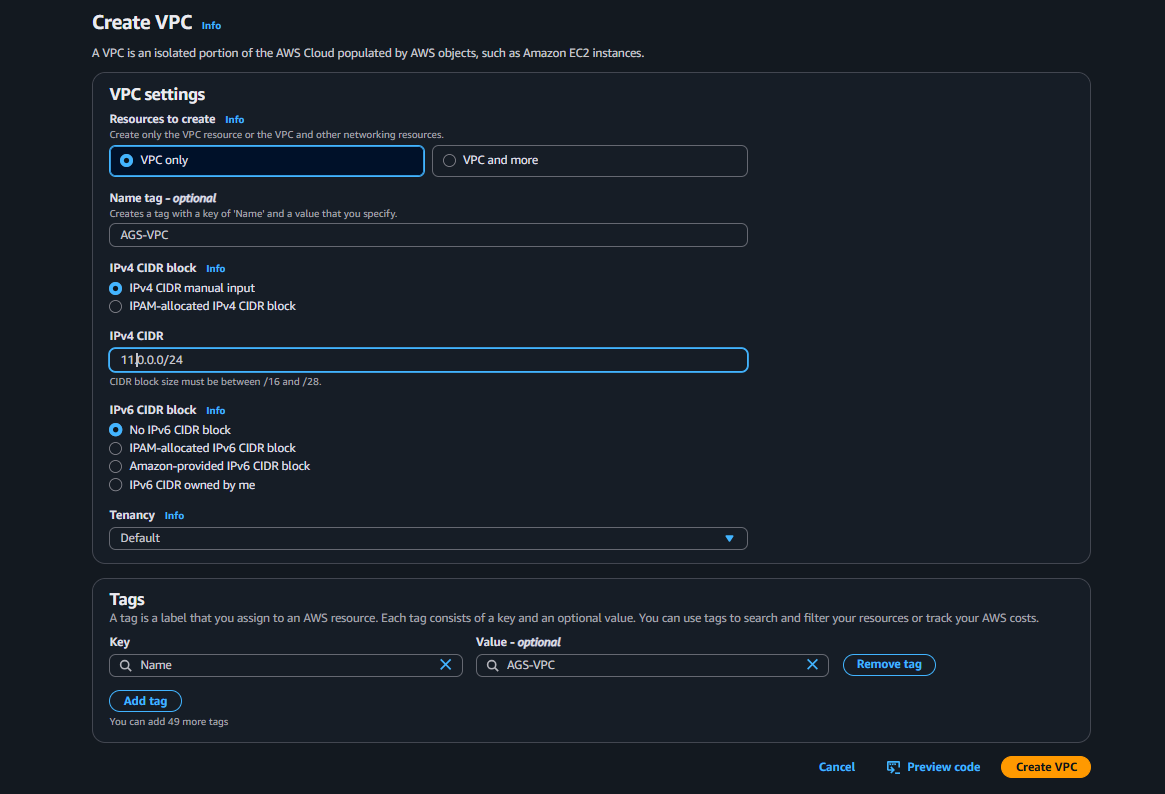
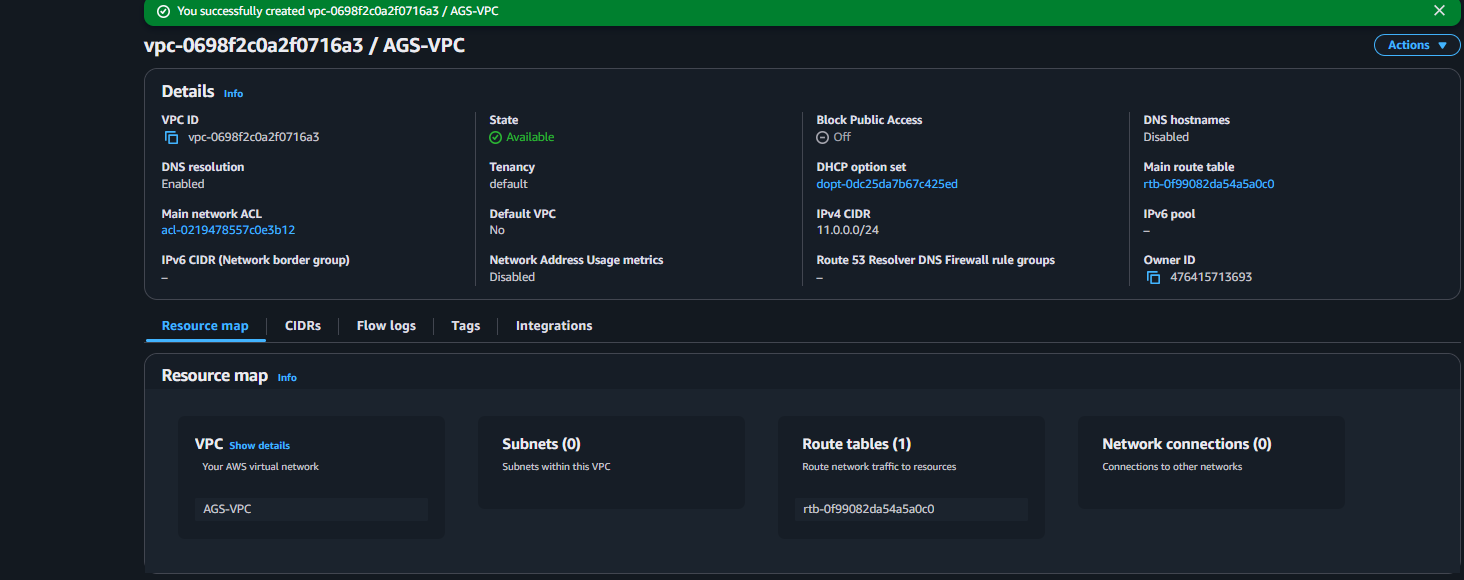
1. Create one vpc in N.virginia region.

