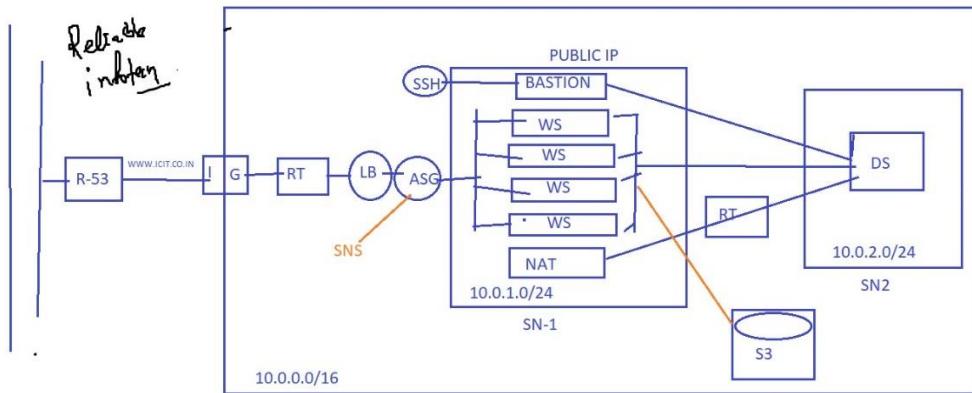


AWS Project using VPC

This is a simple AWS project for beginner. This Project consist of Virtual Private Cloud(VPC), Elastic Compute Cloud(EC2), Route 53, ASG, Loadbalancer.

Using this all services you will deploy a website on your public network simultaneously you will connect private and public servers.

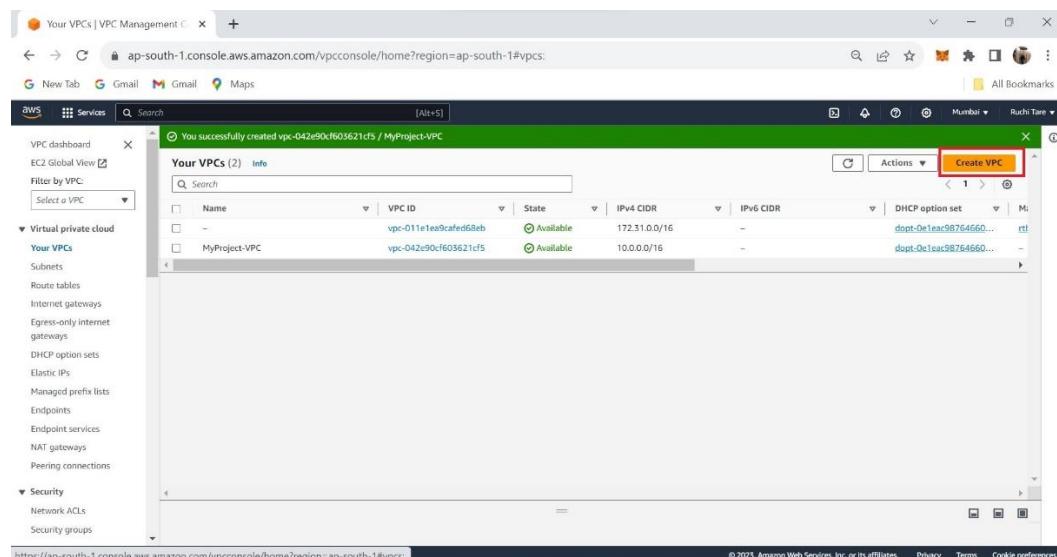


Step 1 : Connect to VPC

Virtual Private Cloud(VPC) : A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud. VPC customers can run code, store data, host websites, and do anything else they could do in an ordinary private cloud, but the private cloud is hosted remotely by a public cloud provider.

1. Login to Aws Console , after that search for VPC.

2. Create a VPC with CIDR 10.0.0.0/16.



VPC Console

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateVpc:createMode=vpcOnly

New Tab Gmail Maps

Mumbai Ruchi Tare

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 - optional
Creates a tag with a key of 'Name' and a value that you specify.
MyProject-VPC

IPv4 CIDR block: [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR: [Info](#)
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block: [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy: [Info](#)
Default

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC Console

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateVpc:createMode=vpcOnly

New Tab Gmail Maps

Mumbai Ruchi Tare

VPC settings

Resources to create: [Info](#)
Create only the VPC resource or the VPC and other networking resources.

IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR: [Info](#)
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block: [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

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: Remove tag

Add tag
You can add 49 more tags.

Create VPC

Cancel

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

You successfully created vpc-0fbe45a91ab841aa7 / MyProject-VPC

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
-	vpc-011e1ea9cafed68eb	Available	172.31.0.0/16	-	dopt-01ea98764660...
MyProject-VPC	vpc-0fbe45a91ab841aa7	Available	10.0.0.0/24	-	dopt-01ea98764660...

3. After creating vpc next step is to create 2 Subnet public and private subnet

4. public subnet CIDR 10.0.1.0/24

5. private subnet CIDR 10.0.2.0/24

First click on create a subnet.

Create subnet

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-069329704812b5403	Available	vpc-011e1ea9cafed68eb	172.31.52.0/20	-
-	subnet-0d07ed759feaae622	Available	vpc-011e1ea9cafed68eb	172.31.0.0/20	-
-	subnet-08bfaf0fe0ff0a8c48	Available	vpc-011e1ea9cafed68eb	172.31.16.0/20	-

Select the VPC that you have created.

The screenshot shows the AWS VPC Console interface. In the top left, there's a search bar with the URL "ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateSubnet". Below the search bar, the AWS logo and "Services" link are visible. The main content area is titled "VPC". Under "VPC ID", it says "Create subnets in this VPC." and lists two VPC IDs: "vpc-042e90cf603621cf5 (MyProject-VPC)" (selected) and "vpc-011e1ea9cafed68eb (default)". A "Search" input field is present. On the right, there are tabs for "CloudShell" and "Feedback", along with copyright information: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Enter Public Subnet name and CIDR and Availability Zone

The screenshot shows the AWS VPC Console interface for creating a subnet. The URL in the address bar is "ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateSubnet". The main content area is titled "Subnet 1 of 1". It has sections for "Subnet name" (with "Public Subnet" entered), "Availability Zone" (set to "Asia Pacific (Mumbai) / ap-south-1a"), and "IPv4 VPC CIDR block" (set to "10.0.0.0/16"). Below these, there's an "IPv4 subnet CIDR block" dropdown set to "10.0.1.0/24" with "256 IPs". There are "Tags - optional" fields for "Remove" and "Add new subnet". At the bottom right are "Cancel" and "Create subnet" buttons. The footer includes "CloudShell", "Feedback", and copyright information: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

Do the same for Private Subnet.

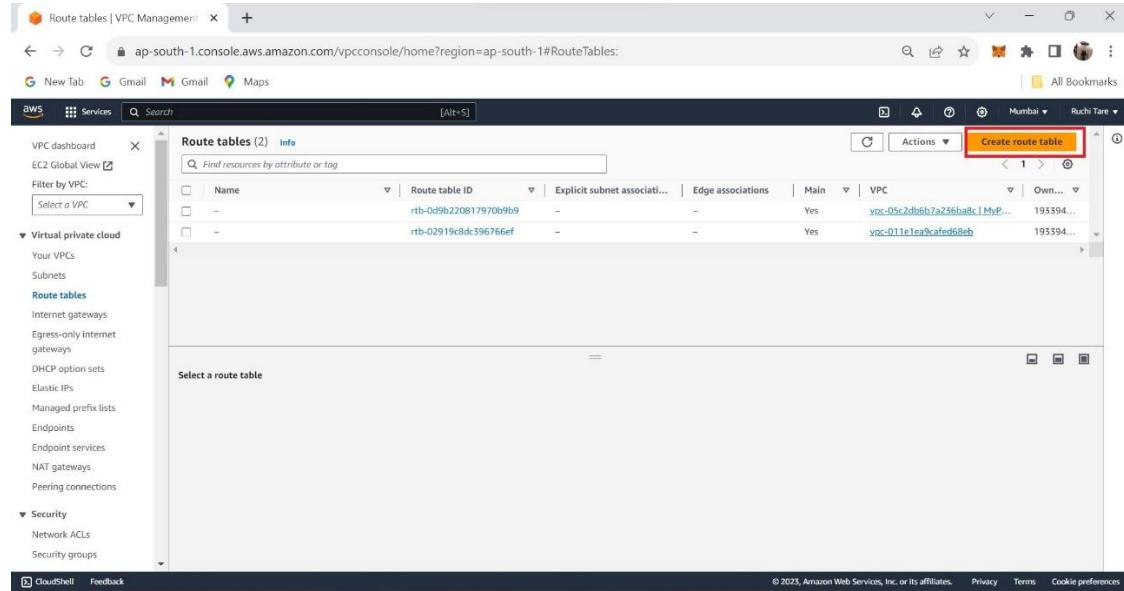
The screenshot shows the 'Create Subnet' wizard in the AWS VPC Console. The 'Subnet name' field is set to 'Private Subnet'. The 'Availability Zone' dropdown is set to 'Asia Pacific (Mumbai) / ap-south-1a'. The 'IPv4 VPC CIDR block' dropdown is set to '10.0.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '10.0.2.0/24'. The 'Create subnet' button is highlighted with a red box.

The screenshot shows the 'Subnets (5) Info' table in the AWS VPC Console. It lists five subnets: 'Public Subnet' and 'Private Subnet' under 'Virtual private cloud' and three unnamed subnets under 'EC2 Global View'. The table includes columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. The 'Create subnet' button is visible at the top right of the table area.

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-069329704812b3405	Available	vpc-011e1ea9cafed68eb	172.31.32.0/20	-
-	subnet-0d07ed759feaae622	Available	vpc-011e1ea9cafed68eb	172.31.0.0/20	-
-	subnet-08bfa0f600a8c48	Available	vpc-011e1ea9cafed68eb	172.31.16.0/20	-
Public Subnet	subnet-0d47cf7011f211ca	Available	vpc-042c90cf603621cf5 MyPr...	10.0.1.0/24	-
Private Subnet	subnet-04891b7a7ec2bf7be	Available	vpc-042c90cf603621cf5 MyPr...	10.0.2.0/24	-

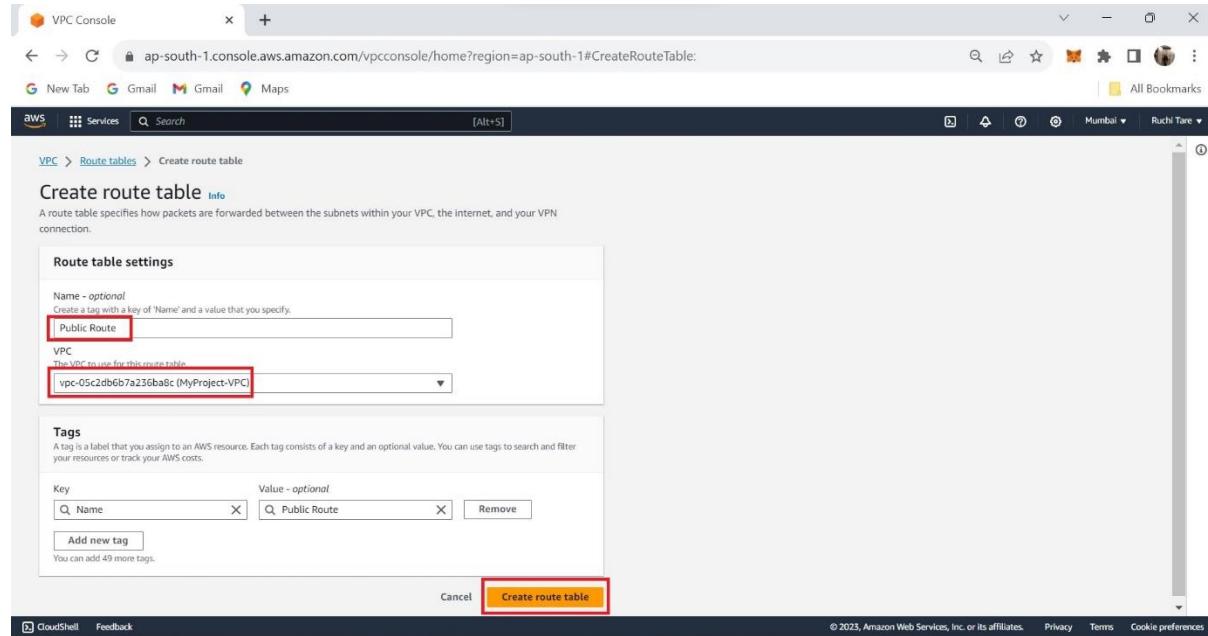
4. Create 2 Route table of which one is public and private

First click on create Route Table

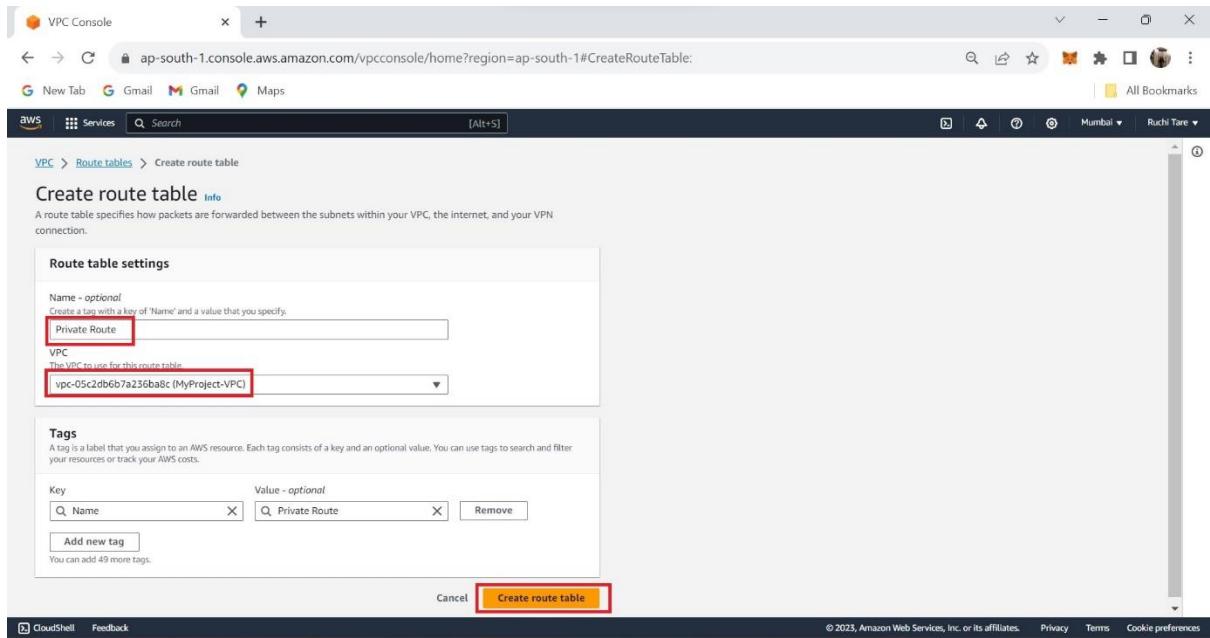


The screenshot shows the AWS VPC Management console with the 'Route tables' section selected. The left sidebar includes options like 'VPC dashboard', 'Virtual private cloud', 'Security', and 'CloudShell'. The main area displays a table of existing route tables with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Owner. A prominent red box highlights the 'Create route table' button at the top right of the table area.

