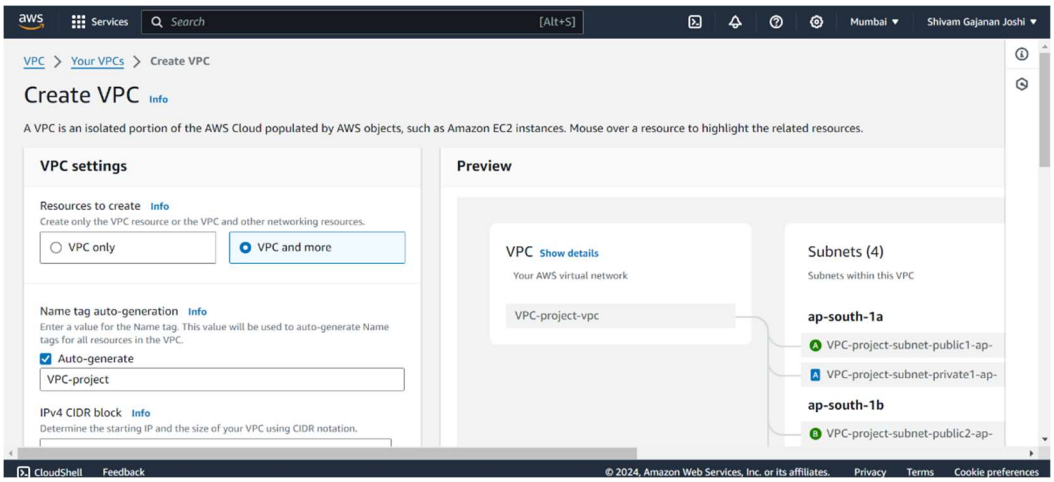
VPC HANDS-ON

By: shivam9joshi@gmail.com

www.linkedin.com/in/shivamgjoshi

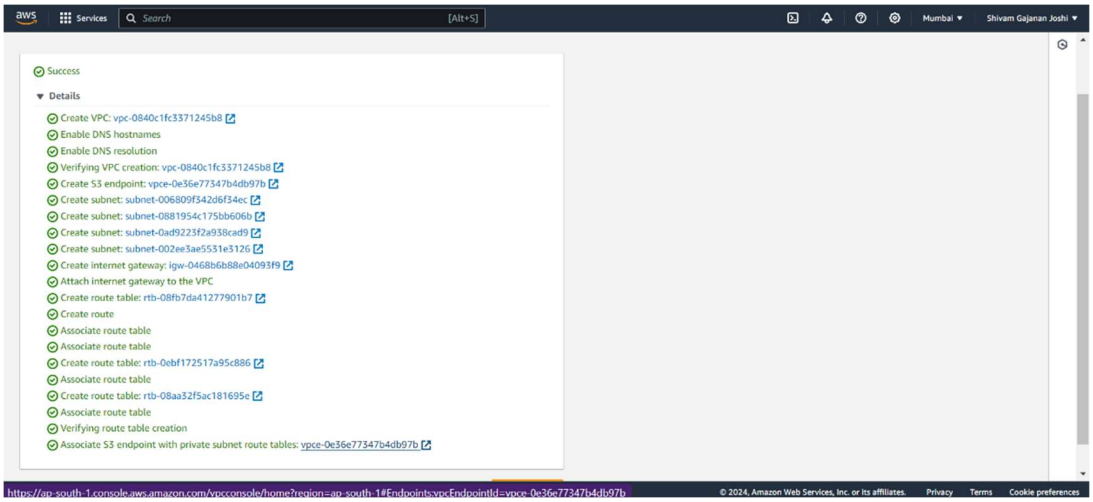
1. **Peering Connections:** VPC Peering Connection allows you to connect two VPCs and route traffic between them privately using private IP addresses. It enables resources in separate VPCs to communicate with each other as if they were within the same network. Peering connections do not require a separate gateway or physical hardware and can be established within the same AWS region or across different AWS regions.
2. **Traffic Mirroring:** Traffic Mirroring allows you to copy network traffic from Amazon EC2 instances, Elastic Network Interfaces (ENIs), or AWS Lambda functions and redirect it to monitoring and security tools for analysis. This feature is particularly useful for troubleshooting, network monitoring, intrusion detection, and compliance purposes. By mirroring network traffic, you can gain insights into the traffic patterns and potential security threats within your VPC.
3. **Transit Gateways:** Transit Gateways are a centralized hub that simplifies network architecture by allowing you to connect multiple VPCs, on-premises networks, and VPN connections. They act as a transit point for routing traffic between different networks, providing a scalable and efficient solution for managing connectivity across large and complex network environments. Transit Gateways support features such as route propagation, route tables, and VPN attachments, making it easier to manage and scale your network infrastructure.
4. **VPC Flow Logs:** VPC Flow Logs capture information about the IP traffic flowing to and from network interfaces within your VPC. This includes information such as source and destination IP addresses, ports, protocol, packet counts, and more. Flow logs can be used for various purposes, including network monitoring, troubleshooting, compliance auditing, and security analysis. By analyzing flow log data, you can gain insights into network traffic patterns, identify potential security threats, and troubleshoot connectivity issues within your VPC.
5. **VPN Connections:** AWS Virtual Private Network (VPN) allows you to establish secure connections between your VPCs and on-premises networks or remote locations using encrypted tunnels. VPN connections provide a secure and reliable way to extend your on-premises network to the AWS cloud, enabling seamless communication between resources located in different environments. AWS VPN supports various VPN protocols, including IPsec (Internet Protocol Security) and SSL/TLS (Secure Socket Layer/Transport Layer Security), ensuring compatibility with a wide range of VPN devices and configurations.

