1. Create one VPC in N.virginia region.
2. Go to AWS management console
3. Select VPC
4. To Create a VPC :

* Click “create VPC”

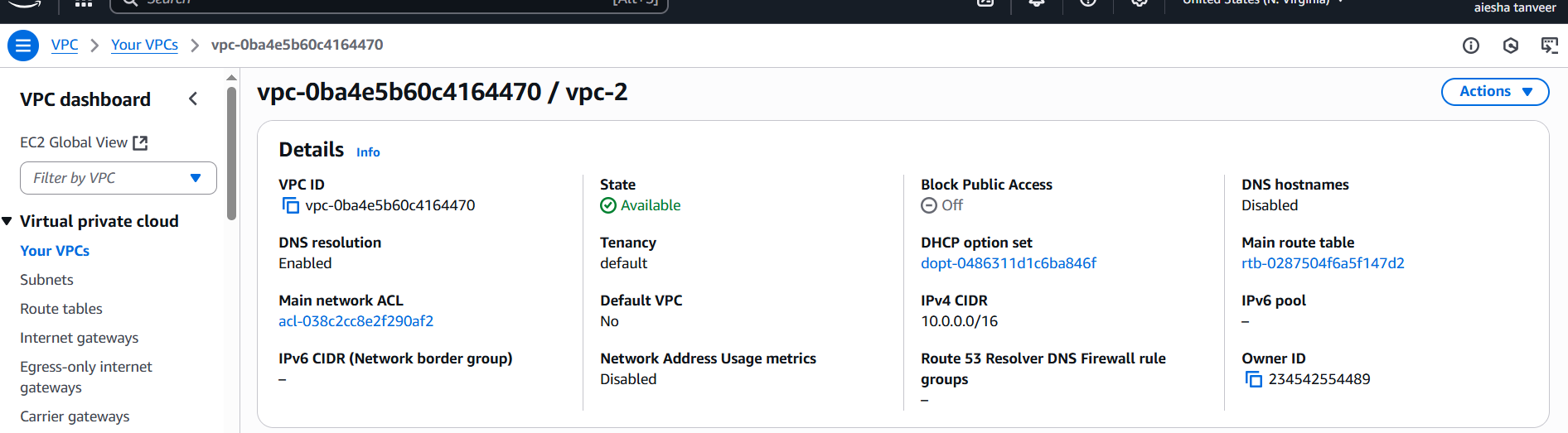
Name: VPC-2

IPV4 CIDR -10.0.0.0/16

Leave IPV6 CIDR as “NO IPV6 CIDR block”

Tenancy: Default

Click create VPC.



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

* **creating public subnet**

**Go to subnets →**click “**create subnet**”

Choose your VPC : VPC-2

Subnet settings :

**.**  **subnet name:** web-pub-subnet

**. availability zone:** us-east-1a

**. IPV4 CIDR block:** 10.0.1.0/24

click create subnet.

* **creating private subnet**

**Go to subnets →** click **“create subnet”**

Choose your VPC: VPC-2

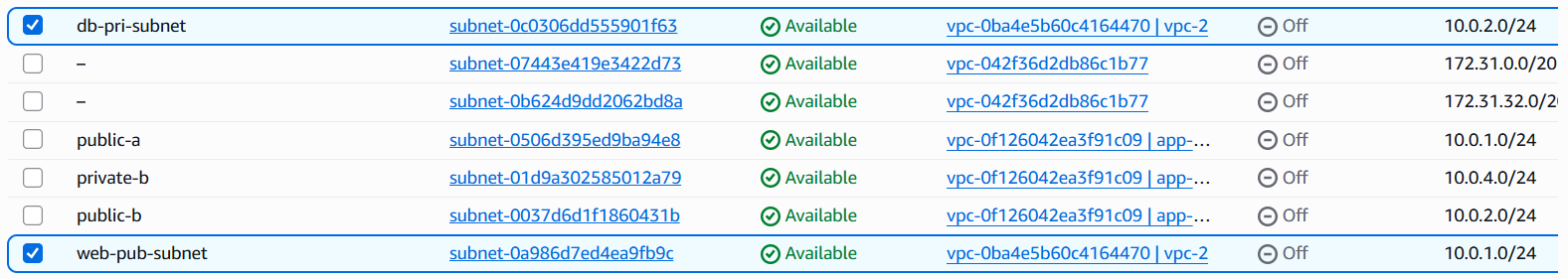
Subnet settings:

**. subnet name:** db-pri-subnet

**. availability zone:** us-east-1b

**. IPV4 CIDR block:** 10.0.2.0/24

Click create subnet.

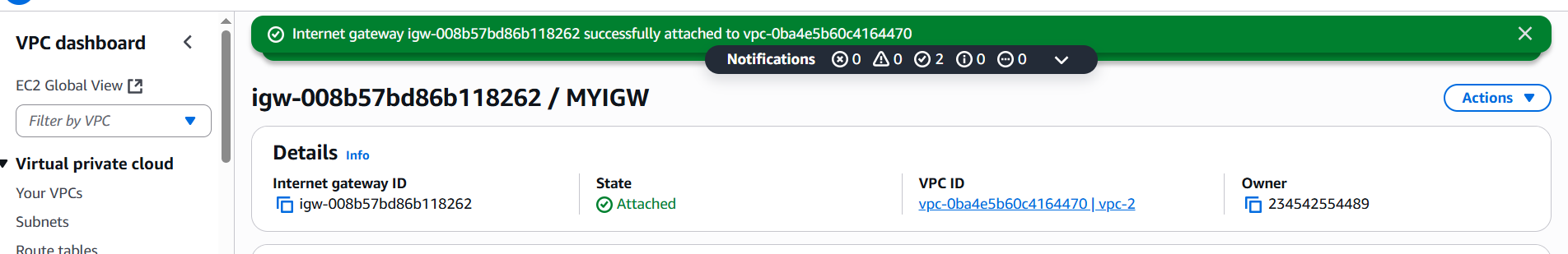


1. Provide the IGW to the VPC.

Go to **Internet gateway →** click **“create internet gateway”**

Name: MYIGW → click create

Select IGW → click **actions > Attach to VPC** →Select **VPC-2**



1. Create one public RT and one private RT.

* **Creating public route table :**

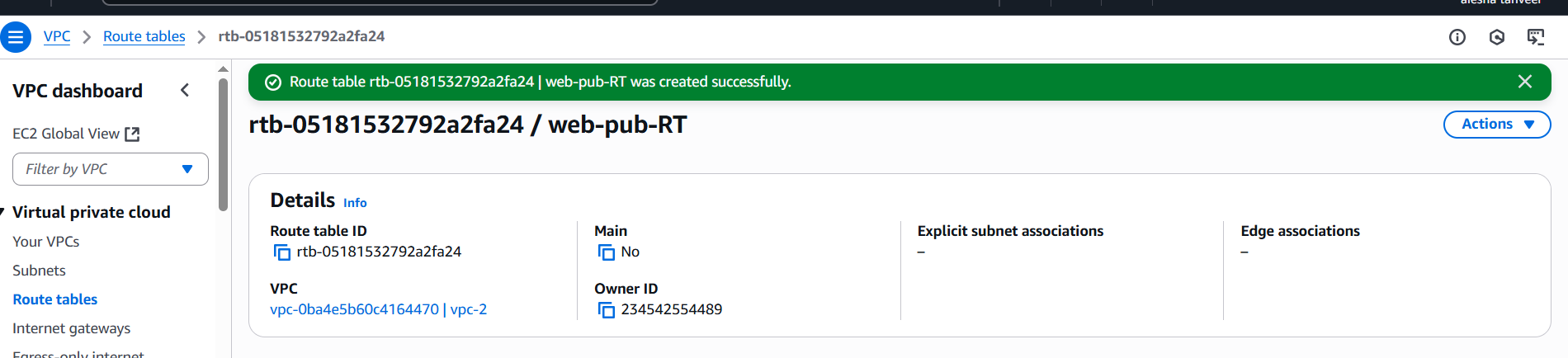
In VPC → Go to Route tables

Click **create route table**

Enter → Name :

VPC : Select your VPC (Select VPC-2)

Click create Route table.



