1)Create one VPC in N.virginia region.

Step 1: Select N. Virginia Region

1. Go to the AWS Management Console  
2. In the top-right corner, click on the region selector  
3. Select N. Virginia (us-east-1)

Step 2: Go to VPC Dashboard

1. In the Services menu, search for VPC  
2. Click “VPC Dashboard”

Step 3: Create a VPC

1. Click “Create VPC”  
2. Choose “VPC only” option

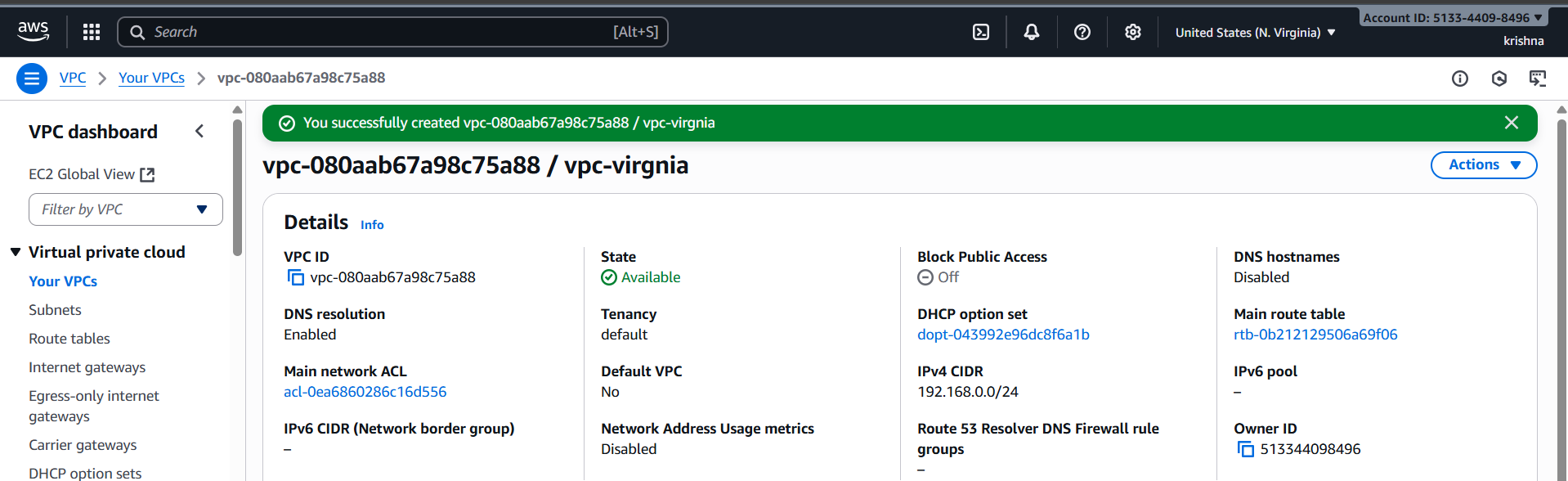
Step 4: Configure VPC Settings

1. Name tag: MyVPC (or any name)  
2. IPv4 CIDR block: 192.168.0.0/24(or your preferred block)

3. Leave IPv6 CIDR block as "No IPv6 CIDR block"  
4. Tenancy: Default  
5. Click Create VPC

Step 5: Confirm VPC Creation

1. After successful creation, click “View VPC”  
2. You will see your VPC listed with the specified CIDR



2)Create two subnets. One Public subnet and one private subnet.

**Step 1: Create a VPC**

1. Go to **VPC Dashboard** → Click **"Your VPCs"**
2. Click **"Create VPC"**
3. Choose **VPC only**
4. Set:
   * **Name tag:** vpc-virginia
   * **IPv4 CIDR block:** 192.168.0.0/24
5. Leave everything else as default (or modify if needed)
6. Click **"Create VPC"**

**Step 2: Create the Subnets**

**🔹 Private Subnet: 192.168.0.0/28**

1. Go to **Subnets** → Click **"Create subnet"**
2. Choose your **VPC: MyVPC**
3. Subnet settings:
   * **Subnet name:** Pri-Sub-virginia
   * **Availability Zone:** (Pick one, e.g., us-east-1a)
   * **IPv4 CIDR block:** 192.168.0.0/28
4. Click **Create subnet**

**🔹 Public Subnet: 192.168.0.16/28**

1. Still in **Subnets** → Click **"Create subnet"**
2. Same **VPC: MyVPC**
3. Subnet settings:
   * **Subnet name:** PublicSubnet
   * **Availability Zone:** (Can be same or different)
   * **IPv4 CIDR block:** 192.168.0.16/28
4. Click **Create subnet**

**Step 3: Make Public Subnet Truly Public**

To enable internet access in the **public subnet**, you need:

**🔹 3.1. Create an Internet Gateway (IGW)**

1. Go to **Internet Gateways** → Click **"Create internet gateway"**
2. Name it: MyIGW → Click **Create**
3. Select the IGW → Click **Actions > Attach to VPC** → Select MyVPC

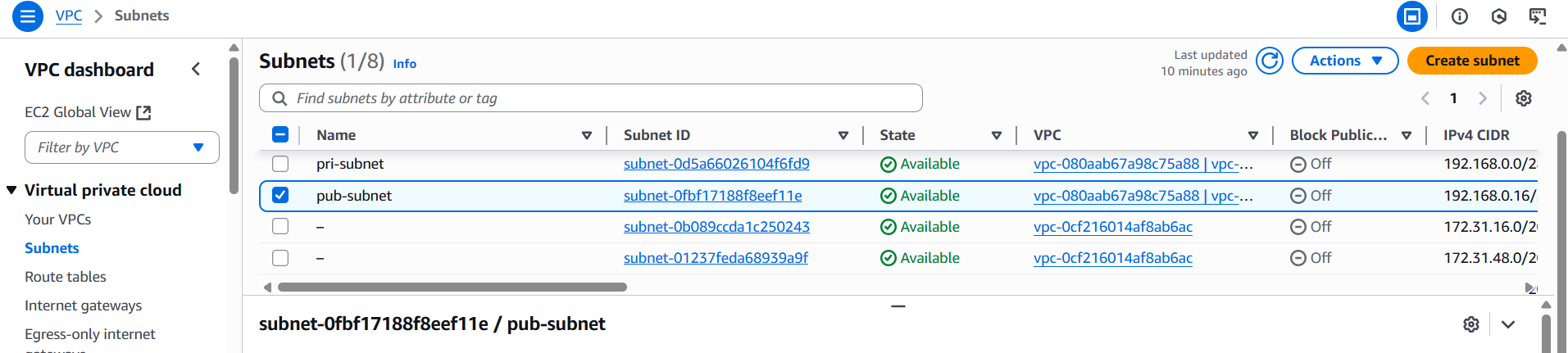
**🔹 3.2. Update Route Table**

1. Go to **Route Tables**
2. Find the route table associated with your **Public Subnet** (or create one if needed)
3. Select the route table → **Edit routes**
4. Add:
   * **Destination:** 0.0.0.0/0
   * **Target:** Select your Internet Gateway (MyIGW)
5. Save routes

Note: Create two Route Tables for each Subnets, one for public subnet and another for private subnet

**🔹 3.3. Associate Public Subnet with the Route Table**

1. In the route table → Go to **Subnet Associations**
2. Click **Edit subnet associations**
3. Select **PublicSubnet** and **Save**

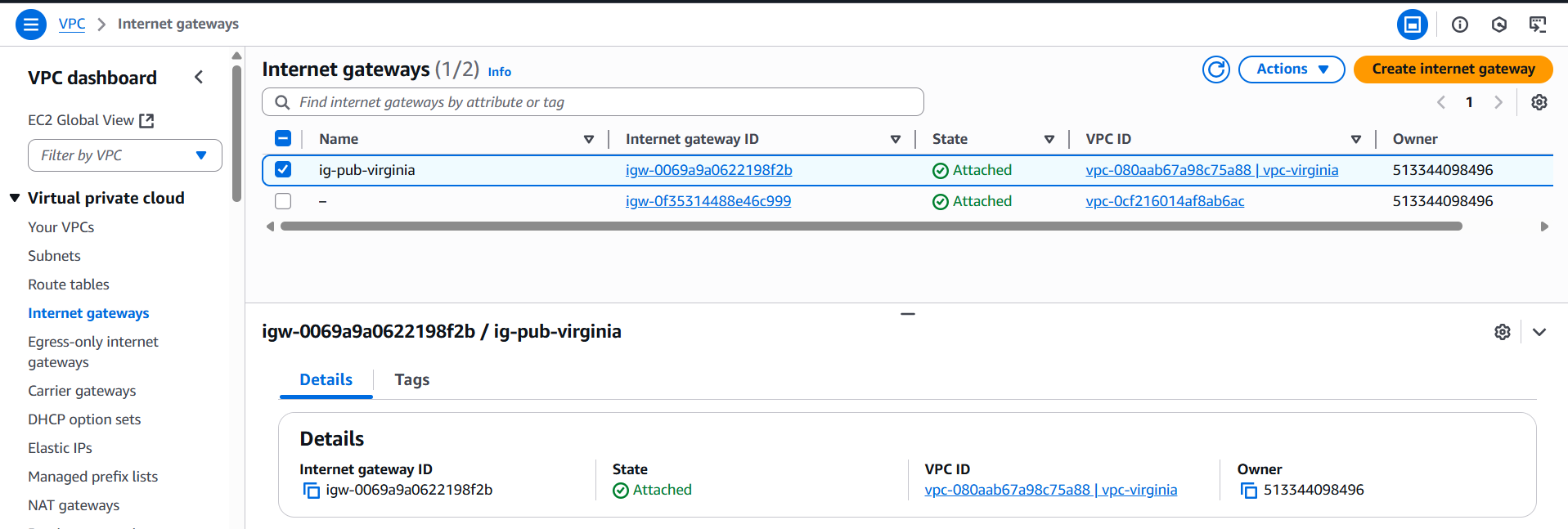




3)Provide the IGW to the vpc.

**Create an Internet Gateway (IGW)**

1. Go to **Internet Gateways** → Click **"Create internet gateway"**
2. Name it: MyIGW → Click **Create**
3. Select the IGW → Click **Actions > Attach to VPC** → Select MyVPC



4)Create One public RT and one private RT.