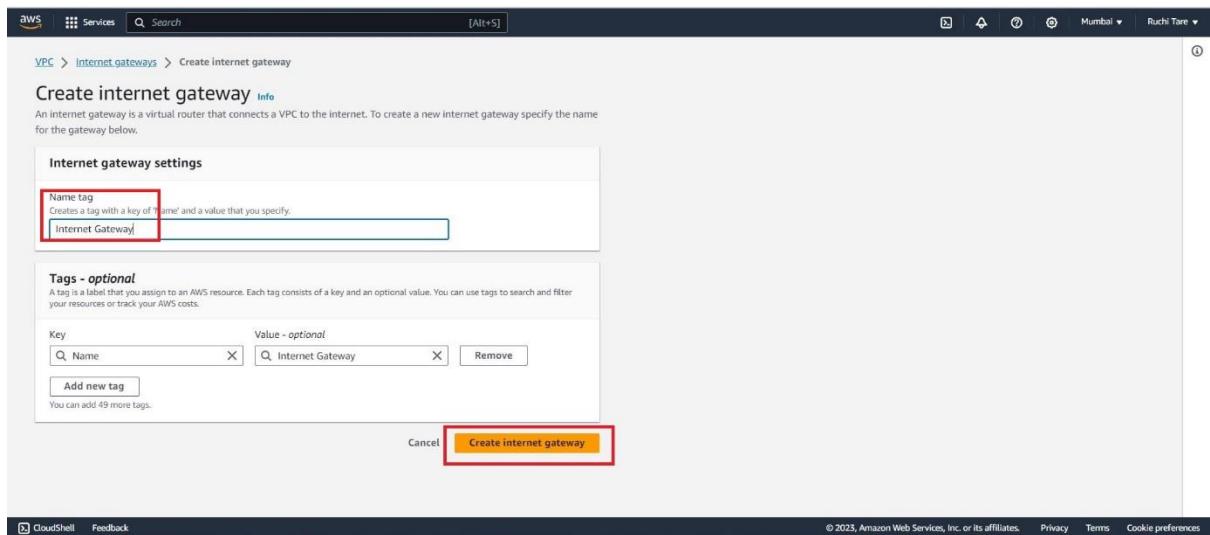
Create a public route table , add the created vpc and the name for route table do the same for private route table



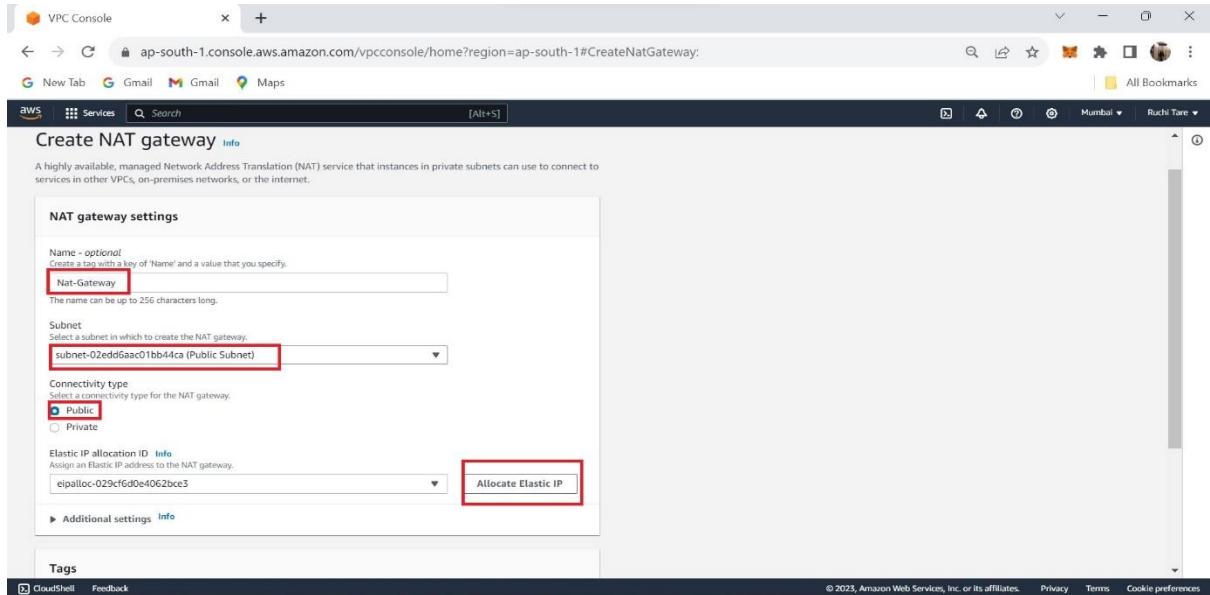
The screenshot shows the 'Create route table' wizard in the AWS VPC Console. The first step, 'Route table settings', is displayed. It requires a 'Name' (set to 'Public Route') and a 'VPC' (set to 'vpc-05c2db6b7a236ba8c (MyProject-VPC)'). Below these, there's a 'Tags' section where a single tag ('Name: Public Route') is added. At the bottom of the form is a 'Create route table' button, which is also highlighted with a red box.



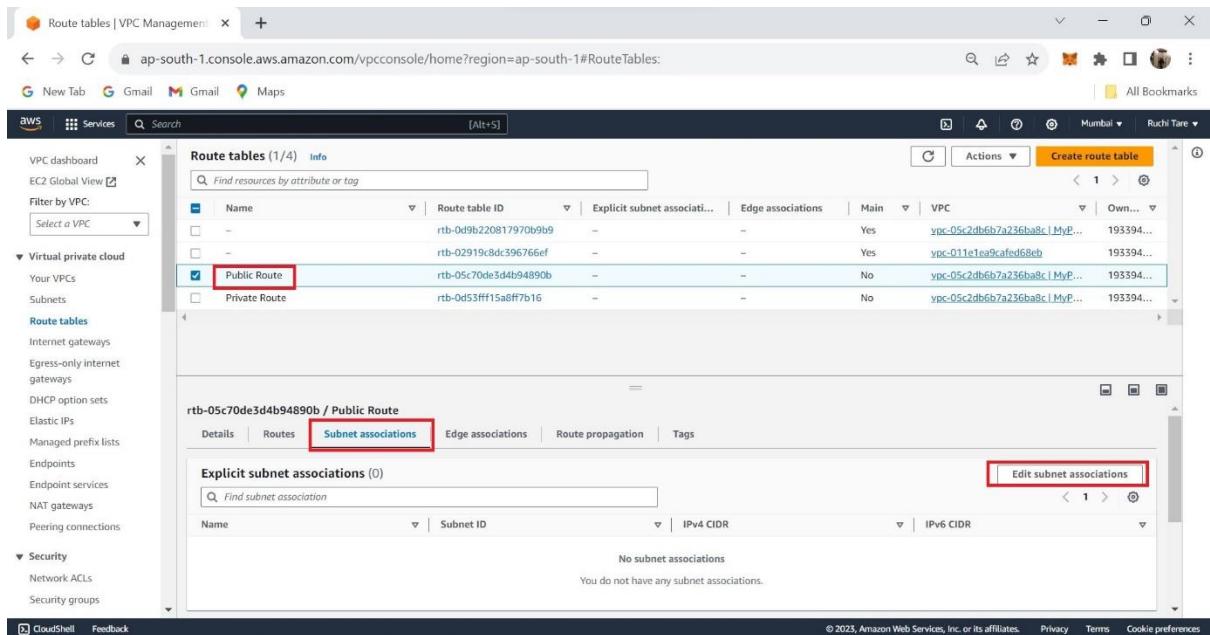
5. Create a Internet Gateway for connecting Internet to Subnets and also attach that with vpc



6. Create a NAT gateway for connecting private subnet through internet, we will place Nat gateway in the public subnet also we will attach elastic IP to it.



7. Next step is to Associate the subnets to the Route Table , Public route to public subnet and private route to the private subnet.



VPC Console

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-05c70d...

New Tab Gmail Maps All Bookmarks Mumbai Ruchi Tare

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
Private Subnet	subnet-0d55964ff5c7435b	10.0.2.0/24	-	Main (rtb-0d9b220817970b9b9)	
<input checked="" type="checkbox"/> Public Subnet	subnet-02edd6aac01bb44ca	10.0.1.0/24	-	Main (rtb-0d9b220817970b9b9)	

Selected subnets

subnet-02edd6aac01bb44ca / Public Subnet X
--

Cancel **Save associations**

Route tables | VPC Management

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#RouteTables:

New Tab Gmail Maps All Bookmarks Mumbai Ruchi Tare

Route tables (1/4) Info

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC	Own...
-	rtb-0d9b220817970b9b9	-	-	Yes	vpc-05c2db6b7a236ba8c MyP...	193394...
-	rtb-02919c8dc596766ef	-	-	Yes	vpc-011e1ea9cafed68eb	193394...
<input checked="" type="checkbox"/> Public Route	rtb-05c70de3d4b94890b	-	-	No	vpc-05c2db6b7a236ba8c MyP...	193394...
Private Route	rtb-0d53fff15a8ff7b16	-	-	No	vpc-05c2db6b7a236ba8c MyP...	193394...

rtb-05c70de3d4b94890b / Public Route

Details Routes **Subnet associations** Edge associations Route propagation Tags

Explicit subnet associations (0)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
------	-----------	-----------	-----------

No subnet associations
You do not have any subnet associations.

Edit subnet associations

7. Route the Internet Gateway with public route for providing internet connectivity.

You have successfully updated subnet associations for rtb-0d53fff15a8ff7b16 / Private Route.

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC	Own...
<input checked="" type="checkbox"/> Public Route	rtb-05c70de3d4b94890b	subnet-02eddf6aac01bb4...	-	Yes	vpc-05c2db6b7a236ba8c MyP...	193394...
<input type="checkbox"/> Private Route	rtb-0d53fff15a8ff7b16	subnet-0d53fff15a8ff7b16	-	No	vpc-05c2db6b7a236ba8c MyP...	193394...

rtb-05c70de3d4b94890b / Public Route

Details **Routes** Subnet associations Edge associations Route propagation Tags

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Click on public subnet → route → Edit route → Add Route → select 0.0.0.0/0 → Internet gateway → save changes.

Y VPC Console

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1>EditRoutes:RouteTableId=rtb-05c70de3d4b94890b

Y New Tab G Gmail M Gmail Maps

aws Services Q Search [Alt+S]

VPC > Route tables > rtb-05c70de3d4b94890b > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-0c75b4a58d74c90a1		

Add route Cancel Preview **Save changes**

DO the same process with the Private Subnet but instead of Internet gateway you will have to add NAT Gateway , you will provide internet to the private subnet through NAT Gateway.

Route tables (1/4) Info

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Own...
-	rtb-0d9b220817970b9b	-	-	Yes	vpc-05c2db6b7a236ba8c MyP...	193394...
-	rtb-02919c8dc39676ef	-	-	Yes	vpc-011e1ea9cafed68eb	193394...
Public Route	rtb-05c70de3d4b94890b	subnet-02edd6aac01bb4...	-	No	vpc-05c2db6b7a236ba8c MyP...	193394...
Private Route	rtb-0d53ff15a8ff7b16	subnet-0d55964ffcc5c743...	-	No	vpc-05c2db6b7a236ba8c MyP...	193394...

rtb-0d53ff15a8ff7b16 / Private Route

Routes (1)

Destination	Target	Status	Propagated
10.0.0.16	local	Active	No

YPC > Route tables > rtb-0d53ff15a8ff7b16 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.16	local	Active	No
0.0.0.0/0	NAT Gateway nat-0015f57ba2664dd93	-	No

Add route

Cancel Preview **Save changes**

Step 2: Now create EC2 environment:

1. Go to EC2 and create two instance
2. 1st instance is Bastin Server which is a Public server and this server is running on the Linux server for that for authentication we required Key , so download a .ppk file.

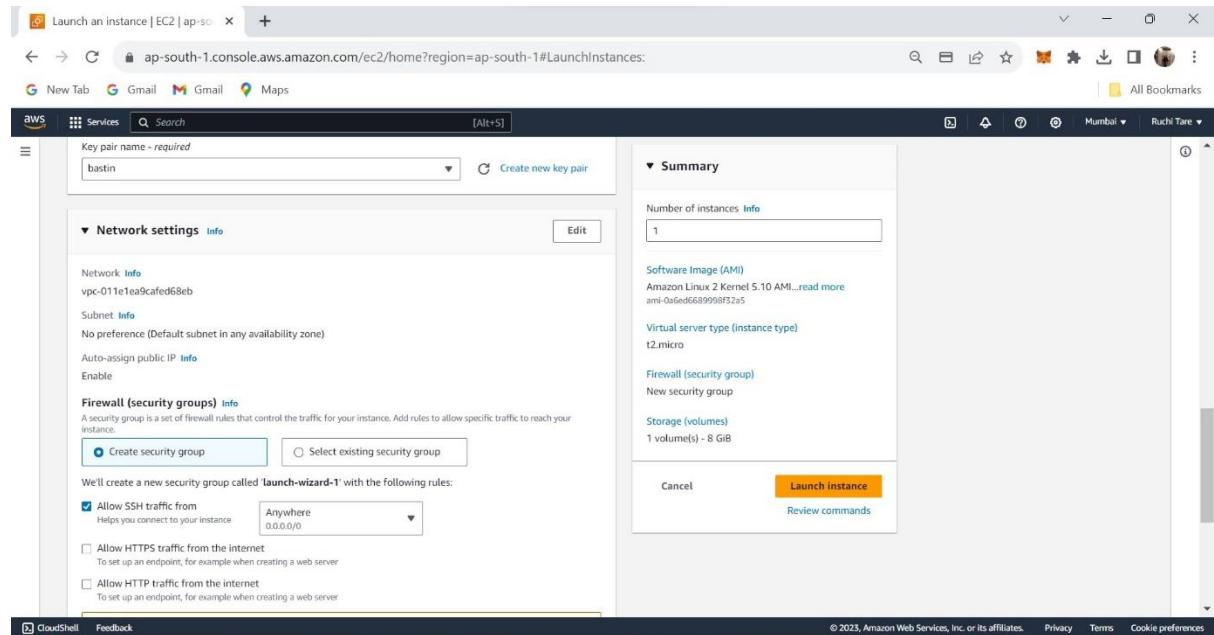
Write the name of the server you and select the operating system as linux.

