**AWS 2-Tier Project**

**VPC with public-private subnet in Production**

**About the Project:**

This example demonstrates how to create a VPC that you can use for servers in a production environment.To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.


        A VPC with subnets in two Availability Zones.
      

**Overview:**

The VPC has public subnets and private subnets in two Availability Zones.

Each public subnet contains a NAT gateway and a load balancer node.

The servers run in the private subnets, are launched and terminated by using an

Auto Scaling group, and receive traffic from the load balancer.

The servers can connect to the internet by using the NAT gateway.

**Resources required:**

- Auto Scaling Group

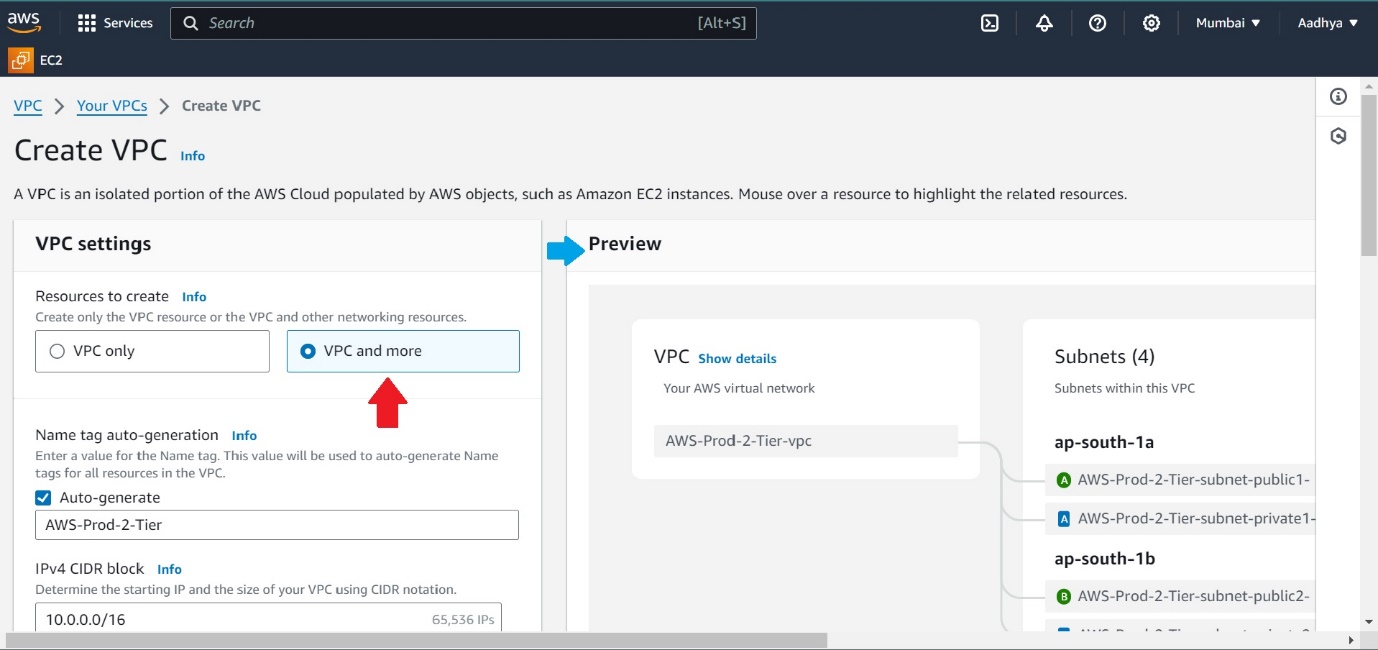
- Load Balancer

- Target Group

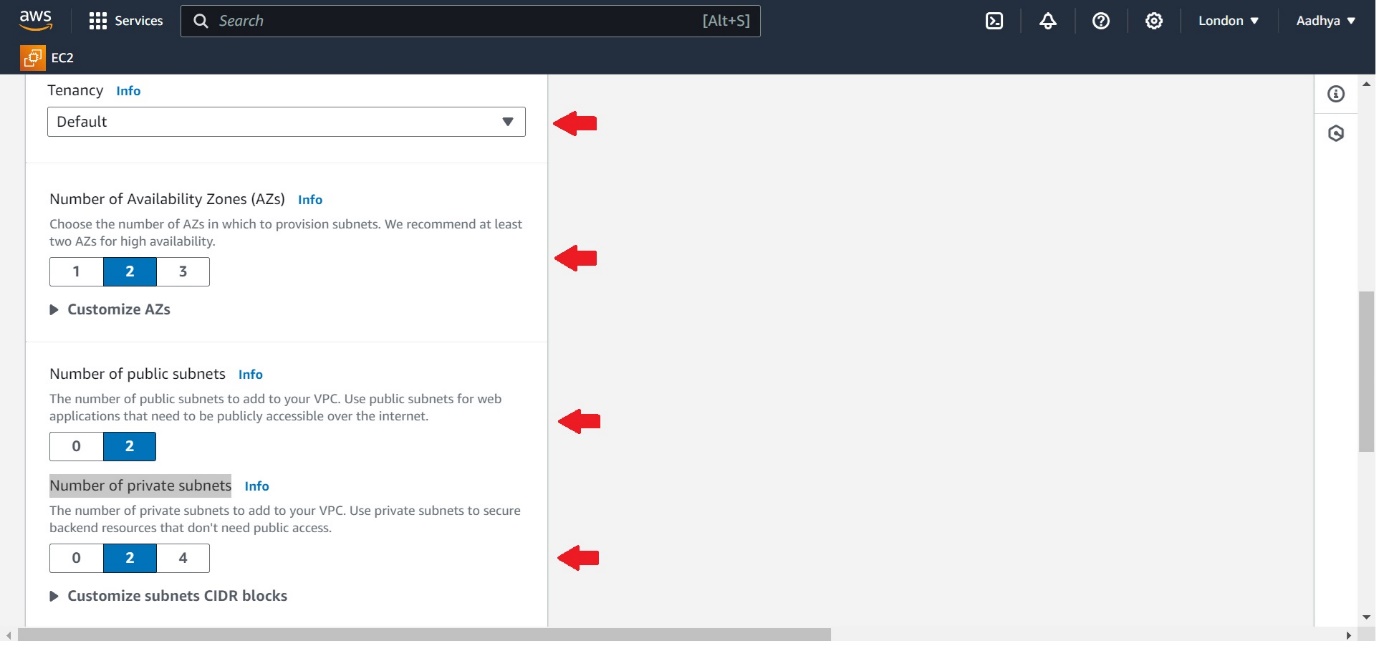
- Bastion Host or Jump Server

**Steps:**

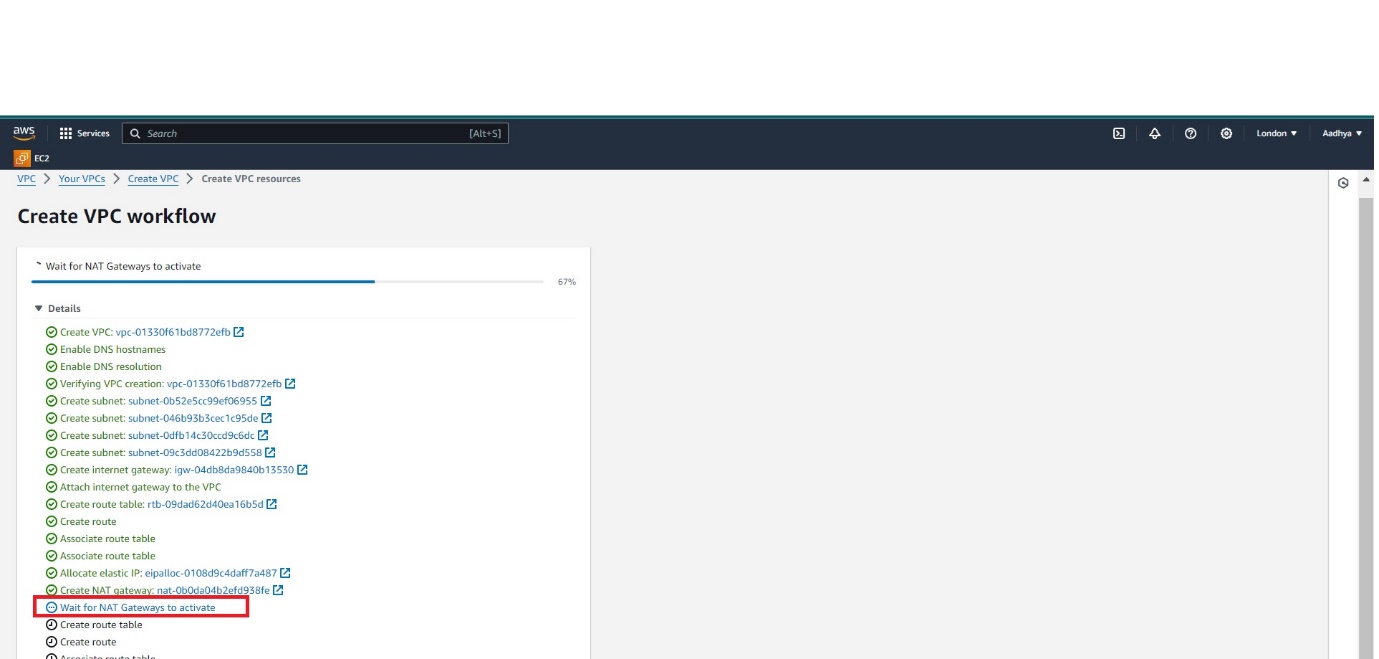
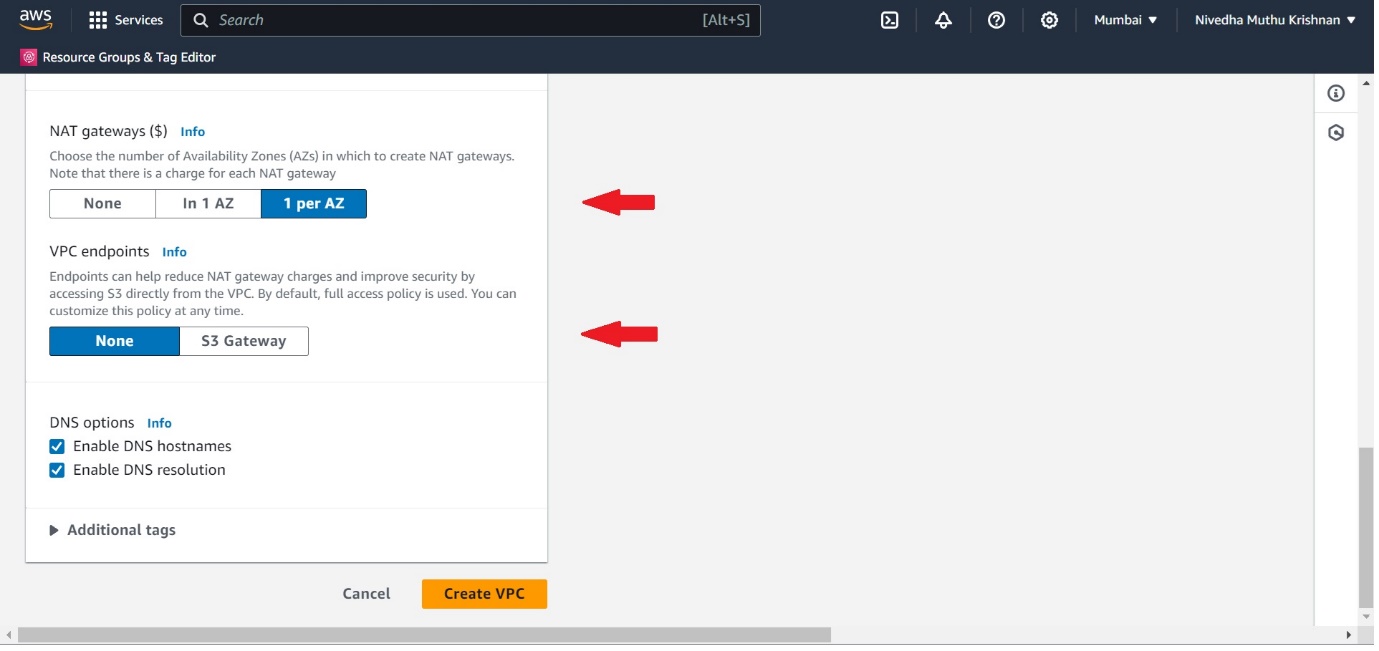
1. Login into AWS console and go to VPC dashboard to create VPC.
2. Click on Create VPC.
3. In VPC settings, select VPC and more. Once you select VPC and more you can see a preview diagrammatic representation at the right side.
4. Select the name of the project – AWS-Prod-2-Tier. In IPv4 CIDR Block range as 10.0.0.0/16.



