

Lesson 03 Demo 05

Setting Up a Web Server on an EC2 Instance

Objective: To demonstrate the process of setting up a web server on AWS, including creating a VPC, configuring essential components, and launching an EC2 instance with a basic web server

Tools required: AWS Management Console

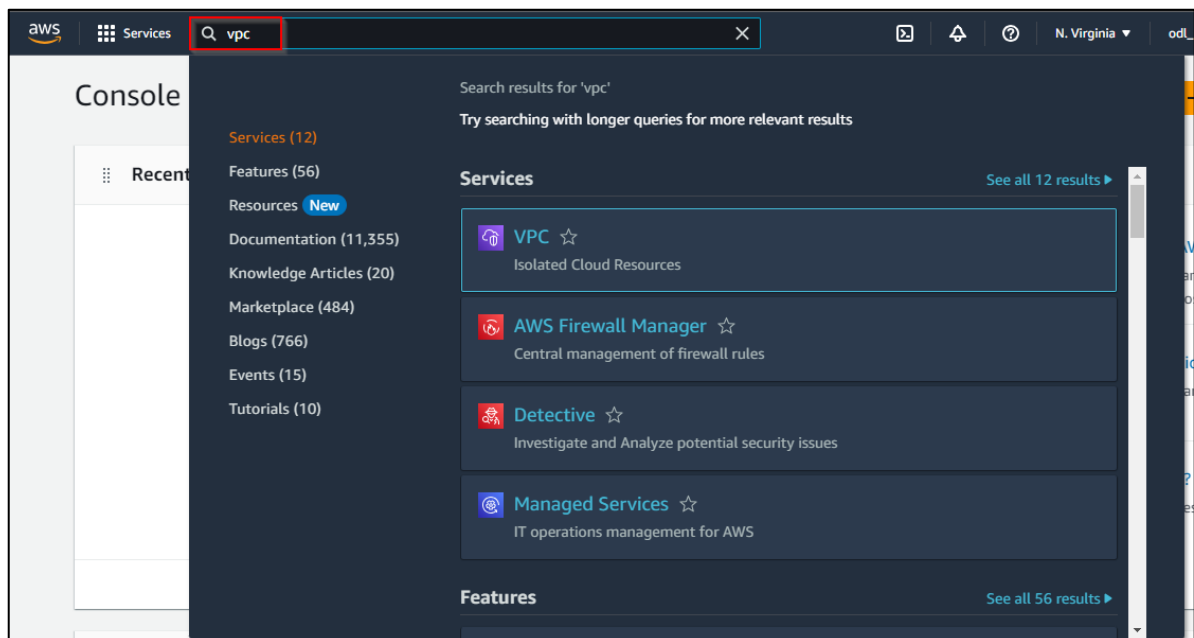
Prerequisites: NA

Steps to be followed:

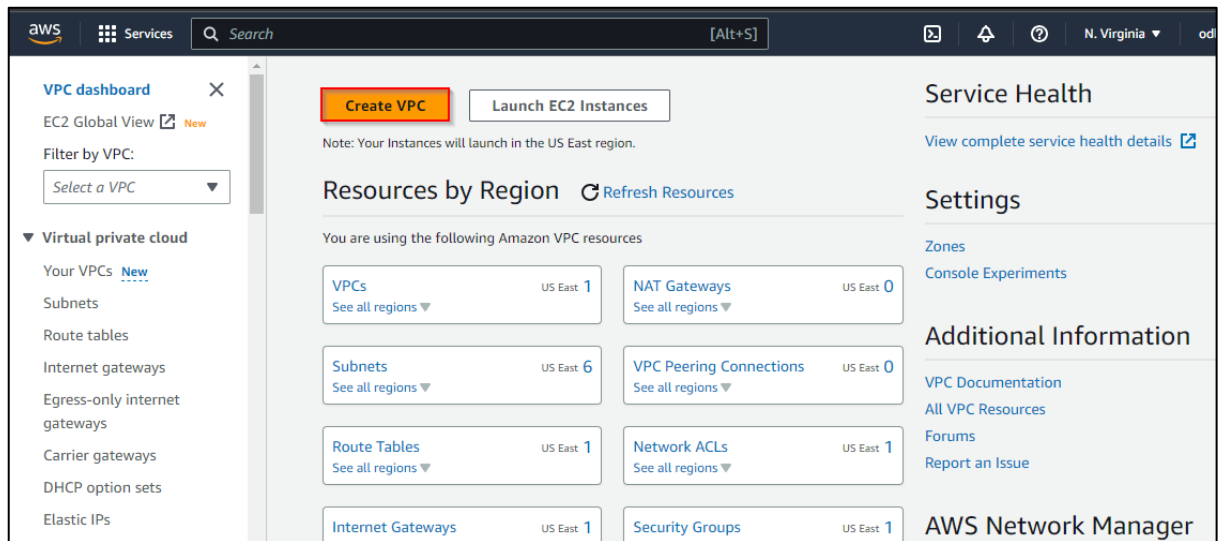
1. Create a VPC, a subnet, and an internet gateway
2. Create Route tables
3. Create an EC2 web server instance

