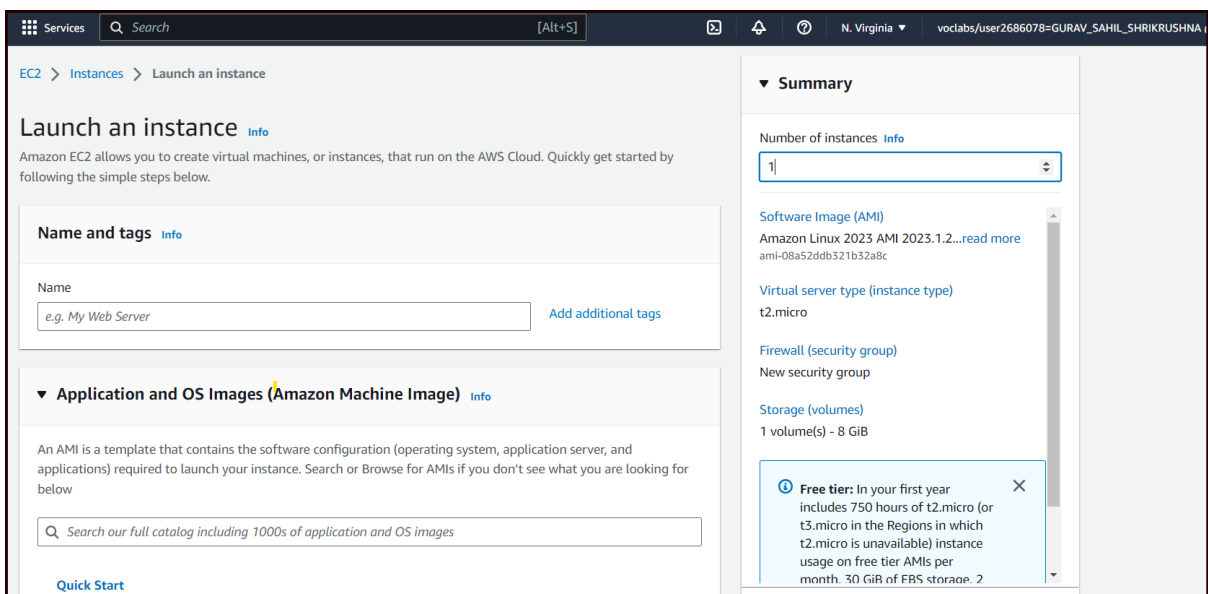


Fig 5

If we create more than one instance then how can we identify our instance? For that reason inside name and tags we write some name for instance so later we can identify them. so here in Fig 6 you can see i named it Linux. Then we need to select the Amazon Machine Image. In this we can specify which operating system (OS) and application server you need to launch in your instance. Here I select Amazon Linux.

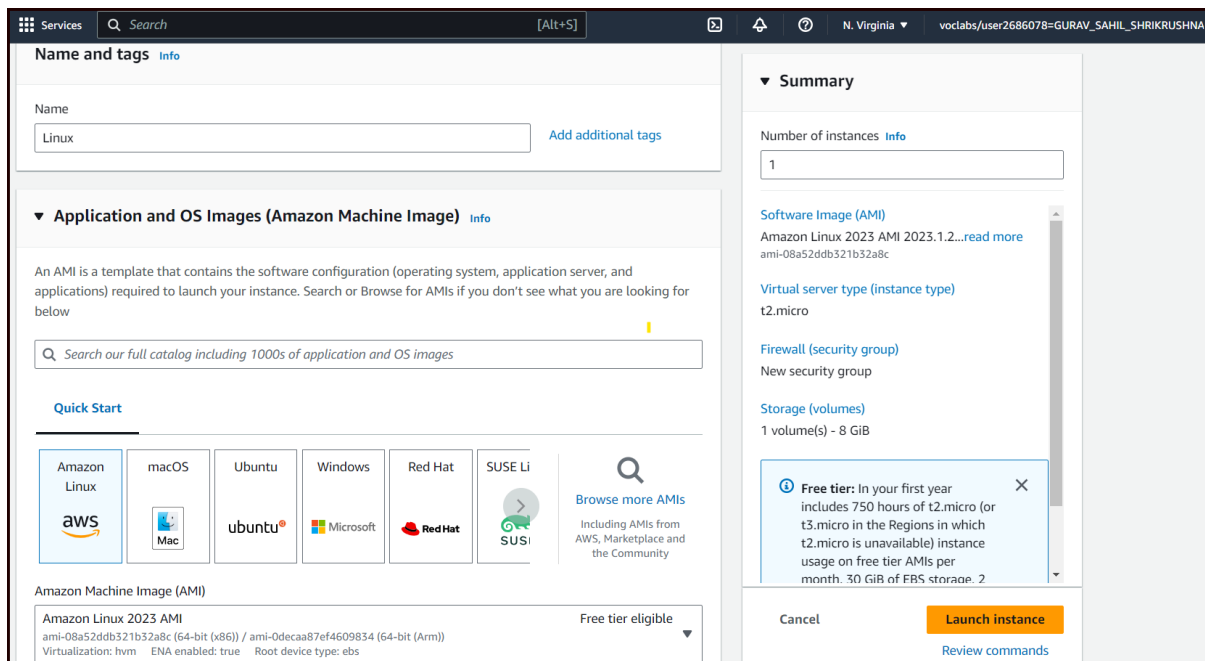


Fig 6

After scrolling the page we will see other details like Fig 7. In instance type we can choose our computing power, memory, networking or storage requirements. Here I have selected default instance type. In t2.micro instance type features 1 virtual CPU and 1 GiB memory. Fig 7 below instance type has key pair (login) it is used for securely connect to your instance. It is a set of security credentials that you use to prove your identity when connecting to an Amazon EC2 instance. There default key pair is available but I will create new key pair for windows EC2 instance. For this click on "Create New Key Pair".