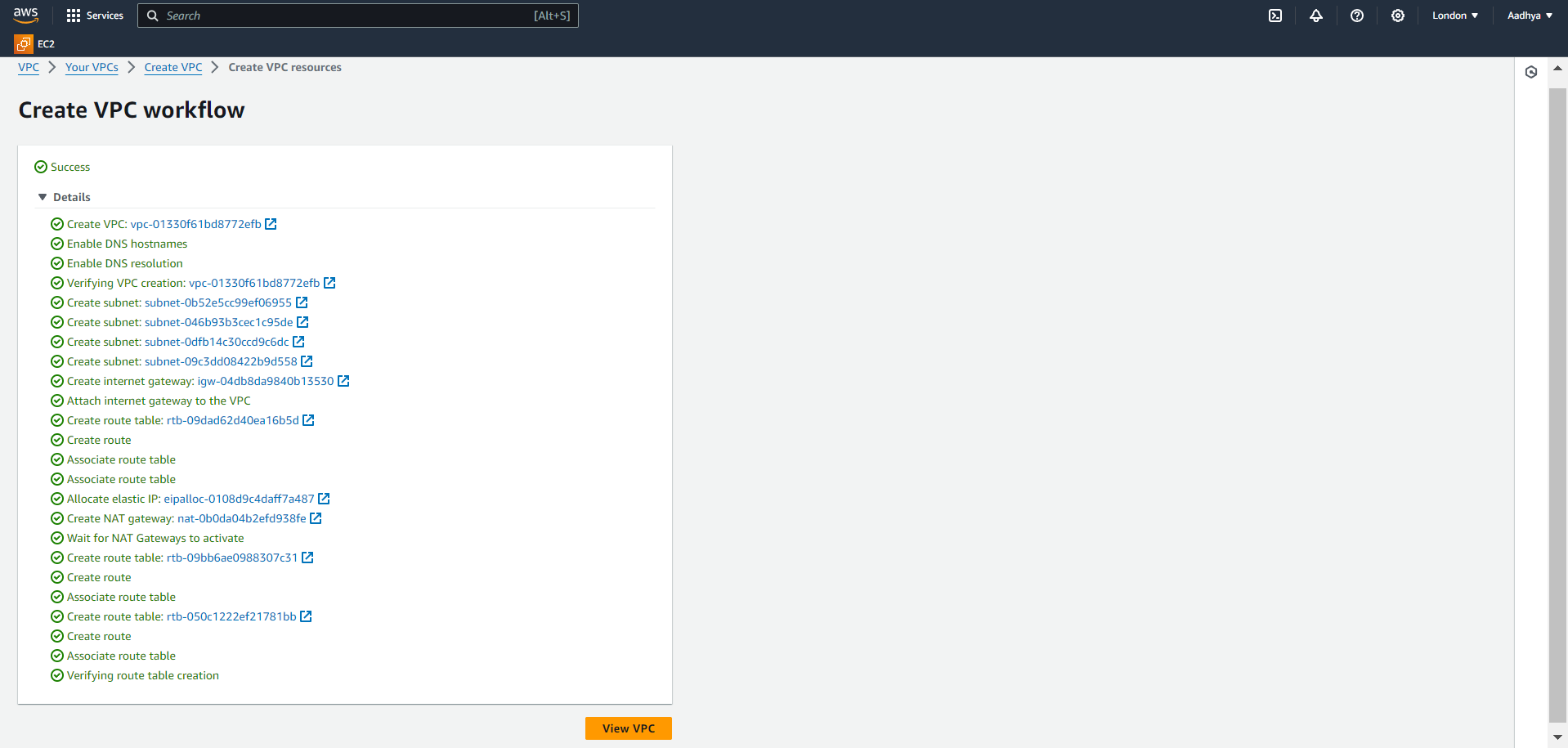
1. Keep the Tenancy as Default itself.
2. Select 2 from all Number of Availability Zones (AZs), Number of public subnets & Number of private subnets.

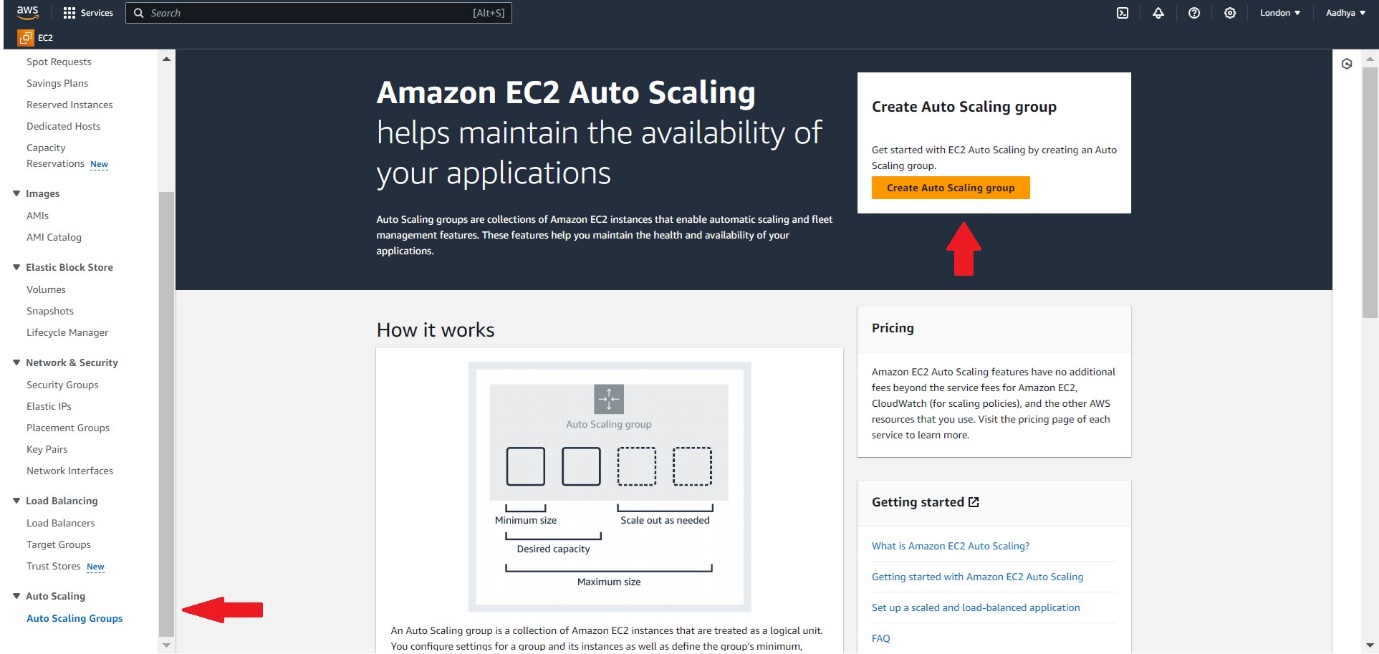


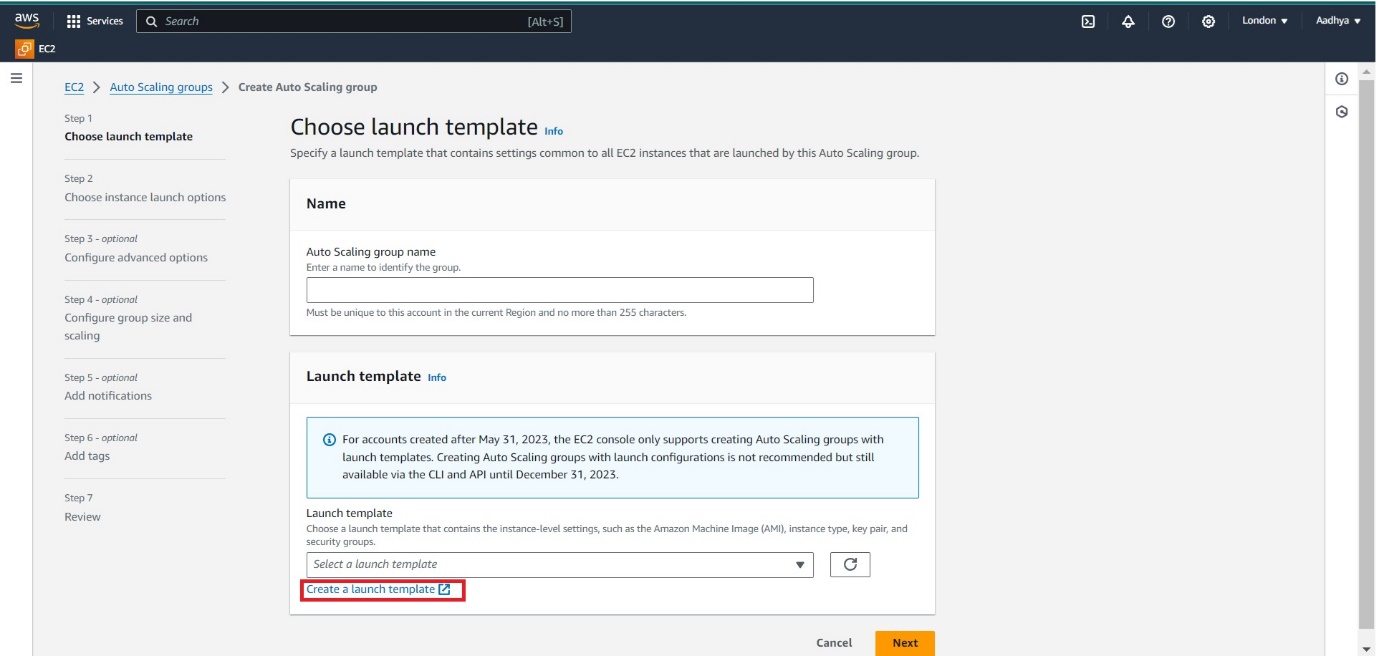
1. In NAT gateways, select - In 1 AZ.
2. Select None in VPC endpoints.
3. Keep all other options as same and click on Create VPC.
4. Now you can see VPC is getting created, and in while creating NAT gateway it takes time so wait till it gets completed.



