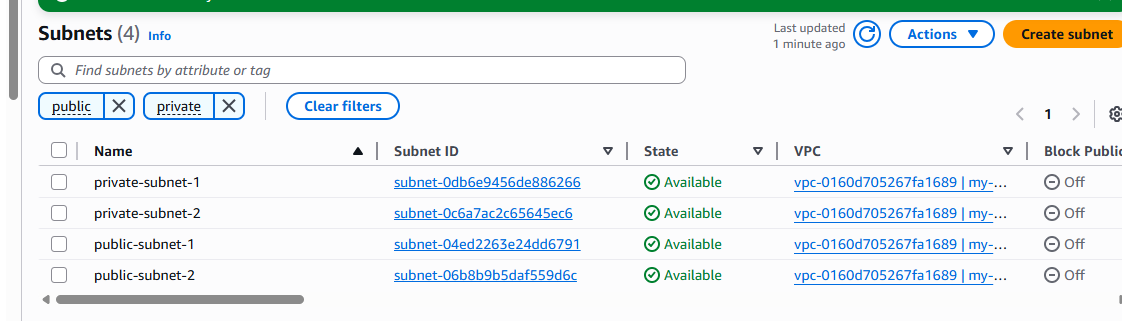
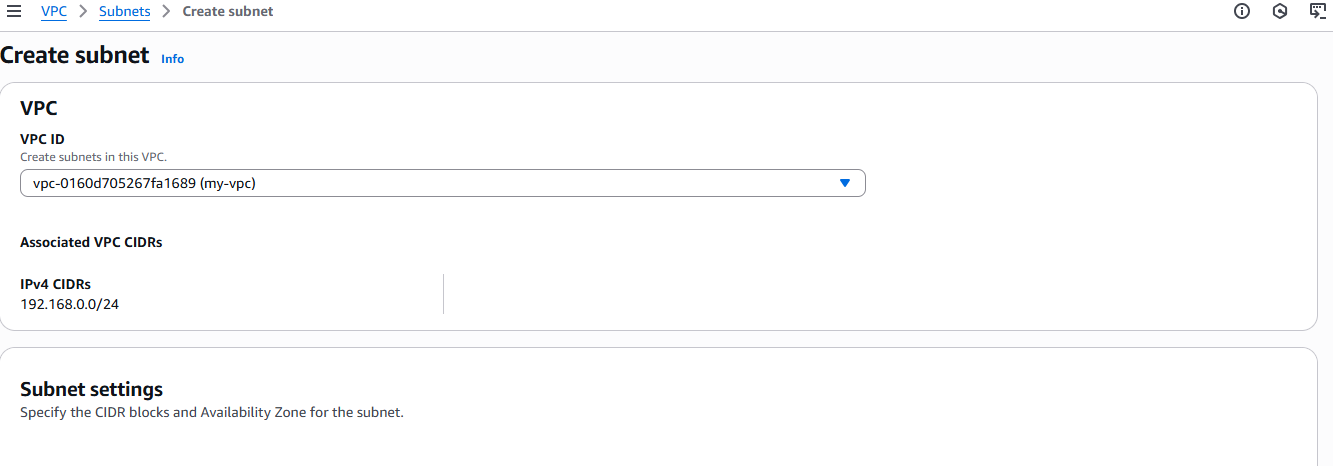
**VPC task**