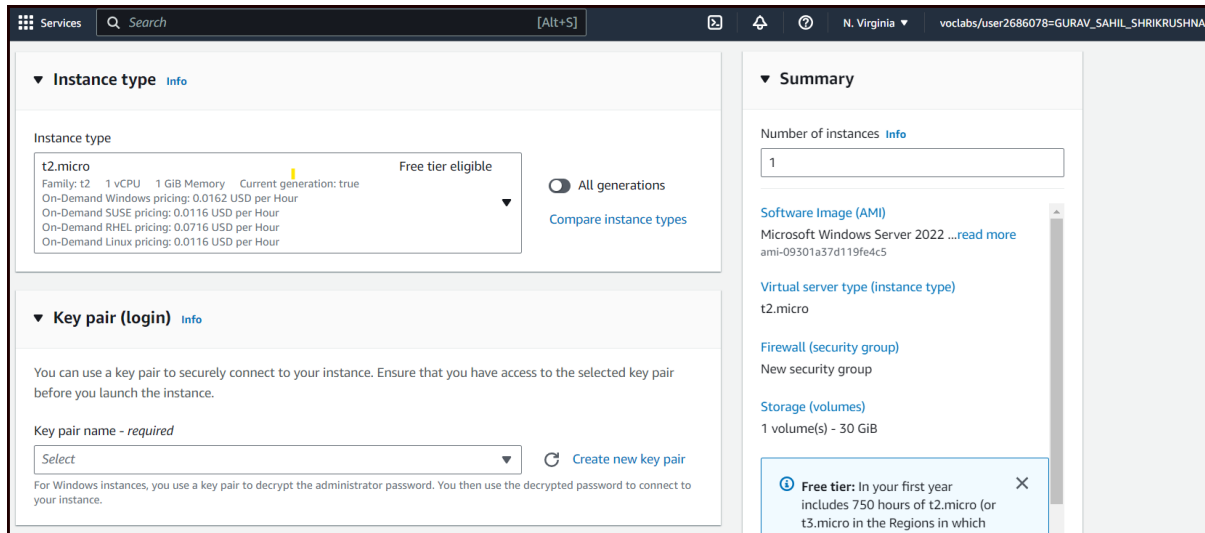


Fig 7

After clicking on "Create New Key Pair" we will get an interface like Figure 8. We can create n number of keys. To identify the key later, we need to give it a name. so here i will give name Linux\_key. Other options we choose according to our usage. Here I use default values. Then click "Generate Key Pair".

The screenshot shows a 'Create key pair' dialog box with a close button (X) in the top right corner. The dialog is divided into several sections:

- Key pair name:** A text input field with a placeholder 'Enter key pair name'. Below it, a note states: 'The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.'
- Key pair type:** Two radio button options are shown. The first is 'RSA' (selected), described as 'RSA encrypted private and public key pair'. The second is 'ED25519', described as 'ED25519 encrypted private and public key pair (Not supported for Windows instances)' and is currently disabled.
- Private key file format:** Two radio button options are shown. The first is '.pem' (selected), described as 'For use with OpenSSH'. The second is '.ppk', described as 'For use with PuTTY'.
- Warning box:** A yellow box with a warning icon contains the text: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more' with a link icon.
- Buttons:** At the bottom right, there are two buttons: 'Cancel' and 'Create key pair' (highlighted in orange).

Fig 8

After clicking on "Generate Key Pair" we get the key. we can see in figure 9.

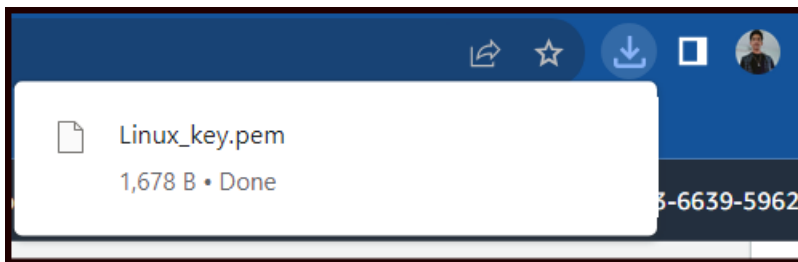


Fig 9

After downloading the key we will get an interface like Figure 10. We haven't changed anything in that network setting, it's default