First go to vpc

Next give the required details to it and create vpc



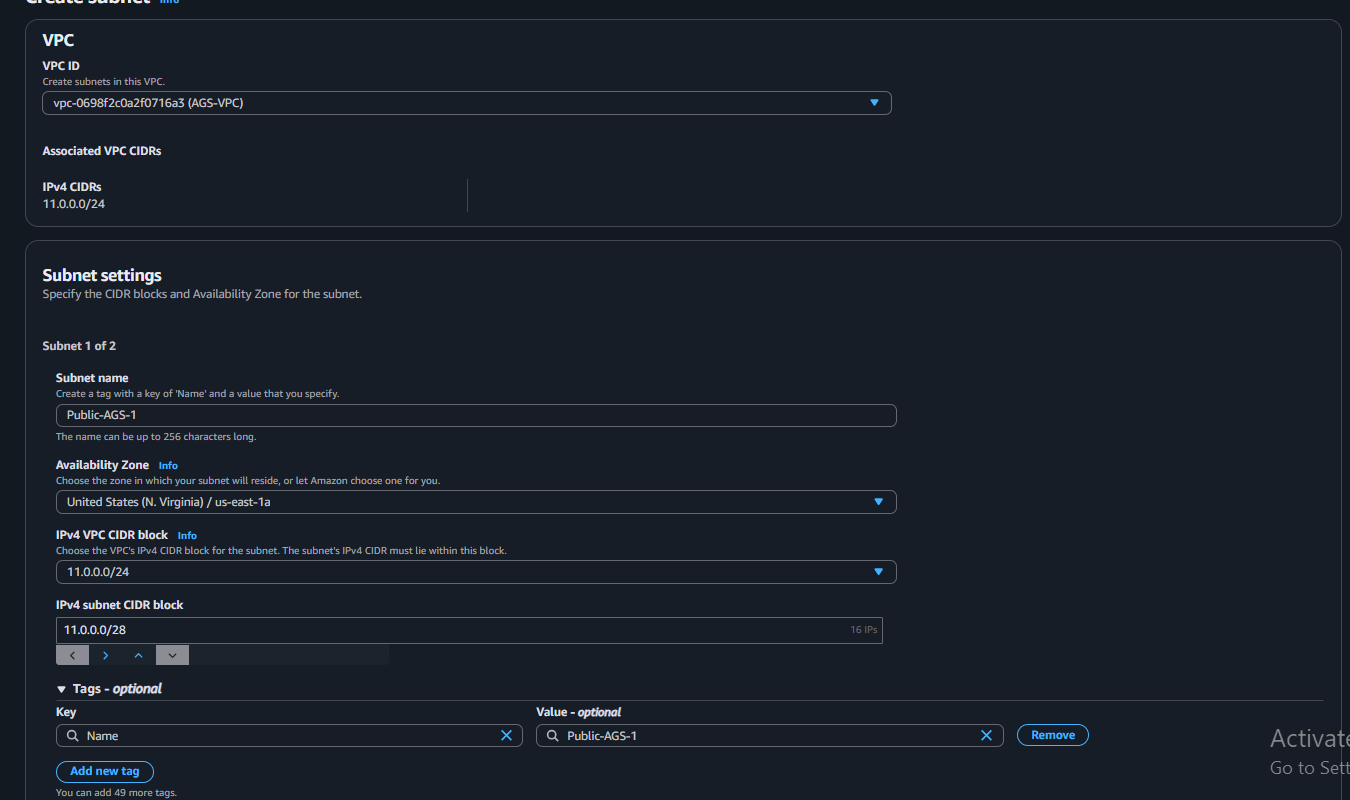


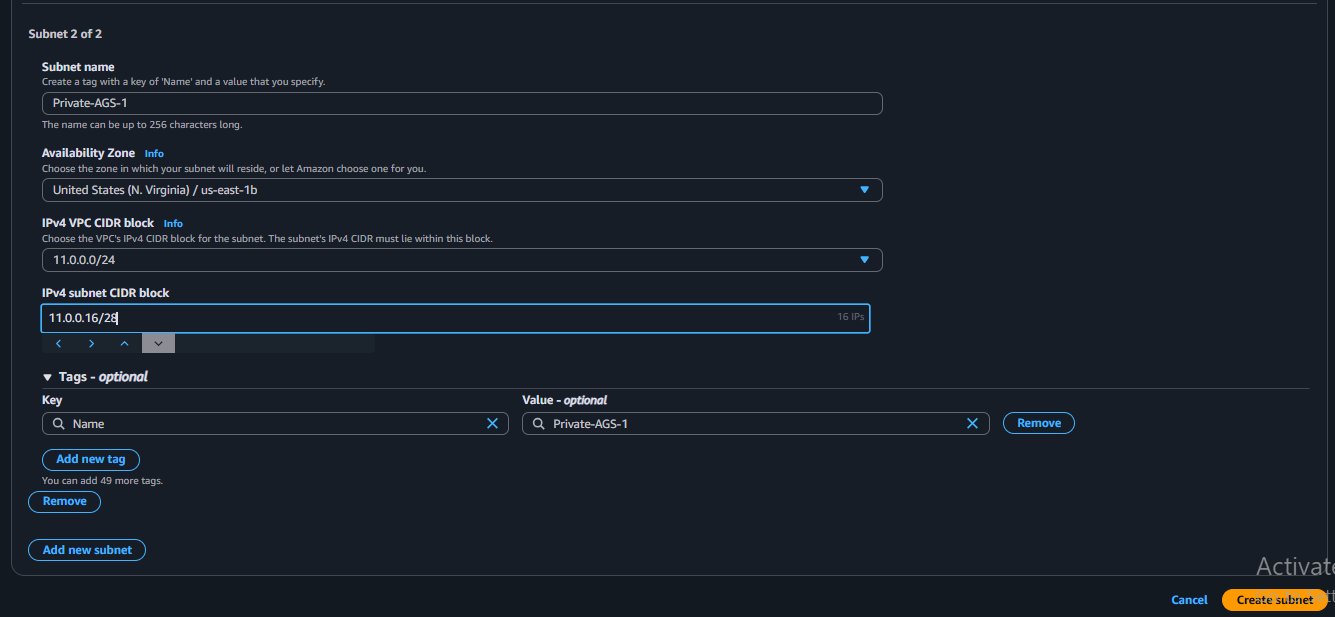
1. Create two subnets. One Public subnet and one private subnet.

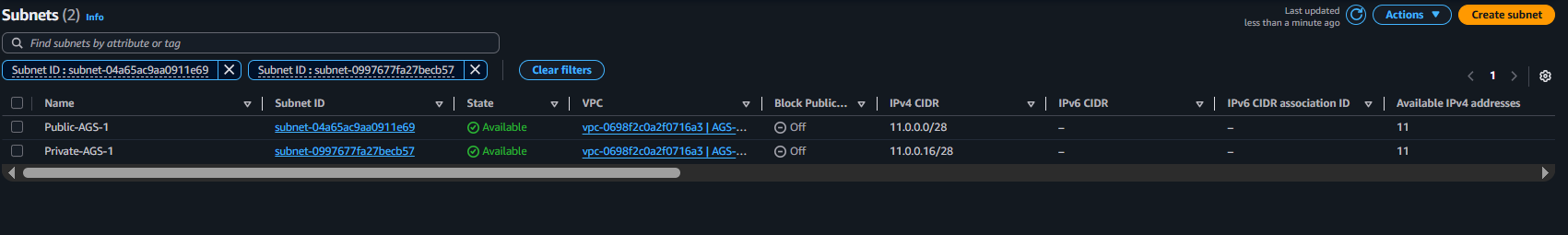
First go the the subnets

Next add the vpc to it and create two subnets

One in public and another in the private



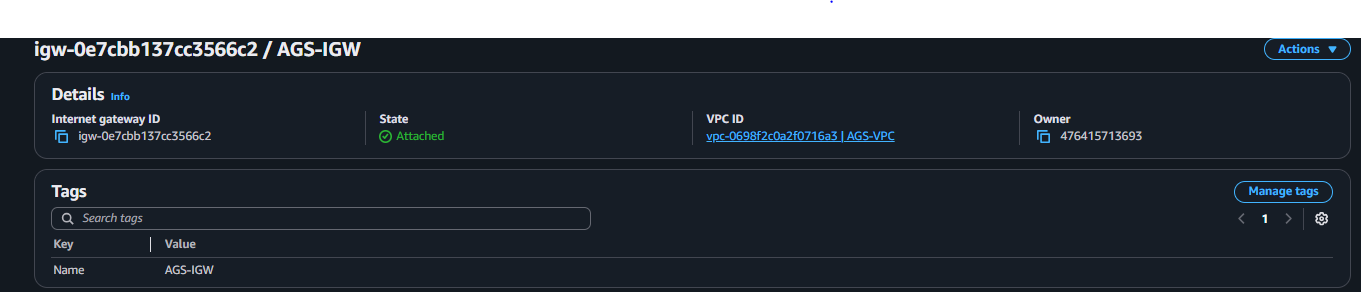




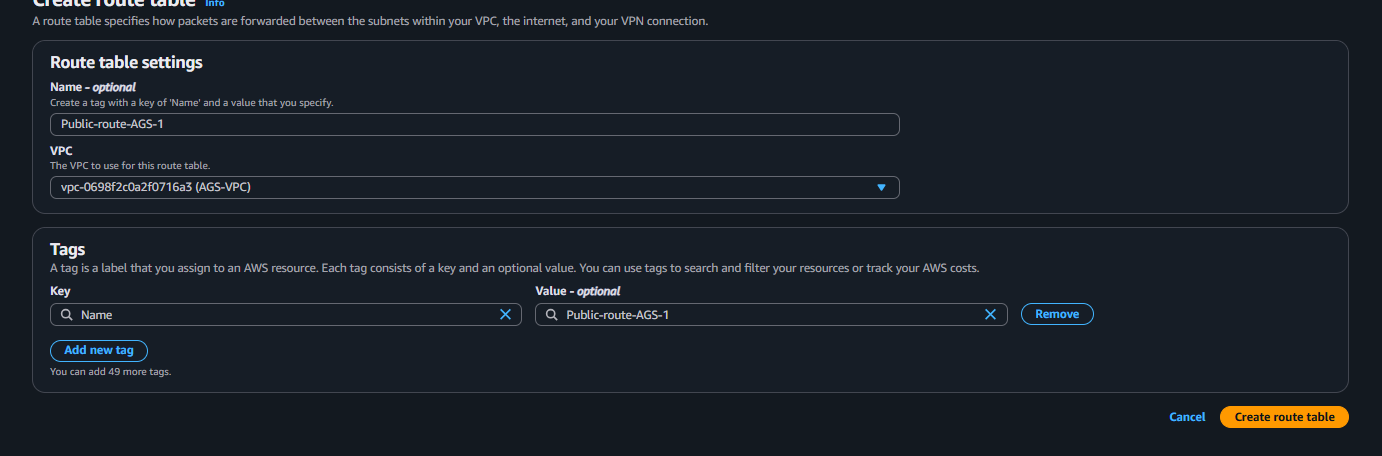
1. Provide the IGW to the vpc.

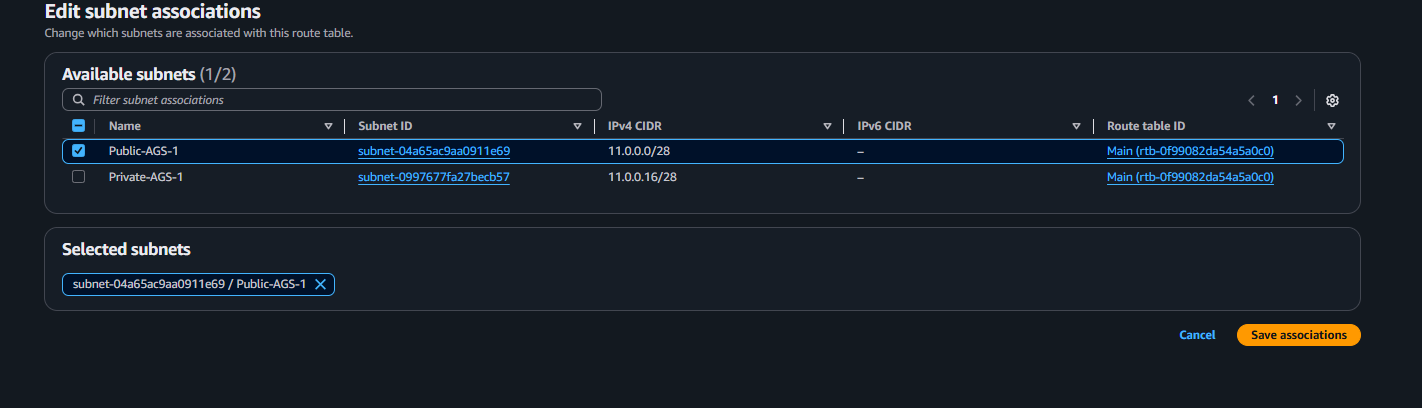
First go the internet gateway and create one