Step 1: Create a VPC, a subnet, and an internet gateway

1.1 In the console navigation pane, search for and select **VPC**



1.2 In the VPC dashboard, click on **Create VPC**



1.3 Create a name tag, such as **my-subnets**, and set the IPv4 CIDR to **10.0.0.0/24**

The screenshot shows the 'Create VPC' wizard in the AWS console. The 'VPC settings' section is active. Under 'Resources to create', the 'VPC only' radio button is selected. The 'Name tag - optional' field is filled with 'my-subnets'. Under 'IPv4 CIDR block', the 'IPv4 CIDR manual input' radio button is selected, and the 'IPv4 CIDR' field is filled with '10.0.0.0/24'.

1.4 Click on **Create VPC**

Tenancy [Info](#)

Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Value - optional**

Q Name X Q my-subnets X Remove tag

Add tag

You can add 49 more tags

Cancel **Create VPC**

VPC dashboard X

EC2 Global View [New](#)

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs [New](#)

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

You successfully created vpc-09cbeeeb7ef3c059b / my-subnets

VPC > Your VPCs > vpc-09cbeeeb7ef3c059b

vpc-09cbeeeb7ef3c059b / my-subnets [Actions](#)

Details [Info](#)

VPC ID vpc-09cbeeeb7ef3c059b	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-040a7245f6d78c68d	Main route table rtb-0eca8e9b958f60aad	Main network ACL acl-05d9cad570a04099e
Default VPC No	IPv4 CIDR 10.0.0.0/24	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 025995114484	

[Resource map](#) [New](#) [CIDRs](#) [Flow logs](#) [Tags](#)

1.5 Navigate to **Subnets** on the left side of the screen and click **Create subnet**

aws Services Search [Alt+S] N. Virginia od_user_1036229 @ 0259-9511-4484

VPC dashboard X

EC2 Global View [New](#)

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs [New](#)

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

Subnets (6) [Info](#)

Find resources by attribute or tag

[Actions](#) **Create subnet**

	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	-	subnet-0ccdc947220031deb	Available	vpc-034cc6a8ed7395da5	172.31.0.0/24
<input type="checkbox"/>	-	subnet-0cfe163b17a96e66f	Available	vpc-034cc6a8ed7395da5	172.31.0.0/24
<input type="checkbox"/>	-	subnet-04492488ceb601876	Available	vpc-034cc6a8ed7395da5	172.31.0.0/24
<input type="checkbox"/>	-	subnet-0306cad2bea6607a3	Available	vpc-034cc6a8ed7395da5	172.31.0.0/24
<input type="checkbox"/>	-	subnet-096568035ddc1f771	Available	vpc-034cc6a8ed7395da5	172.31.0.0/24

Select a subnet

1.6 Select the previously **created VPC**, name the subnet as **my-subnets**, and click on **Create subnet**

aws Services Search [Alt+S]

VPC > Subnets > Create subnet

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

VPC ID	CIDR block	Default
vpc-09cbeeeb7ef3c059b (my-subnets)	10.0.0.0/24	✓
vpc-034cc6a8ed7395da5	172.31.0.0/16	(default)

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Create a tag with a key of 'Name' and a value that you specify.

my-subnets

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

IPv4 CIDR block [Info](#)

10.0.0.0/24

▼ **Tags - optional**

Key	Value - optional	
Name	my-subnets	Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel **Create subnet**

1.7 Enter an IPv4 CIDR block for the subnet, such as 10.0.0.0/24

The screenshot shows the AWS console interface for creating a subnet. The 'IPv4 CIDR block' field is highlighted with a red box and contains the text '10.0.0.0/24'. The 'Availability Zone' is set to 'No preference'. The 'Tags' section shows a key-value pair: 'Name' with value 'my-subnet'.