1. Go to the AWS Management console
2. Search for VPC > Create VPC
3. Enter VPC name, select **VPC and more, Number of Availability Zones (AZs) = default, Number of public subnets and private subnets = 2, NAT gateway= none, VPC endpoint = S3 gateway**, click **create VPC.**

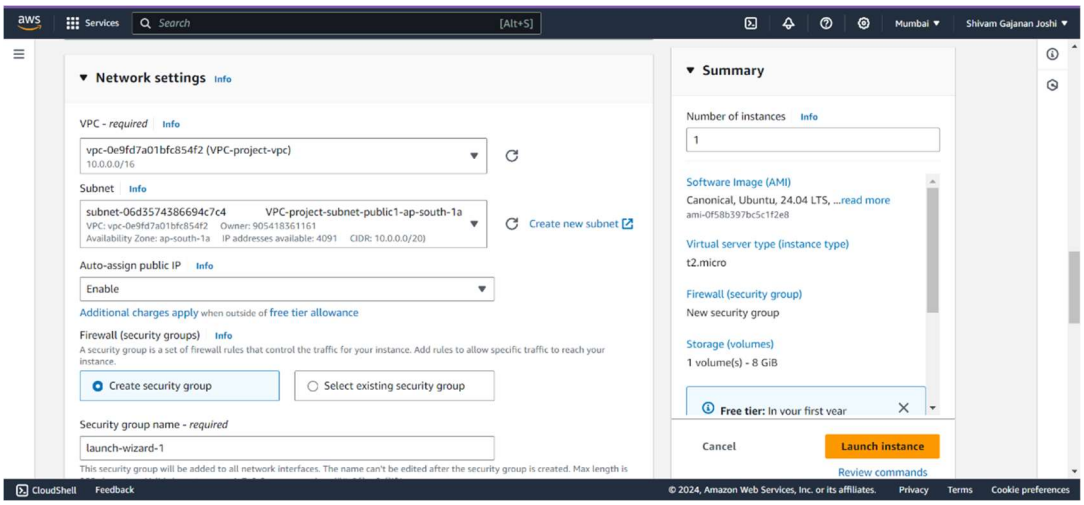


4. A preview of your VPC Network which shows how it will be interconnected with its components.

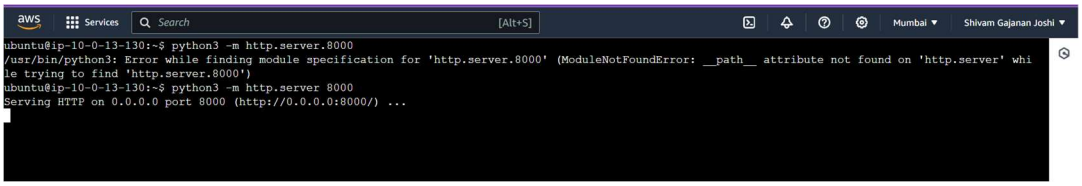
5. The below image shows the successful creation of VPC components.