The screenshot shows the 'Launch instance | EC2' wizard on the AWS console. The current step is 'Name and tags'. A single instance is selected. The 'Name' field contains 'Bastio Server'. The 'Software Image (AMI)' dropdown is set to 'Amazon Linux 2023 AMI 2023.2.2...'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Storage (volumes)' section shows '1 volume(s) - 8 GB'. The 'Summary' panel on the right shows 'Number of instances: 1'. The 'Launch instance' button is highlighted in orange at the bottom right.

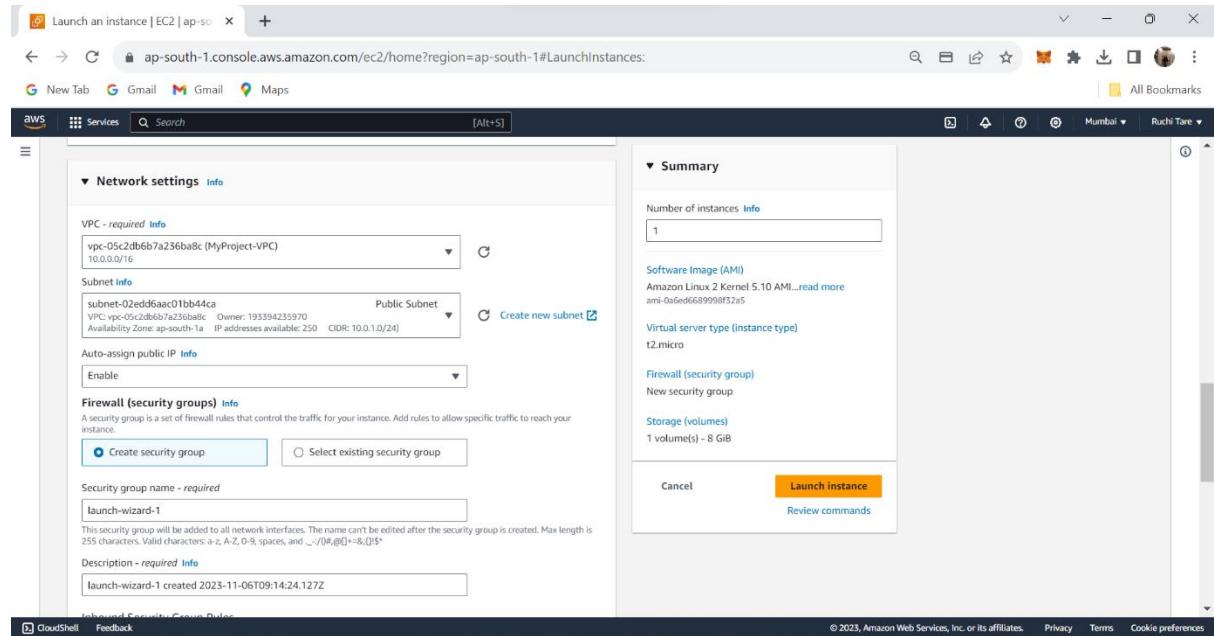
Select the Amazon Linux 2 AMI as your operating system. Select t2micro instance type.

The screenshot shows the 'Launch instance | EC2' wizard on the AWS console. The current step is 'Amazon Machine Image (AMI)'. The 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' is selected. The 'Free tier eligible' status is shown. The 'Description' section details 'Amazon Linux 2 Kernel 5.10 AMI 2.0.20231101.0 x86_64 HVM gp2'. The 'Architecture' is '64-bit (x86)' and the 'AMI ID' is 'ami-0a6ed6689998f32a5'. The 'Verified provider' badge is present. The 'Instance type' dropdown is set to 't2.micro'. The 'Summary' panel on the right shows 'Number of instances: 1'. The 'Launch instance' button is highlighted in orange at the bottom right.

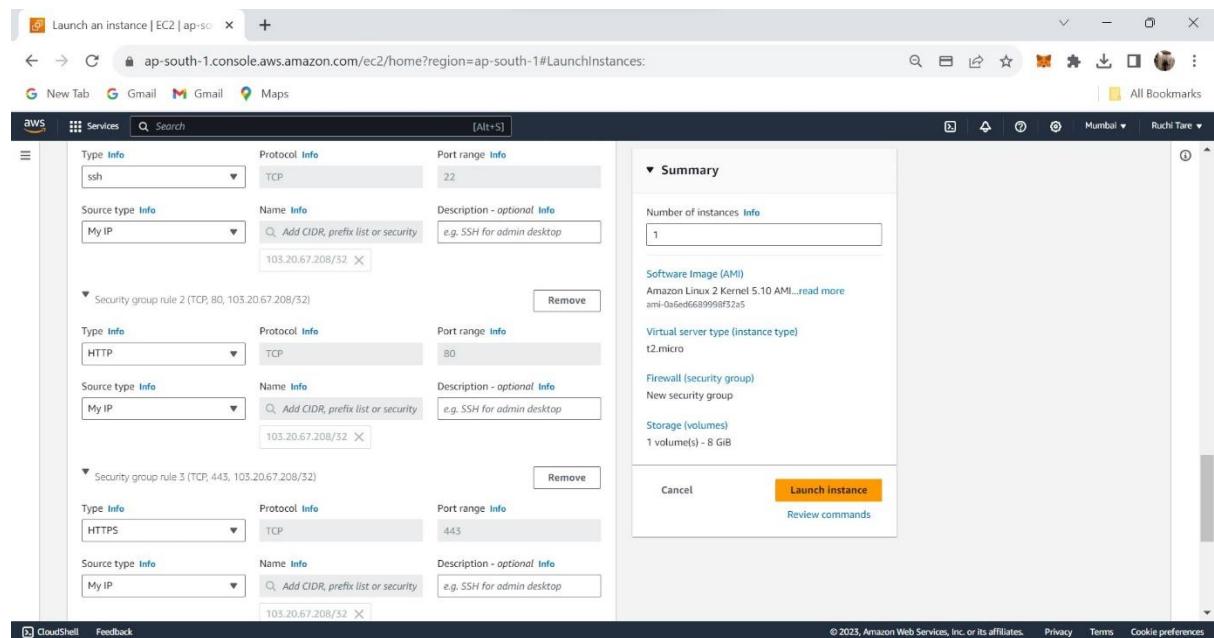
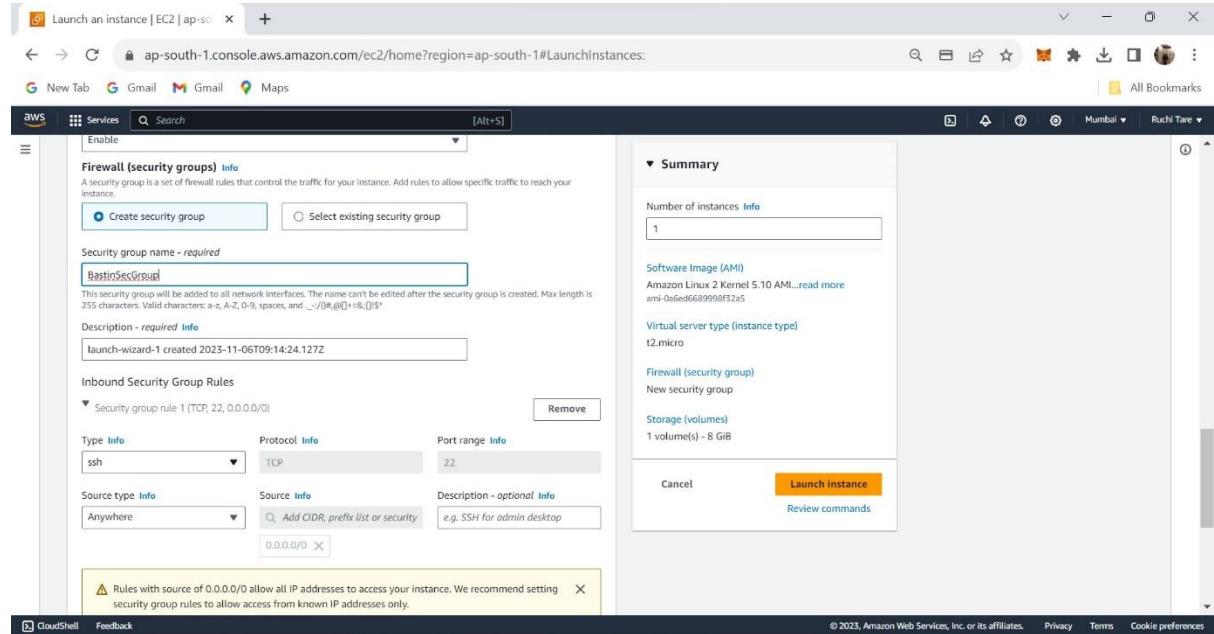
Select the key pair you have created and under network setting select the created VPC with the public subnet, enable the IP addresses.



Create a new security group , give name to that group,

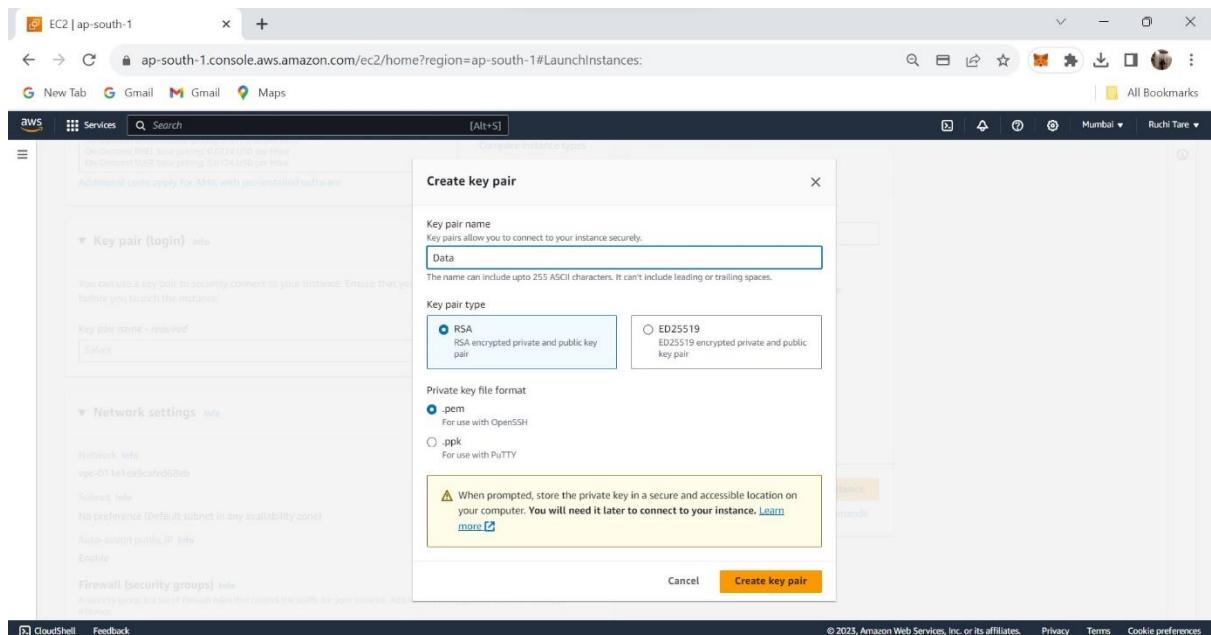
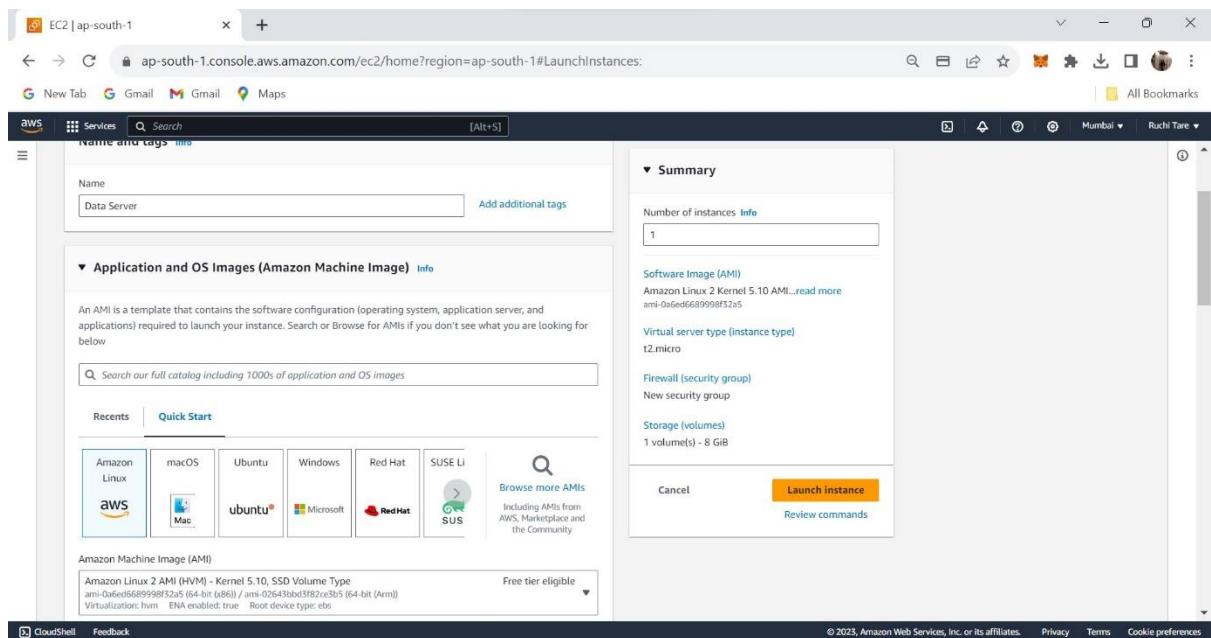


Add the security rule of SSH with MyIP followed by MySql, HTTP and HTTPS with the security type as MyIP.



After the above steps are done recheck it and the click on the Launch the Instance button.

3. 2nd instance is Data instance with the Data.pem key followed by its name and new Security group with security rule as SSH and MySql and Http and Https.



EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

New Tab Gmail Maps Mumbai Ruchi Tare All Bookmarks

aws Services Search [Alt+S]

Network settings

VPC - required Info
vpc-05c2db6b7a236ba8c (MyProject-VPC)
10.0.0.0/16

Subnet Info
subnet-0d55964ff5c7435b Private Subnet
VPC vpc-05c2db6b7a236ba8c Owner: 193394235970 Availability Zone: ap-south-1a IP addresses available: 251 CIDR: 10.0.2.0/24

Create new subnet

Auto-assign public IP Info
Disable

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 launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-/()#@+=&{|}^<

Description - required Info
launch-wizard-1 created 2023-11-06T09:18:26.098Z

Inbound Security Group Rules

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0a6ed6689998f32a5

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

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

New Tab Gmail Maps Mumbai Ruchi Tare All Bookmarks

aws Services Search [Alt+S]

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 DataSecGroup

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-/()#@+=&{|}^<

Description - required Info
launch-wizard-1 created 2023-11-06T09:18:26.098Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 103.20.67.208/32)
Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info Name Info Description - optional Info
My IP e.g. SSH for admin desktop

Add CIDR, prefix list or security
My IP 103.20.67.208/32 X

Security group rule 2 (TCP, 0)
Remove

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-0a6ed6689998f32a5

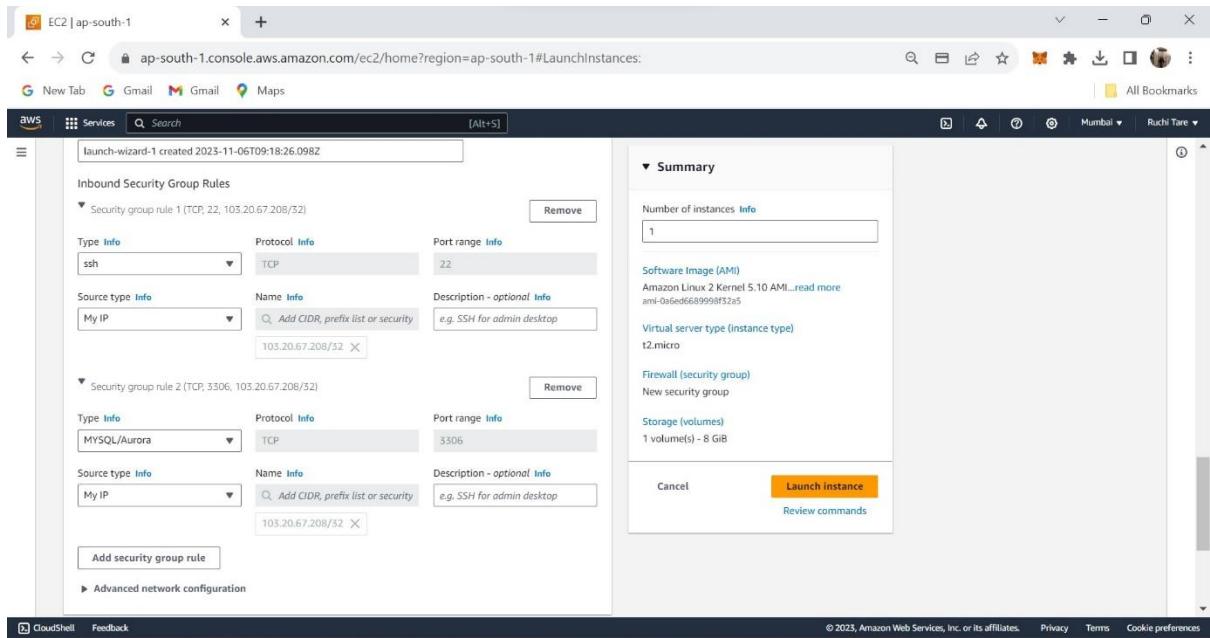
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Cancel Launch instance Review commands

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



4. Connect the Bastin Server.
5. To connect bastin server you will have to first download putty app
<https://www.putty.org/>
6. Open the putty server paste your bastin Public IP in the addresses of your putty then go to SSH→Auth→Credential then browser your .ppk key of your bastin server and select that key your bastin terminal will get open

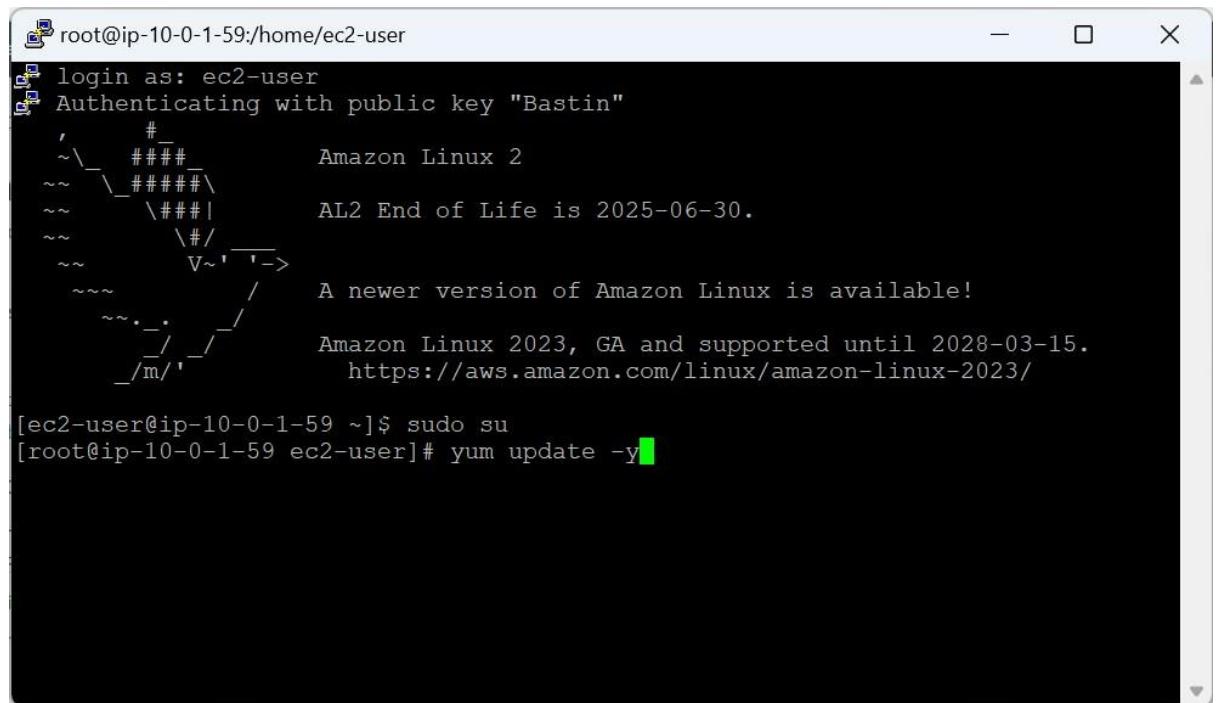
```
ec2-user@ip-10-0-1-59:~$ login as: ec2-user
[ec2-user@ip-10-0-1-59 ~]$ Authenticating with public key "Bastin"
[ec2-user@ip-10-0-1-59 ~]$ ,#
[ec2-user@ip-10-0-1-59 ~]$ ~\_\#\#\# Amazon Linux 2
[ec2-user@ip-10-0-1-59 ~]$ ~~\_\#\#\#\#\backslash AL2 End of Life is 2025-06-30.
[ec2-user@ip-10-0-1-59 ~]$ ~~\#\#| V~'-->
[ec2-user@ip-10-0-1-59 ~]$ ~~\#\# / A newer version of Amazon Linux is available!
[ec2-user@ip-10-0-1-59 ~]$ ~~\#\# / Amazon Linux 2023, GA and supported until 2028-03-15.
[ec2-user@ip-10-0-1-59 ~]$ https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-0-1-59 ~]$
```

Login to Bastin server through the command **ec2-user**.

Then write the command

Sudo su

Yum update -y in your bastin terminal

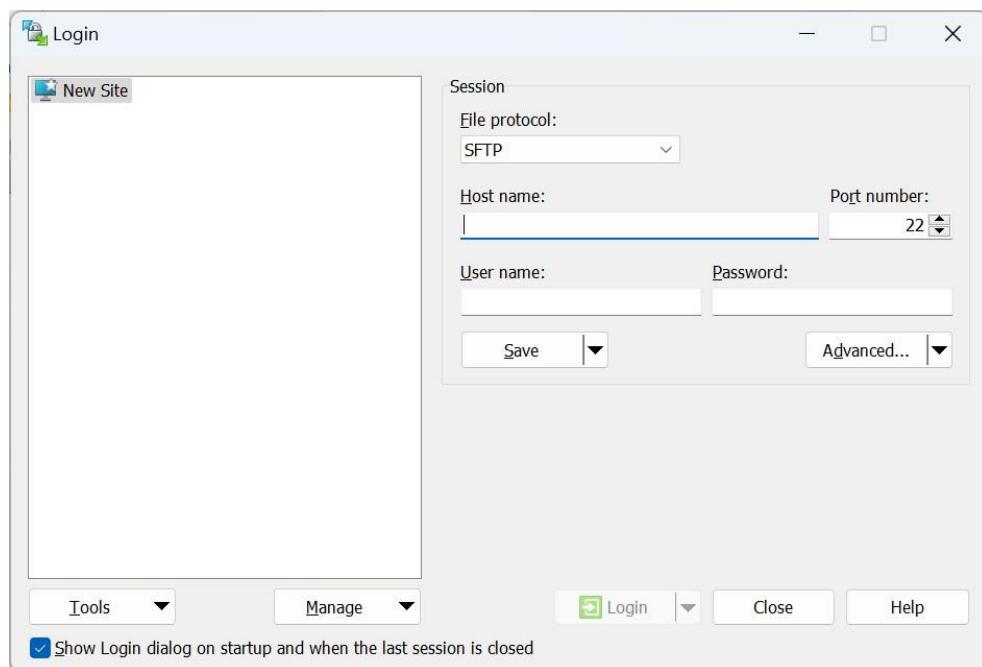


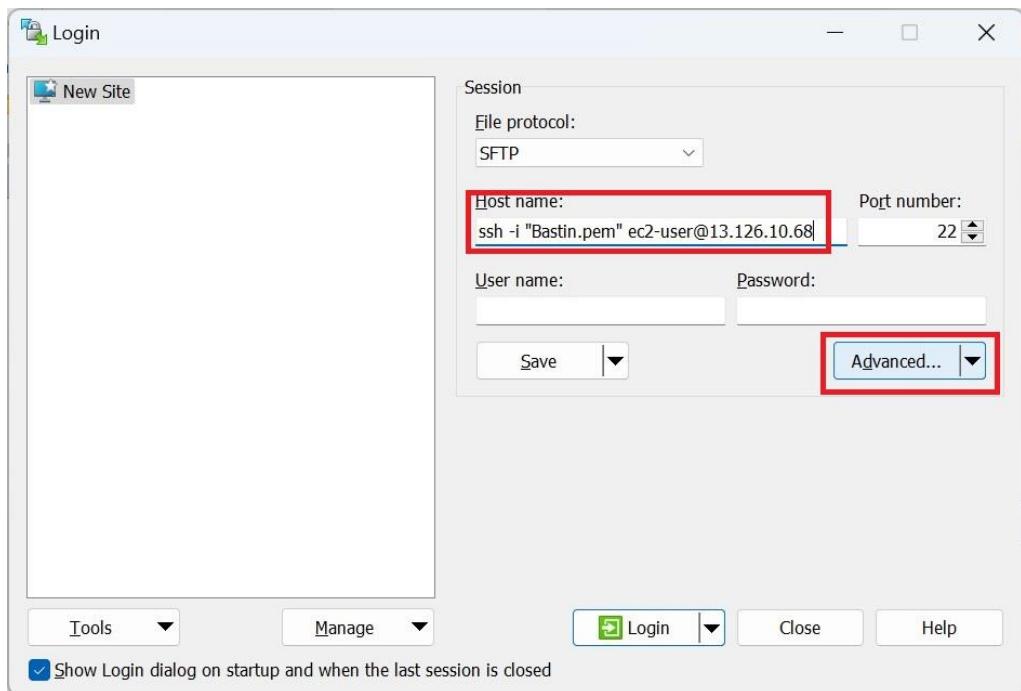
```
root@ip-10-0-1-59:/home/ec2-user
[ec2-user@ip-10-0-1-59 ~]$ sudo su
[root@ip-10-0-1-59 ec2-user]# yum update -y
```

7. Download the Winscp in your computer <https://winscp.net/eng/download.php>.

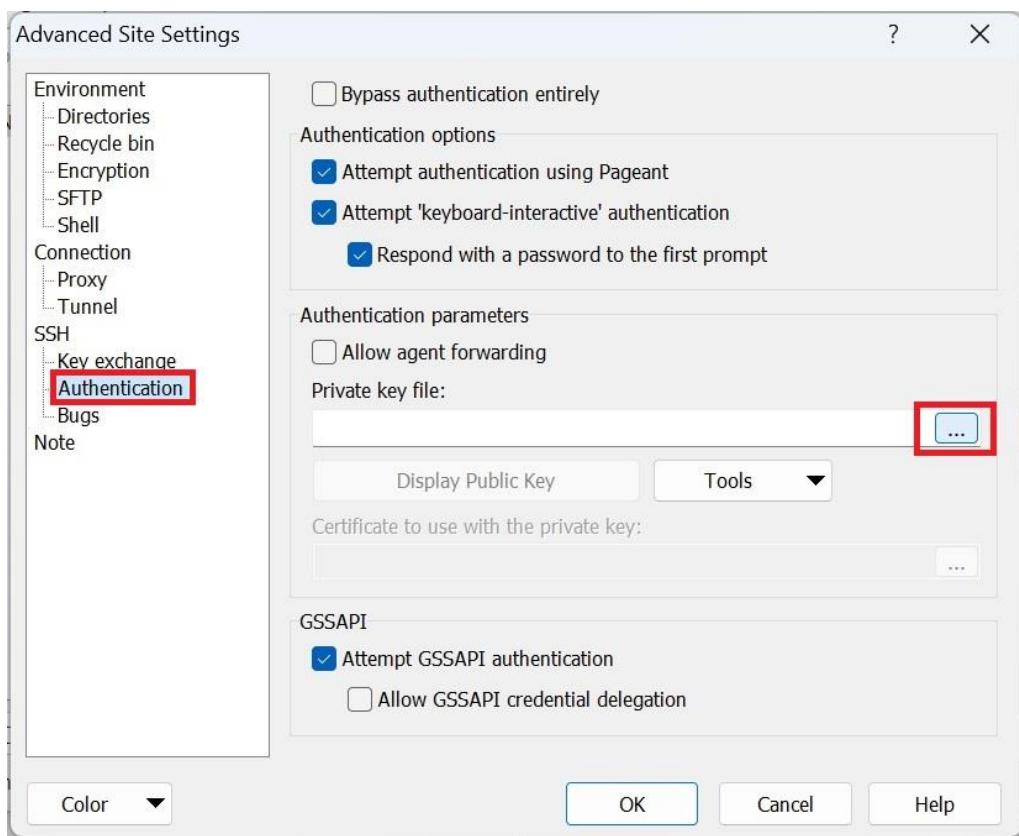
Once the Winscp is downloaded and installed properly login into it with bastin hostname.

To copy the hostname go to bastin instance → select the bastin instance → connect → ssh → example url.

