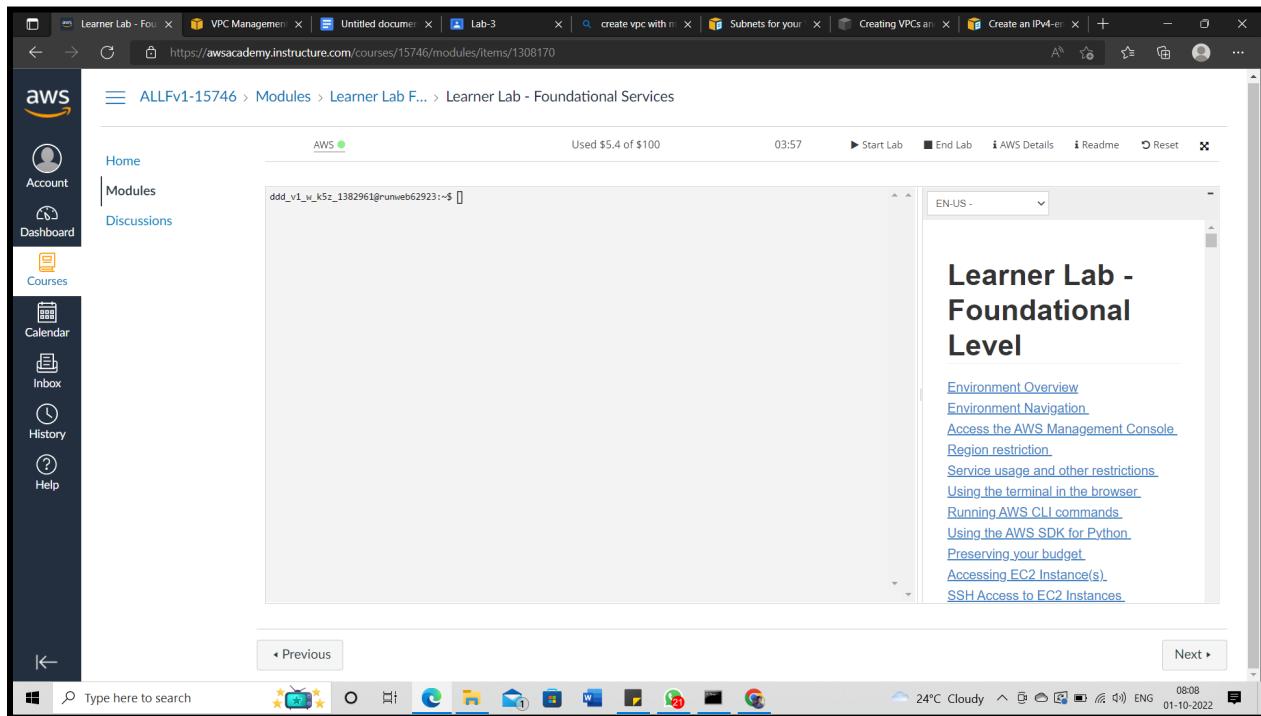


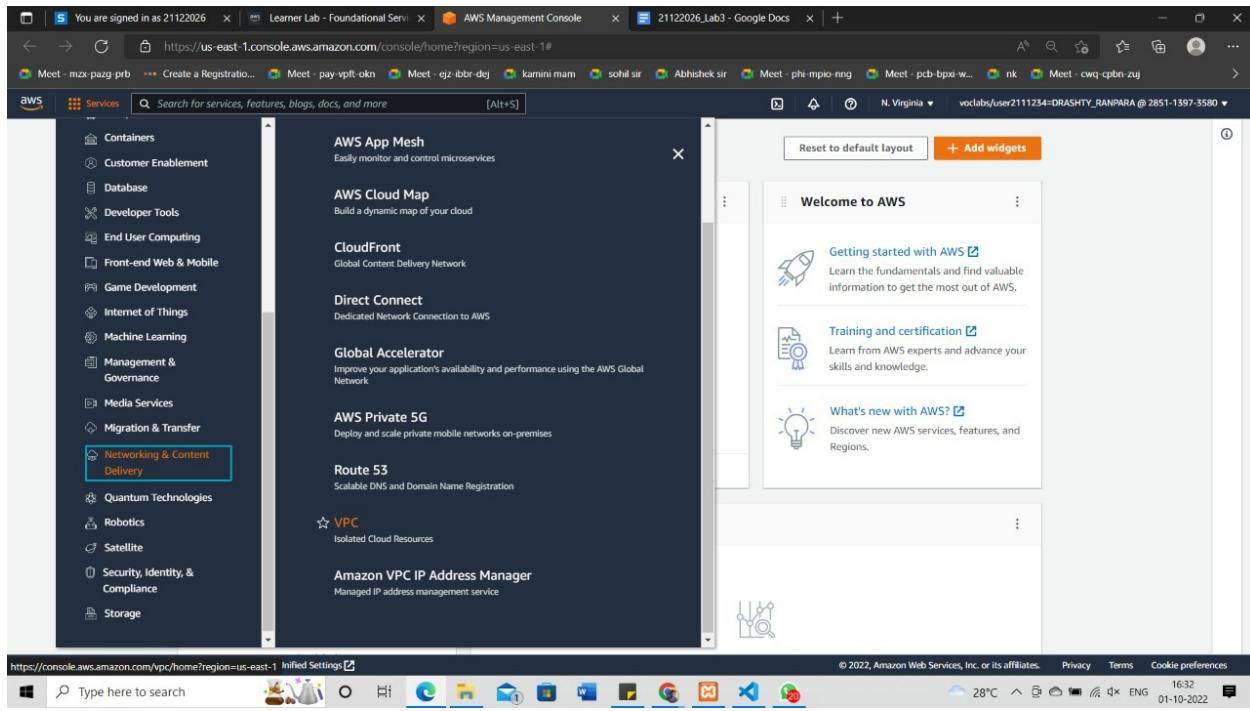
Name: - Drashty Ranpara Roll No: - 21122026

