VPC Project

Accessing a simple Python application from private subnet

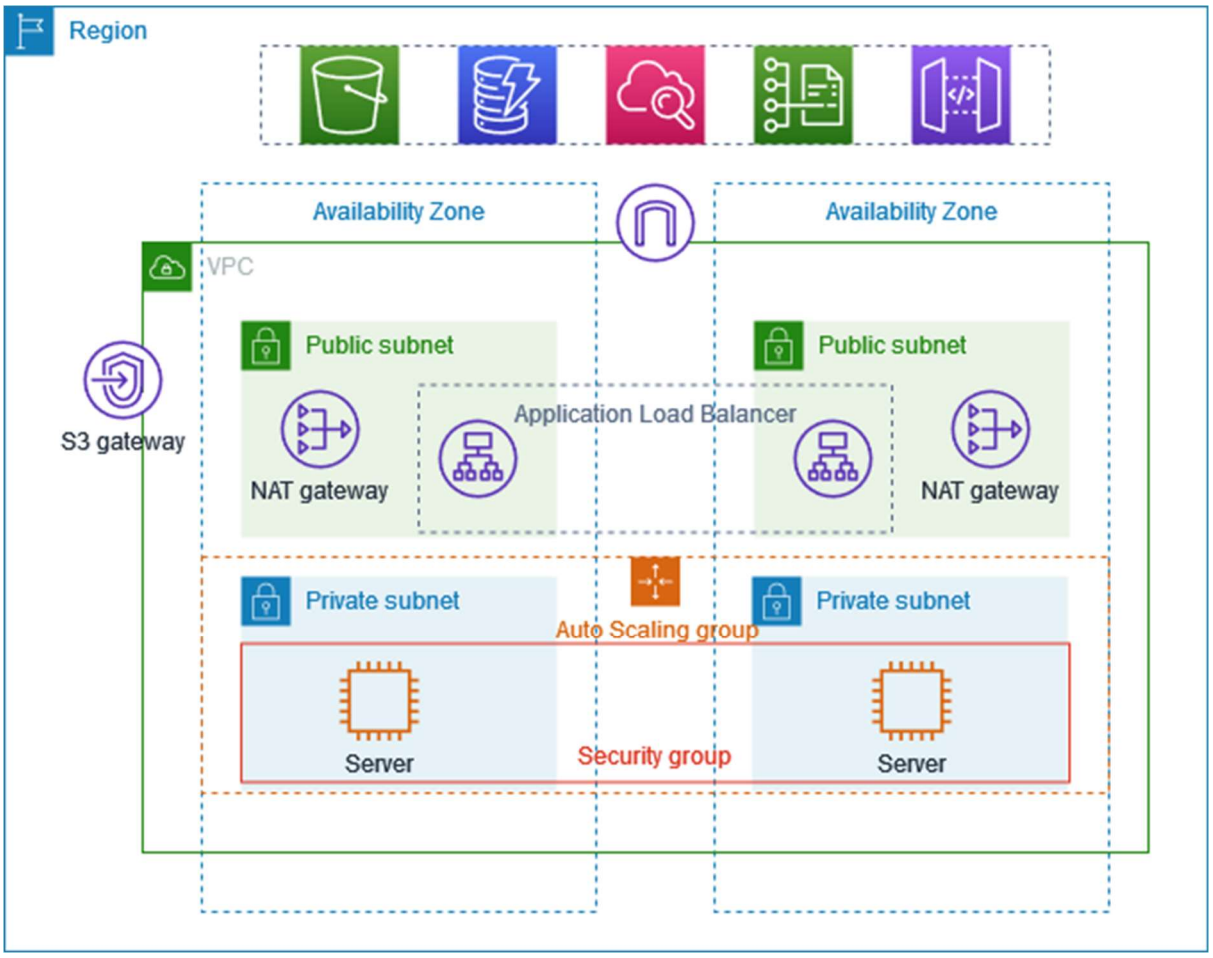
By: Shivam Joshi

https://www.linkedin.com/in/shivamgjoshi

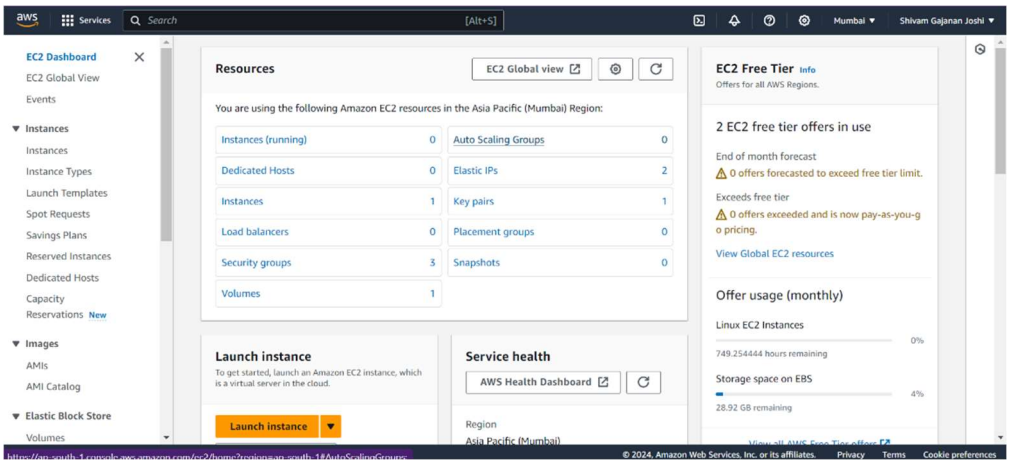
Services Used:

- Application Load Balancer
- Autoscaling groups
- VPC
- EC2

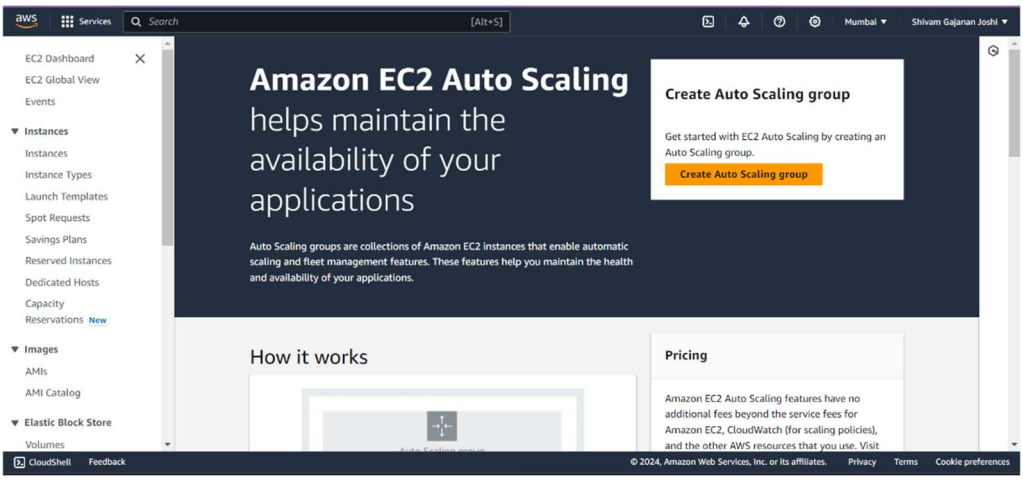
Architecture



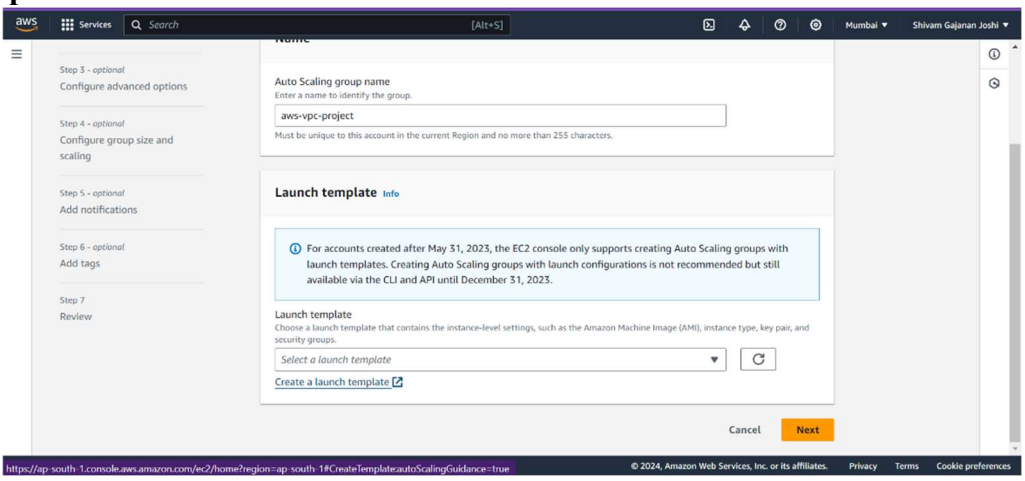
1. Go to AWS console, search EC2. Click on **Auto scaling groups**



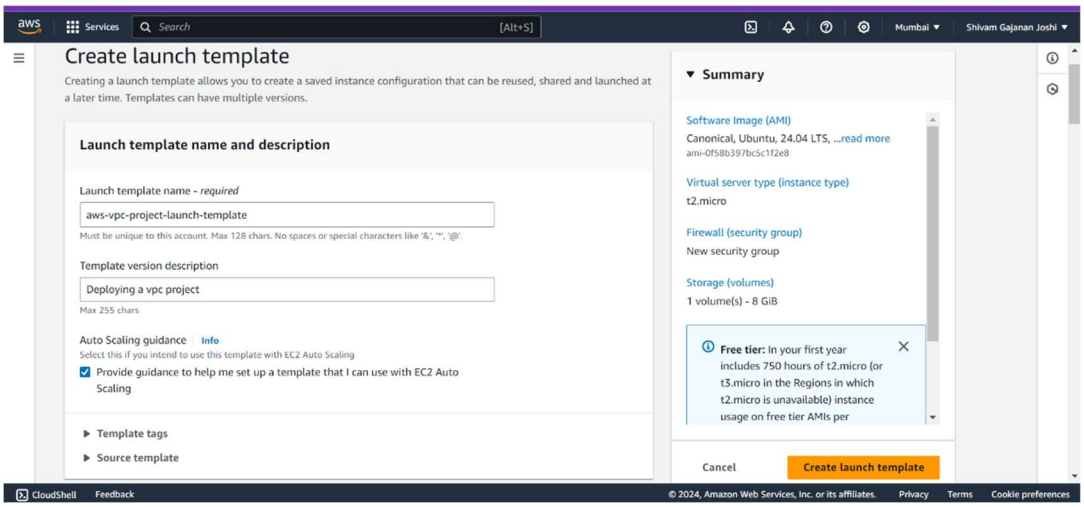
2. **Create Auto Scaling group**



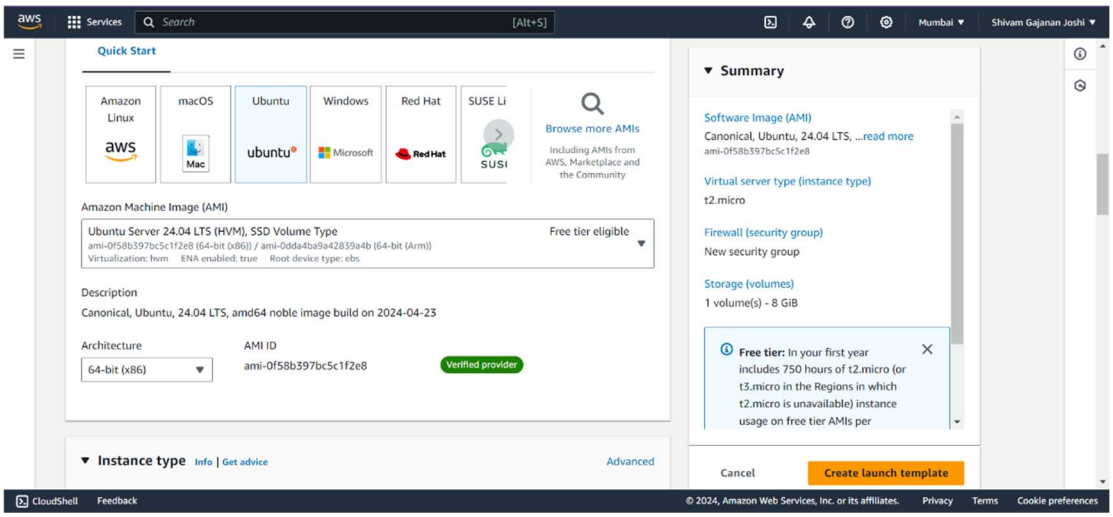
3. We need to specify the EC2 configuration for autoscaling group. Click on **Launch template**