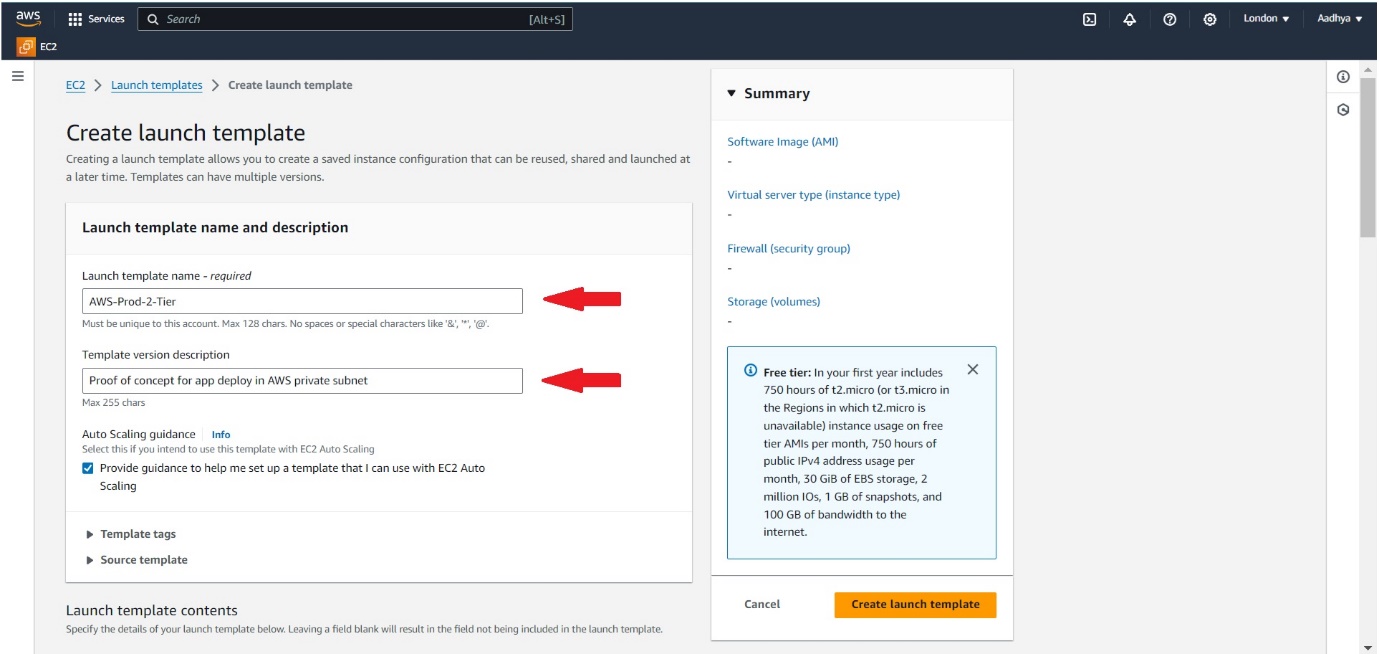
1. Now you can see VPC has been created successfully. Just check if all we created is present or not.



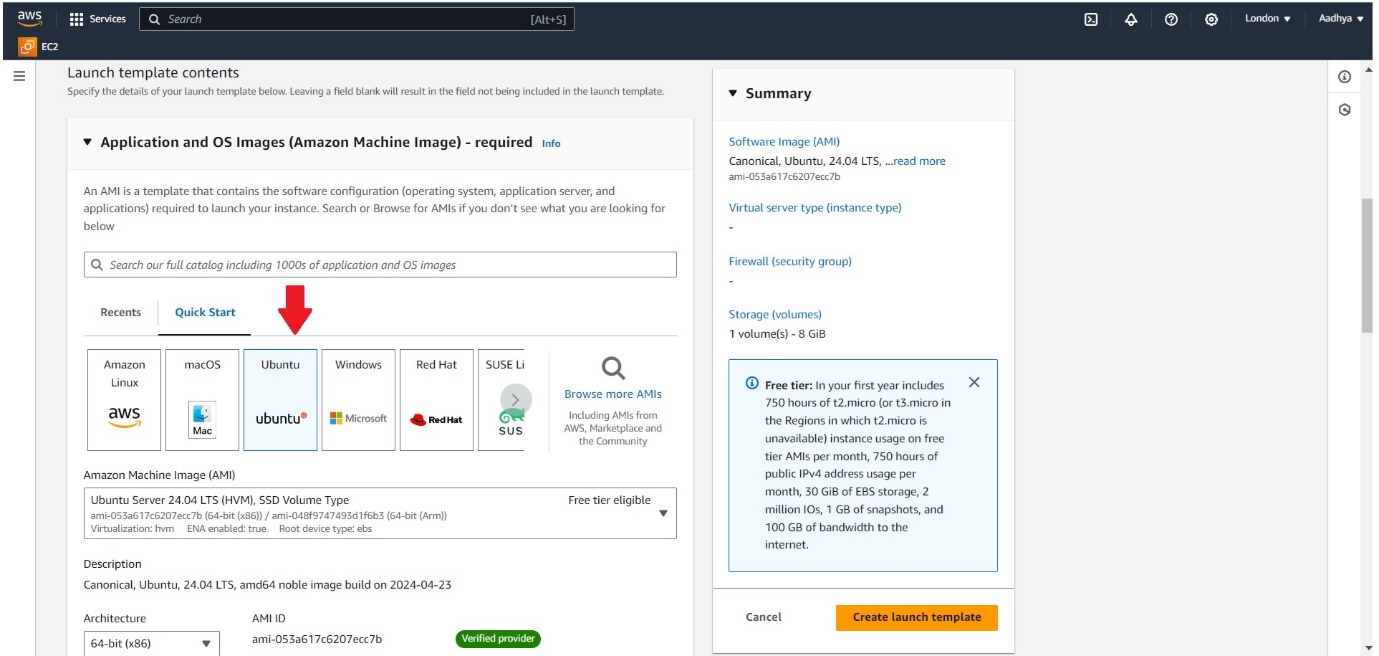
1. Now go to EC2 dashboard, to find ASG(Auto Scaling Group) and then click on Create Auto Scaling Group.
2. Once you are in to ASG page, we have to create ASG template so click on Create a launch template it will automatically redirects to a new tab to create a new template.

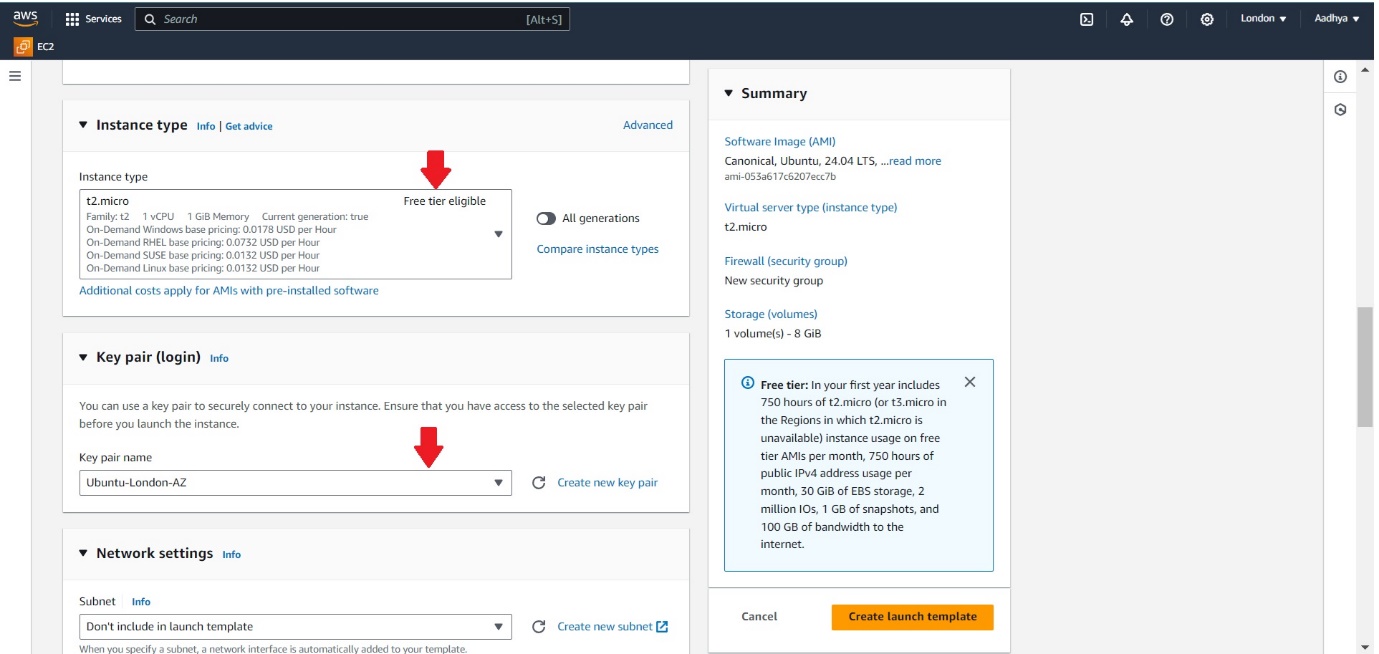


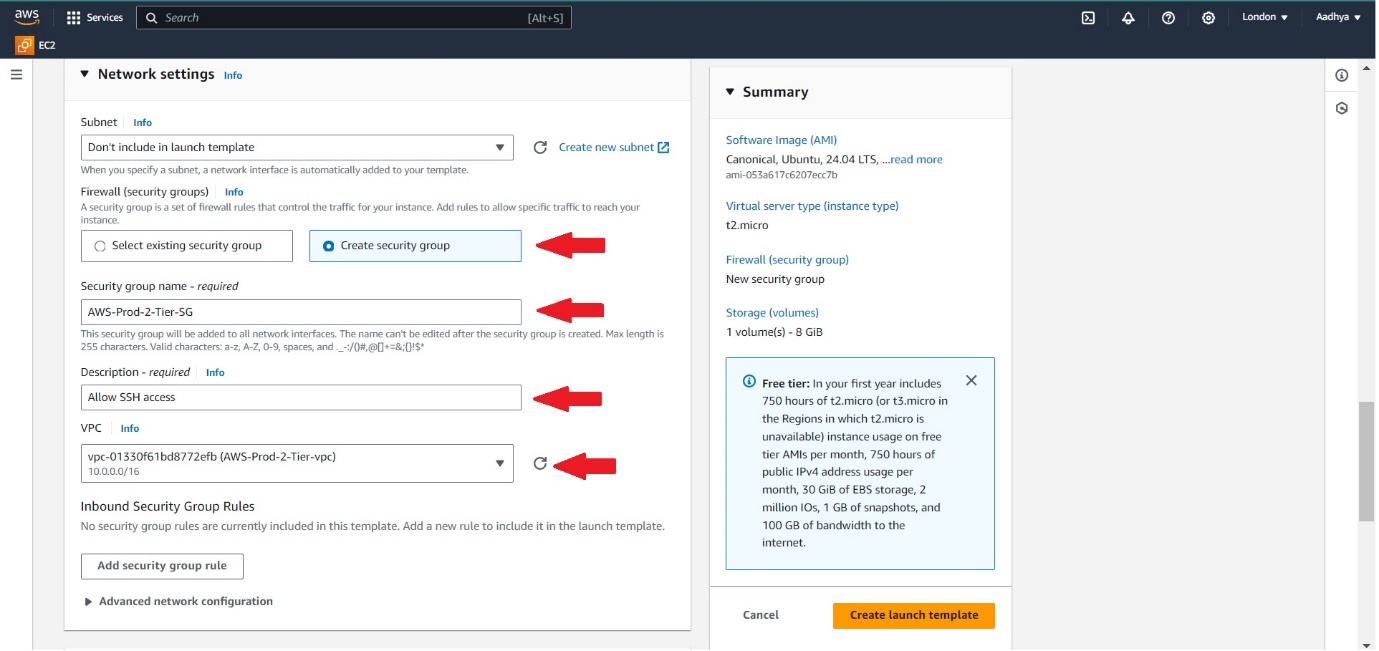
1. In ASG Template page, Launch template name required- AWS-Prod-2-Tier, Template version description- Proof of concept for app deploy in AWS private subnet.