**Step 1: Create Public Route Table**

1. Go to the **VPC Dashboard** → **Route Tables**
2. Click **Create route table**
3. Enter:
   * **Name tag:** rt-pub-virginia
   * **VPC:** Select your VPC (e.g., MyVPC)
4. Click **Create route table**

**Step 2: Add Internet Gateway Route to PublicRT**

1. Select the newly created **PublicRT**
2. Go to **Routes** tab → Click **Edit routes**
3. Click **Add route**
4. Set:
   * **Destination:** 0.0.0.0/0
   * **Target:** Select your **Internet Gateway (IGW)** (you must have created and attached one to the VPC)
5. Click **Save routes**

**Step 3: Associate PublicRT with Public Subnet**

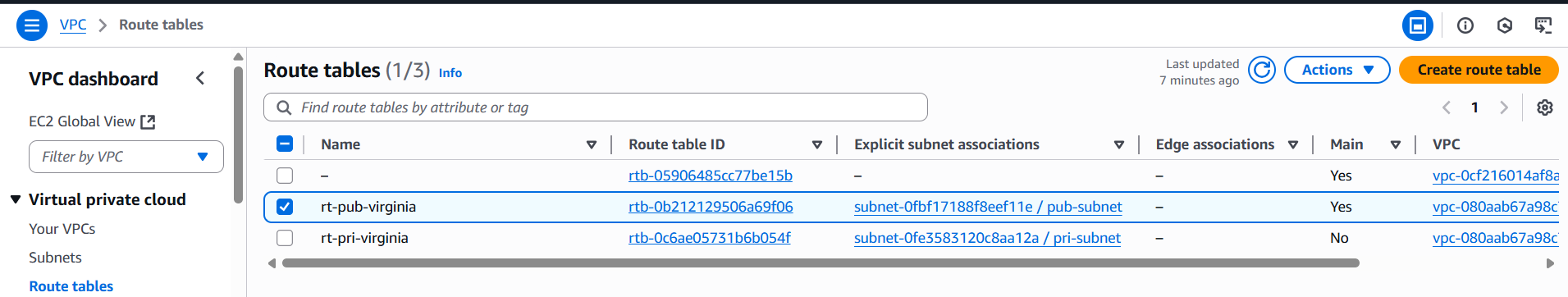
1. In the **PublicRT** details, go to **Subnet Associations** tab
2. Click **Edit subnet associations**
3. Select your **Public subnet** (e.g., 192.168.0.16/28)
4. Click **Save**

**Step 4: Create Private Route Table**

1. Go back to **Route Tables**
2. Click **Create route table**
3. Enter:
   * **Name tag:** rt-pri-virginia
   * **VPC:** Select your VPC (same as before)
4. Click **Create route table**

**Step 5: Associate PrivateRT with Private Subnet**

1. Select the newly created **PrivateRT**
2. Go to **Subnet Associations** tab
3. Click **Edit subnet associations**
4. Select your **Private subnet** (e.g., 192.168.0.0/28)
5. Click **Save**





5)Deploy NAT gateway on public subnet and attach the NAT gateway to private subnet.

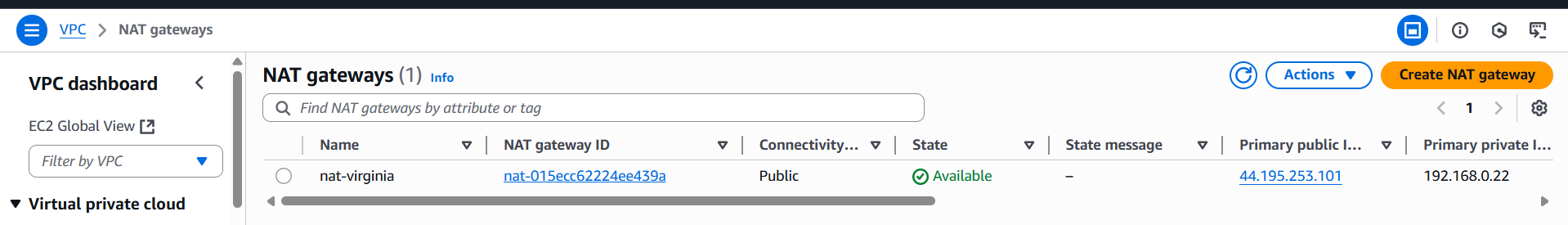
**Step 1: Create an Elastic IP for the NAT Gateway**

1. Go to the **EC2 Dashboard** → **Elastic IPs** (on the left sidebar)
2. Click **Allocate Elastic IP address**
3. Click **Allocate**

**Step 2: Create the NAT Gateway in the Public Subnet**

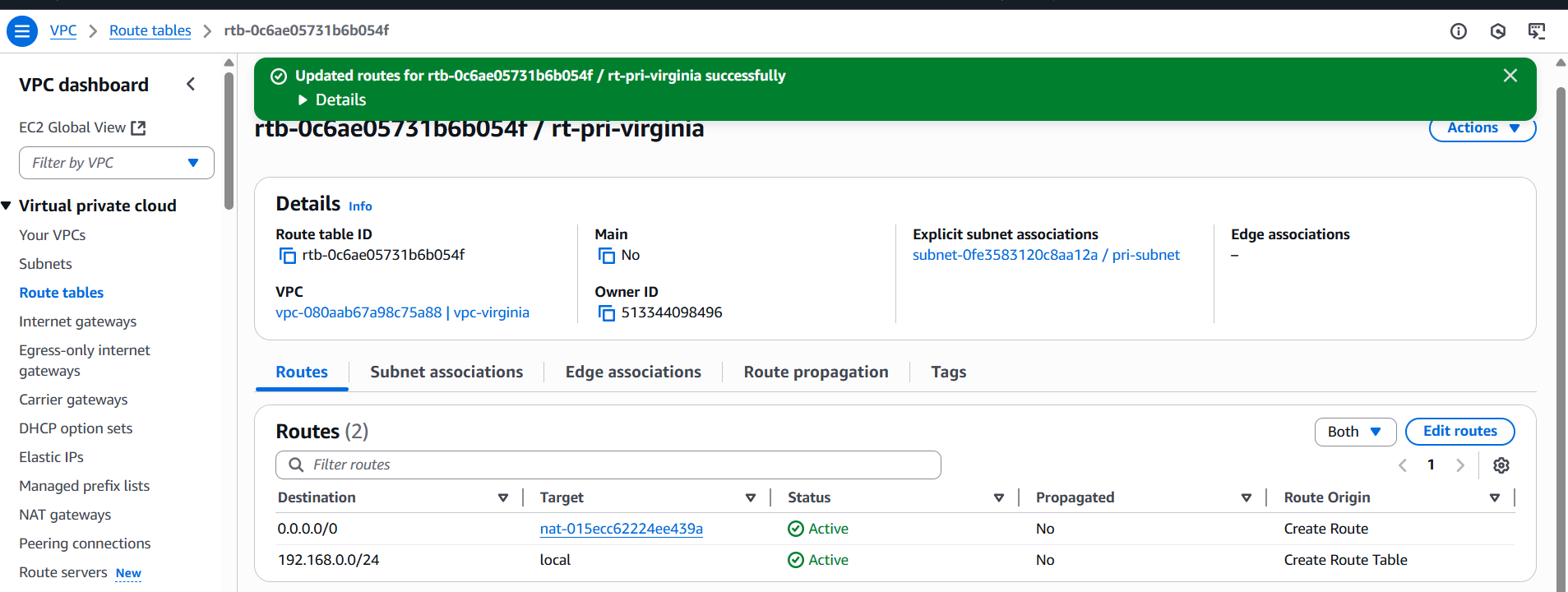
1. Go to the **VPC Dashboard** → **NAT Gateways**
2. Click **Create NAT gateway**
3. Configure:
   * **Subnet:** Select your **Public subnet** (192.168.0.16/28)
   * **Elastic IP allocation ID:** Select the Elastic IP you just allocated
4. Click **Create NAT gateway**

It may take a few minutes to become **available**.



**Step 3: Update the Private Route Table to Use the NAT Gateway**

1. Go to **VPC Dashboard** → **Route Tables**
2. Select your **PrivateRT** (the route table associated with the private subnet)
3. Go to the **Routes** tab → Click **Edit routes**
4. Click **Add route**
5. Enter:
   * **Destination:** 0.0.0.0/0
   * **Target:** Select your **NAT Gateway ID** from the dropdown
6. Click **Save routes**





6)Create Two instances,one in public subnet and one in private subnet.

**Step 1: Launch EC2 Instance in Public Subnet**

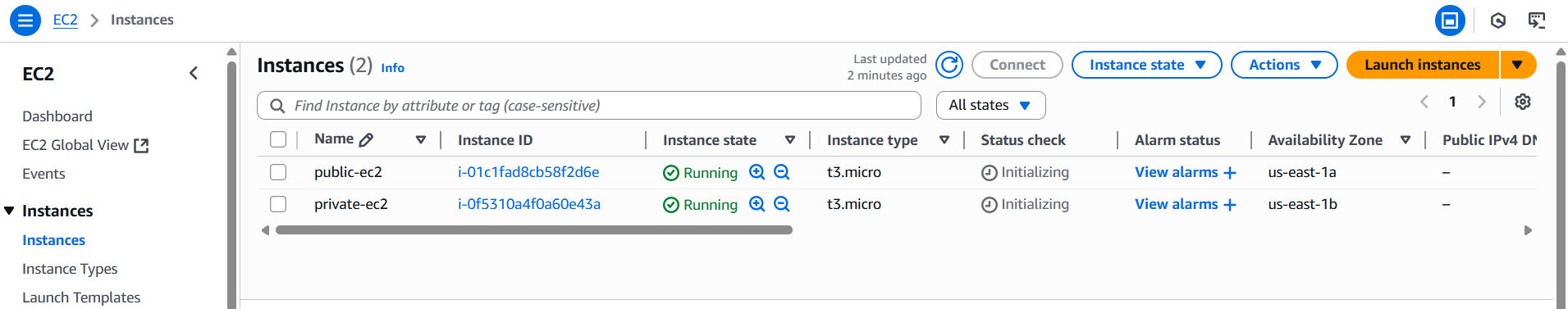
1. Go to **EC2 Dashboard** → Click **Launch Instances**
2. Configure instance:
   * **Name:** Public-ec2
   * **AMI:** Choose an Amazon Linux 2 or Ubuntu AMI
   * **Instance type:** (e.g., t3.micro)
3. **Configure Instance Details:**
   * **Network:** Select your VPC (e.g., MyVPC)
   * **Subnet:** Select your **Public subnet** (192.168.0.16/28)
   * **Auto-assign Public IP:** Enable (important for internet access)
4. **Add Storage:** Leave defaults or adjust
5. **Add Tags:** Optional
6. **Configure Security Group:**
   * Create or select a security group allowing SSH (port 22) from your IP, and HTTP/HTTPS if needed
7. **Review and Launch** the instance
8. Choose or create a key pair for SSH access and launch the instance