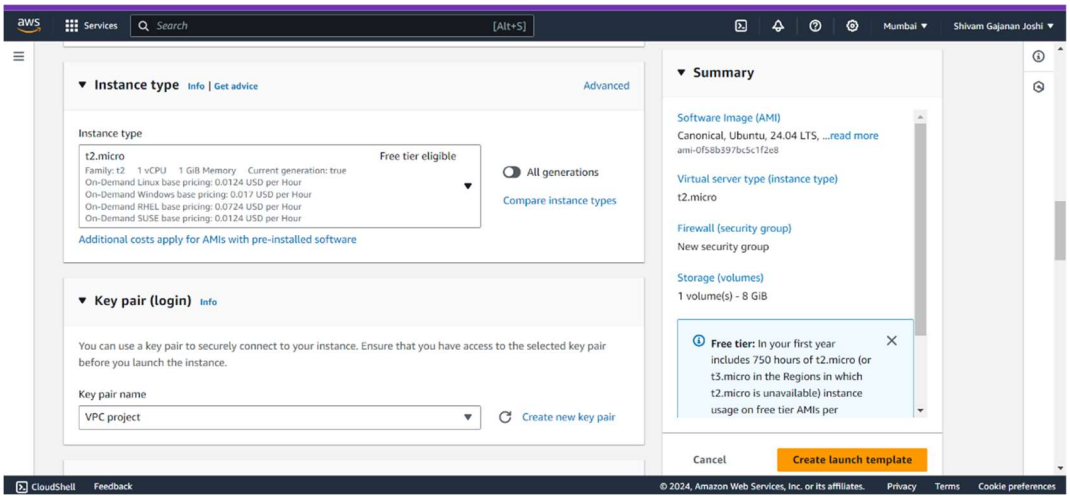
4. Give name for the template and write short description.



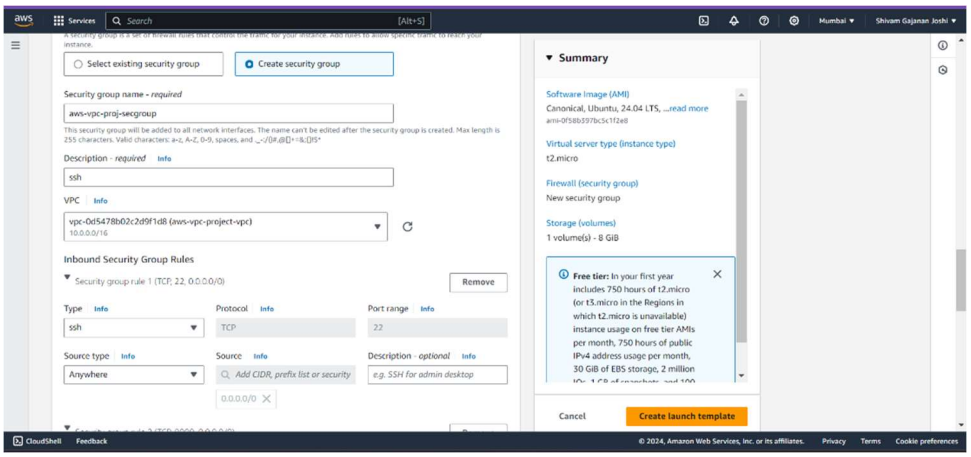
5. Choose AMI as **Ubuntu**.



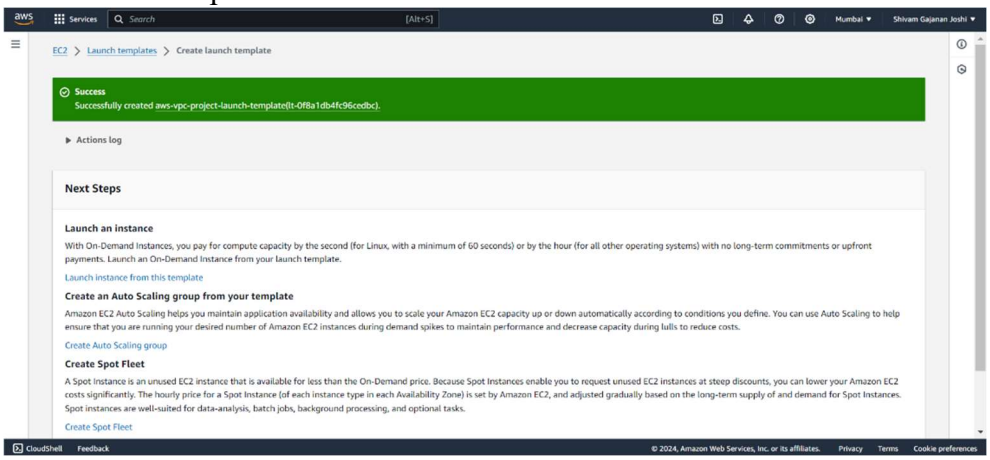
6. Go with free tier configurations. Choose the key pair name, or create on.