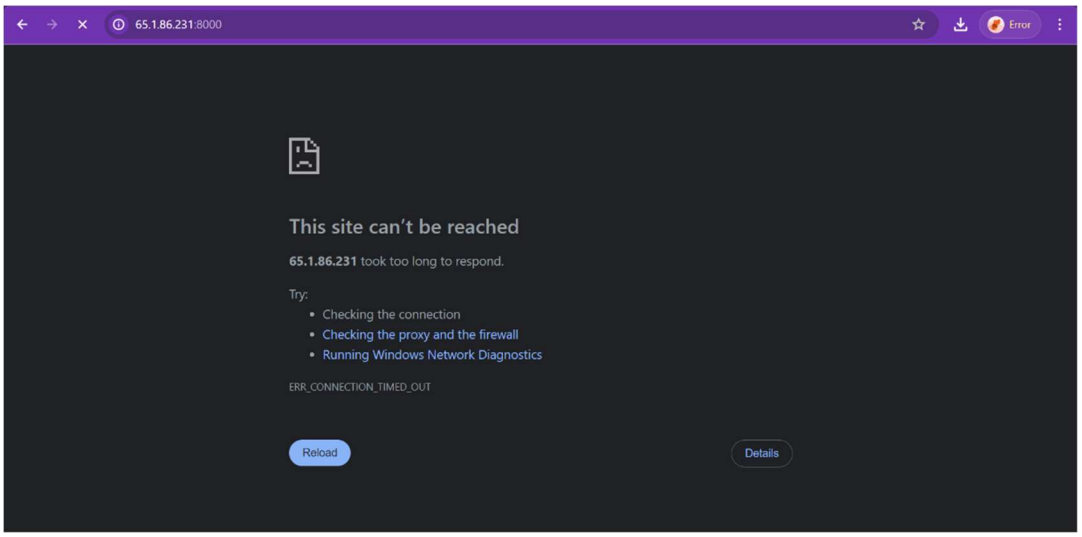
6. Now we will use this VPC in our EC2 instance. So now go to EC2>Launch Instance> Enter a name for EC2> choose AMI>Download the **.pem** file> Under network settings choose edit> and select recently created **VPC> Enable auto assign Public IP(If you keep it default, your EC2 wont start). Click on Launch instance**



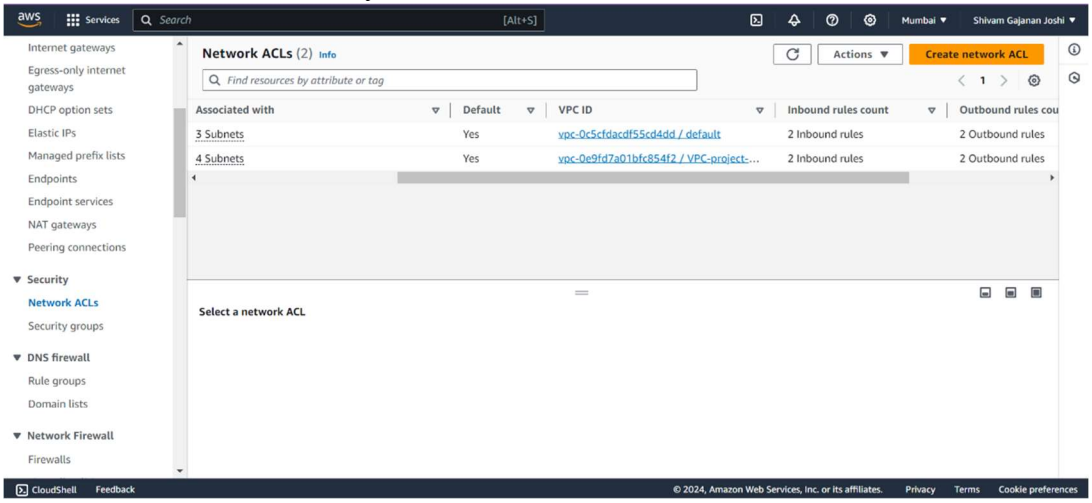
7. Connect to the EC2 using SSH and update the packages (**sudo apt update**) and check if python3 is installed(run the command => **python3**).
8. Now we will sun a simple http server using python, for that use the following command(**python3 -m http.server 8000**) we will access the application using(**http://<public ip of EC2>:8000**)



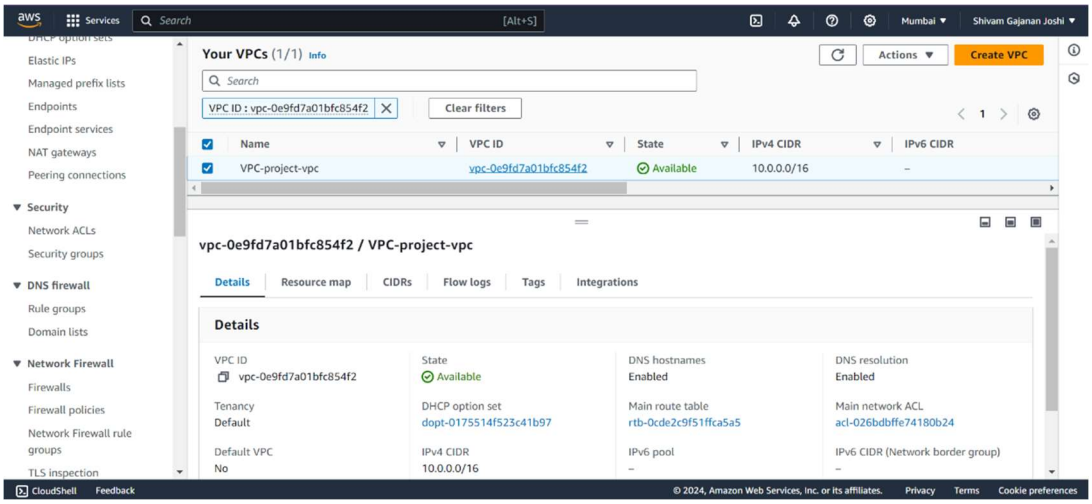
9. Now open new tab in browser and enter (**http://<public ip of EC2>:8000**), oops! We cant access the application, but why? It is because we have not enabled the inbound traffic to our instance.