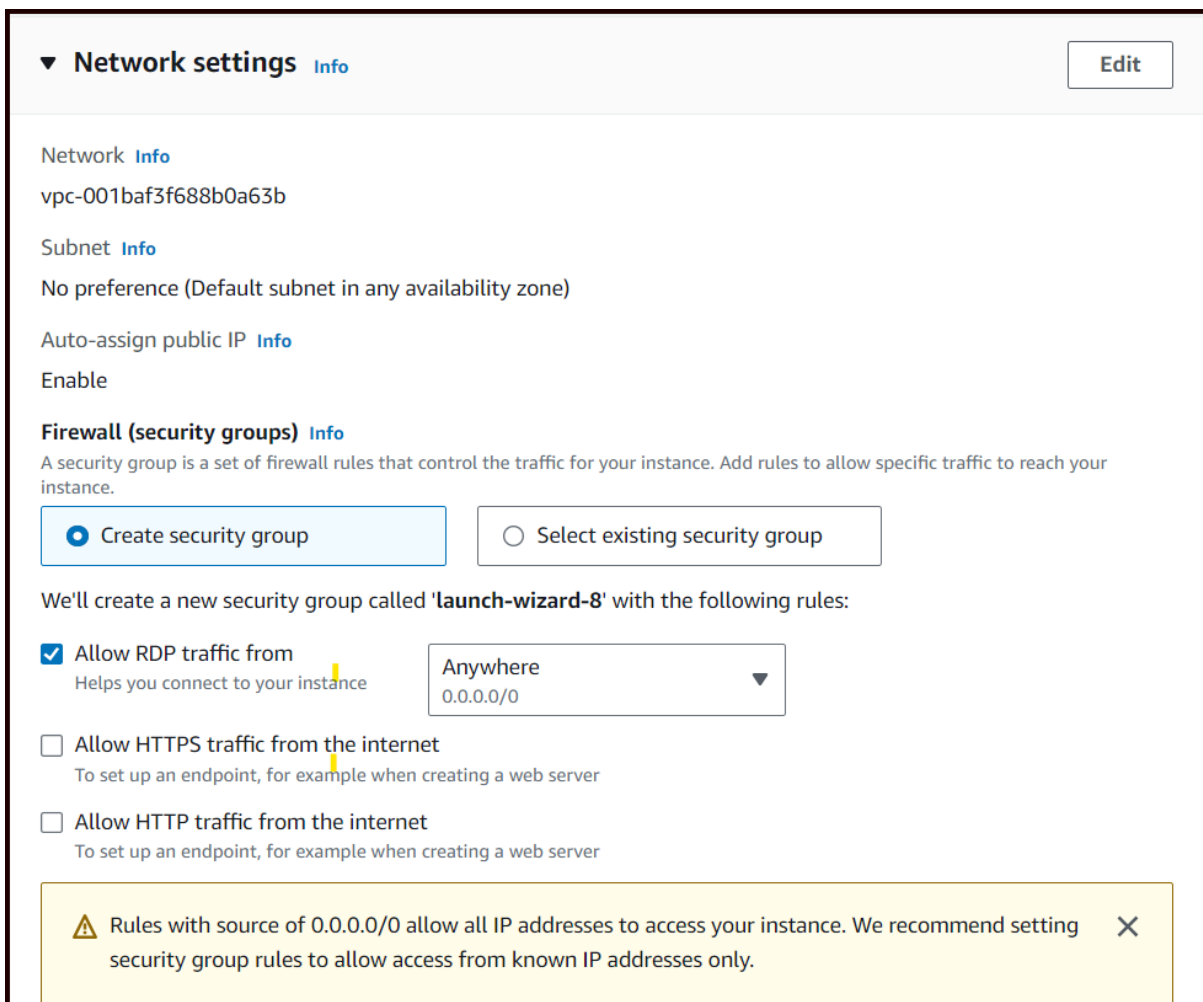


Fig 10

After setting the network, we will get the interface like Figure 11. We have not changed anything in that configuration storage, it is default. Then click on "Launch instance".