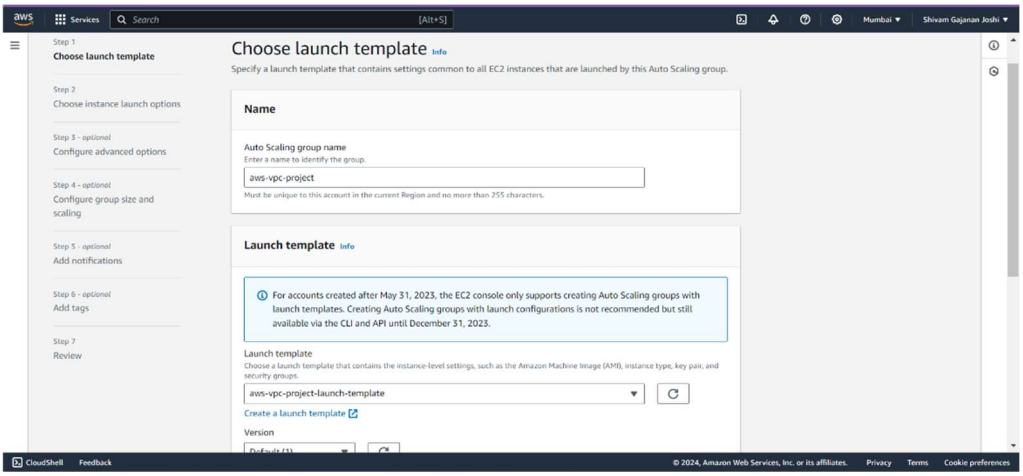
7. Click on **create security group,** and **select the VPC created for this project.** Click on edit **inbound rules, add SSH rule.**



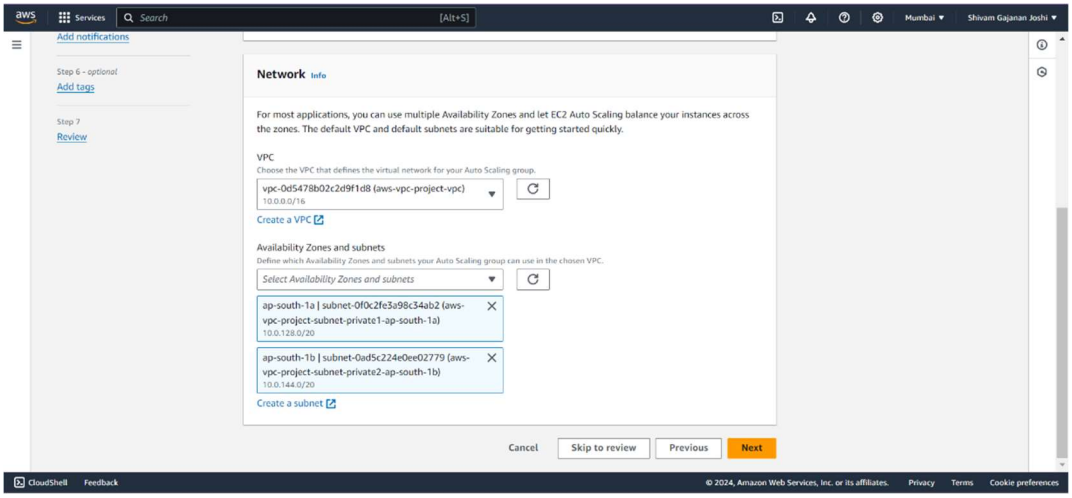
8. Click on Launch template.



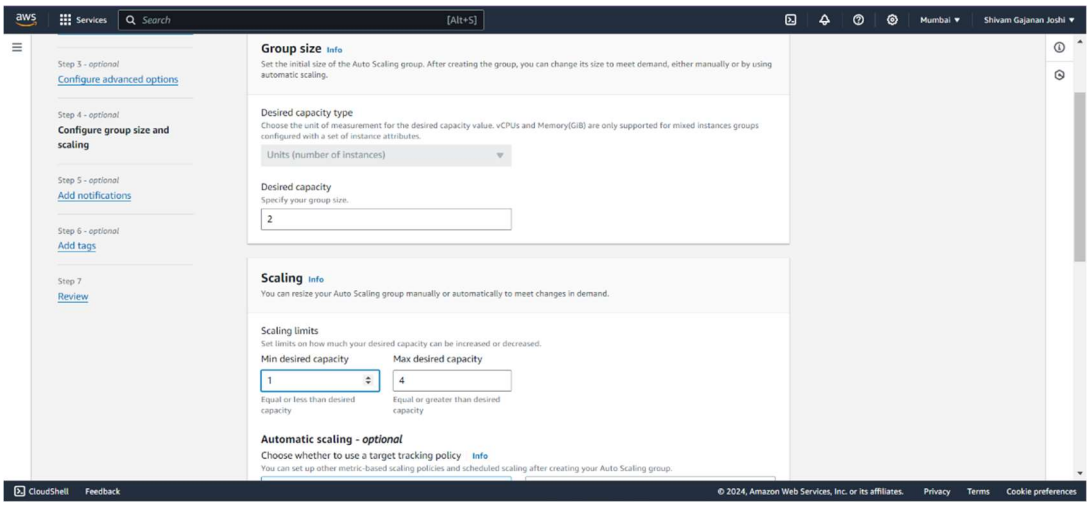
9. Once it is created go to **autoscaling groups, and select the recently created template.**



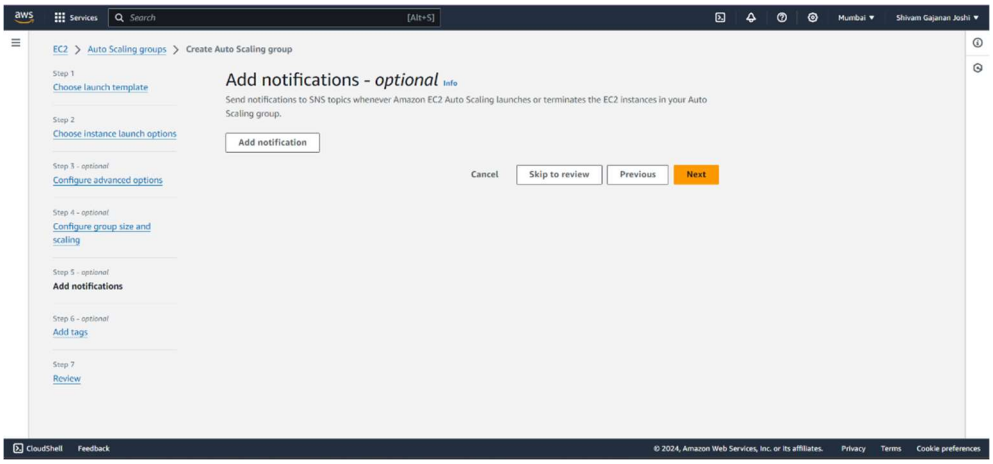
10. Under networking section, select VPC of this project and **select the 2 private subnets.**