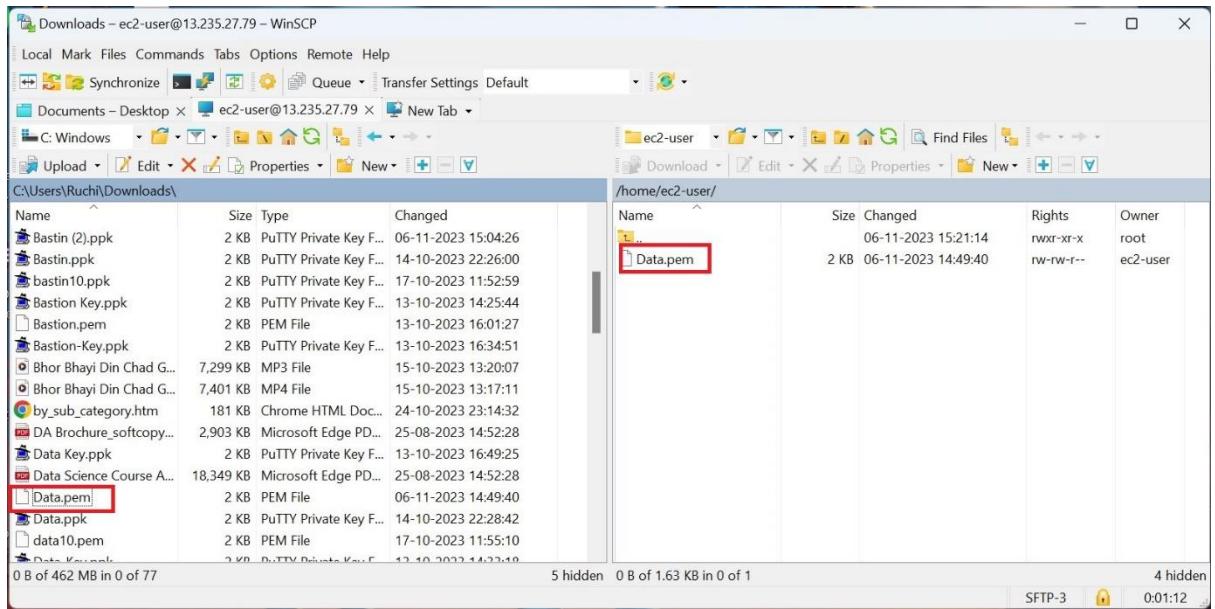




After copying the hostname then paste the url and then click on advance.



Click on Authentication on SSH and the browser for your bastin .ppk key select that key and click on ok.



After that on the left side check for your data.pem key and drag and paste it on the right side of your winscp.

- Come back to your console and in EC2 service select the Data instance copy the private key of DATA and paste it in the security group of BASTIN Security group MYSQL.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address
Data Server	i-09b72dc685285c2e6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-	-
Bastin Server	i-0f16d3fae044f108d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-	13.235.27.79

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ad
Data Server	i-09b72dc685285c2e6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-	-
Bastin Server	i-0f16d3fae044f108d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	13.235.27.79	

Instance: i-0f16d3fae044f108d (Bastin Server)

Details Security Networking Storage Status checks Monitoring Tags

Security details

Inbound rules

Inbound rules Info

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional info
sgr-099f0f648c31cae59	HTTPS	TCP	443	Custom	103.20.67.208/32
sgr-08cf9f0c29506250e	HTTP	TCP	80	Custom	103.20.67.208/32
sgr-04884193cc298d74	MySQL/Aurora	TCP	3306	Custom	10.0.2.131/32 10.0.2.131/32
sgr-0706db3b74595eae	SSH	TCP	22	Custom	103.20.67.208/32

Add rule

9. Copy the bastin private key and paste it to Data Security Group SSH.

Instances (1/4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ad
Data Server	i-09b72dc685285c2e6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-	-
Bastin Server	i-0f16d3fae044f108d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-	13.235.27.79
Bastin Server	i-0995452fe1fd1623	Terminated	t2.micro	-	No alarms	ap-south-1a	-	-
Bastin Server	i-0f18ad801e1fc7880	Terminated	t2.micro	-	No alarms	ap-south-1a	-	-

Instance: i-0f16d3fae044f108d (Bastin Server)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Private IPv4 address copied: 10.0.1.147

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group ID	Type Info	Protocol Info	Port range info	Source info	Description - optional info
sgr-037f9bcdf3cd1c5b1	SSH	TCP	22	Custom Q. 10.0.1.147/32 10.0.1.147/32 X	
sgr-041655c87b1f60610	MySQL/Aurora	TCP	3306	Custom Q. 103.20.6.7/32/32 X	

Add rule

Cancel | Preview changes | Save rules

10. Select the Data instance → connect and copy the entire example url and paste it in Bastin CMD.

Connect to instance

Connect to your instance i-09b72dc685285c2e6 (Data Server) using any of these options

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID
i-09b72dc685285c2e6 (Data Server)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is Data.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 Data.pem
4. Connect to your instance using its Private IP:
10.0.2.131

Example:
 ssh -i 'Data.pem' ec2-user@10.0.2.131

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

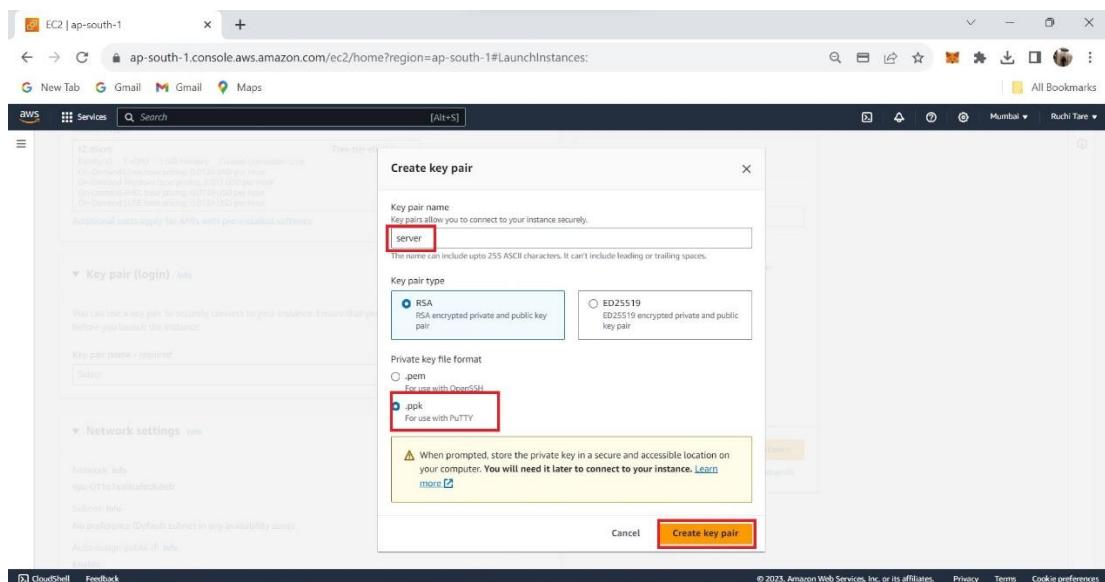
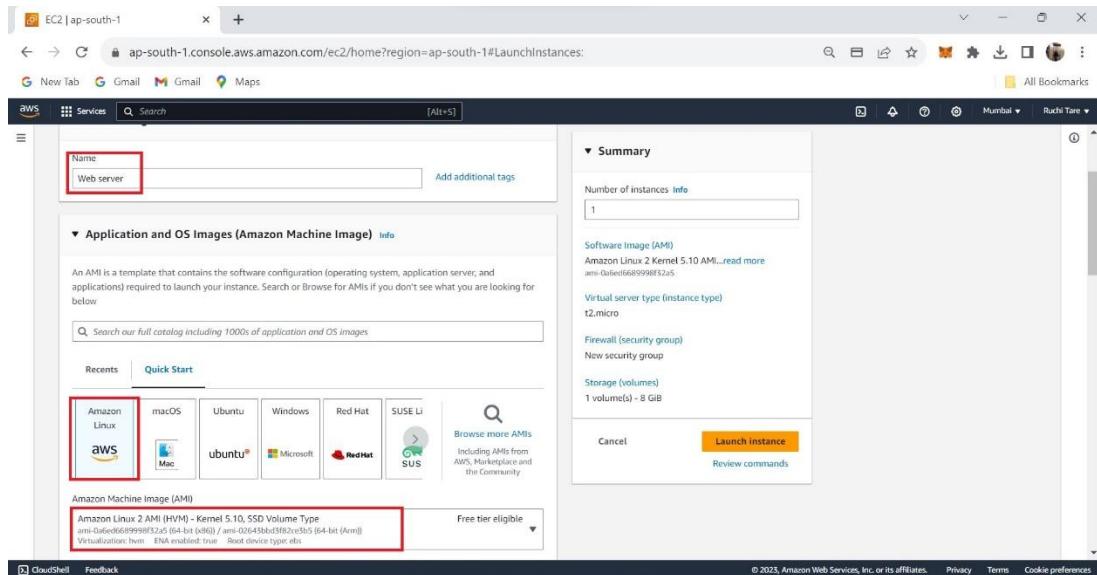
Cancel

```
root@ip-10-0-1-147:/home/ec2-user
[ec2-user@ip-10-0-1-147 ~]$ ssh -i "Data.pem" ec2-user@10.0.2.131
```

root@ip-10-0-1-147:~\$ sudo su
[root@ip-10-0-1-147 ~]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
| 3.6 kB 00:00
No packages marked for update
[root@ip-10-0-1-147 ~]#

Step 3:

1. Launch a Web Server name “Instance” with the Linux OS along with key pair and Security group of HTTP&HTTPS.



EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

New Tab Gmail Maps All Bookmarks Mumbai Ruchi Tare

Services Search [Alt+S]

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... read more
am-0a6ed6899fbf32a5

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Network settings

VPC Required info: VPC: vpc-05c2db6b7a236ba8c (MyProject-VPC) 10.0.0.1/16

Subnet info: Subnet: subnet-02dd6aae01bb44ca Public Subnet: VPC: vpc-05c2db6b7a236ba8c Owner: 193394235970 Availability Zone: ap-south-1a IP addresses available: 249 CIDR: 10.0.1.0/24

Auto-assign public IP: 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 ServerSecGrp

Description - required: launch-wizard-1 created 2023-11-06T10:18:40.101Z

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

New Tab Gmail Maps All Bookmarks Mumbai Ruchi Tare

Services Search [Alt+S]

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... read more
am-0a6ed6899fbf32a5

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Type Info: ssh

Protocol Info: TCP

Port range Info: 22

Source type Info: My IP

Name Info: 103.20.67.208/32

Description - optional Info: e.g. SSH for admin desktop

Type Info: HTTP

Protocol Info: TCP

Port range Info: 80

Source type Info: My IP

Name Info: 103.20.67.208/32

Description - optional Info: e.g. SSH for admin desktop

Type Info: HTTPS

Protocol Info: TCP

Port range Info: 443

Source type Info: My IP

Name Info: 103.20.67.208/32

Description - optional Info: e.g. SSH for admin desktop

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Instances | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:instanceState=running;v=3;\$case=true%5C...

New Tab Gmail Maps All Bookmarks Mumbai Ruchi Tare

Services Search [Alt+S]

Instances (3) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 add...
Data Server	i-09b72dc685285c2e6	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	-	-
Bastin Server	i-0f16d3fae044f108d	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	-	13.235.27.79
Web server	i-0962850dc415fb90a	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	-	13.233.93.165

EC2 Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Recycle Manager

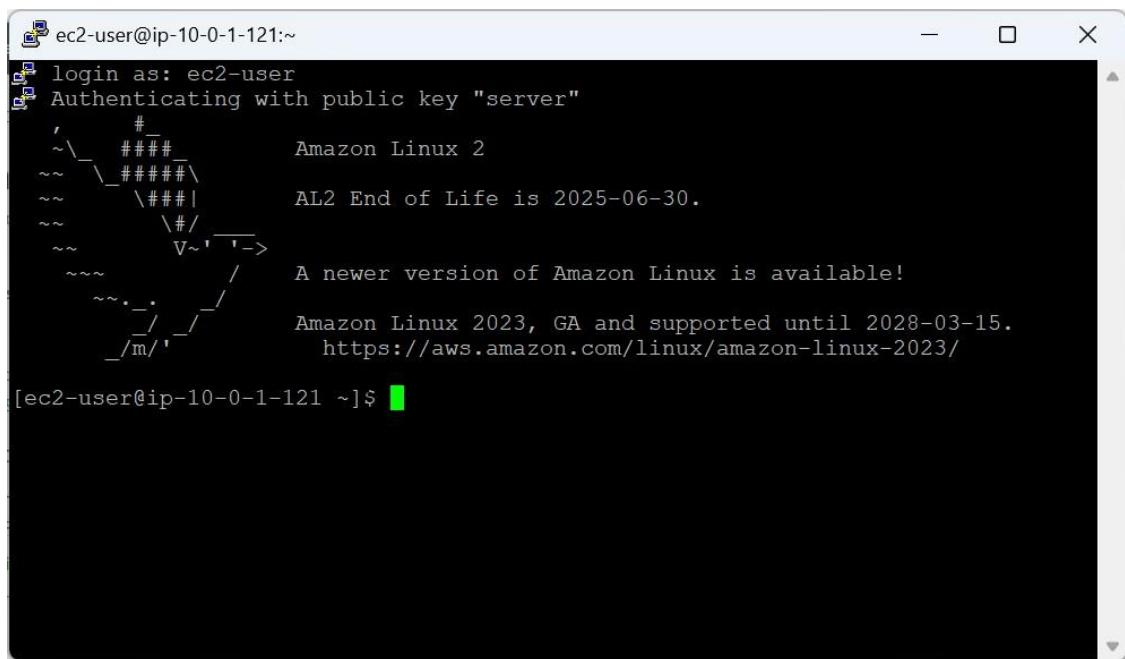
Select an instance

CloudShell Feedback

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

2. After launched connect it and write the below command.

```
ec2-user
sudo su
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
cd /var/www/html
nano index.html
<a href="your website name">Website name</a>
Ctrl x
Y
Ctrl c+enter.
```



The screenshot shows a terminal window with the following text:

```
ec2-user@ip-10-0-1-121:~$ login as: ec2-user
Authenticating with public key "server"
'          #
~\_\_###           Amazon Linux 2
~~\_\_\#\#\#\`       AL2 End of Life is 2025-06-30.
~~\_\#\#\|           V~,'-->
~~\_\#\#/\           A newer version of Amazon Linux is available!
~~\_\#\#\`/           Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/\_\_`/           https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-0-1-121 ~]$
```

```
root@ip-10-0-1-121:/home/ec2-user
login as: ec2-user
Authenticating with public key "server"
'__#_#
~~\###\ Amazon Linux 2
~~ \###| AL2 End of Life is 2025-06-30.
~~ \#/ V~'-->
~~ / A newer version of Amazon Linux is available!
~~ . / Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/,-/ https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-10-0-1-121 ~]$ sudo su
[root@ip-10-0-1-121 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[root@ip-10-0-1-121 ec2-user]# yum install httpd -y
```