* **Creating private route table :**

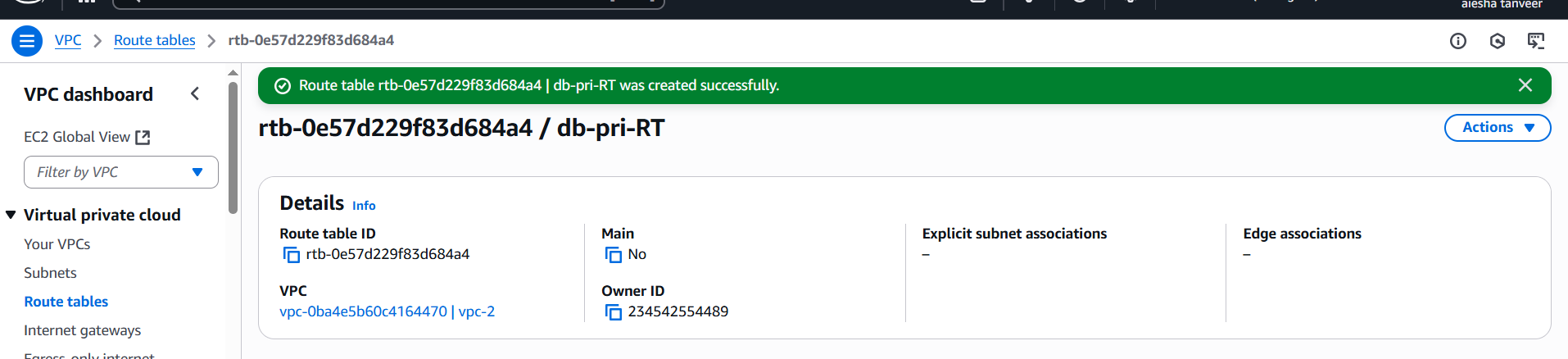
In VPC → Go to Route tables

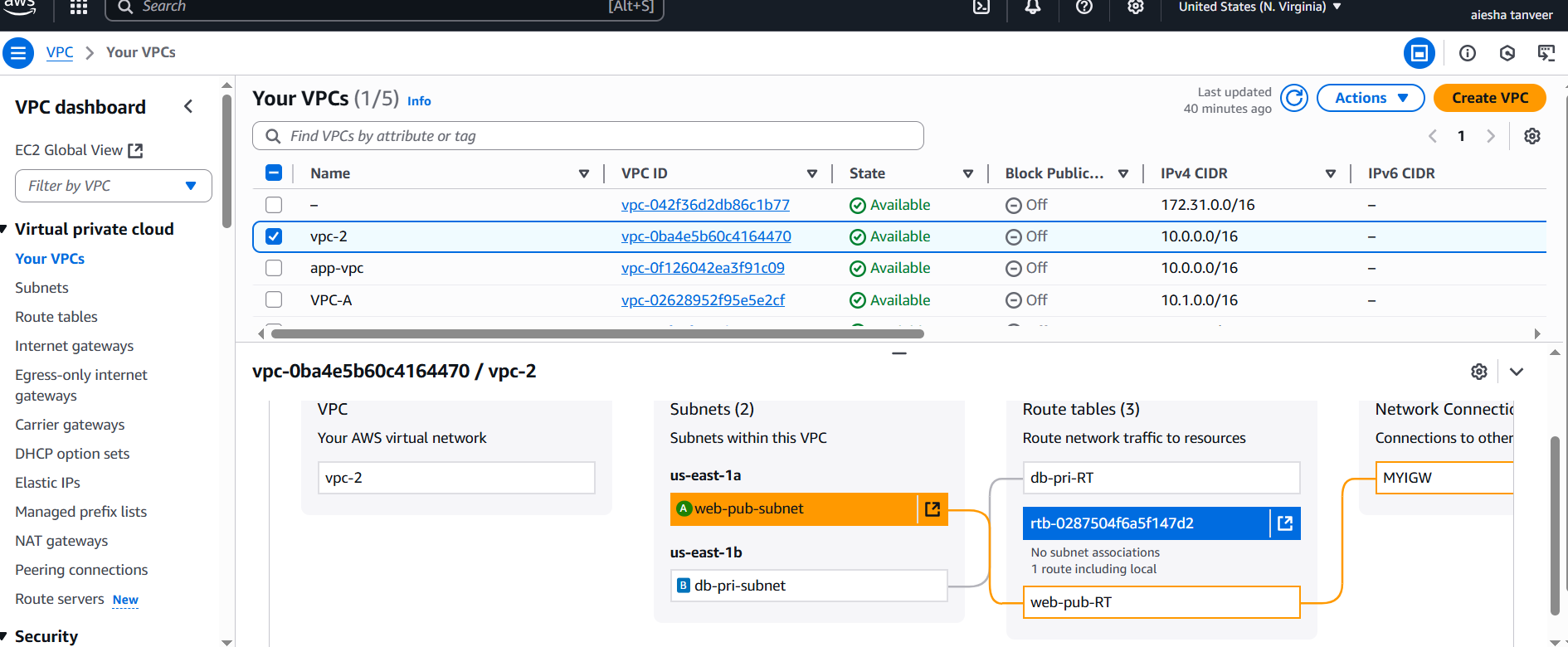
Click **create route table**

Enter → Name :

VPC : Select your VPC (vpc-2)

Click create route table.





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

* **Create an Elastic IP for the NAT Gateway**

Go to EC2 dashboard → Elastic Ips

Click Allocate Elastic IP address

Click Allocate.

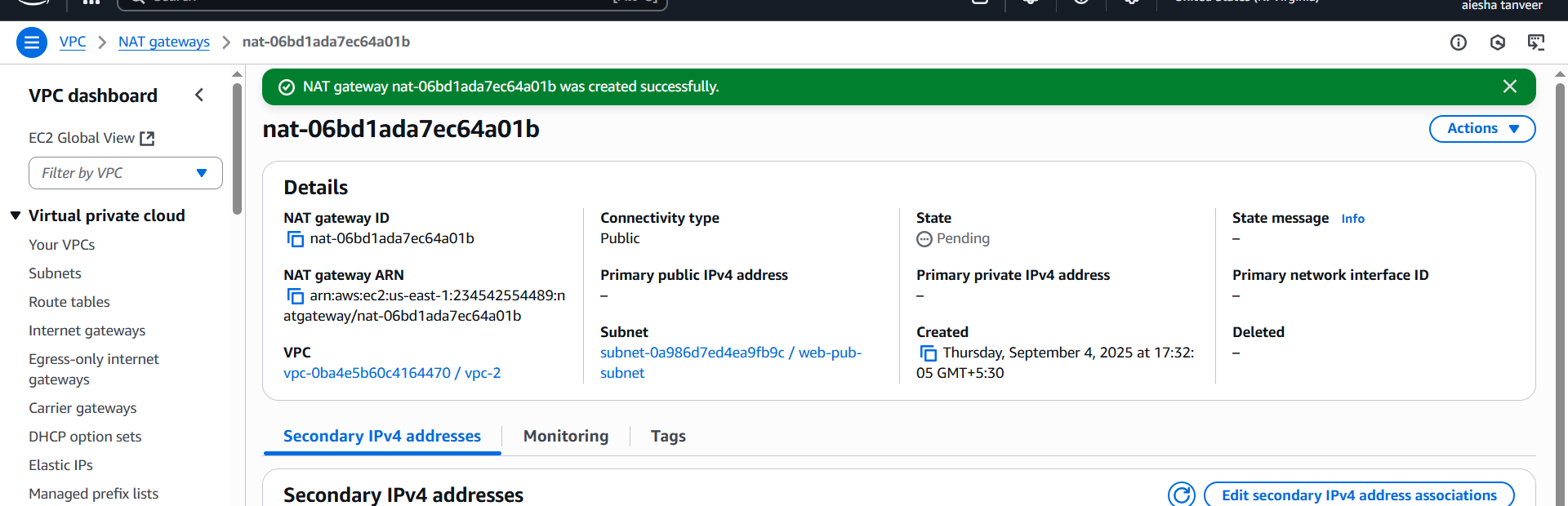
* **Create the NAT Gateway in the public subnet**

Go to VPC → NAT Gateways

Click Create NAT Gateway

**Configure:**

* **Subnet:**
* **Elastic IP allocation ID:**

Click **create NAT Gateway**.

* **Now make private Route table to use the NAT Gatway:**

Go to VPC Dashboard → Route tables

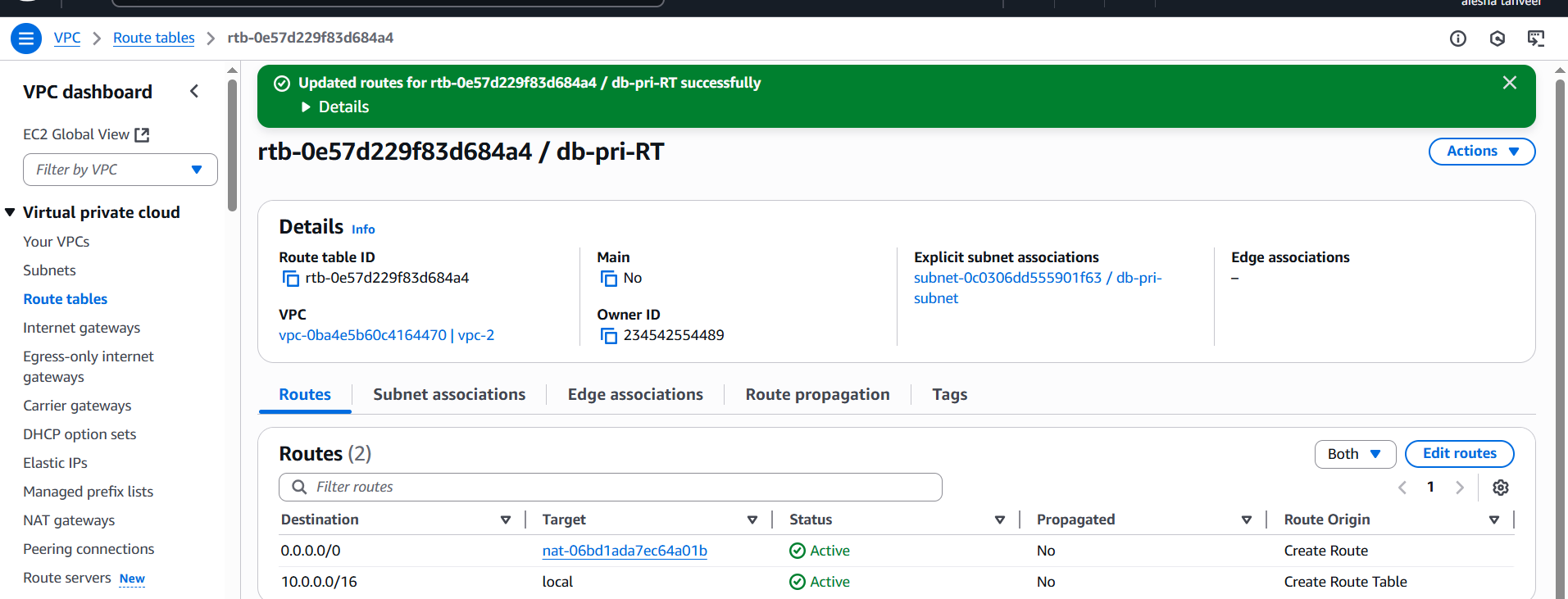
Select your private RT which is associated with the private subnet