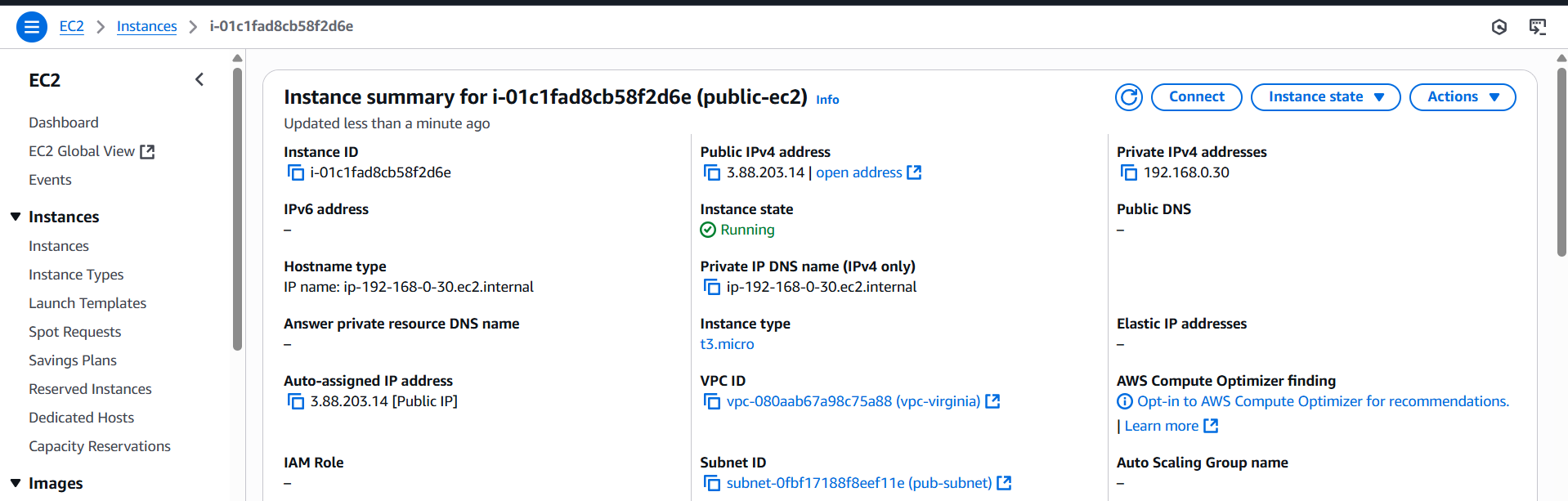
**Step 2: Launch EC2 Instance in Private Subnet**

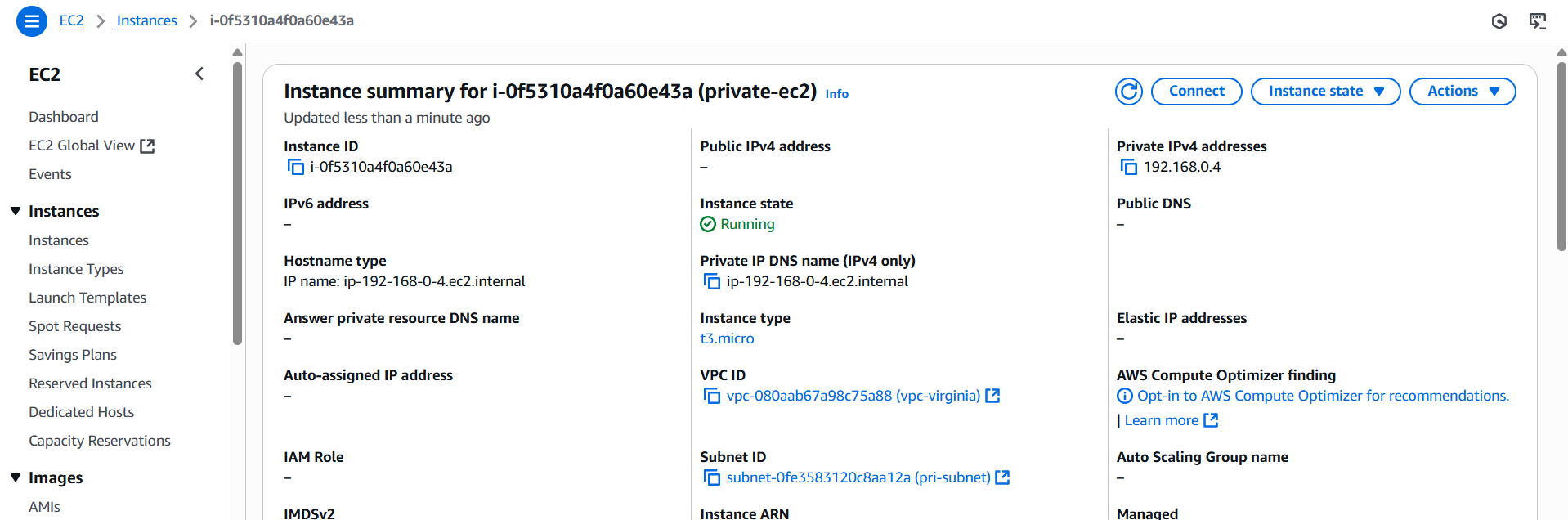
1. Go again to **Launch Instances**
2. Configure instance:
   * **Name:** Private-ec2
   * **AMI:** Same as above or as needed
   * **Instance type:** e.g., t2.micro
3. **Configure Instance Details:**
   * **Network:** Select your VPC (MyVPC)
   * **Subnet:** Select your **Private subnet** (192.168.0.0/28)
   * **Auto-assign Public IP:** **Disable** (private subnet instances should NOT have public IP)
4. **Add Storage, Tags:** As desired
5. **Configure Security Group:**
   * Create or select a security group allowing SSH only from the **public instance’s security group** (for example), or from a Bastion host if you have one
6. **Review and Launch**
7. Choose the same or another key pair and launch

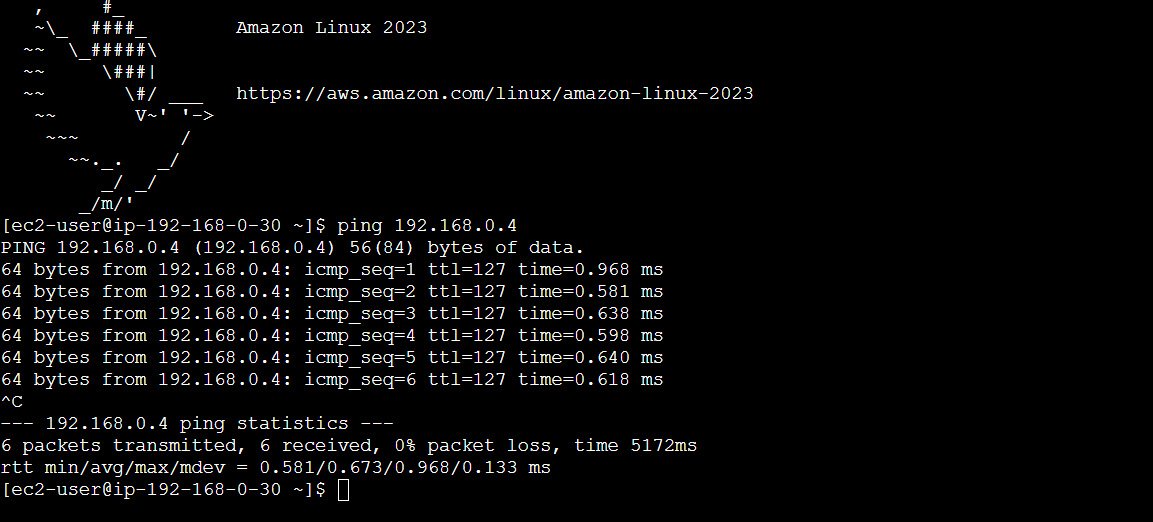
**Step 3: Accessing the Instances**

* **Public Instance:**  
  You can SSH directly via its **public IP** (because it has a public IP and the subnet is public).
* **Private Instance:**  
  It **has no public IP**, so you **cannot SSH directly** from the internet.  
  To access it, you typically:
  + SSH into the **Public Instance** (acting as a bastion host)
  + From the public instance, SSH into the private instance using its **private IP**









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

**Step 1: Connect to Your Instances**

**Public Instance**

* SSH directly using its **public IP**:

ssh -i "linux-test.pem" [ec2-user@3.88.203.14](mailto:ec2-user@3.88.203.14)

**Private Instance**

* SSH **through the public instance** (bastion/jump host):

Firstly, cat the linux.pem and copy the key then create a pem file using

Vi private.pem and paste the key

Give permissions chmod 700 private.pem

Next, connect using

ssh ec2-user@ 192.168.0.4

**Step 2: Install Apache Server on Both Instances**

Run these commands on **both** instances (public and private):

For **Amazon Linux 2** :