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

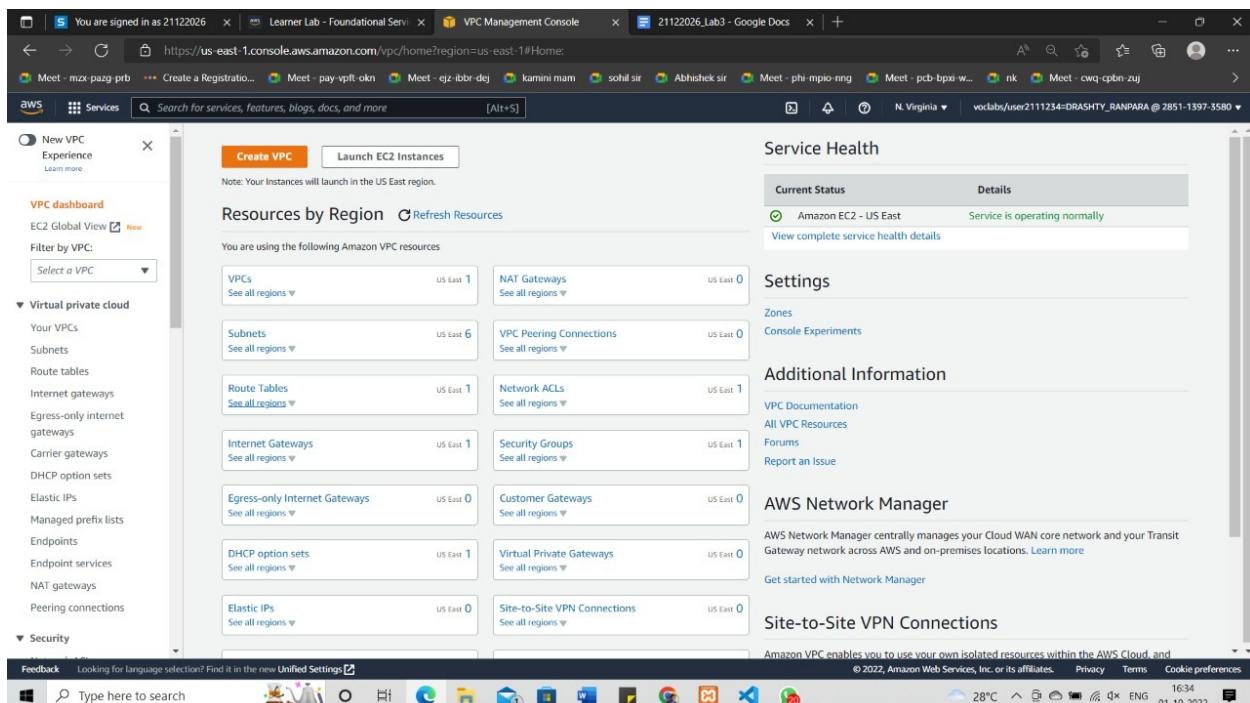
- Open the lab and start the lab



- 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.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only

VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
project

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16	65,536 IPs
-------------	------------

IPv6 CIDR block Info

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 28°C ENG 01-10-2022

- 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.

Preview

VPC Show details
Your AWS virtual network

Subnets (4) Subnets within this VPC

Route tables (3) Route network traffic to resources

us-east-1a

- project-subnet-public1-us-east-1a
- project-subnet-private1-us-east-1a

us-east-1b

- project-subnet-public2-us-east-1b
- project-subnet-private2-us-east-1b

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 28°C ENG 01-10-2022

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

The screenshot shows the AWS VPC Management Console interface. On the left, there's a configuration panel with fields for 'Number of public subnets' (set to 2), 'Number of private subnets' (set to 4), and other settings like NAT gateways and DNS options. On the right, a 'Preview' section shows a hierarchical network diagram. It starts with a 'VPC' node labeled 'project-vpc'. This VPC contains two 'Subnets (4)' groups: 'us-east-1a' and 'us-east-1b'. Each subnet group contains two subnets: 'project-subnet-public1-us-east-1a', 'project-subnet-private1-us-east-1a' in 'us-east-1a', and 'project-subnet-public2-us-east-1b', 'project-subnet-private2-us-east-1b' in 'us-east-1b'. Each subnet is associated with a 'Route tables (3)' group, which includes 'project-rtb-public', 'project-rtb-private1-us-east-1a', and 'project-rtb-private2-us-east-1b'.

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

The screenshot shows the 'Create VPC workflow' step titled 'Creating VPC Resources'. A message box says 'Thank you for using the new create VPC experience. Let us know what you think.' Below it, a list of steps is shown, all of which have been completed (indicated by a green checkmark icon). The steps include:

- Create VPC: vpc-09f08786c1e0d31eb
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-09f08786c1e0d31eb
- Create S3 endpoint: vpce-0ebeb6488b7a42b1f
- Create subnet
- Create subnet
- Create subnet
- Create subnet
- Create internet gateway
- Attach internet gateway to the VPC
- Create route table
- Create route
- Associate route table
- Associate route table
- Create route table
- Associate route table

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

VPC ID: vpc-09f08786c1e0d31eb

State: Available

DNS hostnames: Enabled

Main route table: rtb-0e0e141f97ec2e06

IPv4 CIDR: 10.0.0.0/16

Owner ID: 26174333352

Note: Failed to load rule groups

- 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

Compute

- AWS App Runner
- Batch
- EC2
- EC2 Image Builder
- Elastic Beanstalk
- Lambda
- Lightsail
- AWS Outposts
- Serverless Application Repository

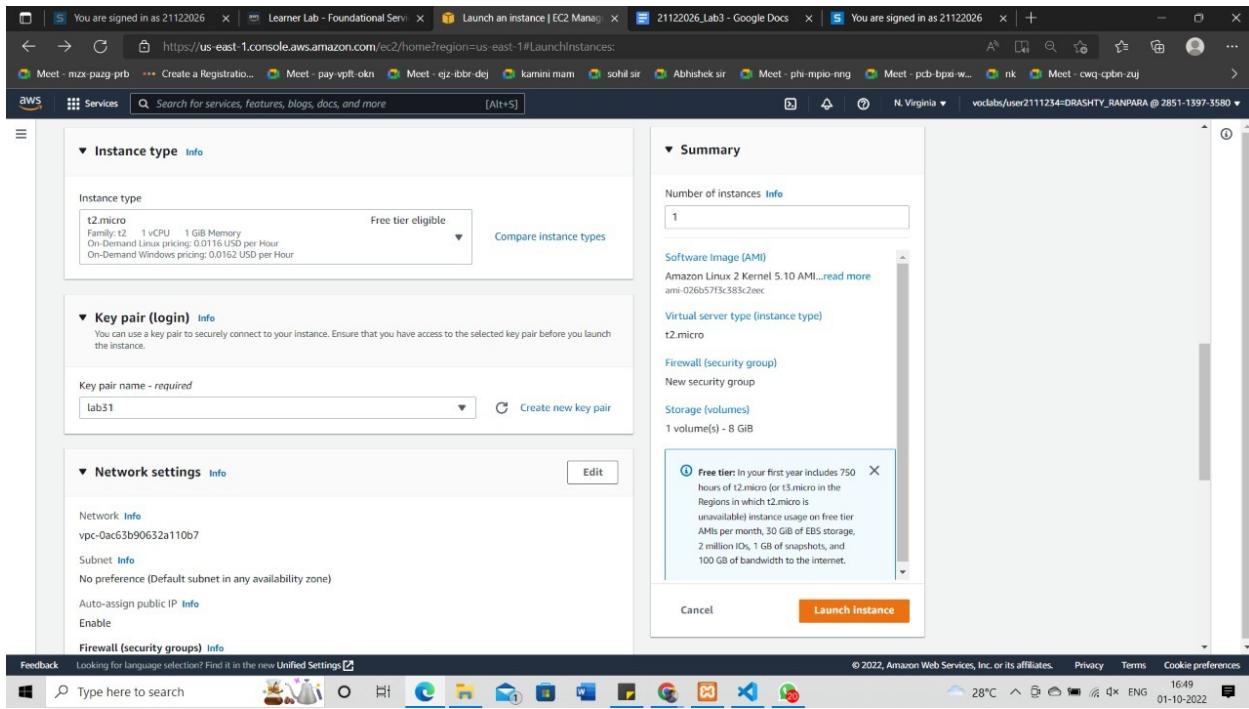
- You will get the Screen like this and Click on the Launch Instance

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area is titled 'Resources' and shows summary statistics for instances, dedicated hosts, elastic IPs, key pairs, load balancers, placement groups, security groups, and snapshots. Below this, there's a callout for launching Microsoft SQL Server Always On availability groups. To the right, there's a 'Service health' section showing the status of the service as 'operating normally'. Further right is an 'Account attributes' panel with information about supported platforms (VPC), default VPC (vpc-0ac63b90632a110b7), and other settings like EBS encryption and zones. At the bottom right, there's an 'Explore AWS' section with links to GuardDuty, Graviton, and price performance.