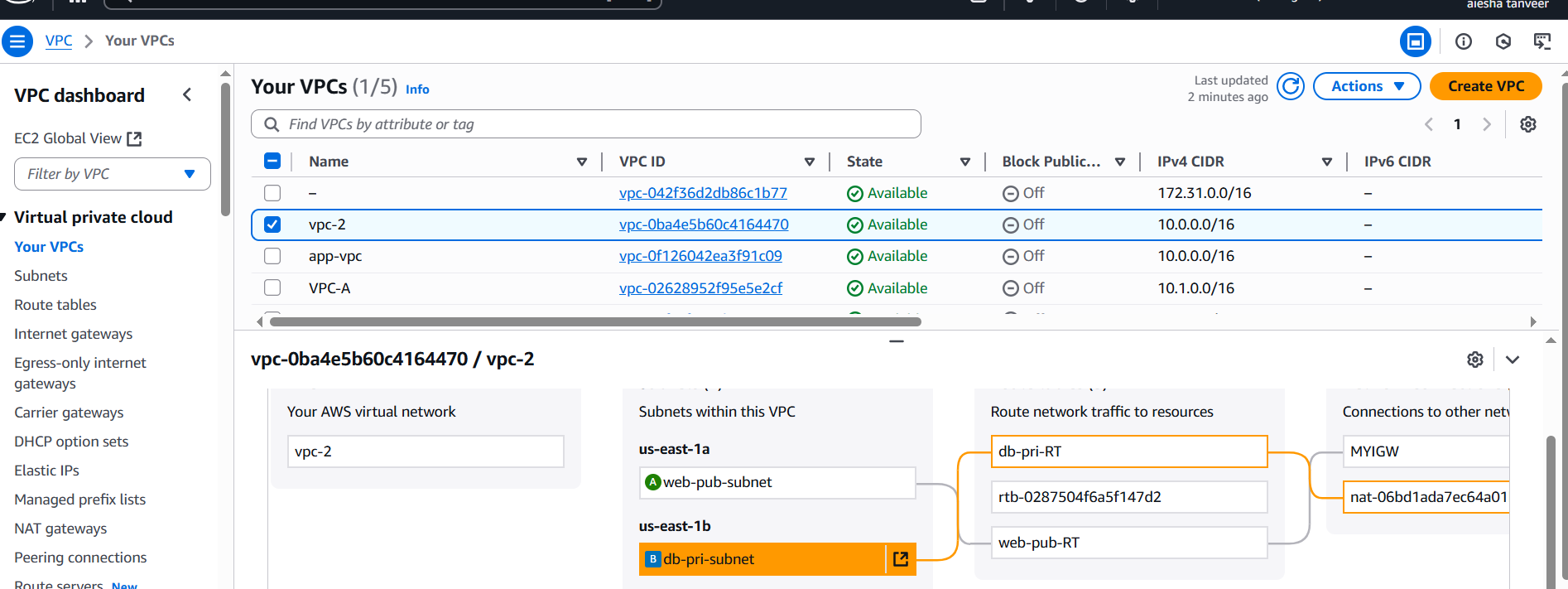
Go to Routes tab → click Edit Routes

Add routes. → **Destination:**0.0.0.0/0

**Target:** From dropdown box select NAT Gateway ID

Save Changes.





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

**Launch EC2 Instance in public subnet**

Go to EC2 → Launch instance

Name: public-ec2

AMI : select Amazon Linux 2 or U buntu

Instance type: t3.micro

Key-pair : create/ select from existing one

**Network settings:**

Select your VPC: VPC-2

Subnet: Select your public subnet (i.e:web-pub-subnet)

Auto-assign public IP: Enable

Use Default storage

**Security Group:**

Select **SSH** security group allowing port no:22 from your IP

Use HTTP/HTTPS if needed

Launch instance.

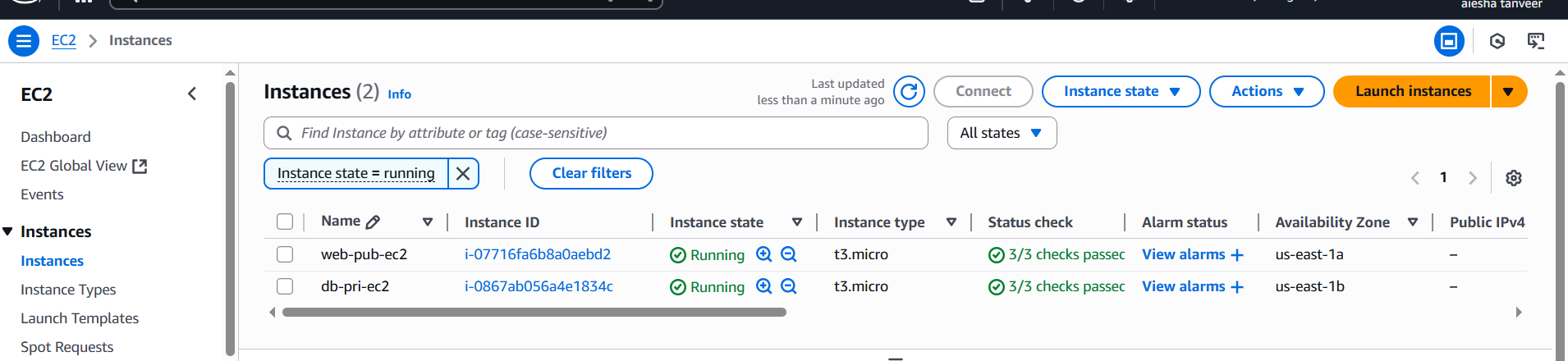
* In the similar way create EC2 from private subnet

Use the same key-pair

In Network settings →Use your private subnet (i.e:db-pri-subnet)

Auto-assign public IP : Disable

Launch instance.



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

Connect each instance via SSH

Install Apache server → run the following commands: sudo yum update -y

Sudo yum install httpd -y

Start &enable apache using commands: sudo systemctl start httpd

Sudo systemctl enable httpd

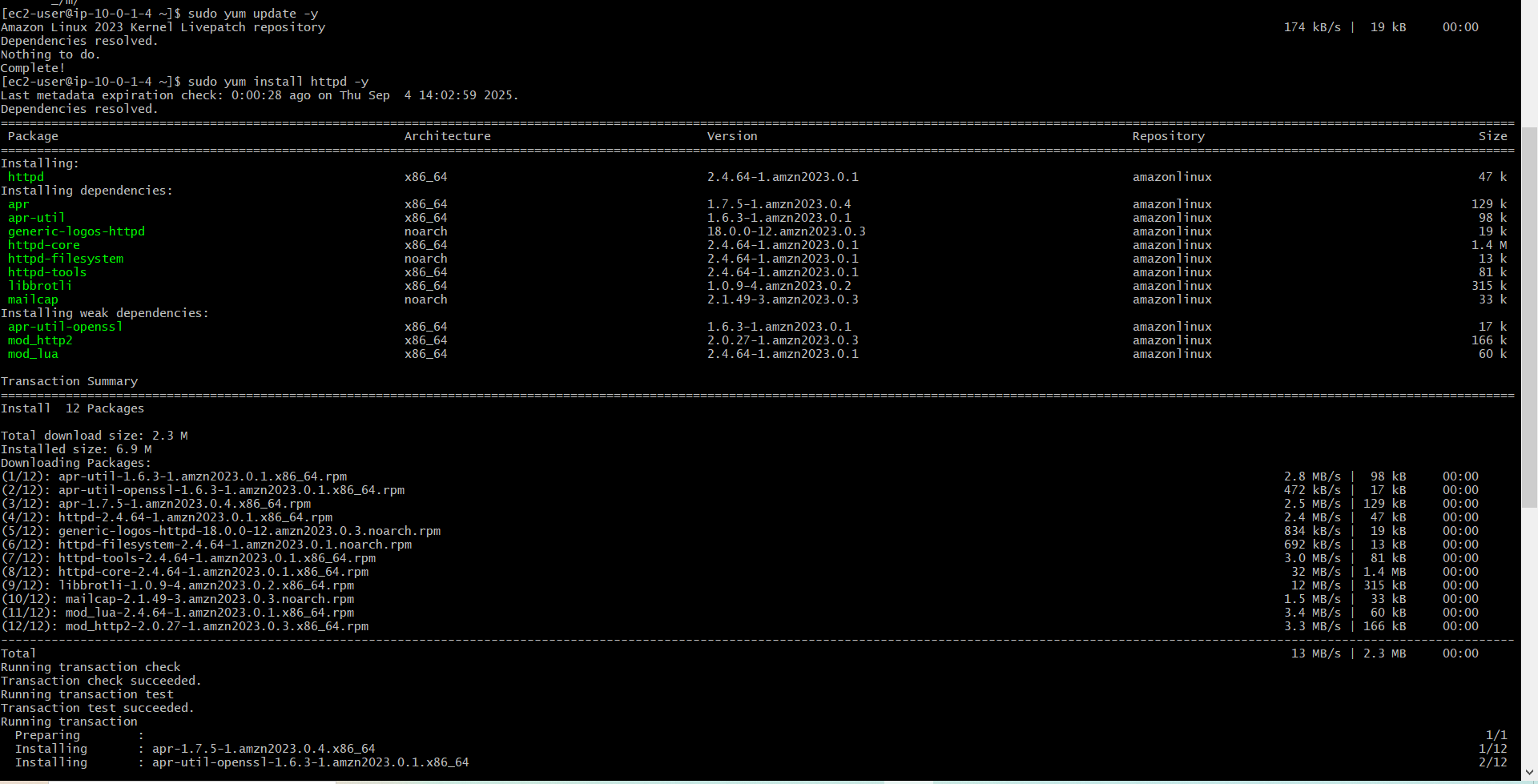
Add sample index.html using → echo “<h1>welcome to apache on $(hostname)</h1>” | sudo tee /var/www/html/index.html