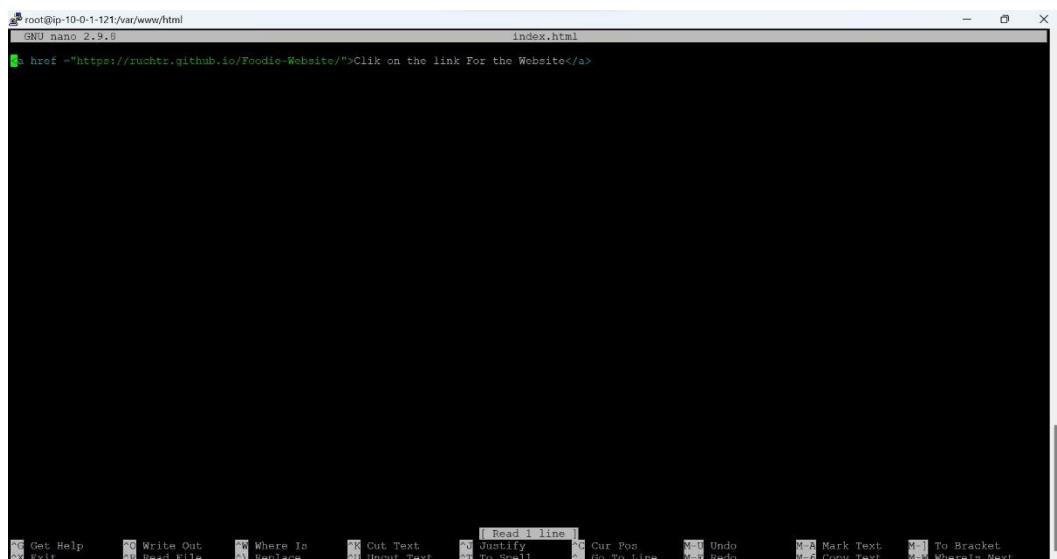
```
root@ip-10-0-1-121:/home/ec2-user
Verifying : httpd-2.4.58-1.amzn2.x86_64 4/9
Verifying : apr-1.7.2-1.amzn2.x86_64 5/9
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64 6/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 7/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 8/9
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.58-1.amzn2

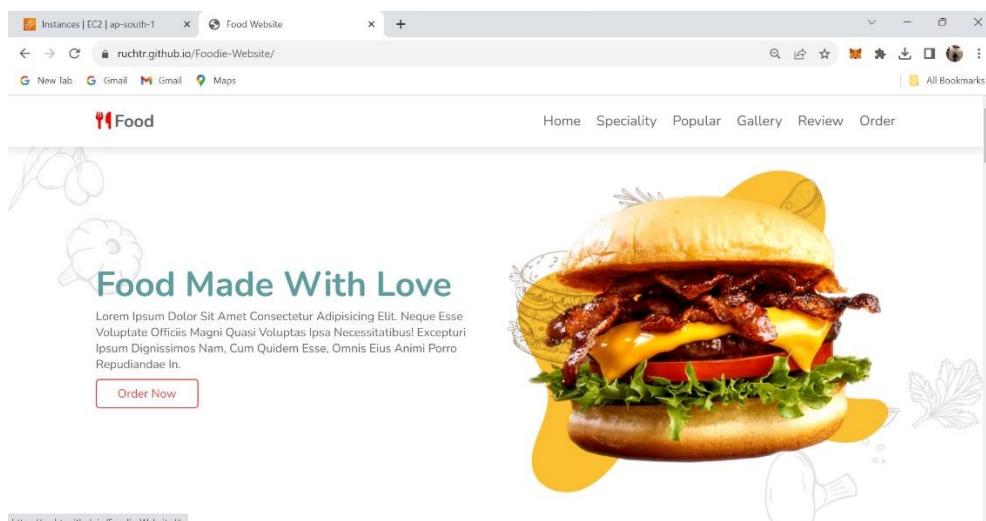
Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.58-1.amzn2
httpd-tools.x86_64 0:2.4.58-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-10-0-1-121 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-121 ec2-user]# chkconfig httpd on
```

```
root@ip-10-0-1-121:/var/www/html
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64 9/9
Installed:
httpd.x86_64 0:2.4.58-1.amzn2
Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httdp.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.58-1.amzn2
httpd-tools.x86_64 0:2.4.58-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1
Complete!
[root@ip-10-0-1-121 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-121 ec2-user]# chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-1-121 ec2-user]# cd /var/www/html
[root@ip-10-0-1-121 html]# nano index.html
```



3. Come to aws console and copy the web server instance URL and paste it on the new tab



4. After that create a AMI of the web server with same key pair and security group.

The screenshot shows three sequential steps in the AWS EC2 console:

- Instances (1/3) Info:** The user has selected the "Web server" instance (i-0962850dc415fb90a). In the Actions menu, the "Create image" option is highlighted with a red box.
- Create Image | EC2 | ap-south-1:** The "Image name" field is filled with "AMI Image". The "Image description" field contains "Image description". The "Storage type" section shows an EBS volume named "/dev/xvda" with a size of 8 GiB and an IOPS of 100. The "Delete on termination" checkbox is checked.
- Images | EC2 | ap-south-1:** The newly created AMI, "AMI Image", is listed in the "Amazon Machine Images (AMIs)" table. It has the ID ami-0d643f8a0ba36f68c, is owned by the user, and is in a Pending state.

The screenshot shows the AWS EC2 Images console. In the center, there is a table titled "Amazon Machine Images (AMIs) (1) Info". The table has columns for Image, AMI ID, AMI name, Source, Owner, Visibility, and Status. The first row contains the following data:

Image	AMI ID	AMI name	Source	Owner	Visibility	Status
ami-0d64f8a0ba36f6bc	AMI Image	193394235970/AMI Image	193394235970	Private		Available

A red box highlights the "Status" column header and the "Available" status of the AMI.

Note: AMI takes time to get available.

Once it is available the next step is to launch a template.

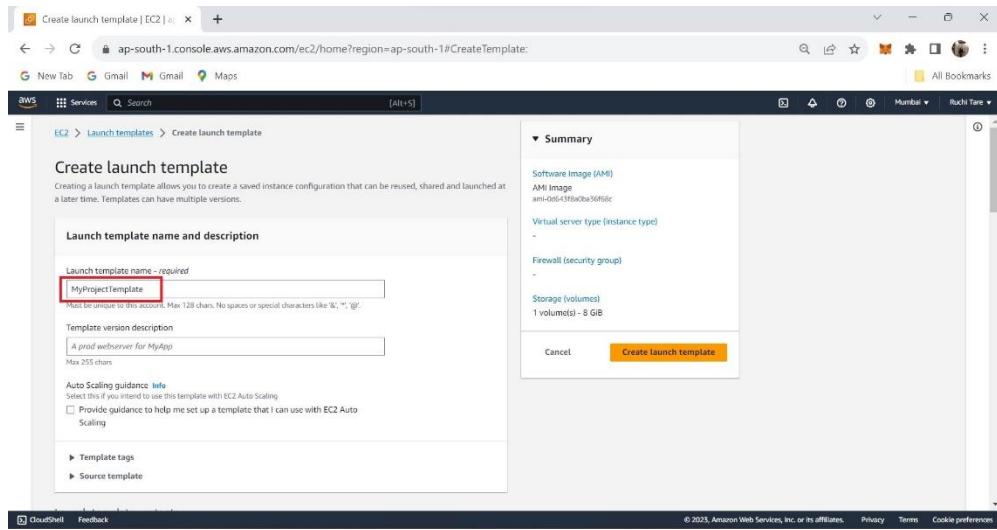
A template is like a pre-made set of instructions or a blueprint that tells a computer system, like AWS, how to create and configure resources. It's a handy way to automate the process of setting up things like servers or services with specific settings, saving time and ensuring consistency in the deployment of resources

Step 4:

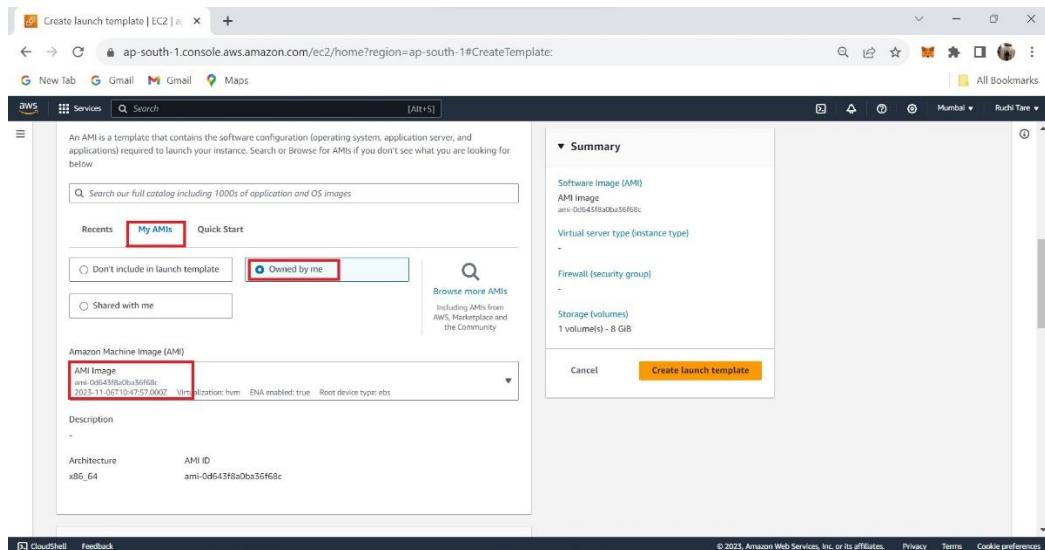
1. Go to the launch template in EC2 click in create launch template.

The screenshot shows the AWS EC2 Launch Templates console. The main heading is "EC2 launch templates" with the subtext "Streamline, simplify and standardize instance launches". Below this, there is a section titled "Benefits and features" with two items: "Streamline provisioning" and "Simplify permissions". To the right, there is a "New launch template" card with a "Create launch template" button, which is highlighted with a red box. At the bottom right of the card, there are links for "Documentation" and "API reference".

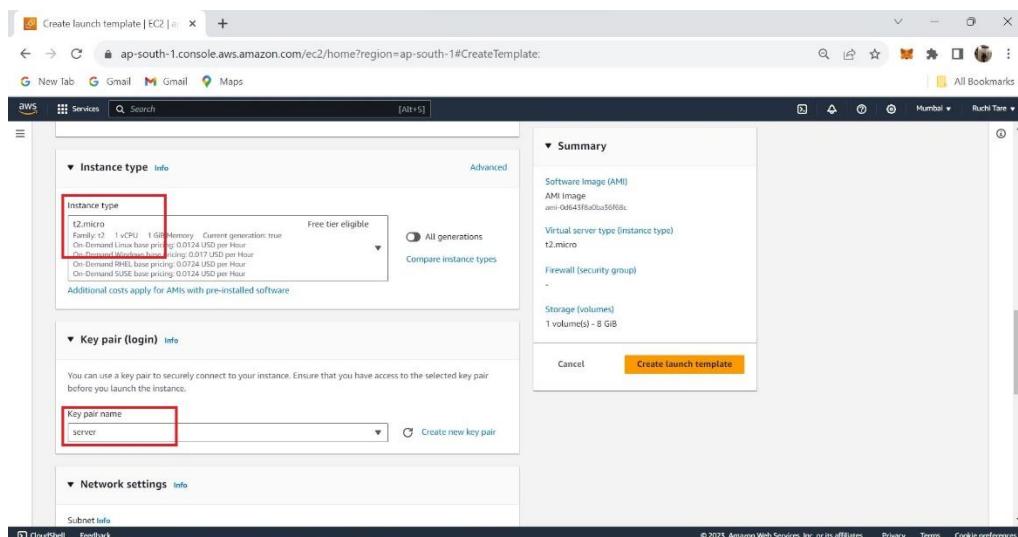
2. Give name to your template.

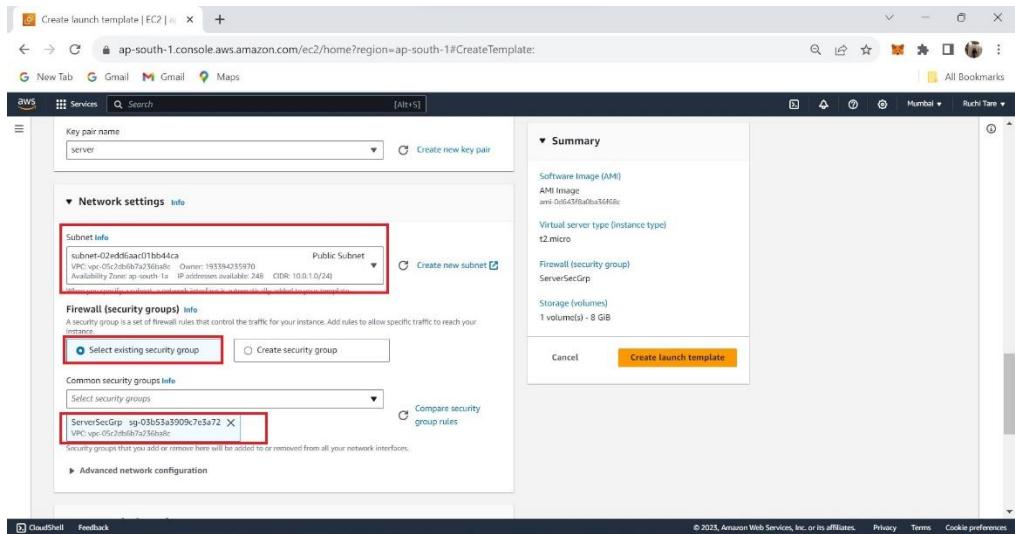


3. Click on My Ami ad your created ami will be visisble.

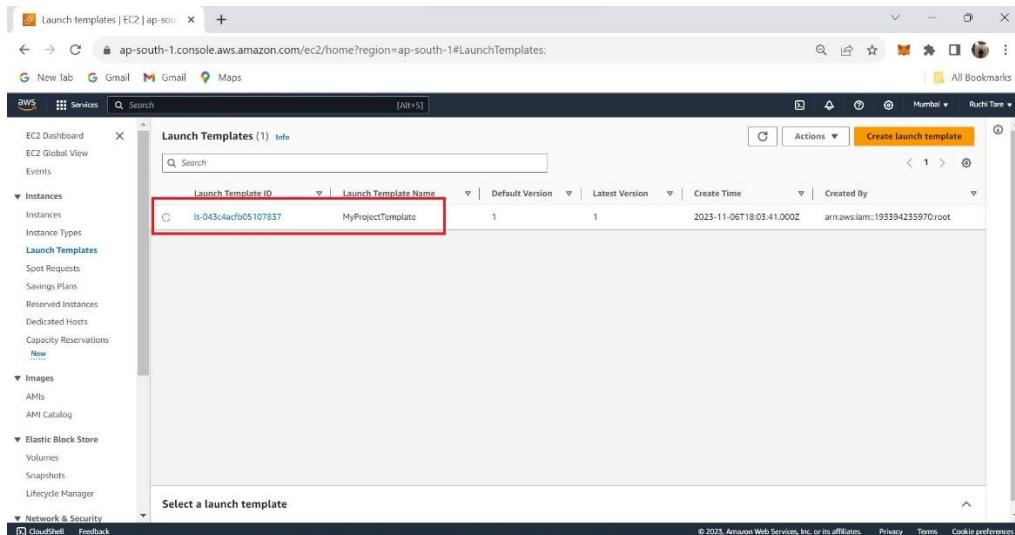


4. Select the same key pair , subnet should be public and security group should be same.





Click on the launch button.