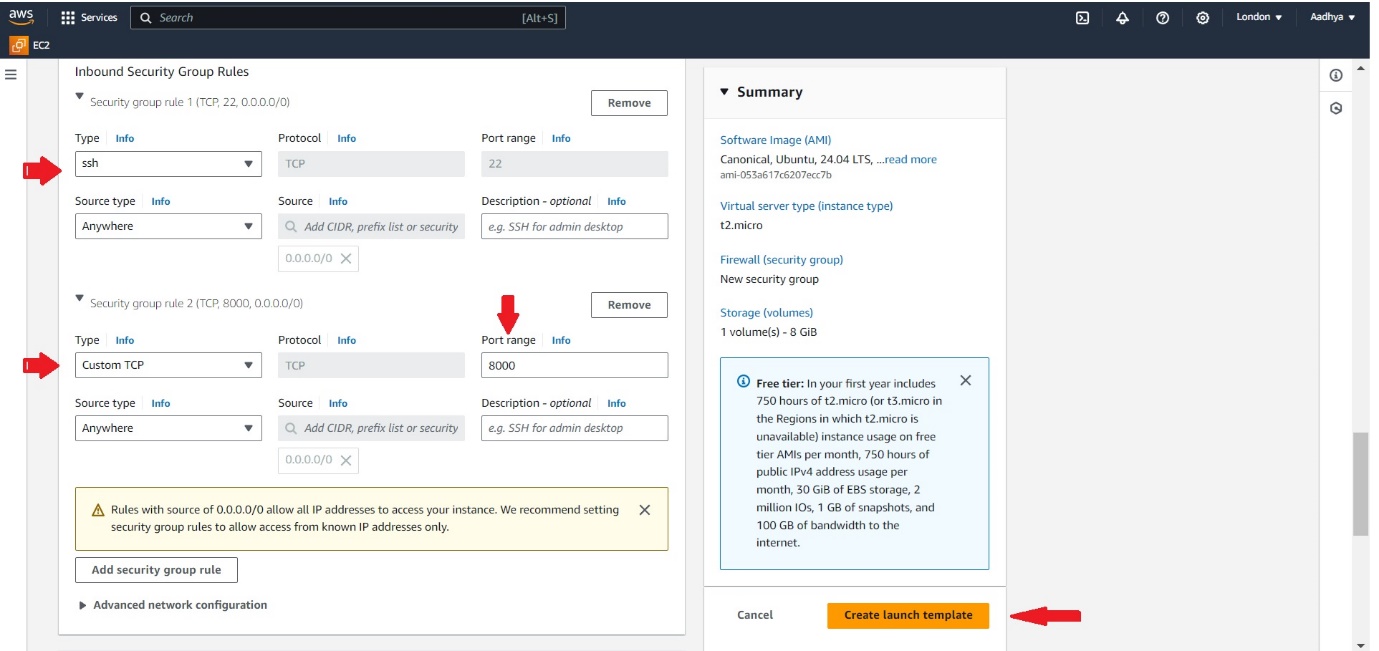


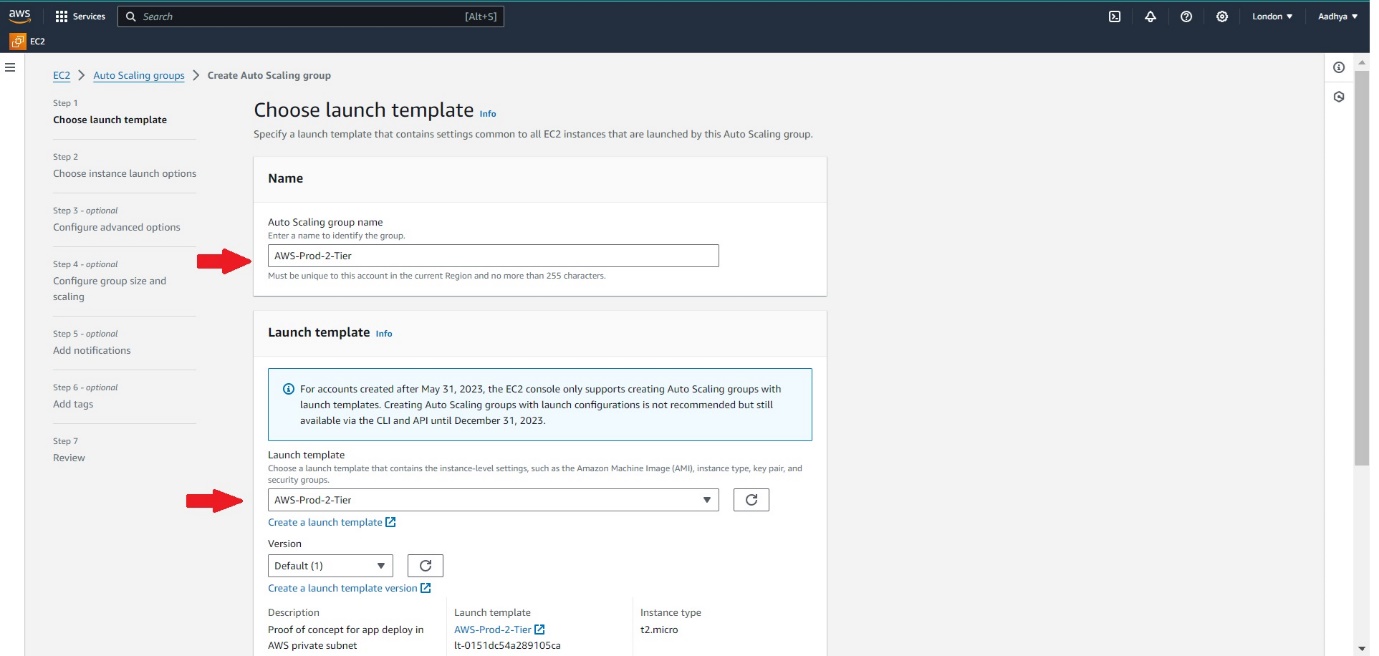
1. Select Application and OS Images (Amazon Machine Image) – required- as Ubuntu AMI.

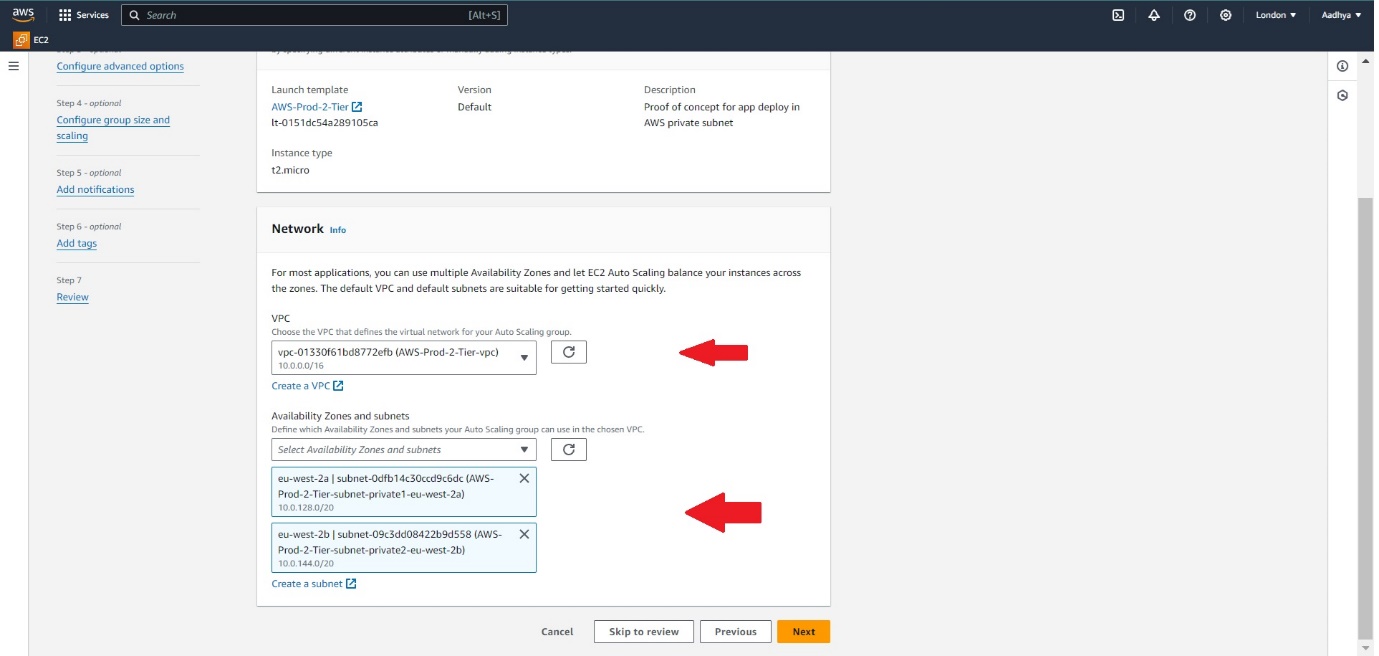


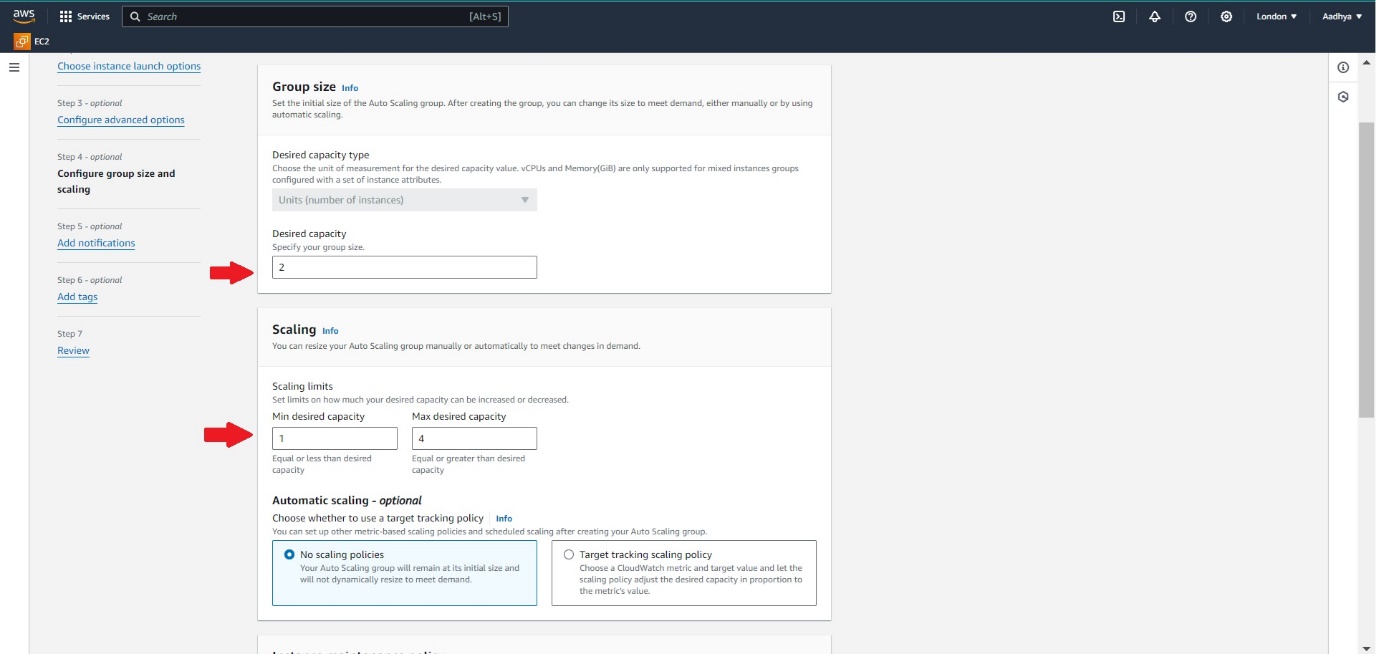
1. In Instance type select t2.micro. Select the available keypair or you can also create new. 
2. In Network settings, don’t touch any subnet configuration. Select Create security group and give the sg name as AWS-Prod-2-Tier-SG, description as Allow SSH access. Select the VPC(AWS-Prod-2-Tier) which we created earlier.