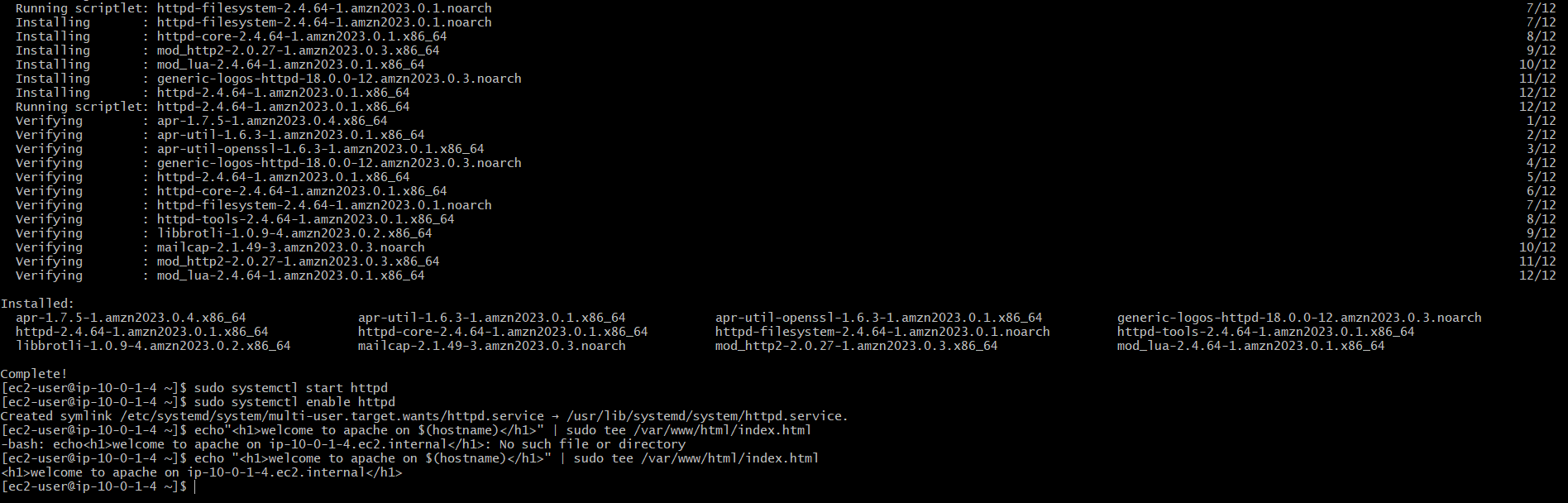
* **Adjust security group for public instance**

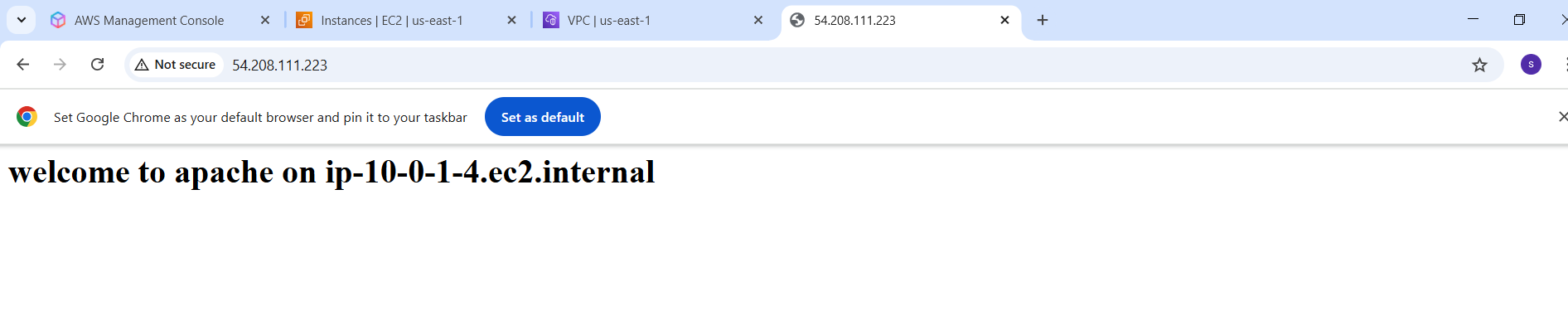
Go to EC2 → security groups

Edit inbound rules

Add rule : HTTP → port no: 80 → source :0.0.0.0./0







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

**1.** Go to ec2 console

Navigate to ec2 → load balancers

Click “create load balancer”

Choose Application load balancer → click create

**2. configure load balancer**

Name: MyALB

Scheme: internet-facing

IP Address type: IPV4

Listeners : HTTP (PORT NO: 80)

Availability zones : select your vpc

Check both subnets for diff AZs

1. **Configure security group**

Select security group that allows HTTP PORT:80 Inbound

1. **Configure target group**

Target group name : MytargetGroup

Target type : Instance

Protocol : HTTP Port no: 80

VPC : Choose your VPC

Click Next

1. **Register targets (EC2 Instances)**

Select both EC2instances

Click add to registered

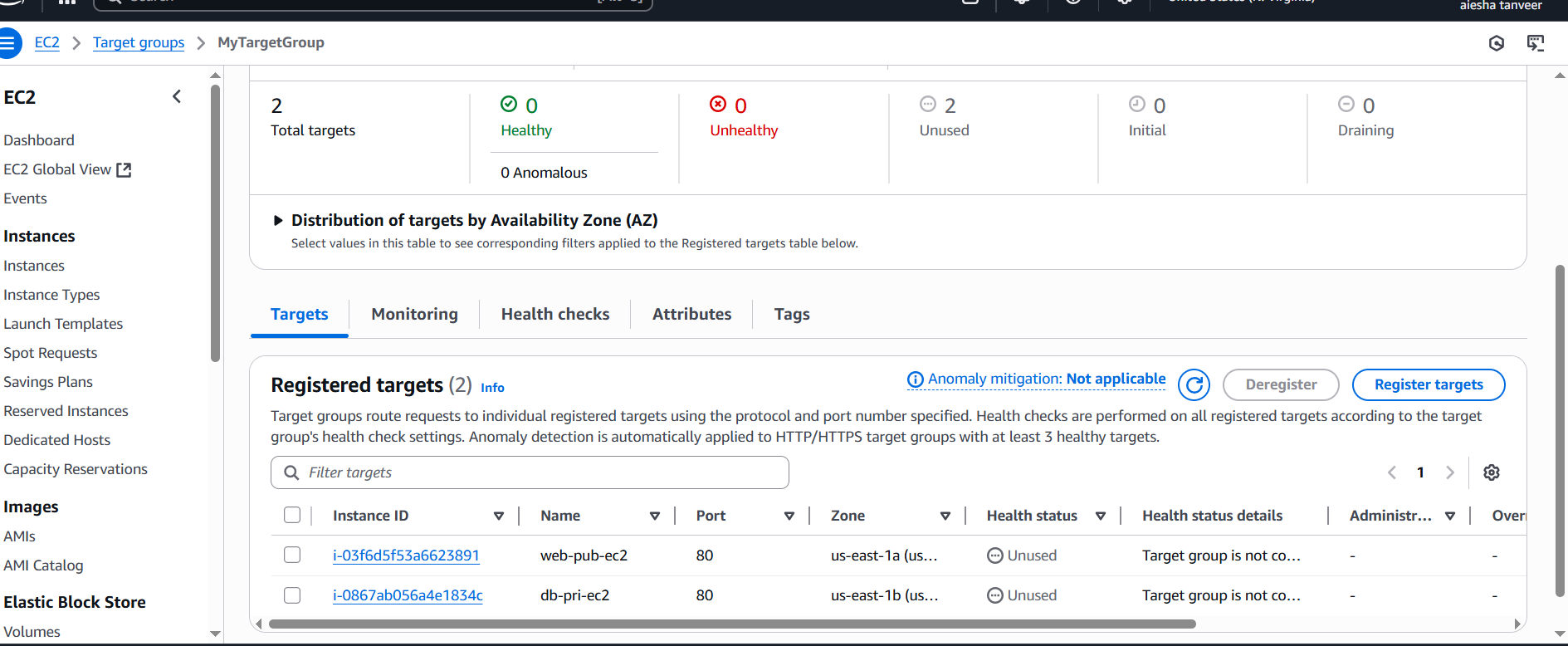
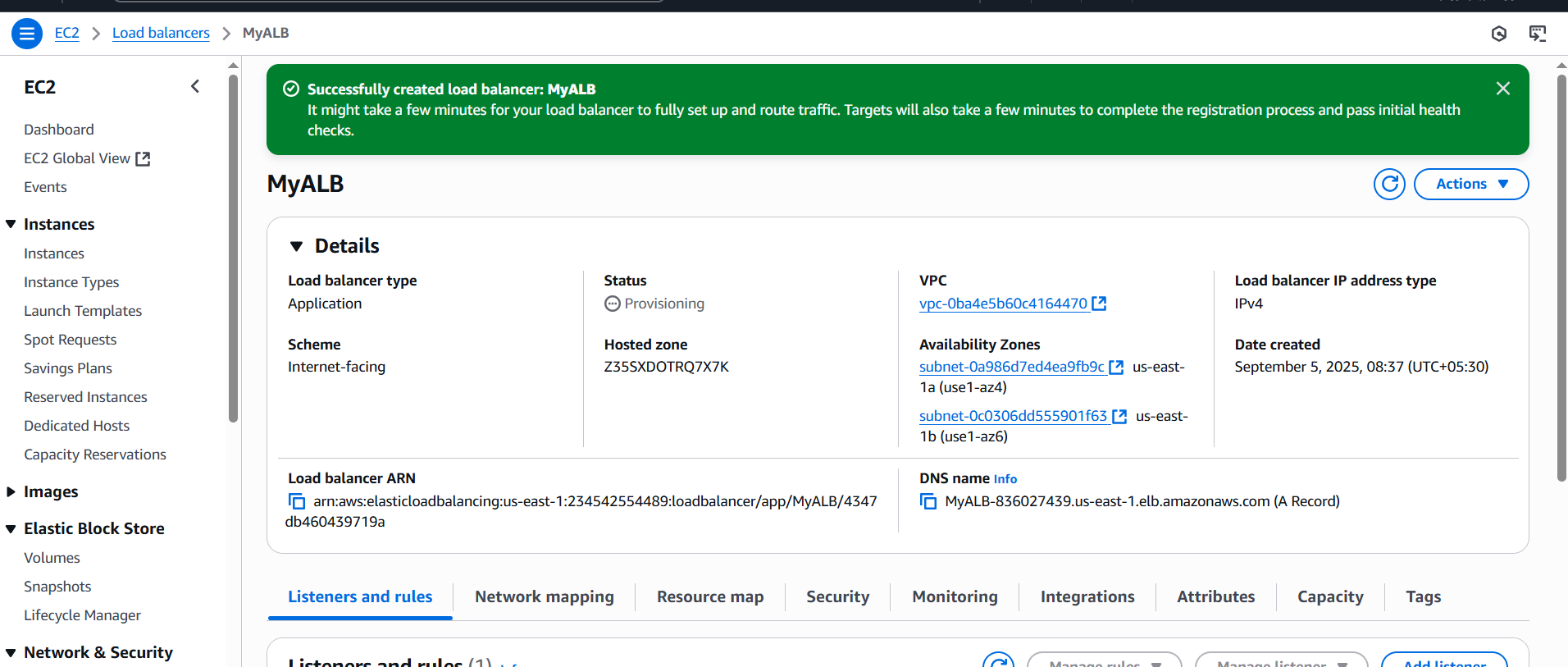
Click Create target group

