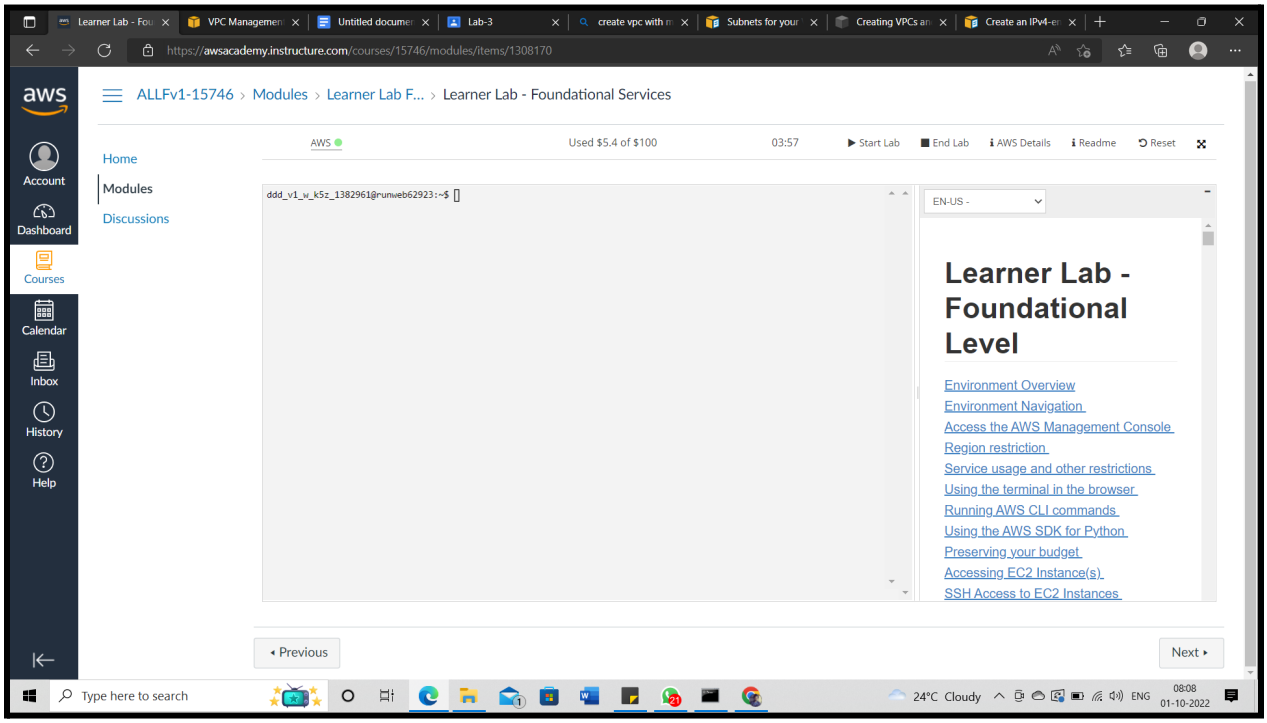
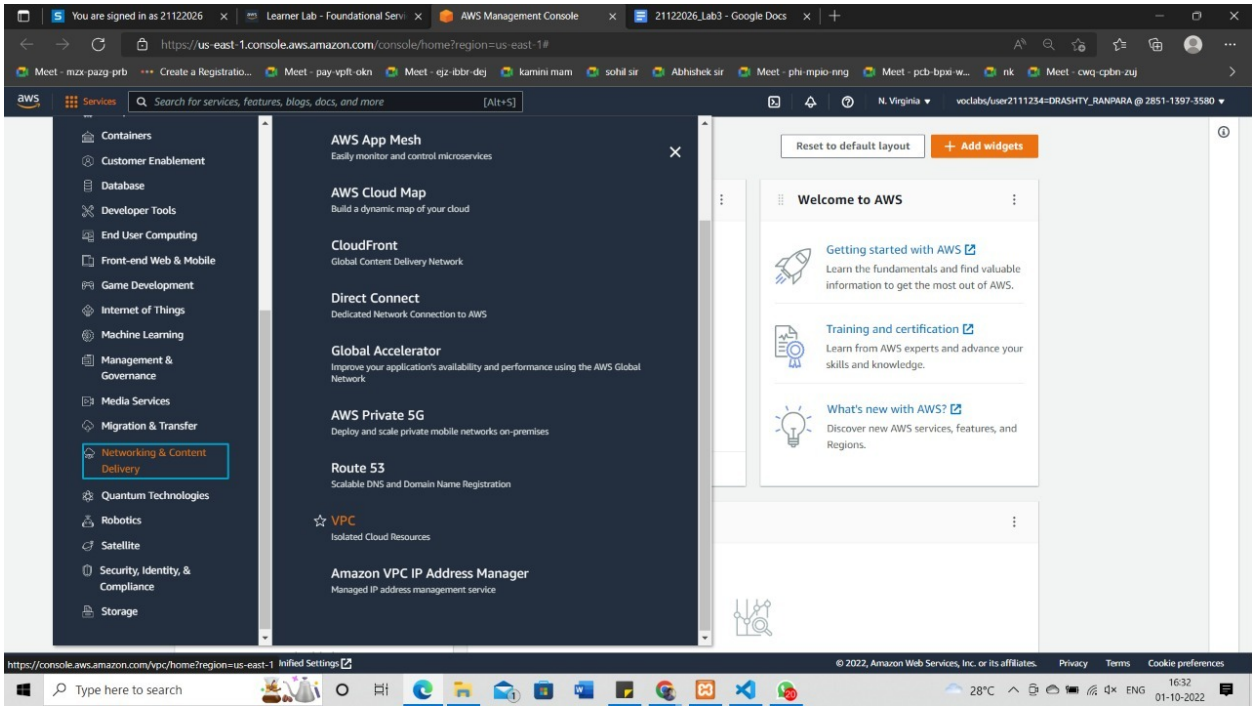
Name: - Drashty Ranpara Roll No: - 21122026

1. Create a VPC with minimum 2 subnets in any region.

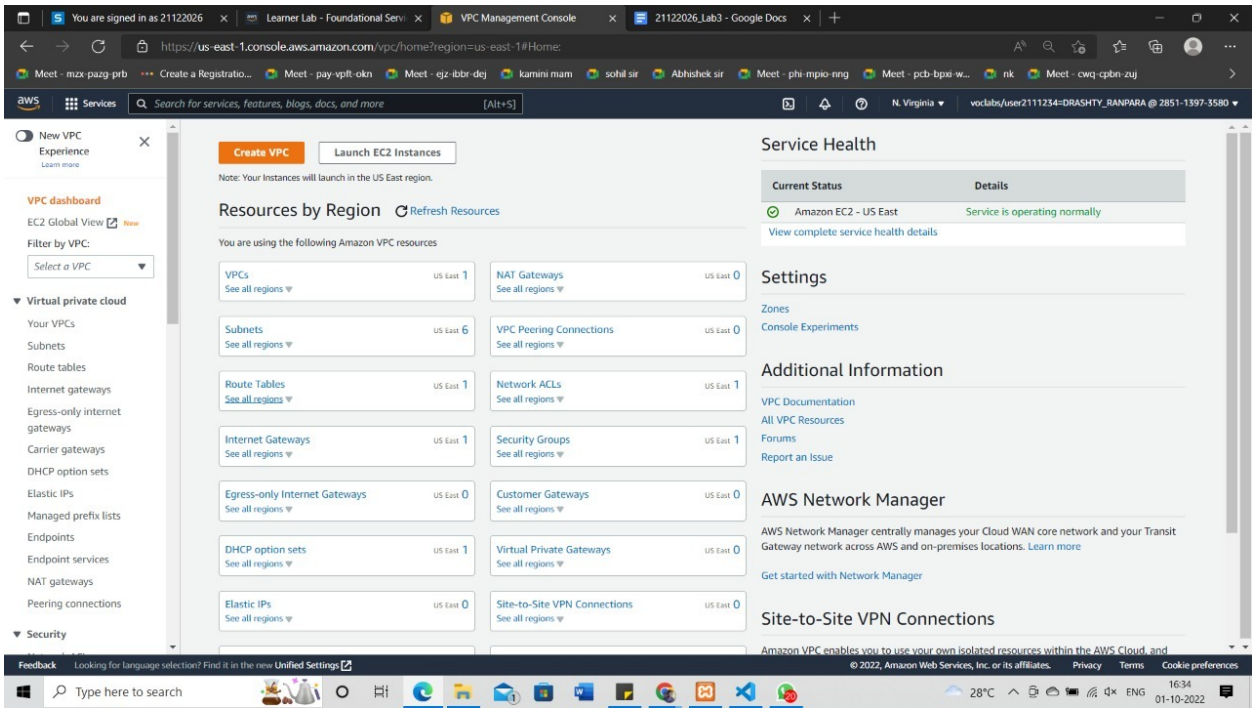
- Open the lab and start the lab



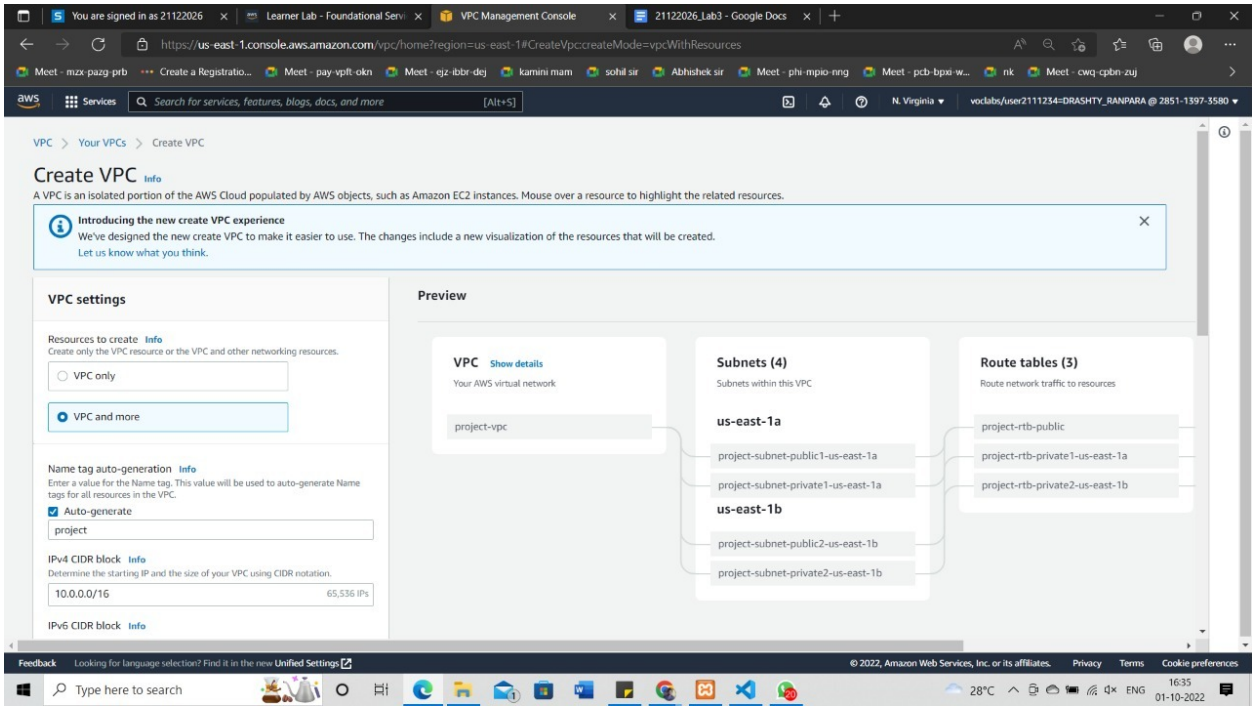
- Navigate to the Network & Content delivery and select the vpc



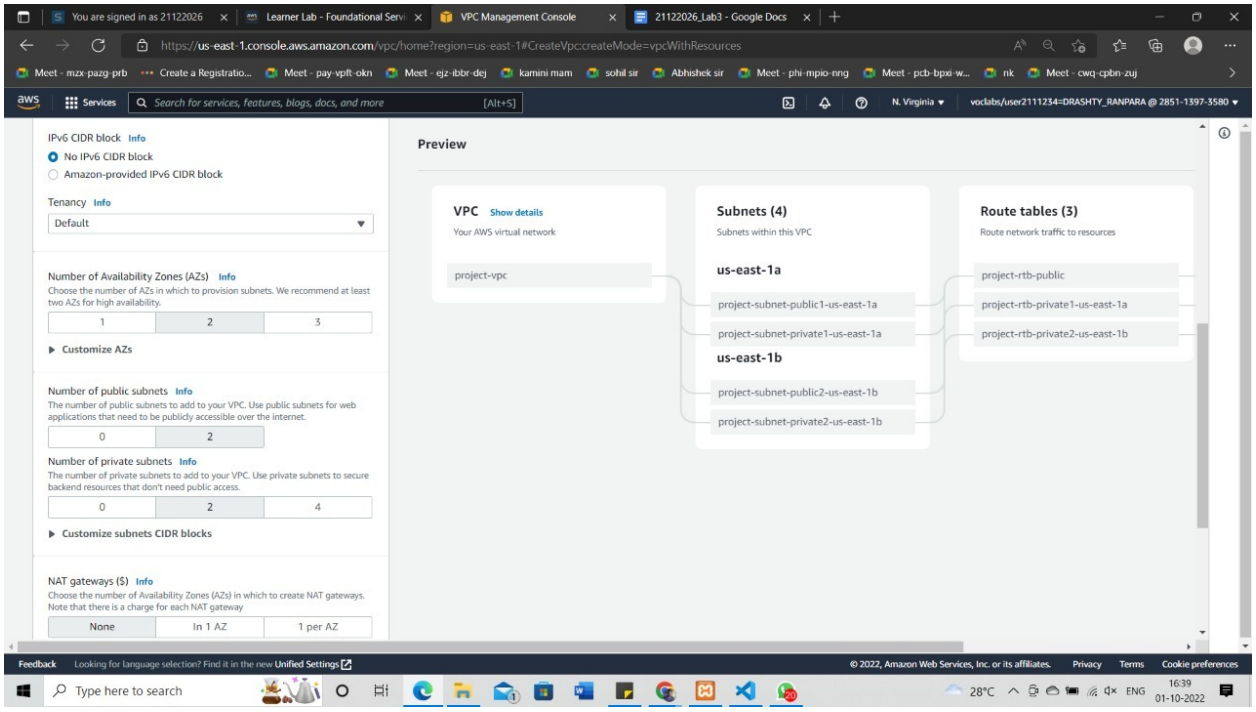
- Click on the Create VPC



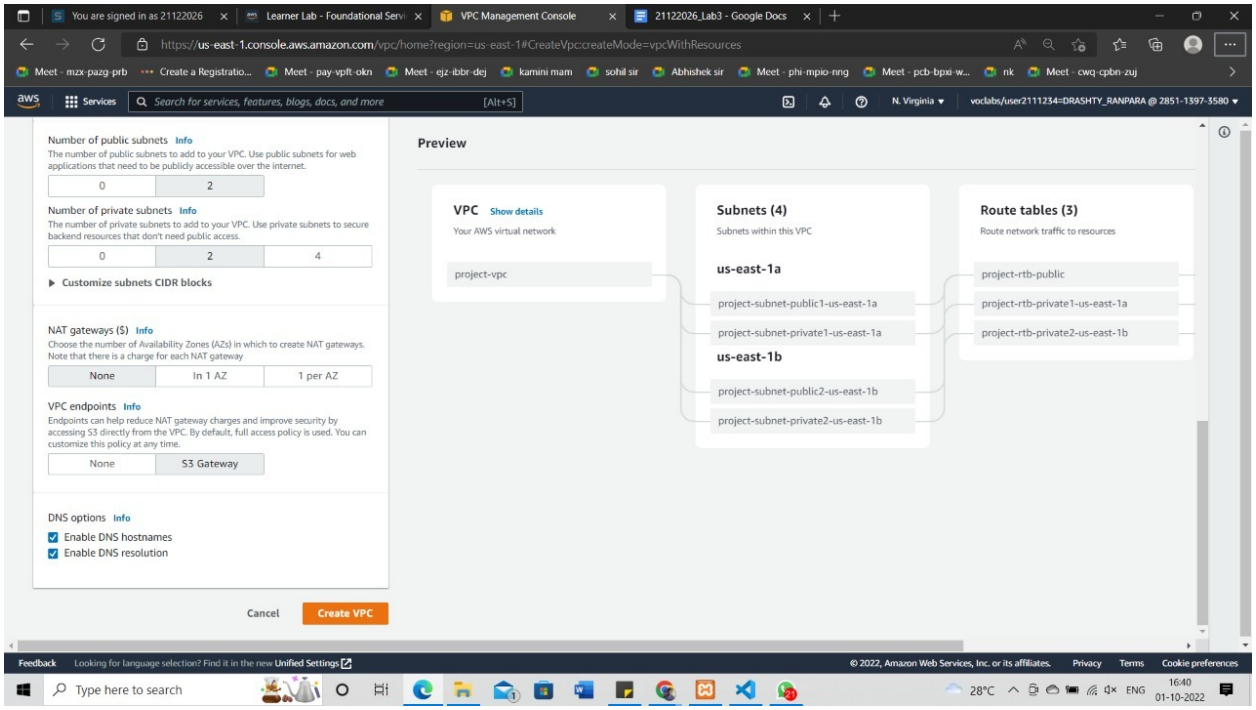
- Select the VPC and more give the name of the vpc and set the CIDR block for the IPv4.



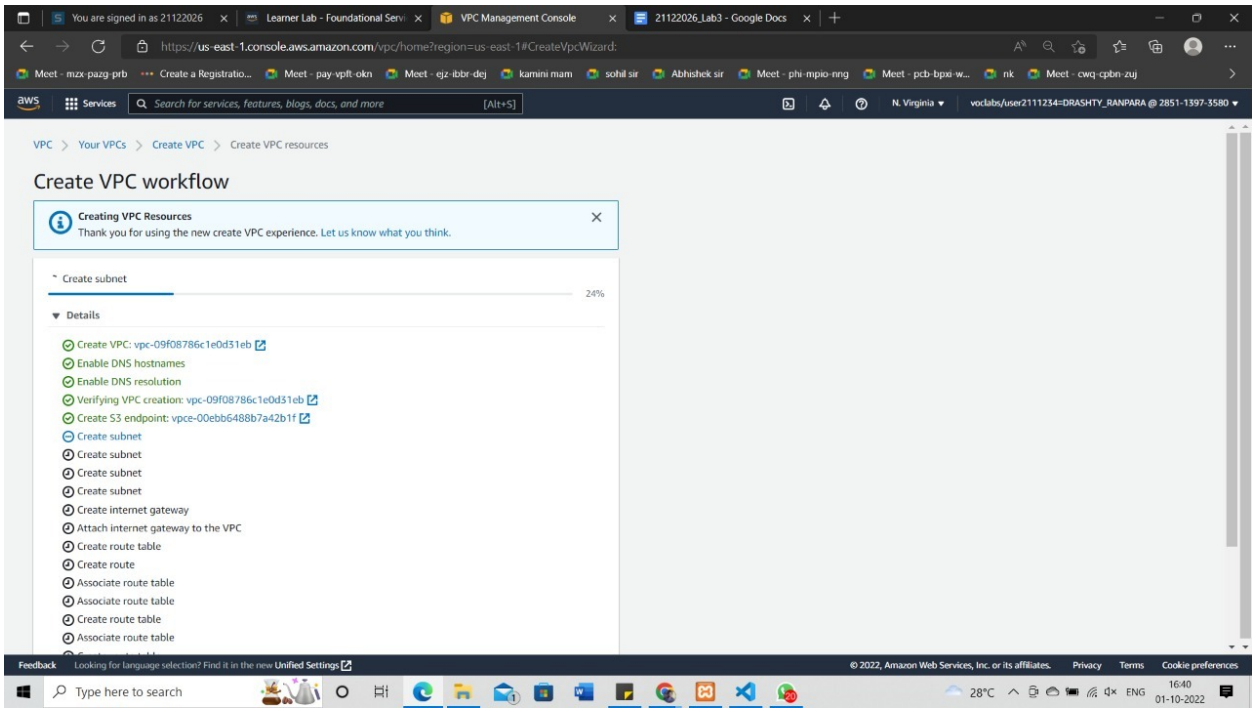
- Select the No IPv6 CIDR block and select 2 for the number of the availability zone and select the availability zone for both the region. Select the 2 public subnets and 2 private subnet for the ip address.