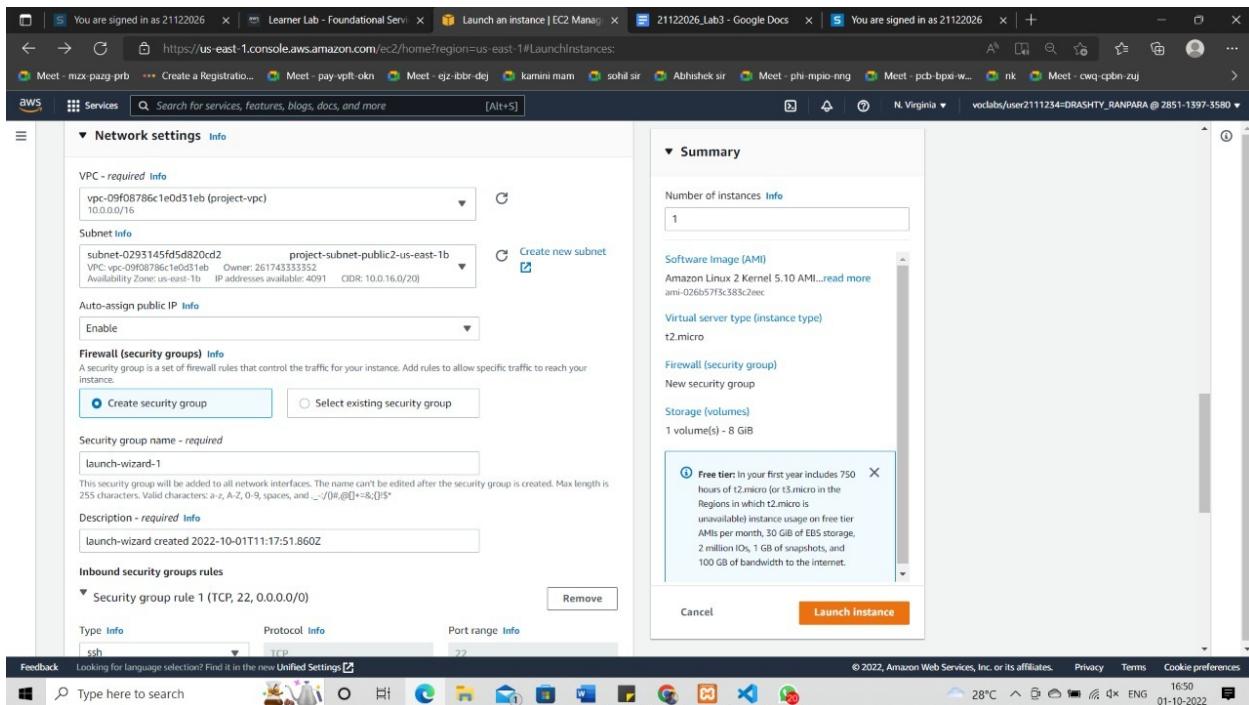
- Give the name of the instance Select the Application and OS Image for the instance

This screenshot shows the 'Launch an instance | EC2 Management' page. It starts with a 'Name and tags' section where 'lab3-1' is entered. Below that is a 'Application and OS Images (Amazon Machine Image)' section. A search bar is at the top, followed by a 'Quick Start' grid showing icons for Amazon Linux, macOS, Ubuntu, Windows, and Red Hat. A tooltip for the 'Quick Start' grid says 'Including AMIs from AWS Marketplace and the Community'. Below the grid, a specific AMI is selected: 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type'. This section includes details like 'Free tier eligible', 'Virtualization: hvm', and 'Root device type: ebs'. At the bottom of this section, there's a 'Description' field containing the text 'Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1 x86_64 HVM gp2'. The right side of the screen shows a 'Summary' panel with a 'Number of instances' set to 1, a 'Software image (AMI)' section for 'Amazon Linux 2 Kernel 5.10 AMI...', a 'Virtual server type (instance type)' set to 't2.micro', and a 'Firewall (security group)' section. A tooltip for the 'Free tier' says: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right is a 'Launch instance' button.