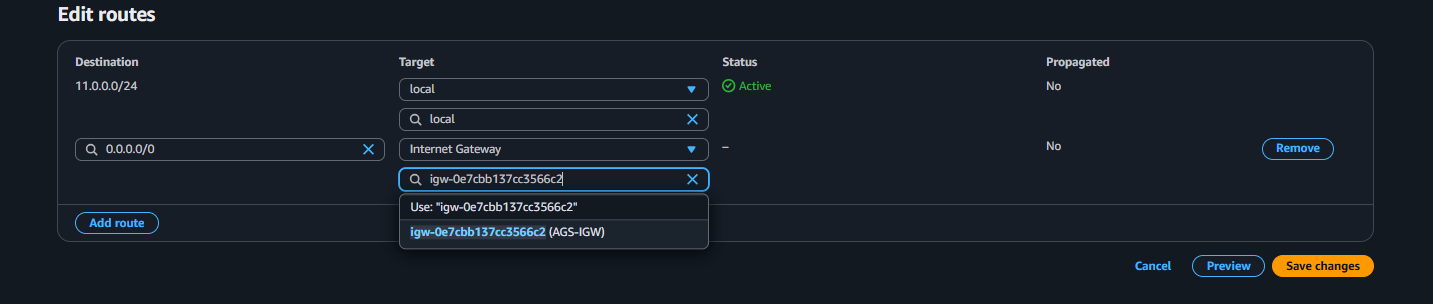
Next add the vpc to it

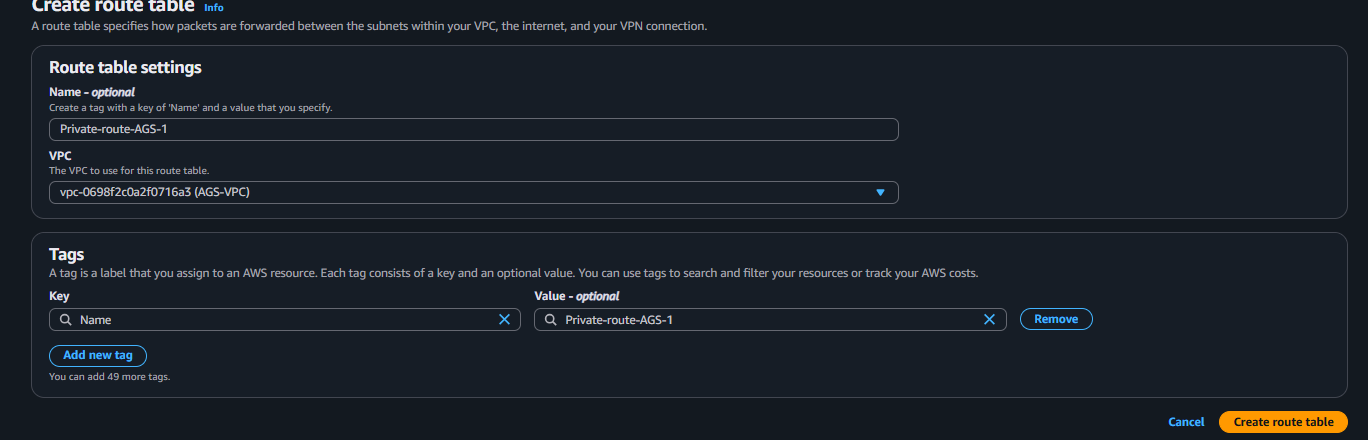


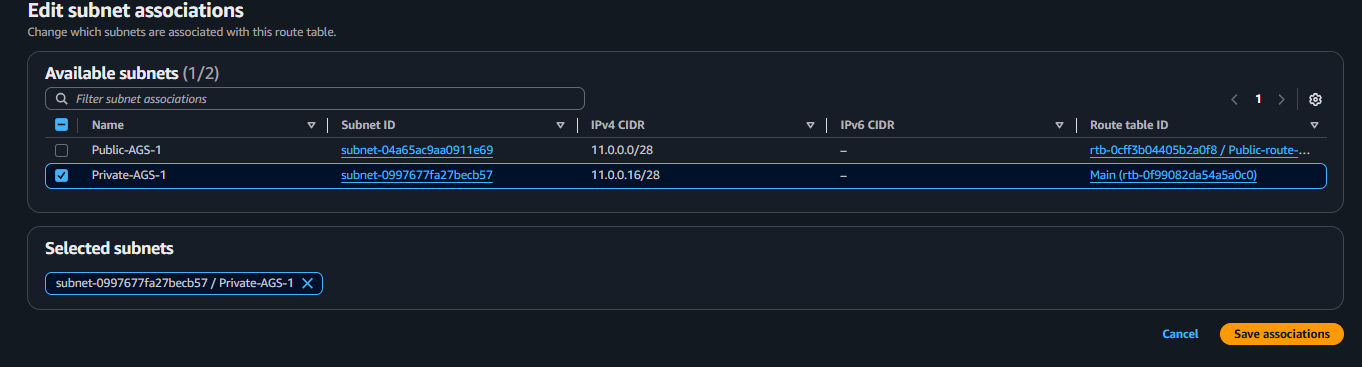
1. Create One public RT and one private RT.