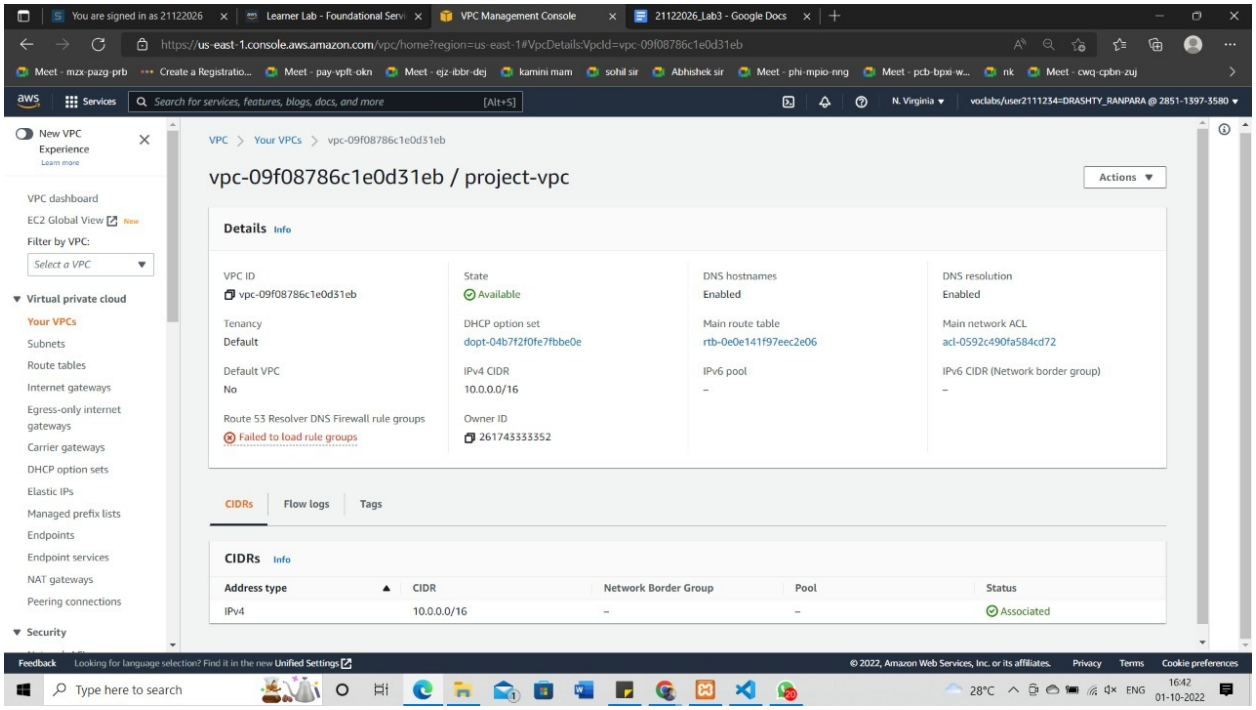
- You can Customize the subnet CIDR block for the availability zone. And then click on the Create VPC.



- After clicking on the create vpc button you will get the screen like this.



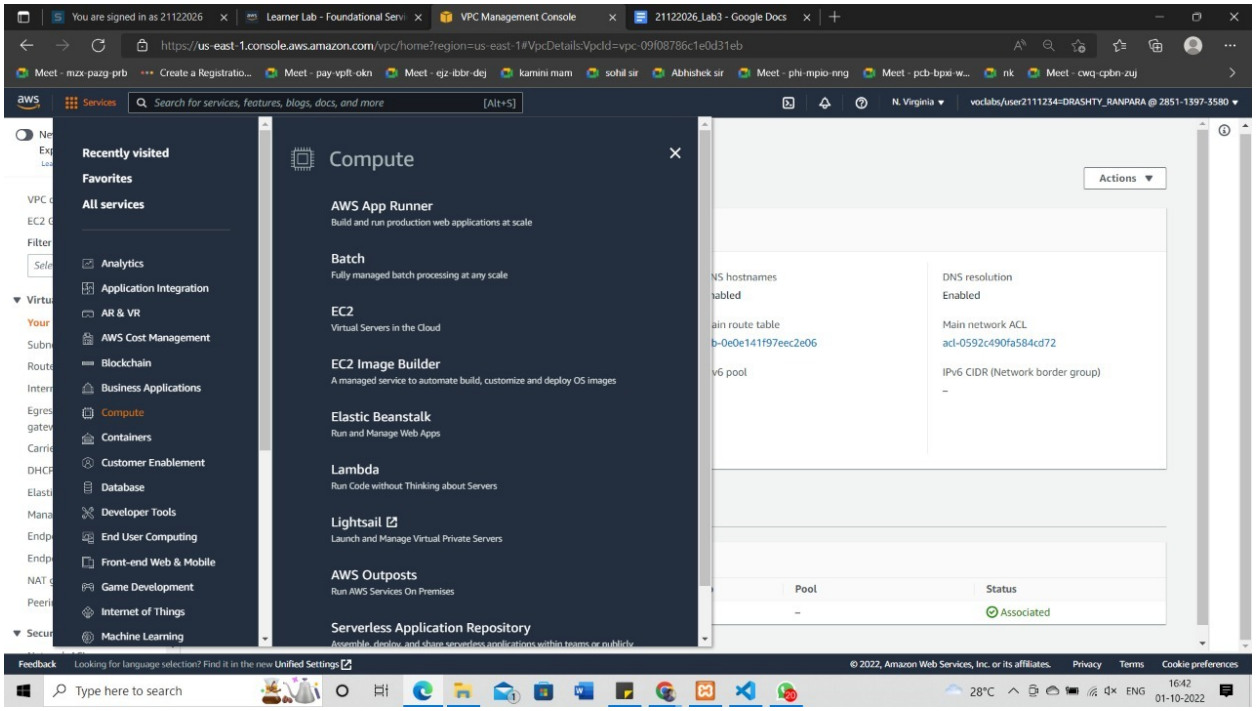
- After Completion of the process you can see the details of the vpc.



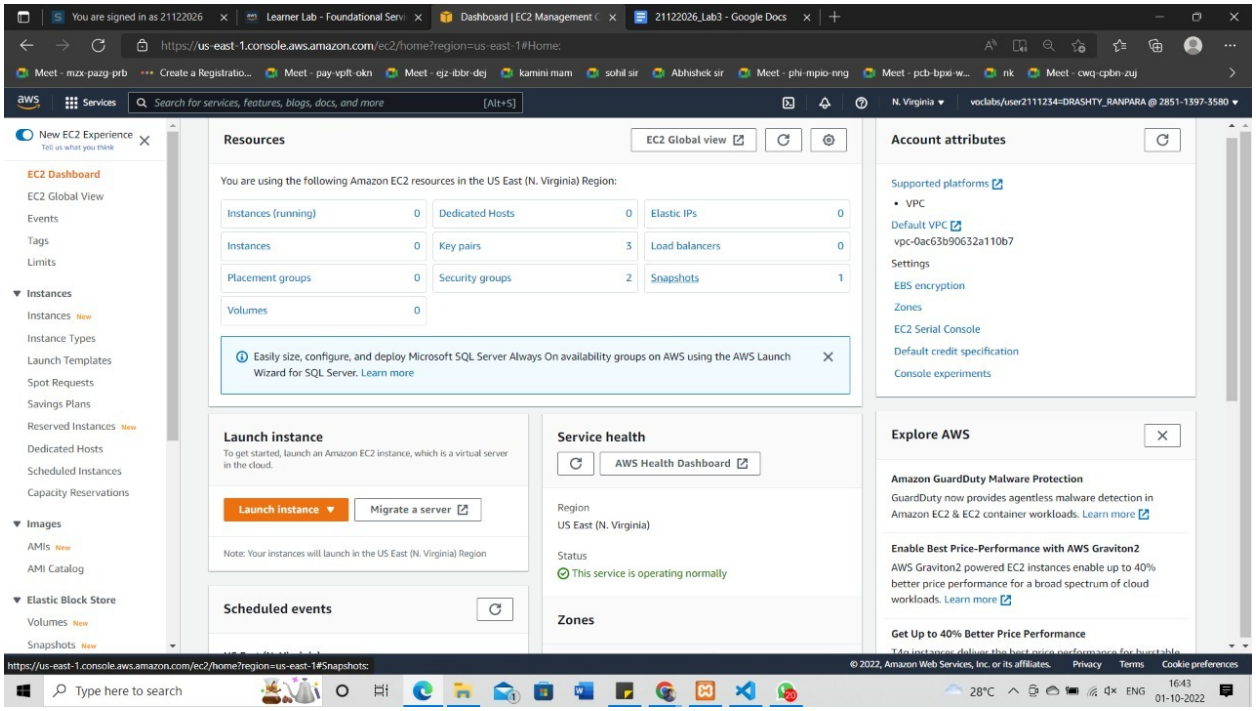
- Hurray!!!! Your First Task Done.

2. Attach 2 instances to the first subnet in the VPC. Ensure public IPs are assigned to both the instances.

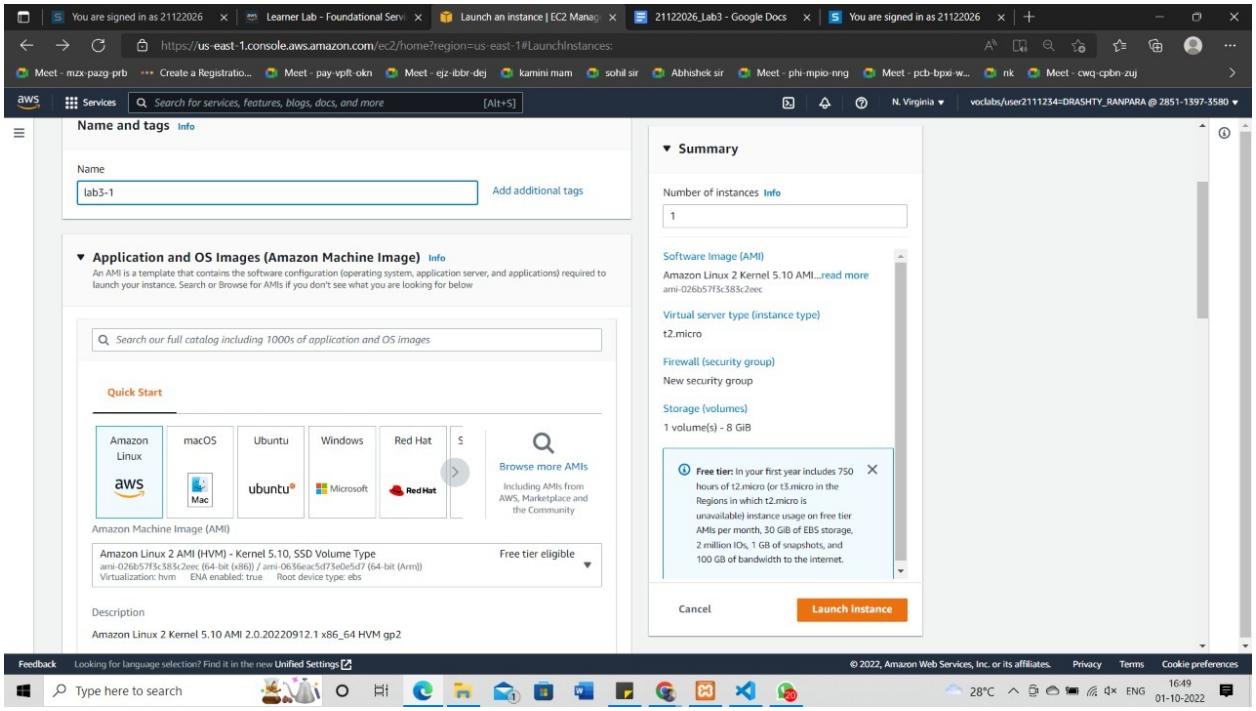
- First navigate to the compute-> EC2



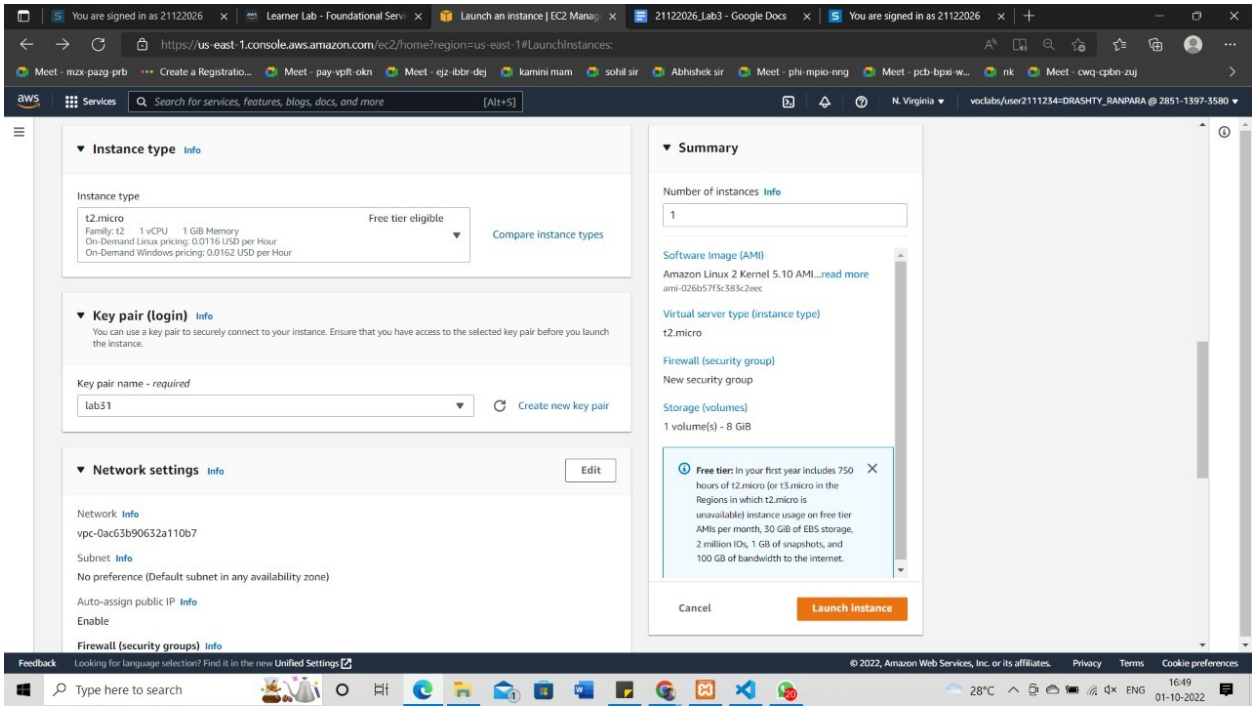
- You will get the Screen like this and CLick on the Launch Instance



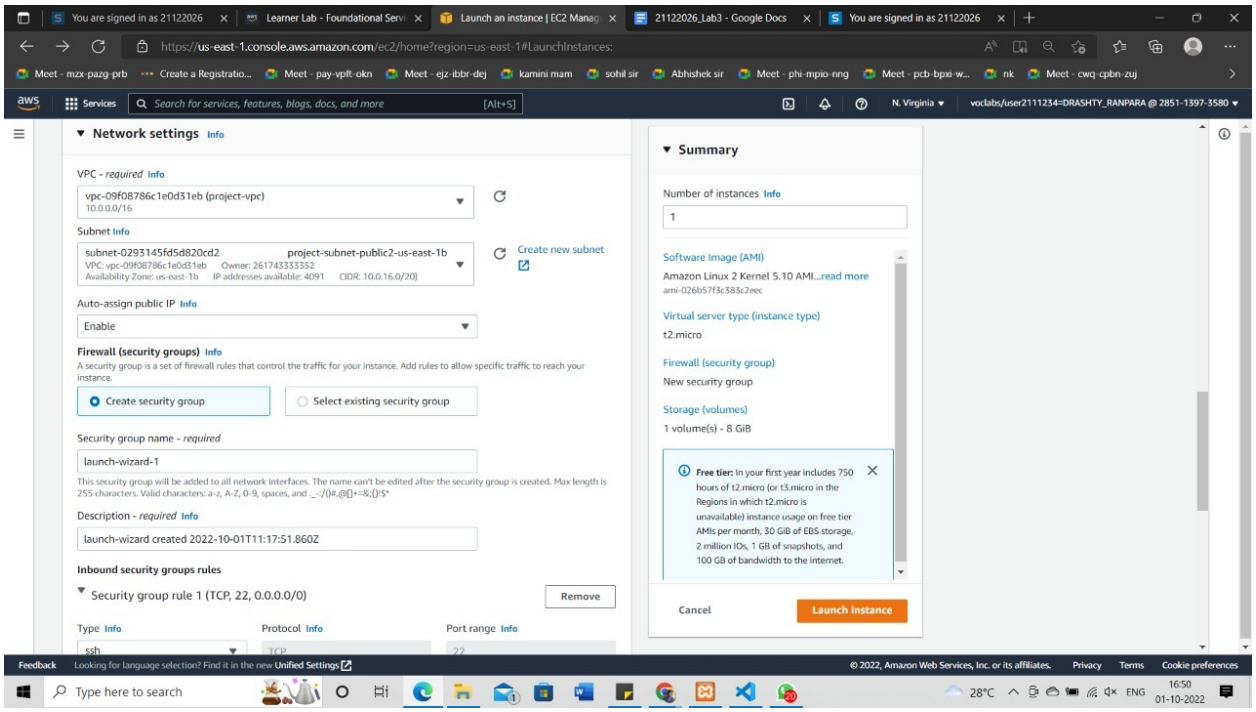
- Give the name of the instance Select the Application and OS Image for the instance