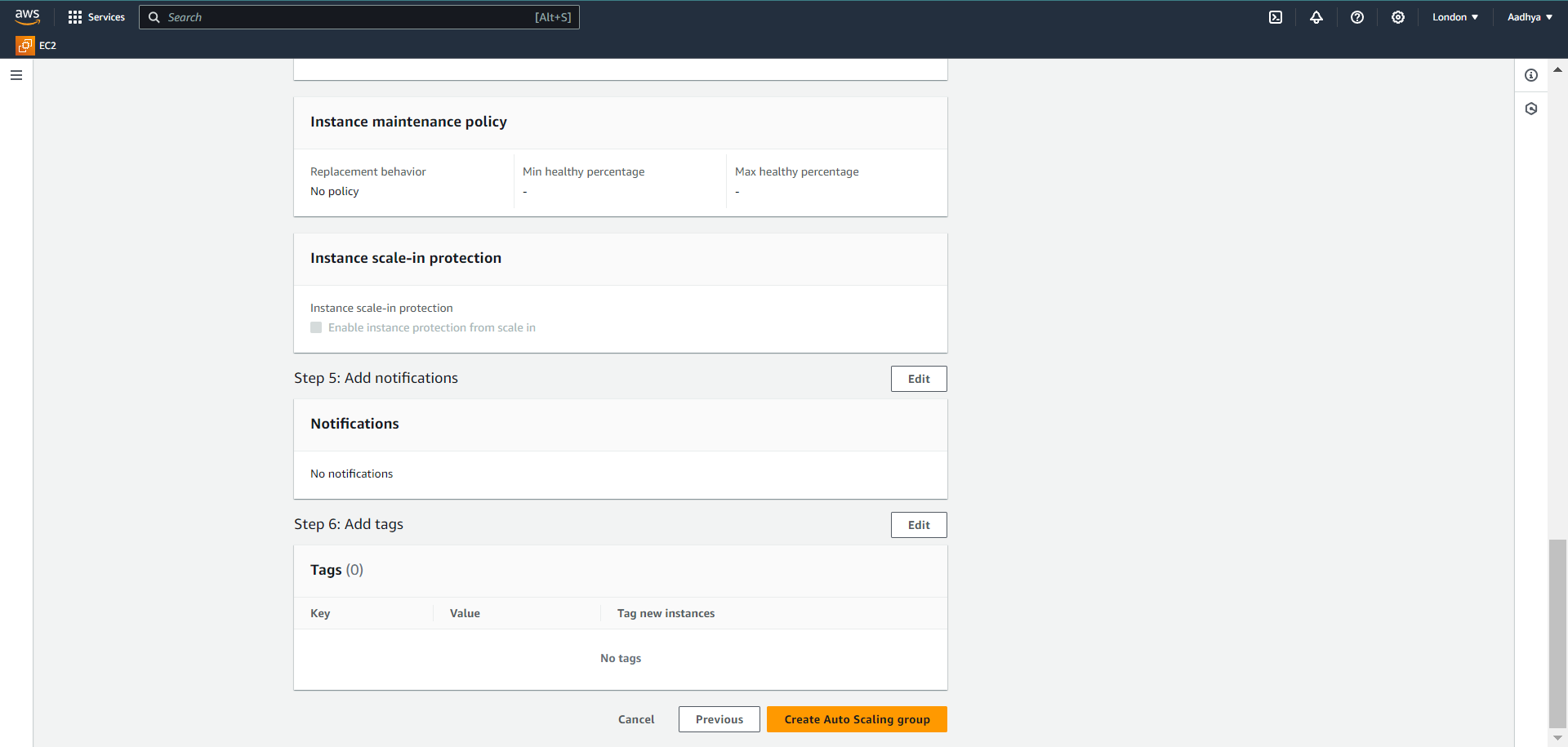


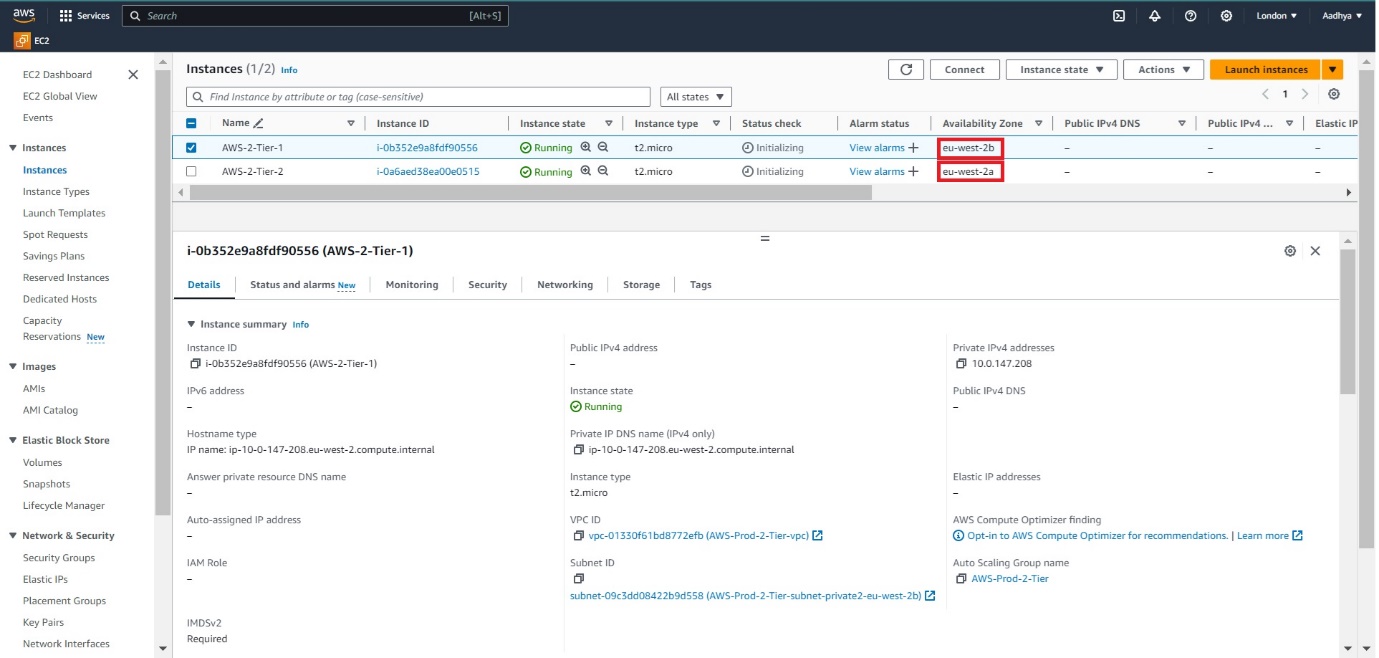
1. In Inbound Security Group Rules, add SSH(from anywhere) and custom TCP with port range 8000(from anywhere). Leave all other configurations untouched and click on Create launch template.
2. Now go back to ASG page and refresh once so that you can see the created template in the options. Give the Auto Scaling group name - AWS-Prod-2-Tier and then click on Next.



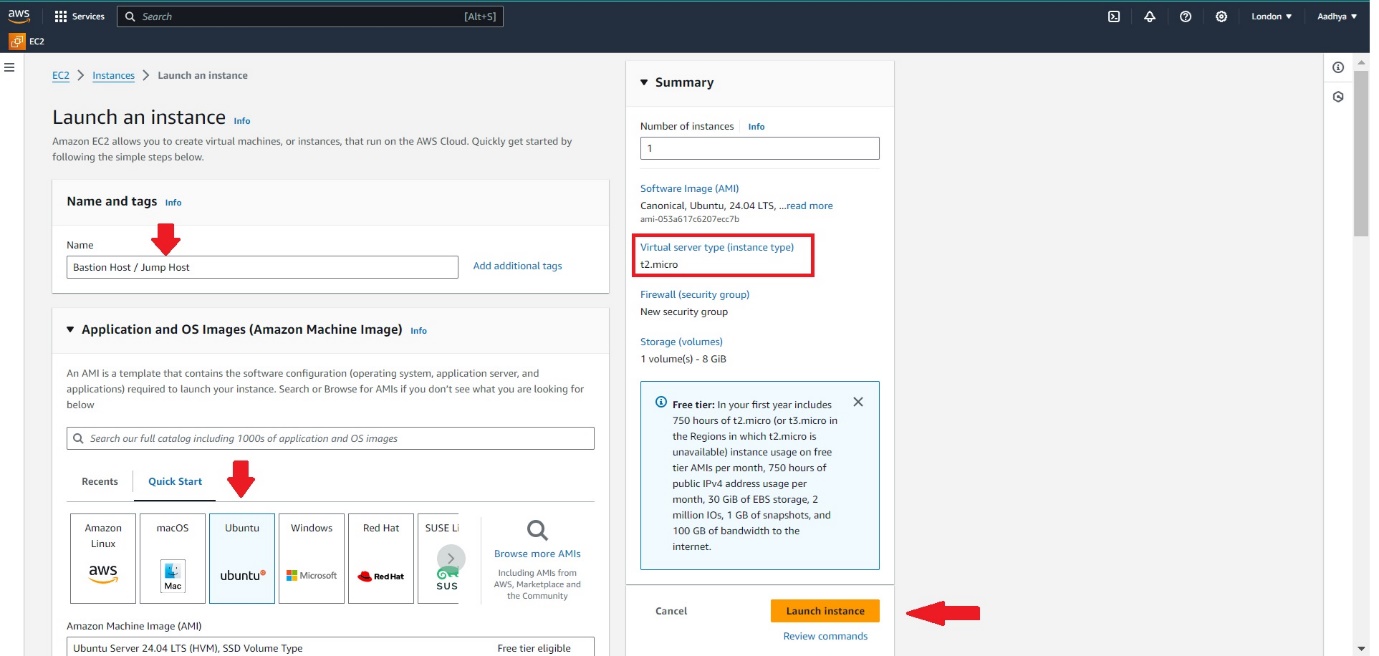
1. In Network, select the VPC created and in Availability Zones and subnets – select the 2 available private subnets. Now click on Next.
2. Configure advanced options – optional page don’t change any. Leave as it is and click on Next.
3. Give 2 in Desired capacity and in Scaling Min as 1 and maximum as 4. Don’t touch automatic scaling configurations. Click on Next.

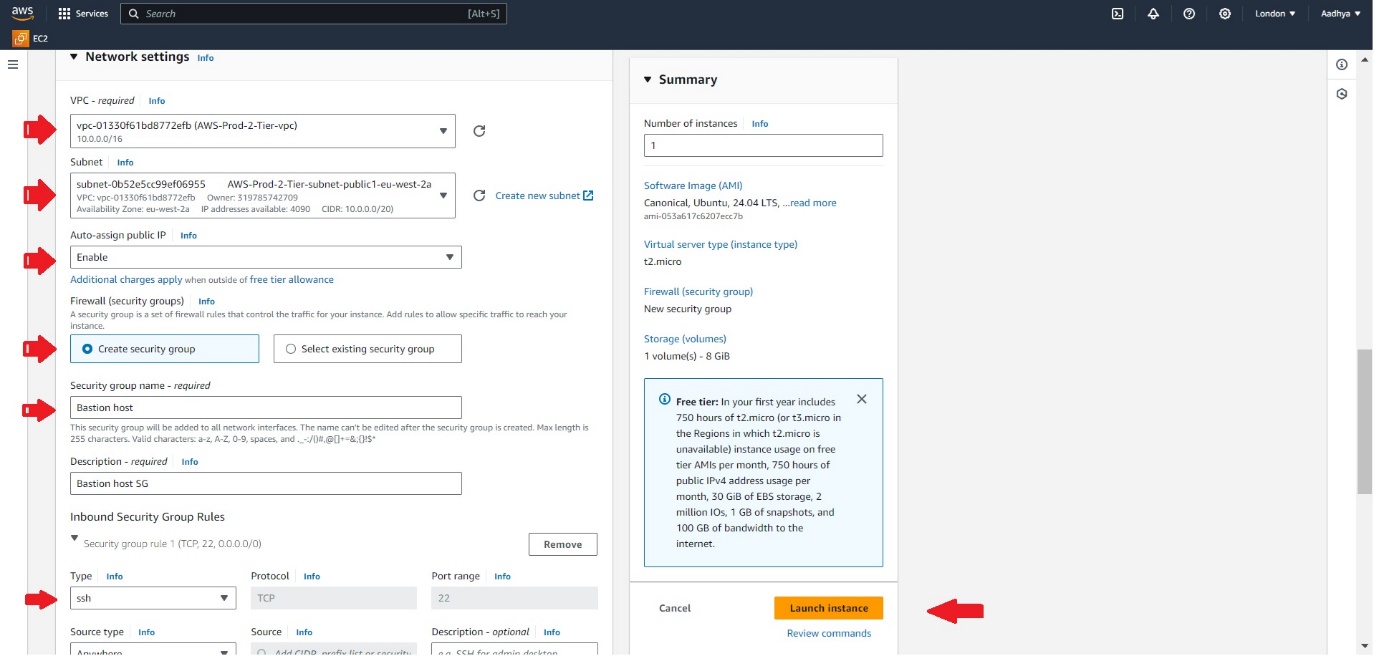


1. Don’t touch any other options and directly click on Create Auto Scaling group.
2. Make sure if the instances are created and also see if it is created in two different AZ’s.