1.8 Navigate to the Internet gateways, and click on Create internet gateway

The screenshot shows the AWS console interface for managing Internet gateways. The 'Create internet gateway' button is highlighted with a red box. The table below shows the existing Internet gateway.

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0ccd995781dee5d15	Attached	vpc-034cc6a8ed7395da5	0259951

1.9 Name it as **IG-4**, and click on **Create internet gateway**

aws Services Search [Alt+S]

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

IG-4

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Q Name X Q IG-4 X Remove

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

1.10 Select the created internet gateway, click **Actions**, and choose **Attach to VPC**

aws Services Search [Alt+S]

VPC dashboard X

EC2 Global View New

Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

The following internet gateway was created: igw-038247b56eb71d418 - IG-4. You can now attach to a VPC to enable the VPC to communicate with the internet. Attach to a VPC

Internet gateways (1/2) Info

Filter internet gateways

	Name	Internet gateway ID	State	Owner
<input checked="" type="checkbox"/>	IG-4	igw-038247b56eb71d418	Detached	02599
<input type="checkbox"/>	-	igw-0ccdd995781dee5d15	Attached	02599

igw-038247b56eb71d418 / IG-4

Actions

- View details
- Attach to VPC
- Detach from VPC
- Manage tags
- Delete internet gateway

1.11 Select the **Available VPCs**, and click on **Attach internet gateway**

VPC > Internet gateways > Attach to VPC (igw-038247b56eb71d418)

Attach to VPC (igw-038247b56eb71d418) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

▶ AWS Command Line Interface command

[Cancel](#) [Attach internet gateway](#)

Step 2: Create Route tables

2.1 Click on **Route tables** > **Create route table**

aws Services Search [Alt+S] N. Virginia odl_user_1036229 @ 0259-9511-4484

VPC dashboard X

EC2 Global View [New](#)

Filter by VPC: [Select a VPC](#)

Virtual private cloud

- Your VPCs [New](#)
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet

Route tables (2) [Info](#)

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associati...	Edge associations	Main
<input type="checkbox"/>	-	rtb-02686a8b2369fb172	-	-	Yes
<input type="checkbox"/>	-	rtb-0eca8e9b958f60aad	-	-	Yes

[Create route table](#)

2.2 Name the route table as **route04**, select the VPC, and click **Create route table**

connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

route04.

VPC
The VPC to use for this route table.

vpc-09cbeeb7ef3c059b (my-subnets)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Value - optional**

Q Name X Q route04. X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

2.3 Under the **Edit routes** section, click on **Add route** and save the changes

VPC > Route tables > rtb-0aaef0df466906018 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/24	Q local X	Active	No
Q	Q	-	No Remove

Add route

Cancel Preview Save changes

2.4 Add the destination value as **0.0.0.0/0**, set the target as the **IG-4 internet gateway**, and click on **Save changes**

VPC > Route tables > rtb-0aaef0df466906018 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/24	local	Active	No
0.0.0.0/0	igw-086b9ff94e0be595b	-	No

[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

2.5 Click on **Actions**, and select **Edit subnet associations**

aws Services Search [Alt+S] N. Virginia odl_user_1036229 @ 0259-9511-4484

VPC dashboard EC2 Global View New

Filter by VPC: Select a VPC

Virtual private cloud

- Your VPCs New
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

Route table rtb-068aad3948de21fc4 | route04. was created successfully.

VPC > Route tables > rtb-068aad3948de21fc4

rtb-068aad3948de21fc4 / route04.

You can now check network connectivity with Reachability Analyzer

[Reachability Analyzer](#)

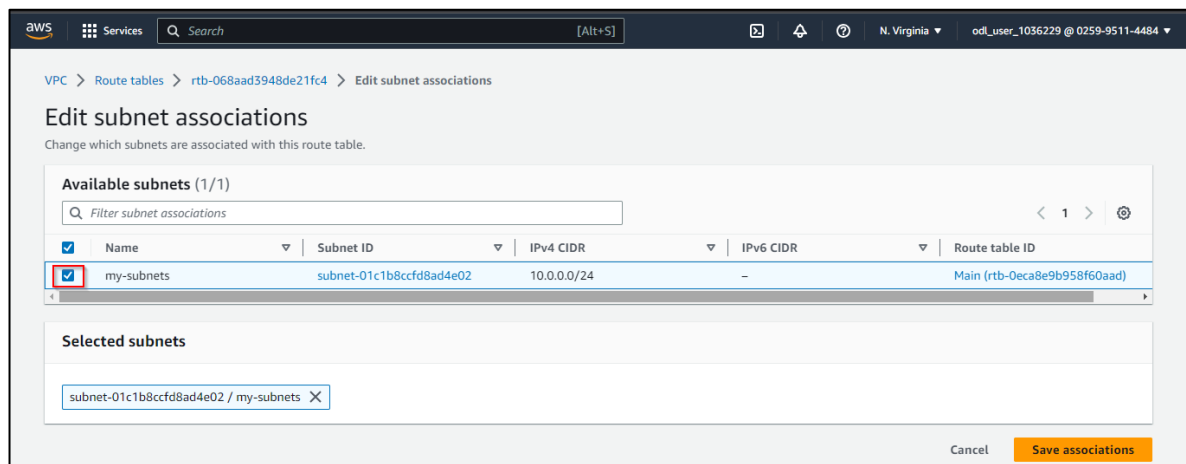
Details [Info](#)