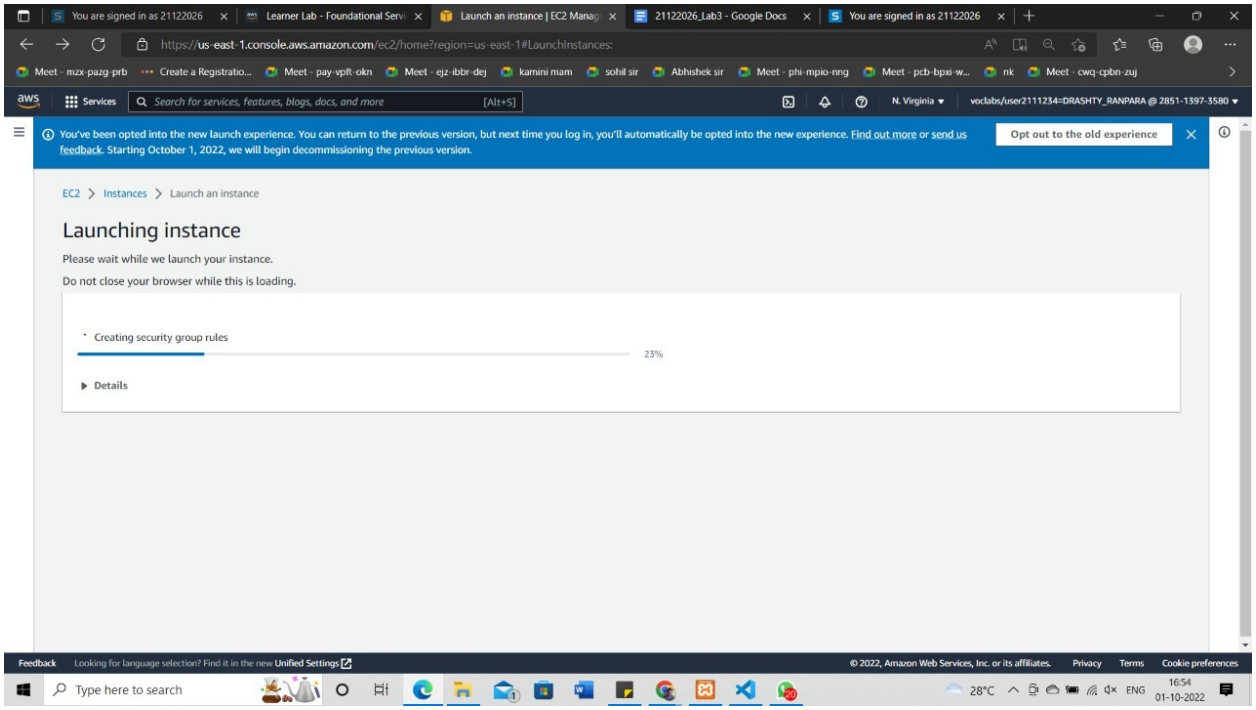
- Create the key value pair



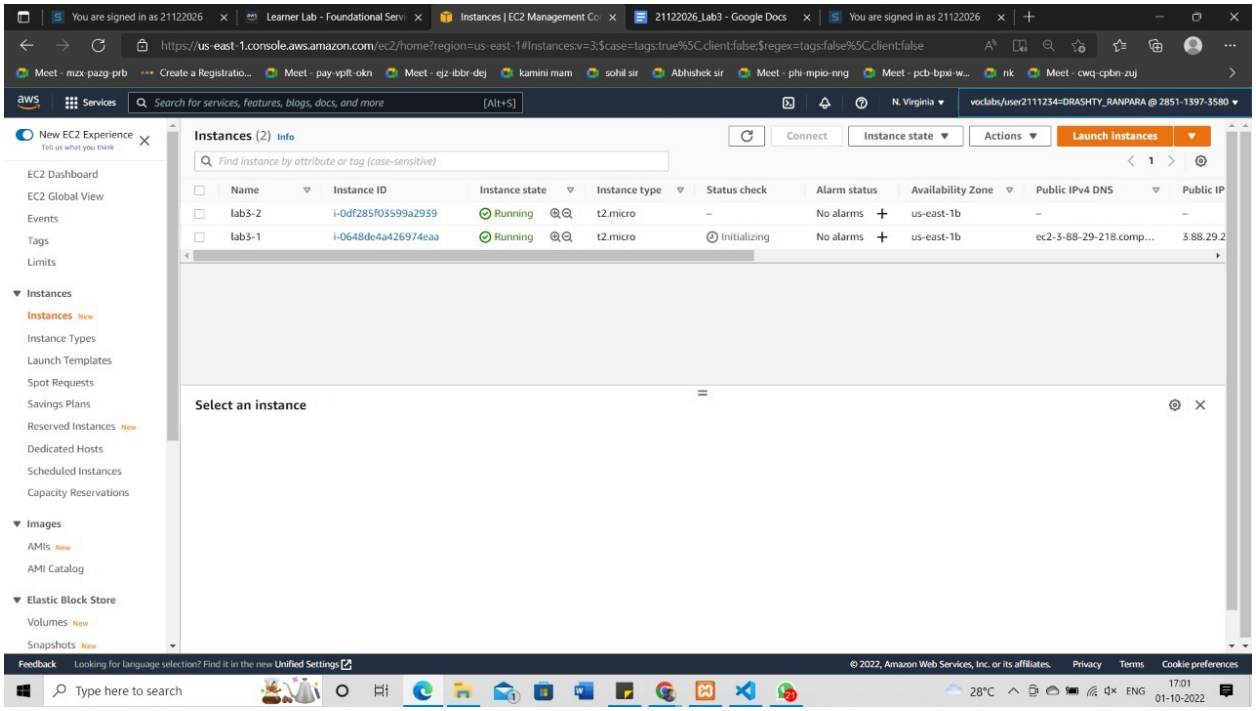
- Change the network setting. Select the VPC which you created and select the public subnet and Enable the Auto Assign public ip.



- After setting all these things click on the Launch Instance. And you will get the Screen like this.



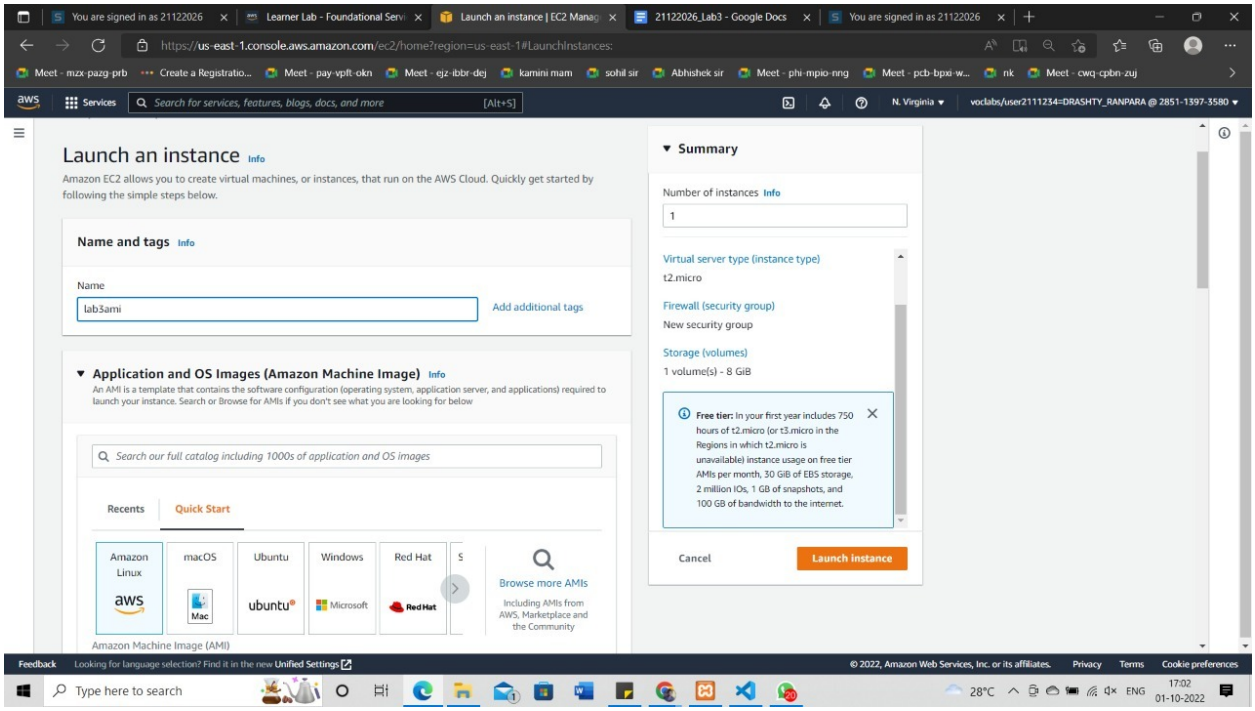
- Again same create second instance with the different tag but remain same the network setting we did for the first instance.
- After creating the second instance you will see the instances.



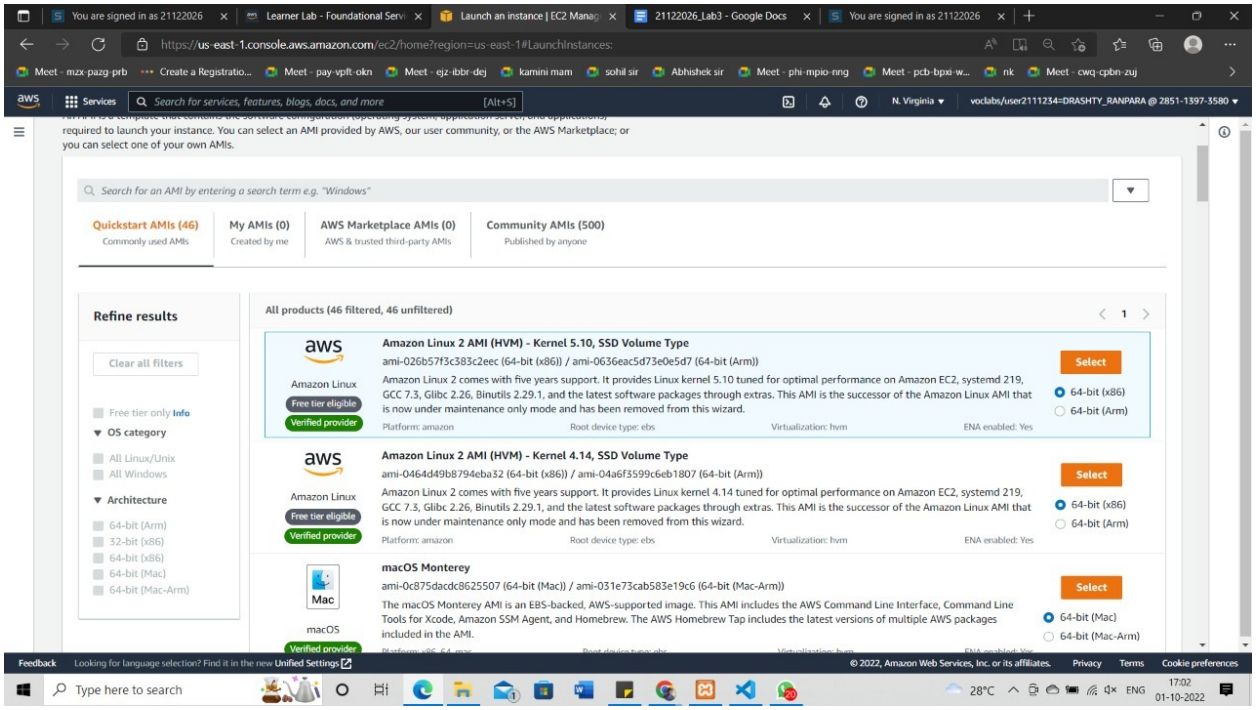
- Hurray!!! Task 2 Now Completed

3. Create an AMI of your own and create 2 instances from the AMI and attach to the second subnet, also provide S3 full access policy to the existing role(LabRole) and attach the LabRole to both the instances.

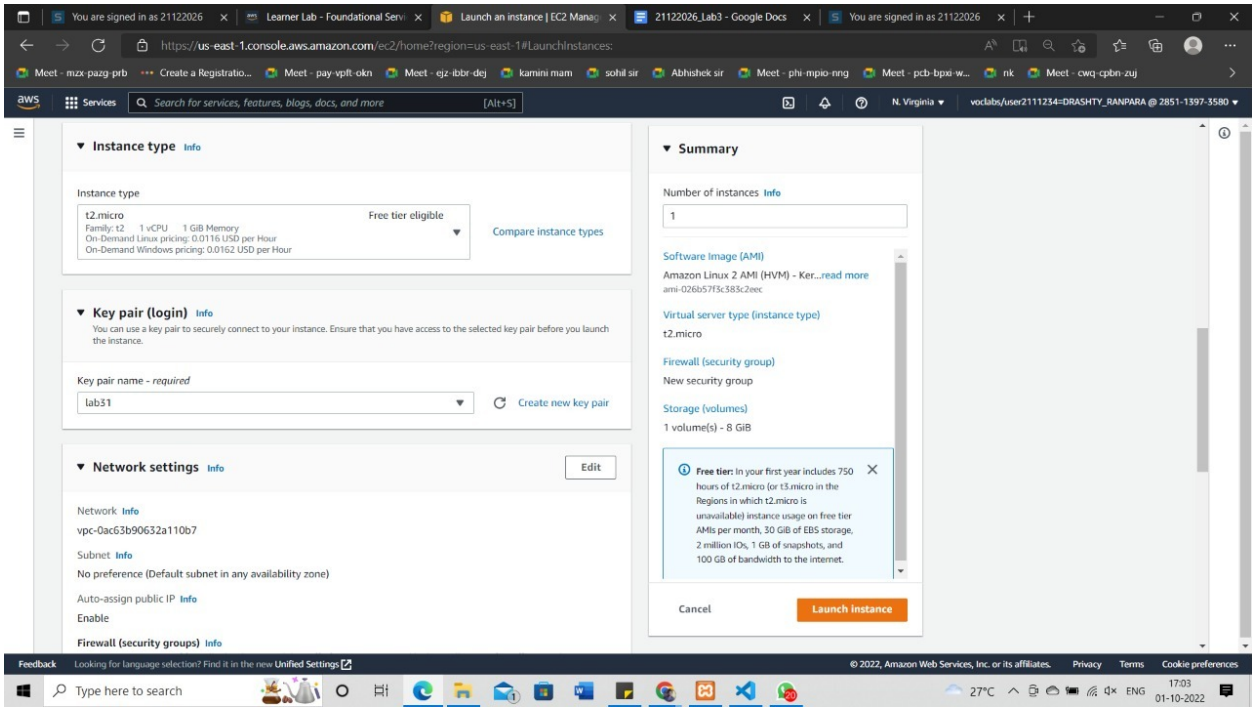
- Again Navigate to the computer->EC2.
- Click on the Launch Instance



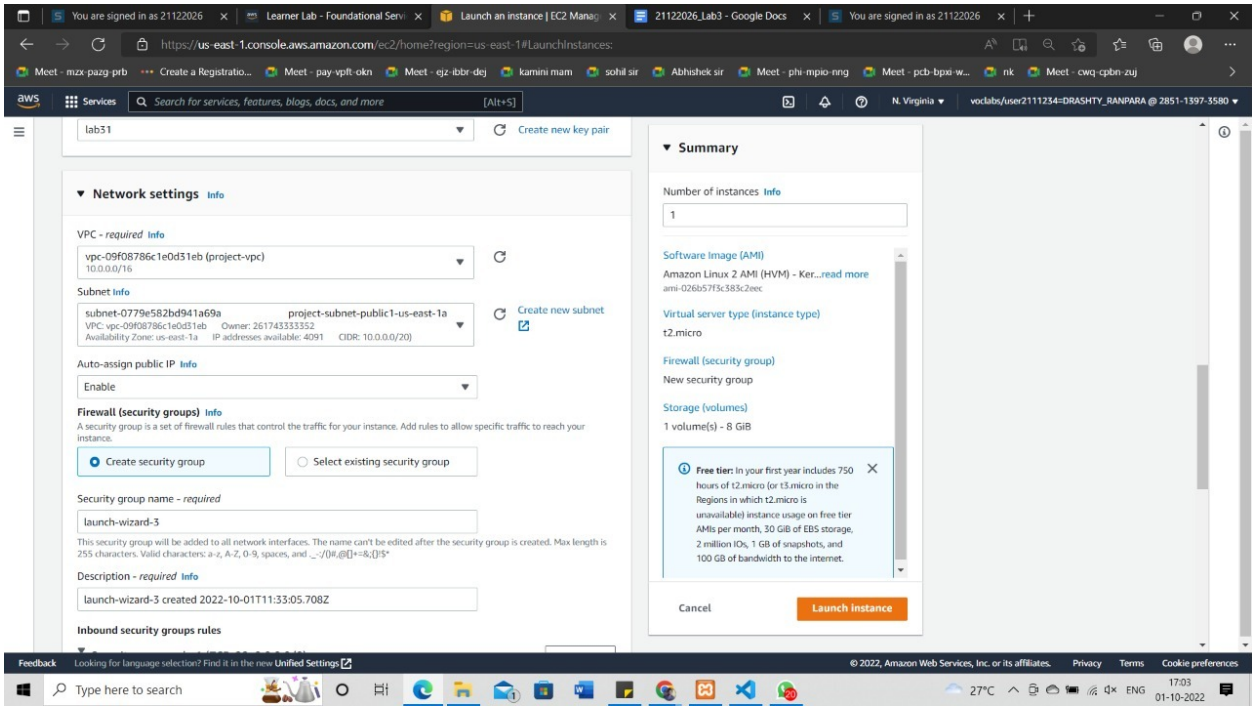
- Give the ame to the ami instance
- In the application and OS Images section click on the search button **Browse more AMIs.**
- Select the any ami for the creating your own ami.