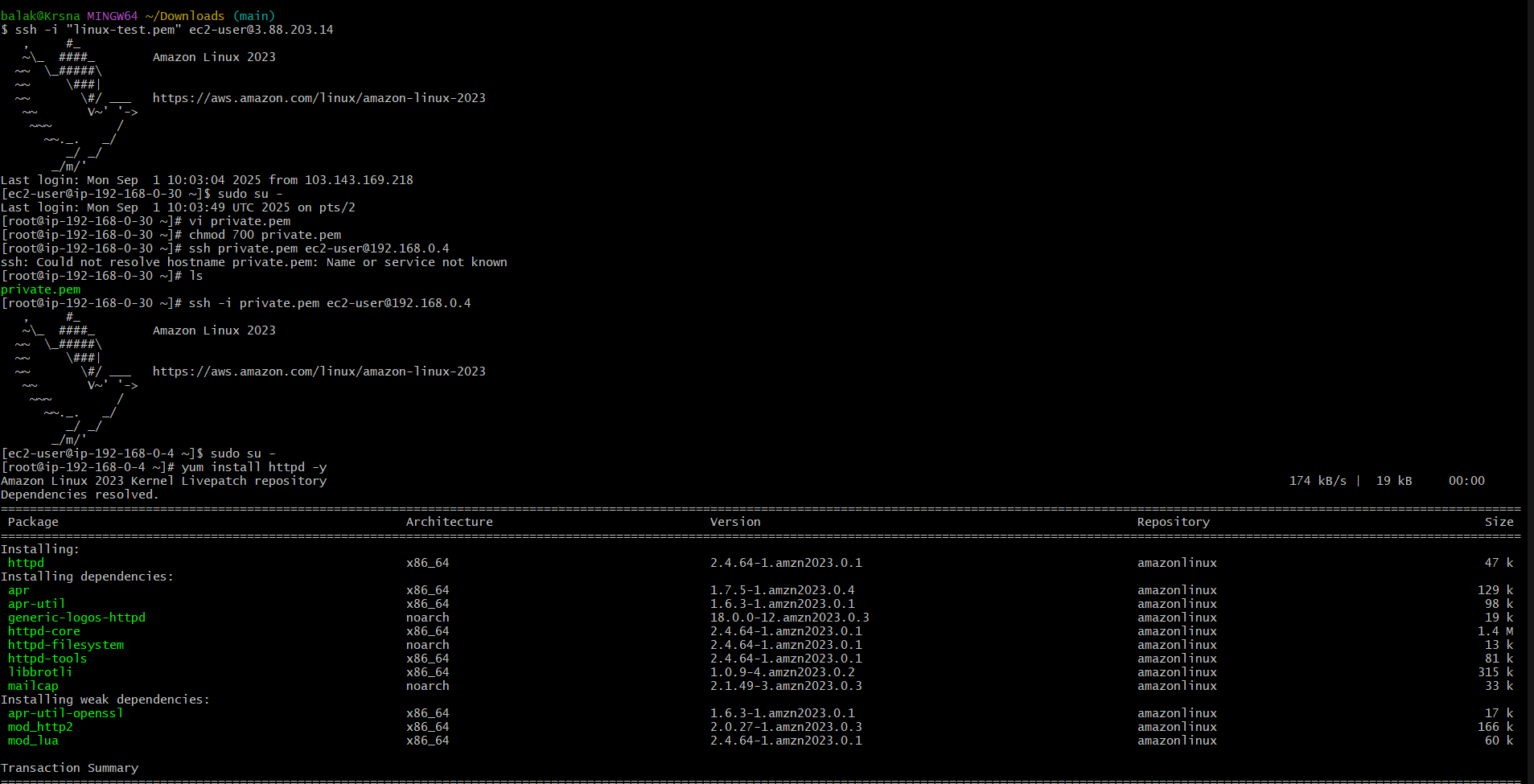
sudo yum update -y

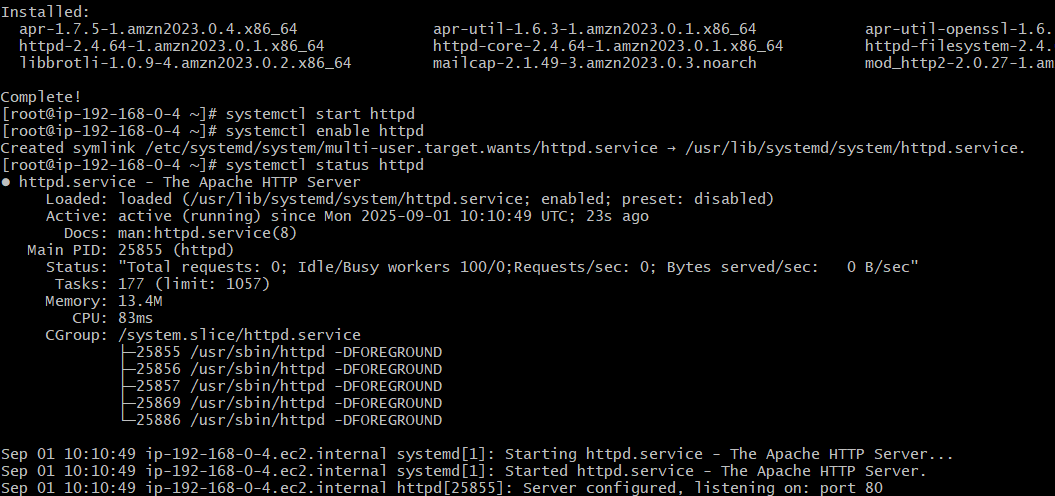
sudo yum install -y httpd

sudo systemctl start httpd

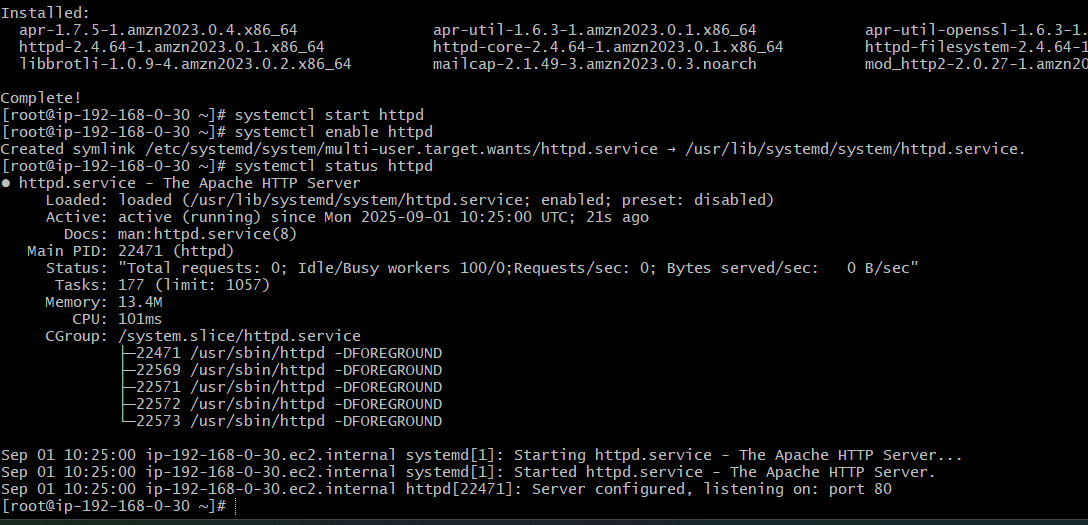
sudo systemctl enable httpd

Installed in private EC2 instance.





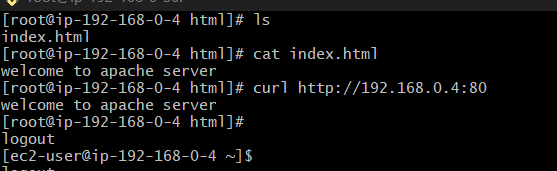
Installed in public EC2 instance.

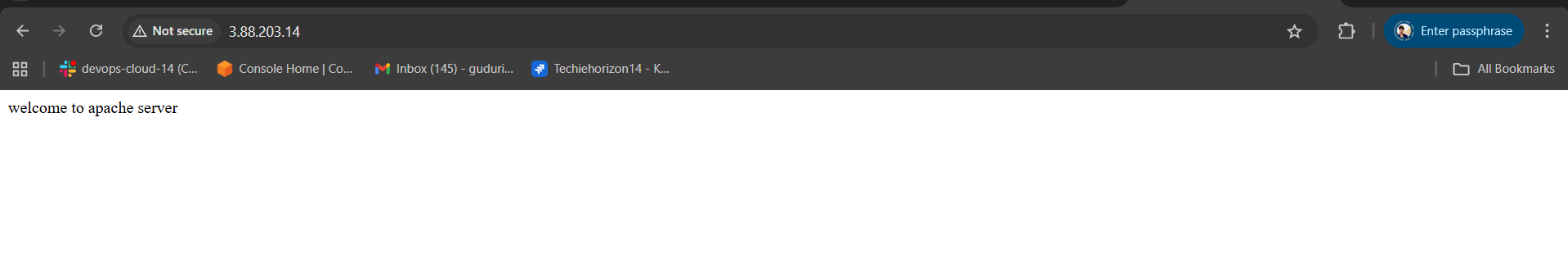


**Step 3: Create a Sample index.html File**

On each instance, create a simple web page to identify the instance:

echo "Welcome to the Apache Server" >> /var/www/html/index.html

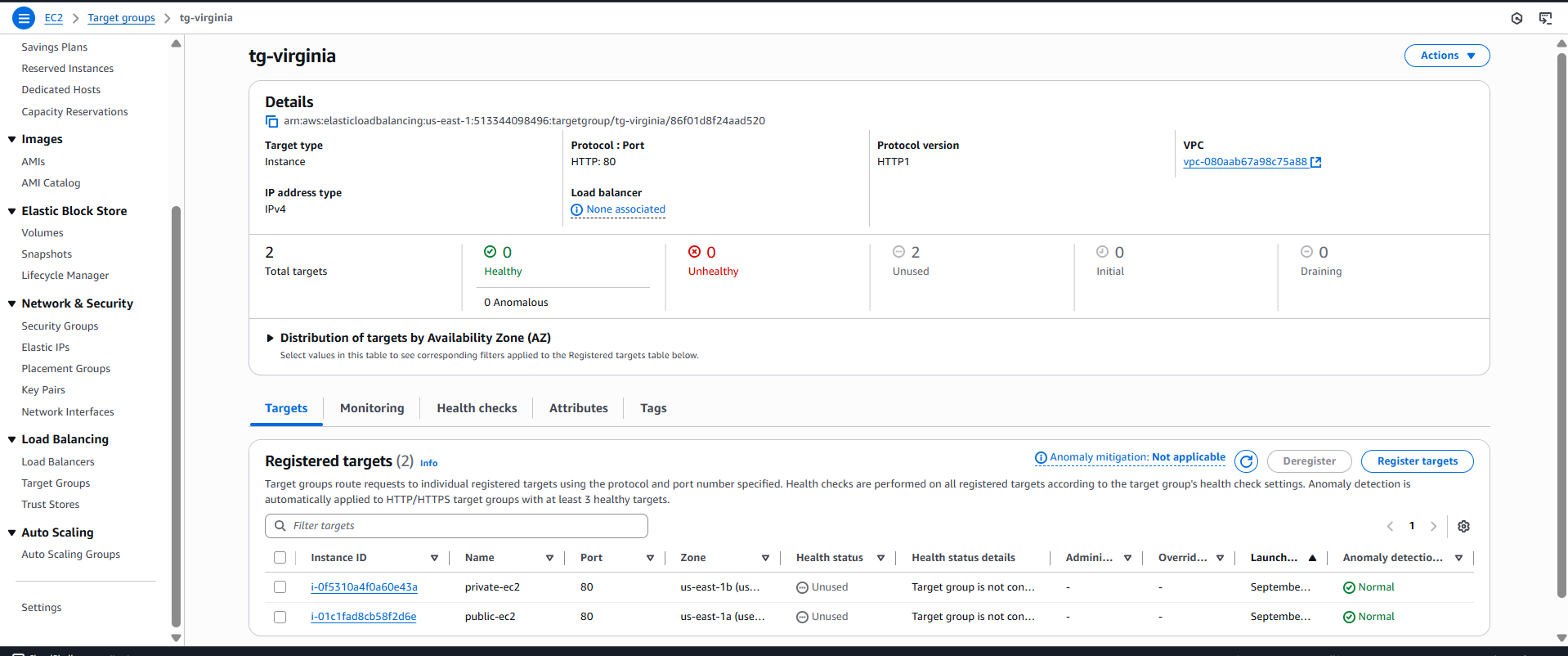




8)Create one application load balancer and attach the load balancer to both the ec2 instances.

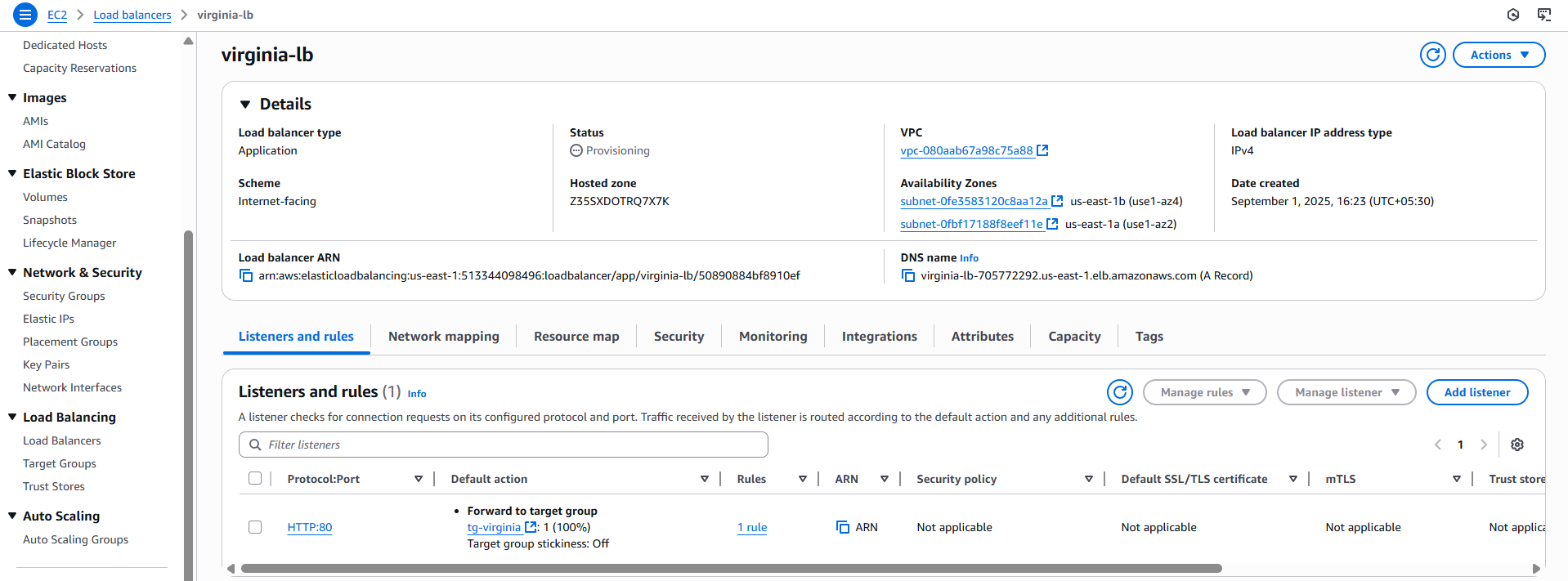
**Step 1: Create Target Group**

1. Go to **EC2 Dashboard → Target Groups**
2. Click **Create target group**
3. Choose:
   * **Target type:** Instance
   * **Protocol:** HTTP
   * **Port:** 80
   * **VPC:** Select your VPC
4. Name: tg-virginia
5. Click **Next**
6. Register targets:
   * Select **both EC2 instances**
   * Click **Include as pending**
7. Click **Create target group**



**Step 2: Create Application Load Balancer**

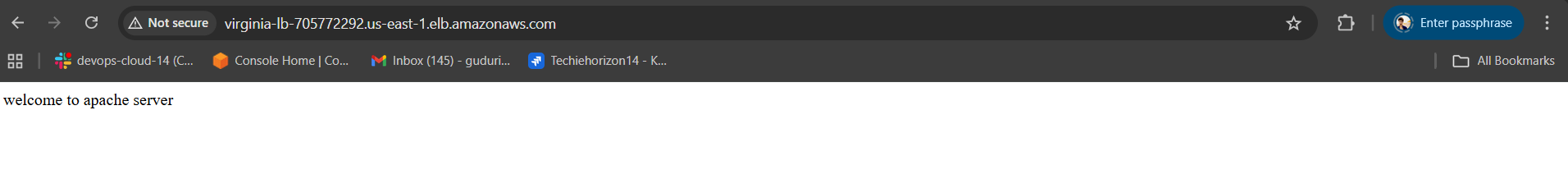
1. Go to **EC2 Dashboard → Load Balancers**
2. Click **Create Load Balancer → Application Load Balancer**
3. Configure:
   * **Name:** virginia-lb
   * **Scheme:** Internet-facing
   * **IP address type:** IPv4
4. **Network Mapping:**
   * Select your VPC
   * Choose **2 subnets** (at least 1 public and 1 private — must be in different AZs or enable cross-zone load balancing)
5. **Security Groups:**
   * Select the default SG which we have access to http 80
6. **Listeners:**
   * HTTP (port 80) → Forward to tg-virginia
7. Click **Create Load Balancer**



**Step 3: Test the Load Balancer**

* Wait for the ALB to become **Active**
* Go to **EC2 → Load Balancers → virginia-lb**
* Copy the **DNS name** and open in your browser:

virginia-lb-705772292.us-east-1.elb.amazonaws.com



9)Store Application load balancer logs to s3.

**🔹 Step 1: Create or Choose an S3 Bucket**

1. Go to the **S3 Console**
2. Click **Create bucket** (or use an existing one)
3. Name : s3-krsna
4. Uncheck **Block all public access** only if you want others to access logs (optional)
5. Click **Create bucket**

**🔹 Step 2: Update the S3 Bucket Policy for ALB Logging**

1. Go to **S3 Console**
2. Open bucket: s3-krsna
3. Go to the **Permissions** tab
4. Click **Edit** under **Bucket policy**
5. Paste the following policy:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowALBAccessLogs",

"Effect": "Allow",

"Principal": {

"Service": "logdelivery.elasticloadbalancing.amazonaws.com"

},

"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::s3-krsna/AWSLogs/\*",

"Condition": {

"StringEquals": {

"s3:x-amz-acl": "bucket-owner-full-control"