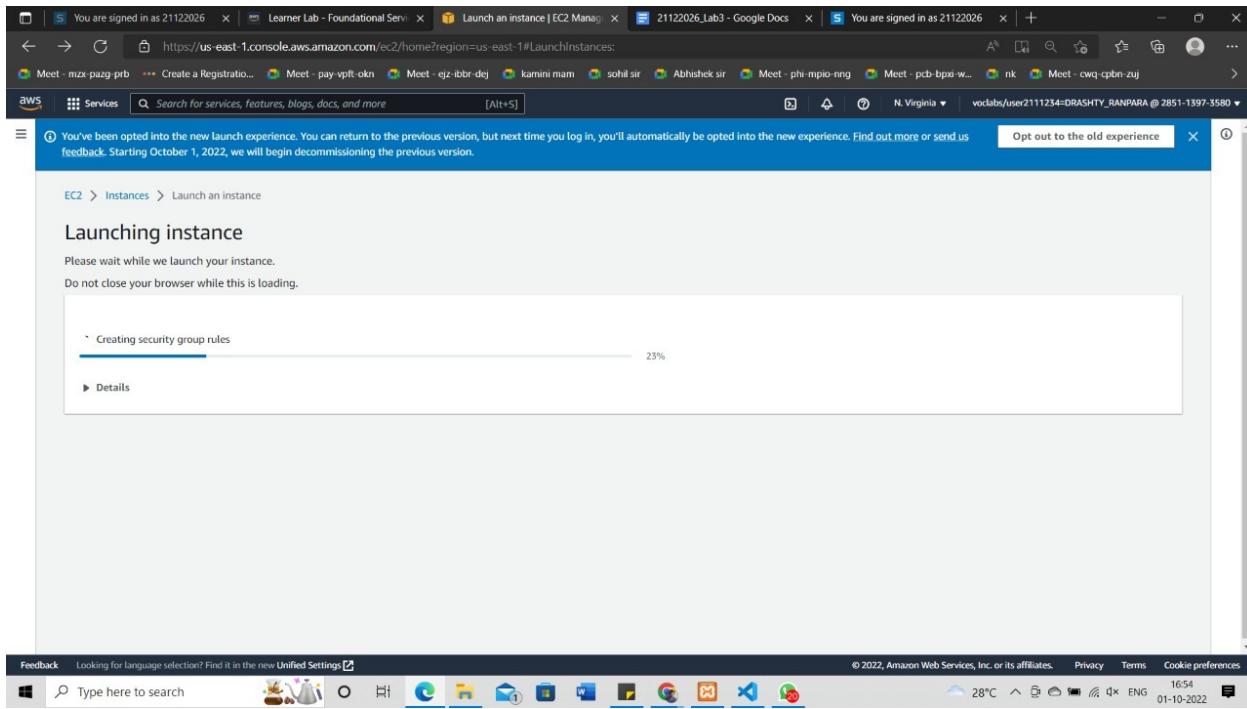
- Create the key value pair



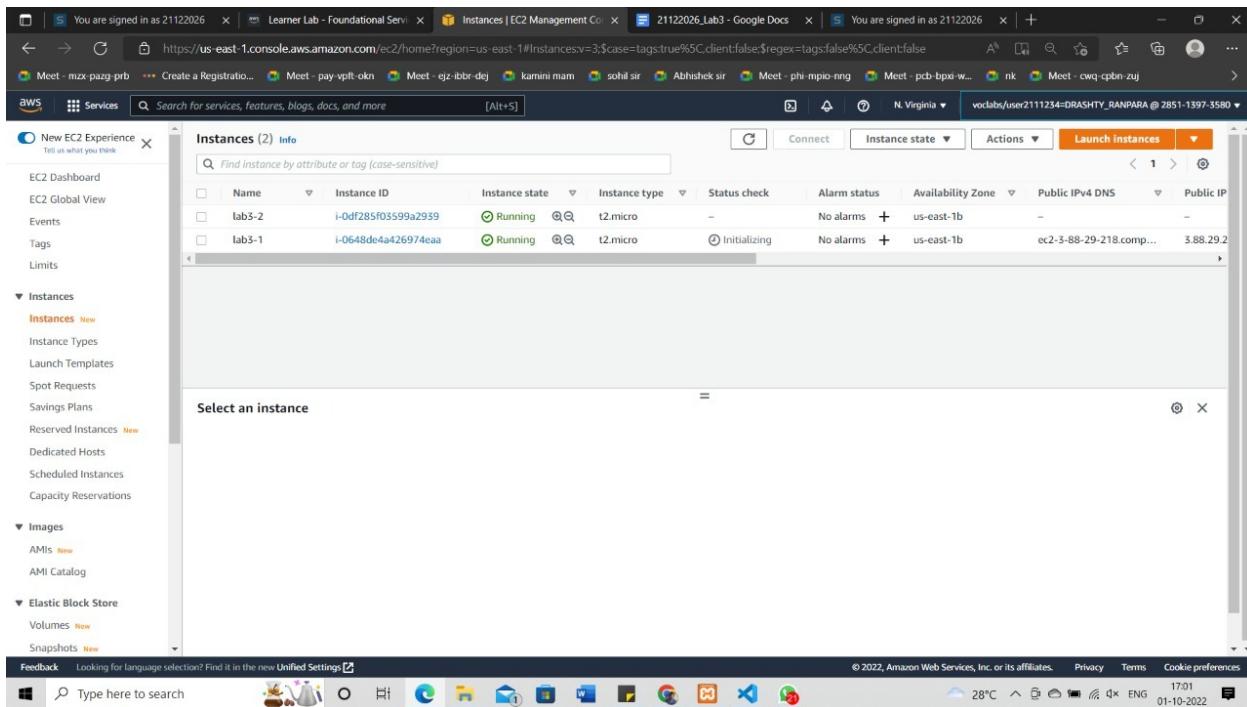
- Change the network setting. Select the VPC which you created and select the public subnet and Enable the Auto Assign public ip.



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



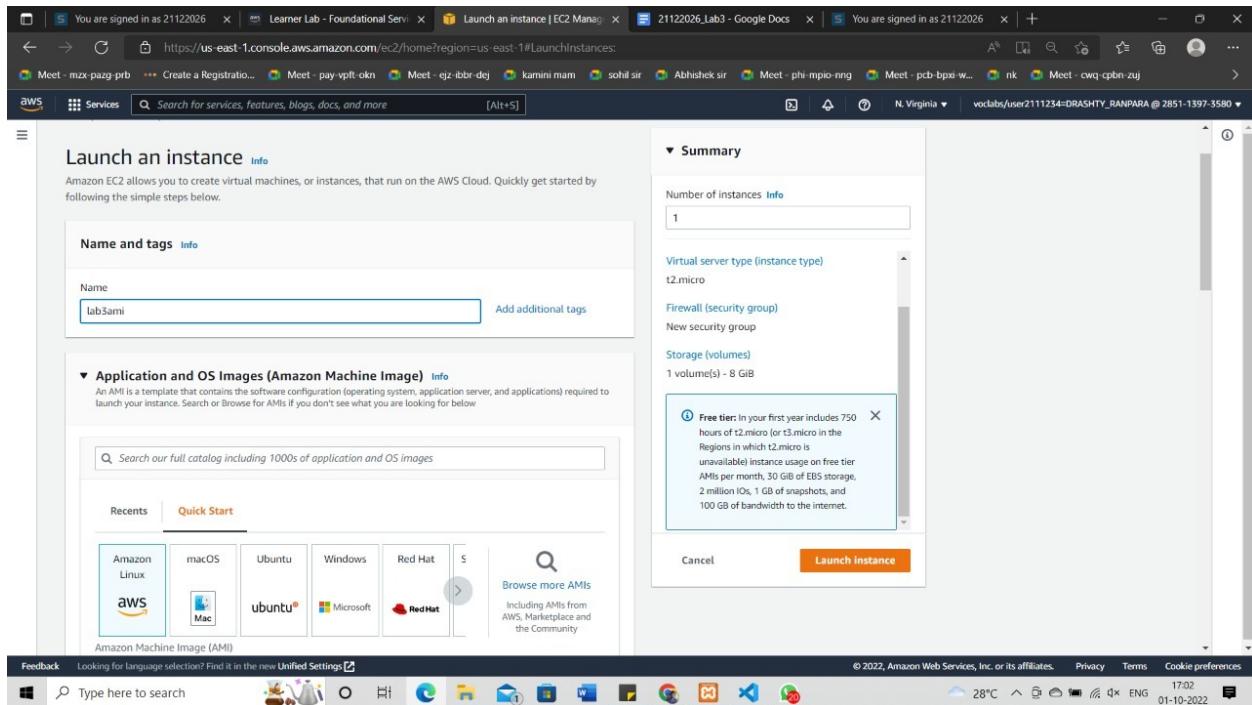
- Again same create second instance with the different tag but remain same the network setting we did for the first instance.
- After creating the second instance you will see the instances.



- 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.

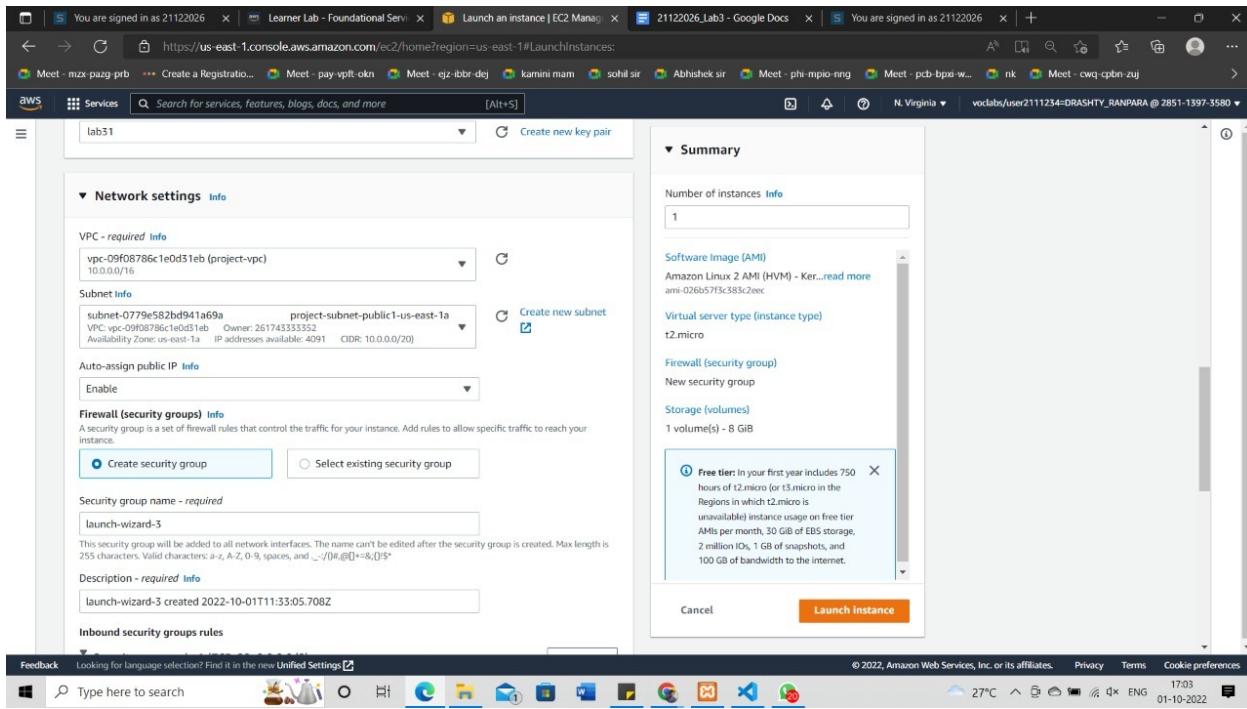
- Again Navigate to the computer->EC2.
- Click on the Launch Instance