1. **Create VPC with 2 private and 2 public subnets.**

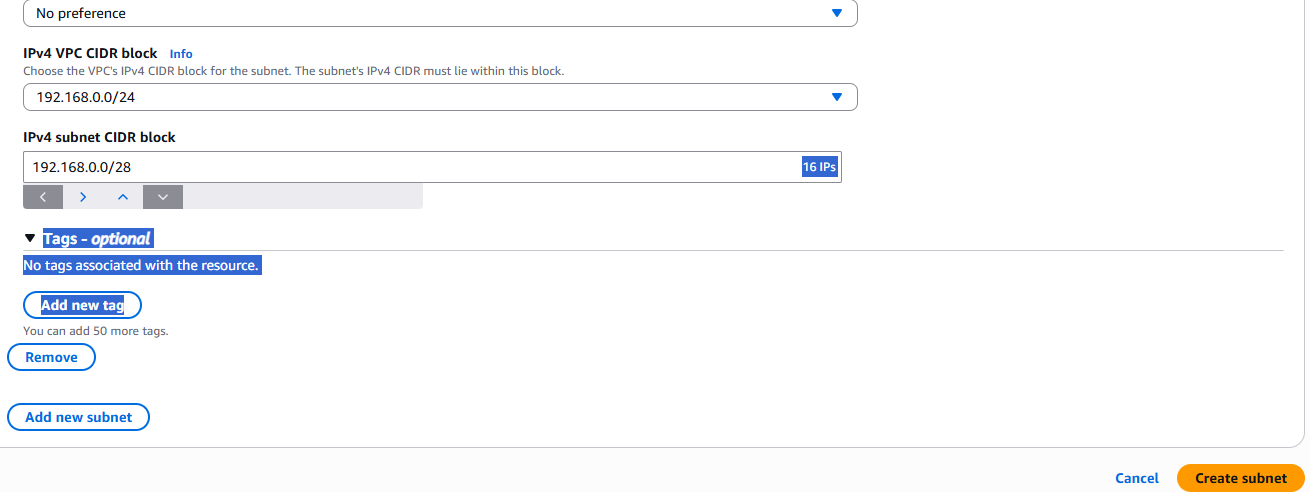
Amazon VPC ---> Subnet ---> Create Subnet