Your template is created.

Step 5:

Once the template is launched, next step is to launch a classic load balancer.

1. Select the load balancer from EC2 service and click on create load balancer.

The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a sidebar with various EC2 services like Instances, AMIs, and Network & Security. The main area is titled 'Load balancers' and contains a table with columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. A red box highlights the 'Create load balancer' button at the top right of the table.

2. Scroll Down and click on previous generation → classic load balancer → create.

The screenshot shows the 'Select Create ELB Wizard' page. It compares Application Load Balancers (ALBs) and Network Load Balancers (NLBs). Below this, it shows the 'Classic Load Balancer - previous generation' section, which is highlighted with a red box. A diagram illustrates how traffic flows from clients through an Application Load Balancer (ALB) to a Network Load Balancer (NLB), which then routes traffic to EC2 instances. A red box highlights the 'Create' button in this section.

3. Give the name for the load balancer and select internet – facing.

The screenshot shows the 'Create Classic Load Balancer' wizard. In the 'Basic configuration' step, the 'Load balancer name' field is highlighted with a red box. The name 'MyProjectLoadBalancer' is entered. Below this, the 'Scheme' section shows 'Internet-facing' selected, indicated by a red box. The 'Network mapping' section at the bottom is also partially visible.

4. Select the created VPC and select public subnet.

VPC **Info**
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created. When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. [Learn more](#)

MyProject-VPC
vpc-05c2e6b7a236ba8c
IPv4: 10.0.0.0/16

Settings
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

ap-south-1a (aps1-a#1)
Subnet
subnet-02edd6aac01bb44ca **Public Subnet**

Security groups **Info**
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups
Select up to 5 security groups

5. Select the same security group which you have created for Web server instance.

Security groups **Info**
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups
Select up to 5 security groups

ServerSecGrp
sg-03b53a3909c7e5a72 VPC vpc-05c2e6b7a236ba8c
default
sg-010xbae41e2cf3d3a VPC vpc-05c2e6b7a236ba8c

Listeners and routing **Info**
A listener is a process that checks for connection requests using the protocol and port you configure. The settings you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80
Instance HTTP:80

Listener protocol	Listener port	Instance protocol	Instance port
HTTP	80	HTTP	80

6. Select the Web server instance as you are going to create the load balancer of web server which will balance the load of your server → then click on create load balancer.

Instances (1)
You can add instances to register as targets of the load balancer. Alternatively, after your load balancer is created, you can add it to an Amazon EC2 Auto Scaling group. [Learn more](#)
The correct number of instances to handle the load for your application. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone.

Filter instances

Instance ID	Name	State	Security groups
i-0962850dc415fb90a	Web server	Running	ServerSecGrp

Attributes
Creating your load balancer using the console gives you the opportunity to specify additional features at launch. You can also find and adjust these settings in the load balancer's "Attributes" section after your load balancer is created.

Enable cross-zone load balancing
With cross-zone load balancing, your load balancer routes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer mode distributes requests evenly among the registered instances in its Availability Zone only. Classic Load Balancers created with the AWS CLI have cross-zone load balancing disabled by default. After you create a Classic Load Balancer, you can enable or disable cross-zone load balancing at any time.

7. The Load Balancer is created.

The screenshot shows the AWS EC2 Load Balancers page. The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, CloudShell, and Feedback. The main content area is titled 'Load balancers (1)' and contains a table with one row. The table columns are Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. The row shows 'MyProjectLoadBalancer' as the Name, 'MyProjectLoadBalancer-85...' as the DNS name, '-' as the state, 'vpc-05c2db6b7a236b...' as the VPC ID, 'ap-south-1a (ap-s1-a21)' as the Availability Zones, 'classic' as the Type, and 'November 6, 2023, 23:45 (..)' as the Date created. There is a 'Create load balancer' button at the top right of the table. At the bottom of the page, it says '0 load balancers selected'. The footer includes links for © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 6:

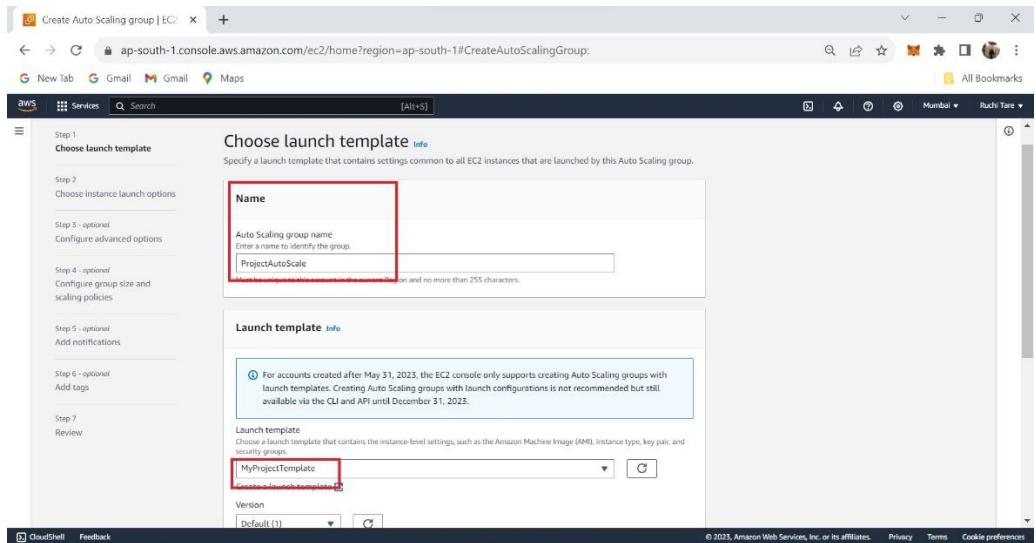
Now that your load balancer is created the next step of yours is to create a Auto Scaling Group.

Auto Scaling Group is used here to Scale the instance in and out when the instance extend its load.

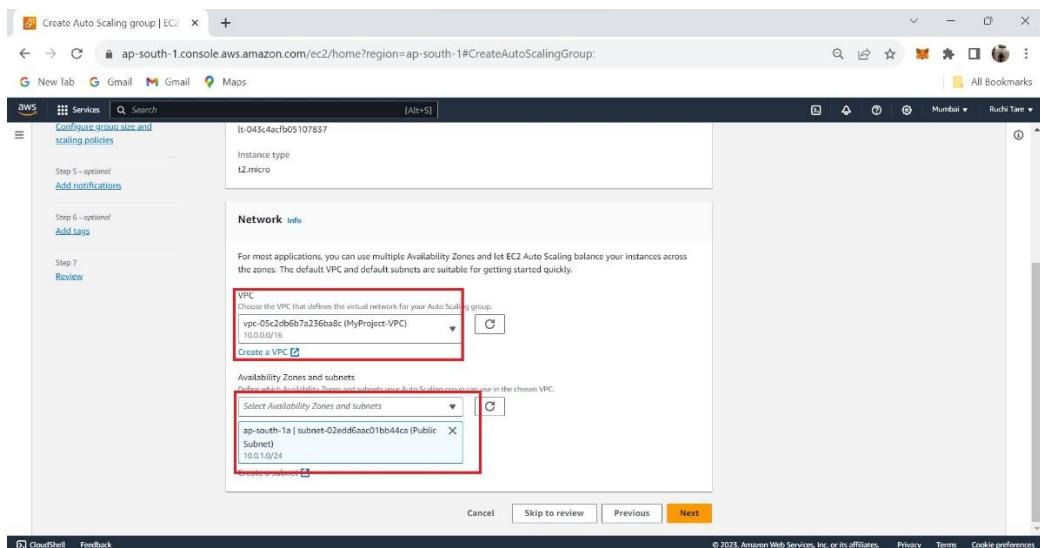
1. Select the Auto Scaling Group in the EC2 service and then click on create Auto Scaling Group.

The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar is identical to the previous screenshot. The main content area features a large banner with the text 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below the banner, there is a section titled 'How it works' with a diagram showing four squares representing EC2 instances in an 'Auto Scaling group'. To the right of the diagram are sections for 'Pricing' and 'Getting started'. A prominent orange 'Create Auto Scaling group' button is located in the center of the main content area. The footer includes links for © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

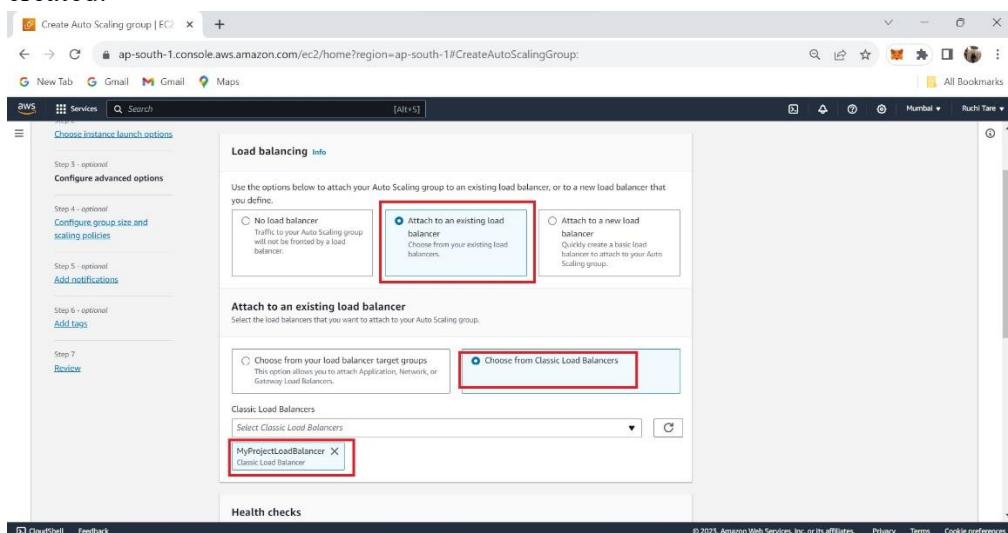
2. Give your Auto Scaling Group a name and select your created template.



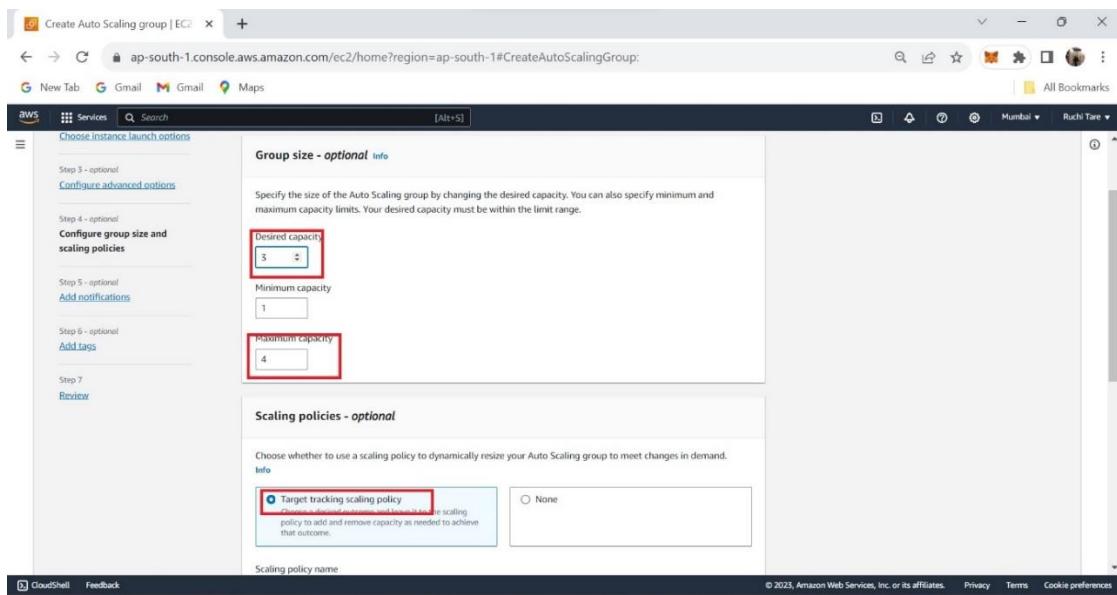
3. Select the VPC that you have created and public subnet.



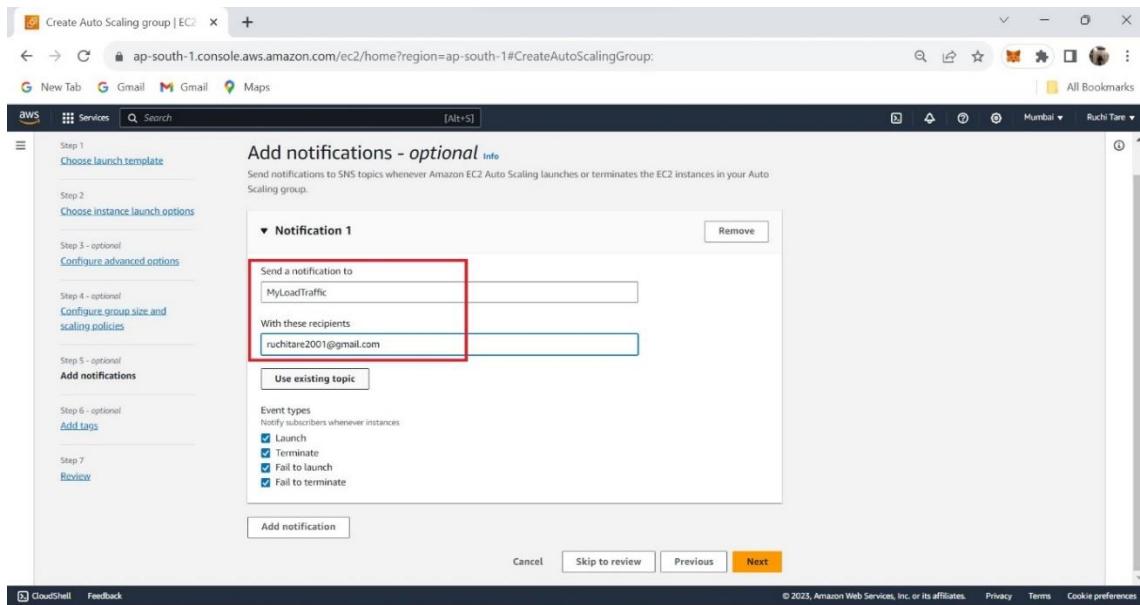
4. Click on next button and next step is to attach load balancing → Attach existing load balancer → choose from classic load balancer → Select the load balancer that you have created.