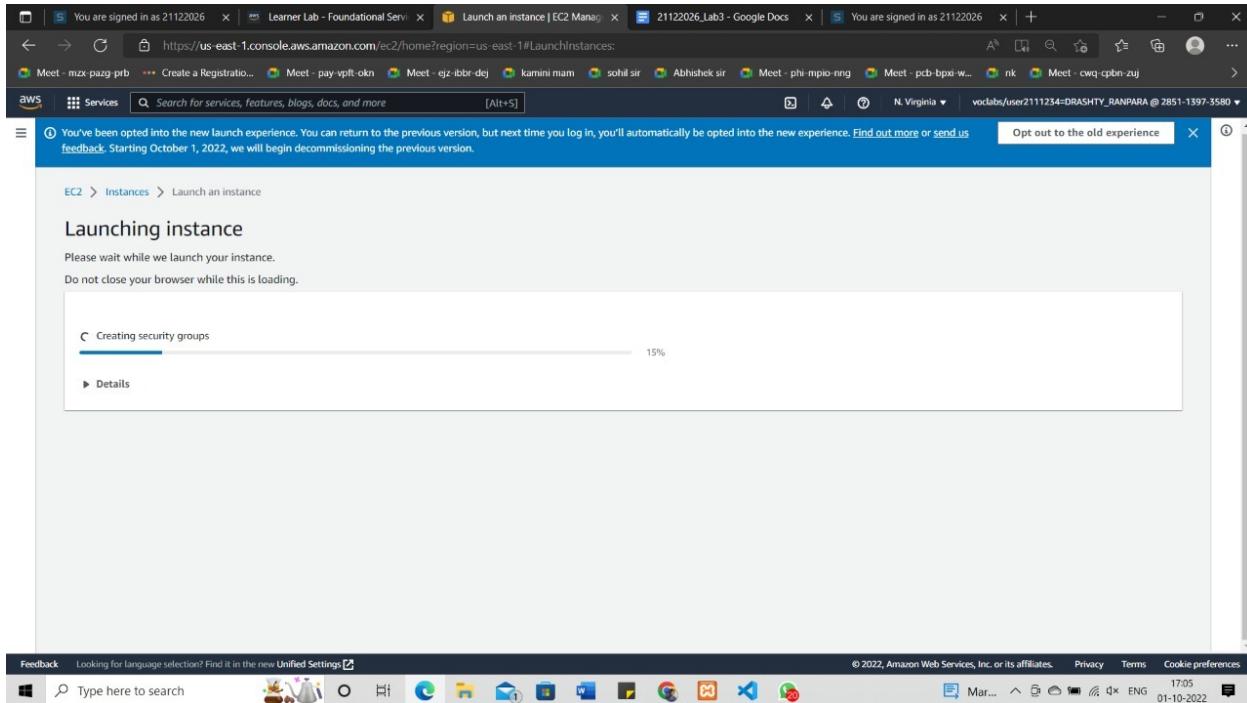
- Give the ame to the ami instance
- In the application and OS Images section click on the search button **Browse more AMIs**.
- Select the any ami for the creating your own ami.

- 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.

The screenshot shows the AWS EC2 Instances launch success page. At the top, there's a banner about the new launch experience. Below it, a green success message says "Successfully initiated launch of instance i-04129dde29cbde38f". A "Launch log" link is available. The "Next Steps" section includes links for estimated charges, connecting to the instance, and more resources. A "View all instances" button is at the bottom right.

- 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.

The screenshot shows the AWS EC2 Instances list page. On the right, for the instance "lab3ami", there's a "Actions" dropdown menu. Under "Image and templates", the "Create image" option is highlighted. The main table lists three instances: "lab3-2", "lab3-1", and "lab3ami". The "lab3ami" row has a checkmark next to it. The "Details" tab is selected for "lab3ami", showing its instance ID, state (Running), type (t2.micro), and other details like public and private IP addresses.

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

The screenshot shows the 'Create image' page in the AWS Management Console. The instance ID is set to 'i-04129dde29cbde38f (lab3ami)'. The image name is 'Dee', and the image description is 'Dee everywhere'. Under 'Instance volumes', there is one EBS volume of size 8 GiB, type 'EBS General Purpose S...', IOPS 100, throughput 100, and it is encrypted. A note states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' The 'Tags - optional' section is visible at the bottom.

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

The screenshot shows the 'Create image' page with the 'Tags - optional' section selected. The 'Tag image and snapshots together' option is selected, and a note says: 'Tag the image and the snapshots with the same tag.' The 'Create Image' button is visible at the bottom right.

- You will get the message like this.

The screenshot shows the AWS EC2 Instances page with three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
lab3-2	i-0df285f03599a2939	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	-	-
lab3-1	i-0648de4a426974eaa	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-88-29-218.comp...	3.88.29.2
lab3ami	i-04129dde29cbde38f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-207-192-31.co...	34.207.1...

A modal window titled "Select an instance" is overlaid on the page, asking the user to choose an instance to delete. The message in the modal reads: "Currently creating AMI ami-09e322829a24f5fb from instance i-04129dde29cbde38f. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI."

- 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.

The screenshot shows the AWS IAM Roles page with a list of roles:

Role Name	Description	Last Used
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked Role)	-
AWSServiceRoleForElastiCache	AWS Service: elasticache (Service-Linked Role)	-
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Service-Linked Role)	-
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
EMR_AutoScaling_DefaultRole	AWS Service: elasticsearch, and 1 more: View	-
EMR_DefaultRole	AWS Service: elasticmapreduce	-
EMR_EC2_DefaultRole	AWS Service: ec2	-
LabRole	AWS Service: secretsmanager, and 38 more: View	17 minutes ago
robomaker_students	AWS Service: lambda, and 3 more: View	-
vocareum	Account: 962395053709	-
vocilabs	Account: 962395053709	-
vocstartsoft	Account: 962395053709	-

A modal window titled "Roles Anywhere" is open at the bottom, with the "Info" tab selected. It says: "Authenticate your non AWS workloads and securely provide access to AWS services." A "Manage" button is located in the top right corner of this modal.