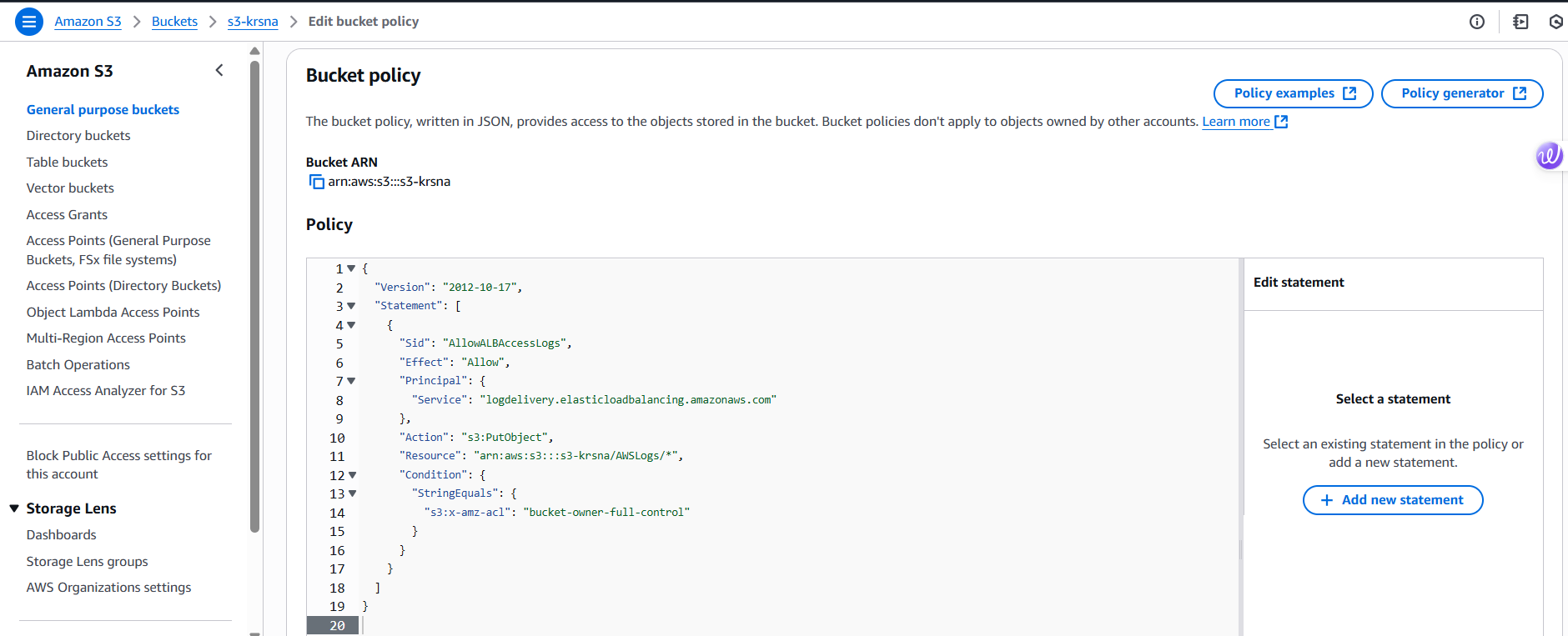
}

}

}

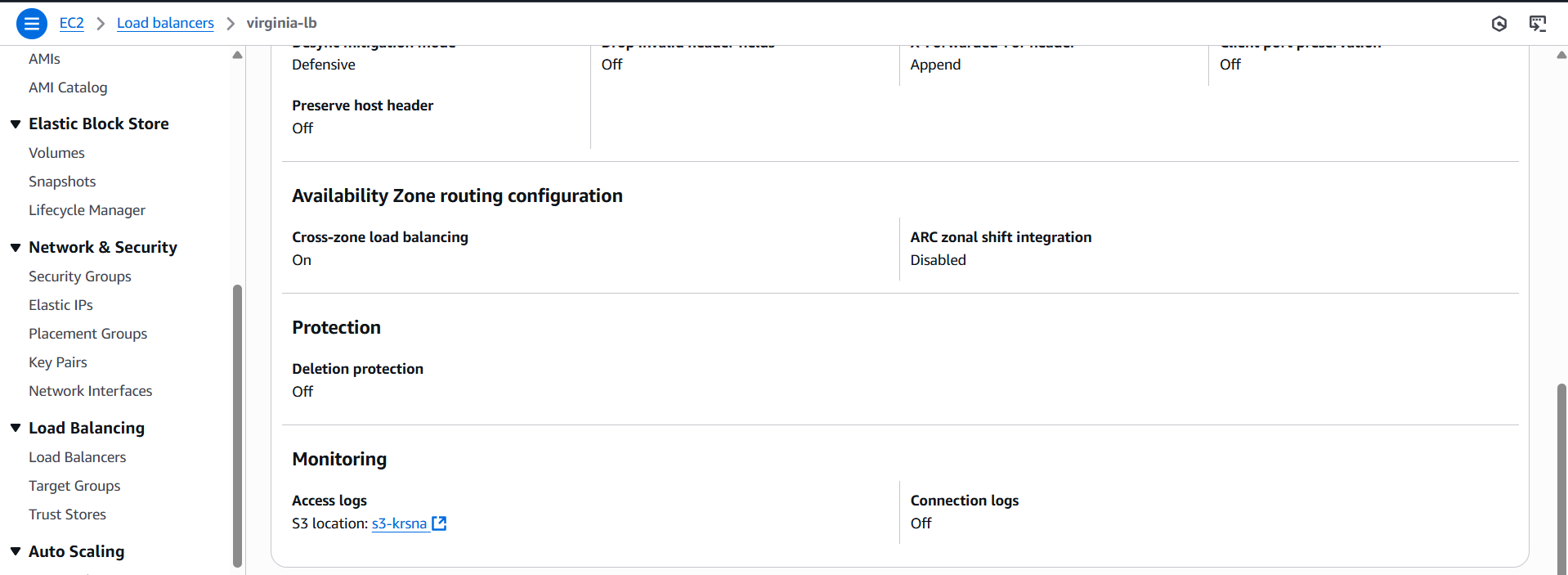
]

}



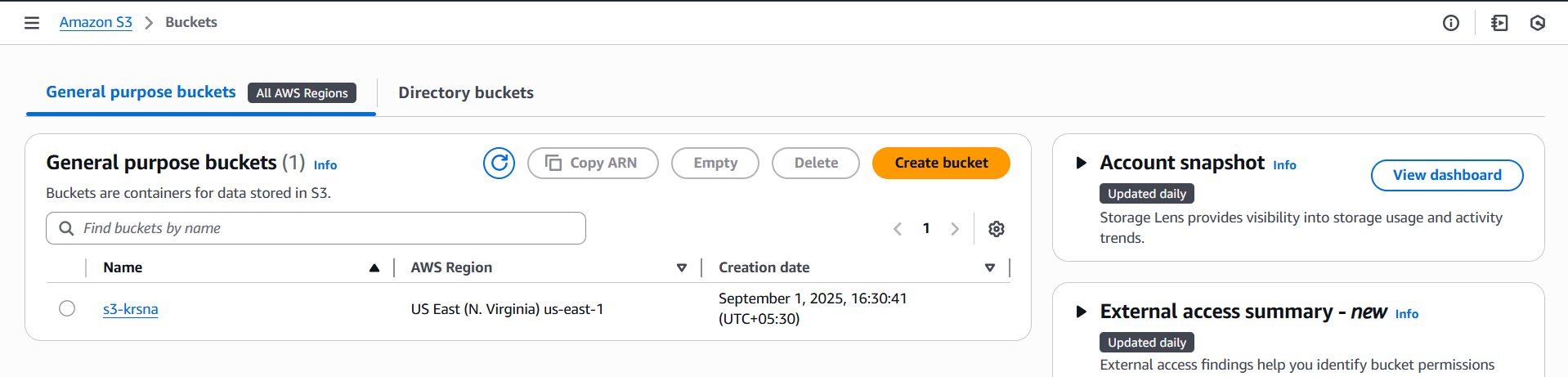
**🔹 Step 3: Enable Access Logging on Your ALB**

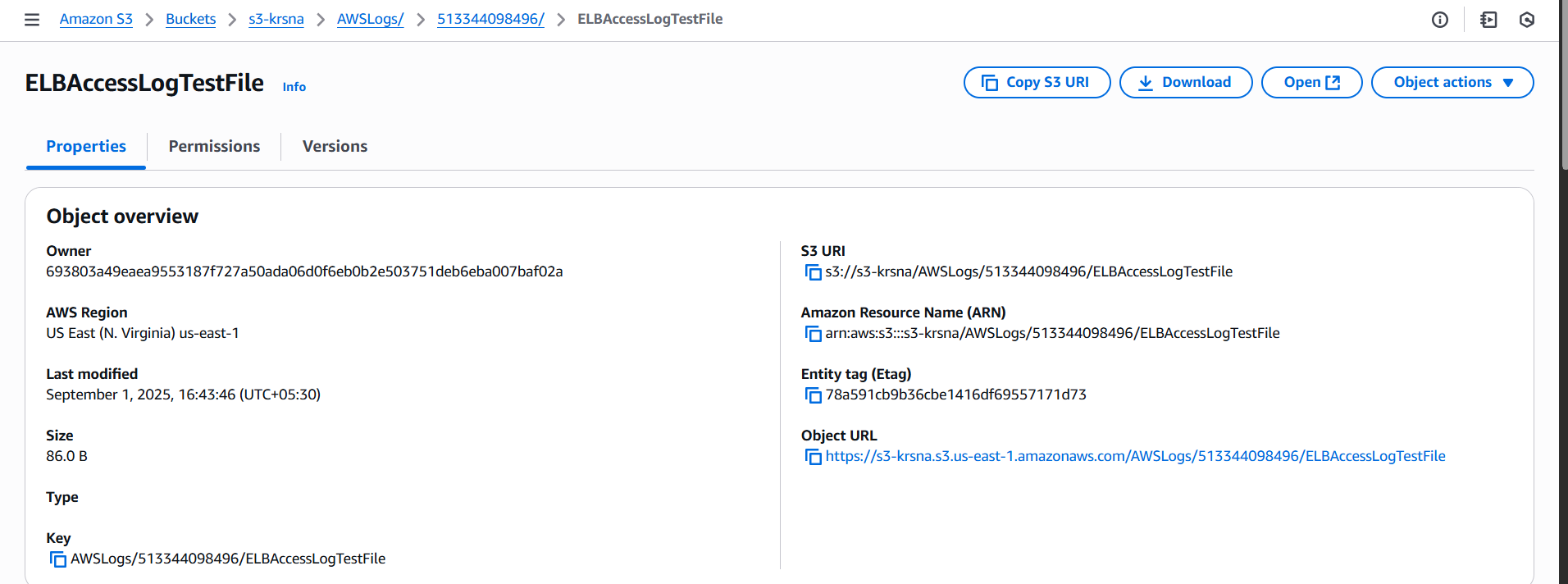
1. Go to **EC2 Console** → **Load Balancers**
2. Select your ALB (virginia-lb)
3. Click the **Attributes** tab
4. Click **Edit attributes**
5. Under **Access logs**:
   * Enable logging
   * **S3 bucket name:** s3-krsna
   * **Prefix (optional):** alb-logs/ (this is the folder inside your bucket where logs go)
6. Click **Save**

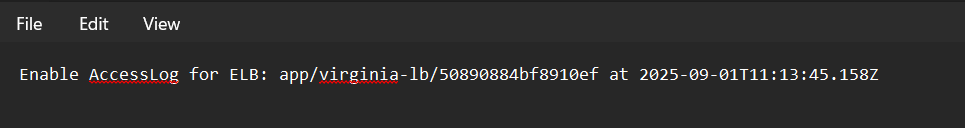


**🔹 Step 4: Verify Logging**

* It may take **5–15 minutes** before the first logs start showing up in S3.
* Check the S3 bucket under the prefix you set (e.g., alb-logs/AWSLogs/your-account-id/...)