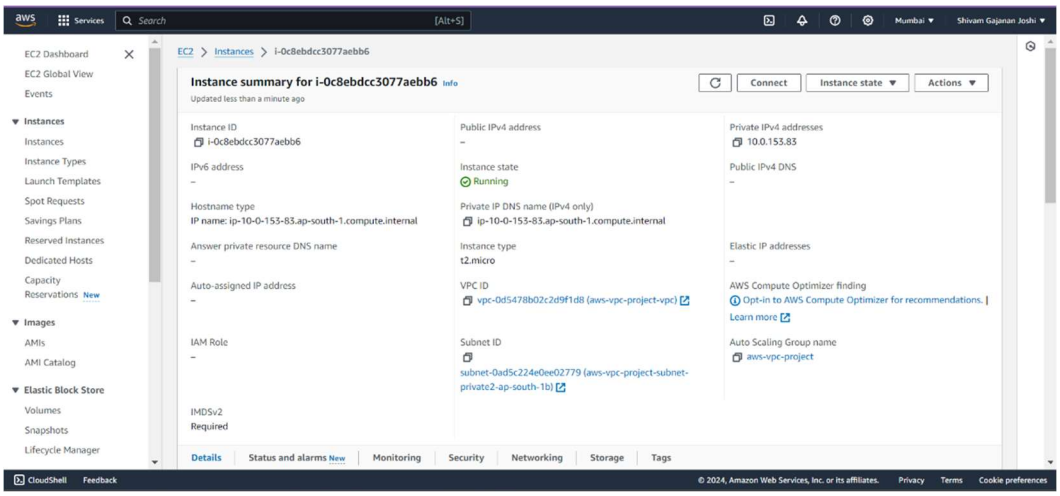
11. Under Group size, select the no. of instance you need in the start, and max capacity instance. These will reach its said limit when the traffic will be more.



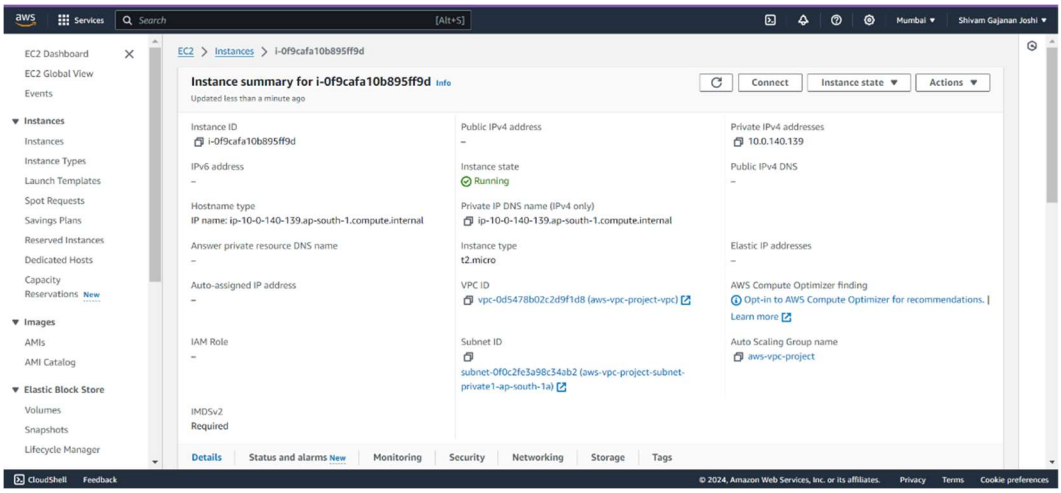
12. Go with default configurations in notifications section. Review and create.



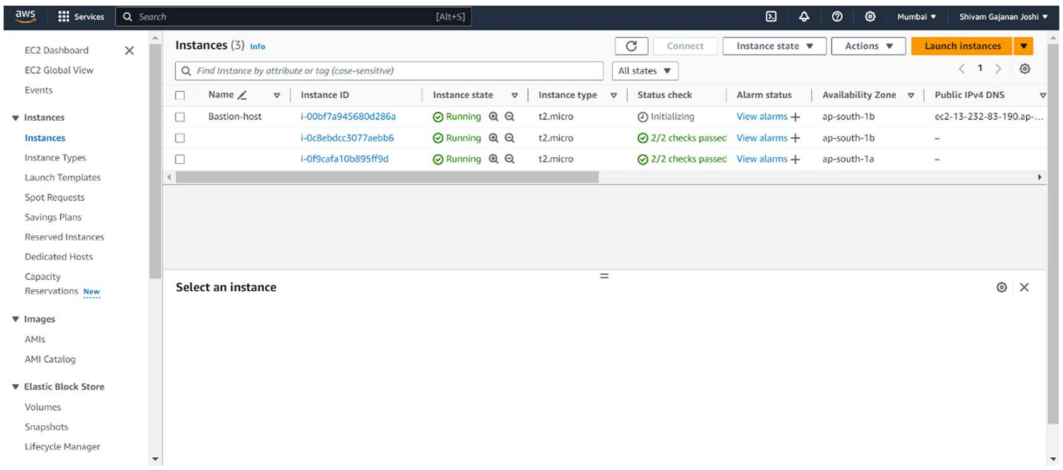
13. Check both the instances and their subnets. This instance is in south1b.



14. Check both the instances and their subnets. This instance is in south1a.



15. Create a Bastion host to access the autoscaling instances, **since they don't have a public IP address, we use a Bastion host machine.** The procedure is same as we follow in creating an EC2 instance, just a mior change in the **networking setting**, while making the configurations select the **VPC in which both the Autoscaling instances are deployed, and enable auto assign IP address option**.