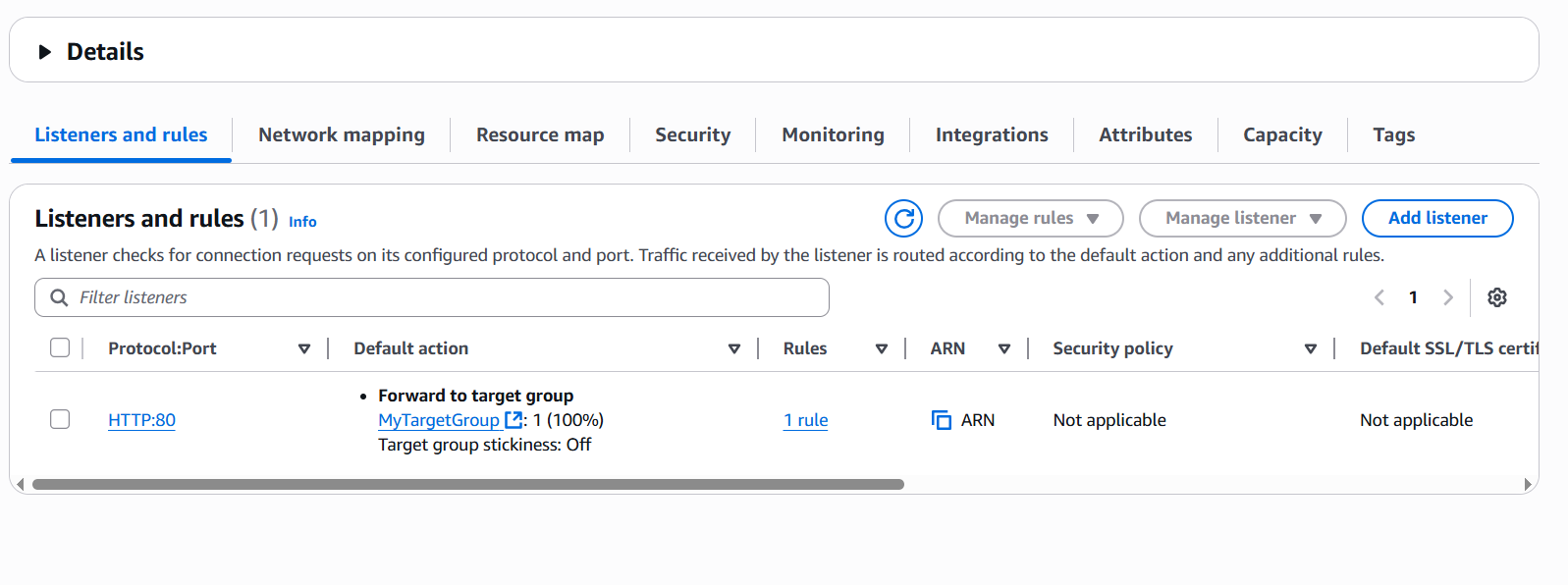
1. **Review and create ALB**

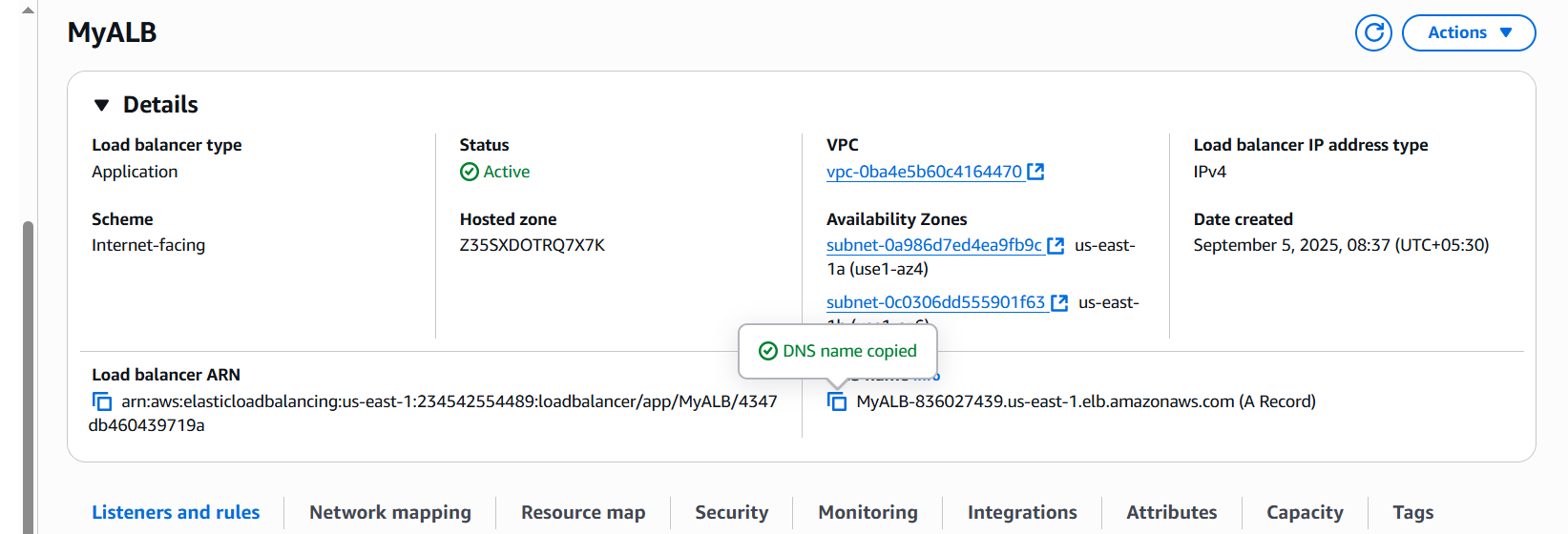
Go back to config of ALB

Select the target group which was created

Review and click create load balancer

Wait for the ALB to become active.



Copy DNS Name and verify in browser…to check whether it is running or not.

1. Store Application load balancer logs to s3.

* **Create or choose an S3 bucket**

Go to S3 console

Click create or use an existing bucket

Name : my-alb-logs-bucket-shabbu

Uncheck Block all public access

Create bucket.