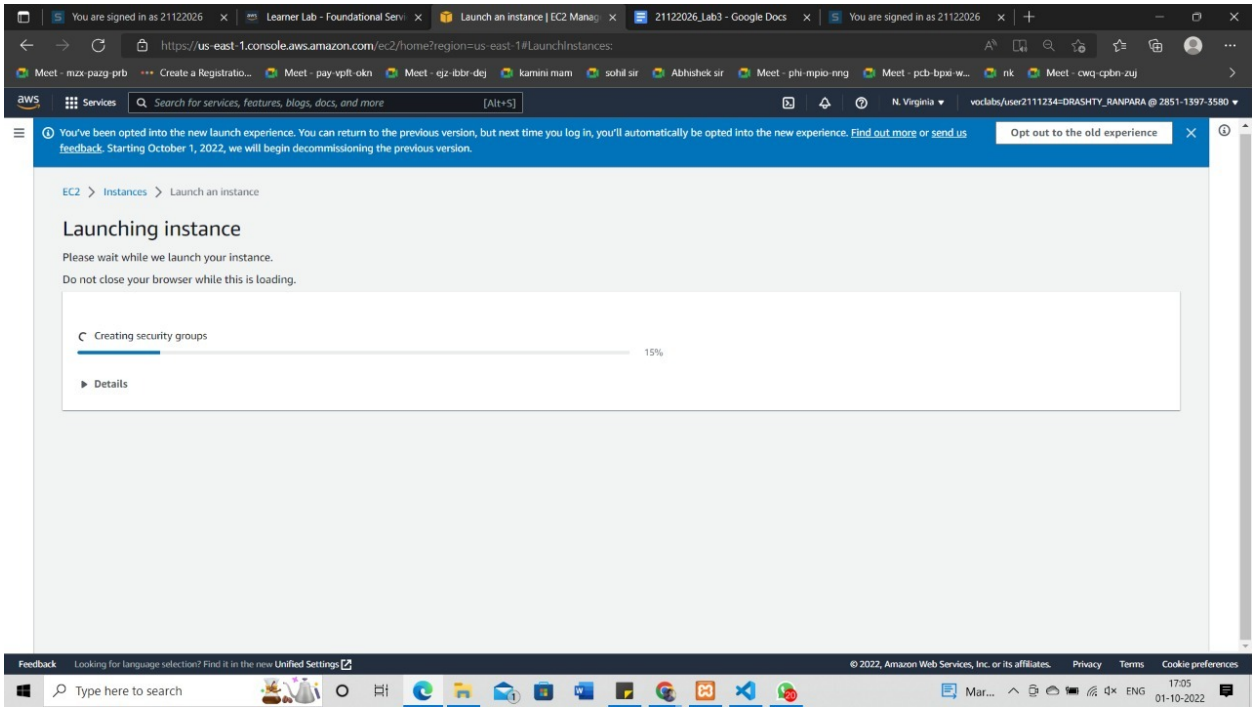
- Create or select the key value pair



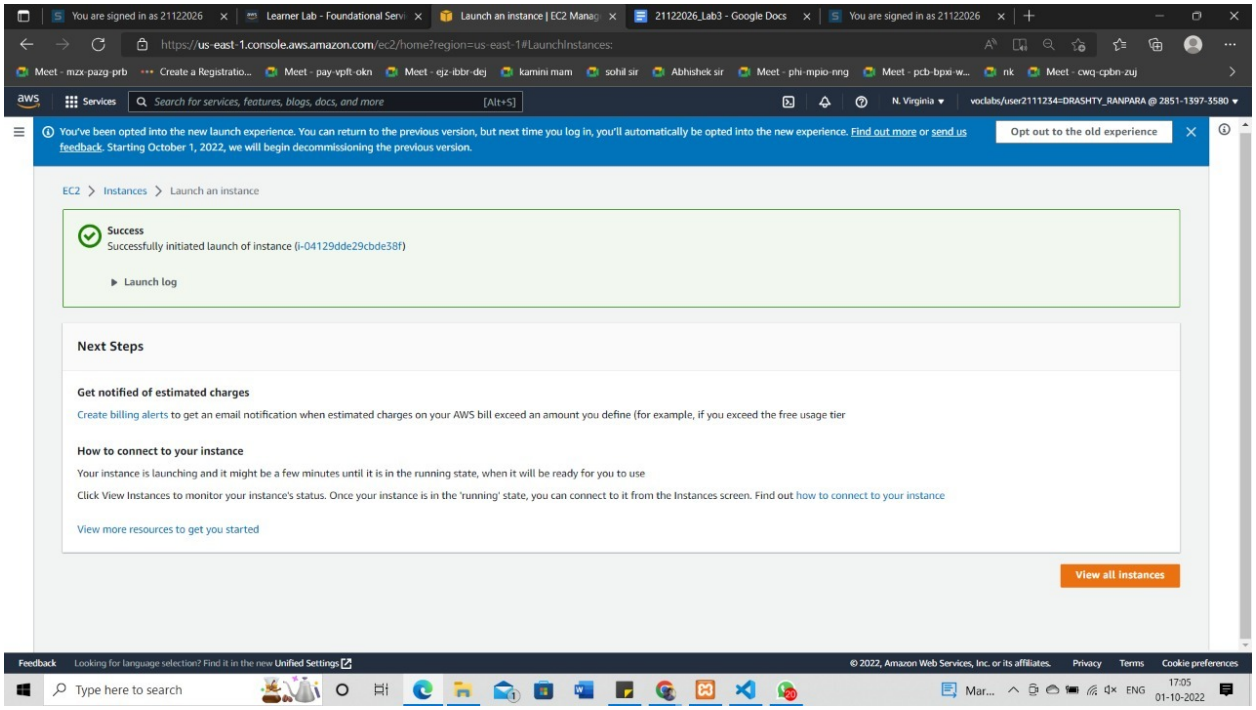
- Click on Edit in Network settings.
- In that section in VPC select the VPC which you created.
- In the subnet select the different region public subnet from the list.



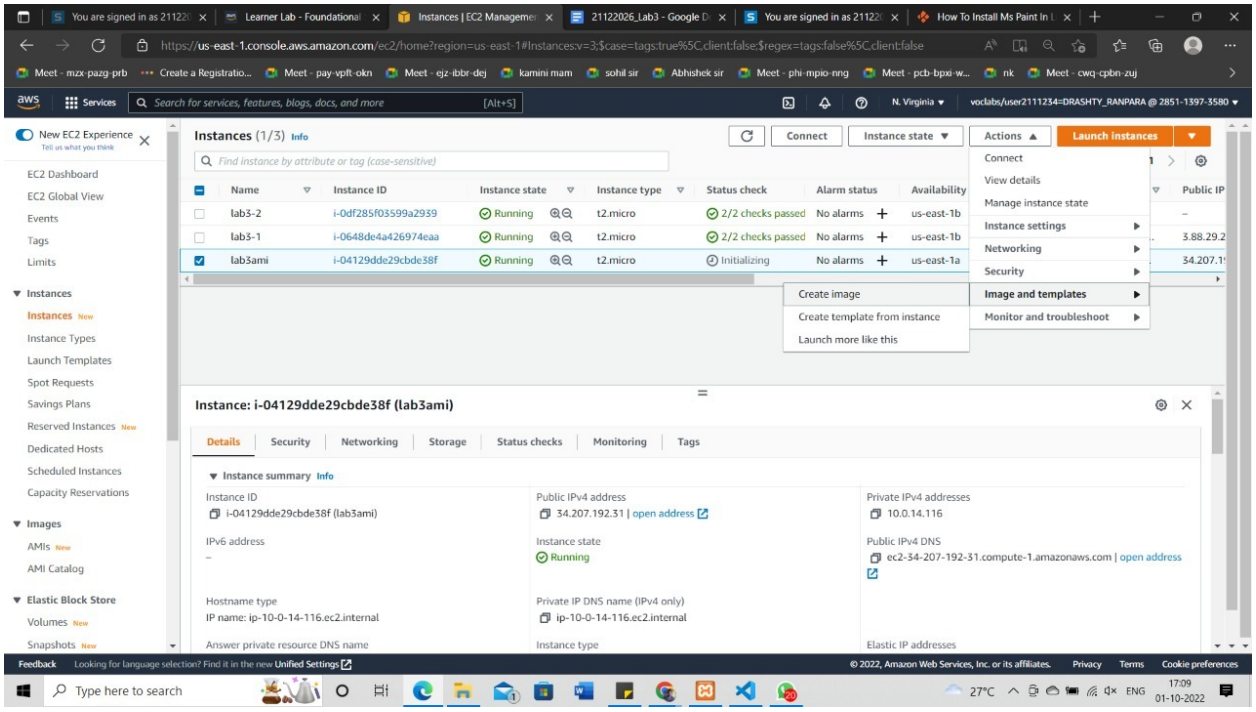
- After all the settings click on the Launch Instance button.



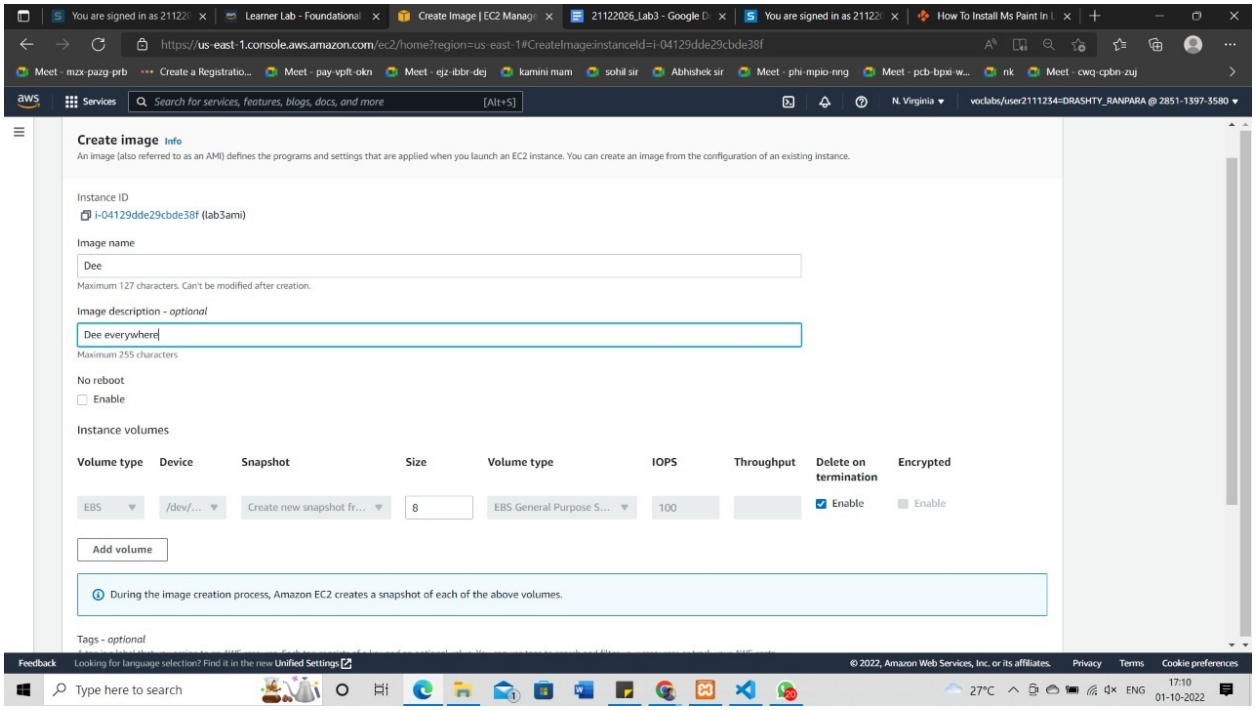
- After successfully creating the ami you will get the following display.



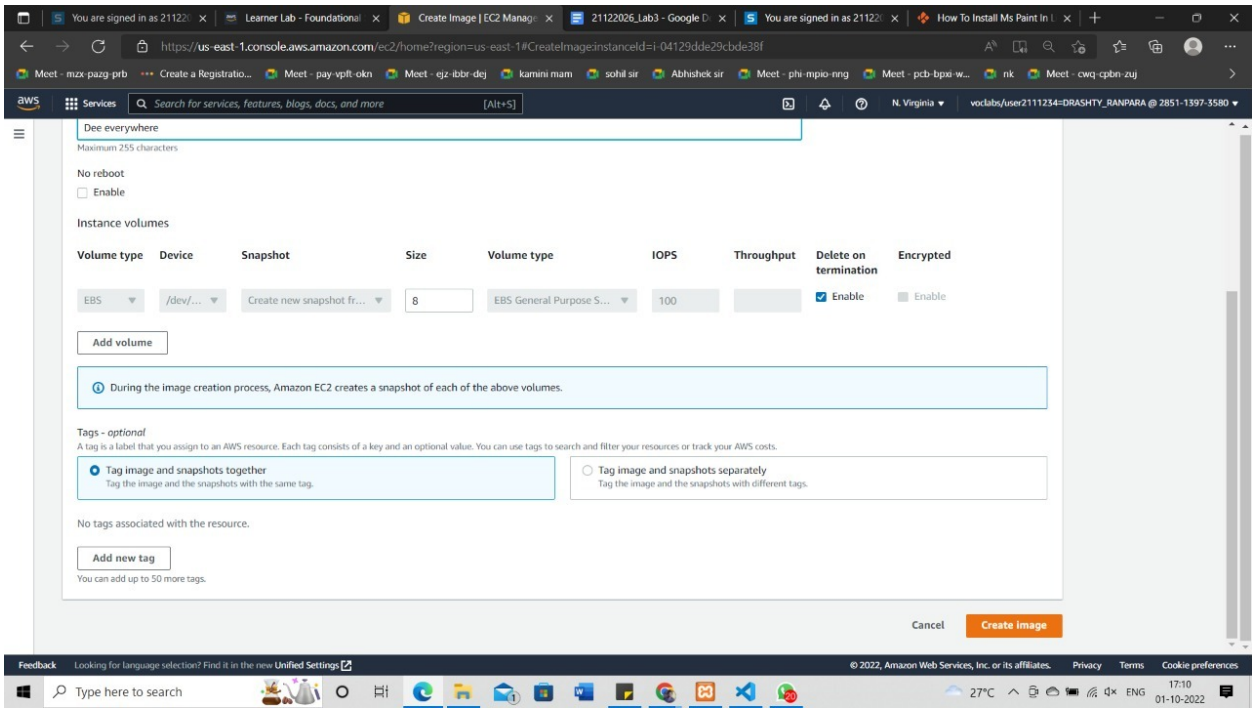
- Now go to the instances click on the ami and then click on the actions button in that section click on the Images and templates in that click on the create image.



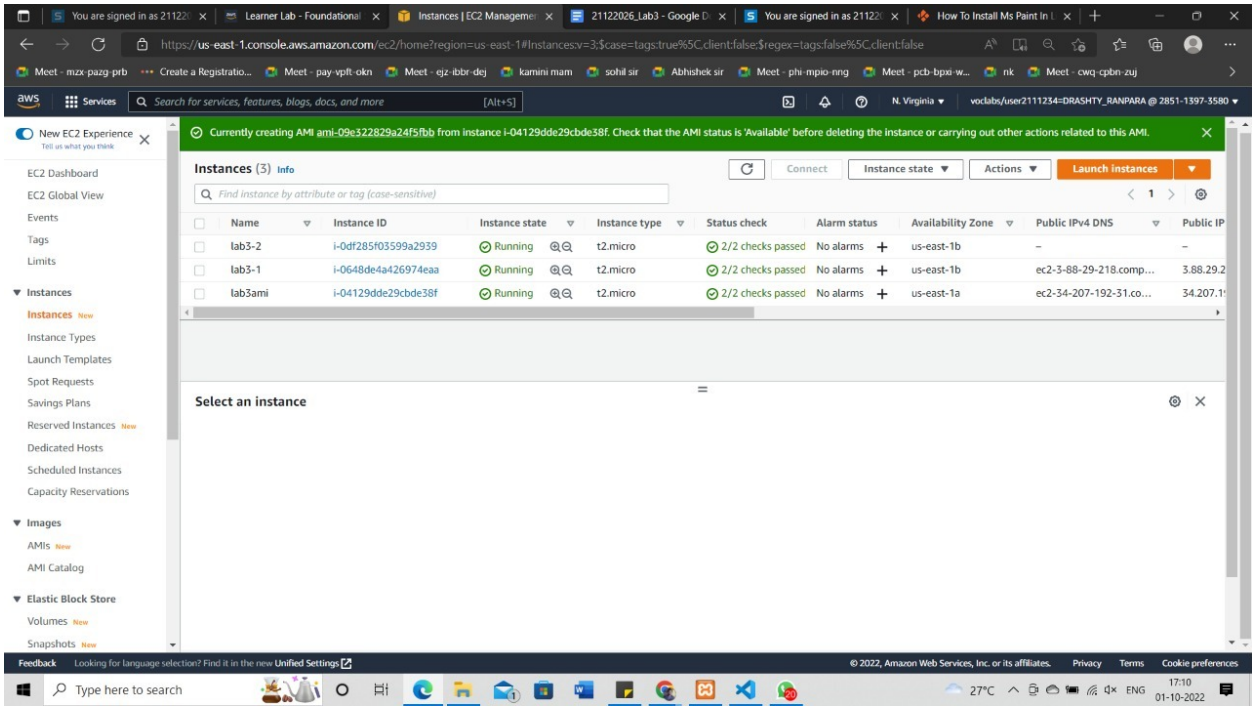
- Write the image name and some description of that image file.



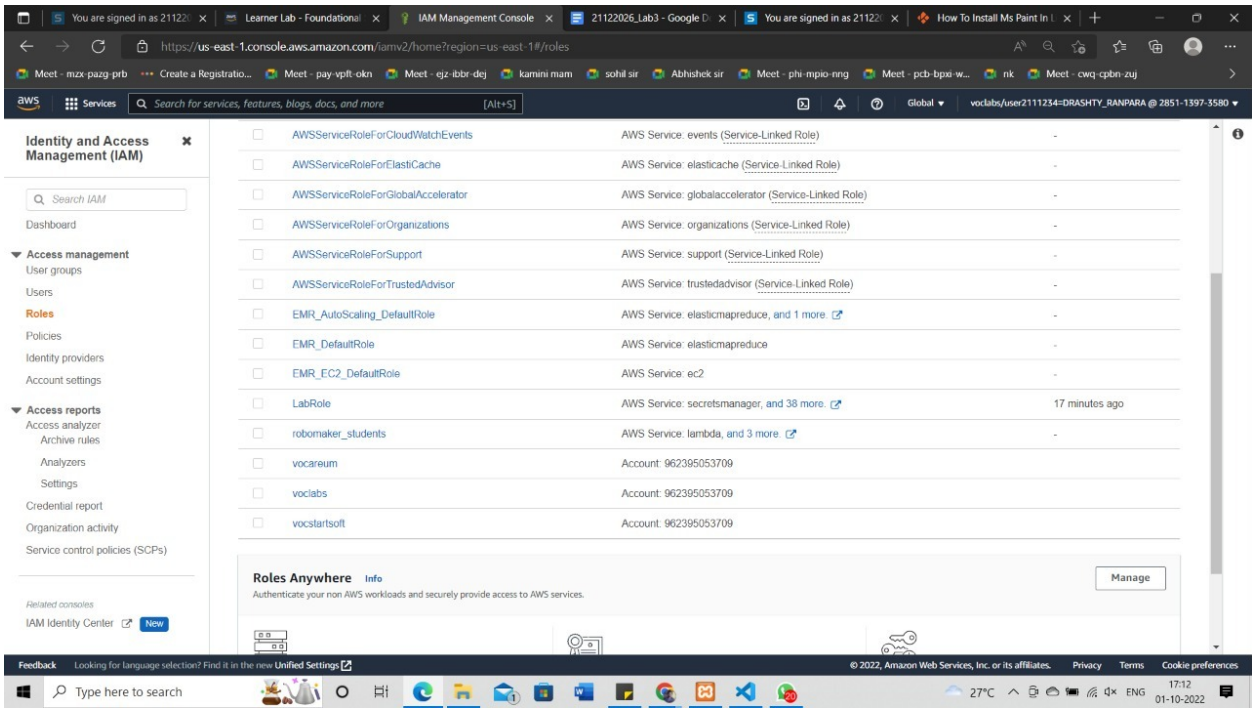
- After changes based on your requirements change the settings and click on the Create Image button.



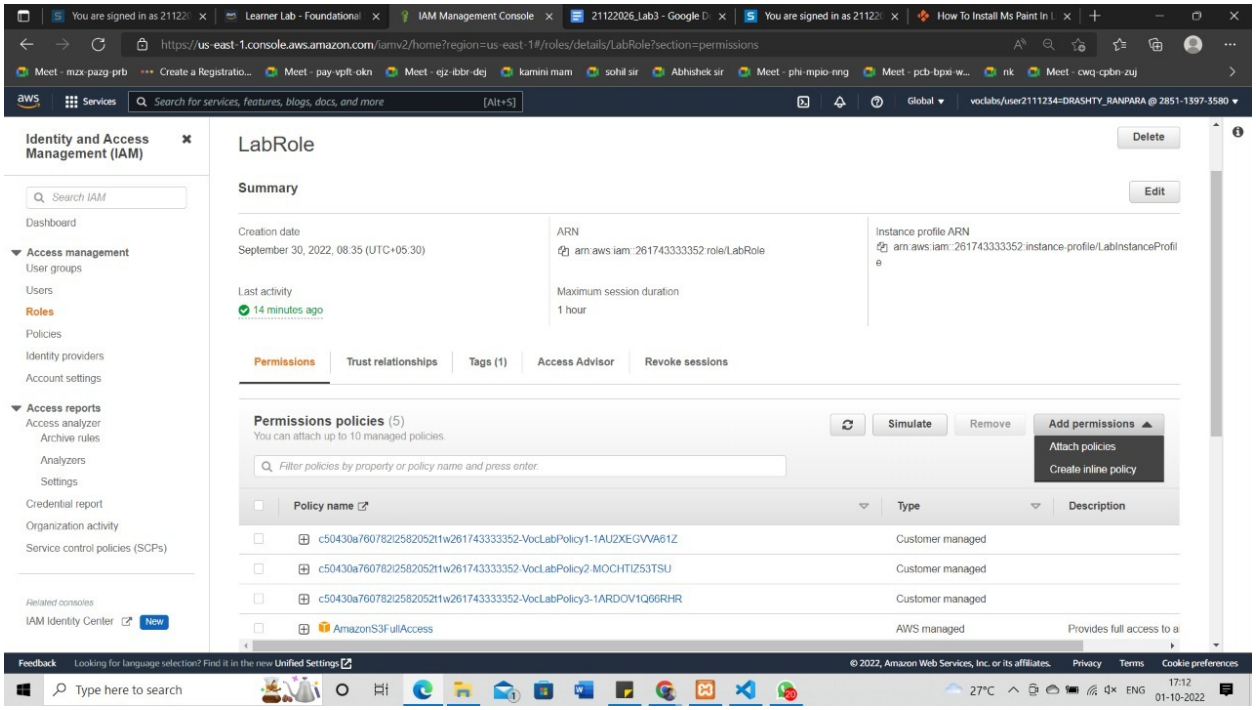
- You will get the message like this.