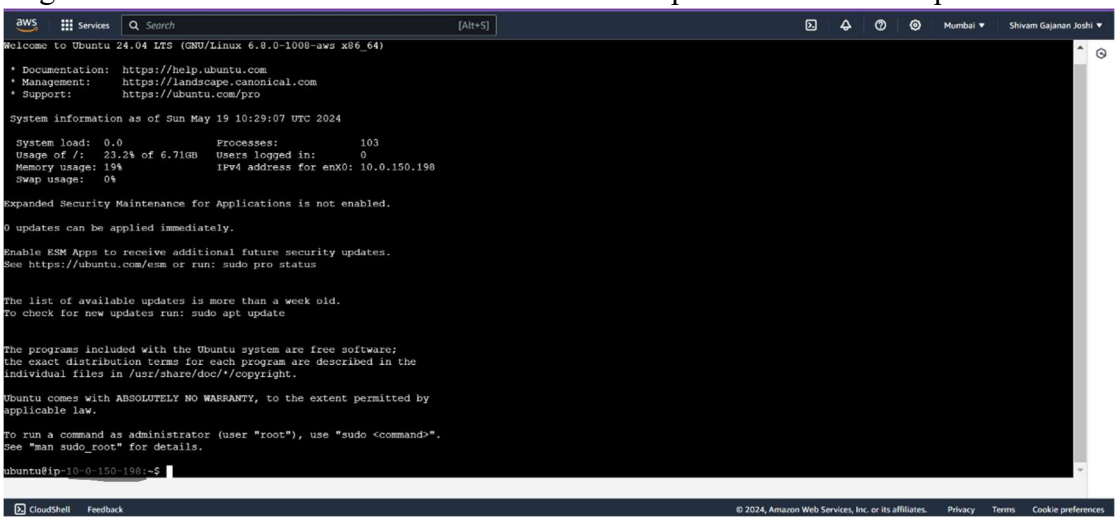


16. Now we need to copy the .pem file from out local machine to our Bastion host machine. For that open the **bash terminal(for windows user)** and type in the below command. **The path of .pem may vary, change it accordingly.**
$ scp -I <path of .pem file> ubuntu@<publicip of Bastion>:/home/ubuntu

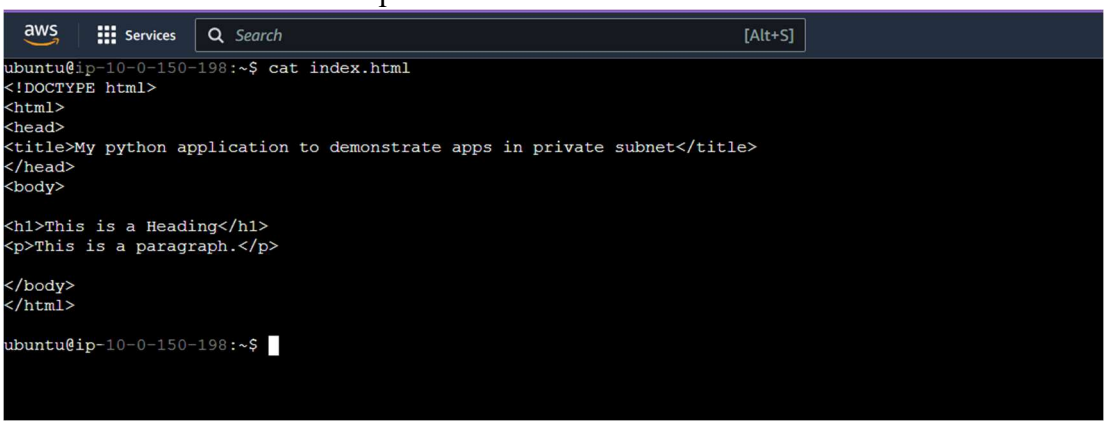
```
HP@SHIVAM MINGW64 ~
$ scp -i "C:\Users\HP\Downloads\awsprodegkeypair.pem" "C:\Users\HP\Downloads\awsprodegkeypair.pem" ubuntu@65.2.130.177:/home/ubuntu
C:\Users\HP\Downloads\awsprodegkeypair.pem                                    100% 1678    49.6KB/s   00:00

HP@SHIVAM MINGW64 ~
$
```