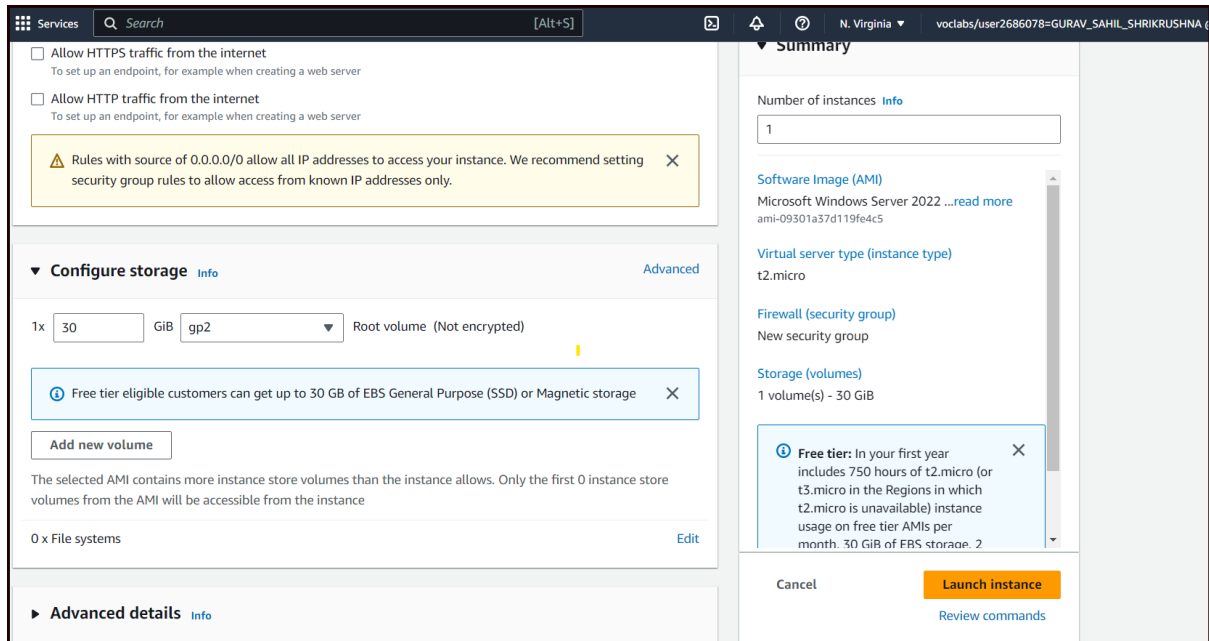


Fig 11

After clicking on "Instances". we can see in the Fig 12 windows instance is created

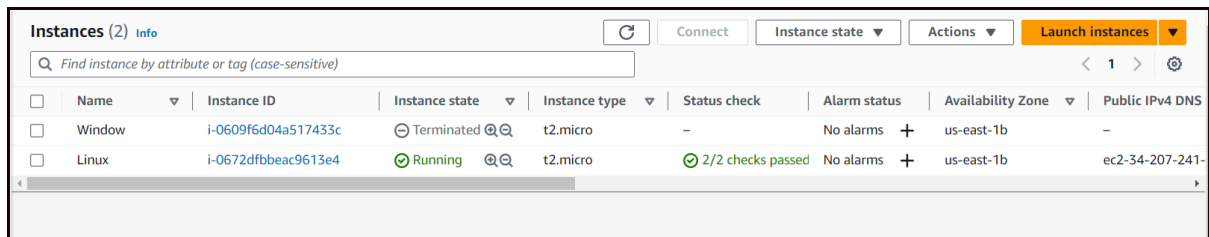


Fig 12

Click on the check box of the given example as in Figure 13 to see the details of the instance. In that we can see information regarding instance details, Security, networking, storage, status check, Monitoring and tags. But in status check we can see that the instance is initializing.

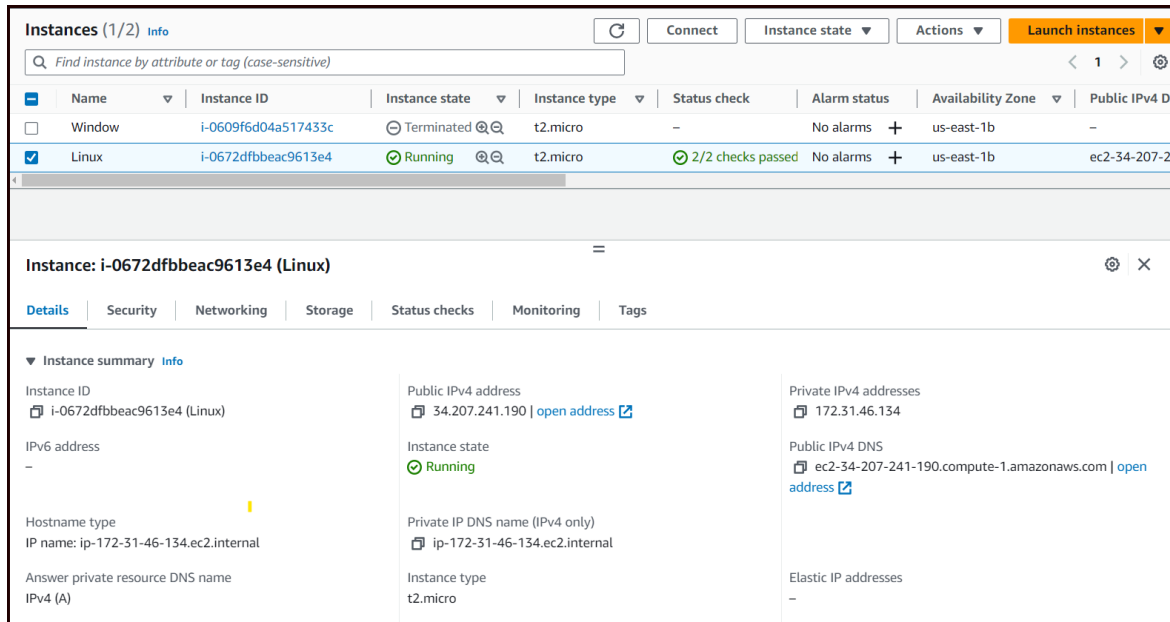


Fig 13

Download PuttyGen. Then open PuTTY Key Generator which is showing interface like Fig 14