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

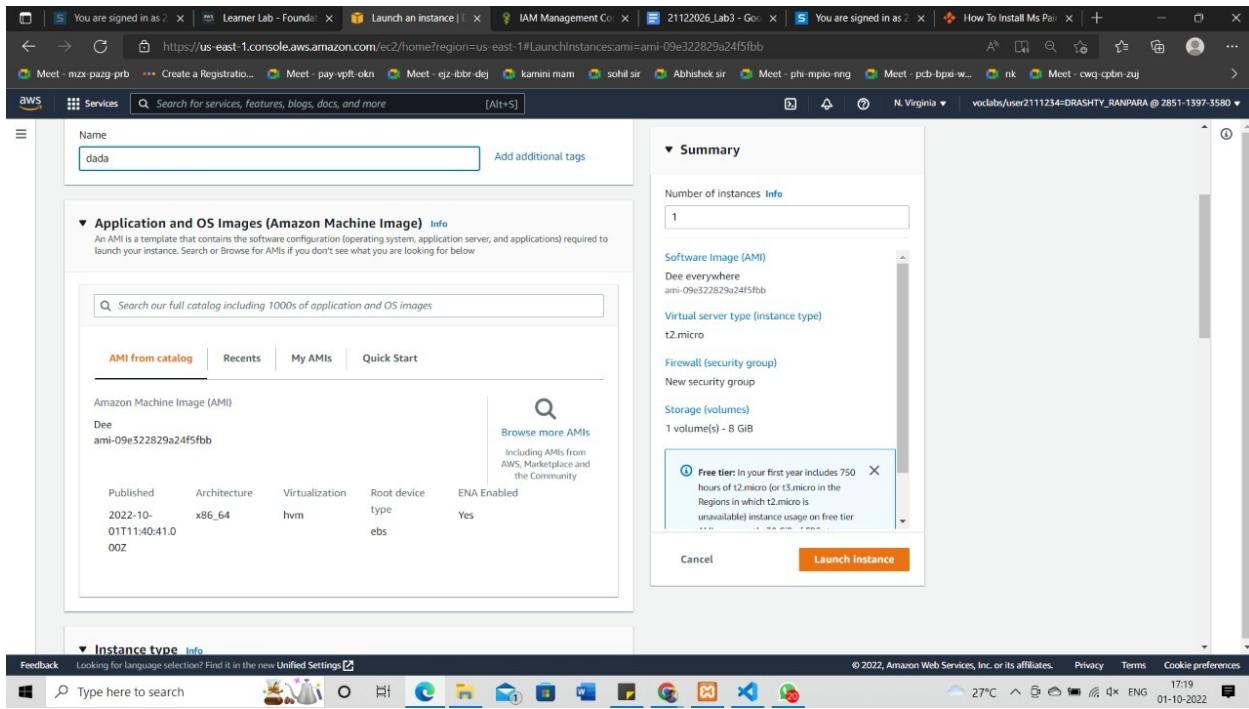
- Write S3 in the find policies search box and select the **AmazonS3FullAccess** policies and click on Attach policies button.

- You can see the policies after attach.

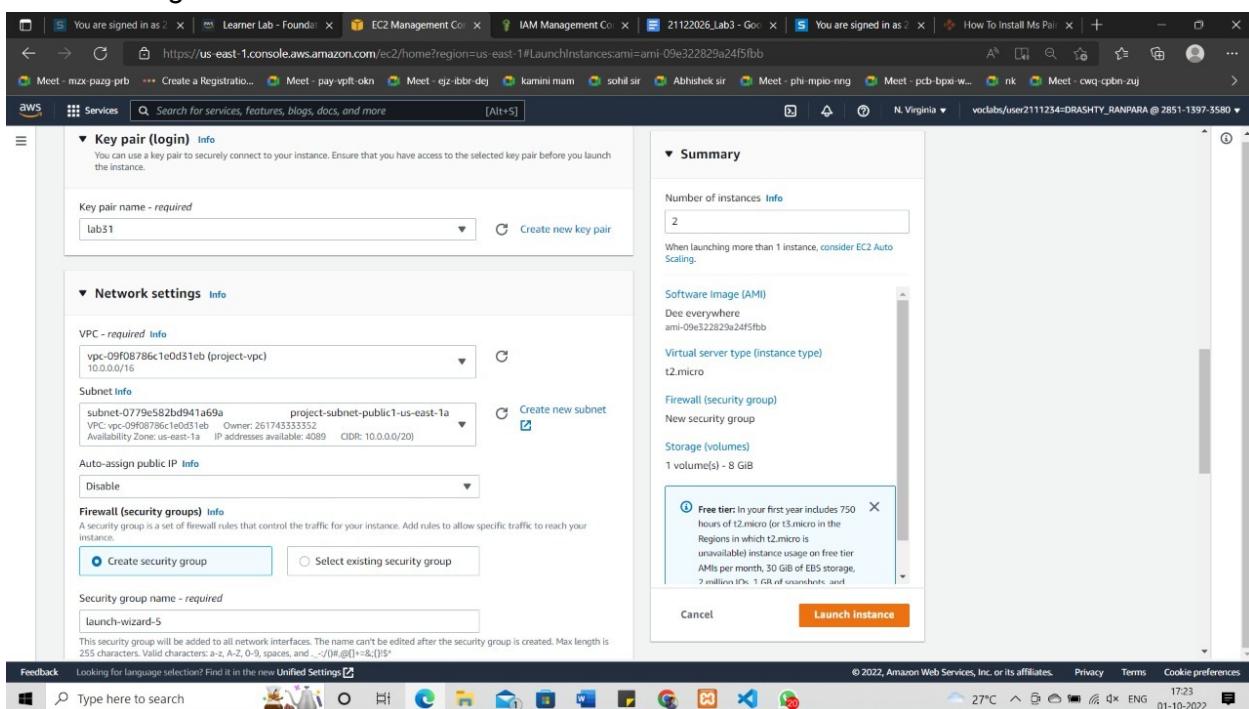
- Now again move to the EC2 Dashboard and go to AMIs.

Name	AMI ID	AMI name	Source	Owner	Visibility	Status
-	ami-09e322829a24f5fb	Dee	26174333352/Dee	26174333352	Private	Available

- 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.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
lab3-2	i-0df285f03599a2939	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	-	-
lab3-1	i-0648de4a426974eaa	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-88-29-218.comp...	3.88.29.2
lab3ami	i-04129ddc29bde38f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-207-192-31.co...	34.207.1
dada	i-050ab89df24a2f6a1	Terminated	t2.micro	Initializing	No alarms	us-east-1a	-	3.83.38.2
dada	i-095b8439be7dde112	Running	t2.micro	-	No alarms	us-east-1a	-	-
dada	i-093edfaae787c7974	Running	t2.micro	-	No alarms	us-east-1a	-	-

- 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.

Elastic IP addresses (1/1)

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID
-	50.17.182.98	Public IP	eipalloc-00f2b45cc5f79c05a	-	-

Summary

Allocated IPv4 address: 50.17.182.98 | Type: Public IP | Allocation ID: eipalloc-00f2b45cc5f79c05a | Reverse DNS record: - | Associated instance ID: - | Private IP address: -

- 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.

Elastic IP address: 50.17.182.98

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance

Network interface

⚠️ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

Instance

Private IP address
The private IP address with which to associate the Elastic IP address.

Reassociation
Specify whether the Elastic IP address can be reassigned with a different resource if it already associated with a resource.
 Allow this Elastic IP address to be reassigned

Associate

- You will get the message like this.

The screenshot shows the AWS EC2 Management Console with a green success banner at the top stating "Elastic IP address associated successfully." Below the banner, the "Elastic IP addresses (1/1)" table is displayed. The table has columns for Name, Allocated IPv4 address, Type, Allocation ID, Reverse DNS record, and Associated instance ID. One row is present with the values: Name is empty, Allocated IPv4 address is 50.17.182.98, Type is Public IP, Allocation ID is eipalloc-00f2b45cc5f79c05a, Reverse DNS record is empty, and Associated instance ID is i-095b8439be7dde112.