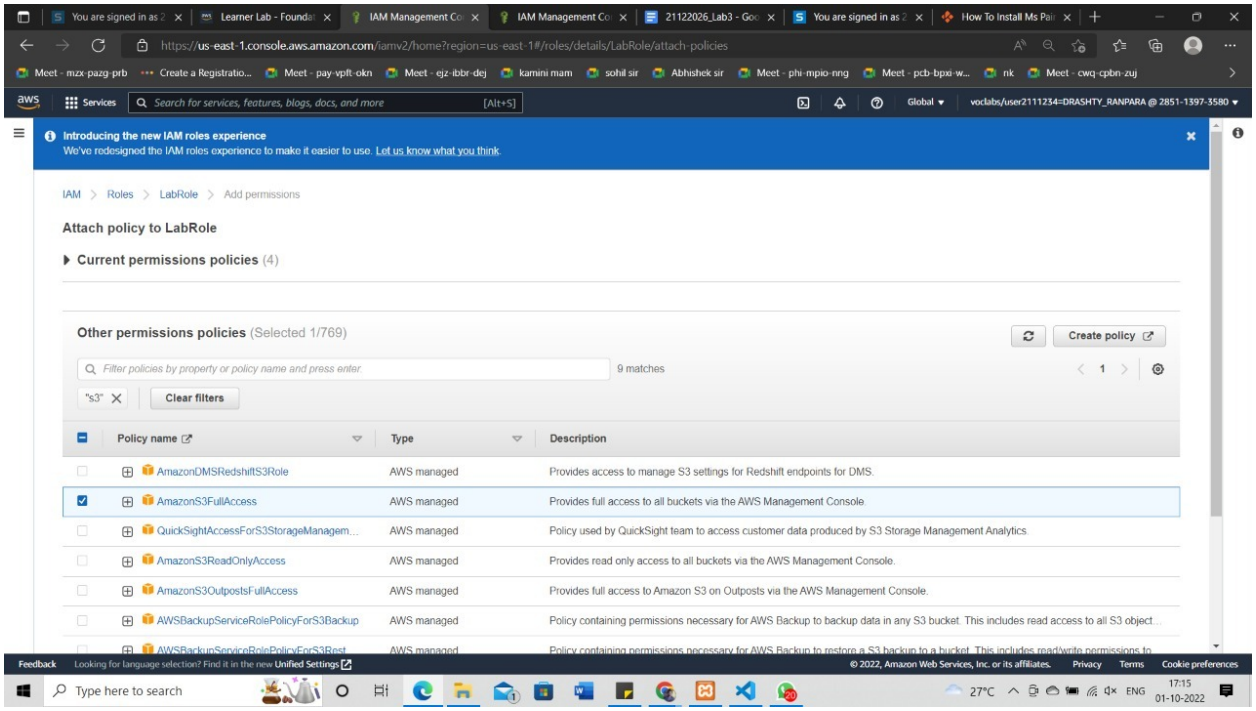
- Now let's First setup the User Role Setting for assigning the IAM Role.
- Navigate to the security, identity, & compliance → IAM → Roles and select the LabRole.



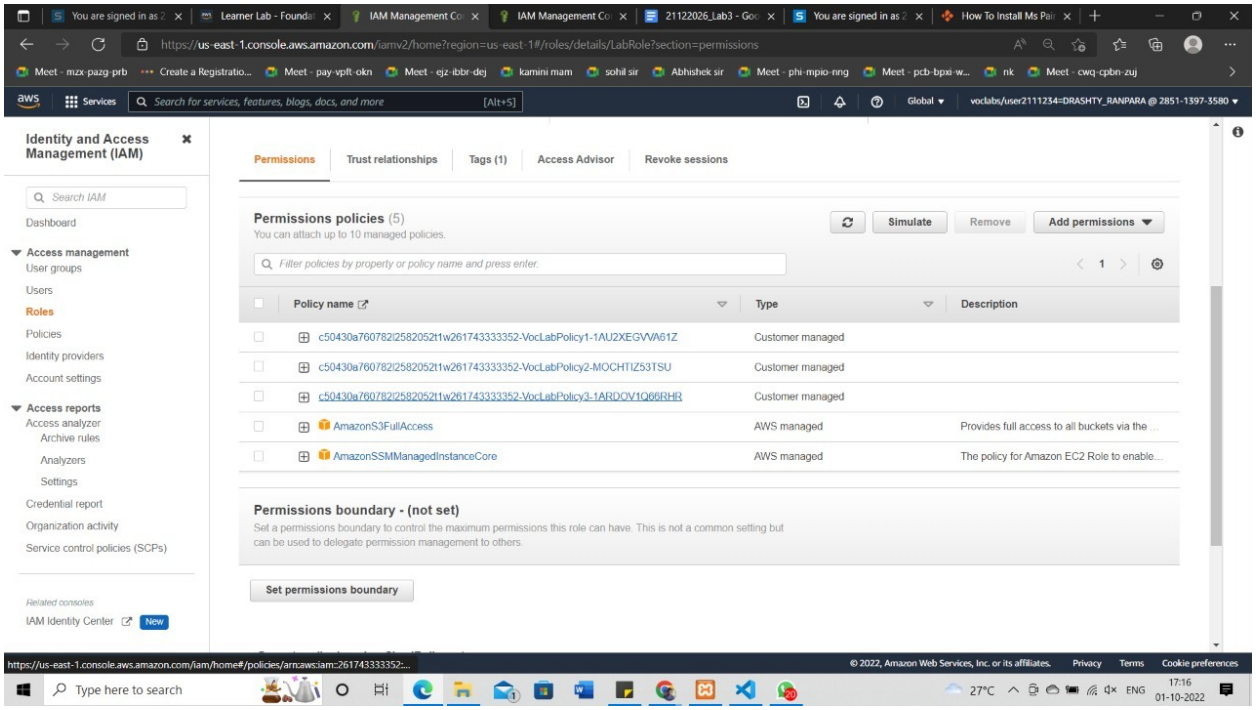
- Click on the Add permissions button and click on the Attach Policies button.



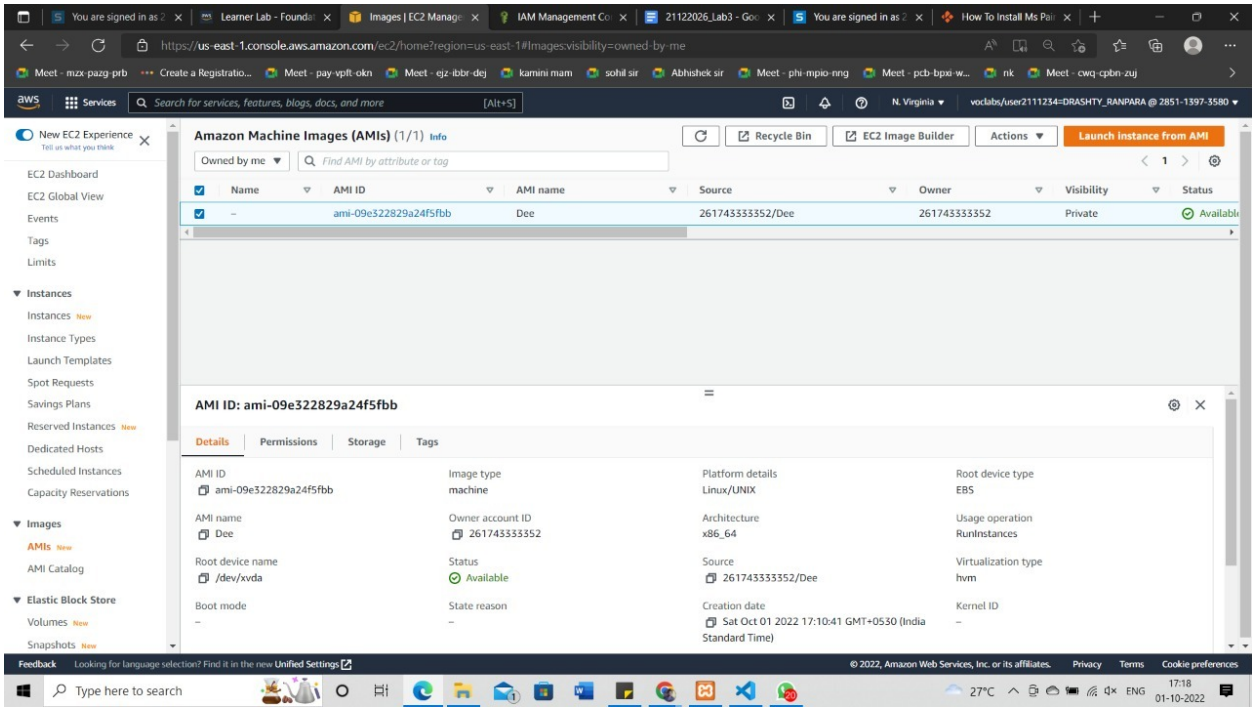
- Write S3 in the find policies search box and select the **AmazonS3FullAccess** policies and click on Attach policies button.



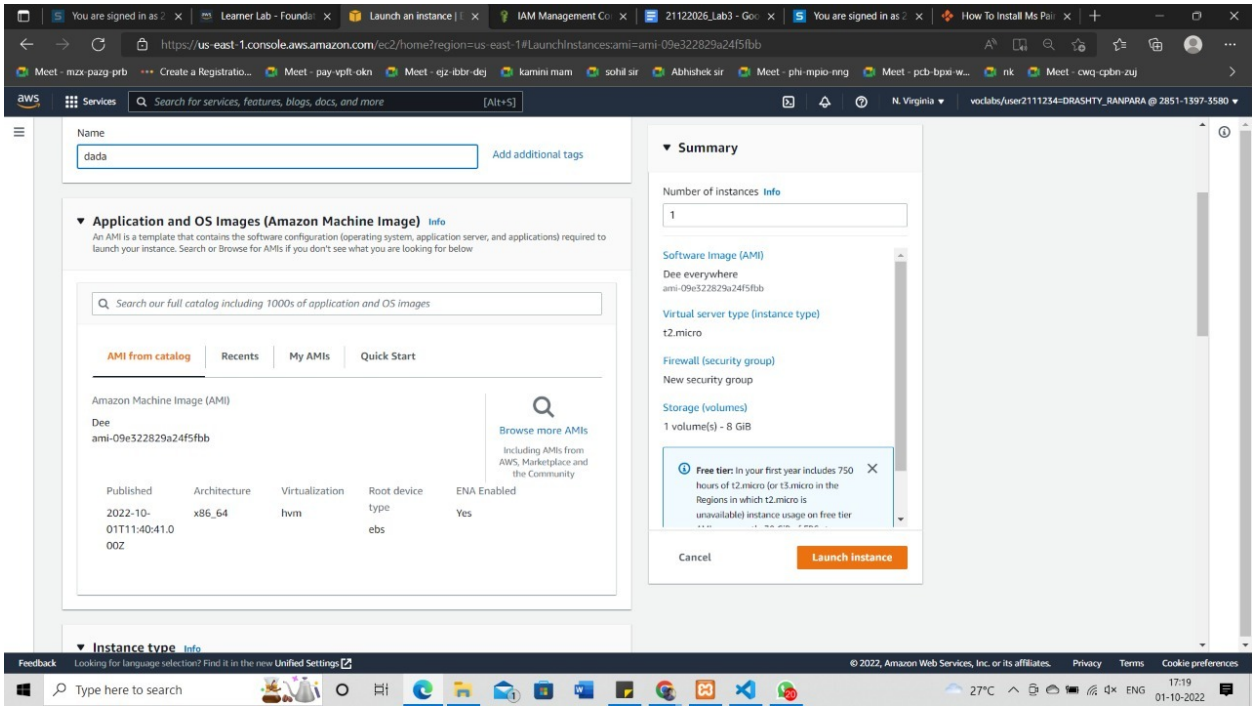
- You can see the policies after attach.



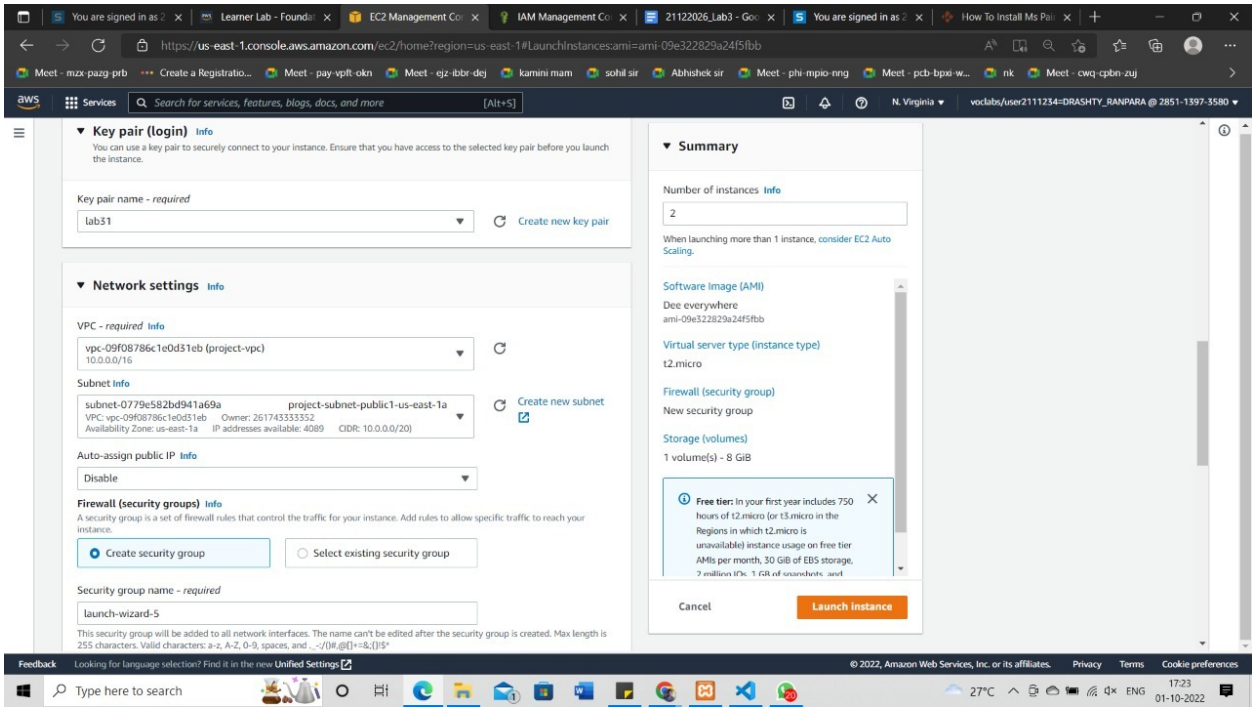
- Now again move to the EC2 Dashboard and go to AMIs.



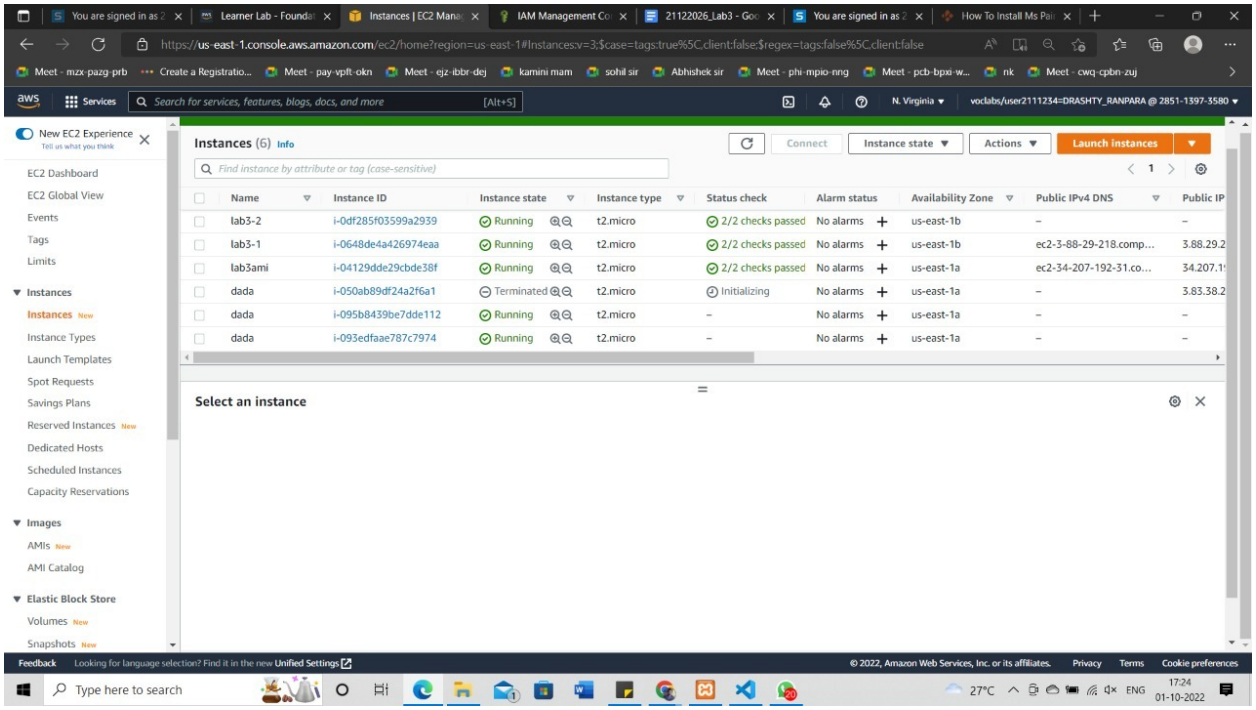
- Click on the checkbox and click on the Launch Instance from AMIs.
- Give the name of the Instance



- Edit in the Network setting and make a network setting same as which we select for the AMIs.
- Change the Number of Instances to 2.



- Click on the Advance detail section In that section IAM instance profile select the LabInstanceProfile and then click on the Launch Instance button.



- Hoorah Task 3 Also Done now.