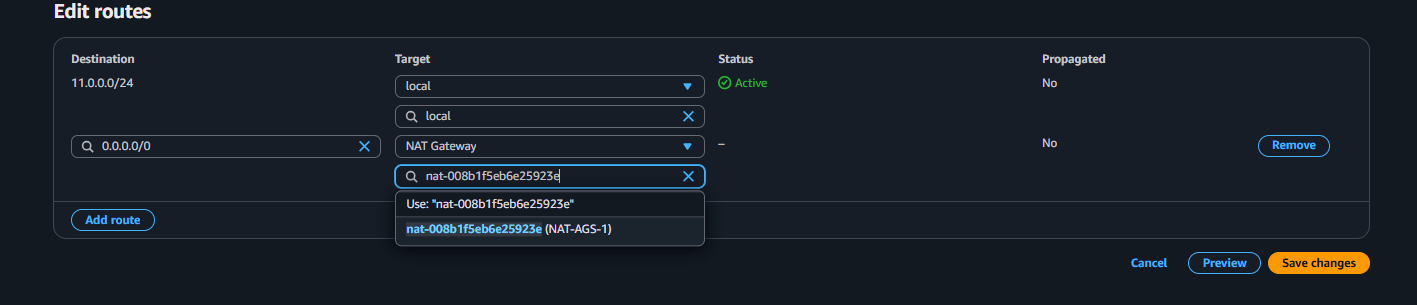




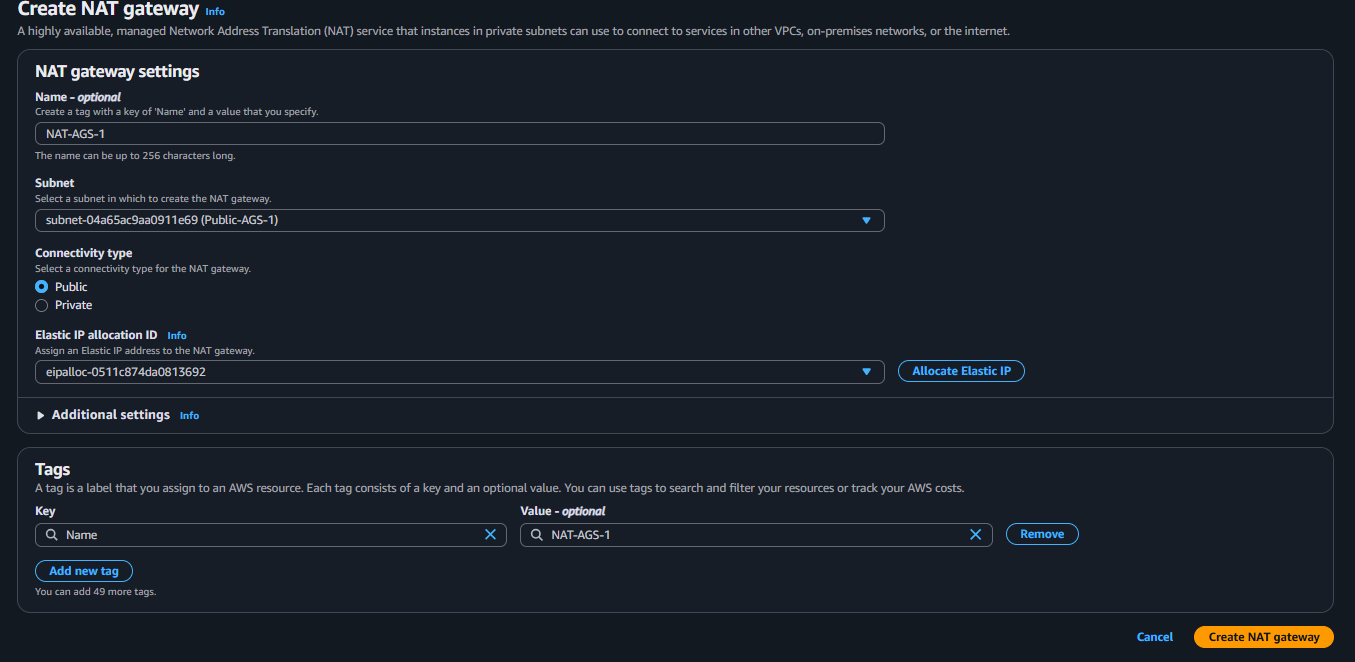


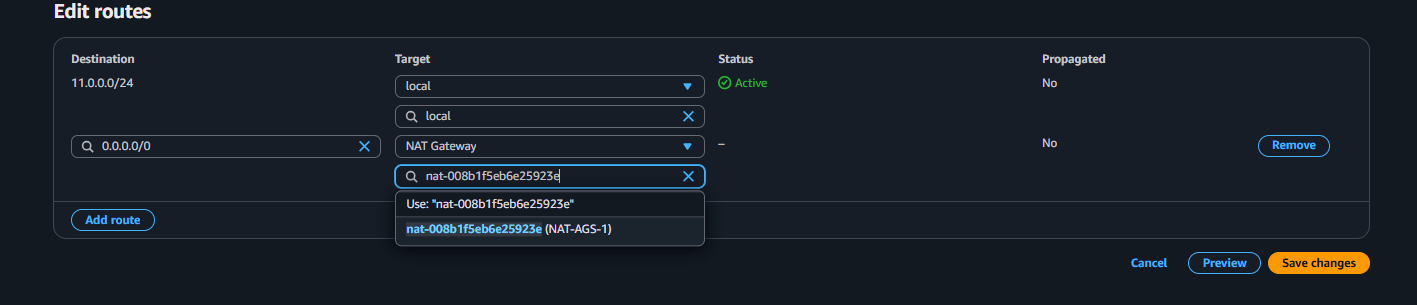






1. Deploy NAT gateway on public subnet and attach the NAT gatewat to private subnet.



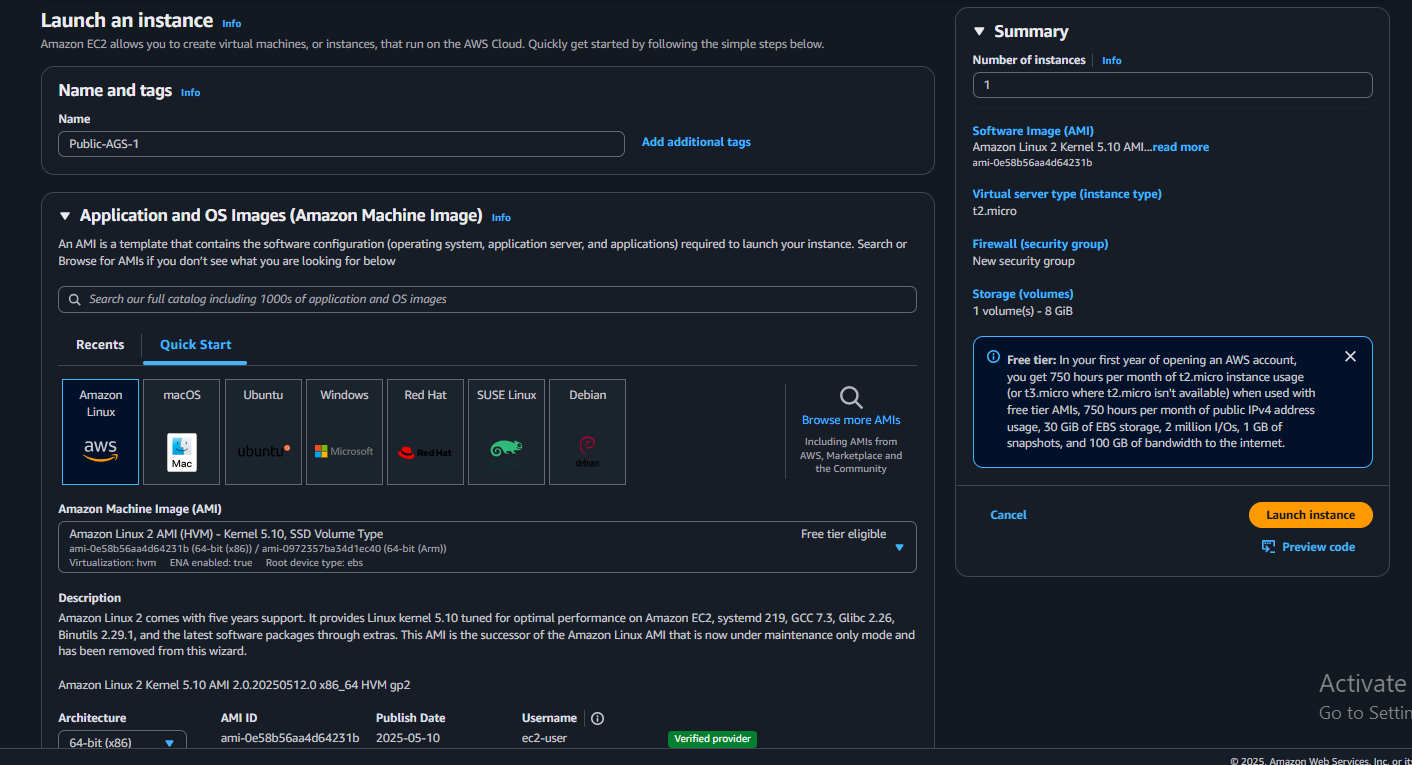


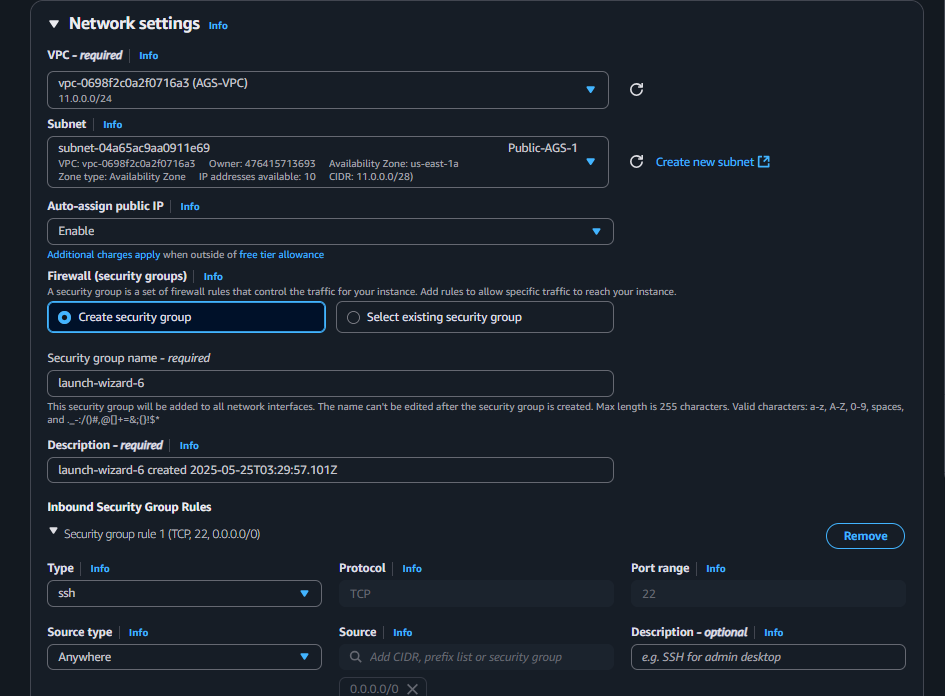
1. Create Two instances,one in public subnet and one in private subnet.

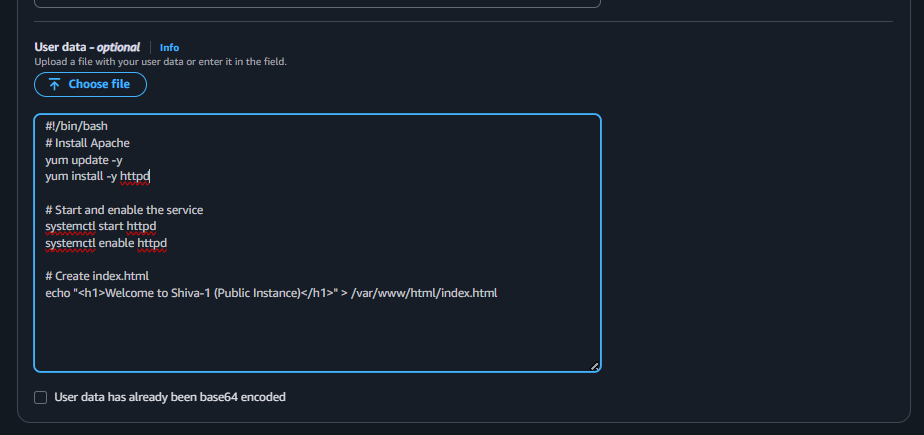
First we have go to EC2

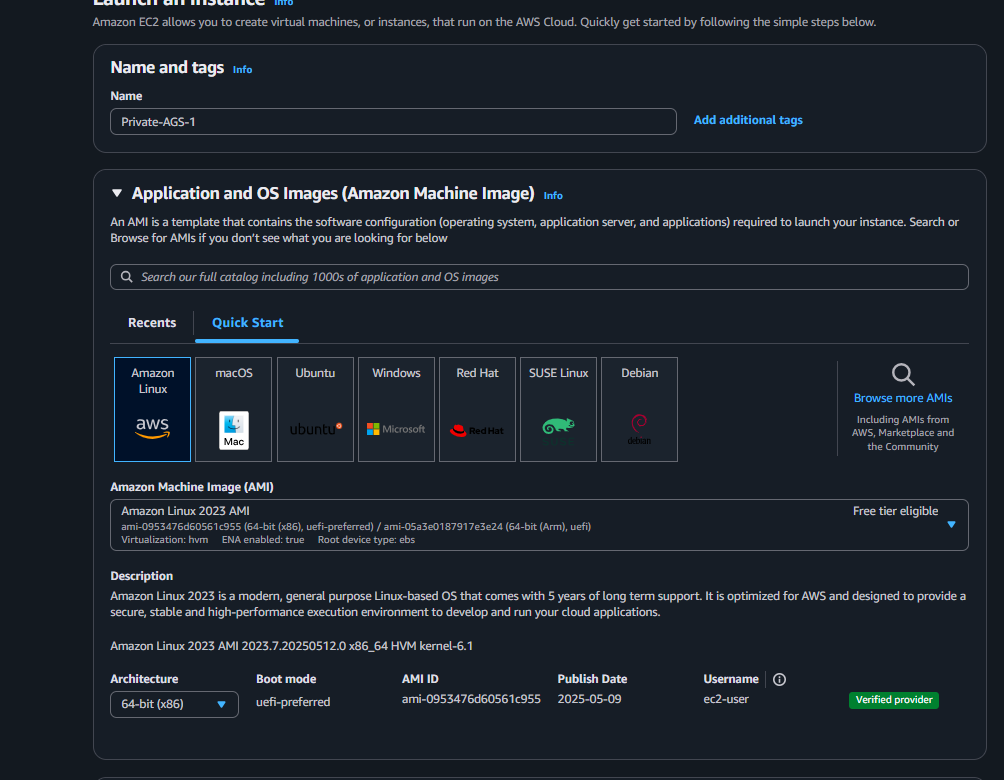
Next lauch two instances one in public and another in private

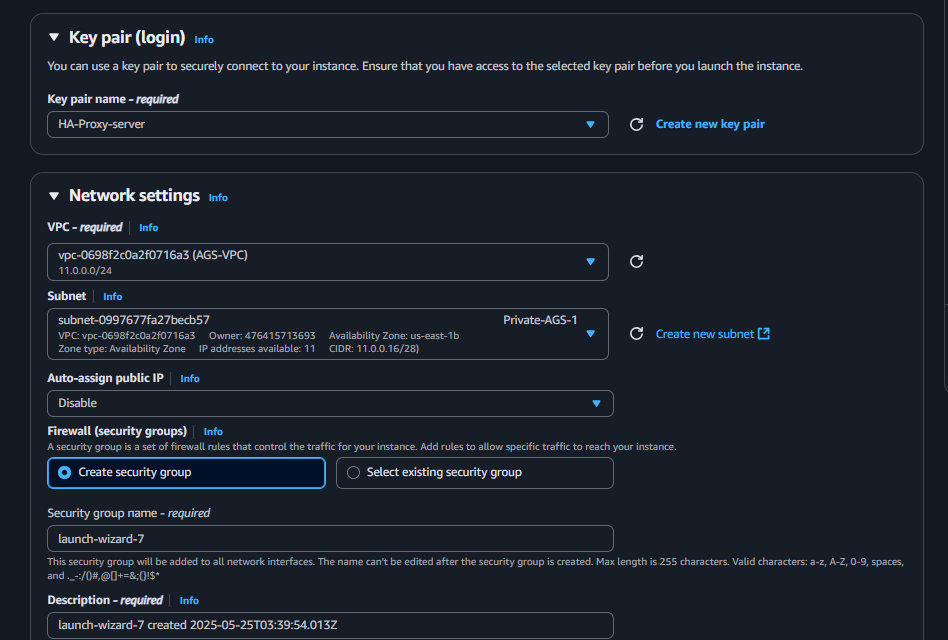
Next while creating add the script to install httpd and check it