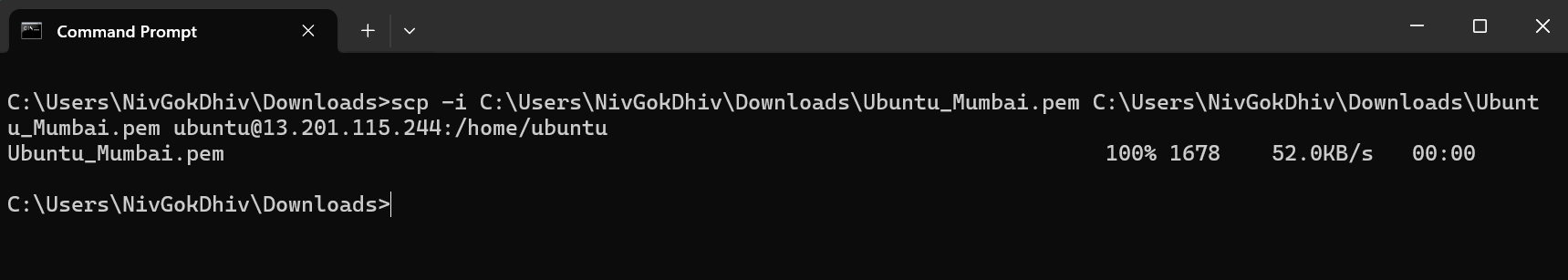
Now to login into the private instances we need a bastion host.

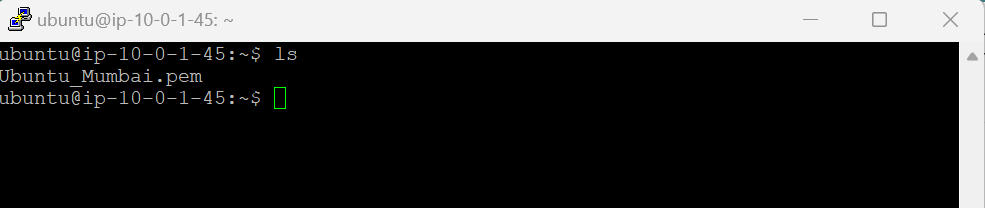
1. Go to EC2 Dashboard, click on Launch Instance. Give the name as Bastion Host / Jump Host, select ubuntu AMI, instance type t2.micro and provide keypair.
2. In network settings, select the created VPC, make sure it is in public subnet, Auto-assign public IP – Enable, Create security group and make sure ssh is added. Now click on Launch instance.



1. Now open CMD and type the below command to

scp -i C:\Users\NivGokDhiv\Downloads\Ubuntu\_Mumbai.pem C:\Users\NivGokDhiv\Downloads\Ubuntu\_Mumbai.pem ubuntu@13.201.115.244:/home/ubuntu





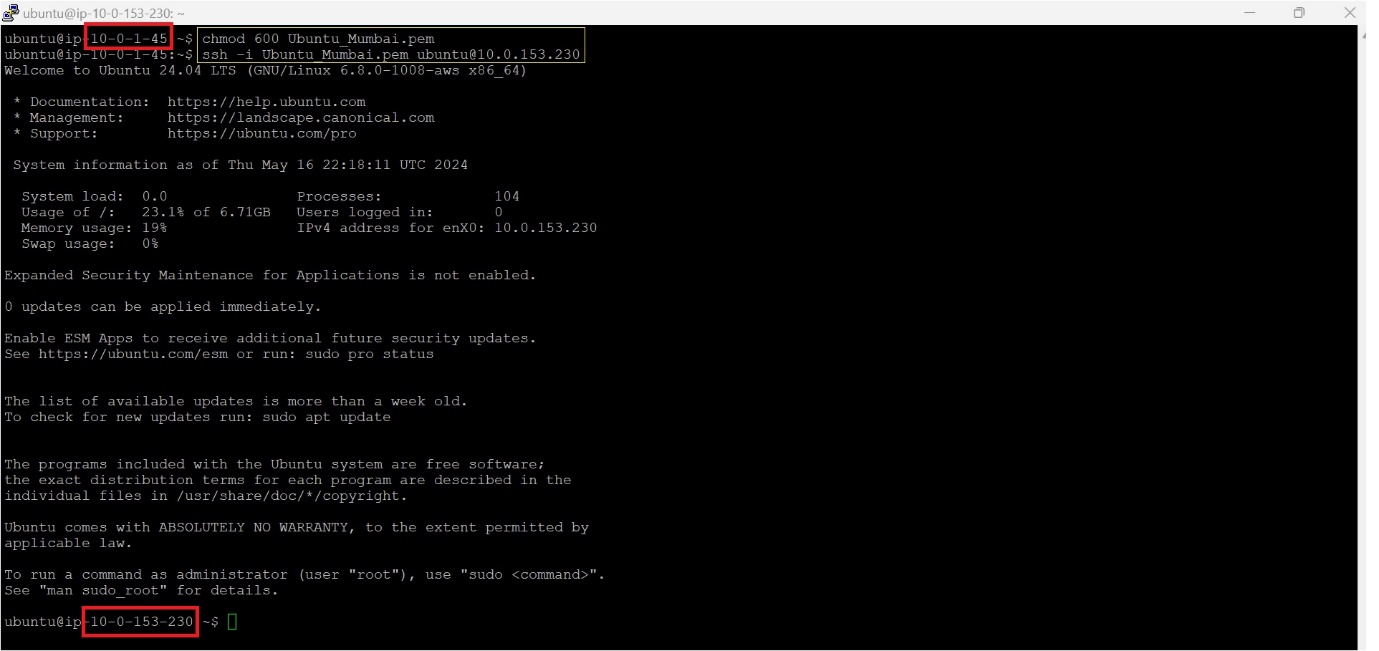
1. Change the permission of the pem file into 600 using the below command.

**Chmod 600** **Ubuntu\_Mumbai.pem**

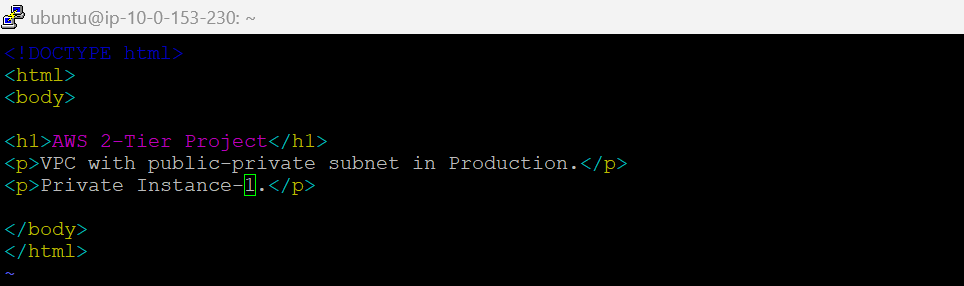
1. Now to login into the private instance, copy any one of the private instance IP- 10.0.153.230 and do ssh from bastion host with the below command via putty.

**ssh -I Ubuntu\_Mumbai.pem** [**ubuntu@10.0.153.230**](mailto:ubuntu@10.0.153.230)

1. Now you can notice that the IP has been changed from Bastion to private IP as shown in the below screenshot.

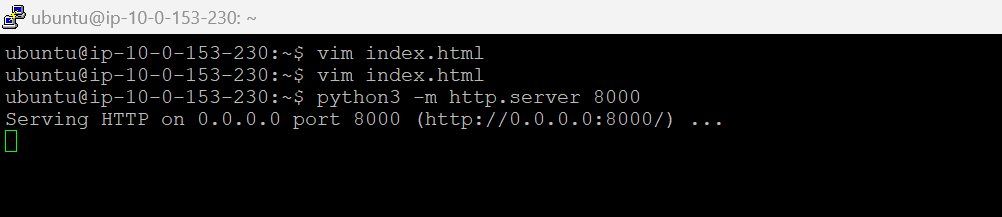


1. Now once u logged inside the private IP, create a index.html file using the below command and give the contents of HTML inside it.