Route table ID rtb-068aad3948de21fc4	Main No	Explicit subnet associations -
VPC vpc-09cbeeb7ef3c059b my-subnets	Owner ID 025995114484	

Actions

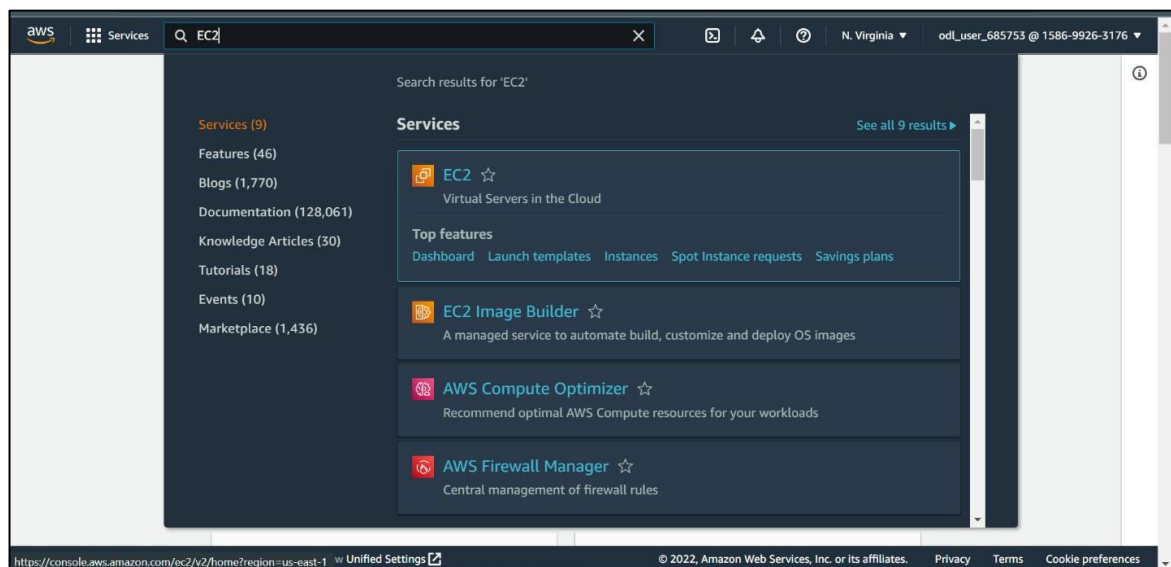
- Set main route table
- Edit subnet associations
- Edit edge associations
- Edit route propagation
- Edit routes
- Manage tags
- Delete
- Troubleshoot
- Trace network reachability