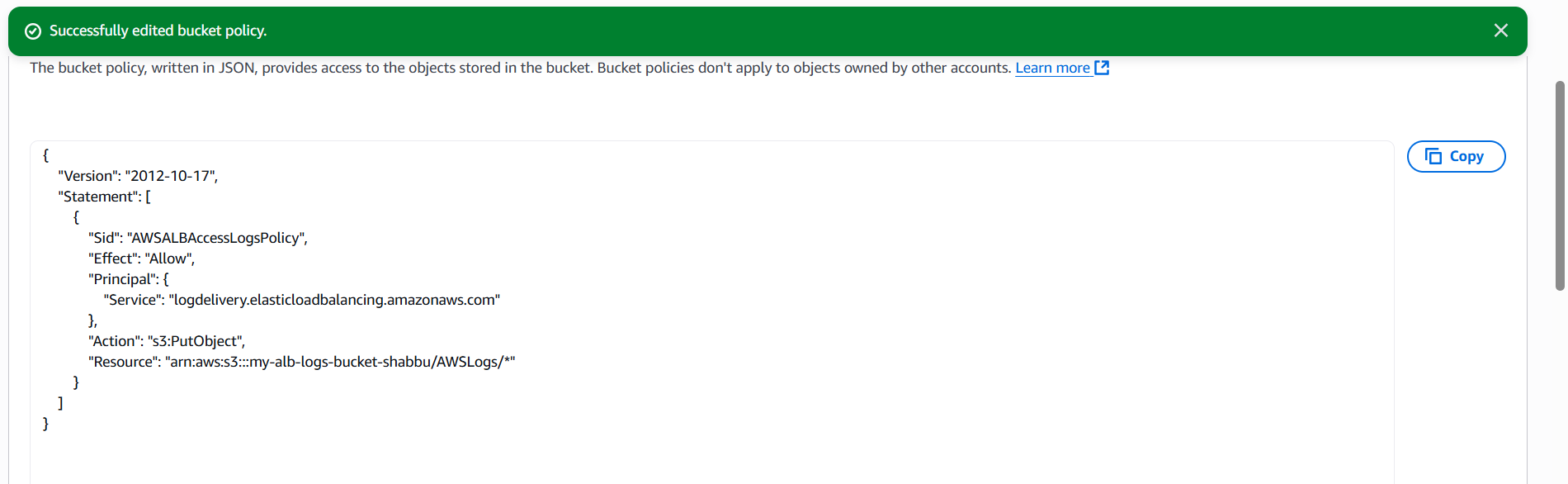
* **Update s3 bucket policy for ALB**

Go to s3 console

Click your s3 bucket : my-alb-logs-bucket-shabbu

Go to permissions tab

Click Edit under bucket policy



Go to ec2 console → load balancers

Select your ALB

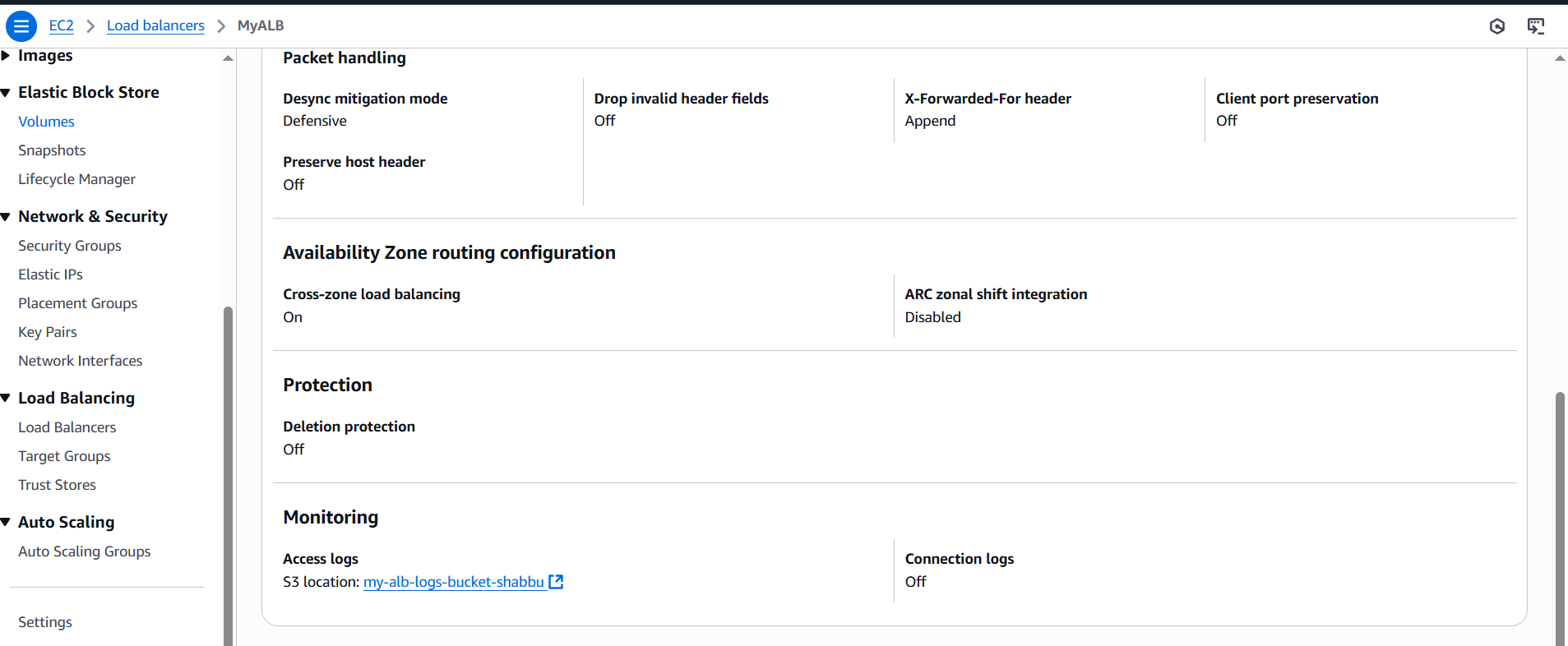
Click the attributes tab

Edit attributes

Under Access logs : Enable logging

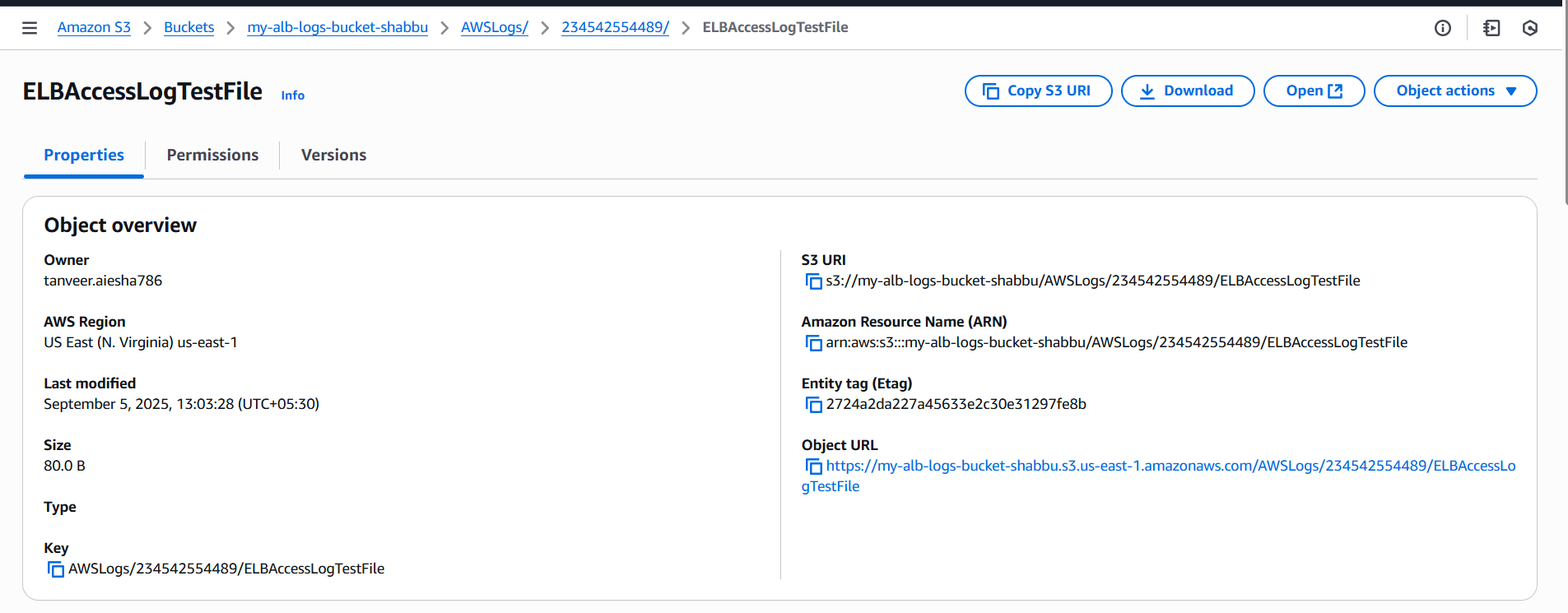
S3 bucket name: my-alb-bucket-shabbu

Save changes



Check the logs under the s3 bucket:my-alb-bucket-shabbu

(i.e: AWSLogs/Account id/ELBAccessLogTestFile)