17. Login to the Bastion host machine and check if the .pem file has been copied or not.

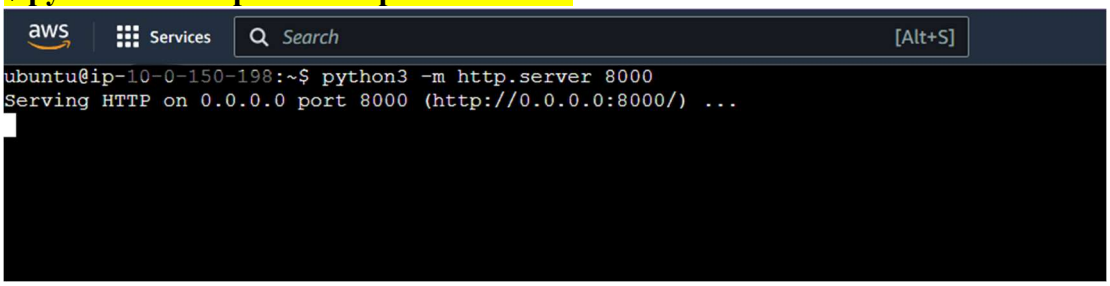


18. Now we need to create a sample html file.

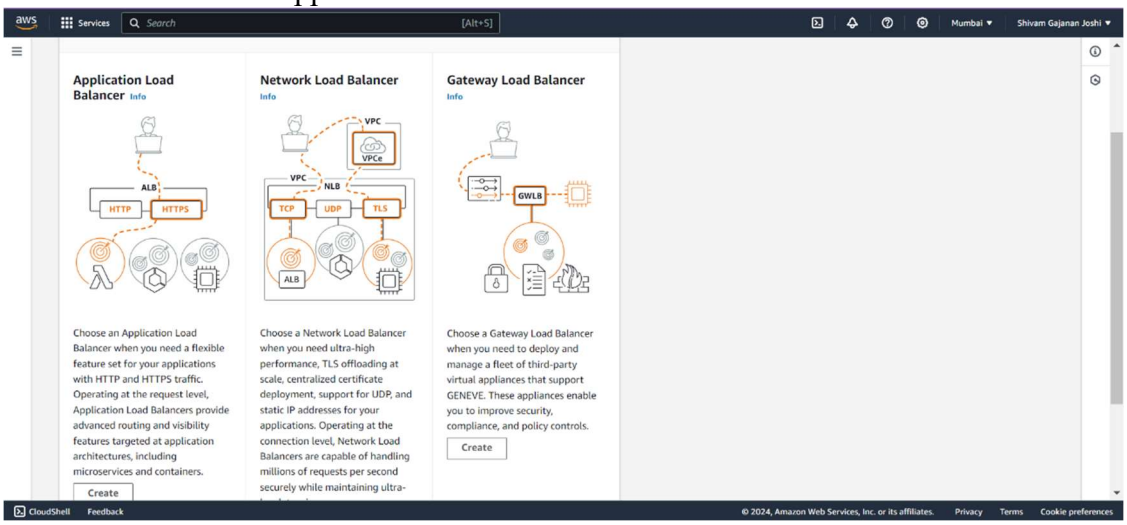


19. Run the application using above command

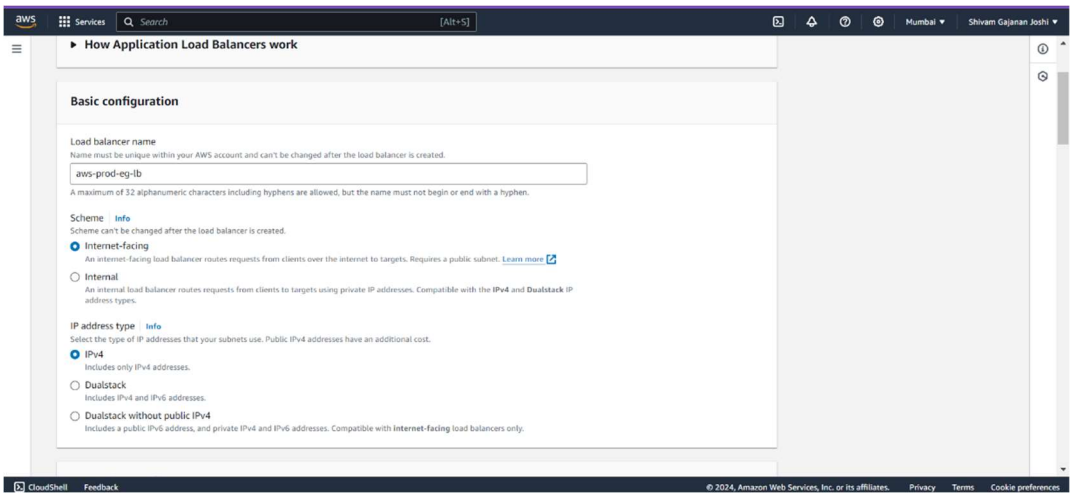
$ python3 -m http.service <port number>



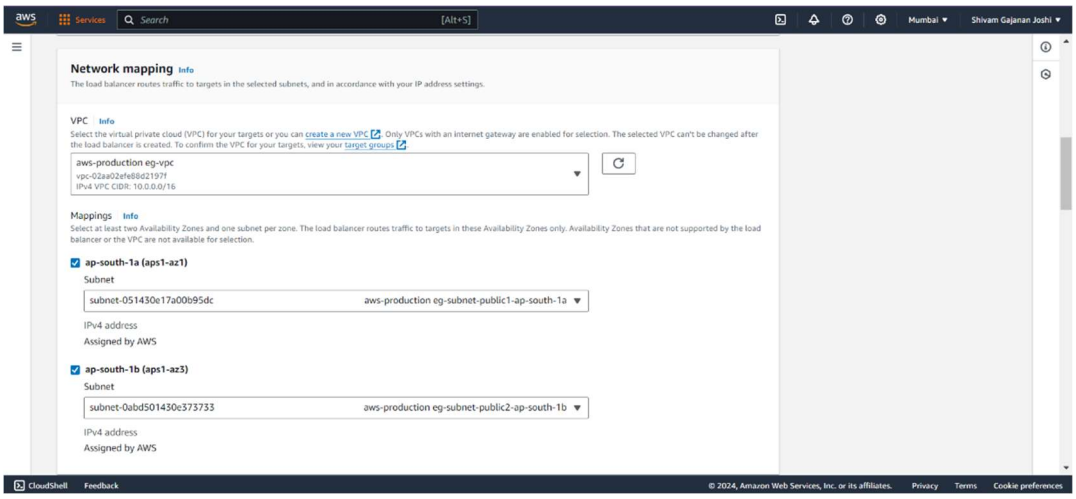
20. It's time to create an Application Load Balancer