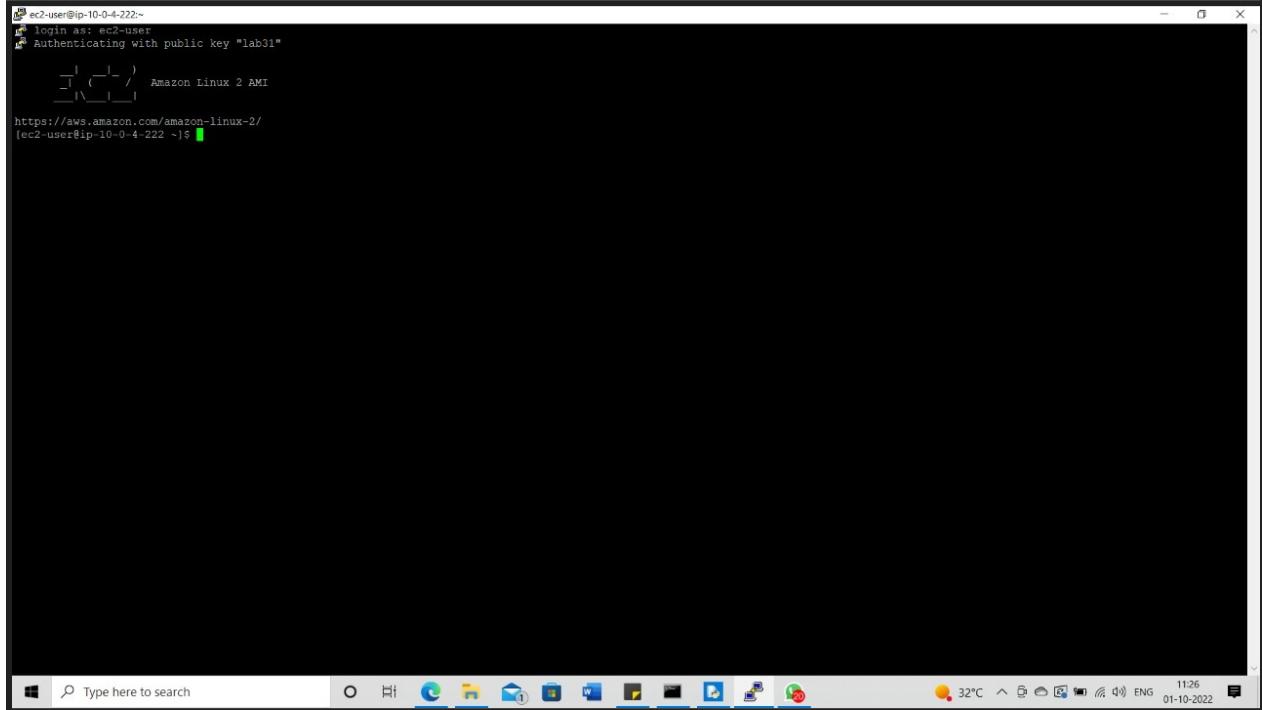
- For confirmation you can see the Associated instance id.

This screenshot shows the same AWS EC2 Management Console interface, but the table below the banner is now titled "Elastic IP addresses (1/1)" and includes additional columns: Reverse DNS record, Associated instance ID, Private IP address, Association ID, Network interface owner account, and Network Border Group. The single row shows the same data as the previous screenshot, plus a Private IP address of 10.0.13.180 and a Network interface owner account of 261743333552.

- 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



```
ec2-user@ip-10-0-4-222:~$ login as: ec2-user
ec2-user@ip-10-0-4-222:~$ Authenticating with public key "lab31"
ec2-user@ip-10-0-4-222:~$ curl -s https://aws.amazon.com/amazon-linux-2/ | tee amazon-linux-extras.repo
ec2-user@ip-10-0-4-222:~$
```

- 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"  
[ec2-user@ip-10-0-4-222 ~]$ ls  
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  
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-mysqld, 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  
amzn2extra-docker  
amzn2extra-kernel-5.10  
amzn2extra-mariadb10.5  
amzn2extra-php8.0  
(1/11): amzn2extra-Cores/2/x86_64/group.gz  
(2/11): amzn2extra-Cores/2/x86_64/updateinfo  
(3/11): amzn2extra-docker/2/x86_64/updateinfo  
(4/11): amzn2extra-docker/2/x86_64/primary_db  
(5/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo  
(6/11): amzn2extra-mariadb10.5/2/x86_64/updateinfo  
(7/11): amzn2extra-php8.0/2/x86_64/updateinfo  
(8/11): amzn2extra-kernel-5.10/2/x86_64/primary_db  
(9/11): amzn2extra-mariadb10.5/2/x86_64/primary_db  
(10/11): amzn2extra-php8.0/2/x86_64/primary_db  
  
| 3.7 kB 00:00:00  
| 3.0 kB 00:00:00  
| 3.0 kB 00:00:00  
| 3.0 kB 00:00:00  
| 3.0 kB 00:00:00  
| 2.9 kB 00:00:00  
| 488 kB 00:00:00  
| 6.4 kB 00:00:00  
| 93 kB 00:00:00  
| 18 kB 00:00:00  
| 76 B 00:00:00  
| 76 B 00:00:00  
| 11 MB 00:00:00  
| 83 kB 00:00:00  
| 181 kB 00:00:00  
  