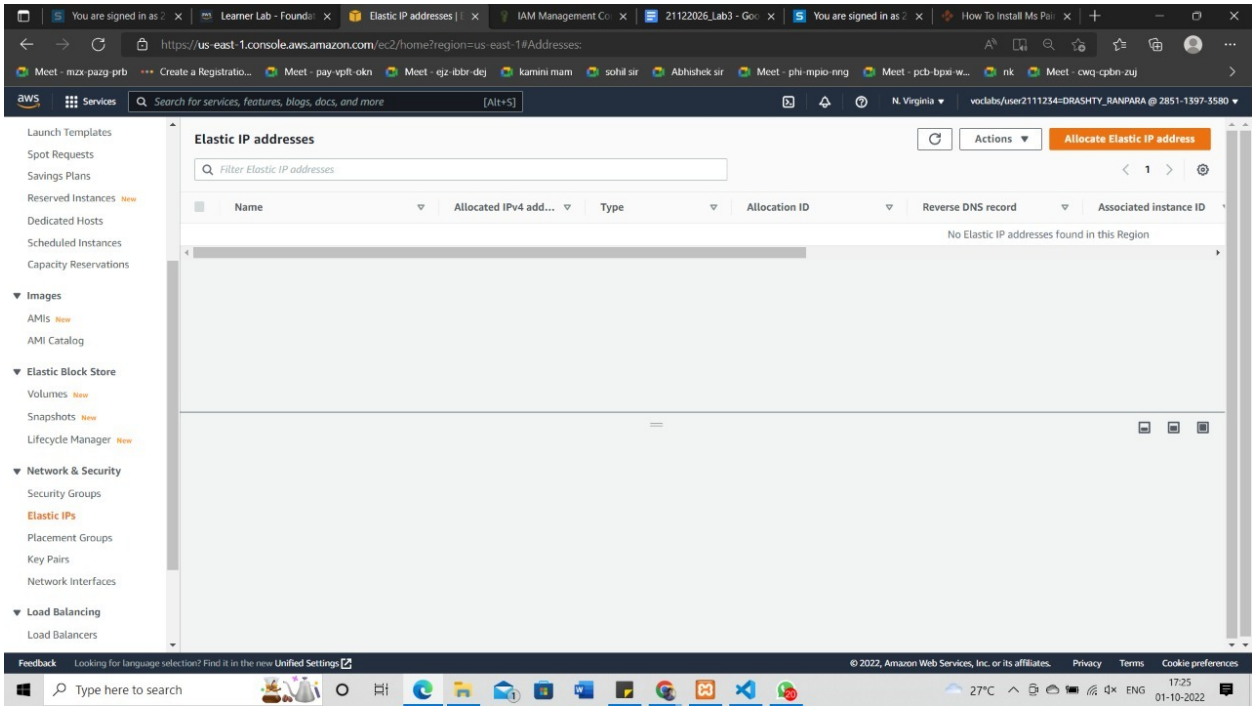
4. Create Elastic IP and associate with any of the instances and demonstrate its usage.

Elastic IP

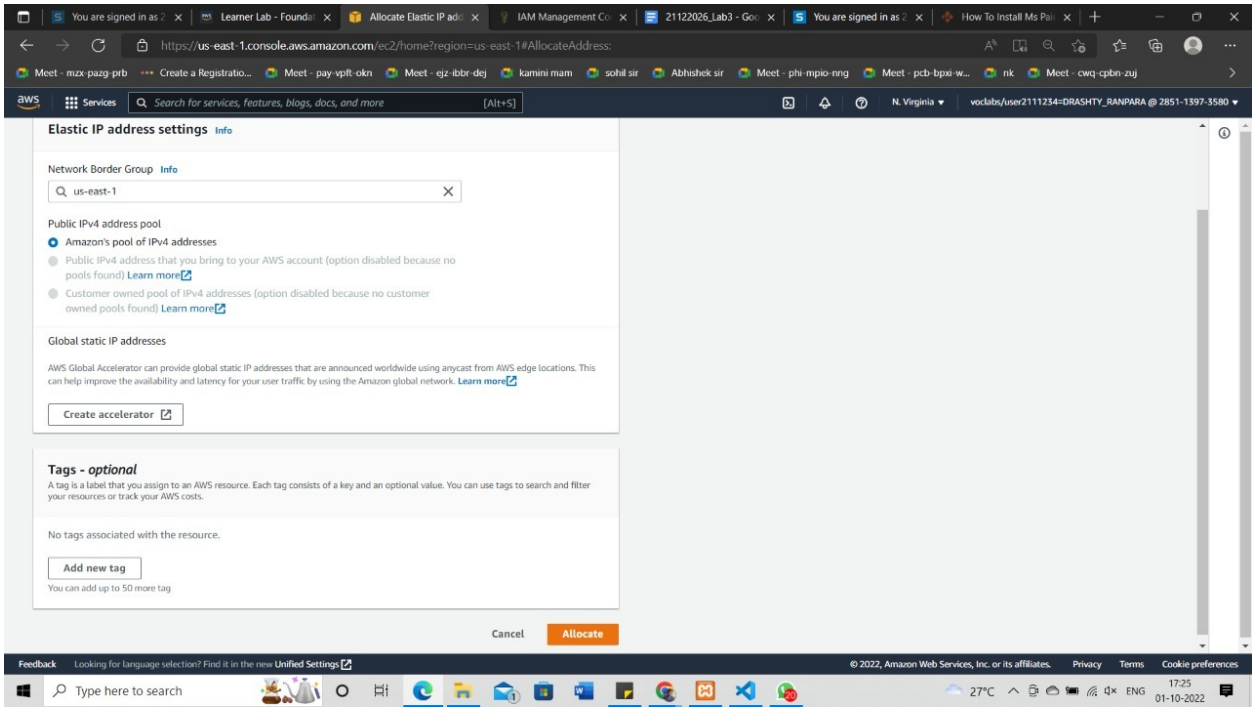
- An Elastic IP (EIP) address is more of a static IPv4 address that is designed for dynamic cloud computing. The main purpose of these IPs is to mask the failure of software or instance from your AWS account.
-
- This is achieved by remapping the address to another instance, which is available in your account as quickly as possible. An IP address is automatically allocated to your AWS account and it will be yours unless you decide to release it. On the other hand, you have the chance to specify the IP in a DNS record to your domain. This will make sure that the specified domain points to your instance.
-
- This address is reachable from the internet like any other public IPv4 address. You can associate this IP address with your instance if your instance does not have any public IPv4 address to enable communication from the internet to the instance. At present, the AWS EIP does not support addresses for IPv6.

Process

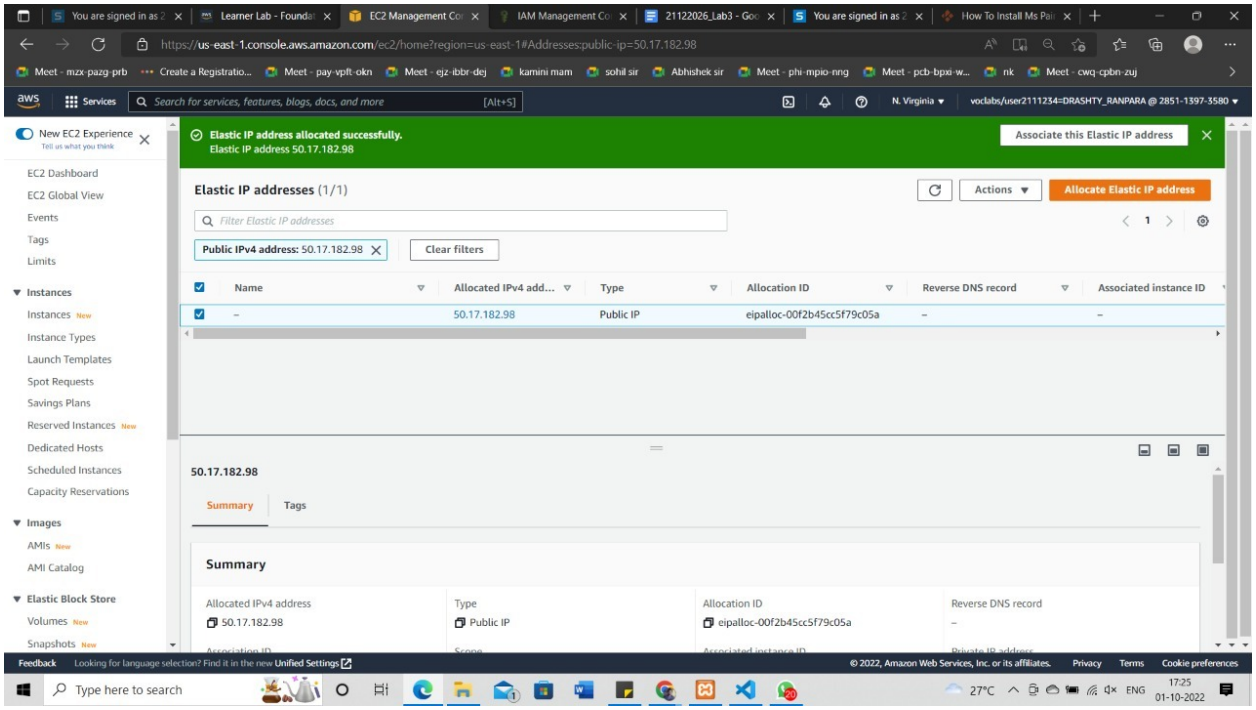
- Navigate to the EC2 Dashboard and click on the Elastic IP Addresses.



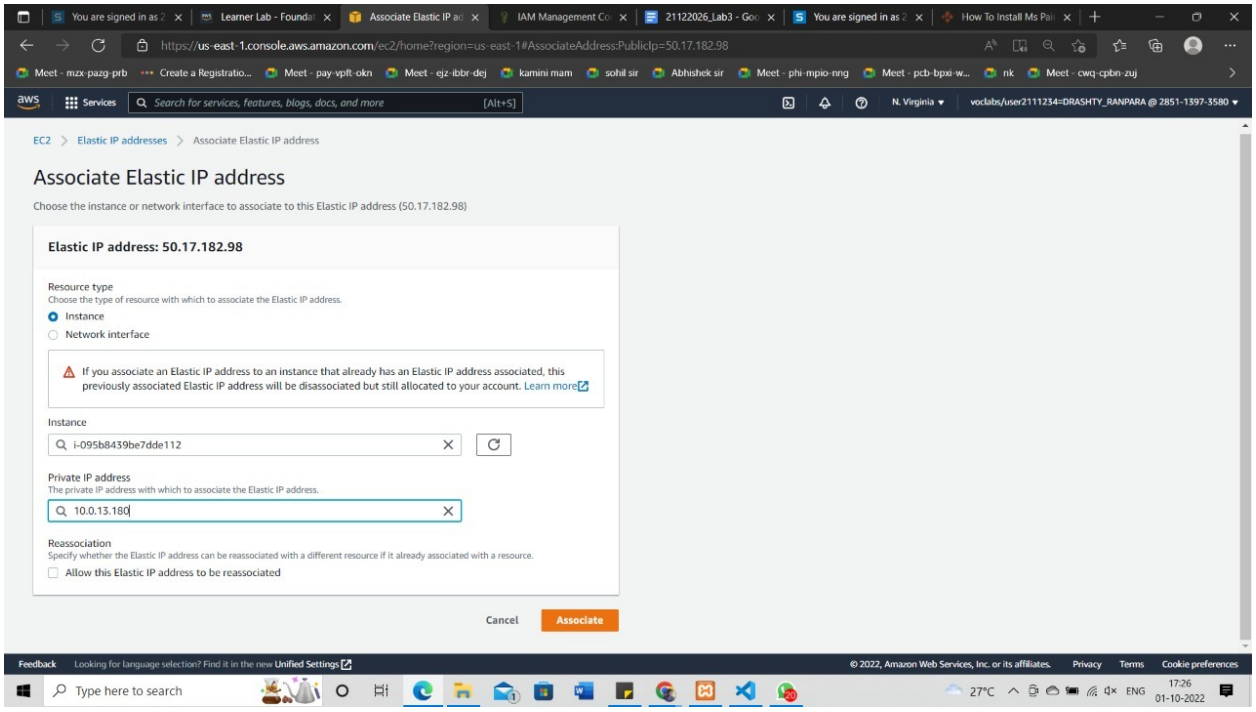
- Click on the Allocate Elastic IP address button
- You will get the new screen
- Select the Network border group and click on the Allocate button.



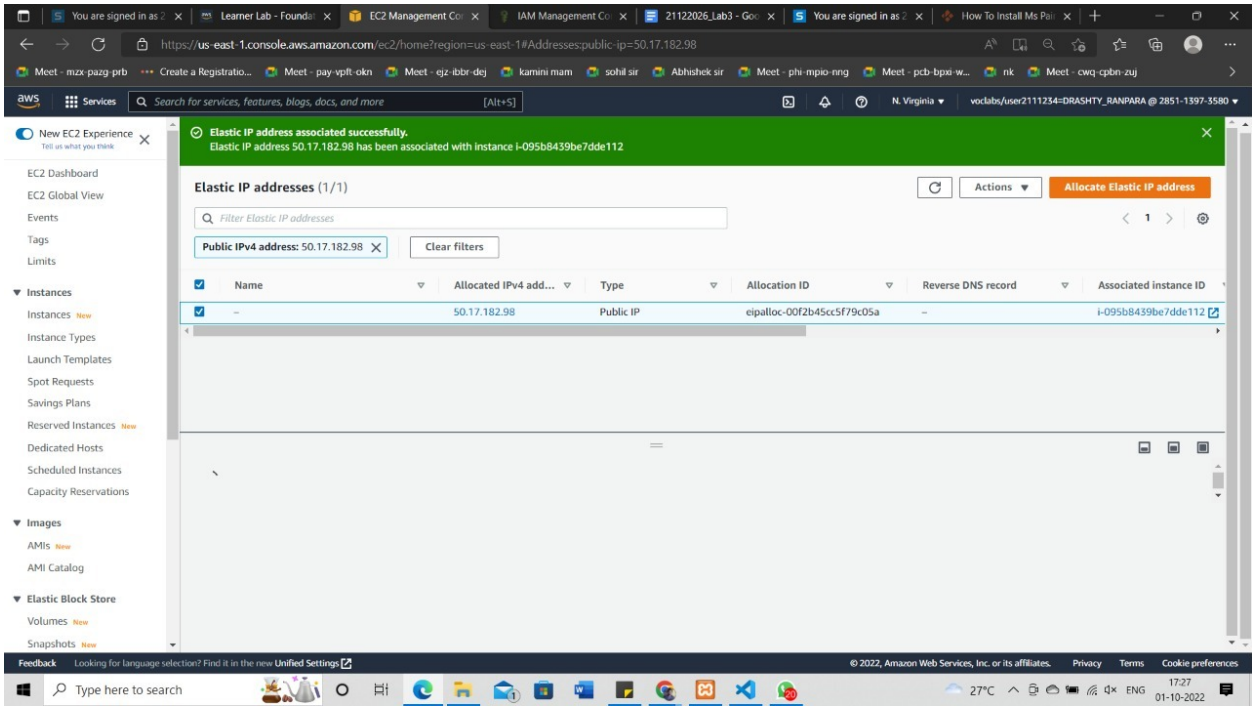
- You get the Elastic IP.



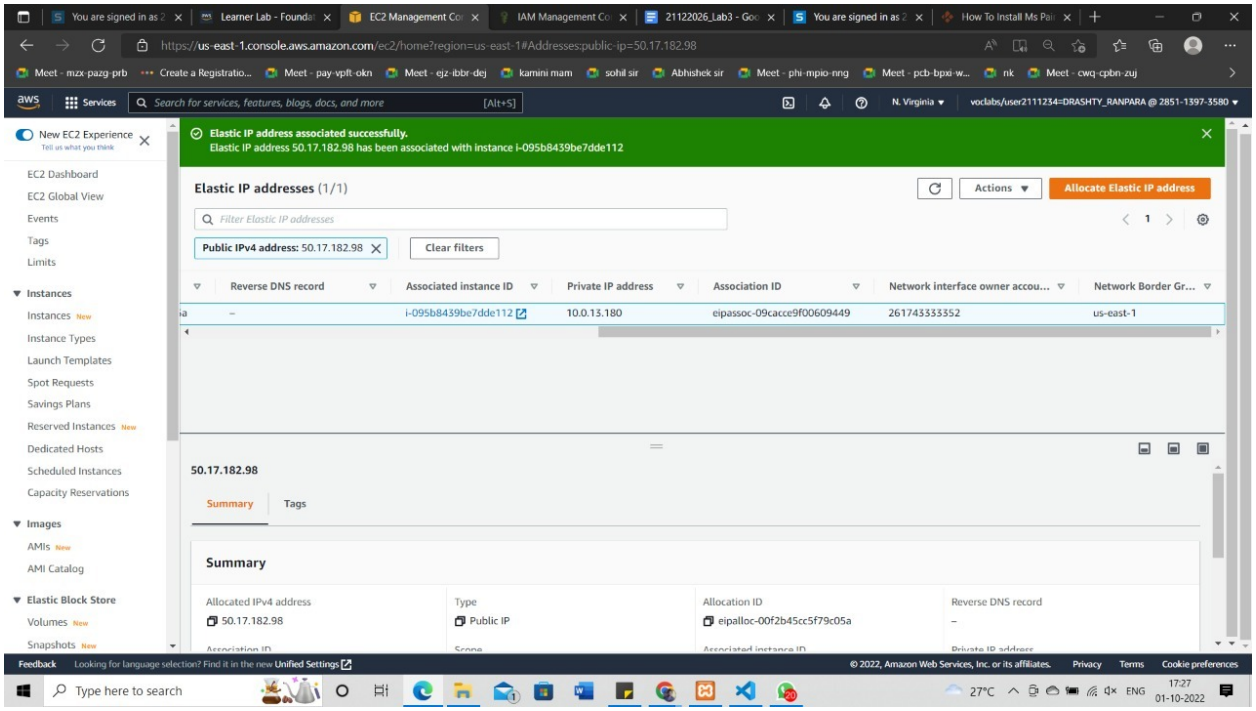
- Check the checkbox of the elastic ip and click on the actions button and click on the Associate Address
- Choose the Instance and select private ip and click on the associate button.



- You will get the message like this.