****

Add your VPC in VPC ID

****

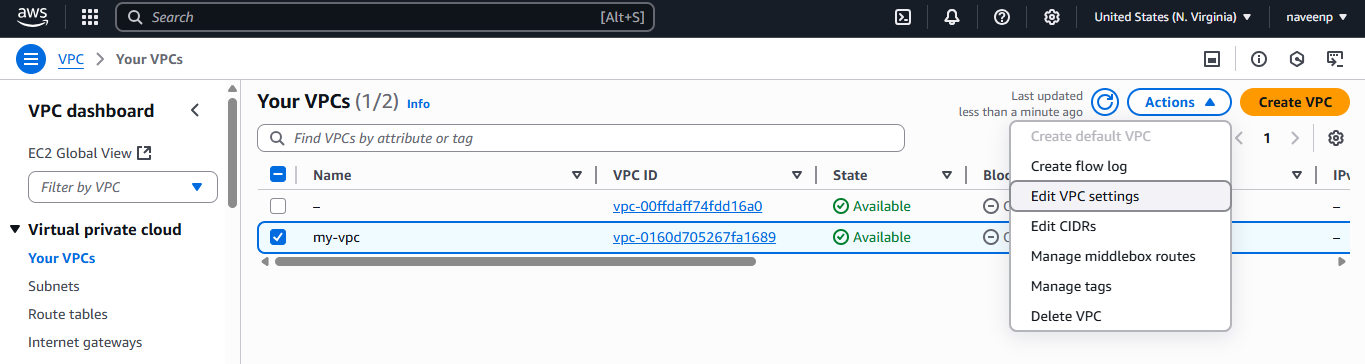
Enter reqired IPV4 subnet CIDR block

****

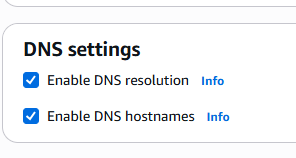
To click create subnet

1. **Enable DNS Hostname in VPC**

VPC ---> Select the created VPC you want to edit ---> Actions



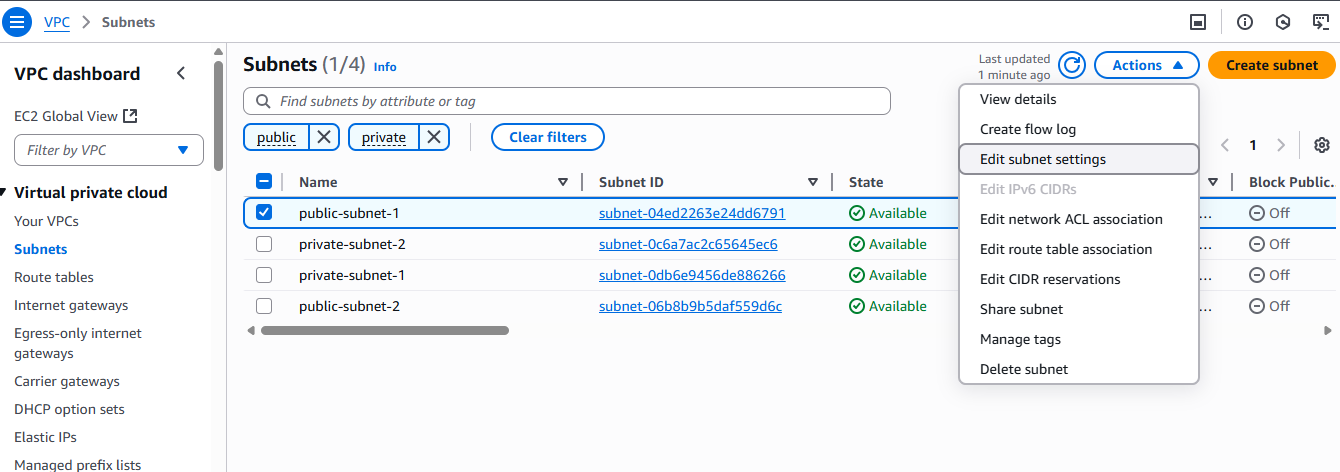
Click Enable DNS hostname in DNS setting