2.6 Select the subnet, and click on **Save associations**

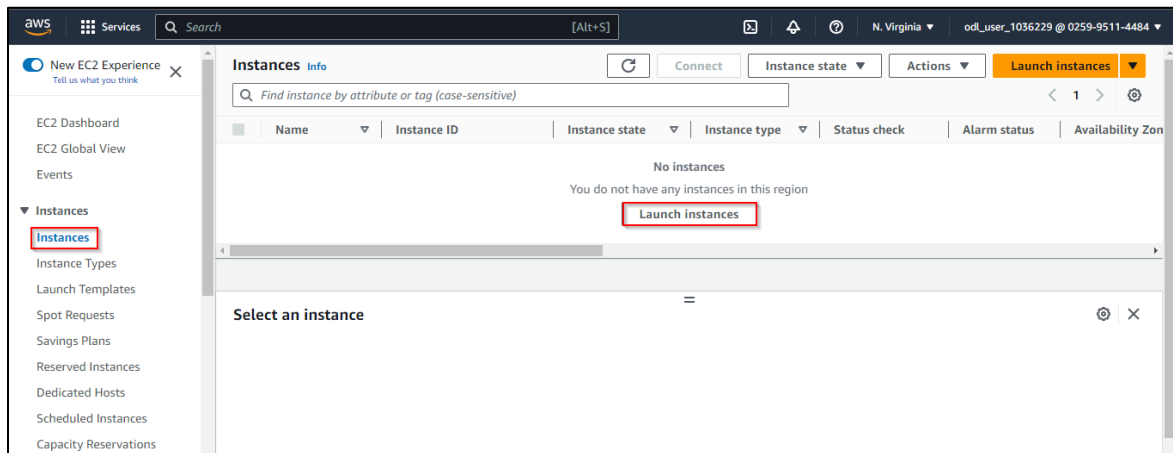


Step 3: Create an EC2 web server instance

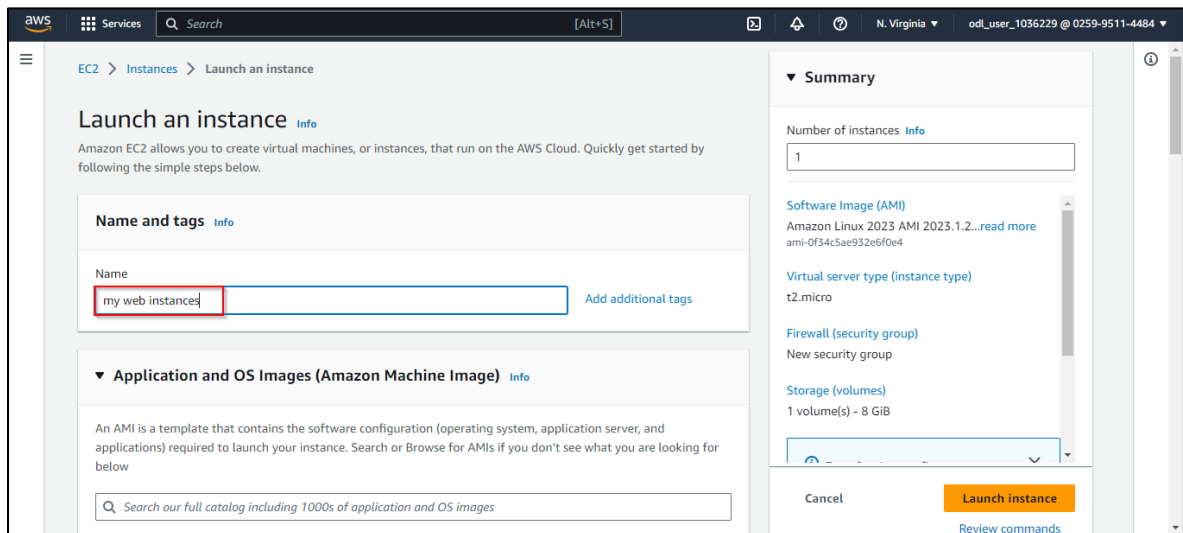
3.1 Navigate to **EC2** in the console navigation pane



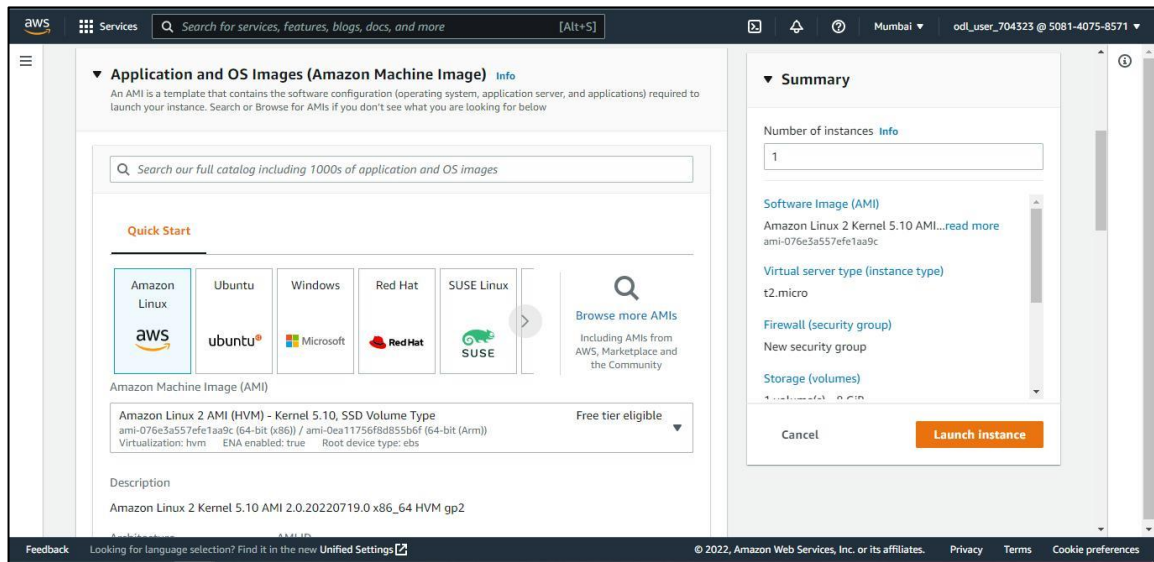
3.2 Click **Instances** and then **Launch instances**