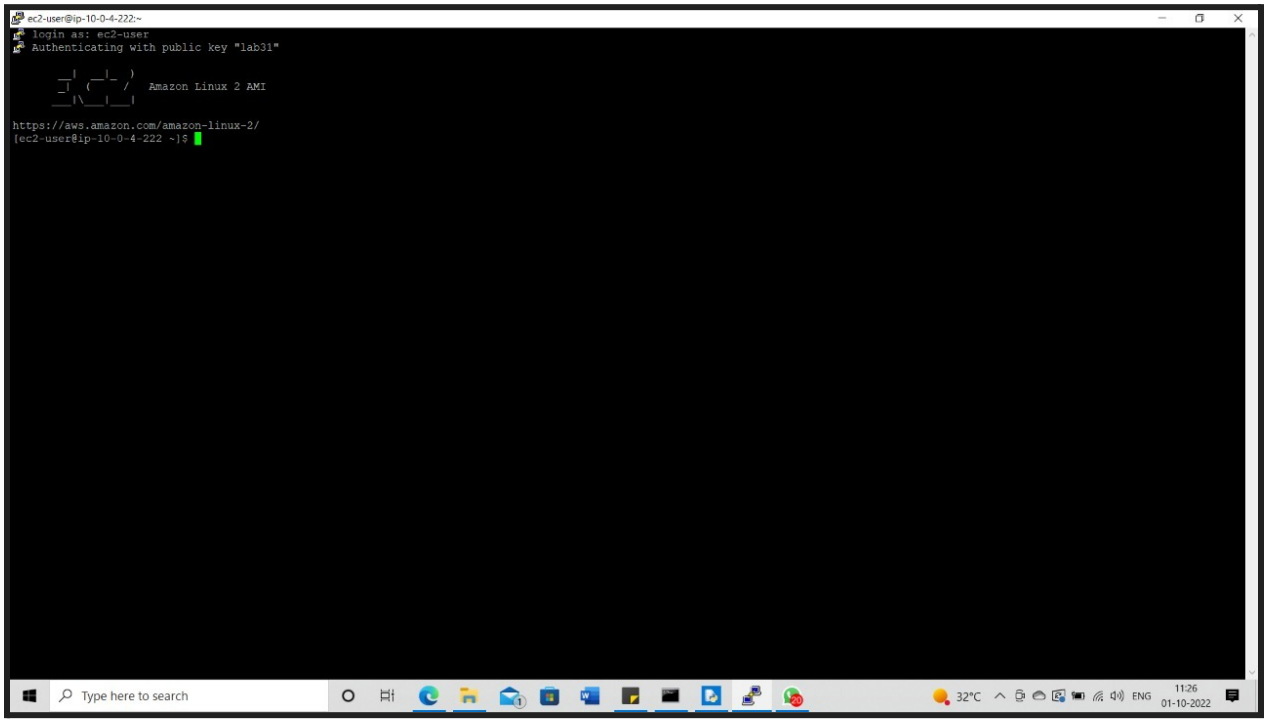


- For confirmation you can see the Associated instance id.

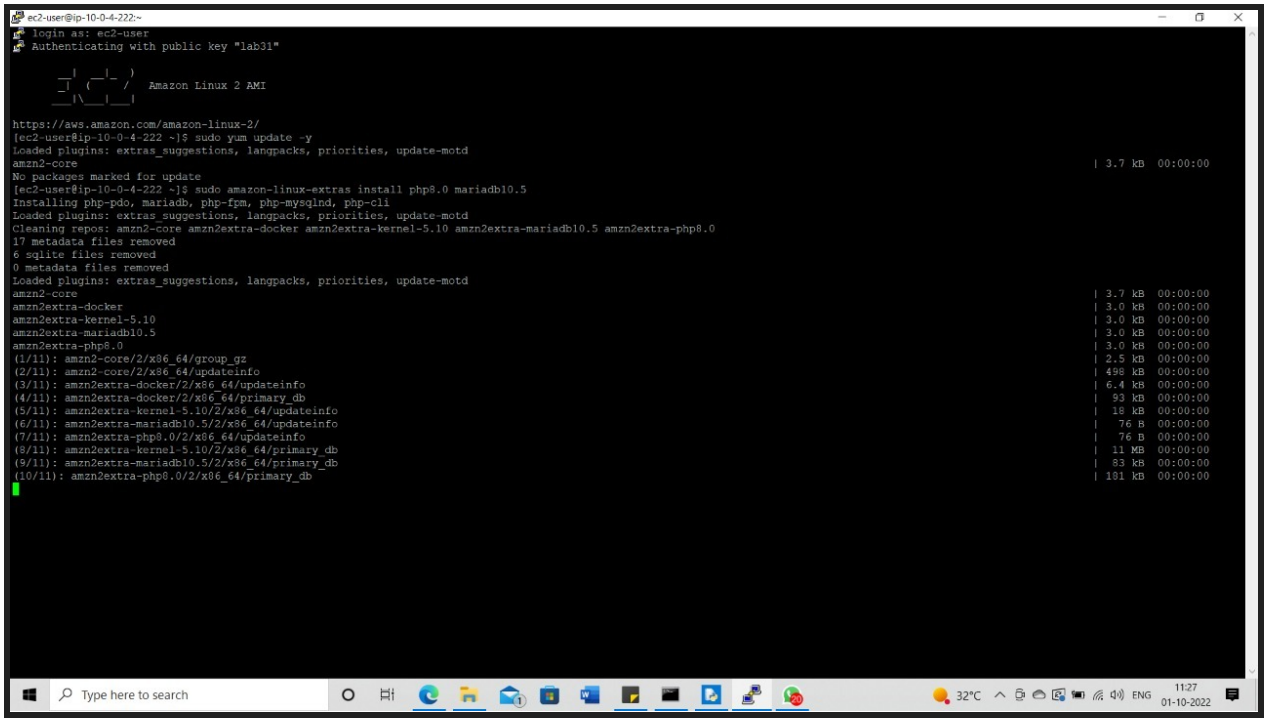


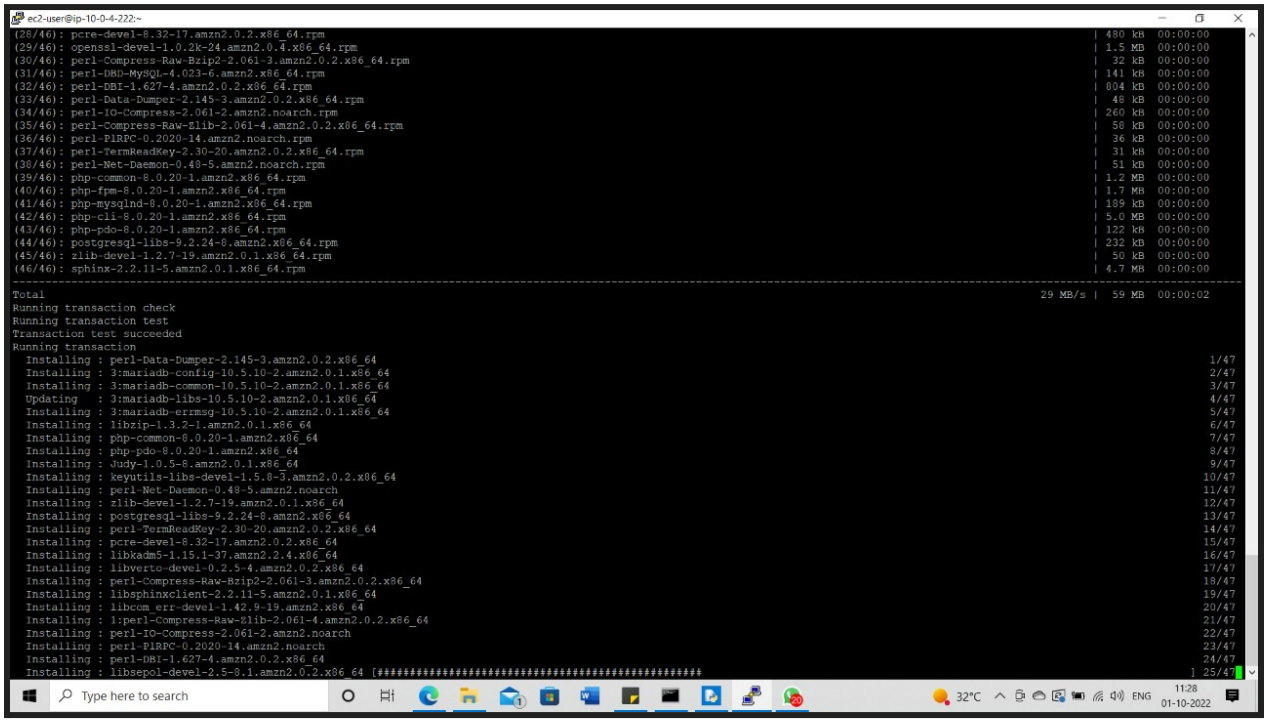
- Hurray!!! Task 4 also completed

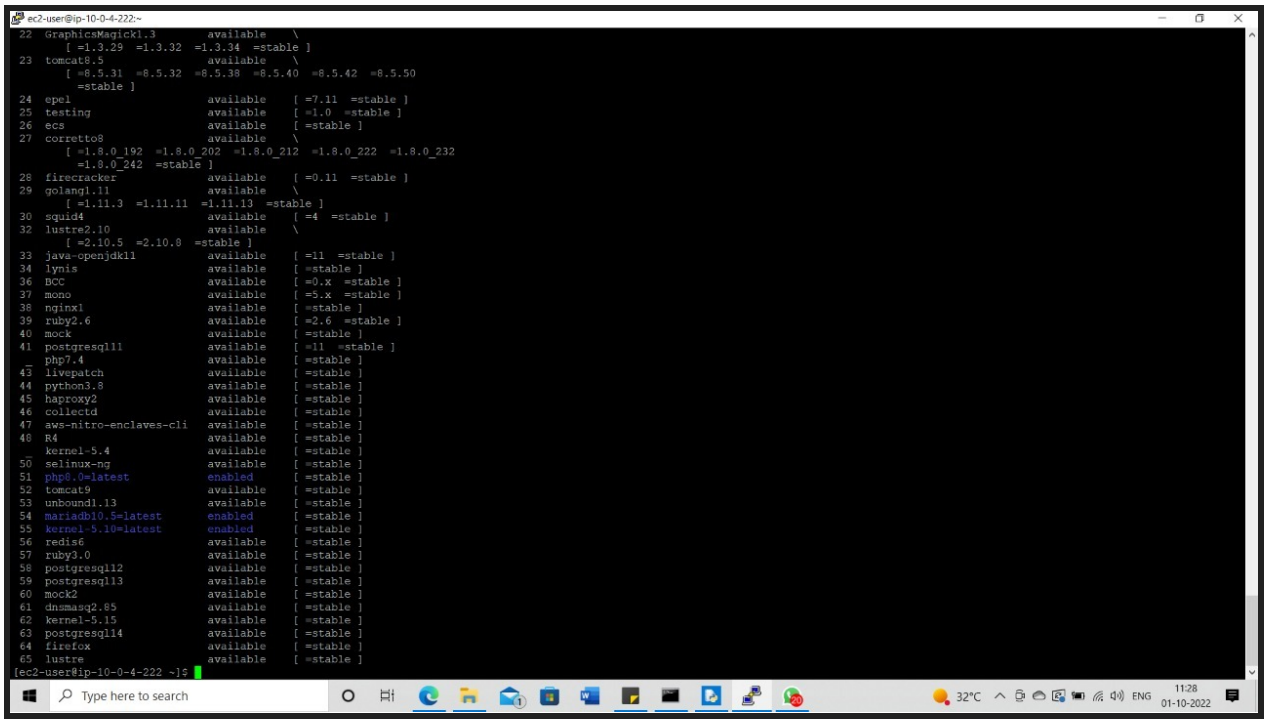
5. Install a web or Application server in any of the instances.
    - Run the Putty and give the host name from the instance and from category goto ssh and click on Auth and select the key pair which is selected during the creating instance.
    - Now enter the username for the login

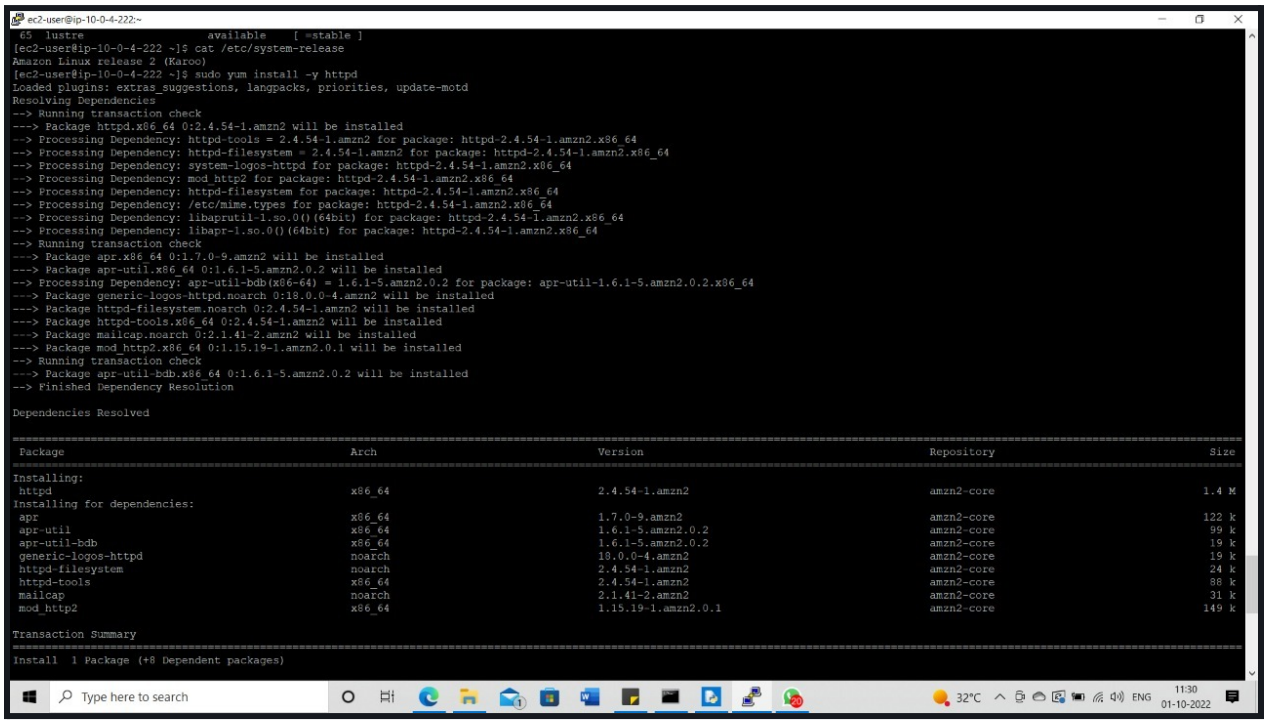


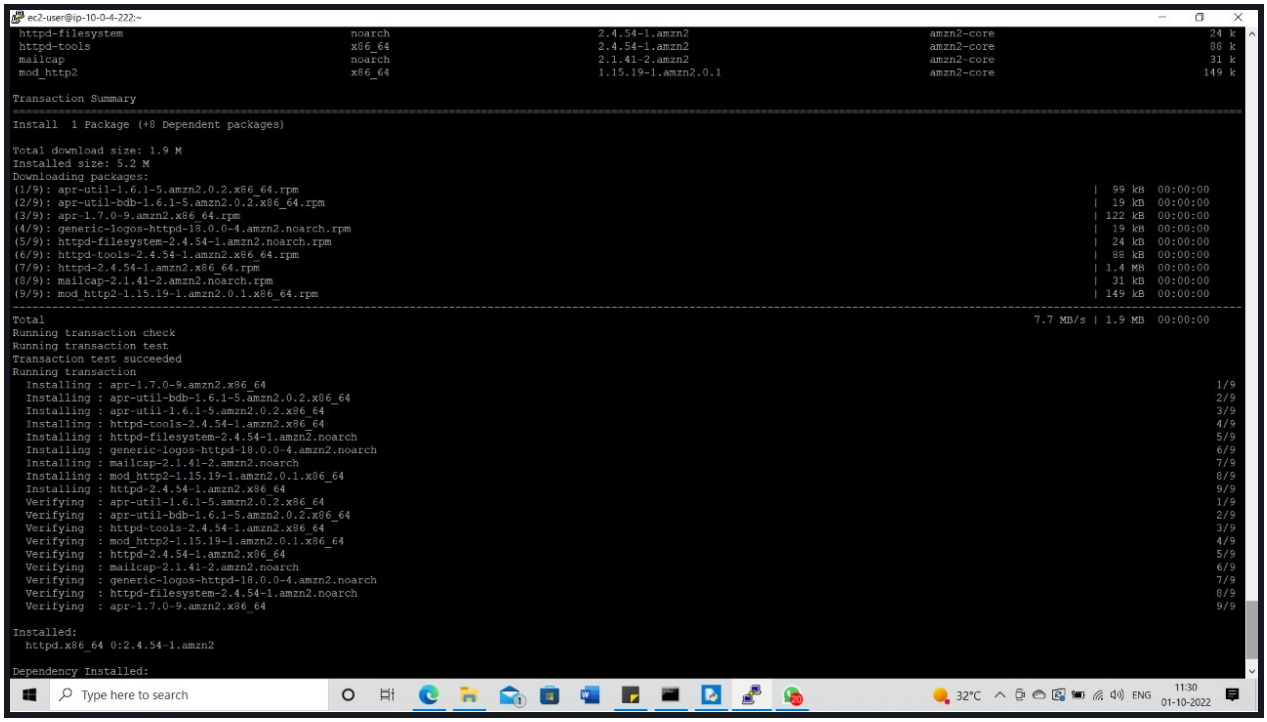
- We are Installing an **Apache web server with PHP and MariaDB.**
- All the commands are in the screenshot please goto through it.


ec2-user@ip-10-0-4-222:~
login as: ec2-user
Authenticating with public key "lab31"
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-222 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
No packages marked for update
[ec2-user@ip-10-0-4-222 ~]$ sudo amazon-linux-extras install php8.0 mariadb10.5
Installing php-pdo, mariadb, php-fpm, php-mysqlnd, php-cli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-mariadb10.5 amzn2extra-php8.0
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
amzn2extra-docker | 3.0 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
amzn2extra-mariadb10.5 | 3.0 kB 00:00:00
amzn2extra-php8.0 | 3.0 kB 00:00:00
(1/11): amzn2-core/2/x86_64/group_gz | 2.5 kB 00:00:00
(2/11): amzn2-core/2/x86_64/updateinfo | 498 kB 00:00:00
(3/11): amzn2extra-docker/2/x86_64/updateinfo | 6.4 kB 00:00:00
(4/11): amzn2extra-docker/2/x86_64/primary_db | 93 kB 00:00:00
(5/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 18 kB 00:00:00
(6/11): amzn2extra-mariadb10.5/2/x86_64/updateinfo | 76 B 00:00:00
(7/11): amzn2extra-php8.0/2/x86_64/updateinfo | 76 B 00:00:00
(8/11): amzn2extra-kernel-5.10/2/x86_64/primary_db | 11 MB 00:00:00
(9/11): amzn2extra-mariadb10.5/2/x86_64/primary_db | 83 kB 00:00:00
(10/11): amzn2extra-php8.0/2/x86_64/primary_db | 181 kB 00:00:00
Type here to search
32°C
ENG
11:27
01-10-2022