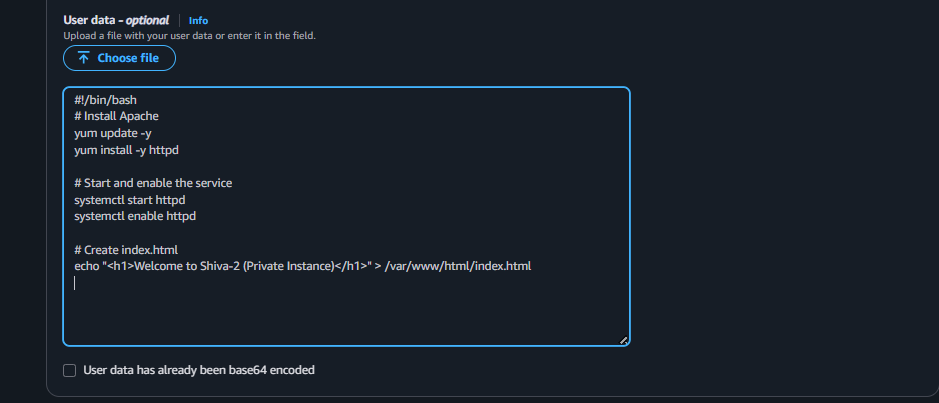












1. Deploy Apache server on both the ec2 instances with sample index.html file.

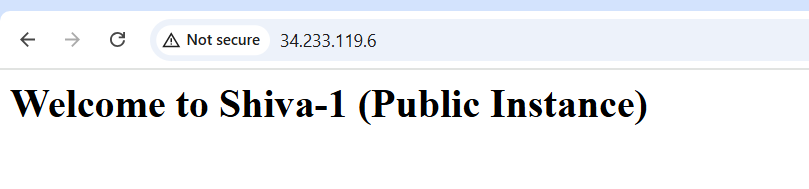
First launch two instances before question I have done this is continue

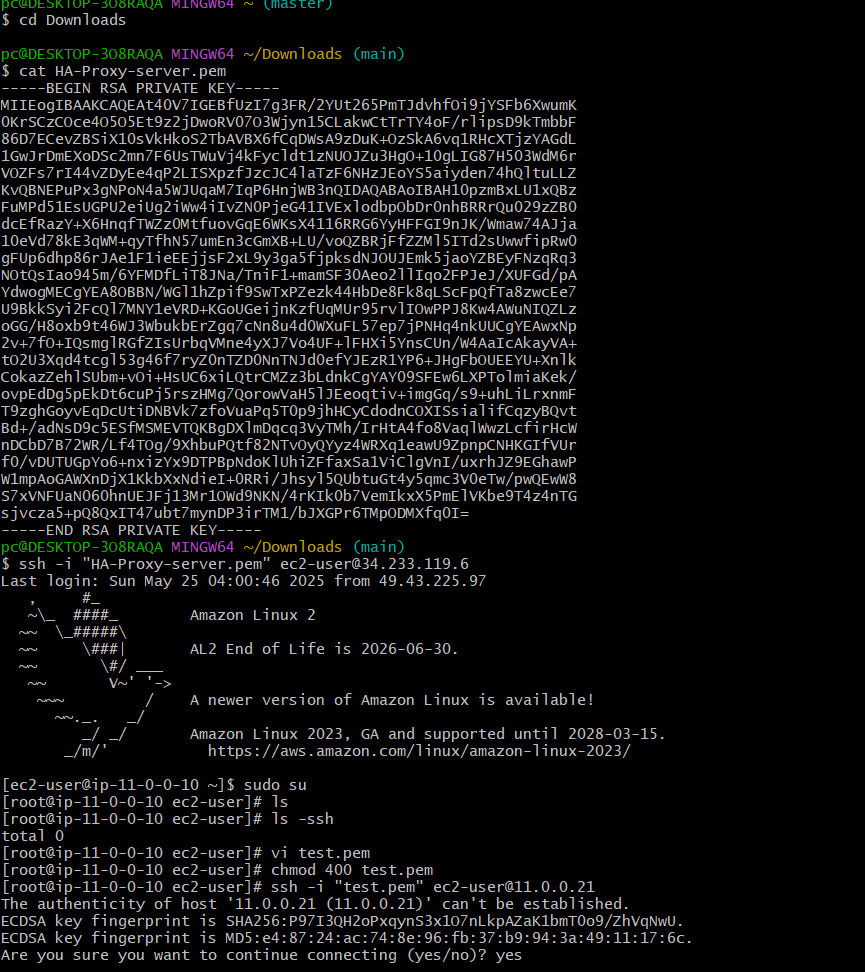
Next check it with public ip in webpage

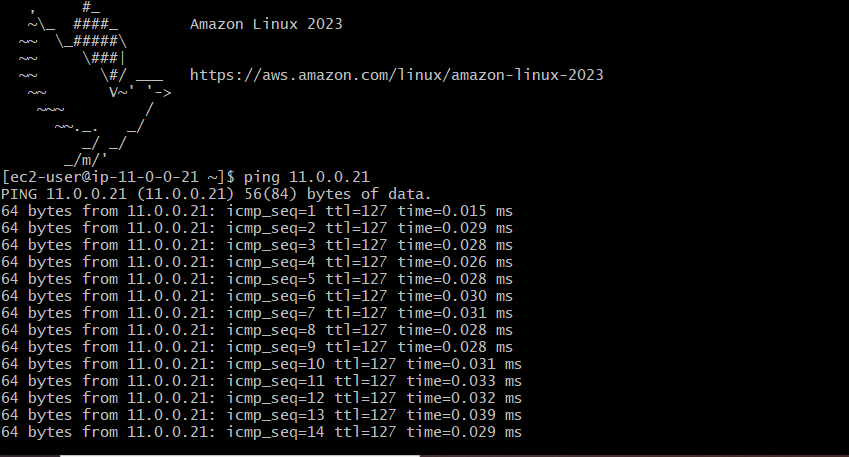
After that go to terminal and download the pem key file to save it as new

Next login to ssh with public and save the new pem key

Next with this new pem key logging to private and check with ping with your private ip if it works ok