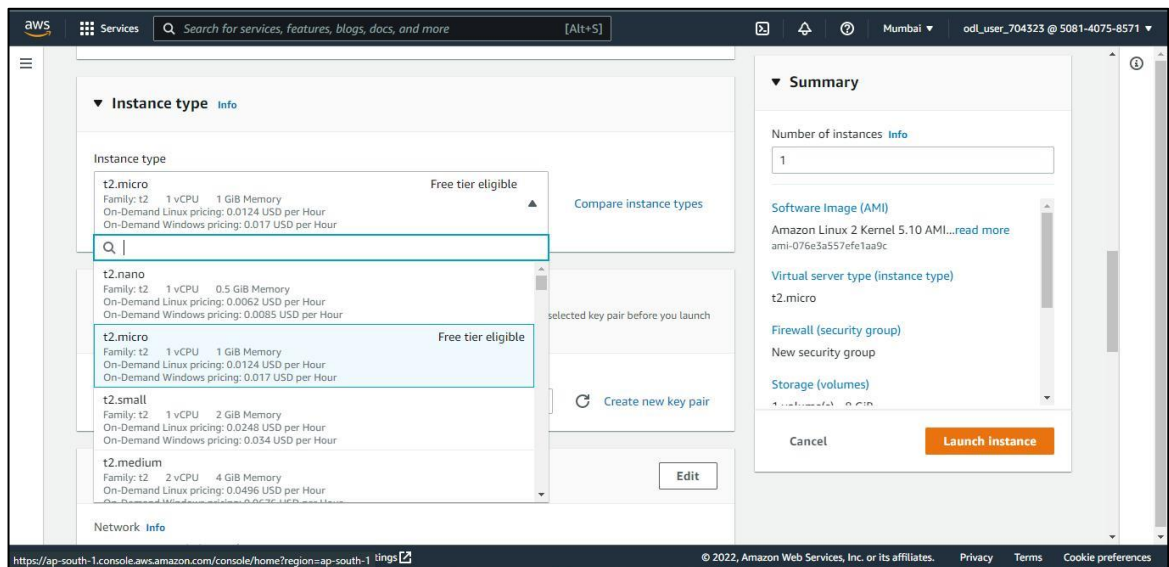
3.3 Provide the instance name, such as **my web instances**



3.4 Choose **Amazon Linux 2 AMI** as the OS image, and select **Kernel 5.10** and **SSD** in the Application and OS image section



3.5 Choose **t2.micro** as the instance type



3.6 Click **Create new key pair**, and use **my web instances** as the key pair name

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

my web instances

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

☒ **.pem**
For use with OpenSSH

☐ **.ppk**
For use with PuTTY

⚠ 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](#)

Cancel **Create key pair**

3.7 Select the previously created VPC and subnet, and enable **Auto-assign Public IP**

Key pair name - required
my web instances [Create new key pair](#)

Network settings [Info](#)

VPC - required [Info](#)
vpc-09cbeeb7ef3c059b (my-subnets) [Create new VPC](#)

Subnet [Info](#)
subnet-01c1b8ccfd8ad4e02 my-subnets [Create new subnet](#)

Auto-assign public IP [Info](#)
Enable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
☒ Create security group ☐ Select existing security group