1. Store the VPC flowlogs to cloudwatch group.

* **Create a cloudwatch logs group**

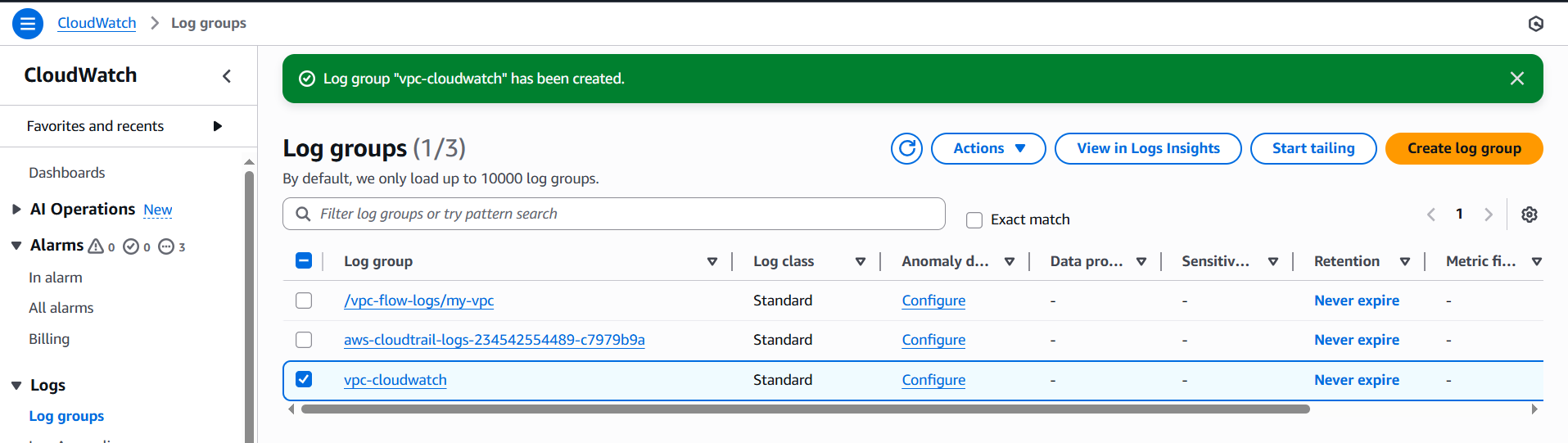
Go to cloudwatch → logs → log groups

Click **create log group**

Enter Name: vpc-cloudwatch

Choose a retention period (i.e: 1 week or never expire)