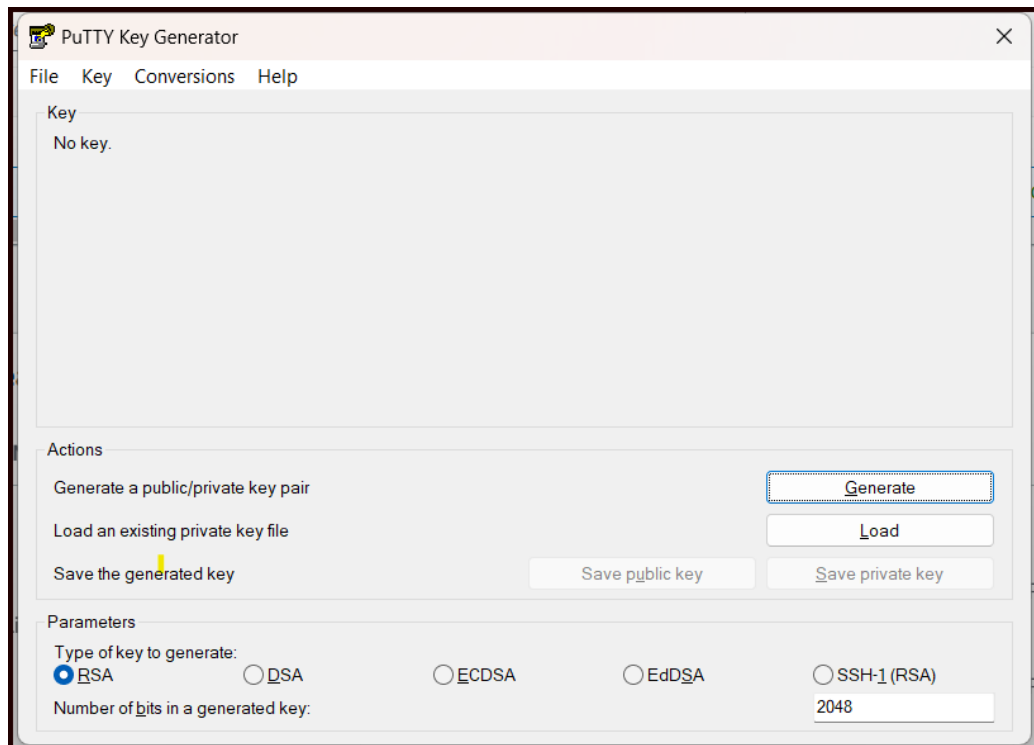


Fig 14

Then click on load, then select the file that we have already downloaded, show in picture 9. Then we will get an interface similar to Figure 15. Then click ok.

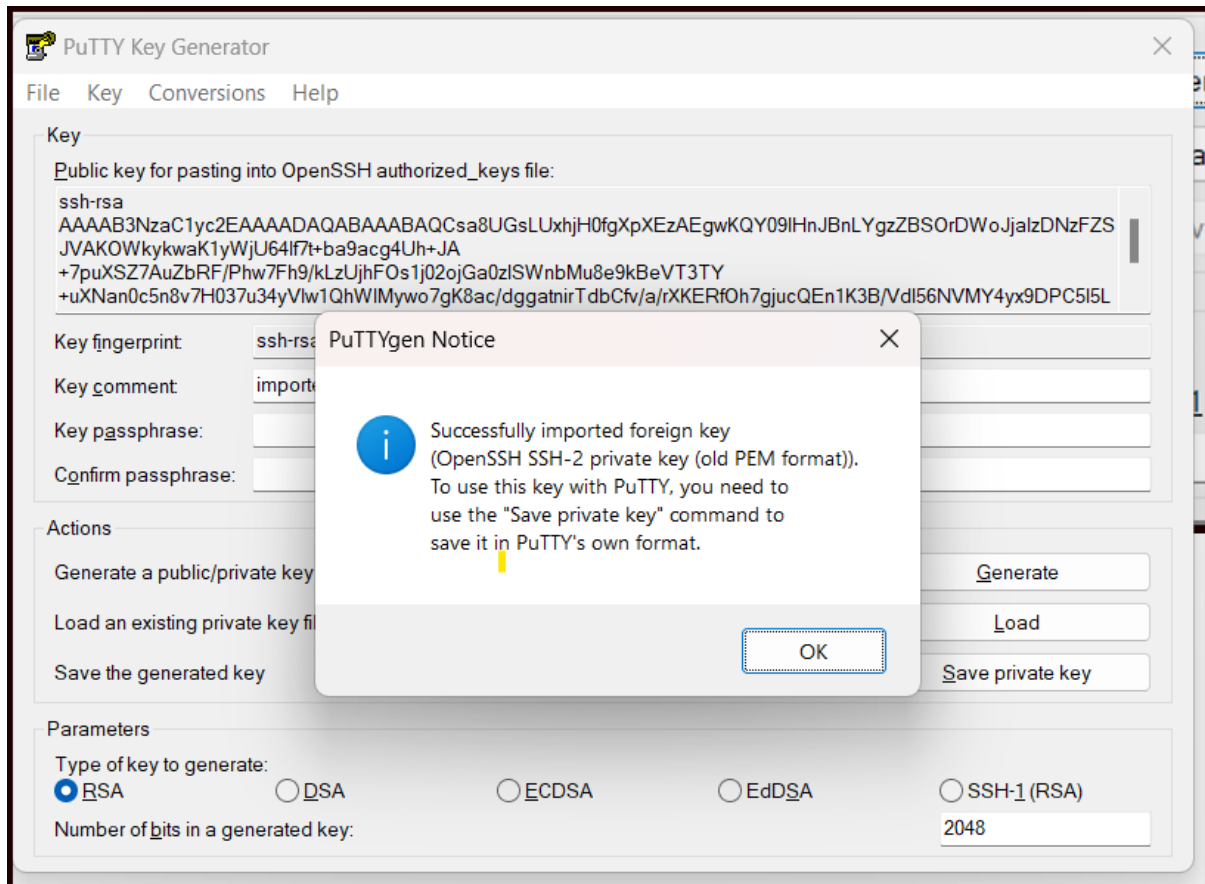


Fig 15

After click on ok we get interface similar to Fig 16. Then Click on “Save private key” and give some name to that file. Here I give that file name is key. And then close the PuTTY key generator.