Security group name - required

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-09538990a0c4fe9be

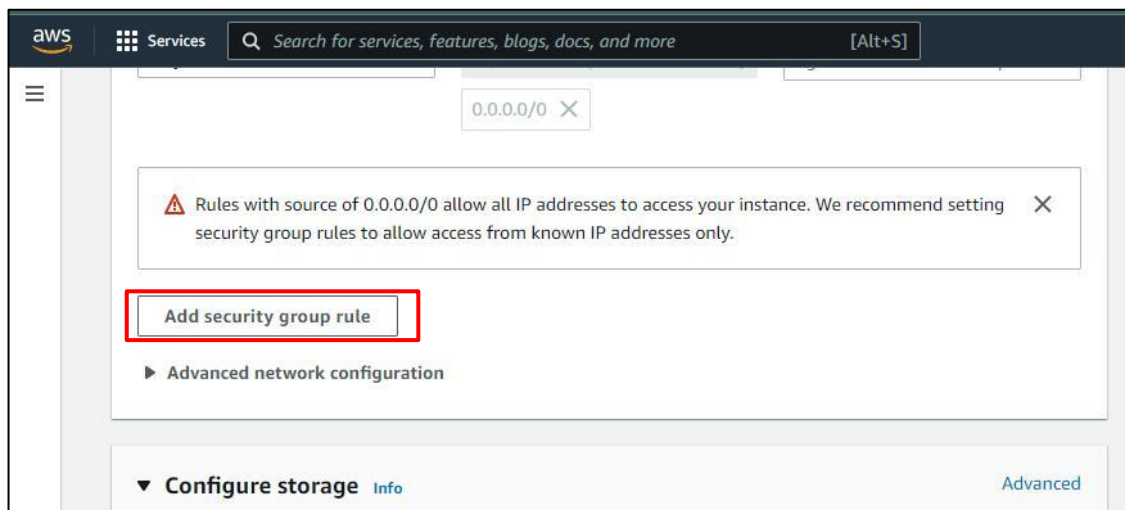
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

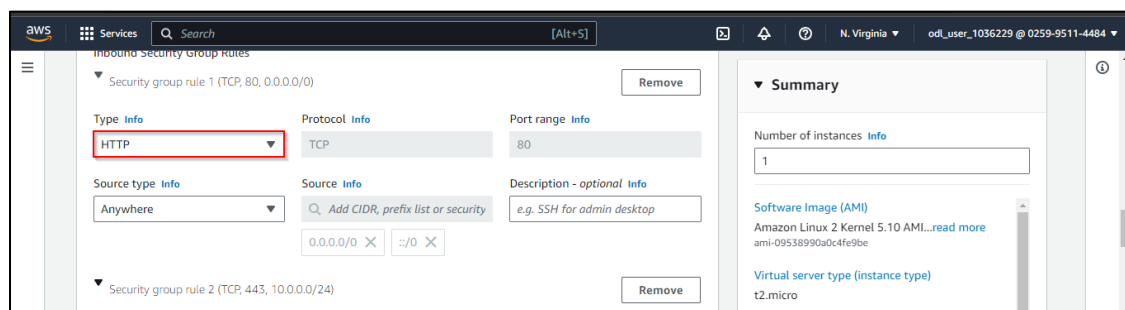
Storage (volumes)
1 volume(s) - 8 GiB

Cancel **Launch Instance** [Review commands](#)

3.8 Next, click on **Add Security group rule**



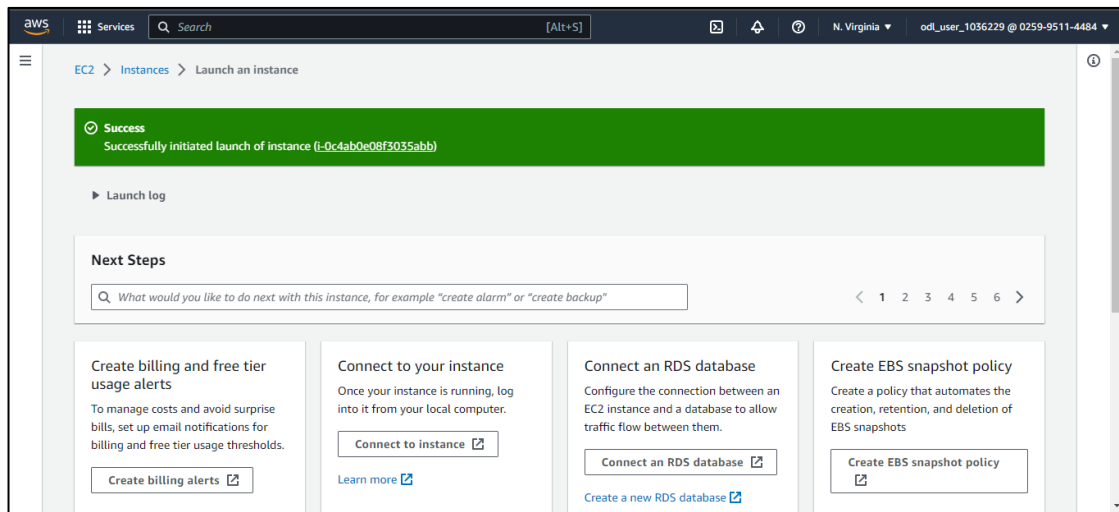
3.9 Click on **Inbound Security rule**, and add **HTTP** inbound rule