Type here to search               32°C ENG 11:27 01-10-2022
```

```
ec2-user@ip-10-0-4-22:~
```

(28/46): pcre-devel-8.32-17.amzn2.0.2.x86_64.rpm
(29/46): openssl-devel-1.0.2k-24.amzn2.0.4.x86_64.rpm
(30/46): perl-COMPRESS-Raw_BZip2-2.061-3.amzn2.0.2.x86_64.rpm
(31/46): perl-BDB-MYSQL-4.023-6.amzn2.x86_64.rpm
(32/46): perl-DBI-1.627-4.amzn2.0.2.x86_64.rpm
(33/46): perl-Data-Dumper-2.145-3.amzn2.0.2.x86_64.rpm
(34/46): perl-IO-Compress-2.061-2.amzn2.noarch.rpm
(35/46): perl-Compress-Raw_Zlib-2.061-4.amzn2.0.2.x86_64.rpm
(36/46): perl-PLRPC-0.2020-14.amzn2.noarch.rpm
(37/46): perl-TermReadKey-2.30-20.amzn2.0.2.x86_64.rpm
(38/46): perl-Net-Daemon-0.48-5.amzn2.noarch.rpm
(39/46): php-common-8.0.20-1.amzn2.x86_64.rpm
(40/46): php-pdo-8.0.20-1.amzn2.x86_64.rpm
(41/46): php-mysqlind-8.0.20-1.amzn2.x86_64.rpm
(42/46): php-cli-8.0.20-1.amzn2.x86_64.rpm
(43/46): php-pdo-8.0.20-1.amzn2.x86_64.rpm
(44/46): postgresql-libs-9.2.24-8.amzn2.x86_64.rpm
(45/46): zlib-devel-1.2.7-19.amzn2.0.1.x86_64.rpm
(46/46): sphinx-2.2.11-5.amzn2.0.1.x86_64.rpm

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-server-10.5.10-2.amzn2.0.1.x86_64	4/47
Installing : mariadb-libs-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-1.5.0-3.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 : liblberto-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 : libssh2client-2.2.11-5.amzn2.0.1.x86_64	19/47
Installing : libcurl-err-devel-1.42.0-19.amzn2.x86_64	20/47
Installing : perl-Compress-Raw_Bzip2-2.061-4.amzn2.0.2.x86_64	22/47
Installing : perl-PLRPC-0.2020-14.amzn2.noarch	23/47
Installing : perl-WHICH-677-4.amzn2.0.3.x86_64	24/47
Installing : libsep-devel-2.5-9.1.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:~$ rpm -q --available | grep -i lustre
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
24 epel                    available   [ =7.11 =stable ]
25 testing                 available   [ =1.0 =stable ]
26 ecs                     available   [ =stable ]
27 corretto8                available   \
28     [ =1.8.0_192 =1.8.0_202 =1.8.0_212 =1.8.0_222 =1.8.0_232
29     =1.8.0_242 =stable ]
30 firecracker              available   [ =0.11 =stable ]
31 goolang1.11              available   \
32     [ =1.11.11 =1.11.13 =stable ]
33 squid4                  available   [ =4 =stable ]
34 lustre2.10                available   \
35     [ =2.10.5 =2.10.8 =stable ]
36 java-openjdk11           available   [ =11 =stable ]
37 lynis                   available   [ =stable ]
38 BCC                     available   [ =0.x =stable ]
39 mono                     available   [ =5.x =stable ]
40 nginx1                  available   [ =stable ]
41 ruby2.6                  available   [ =2.6 =stable ]
42 mock                     available   [ =stable ]
43 postgresql11             available   [ =11 =stable ]
44 phpf7.4                  available   [ =stable ]
45 livepatch                available   [ =stable ]
46 monit                   available   [ =stable ]
47 haproxy2                 available   [ =stable ]
48 collectd                 available   [ =stable ]
49 aws-nitro-enclaves-cli  available   [ =stable ]
50 R4                      available   [ =stable ]
51 kernel-5.4                available   [ =stable ]
52 selinux-ng                available   [ =stable ]
53 php8.0-latest            enabled    [ =stable ]
54 tomcat9                  available   [ =stable ]
55 mariadb10.5=latest       enabled    [ =stable ]
56 kernel-5.10-latest       enabled    [ =stable ]
57 redis                    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 ~]$
```



```
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-httdp for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod httpd 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-httdp.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_httpd.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
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    122 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    99 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    19 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    19 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    24 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    88 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    31 k
httpd           x86_64      2.4.54-1.amzn2          amzn2-core    149 k
Transaction Summary
Install 1 Package (+8 Dependent packages)

[ec2-user@ip-10-0-4-222 ~]$
```



```

[ec2-user@ip-10-0-4-222-~]
httpd-filesystem           noarch          x86_64          2.4.54-1.amzn2
httpd-tools                 noarch          x86_64          2.4.54-1.amzn2
mailcap                     noarch          x86_64          2.1.41-2.amzn2
mod_http2                  noarch          x86_64          1.15.19-1.amzn2.0.1

Transaction Summary
Install 1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Downloading packages:
(1/9): apr-util-1.6.1-5.amzn2.0.2.x86_64.rpm | 99 kB 00:00:00
(2/9): apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64.rpm | 19 kB 00:00:00
(3/9): generic-logos-httdp-18.0.0-4.amzn2.noarch.rpm | 122 kB 00:00:00
(4/9): httpd-filesystem-2.4.54-1.amzn2.noarch.rpm | 19 kB 00:00:00
(5/9): httpd-tools-2.4.54-1.amzn2.noarch.rpm | 24 kB 00:00:00
(6/9): httpd-tools-2.4.54-1.amzn2.x86_64.rpm | 88 kB 00:00:00
(7/9): httpd-2.4.54-1.amzn2.x86_64.rpm | 1.4 MB 00:00:00
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm | 31 kB 00:00:00
(9/9): mod_http2-1.15.19-1.amzn2.0.1.x86_64.rpm | 149 kB 00:00:00

Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.7.0-9.amzn2.x86_64
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64
  Installing : httpd-tools-2.4.54-1.amzn2.x86_64
  Installing : httpd-filesystem-2.4.54-1.amzn2.noarch
  Installing : generic-logos-httdp-18.0.0-4.amzn2.noarch
  Installing : mailcap-2.1.41-2.amzn2.noarch
  Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64
  Installing : httpd-2.4.54-1.amzn2.x86_64
  Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64
  Verifying : httpd-tools-2.4.54-1.amzn2.x86_64
  Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64
  Verifying : httpd-2.4.54-1.amzn2.x86_64
  Verifying : mailcap-2.1.41-2.amzn2.noarch
  Verifying : generic-logos-httdp-18.0.0-4.amzn2.noarch
  Verifying : httpd-filesystem-2.4.54-1.amzn2.noarch
  Verifying : apr-1.7.0-9.amzn2.x86_64

Installed:
  httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
  [1/9] Type here to search  ○  ⌂  32°C  ⌂  ENG  11:30
  [2/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [3/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [4/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [5/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [6/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [7/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [8/9]  ⌂  32°C  ⌂  ENG  01-10-2022
  [9/9]  ⌂  32°C  ⌂  ENG  01-10-2022

```

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

```

[ec2-user@ip-10-0-4-222:~]
$ login as: ec2-user
$ Authenticating with public key "lab01"
Last login: Sat Oct 1 05:56:20 2022 from 106.77.77.145
[ec2-user@ip-10-0-4-222 ~]$ ls
[ec2-user@ip-10-0-4-222 ~]$ cd /var/www
[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]$ 

[ec2-user@ip-10-0-4-222 inc]$ 

```

OneDrive

Screenshot saved
The screenshot was added to your OneDrive.

32°C ⌂ ENG 11:38 01-10-2022

```
ec2-user@ip-10-0-4-222:/var/www/html
$ login as: ec2-user
$ Authenticating with public key "Jab31"
Last login: Sat Oct 1 05:56:20 2022 from 106.77.77.145
[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 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]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$ nano SamplePage.php
[ec2-user@ip-10-0-4-222 html]$
```

```
ec2-user@ip-10-0-4-222:/var/www/inc
$ GNU nano 2.9.8
dbinfo.inc
$ ppp
define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');

?>
```

```

ec2-user@ip-10-0-4-222:/var/www/html
GNU nano 2.5.8                                         SamplePage.php

<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
</body>
</html>

/* Connect to MySQL and select the database. */
$connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " . mysqli_connect_error();

$database = mysqli_select_db($connection, DB_DATABASE);

/* Ensure that the EMPLOYEES table exists. */
verifyEmployeeTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strien($employee_name) || strien($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
}

?>

<!-- Input form -->
<form action=<?php echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
<table border="0">
    <tr>
        <td>NAME</td>
        <td>ADDRESS</td>
    </tr>
    <tr>
        <td><input type="text" name="NAME" maxlength="45" size="30" />
        </td>
        <td><input type="text" name="ADDRESS" maxlength="90" size="60" />
        </td>
    </tr>
    <tr>
        <td><input type="submit" value="Add Data" />
        </td>
    </tr>
</table>

```

Get Help Write Out Where Is Cut Text Uncut Text Justify Undo Mark Text To Bracket Previous Back
 Exit Read File Replace Copy Text Redo Go To Line Redo WhereIs Next Next Forward
 Type here to search 32°C ENG 11:41
 01-10-2022

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