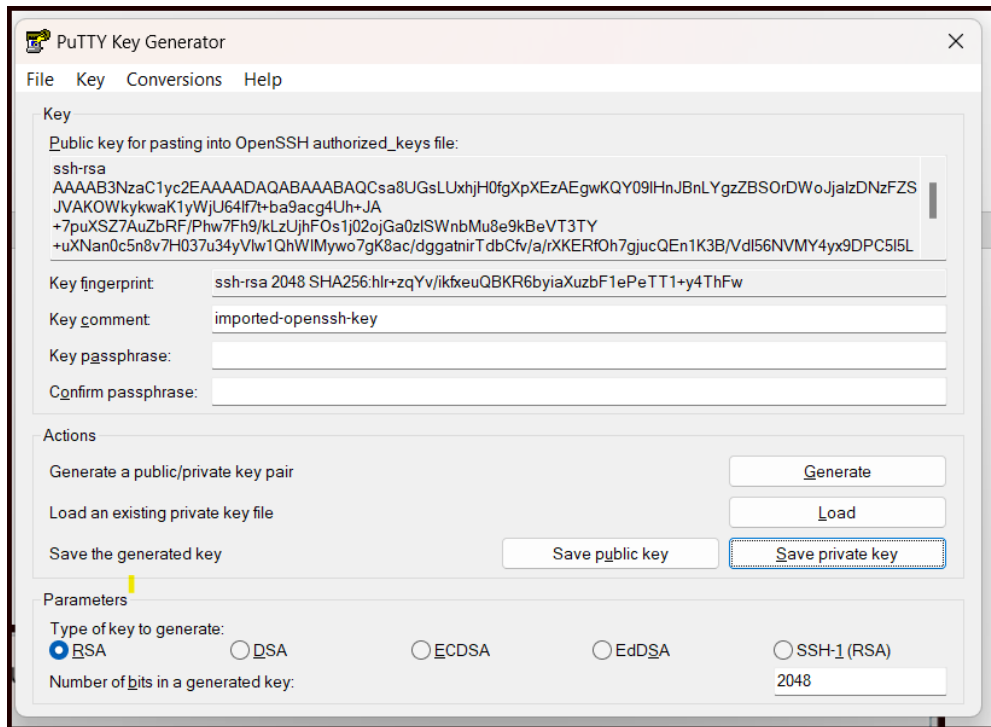
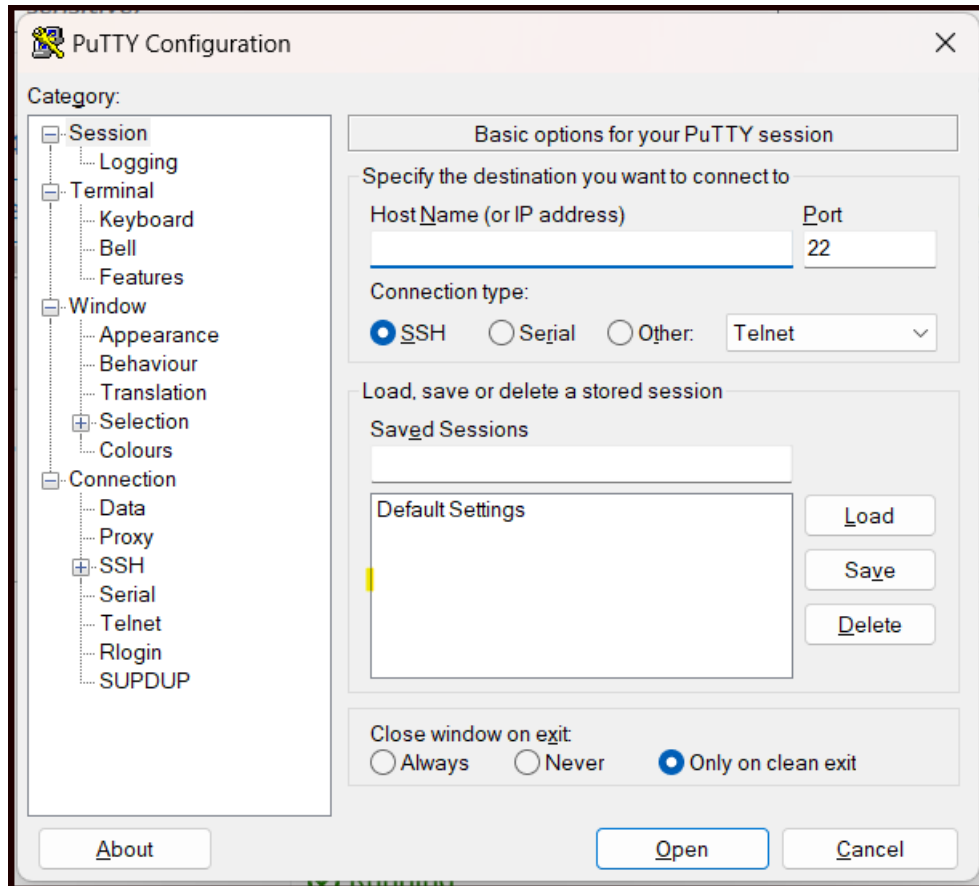