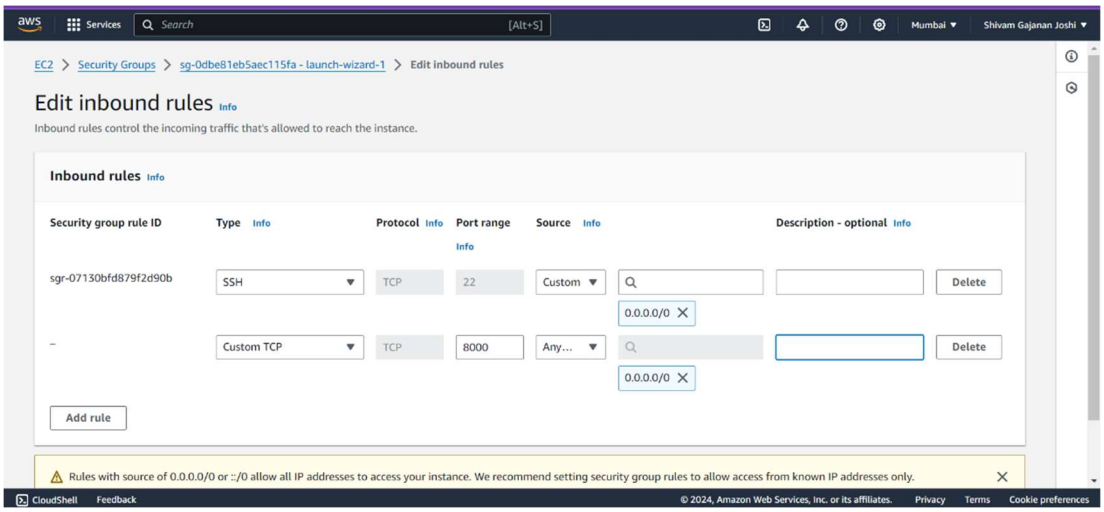
10. Check the NACL of recently created VPC subnet.



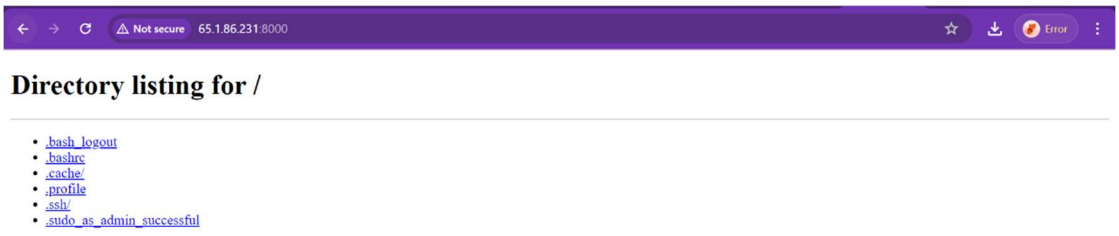
11. Click on Main network ACL on left hand side.



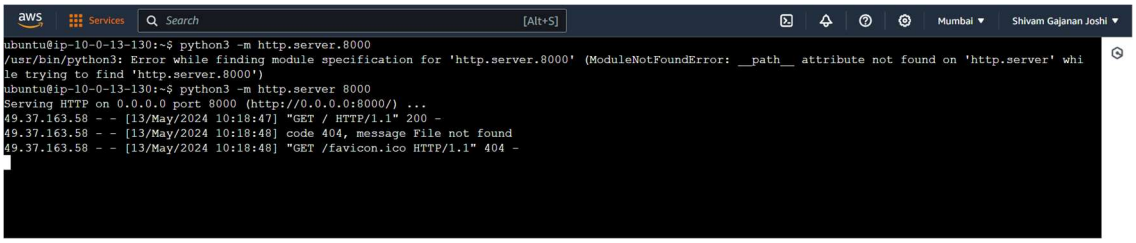
12. Now we need to add a rule in Security group to allow traffic to our EC2 so that we can access our application. Click on inbound> add rule> enter port no. on which your application will be accessible and save it.



13. Refresh the browser to access the application. Yaayy! We now can successfully access our application.



14. We can see the status codes on the EC2 aswell.



NOTE: Please delete the resources to avoid billing charges.