3.10 Click on **Advanced details**

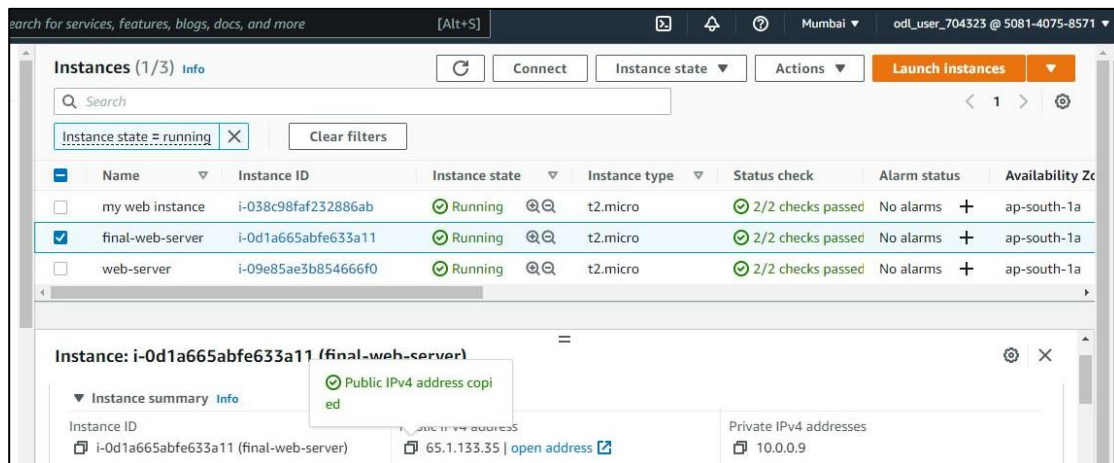
3.11 Add the following code in User data, and click on **Launch instance**:

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo '<h1>VM to DB Connect</h1>' > /var/www/html/index.html
```



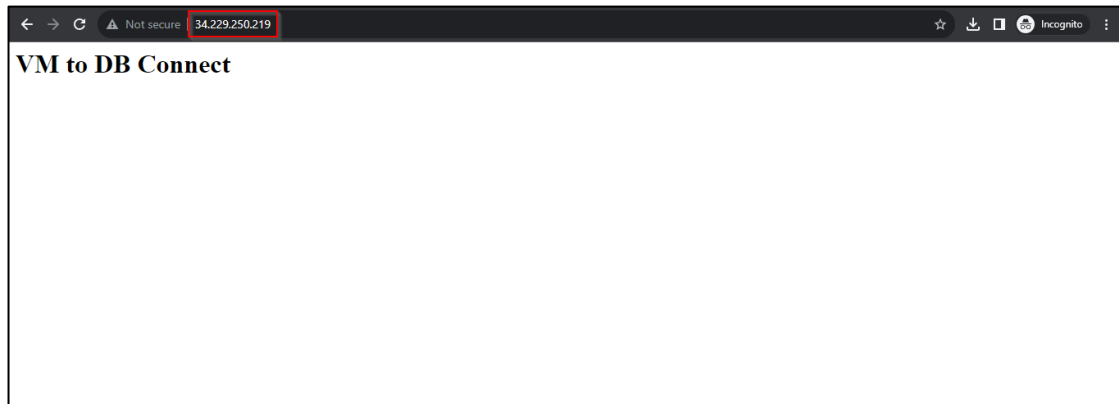
The EC2 web instance is created successfully.

3.12 Select the instance, and copy the public IPv4 address



3.13 Open a new browser tab, paste the public IPv4 address, and use port 80 to access the web server:

54.89.15.146:80



The web instance is now successfully created and accessible via its IPv4 address.

By following these steps, you will be able to successfully set up a web server on an EC2.