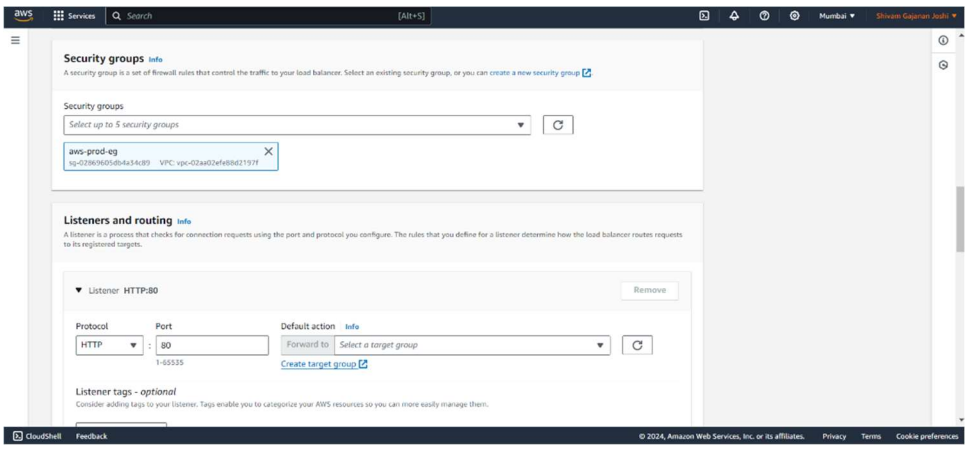
21. Follow the below steps



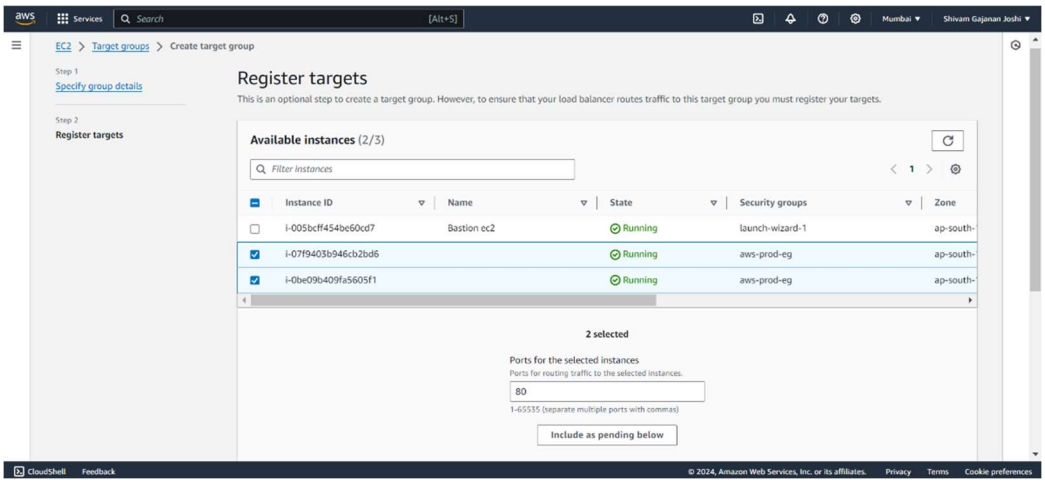
22. Select the VPC created for this project.



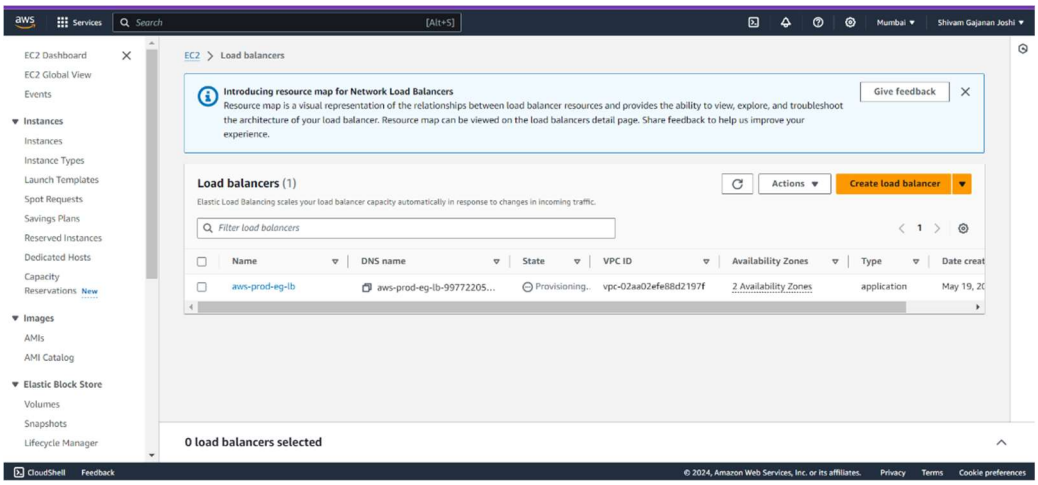
23. In the security groups, add rule to allow traffic via port 8000 (port mentioned in previous step).



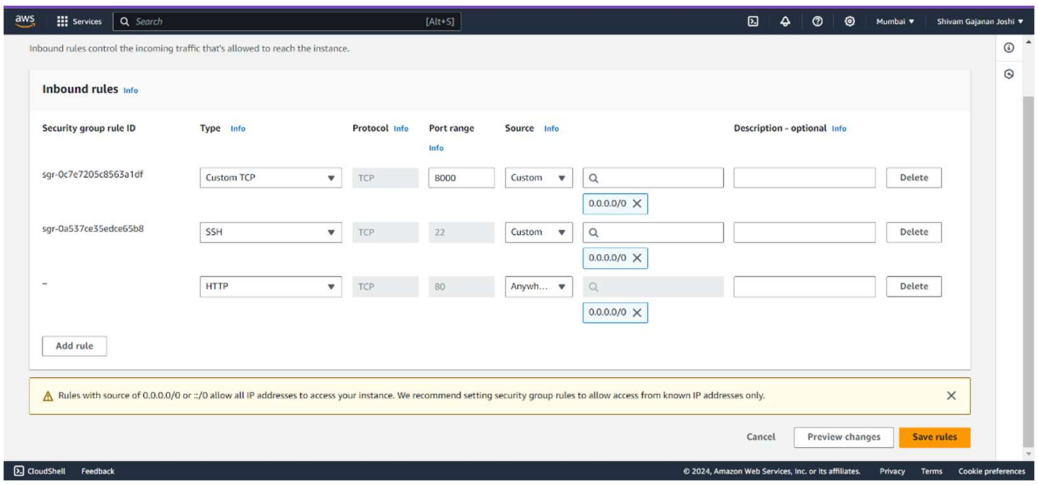
24. Now add the autoscaling machines into the target group and click **Include as pending below.**



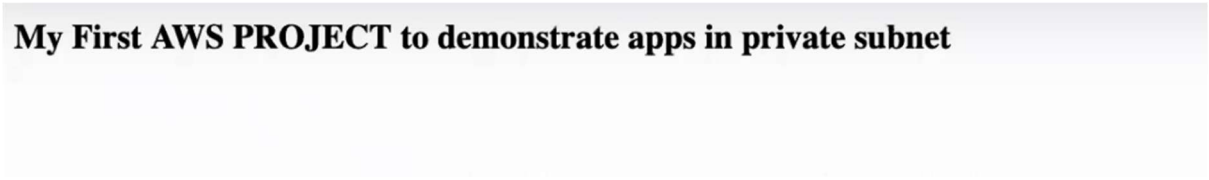
25. Create load balancer.



26. If you face issue with port number or access denied, change the security group rules.



27. Access the application using the LB DNS. It should display the output of HTML file.



NOTE: If you see 502 Bad Gateway, it is because aws shows the successful output of healthy instances.

Please clean up your resources to save cost.