5. Click next → choose the desired capacity of instance you want to create first→then choose the maximum instance you want.



6. Auto Scaling Group will send you the message whenever the a load on one instance exceed and another instance is created so for that we will create an SNS group.
Click on next→Give name title to your message→enter the recipients to whom you want to send message →click on next→ Review your Auto Scaling Group and then click on create.



The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, CloudShell, and Feedback. The main content area displays a table titled 'Instances (7) Info' with a filter 'Instance state = running'. The table columns are Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 address. There are seven rows, each representing a different instance, all of which are currently running.

The above 4 instances is created by the Auto Scaling Group. Where the load is extended beyond 50% of the CPU limit.

Go to the load Balancer and then check whether the status is showing 4 of 4 instances in service which means your load balancer is working properly and it is dividing the load between the instances.

The screenshot shows the AWS Load Balancers page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, CloudShell, and Feedback. The main content area displays a table titled 'Load balancers (1)' with a filter 'Filter load balancers'. The table columns are Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. One row is visible, labeled 'MyProjectLoadBalancer-83...', with its DNS name highlighted in red. Below the table, it says '0 load balancers selected'.

The screenshot shows the AWS Cloud Console interface for managing a Load Balancer. The main navigation bar includes 'Services' (selected), 'Search', and 'Mumbai' (Region). The left sidebar lists various AWS services: Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'MyProjectLoadBalancer' under 'Load balancers'. It shows the following details:

- Details** section:

Load balancer type:	Classic	Status:	4 of 4 instances in service
VPC:	vpc-05c2d0b7a236ba8c	Date created:	November 7, 2023, 00:10 (UTC+05:30)
Scheme:	Internet-facing	Hosted zone:	ZP5YRAFLXTHZK
Availability Zones:	subnet-02edd6aac01bb44ca (ap-south-1a)		
- DNS name info:** MyProjectLoadBalancer-969202507.ap-south-1.elb.amazonaws.com (A Record)
- Distribution of targets by Availability Zone (A2):** A table showing registered instances and their current health status.
- Listeners:** Tab selected, showing a list of configured listeners.

Click again on the DNS of the load balancer then check whether your website is working properly.

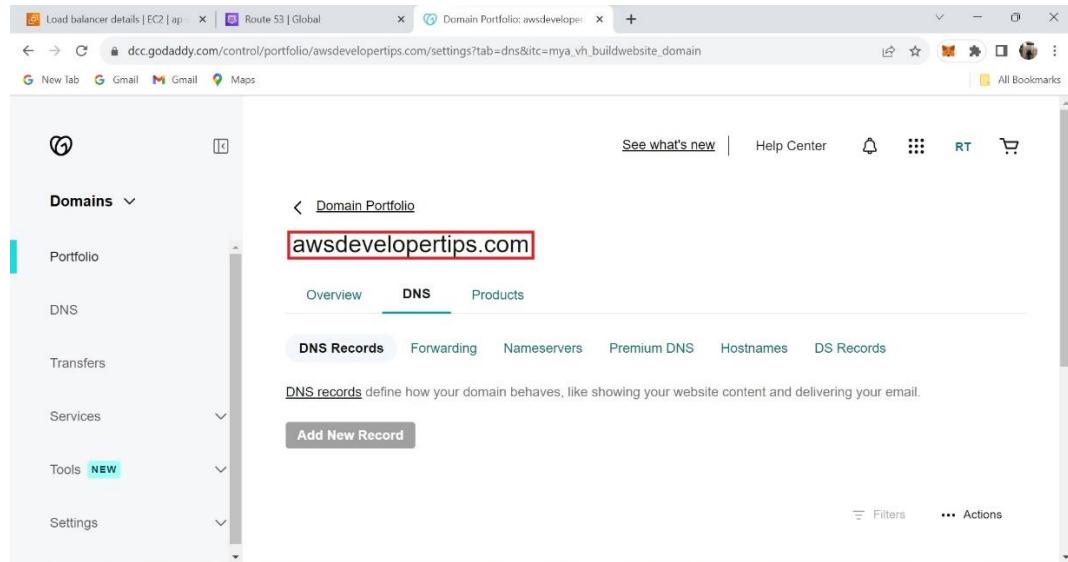
The screenshot shows a web browser window with the following details:

- Address bar: myprojectloadbalancer-969202507.ap-south-1.elb.amazonaws.com
- Page content: A simple message 'Clik on the link For the Website'.

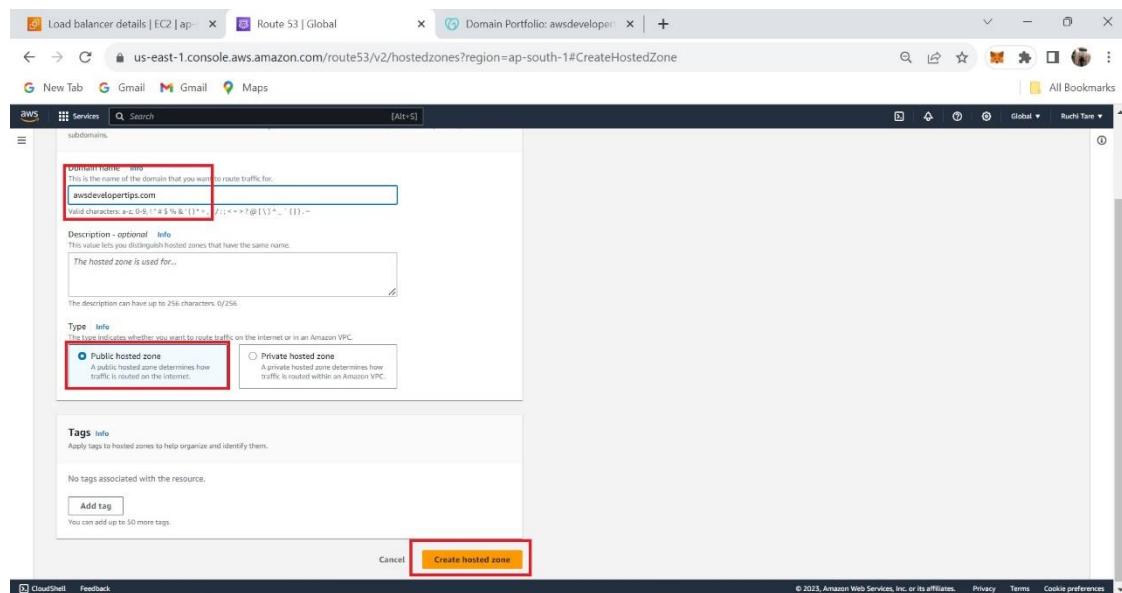
The screenshot shows a web browser displaying a food website. The address bar shows the URL: ruchtr.github.io/Foodie-Website/. The page has a header with the title 'Food' and a sub-header 'Food Made With Love'. Below the header, there is a paragraph of placeholder text (Lorem Ipsum) and a 'Order Now' button. The main feature is a large, detailed image of a bacon cheeseburger. The navigation menu at the top includes Home, Speciality, Popular, Gallery, Review, and Order. The footer contains a link to the website's GitHub repository: https://ruchtr.github.io/Foodie-Website/#.

Step 7: Now we have to host our website on the server for that we are going to use Route 53.

1. Before creating a hosted zone for your website you will have to create a domain name.
2. You can create a domain name in aws itself or else you can buy it from [Godaddy.com](#)
3. Click on the link below and complete the registration details and also buy the domain that you want.



4. Once you have bought the domain name go to Route 53 and click on Hosted click → create hosted zone → add the domain name to your hosted zone and also select public hosted zone → click on create hosted zone.



- Now click on the created hosted zone and copy the name server URL and paste it on the name server of your godaddy account.

The screenshot shows the AWS Route 53 console with the domain 'awsdevelopertips.com' selected. The 'Hosted zone details' section is visible, showing the 'Records' table. The table contains two entries under the 'NS' type:

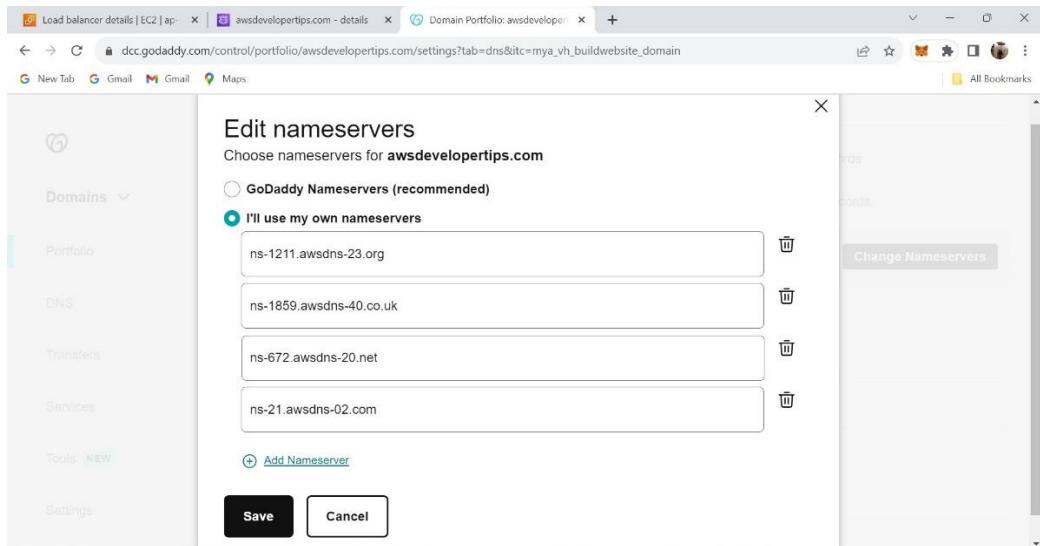
Type	Value	TTL
NS	ns-1211.awsdns-23.org ns-1859.awsdns-40.co.uk	172800
SOA	ns-672.awsdns-20.net ns-21.awsdns-02.com	900

- Go to your godaddy account click on nameservers → change nameserver.

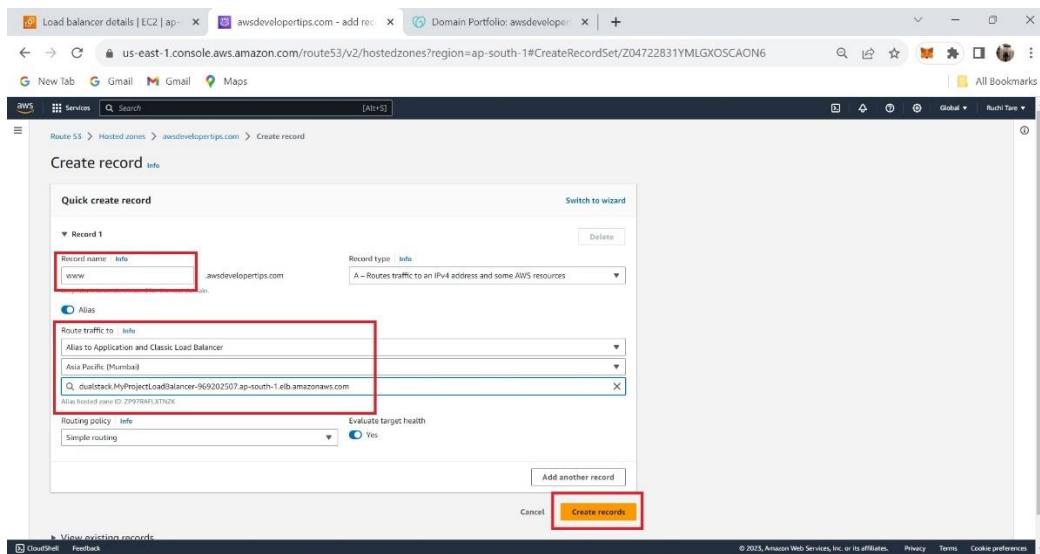
The screenshot shows the GoDaddy DNS settings page for the domain 'awsdevelopertips.com'. The 'Nameservers' tab is active. A red box highlights the 'Change Nameservers' button. The page displays a list of current nameservers:

- ns-1462.awsdns-54.org
- ns-568.awsdns-07.net
- ns-142.awsdns-17.com
- ns-1617.awsdns-10.co.uk

- Paste all the name server that you have copied from your hosted zone.



8. Create a record with subdomain www and load balancer endpoint and then click on create record.



9. Once the record is created click again on the hosted zone and click on the URL which is given by aws and paste it on new tab to see your website.

The screenshot shows the AWS Route 53 service in the AWS Management Console. On the left, there's a navigation sidebar with options like Dashboard, Hosted zones, Health checks, IP-based routing, CDR collections, Domains, Resolver, VPCs, and DNS Firewall. The main area is titled 'Route 53' and shows a message: 'Record for awsdevelopertips.com was successfully created. Route 53 propagates your changes to all of the Route 53 authoritative DNS servers within 60 seconds. Use "View status" button to check propagation status.' Below this, it says 'Public' and lists 'awsdevelopertips.com' under 'Info'. There are buttons for 'Delete zone', 'Test record', and 'Configure query logging', along with an 'Edit hosted zone' button. The 'Records' tab is selected, showing a table with three entries. The first two entries are for 'awsdevelopertips.com' (NS and SOA types). The third entry, 'www.awsdevelopertips.com', is highlighted with a red box and has its details shown on the right side of the screen. The 'Record details' pane shows the following information for the 'www' record:

Record name	Type	Routing policy	Alias	Value/Route traffic to
awsdevelopertips.com	NS	Simple	No	ns-1211.awsdns-23.org ns-1599.awsdns-40.co.uk. ns-672.awsdns-20.net. ns-21.awsdns-02.com
www.awsdevelopertips.com	A	Simple	Yes	dualstack.myprojectloadbalancer-9690202507.ap-south-1.elb.amazonaws.com

On the far right, there are buttons for 'Edit record', 'Record type' (set to 'A'), 'Value' (containing 'dualstack.myprojectloadbalancer-9690202507.ap-south-1.elb.amazonaws.com'), 'Alias' (set to 'Yes'), 'TTL (seconds)' (set to '1'), and 'Routing policy' (set to 'Simple').

As you can see the website is visible through the URL.