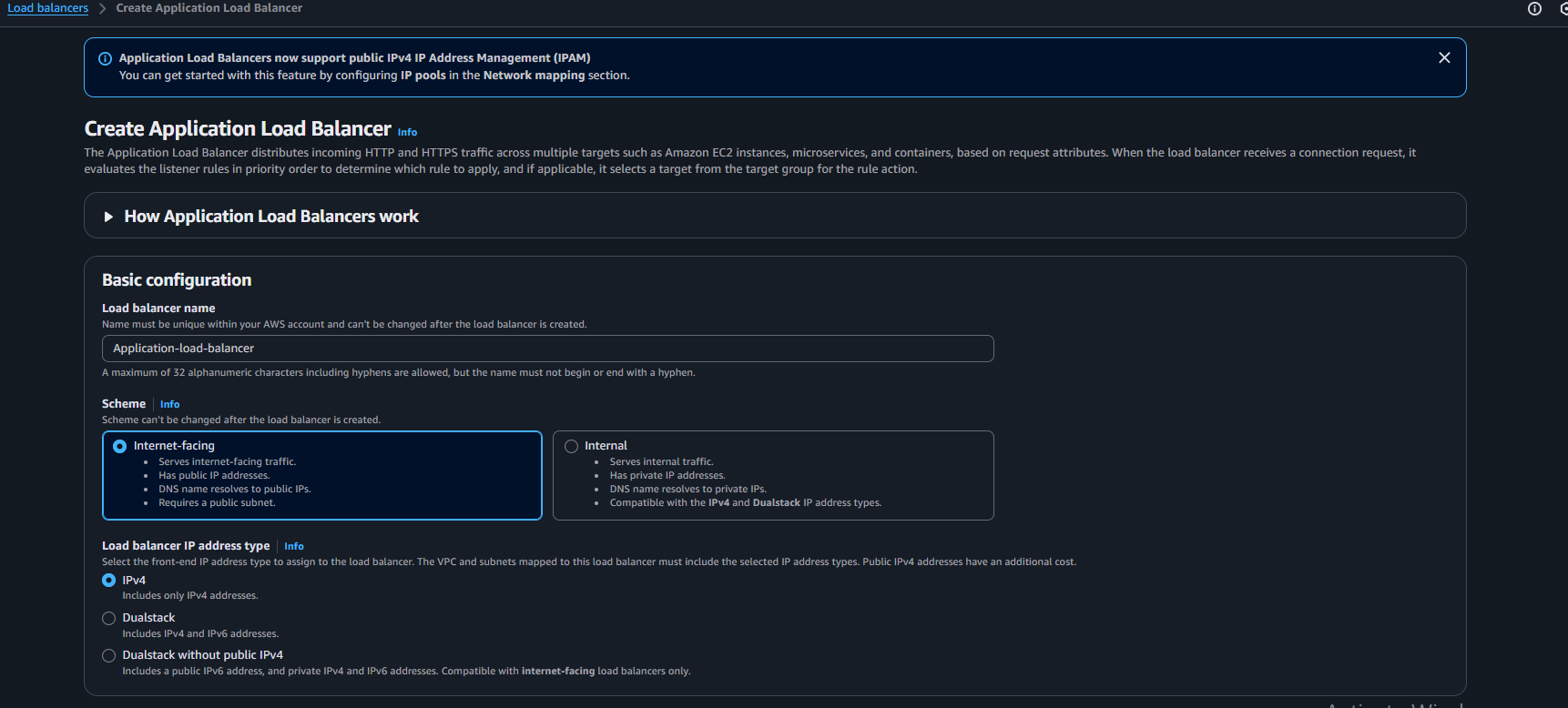


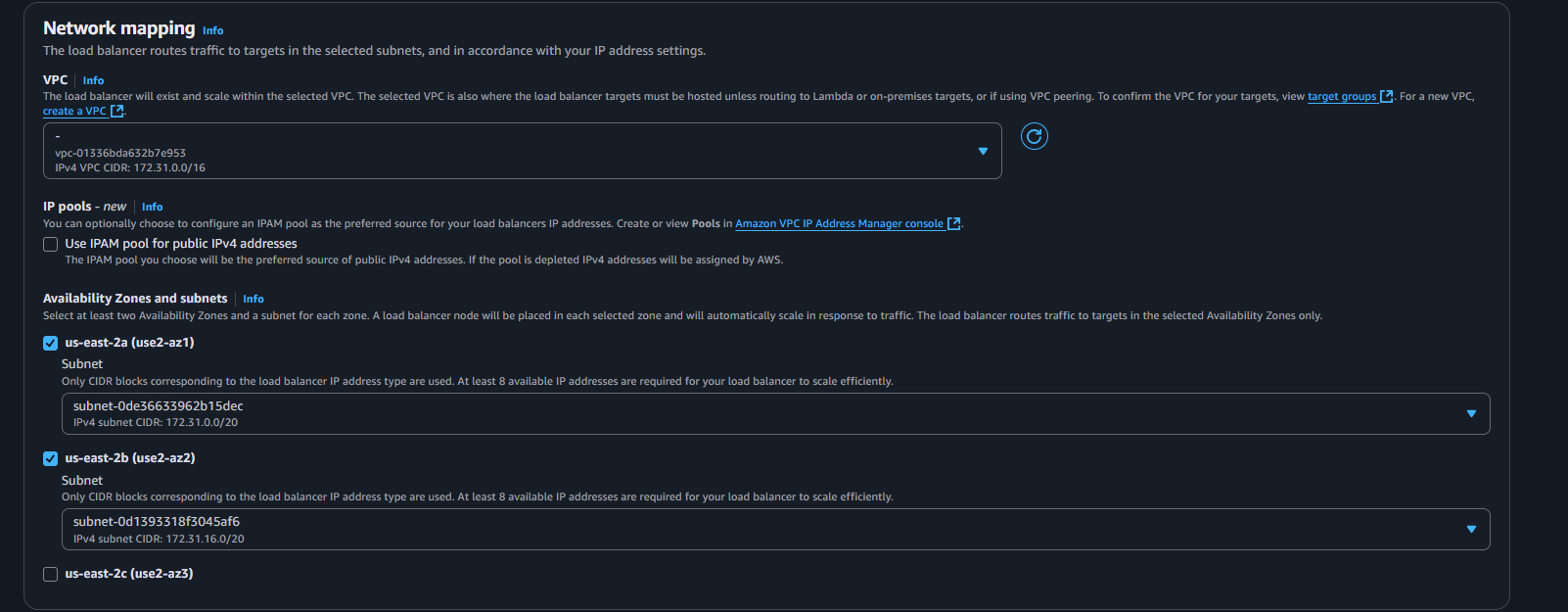


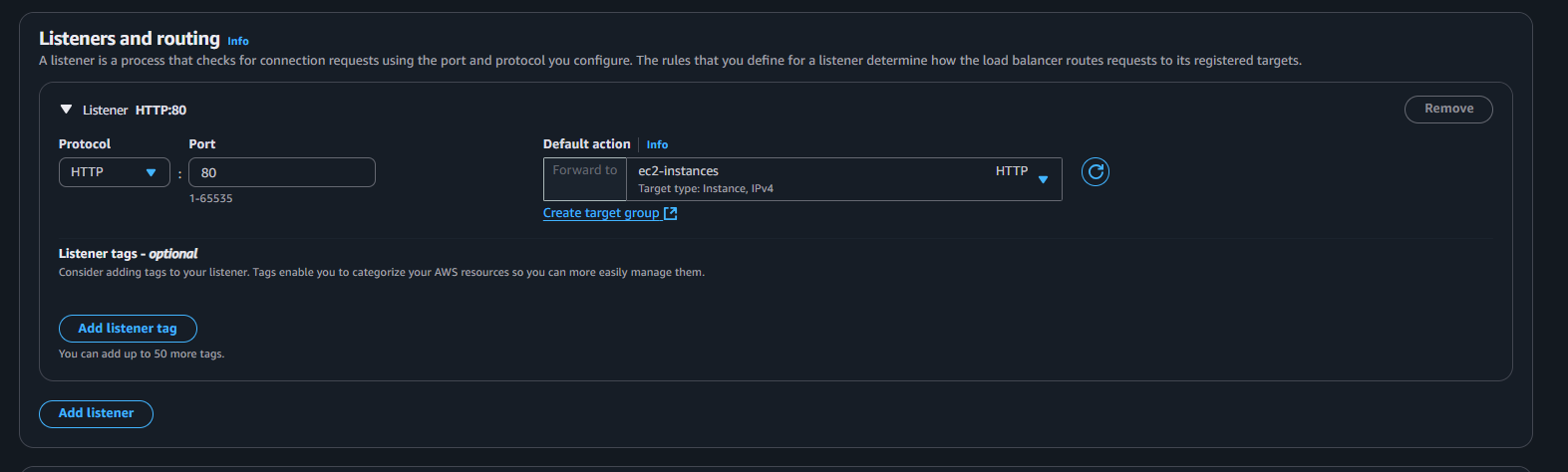


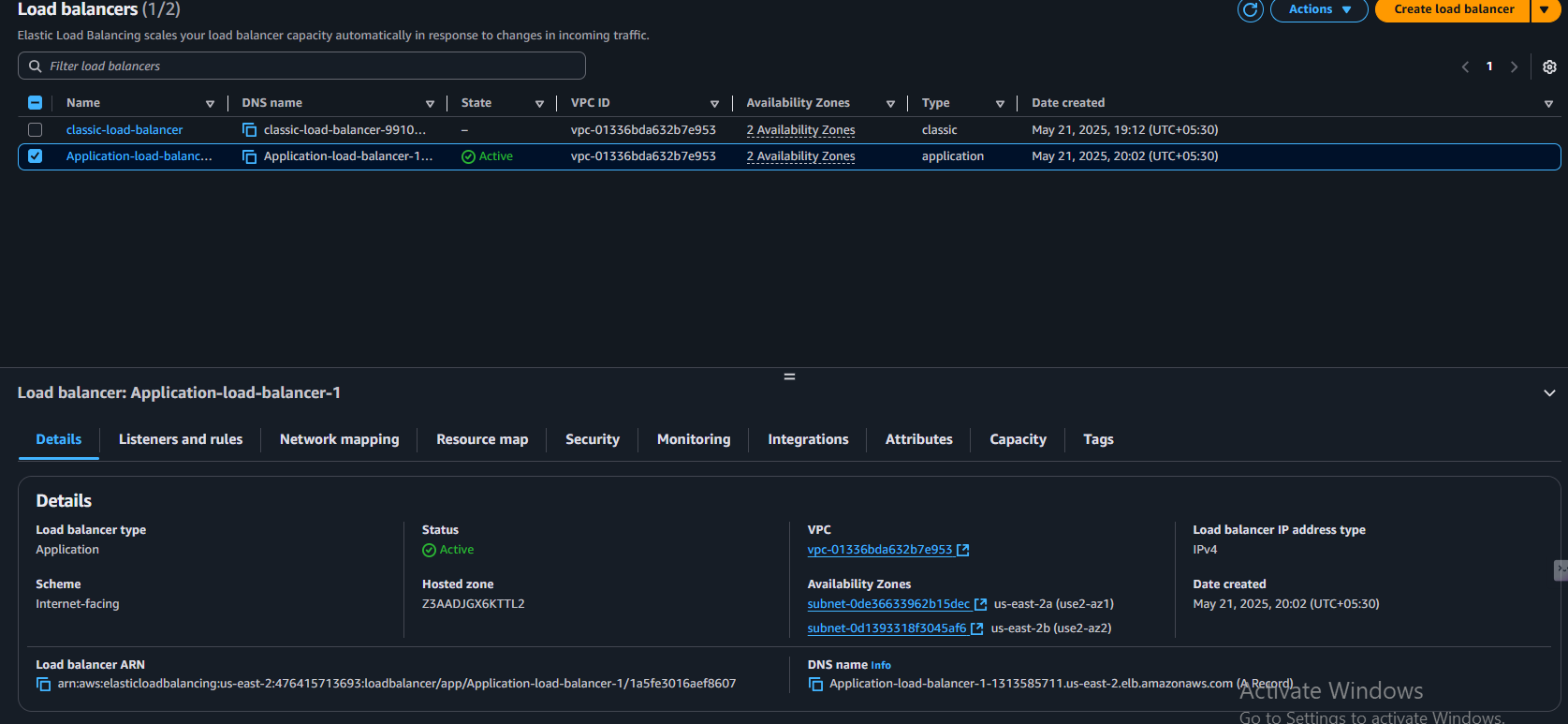
1. Create one application load balancer and attach the load balancer to both the ec2 instances.
2. First we have to create two instances with two availability zones and aslo add the script to install httpd with index.html
3. Next go to Load Balancer and click on create load balancer
4. After that select Application Load Balancer and also add these two instances to it and create while creating before we have to create a group tragets and then we have add the instances to this

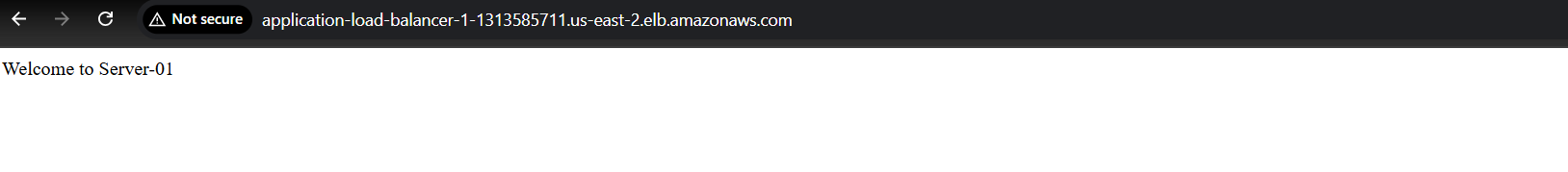
Next open the web page with DNS name and it will show the script what we have saved in two instances when we refresh each time