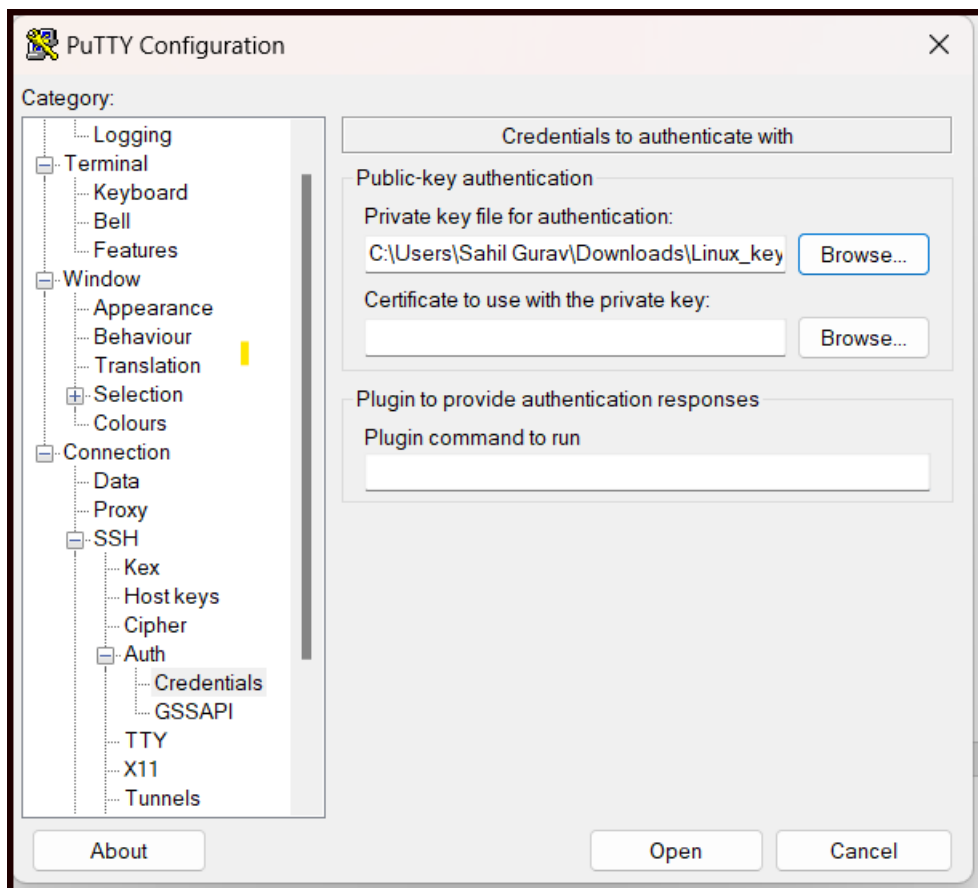
Fig 16

After closing the PuTTY key generator open the PuTTY configuration similar to Fig 17. Copy the public IPv4 address from Fig 13 and then click on SSH -> Auth -> Credentials, then we get an interface like Fig 14.




**Fig 17**

Then select browse button from private key file for authentication and select the file which is already downloaded in Fig 16 we save as file name key. Then click on open.



**Fig 18**

After clicking on Open, the interface like Fig 19 opens. In this we have to type command ec2-user then we get image of bird similar to Fig 19



```
ec2-user@ip-172-31-44-212:~
login as: ec2-user
Authenticating with public key "imported-openssh-key"
#
~\  ####          Amazon Linux 2023
~~~\  #####\
~~~\  #####|
~~~\  \###/      https://aws.amazon.com/linux/amazon-linux-2023
~~~\  V~'  '->
~~~~
~~~.  _
~~~\  /m/'
[ec2-user@ip-172-31-44-212 ~]$
```

**Fig 19**

Enter given commands as shown in Fig 20

**sudo su** :- This command gives you root privileges, which are required to install and configure the Apache web server

**yum update -y** :- updates all installed packages and their dependencies

**yum install httpd -y** :- installs the Apache web server. The -y flag tells yum to answer yes to all prompts