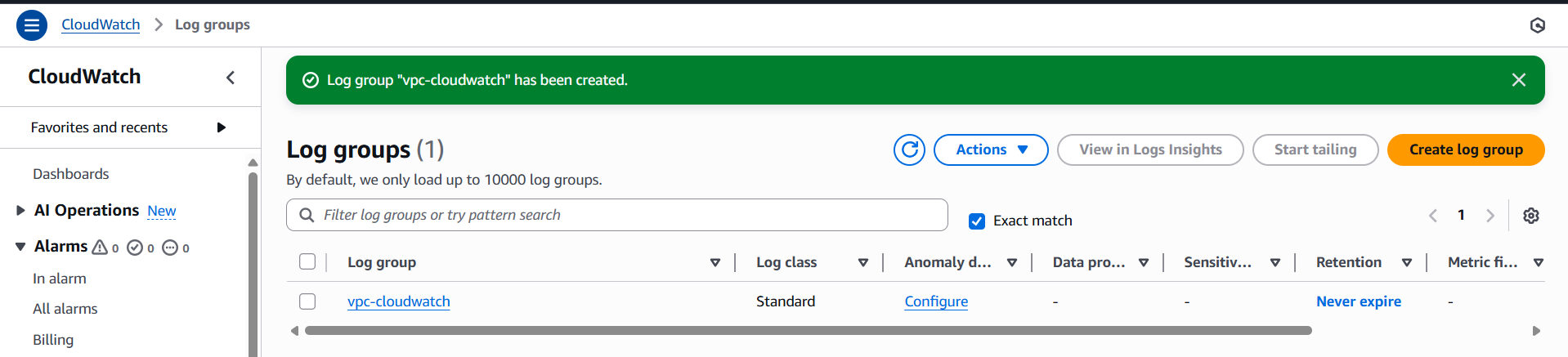




10)Store the VPC flow logs to CloudWatch group.

**Step 1: Create a CloudWatch Log Group**

1. Go to the **CloudWatch Console**
2. Click **Log groups**
3. Click **Create log group**
4. Enter a name: vpc-cloudwatch
5. Choose a retention period (e.g., 1 week or Never Expire)
6. Click **Create**



**🔹 Step 2: Create IAM Role for VPC Flow Logs (if not already created)**

1. Go to the **IAM Console** → **Roles**
2. Click **Create role**
3. **Trusted entity type:** Choose AWS service
4. **Use case:** Select EC2
5. Click **Next**
6. Attach the following policy:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents",

"logs:DescribeLogGroups",

"logs:DescribeLogStreams"

],

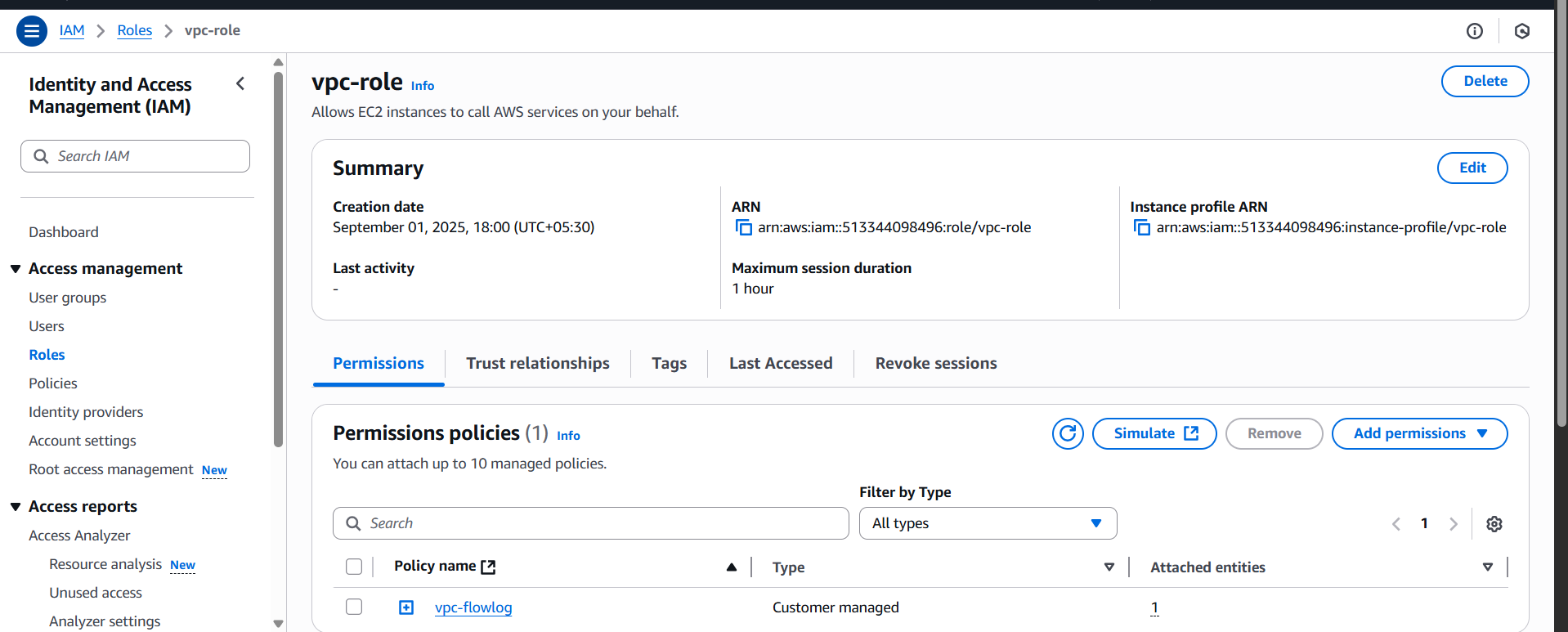
"Resource": "\*"

}

]

}

1. Name the role: vpc-role
2. Click **Create role**



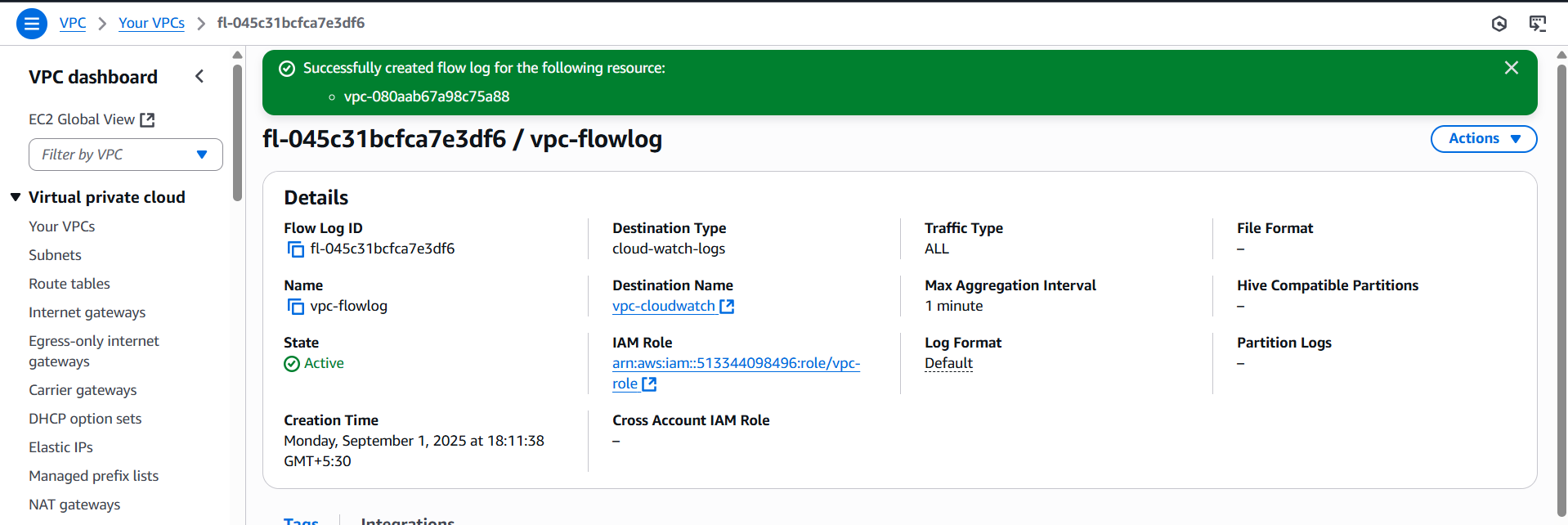
**🔹 Step 3: Enable VPC Flow Logs**

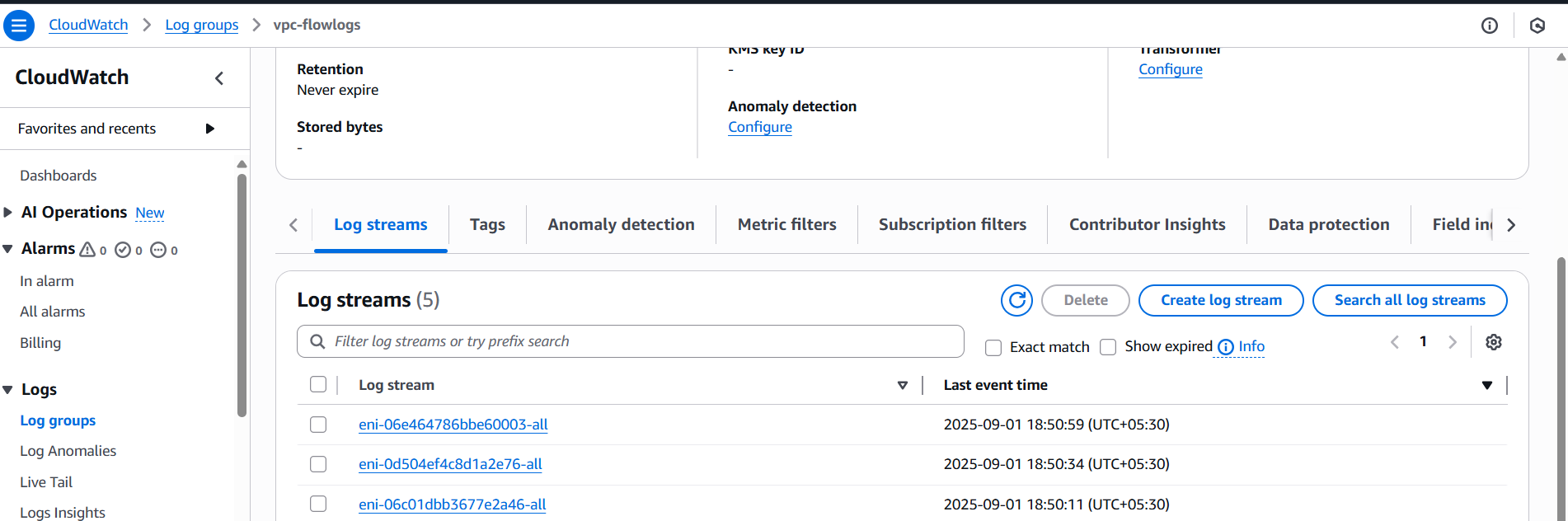
1. Go to the **VPC Dashboard**
2. Click **Your VPCs**
3. Select your VPC (vpc-virginia)
4. Go to **Flow Logs** tab → Click **Create flow log**