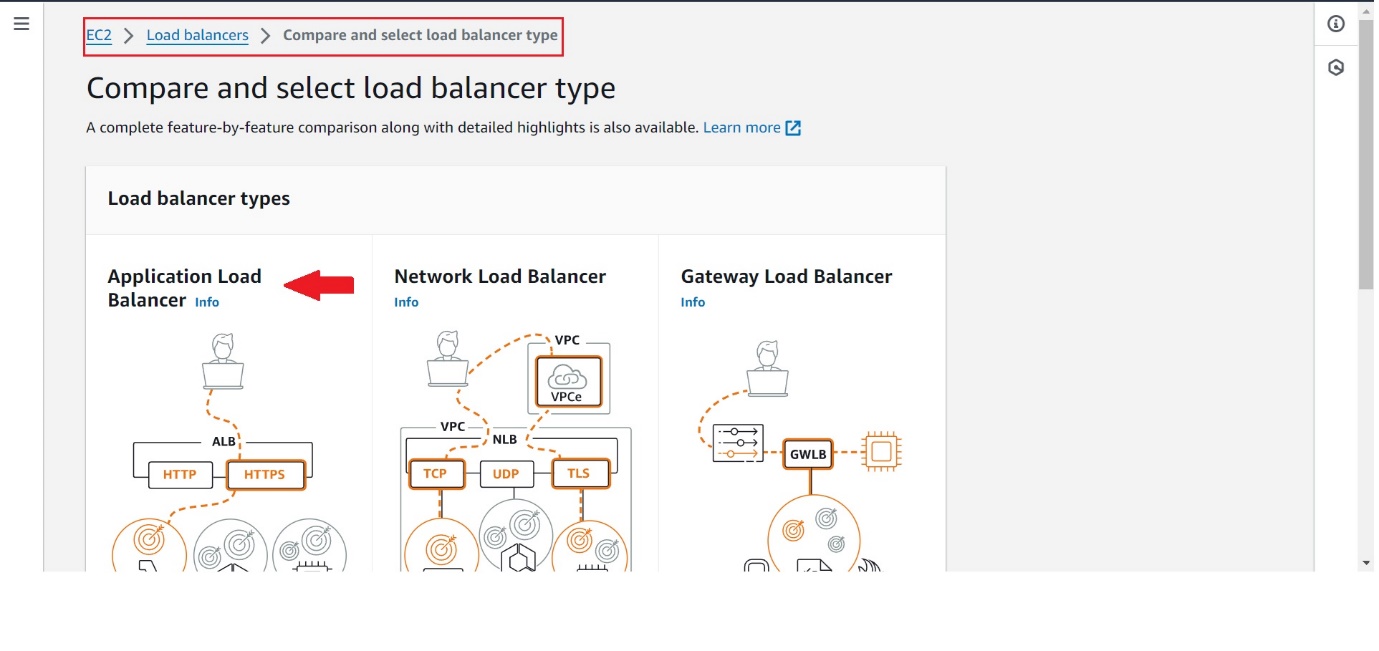


1. Now install the python application and run port 8000 in it using the below command.

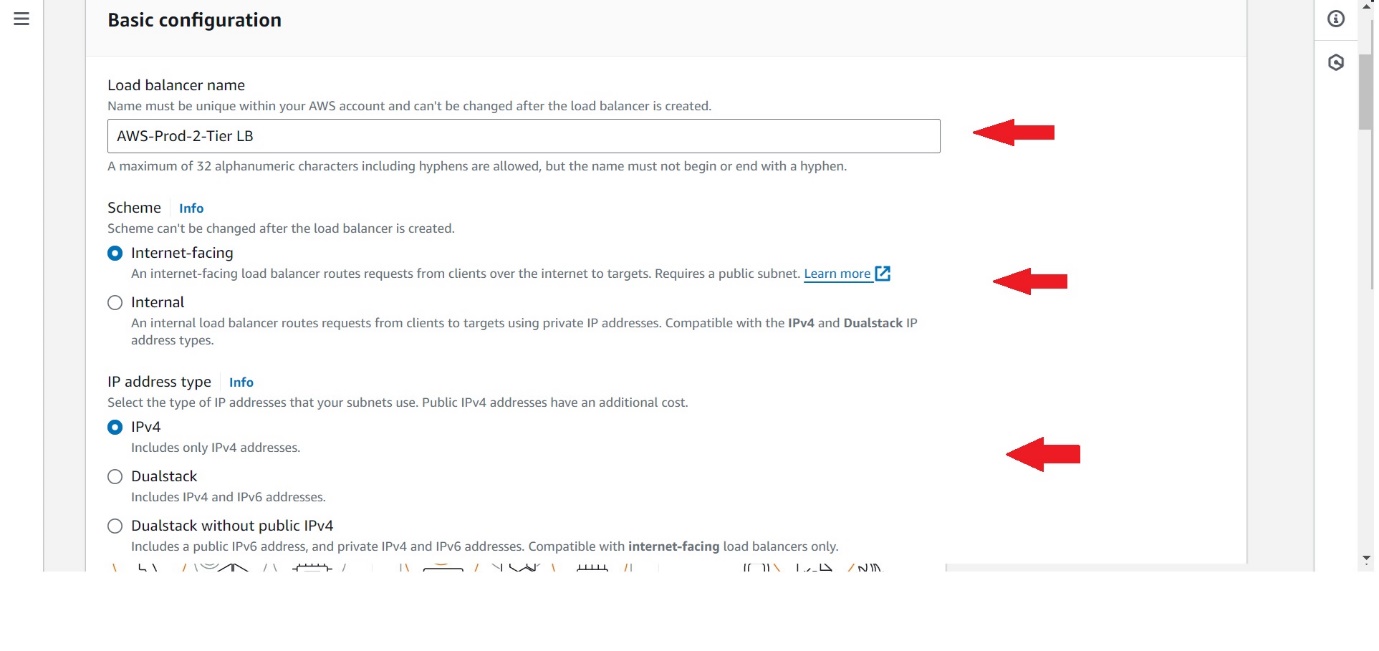
**python3 -m http.server 8000**



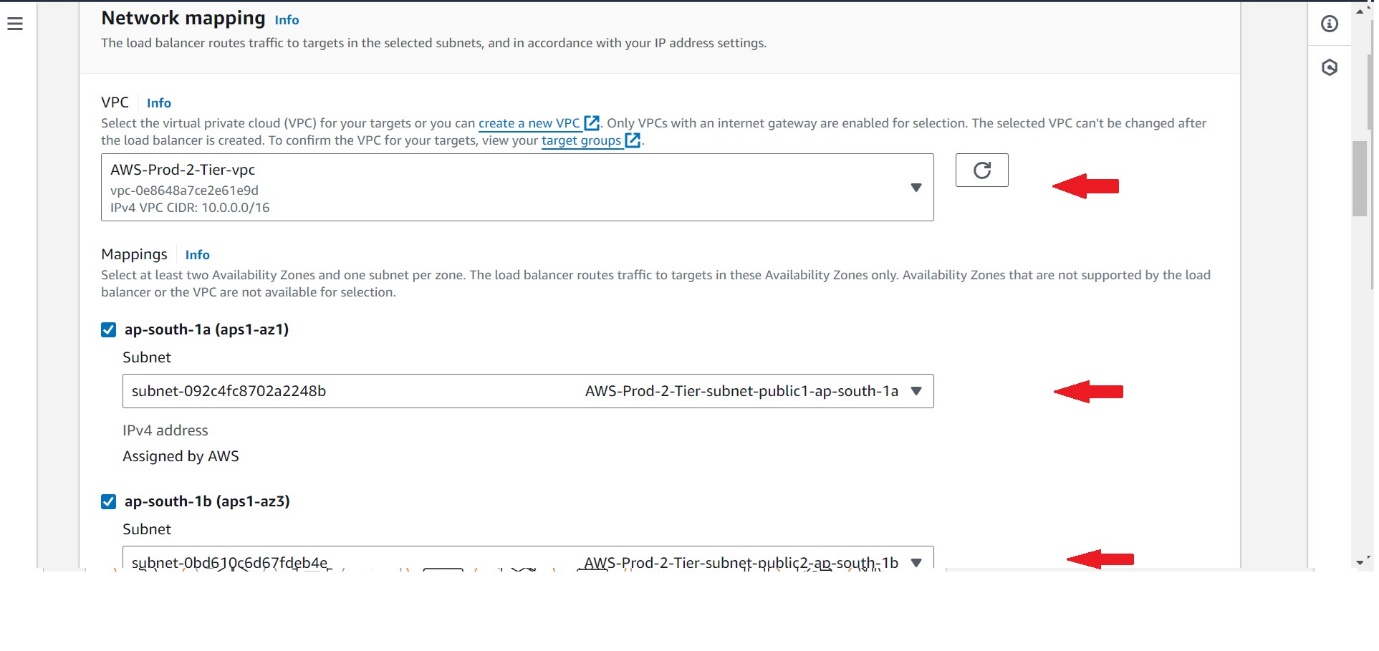
1. Now to balance the load between two private instances now we have to create Load Balancers. Go to EC2 Dashboard, select Load balancers from the left panel, select Application Load Balancer and click on Next.



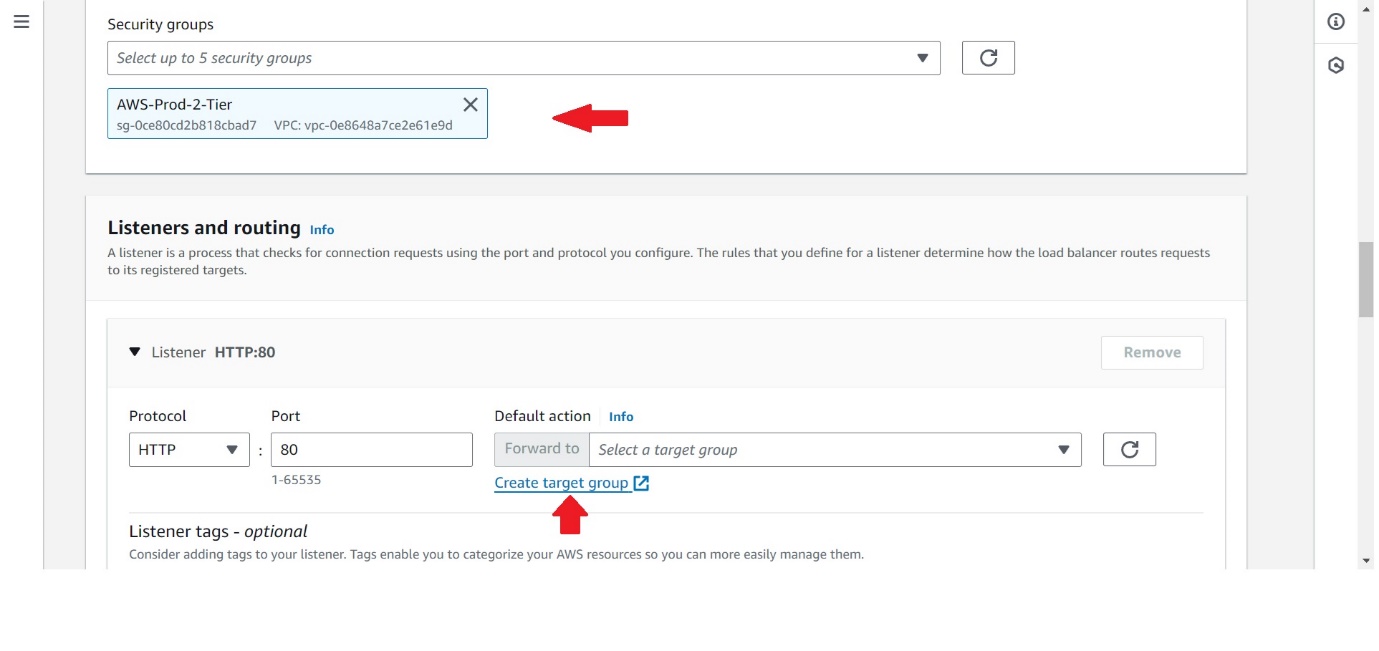
1. Give the Load balancer name AWS-Prod-2Tier-LB, keep the scheme as internet-facing as LB should be always public. IP address type is IPv4.



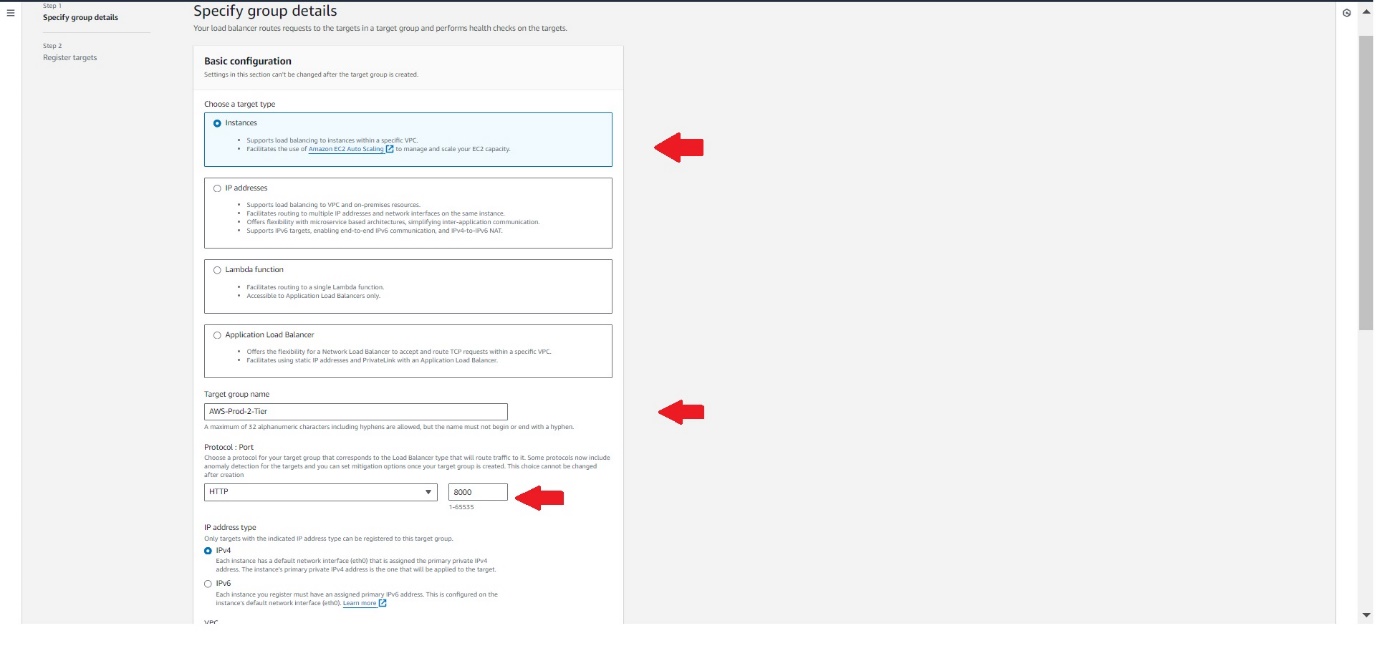
1. In Network Mapping, select the created VPC. Select the available mappings and make sure both mappins are in public subnets , if not change it public subnet from the dropdown.