```
root@ip-172-31-44-212/home/ec2-user
[ec2-user@ip-172-31-44-212 ~]$ sudo su
[root@ip-172-31-44-212 ec2-user]# yum update -y
Last metadata expiration check: 0:14:25 ago on Fri Sep  8 07:30:22 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-44-212 ec2-user]# yum install httpd -y
Last metadata expiration check: 0:14:44 ago on Fri Sep  8 07:30:22 2023.
Dependencies resolved.
=====
Package                Arch      Version                Repository              Size
=====
Installing:
httpd                  x86_64    2.4.56-1.amzn2023     amazonlinux             48 k
Installing dependencies:
apr                    x86_64    1.7.2-2.amzn2023.0.2  amazonlinux            129 k
apr-util              x86_64    1.6.3-1.amzn2023.0.1  amazonlinux            98 k
generic-logos-httpd   noarch    10.0.0-12.amzn2023.0.3 amazonlinux            19 k
httpd-core            x86_64    2.4.56-1.amzn2023     amazonlinux            1.4 M
httpd-filesystem      noarch    2.4.56-1.amzn2023     amazonlinux            15 k
httpd-tools           x86_64    2.4.56-1.amzn2023     amazonlinux            82 k
libbrotli             x86_64    1.0.9-4.amzn2023.0.2  amazonlinux            315 k
mailcap               noarch    2.1.49-3.amzn2023.0.3 amazonlinux            33 k
Installing weak dependencies:
apr-util-openssl     x86_64    1.6.3-1.amzn2023.0.1  amazonlinux            17 k
mod_http2             x86_64    2.0.11-2.amzn2023     amazonlinux            150 k
mod_lua               x86_64    2.4.56-1.amzn2023     amazonlinux            62 k
Transaction Summary
-----
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): httpd-2.4.56-1.amzn2023.x86_64.rpm 625 kB/s | 48 kB 00:00
(2/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.r 3.5 MB/s | 315 kB 00:00
(3/12): apr-1.7.2-2.amzn2023.0.2.x86_64.rpm 5.0 MB/s | 129 kB 00:00
(4/12): mod_lua-2.4.56-1.amzn2023.x86_64.rpm 2.8 MB/s | 62 kB 00:00
(5/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rp 4.2 MB/s | 98 kB 00:00
(6/12): mod_http2-2.0.11-2.amzn2023.x86_64.rpm 4.9 MB/s | 150 kB 00:00
(7/12): httpd-core-2.4.56-1.amzn2023.x86_64.rpm 8.5 MB/s | 1.4 MB 00:00
(8/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x 776 kB/s | 17 kB 00:00
(9/12): httpd-tools-2.4.56-1.amzn2023.x86_64.rp 1.8 MB/s | 82 kB 00:00
```

Fig 20

Enter given commands as shown in Fig 21

**cd /var/www/html** :- changes to the /var/www/html directory, which is the default directory for web content

**echo "Welcome sahil"> index.html** :- creates a file called index.html and writes the text "Welcome sahil" to it.

**ls** :- lists the contents of the current directory

**service httpd start** :- starts the Apache web server

**chkconfig httpd on** :- enables the Apache web server to start automatically when the system boots

```
root@ip-172-31-44-212:/var/www/html
[root@ip-172-31-44-212 ec2-user]# cd /var/www/html
[root@ip-172-31-44-212 html]#
[root@ip-172-31-44-212 html]# echo "Welcome sahil"> index.html
[root@ip-172-31-44-212 html]# ls
index.html
[root@ip-172-31-44-212 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-44-212 html]# chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service -> /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-44-212 html]#
```

Fig 21

Then open the inbound rule and add rule. In that Type is "HTTP" and Source is "Anywhere IPV4" as shown in Fig 22

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional info
sg-089d34084aba9aa3f	SSH	TCP	22	Custom	
-	HTTP	TCP	80	Anywh...	

Fig 22

Copy the public IPV4 address and paste on new tab with prefix http:/ followed by IPV4 address as shown in Fig 23



Fig 23

We get an interface like in Fig 24. Then click on "Instance State" select the option as per your choice. Here I select "Terminate Instance".

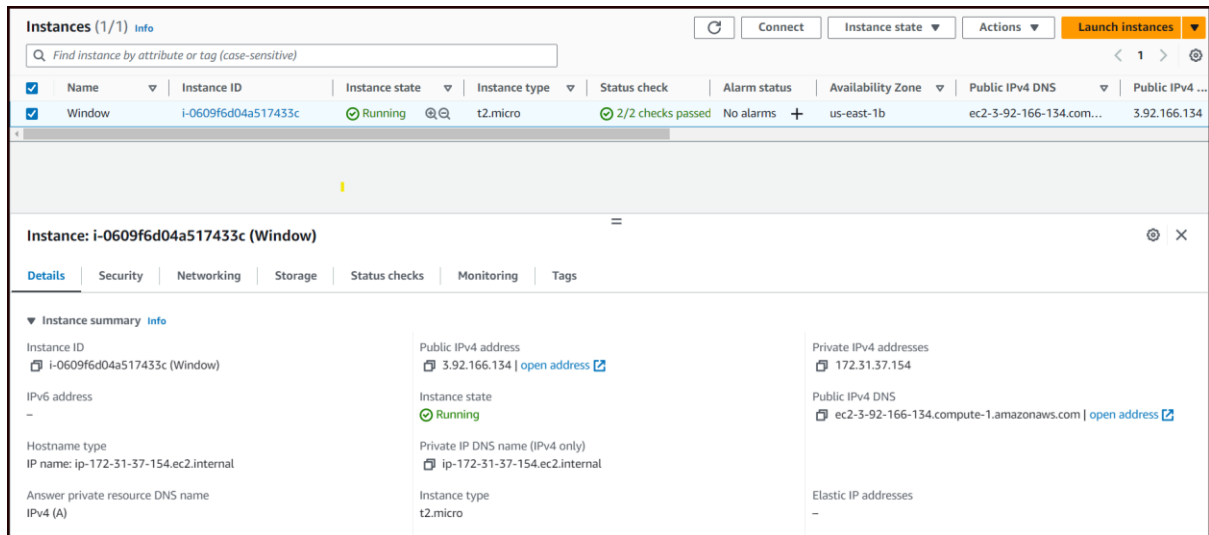


Fig 24