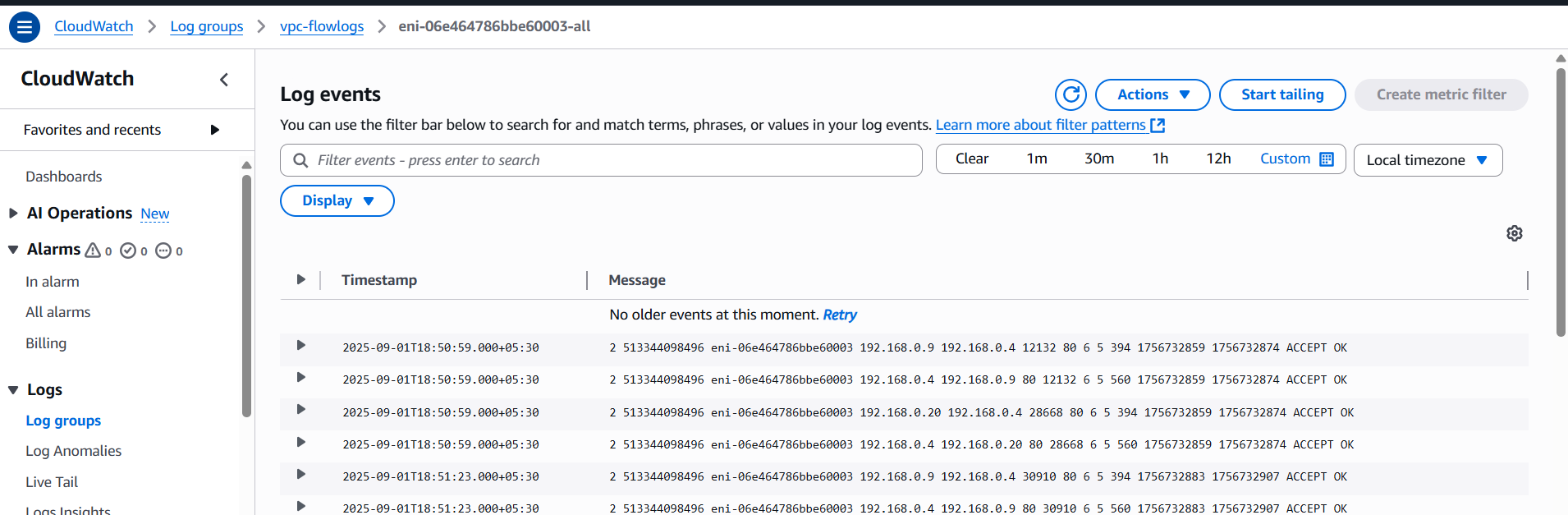
**Configure the Flow Log:**

* **Filter:** All (or Accept / Reject based on what you need)
* **Destination:** Send to CloudWatch Logs
* **Log group:** /vpc-cloudwatch (select the one you created)
* **IAM role:** vpc-role

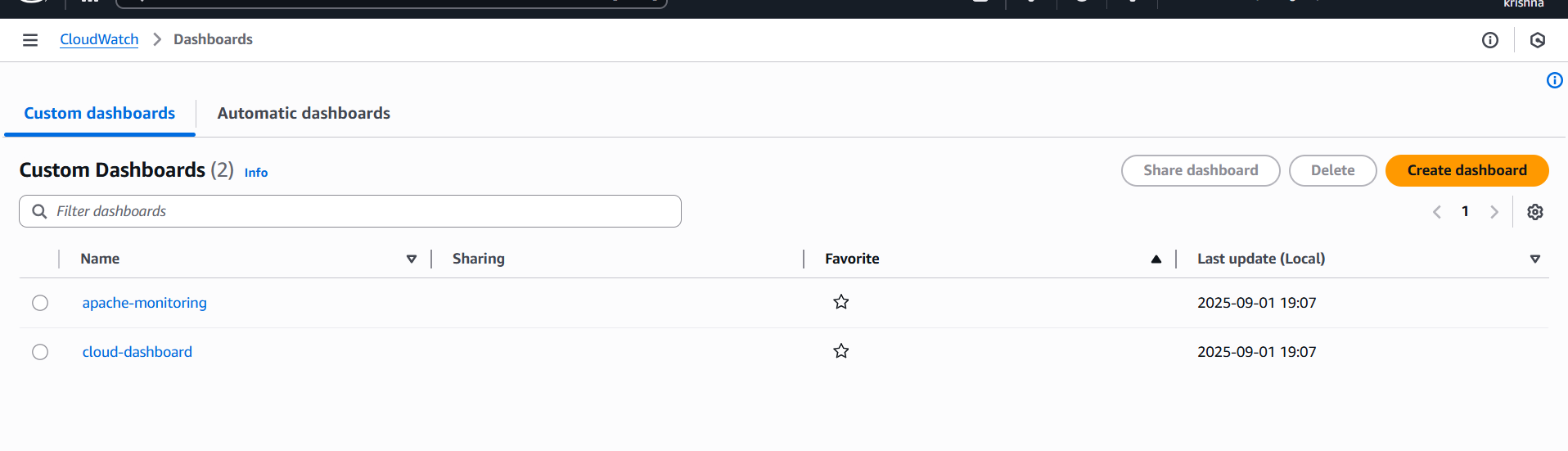
1. Click **Create flow log**

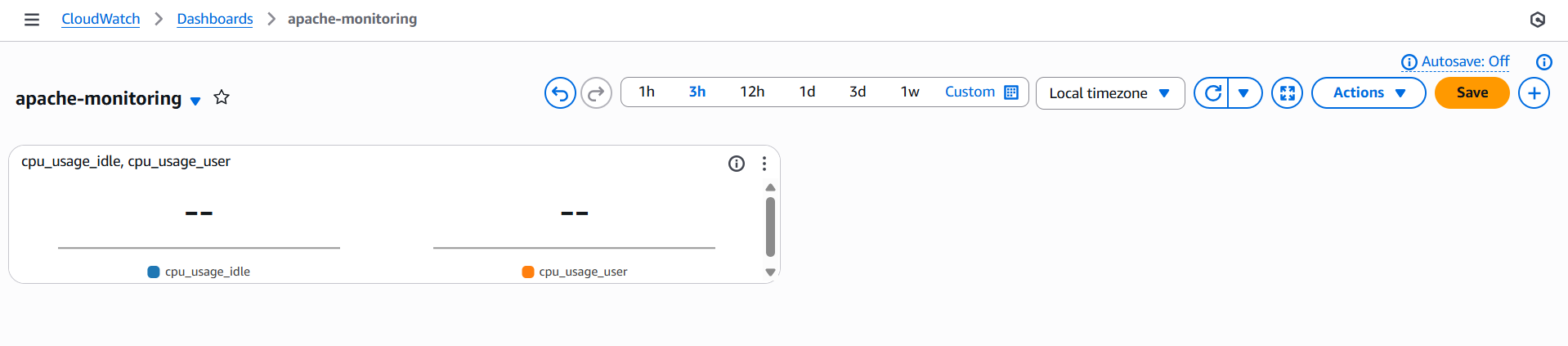


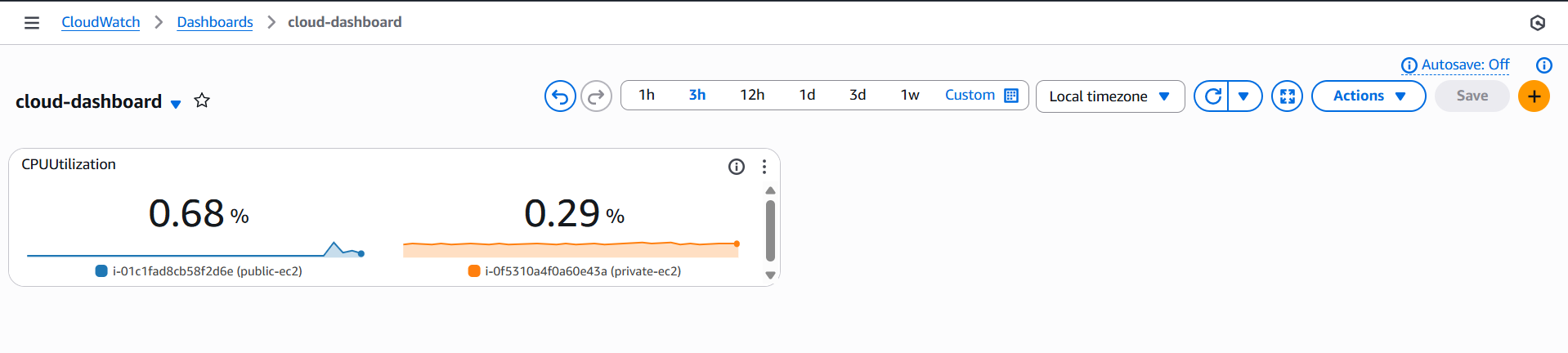




11)Create Monitoring Dashboards to monitor CPU utilization and to monitor Apache service.







12)CPU utilizations more than 70% then it should trigger Autoscaling and launch new instance.