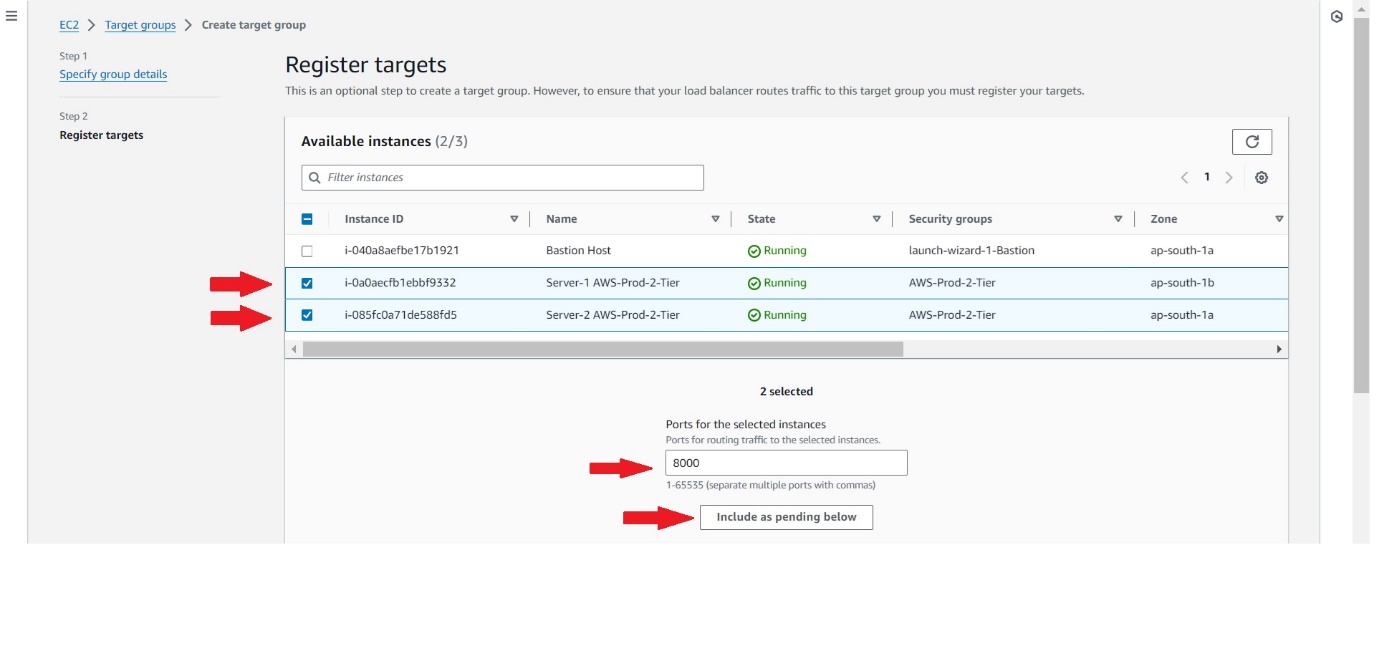
1. Select the VPC security group. In Listners and Routing, Click on Create target group.



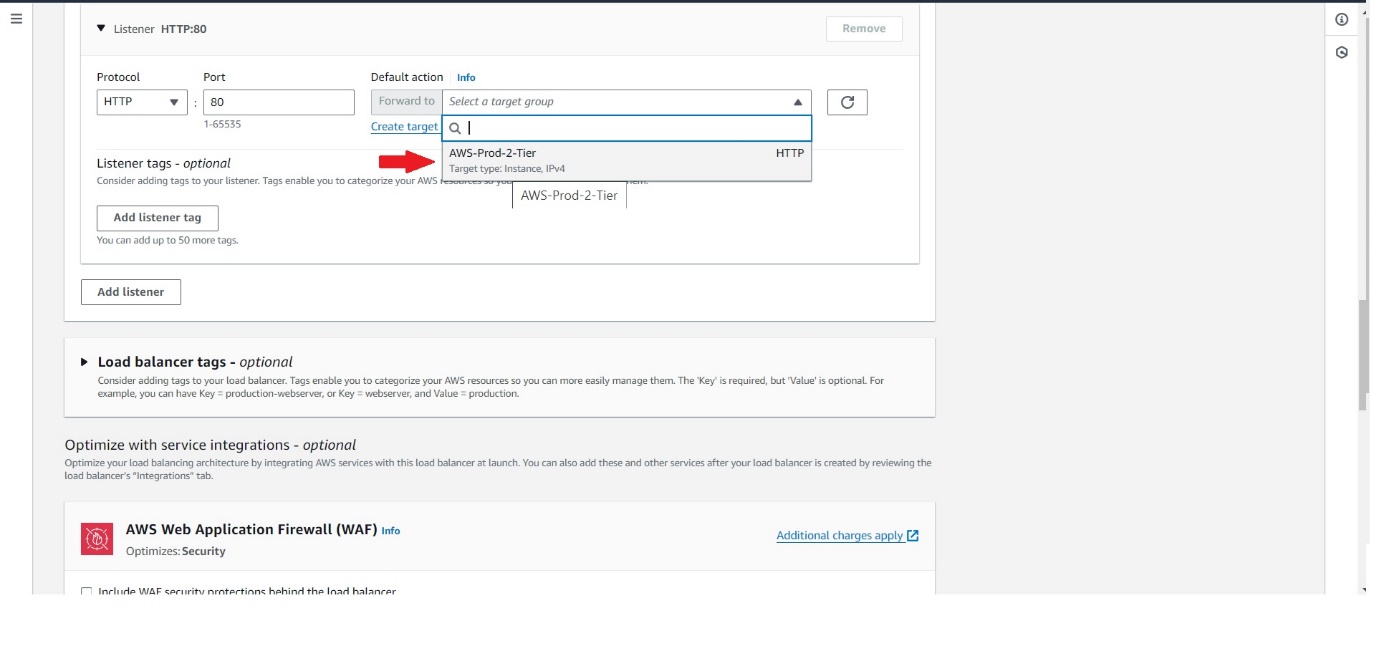
1. In the target group page, specify the basic configuration as Instances, in Protocol-Port select HTTP and give 8000 in it.



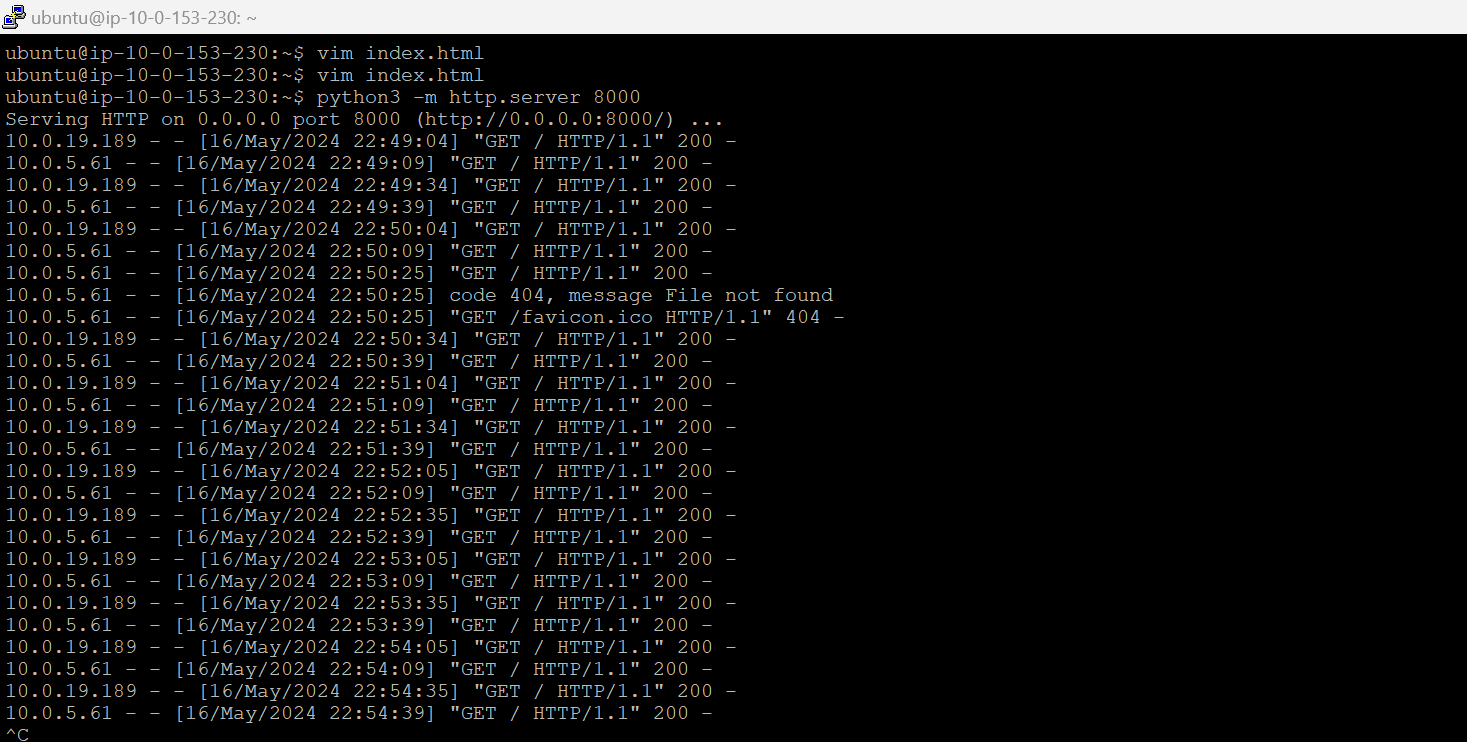
1. Select the two instances other than bastion host and click on include as pending below. In the ports for the selected instances give port number as 8000.

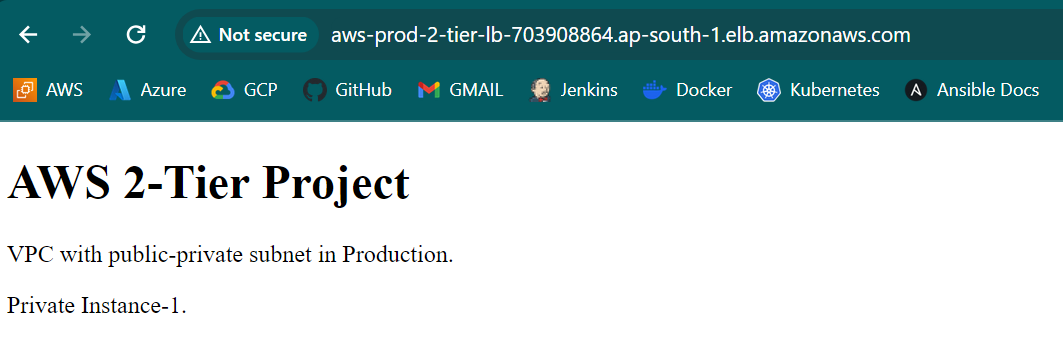


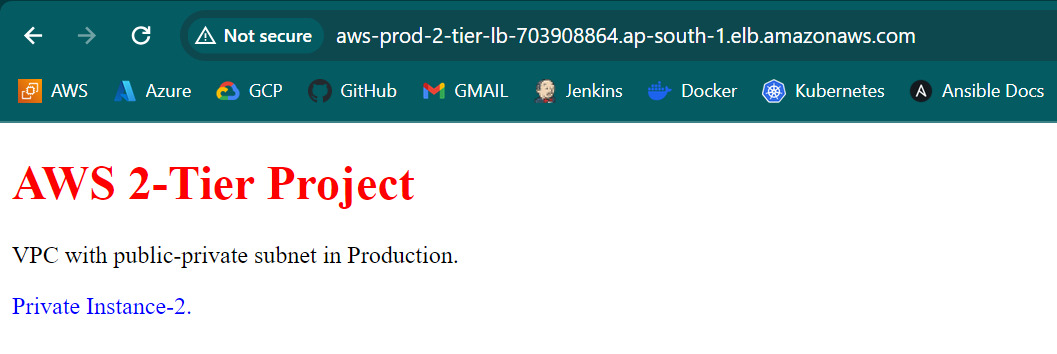
1. Now to LB page again refresh it and select the created target group, and create the Load balancer directly without changing anyother options.



1. Now once the LB is active copy the DNS name and paste it in web page now you can see the index.html files. Now once you see the output repeat the same in server 2 also.







**Deletion of Resources:**

1. Delete the LB(along with the created Target group)
2. Delete the ASG(along with the created Template)
3. Terminate all 3 instances.
4. Now coming to delete the VPC follow the below steps-
5. NAT Gateway
6. Then directly delete the VPC(It will automaticall delete the Subnets, SG and SG)
7. Don’t forget to release the Elastic IP.