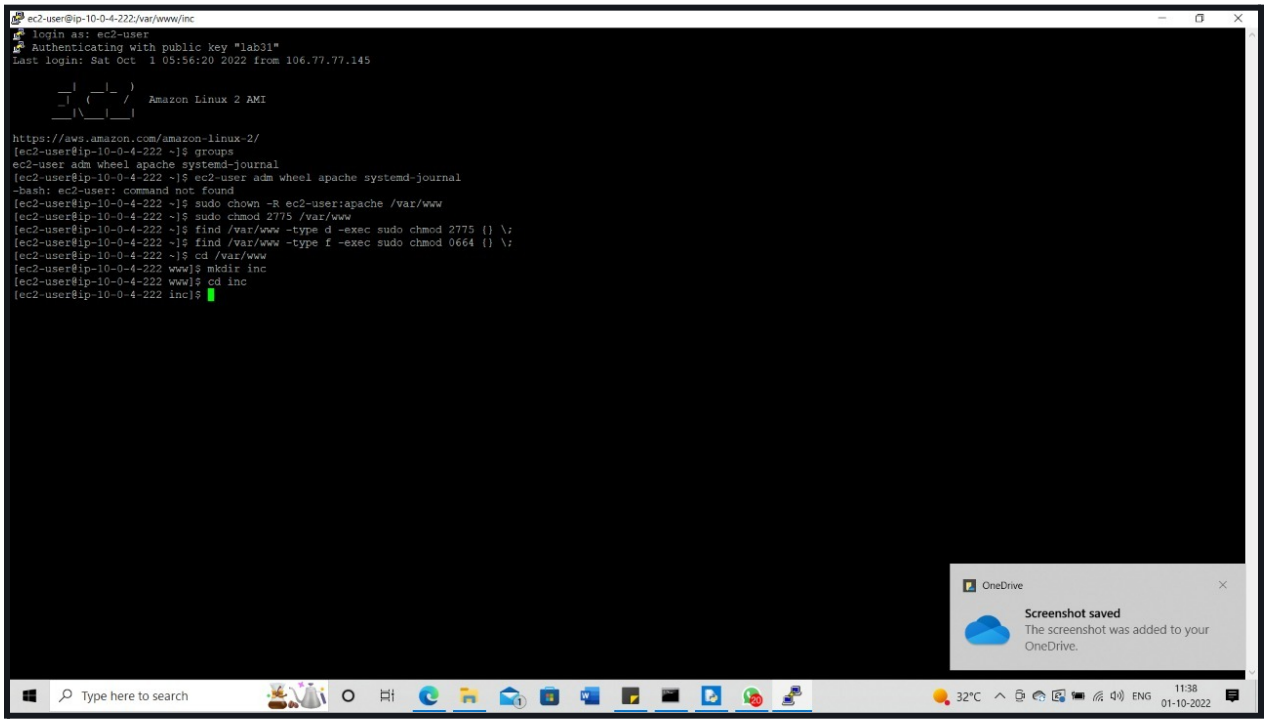

ec2-user@ip-10-0-4-222:~
(28/46): pcre-devel-8.32-17.amzn2.0.2.x86_64.rpm | 480 kB 00:00:00
(29/46): openssl-devel-1.0.2k-24.amzn2.0.4.x86_64.rpm | 1.5 MB 00:00:00
(30/46): perl-Compress-Raw-Bzip2-2.061-3.amzn2.0.2.x86_64.rpm | 32 kB 00:00:00
(31/46): perl-DBD-MySQL-4.023-6.amzn2.x86_64.rpm | 141 kB 00:00:00
(32/46): perl-DBI-1.627-4.amzn2.0.2.x86_64.rpm | 804 kB 00:00:00
(33/46): perl-Data-Dumper-2.145-3.amzn2.0.2.x86_64.rpm | 48 kB 00:00:00
(34/46): perl-IO-Compress-2.061-2.amzn2.noarch.rpm | 260 kB 00:00:00
(35/46): perl-Compress-Raw-Zlib-2.061-4.amzn2.0.2.x86_64.rpm | 58 kB 00:00:00
(36/46): perl-PlRPC-0.2020-14.amzn2.noarch.rpm | 36 kB 00:00:00
(37/46): perl-TermReadKey-2.30-20.amzn2.0.2.x86_64.rpm | 31 kB 00:00:00
(38/46): perl-Net-Daemon-0.48-5.amzn2.noarch.rpm | 51 kB 00:00:00
(39/46): php-common-8.0.20-1.amzn2.x86_64.rpm | 1.2 MB 00:00:00
(40/46): php-fpm-8.0.20-1.amzn2.x86_64.rpm | 1.7 MB 00:00:00
(41/46): php-mysqlnd-8.0.20-1.amzn2.x86_64.rpm | 189 kB 00:00:00
(42/46): php-cli-8.0.20-1.amzn2.x86_64.rpm | 5.0 MB 00:00:00
(43/46): php-pdo-8.0.20-1.amzn2.x86_64.rpm | 122 kB 00:00:00
(44/46): postgresql-libs-9.2.24-8.amzn2.x86_64.rpm | 232 kB 00:00:00
(45/46): zlib-devel-1.2.7-19.amzn2.0.1.x86_64.rpm | 50 kB 00:00:00
(46/46): sphinx-2.2.11-5.amzn2.0.1.x86_64.rpm | 4.7 MB 00:00:00
Total 29 MB/s | 59 MB 00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : perl-Data-Dumper-2.145-3.amzn2.0.2.x86_64 1/47
Installing : 3:mariadb-config-10.5.10-2.amzn2.0.1.x86_64 2/47
Installing : 3:mariadb-common-10.5.10-2.amzn2.0.1.x86_64 3/47
Updating : 3:mariadb-libs-10.5.10-2.amzn2.0.1.x86_64 4/47
Installing : 3:mariadb-errmsg-10.5.10-2.amzn2.0.1.x86_64 5/47
Installing : libzip-1.3.2-1.amzn2.0.1.x86_64 6/47
Installing : php-common-8.0.20-1.amzn2.x86_64 7/47
Installing : php-pdo-8.0.20-1.amzn2.x86_64 8/47
Installing : Judy-1.0.5-8.amzn2.0.1.x86_64 9/47
Installing : keyutils-libs-devel-1.5.8-3.amzn2.0.2.x86_64 10/47
Installing : perl-Net-Daemon-0.48-5.amzn2.noarch 11/47
Installing : zlib-devel-1.2.7-19.amzn2.0.1.x86_64 12/47
Installing : postgresql-libs-9.2.24-8.amzn2.x86_64 13/47
Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64 14/47
Installing : pcre-devel-8.32-17.amzn2.0.2.x86_64 15/47
Installing : libkadm5-1.15.1-37.amzn2.2.4.x86_64 16/47
Installing : libverto-devel-0.2.5-4.amzn2.0.2.x86_64 17/47
Installing : perl-Compress-Raw-Bzip2-2.061-3.amzn2.0.2.x86_64 18/47
Installing : libsphinxclient-2.2.11-5.amzn2.0.1.x86_64 19/47
Installing : libcom_err-devel-1.42.9-19.amzn2.x86_64 20/47
Installing : 1:perl-Compress-Raw-Zlib-2.061-4.amzn2.0.2.x86_64 21/47
Installing : perl-IO-Compress-2.061-2.amzn2.noarch 22/47
Installing : perl-PlRPC-0.2020-14.amzn2.noarch 23/47
Installing : perl-DBI-1.627-4.amzn2.0.2.x86_64 24/47
Installing : libsepol-devel-2.5-8.1.amzn2.0.2.x86_64 [##] 25/47
Type here to search
32°C
ENG
11:28
01-10-2022

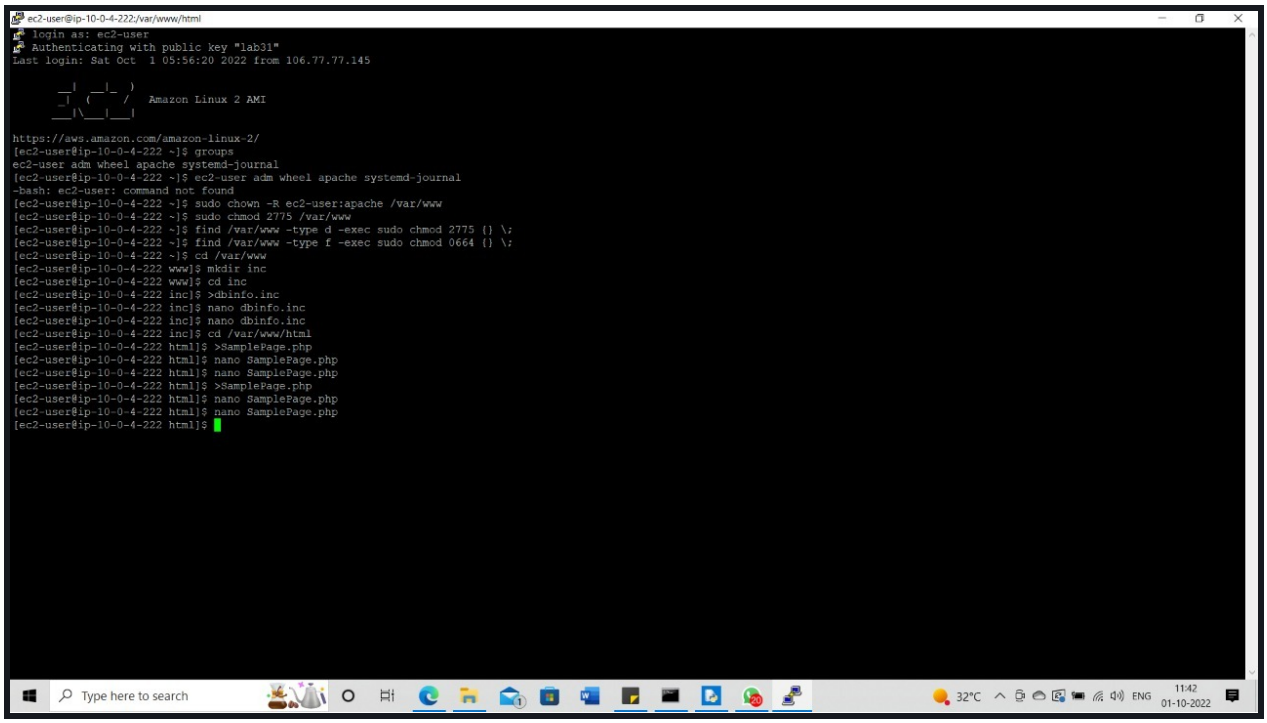

ec2-user@ip-10-0-4-222:~
22 GraphicsMagick1.3 available \
[=1.3.29 =1.3.32 =1.3.34 =stable]
23 tomcat8.5 available \
[=8.5.31 =8.5.32 =8.5.38 =8.5.40 =8.5.42 =8.5.50
=stable]
24 epel available [=7.11 =stable]
25 testing available [=1.0 =stable]
26 ecs available [=stable]
27 corretto8 available \
[=1.8.0_192 =1.8.0_202 =1.8.0_212 =1.8.0_222 =1.8.0_232
=1.8.0_242 =stable]
28 firecracker available [=0.11 =stable]
29 golang1.11 available \
[=1.11.3 =1.11.11 =1.11.13 =stable]
30 squid4 available [=4 =stable]
32 lustre2.10 available \
[=2.10.5 =2.10.8 =stable]
33 java-openjdk11 available [=11 =stable]
34 lynis available [=stable]
36 BCC available [=0.x =stable]
37 mono available [=5.x =stable]
38 nginx1 available [=stable]
39 ruby2.6 available [=2.6 =stable]
40 mock available [=stable]
41 postgresql11 available [=11 =stable]
php7.4 available [=stable]
43 livepatch available [=stable]
44 python3.8 available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
51 php8.0=latest enabled [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5=latest enabled [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis6 available [=stable]
57 ruby3.0 available [=stable]
58 postgresql12 available [=stable]
59 postgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 postgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
[ec2-user@ip-10-0-4-222 ~]$
Type here to search
32°C
ENG
11:28
01-10-2022

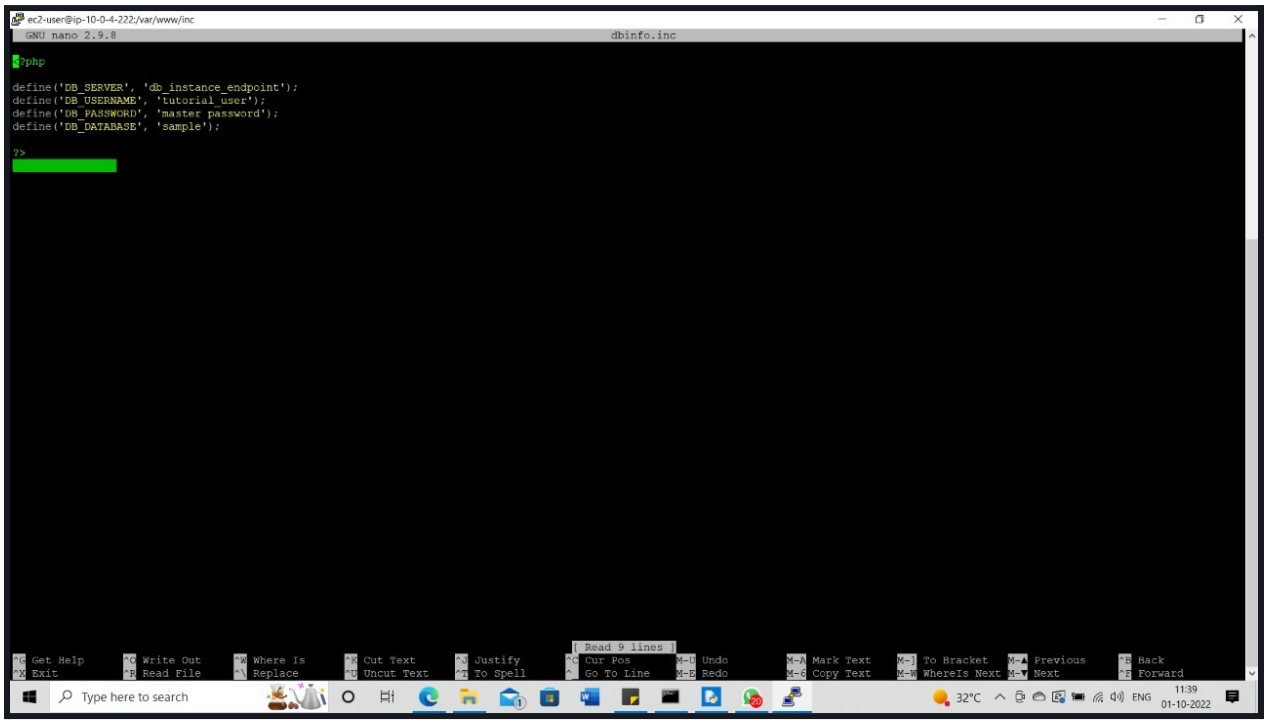

ec2-user@ip-10-0-4-222:~
65 lustre available [=stable]
[ec2-user@ip-10-0-4-222 ~]$ cat /etc/system-release
Amazon Linux release 2 (Karoo)
[ec2-user@ip-10-0-4-222 ~]$ sudo yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
---> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.54-1.amzn2 will be installed
---> Package httpd-tools.x86_64 0:2.4.54-1.amzn2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
httpd x86_64 2.4.54-1.amzn2 amzn2-core 1.4 M
Installing for dependencies:
apr x86_64 1.7.0-9.amzn2 amzn2-core 122 k
apr-util x86_64 1.6.1-5.amzn2.0.2 amzn2-core 99 k
apr-util-bdb x86_64 1.6.1-5.amzn2.0.2 amzn2-core 19 k
generic-logos-httpd noarch 18.0.0-4.amzn2 amzn2-core 19 k
httpd-filesystem noarch 2.4.54-1.amzn2 amzn2-core 24 k
httpd-tools x86_64 2.4.54-1.amzn2 amzn2-core 88 k
mailcap noarch 2.1.41-2.amzn2 amzn2-core 31 k
mod_http2 x86_64 1.15.19-1.amzn2.0.1 amzn2-core 149 k
Transaction Summary
Install 1 Package (+8 Dependent packages)
Type here to search
32°C
ENG
11:30
01-10-2022

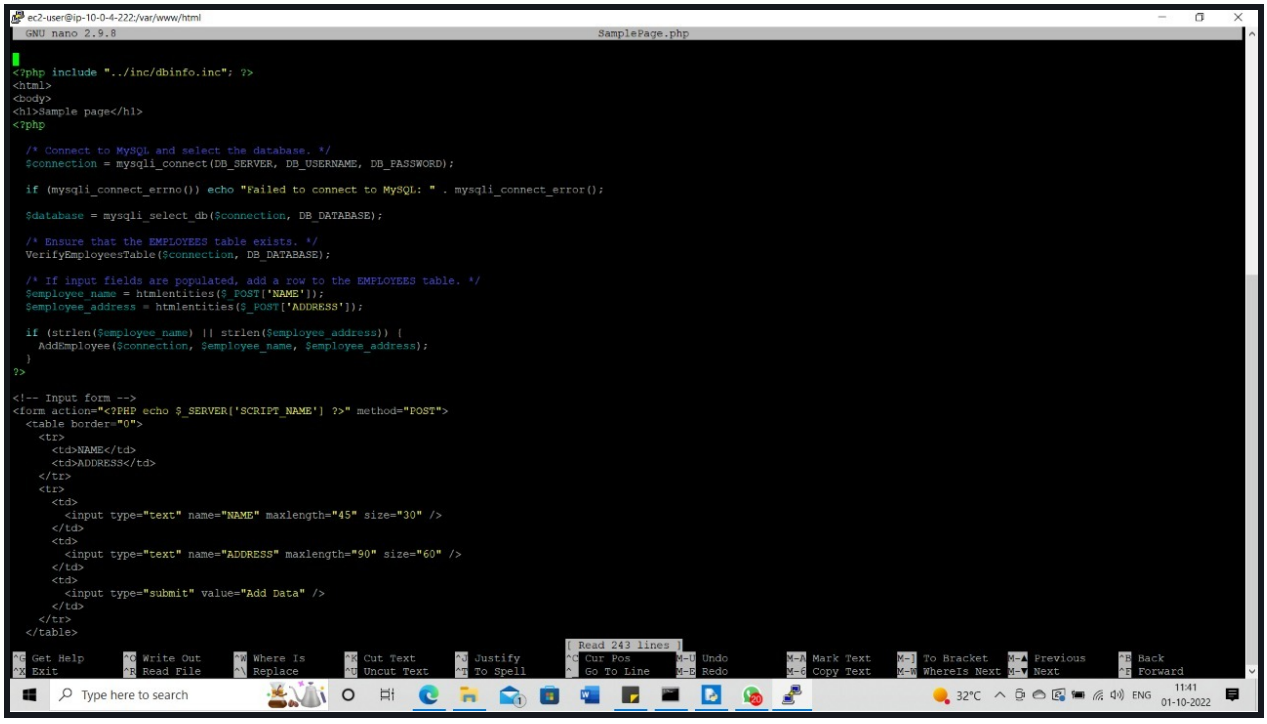


- After this command, make exit command and login again.




ec2-user@ip-10-0-4-222:/var/www/html
login as: ec2-user
Authenticating with public key "lab31"
Last login: Sat Oct 1 05:56:20 2022 from 106.77.77.145
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-222 ~]$ groups
ec2-user adm wheel apache systemd-journal
[ec2-user@ip-10-0-4-222 ~]$ ec2-user adm wheel apache systemd-journal
-bash: ec2-user: command not found
[ec2-user@ip-10-0-4-222 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-10-0-4-222 ~]$ sudo chmod 2775 /var/www
[ec2-user@ip-10-0-4-222 ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-10-0-4-222 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-10-0-4-222 ~]$ cd /var/www
[ec2-user@ip-10-0-4-222 www]$ mkdir inc
[ec2-user@ip-10-0-4-222 www]$ cd inc
[ec2-user@ip-10-0-4-222 inc]$ >dbinfo.inc
[ec2-user@ip-10-0-4-222 inc]$ nano dbinfo.inc
[ec2-user@ip-10-0-4-222 inc]$ nano dbinfo.inc
[ec2-user@ip-10-0-4-222 inc]$ cd /var/www/html
[ec2-user@ip-10-0-4-222 html]$ >SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ >SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$
Type here to search
32°C
ENG
11:42
01-10-2022


ec2-user@ip-10-0-4-222:/var/www/inc
GNU nano 2.9.8
dbinfo.inc
<?php
define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');
?>
[Read 9 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text M-] To Bracket M-▲ Previous ^B Back
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo M-6 Copy Text M-W WhereIs Next M-▼ Next ^F Forward
Type here to search
32°C
ENG
11:39
01-10-2022



- Save and Exit
- Successfully run.
- Hurray All the Task done successfully.