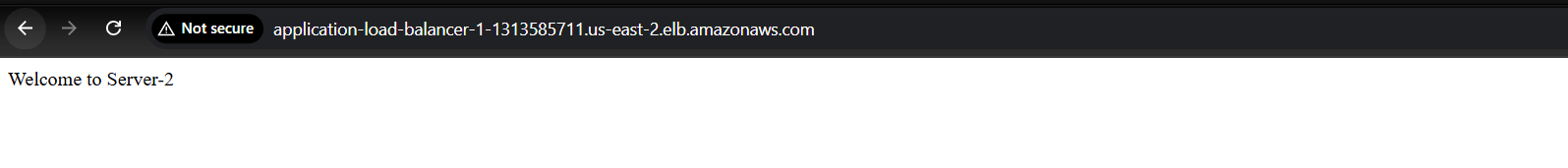












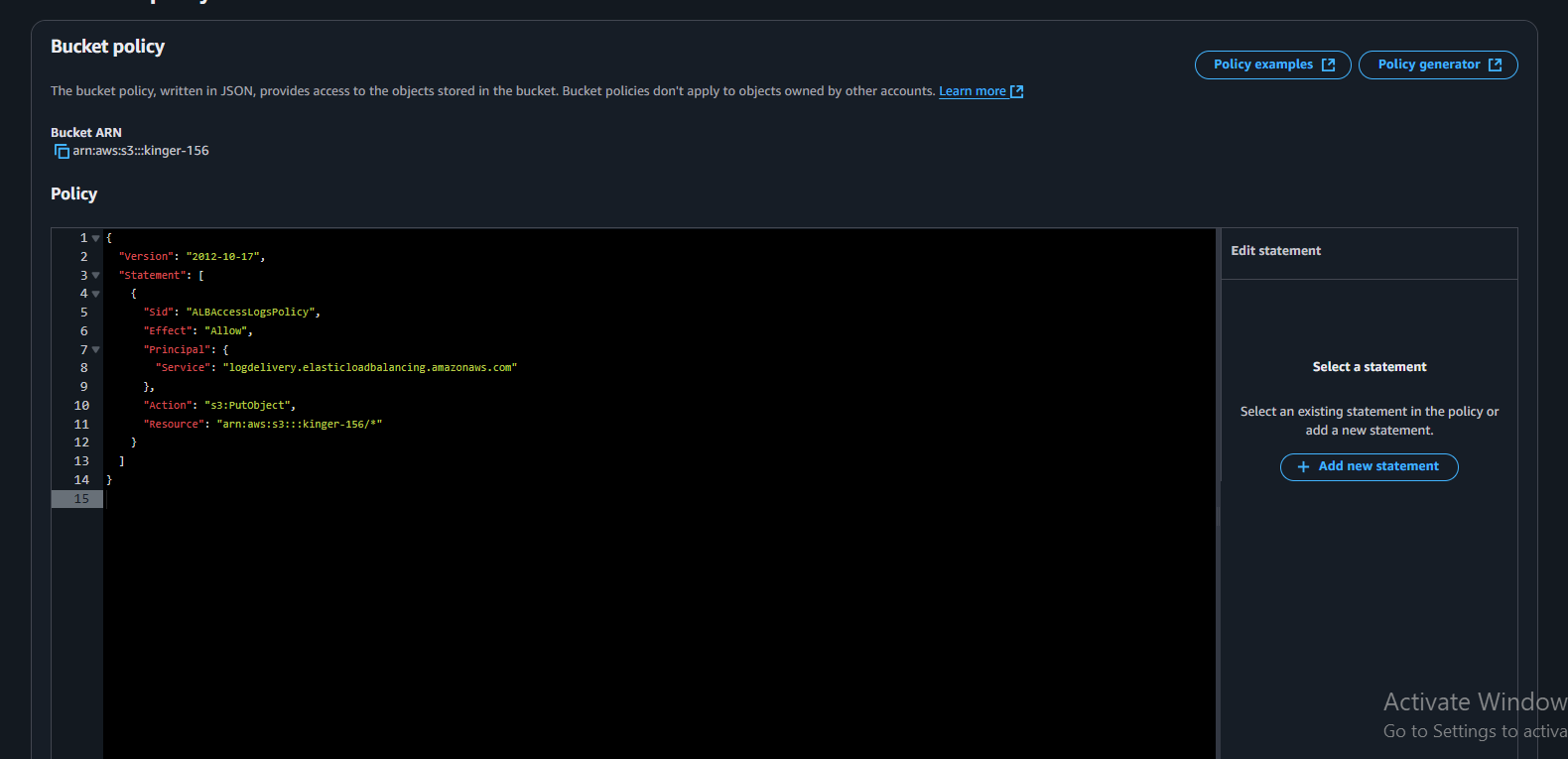
1. Store Application load balancer logs to s3.

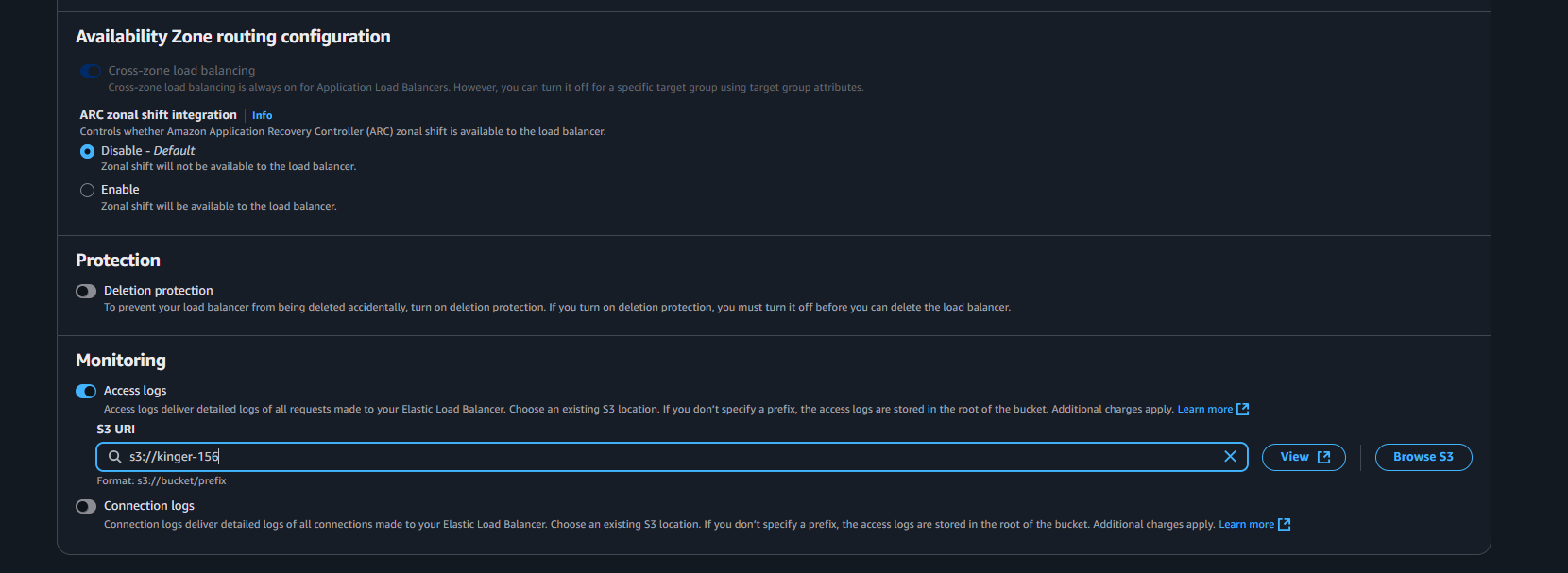
**Created an S3 bucket named kinger-156** i

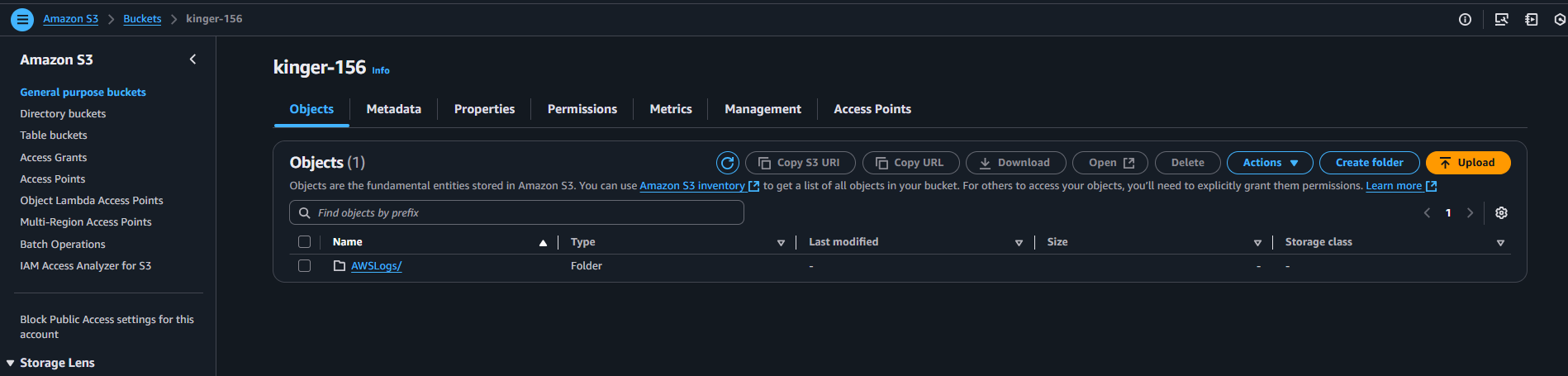
**Enabled Access Logs for your Application Load Balancer (ALB)** and set the log destination to the kinger-156 bucket.