Click **create**



* **Create IAM Role for VPC flowlogs**

Go to IAM Console → roles

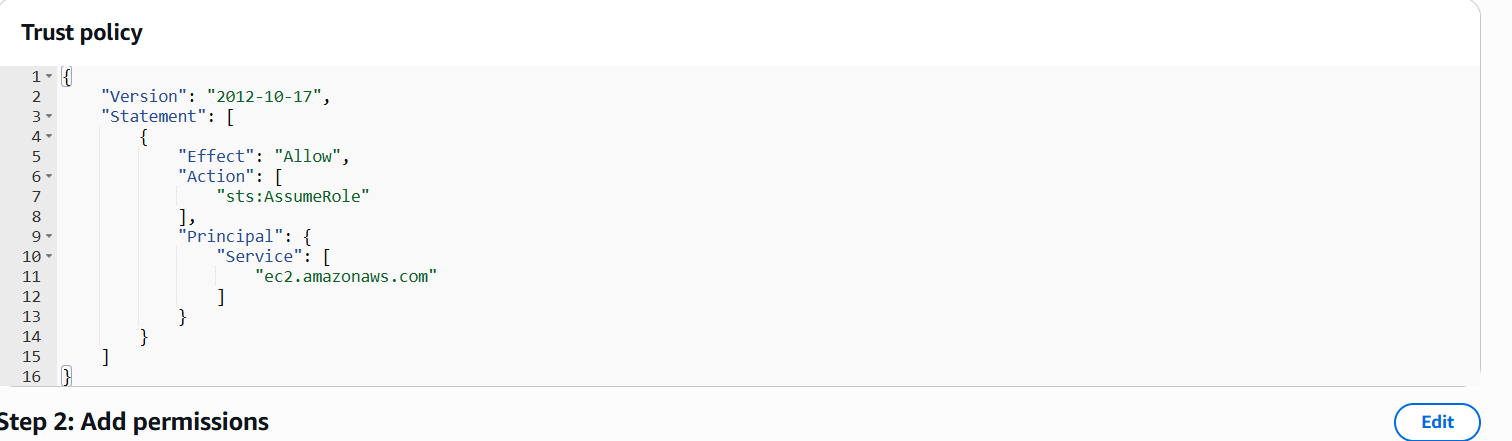
Create Role

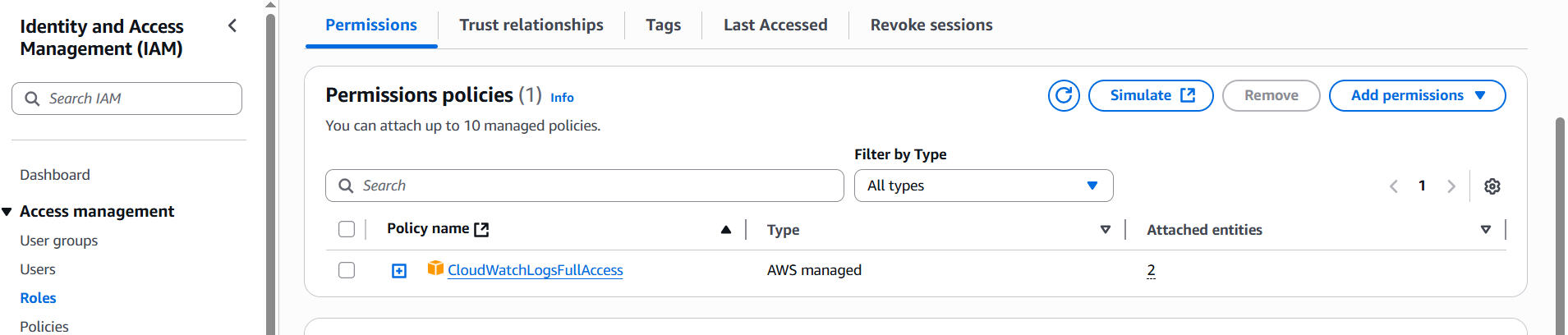
Trusted entity type: choose AWS Service

Use cases: select EC2

Click next

Attach a policy





* **Enable VPC Flowlogs**

Go to VPC Dashboard

Click your VPC

Go to flowlogs tab → click create flow log

* **Configure flow logs**

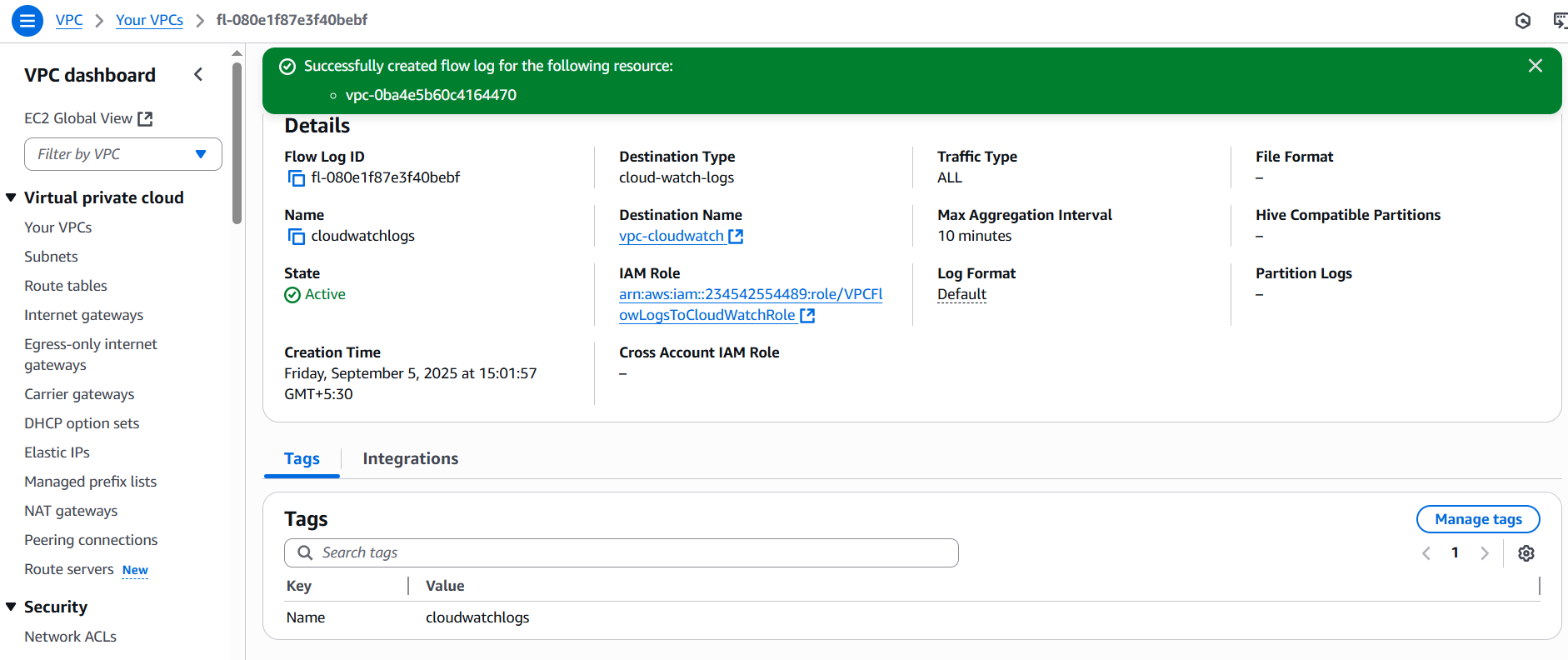
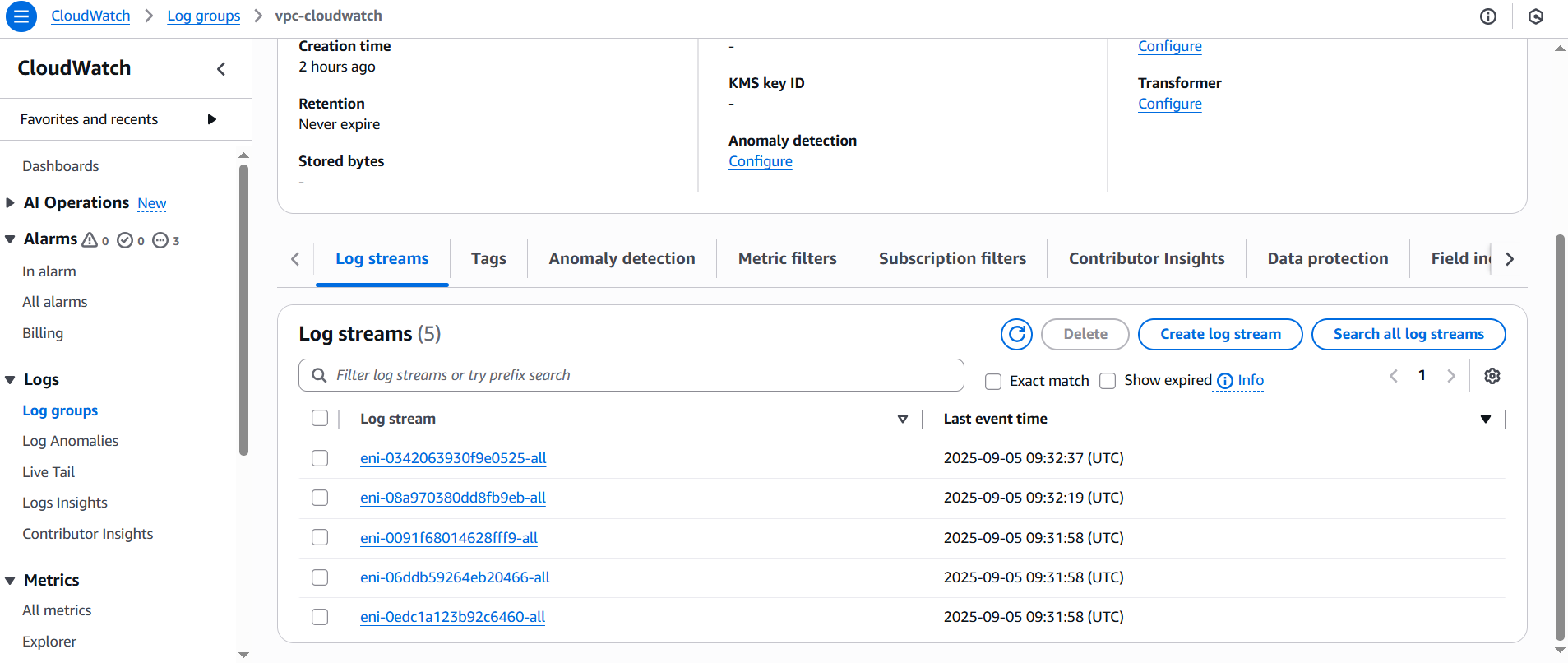
Filter : All

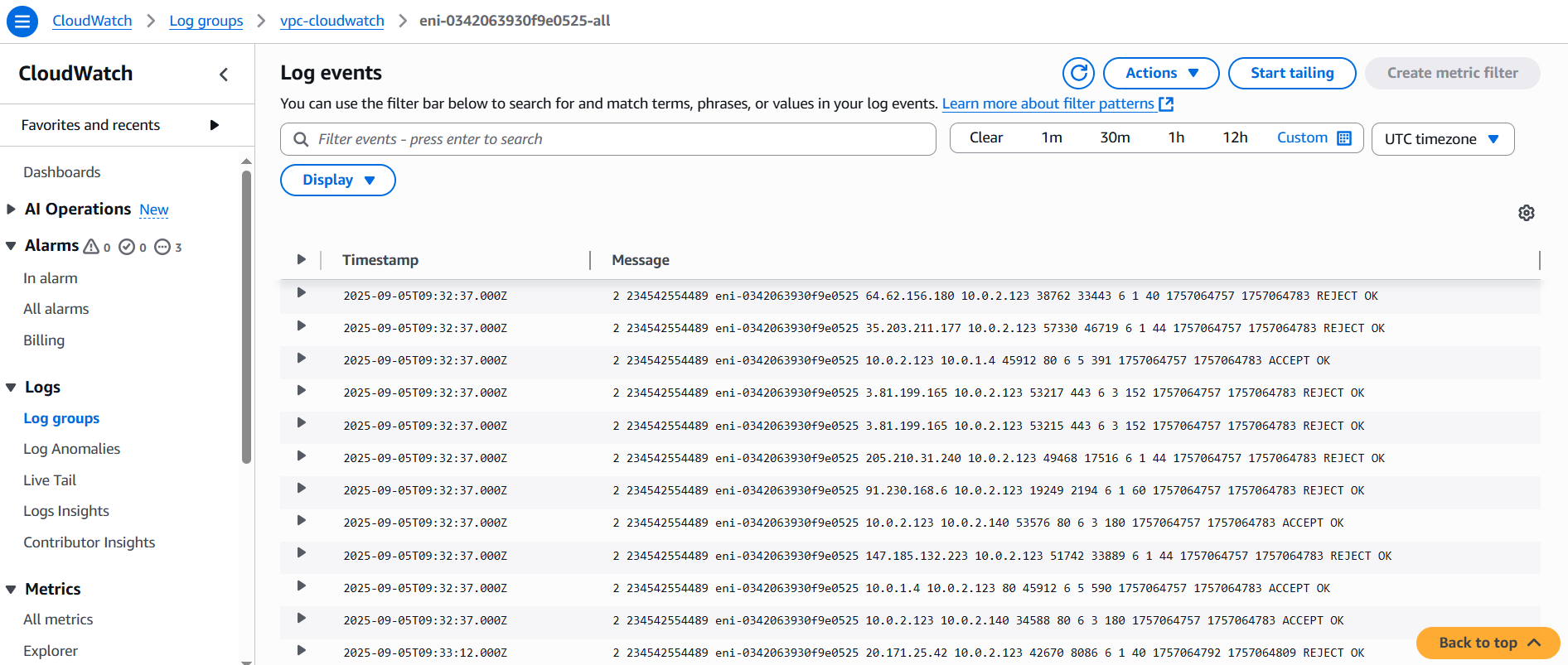
Destination: Send to cloudwatch logs

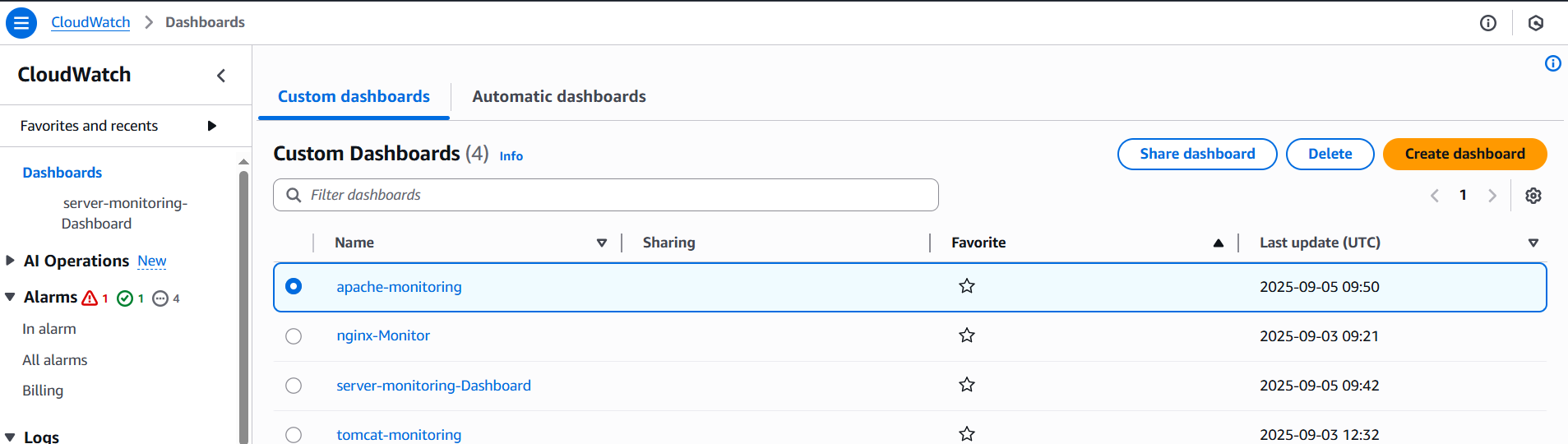
Log group: vpc-cloudwatch

IAM Role:cloudwatchlogsfullaccess

Click create flow log.

11. Create monitoring Dashboards to monitor CPU utilization and monitor Apache server.



12. CPU Utilizations more than 70% then it should trigger autoscaling and launch new instance.