**Made sure connection logs are OFF** (optional for ALB).

**You need to add a bucket policy** to kinger-156 to allow ALB to write logs into it.







1. Store the vpc flow logs to cloudwtach group.

 **Selected your VPC**  
→ AGS-VPC

 **Chose to send logs to CloudWatch Logs**

 **Picked an existing log group**  
→ aws-cloudtrail-logs-476415713693-068239ba

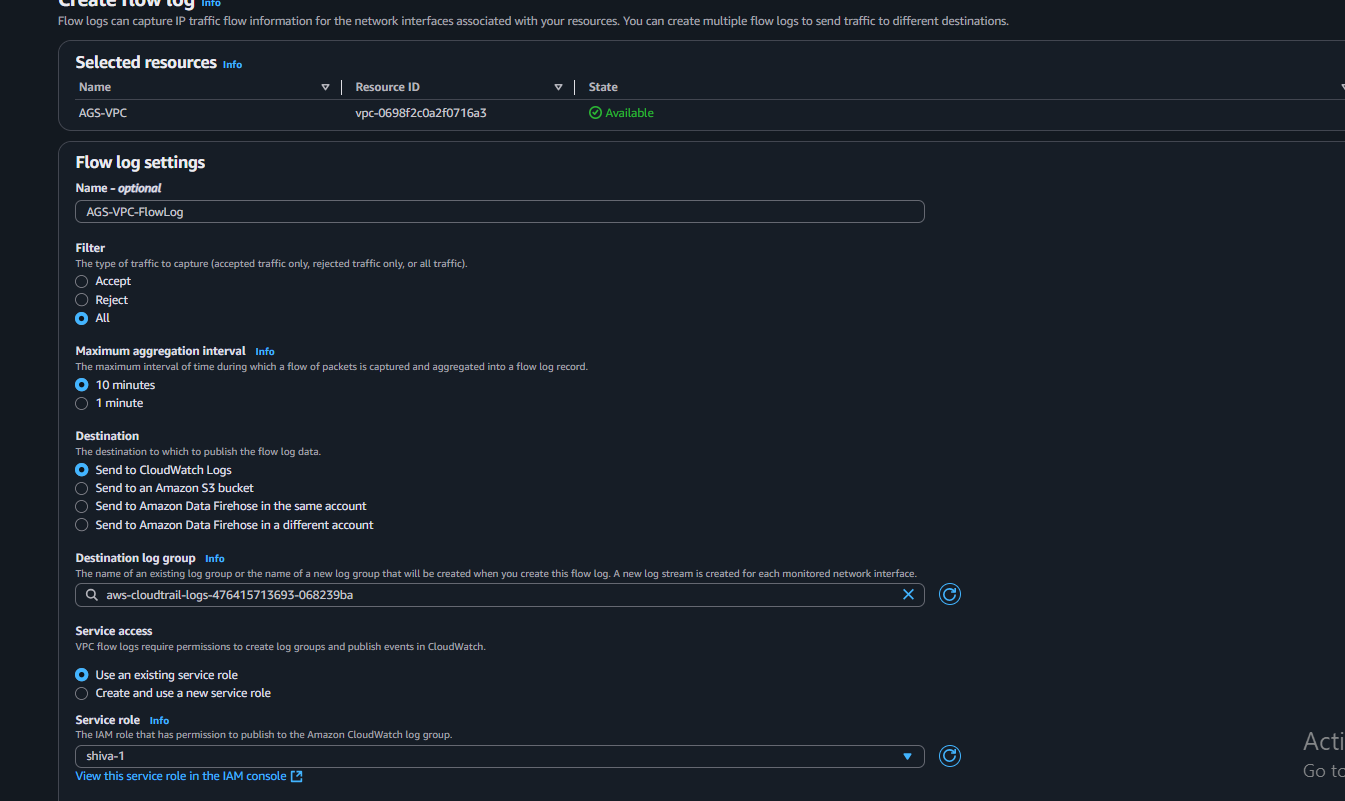
 **Used an IAM role**  
→ shiva-1 (gives permission to send logs)

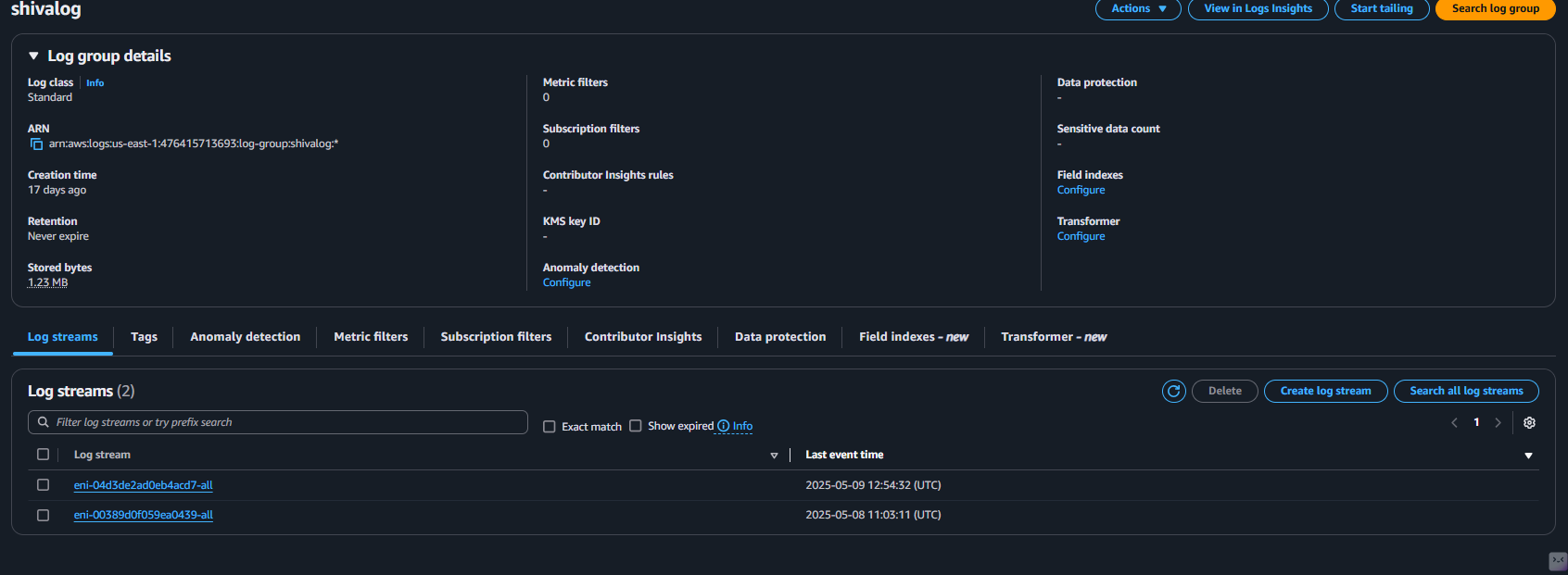
 **Set filter to** All traffic (logs accepted + rejected)

 **Kept default settings (10 min, default format)**

 **Created the flow log** ✅

 **Now logs are showing under ENI log streams** in CloudWatch





1. Create Monitoring Dashboards to monitor cpu utilization and to monitor apache service.

 **Install Apache** on your EC2 instance

 **Start and enable Apache**

 **Enable Apache server-status page** (mod\_status)

 **Restart Apache**

 **Create a CloudWatch role (IAM Role)** and attach it to your EC2 instance

* Attach CloudWatchAgentServerPolicy to the role

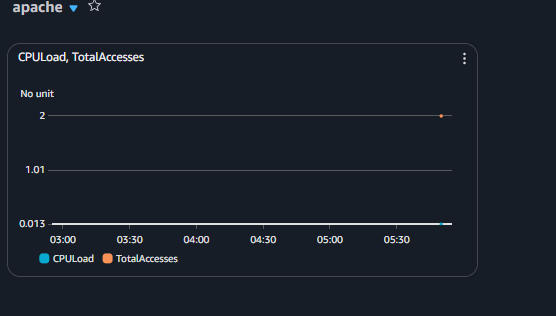
 **Write a small script** that sends Apache data to CloudWatch

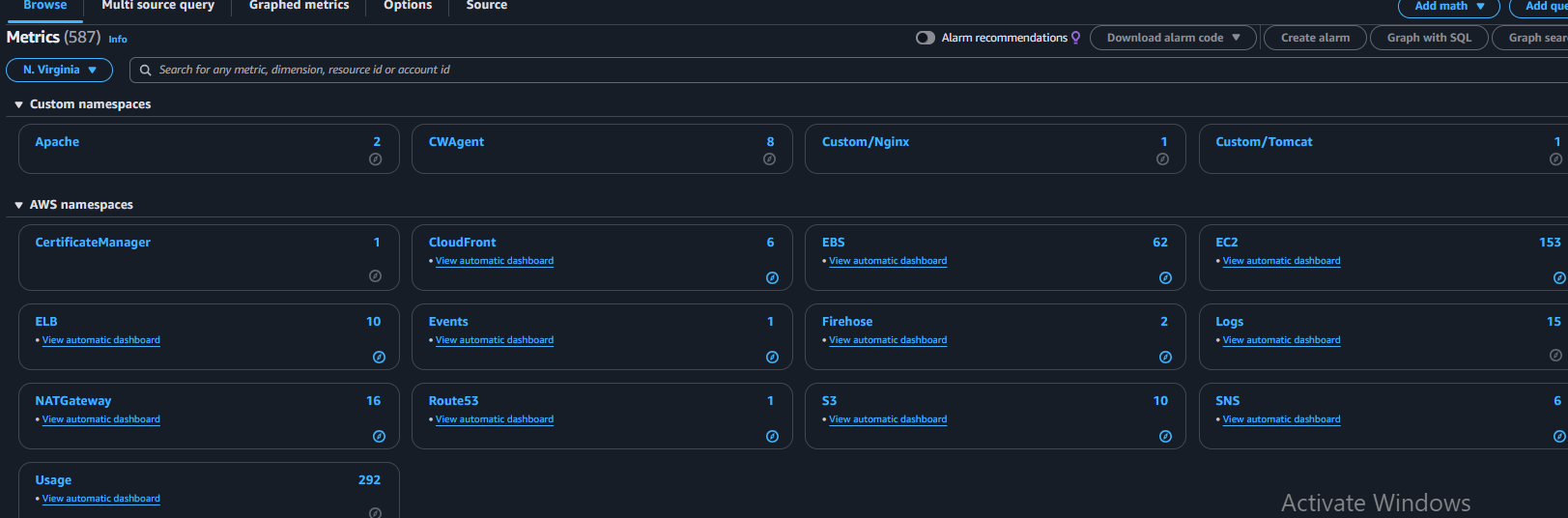
 **Run the script manually**

 **Go to AWS → CloudWatch → Metrics**

 **Select “Apache” namespace**

 **Add CPU Load and Total Accesses to your dashboard**







1. CPU utilizationis more than 70% then it should triggere Autoscaling and launch new instance.

 **Open AWS Console** → Go to **EC2** → Click **Auto Scaling Groups**.

 Click on your **Auto Scaling Group name**.

 Find the **Scaling policies** section.

 Click **Create dynamic scaling policy**.

 Select **Target tracking scaling policy**.

 Name the policy

 For metric, pick **Average CPU utilization**.

 Set the **target value to 70**.

 Click **Create policy** to save