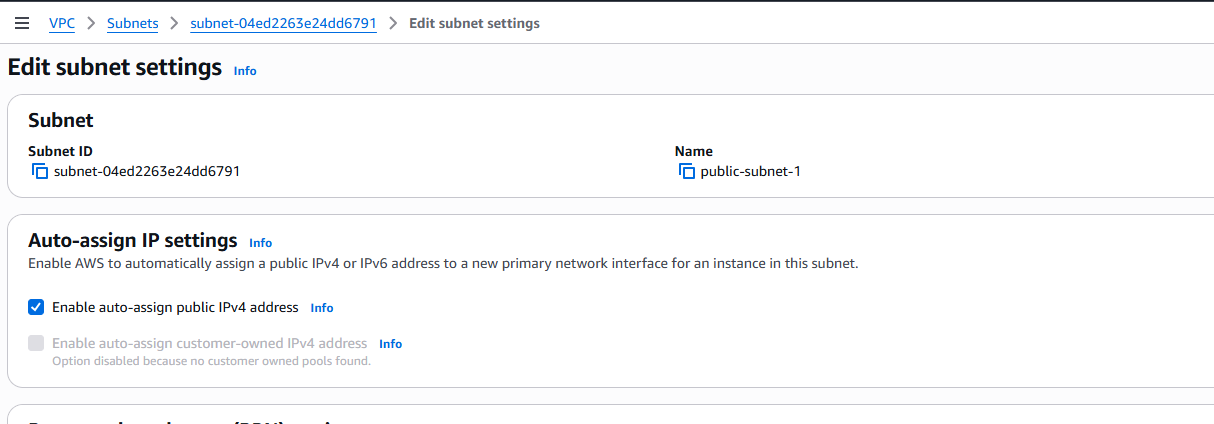
Click save

1. **Enable Auto Assign Public ip in 2 public subnets.**

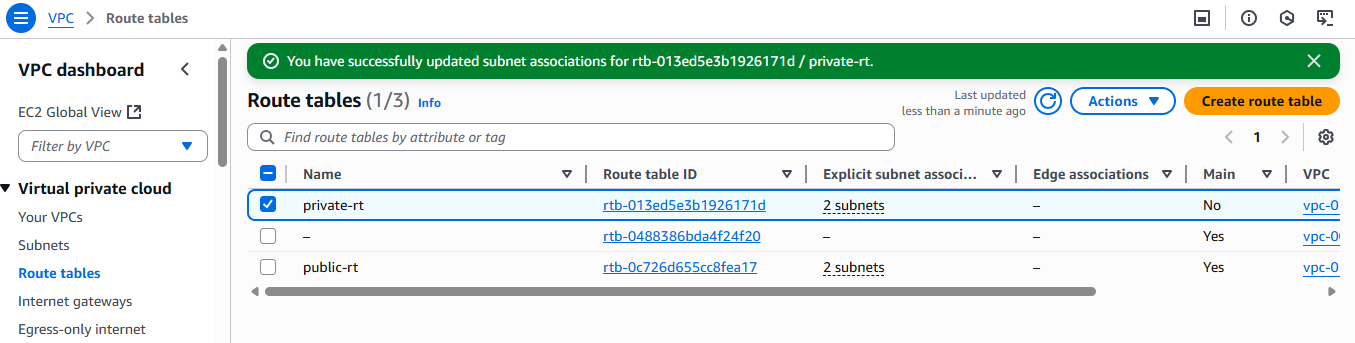
Amazon VPC ---> Subnets --->Select your Public Subnet ---> Actions --->

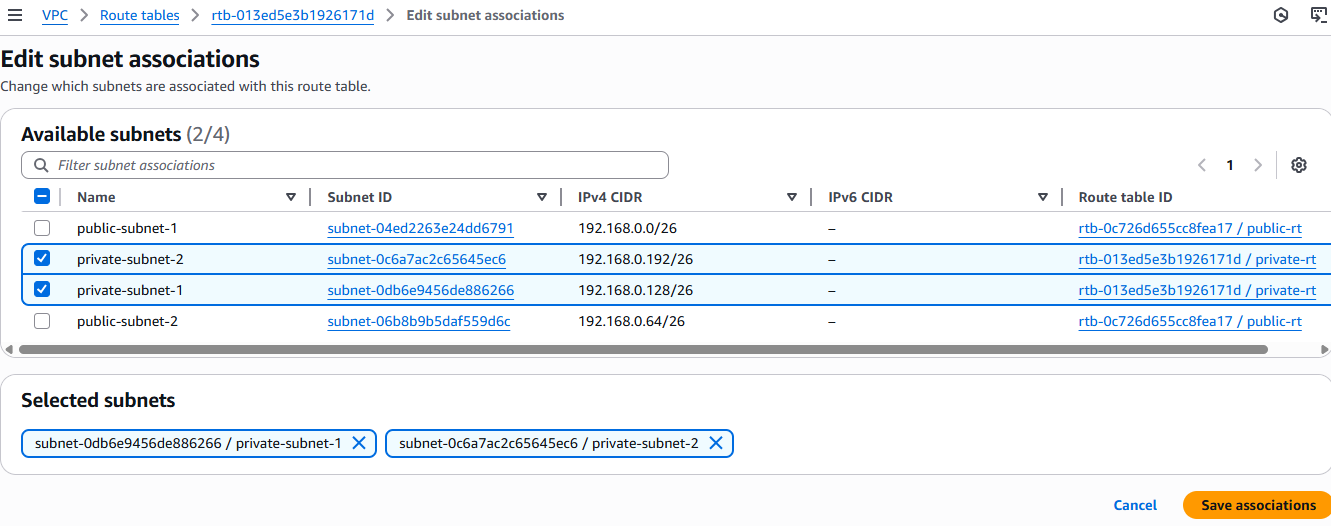
Edit subnet settings ****

**Click** Enable Auto-assign Public Ipv4 address ---> **Click** save

****

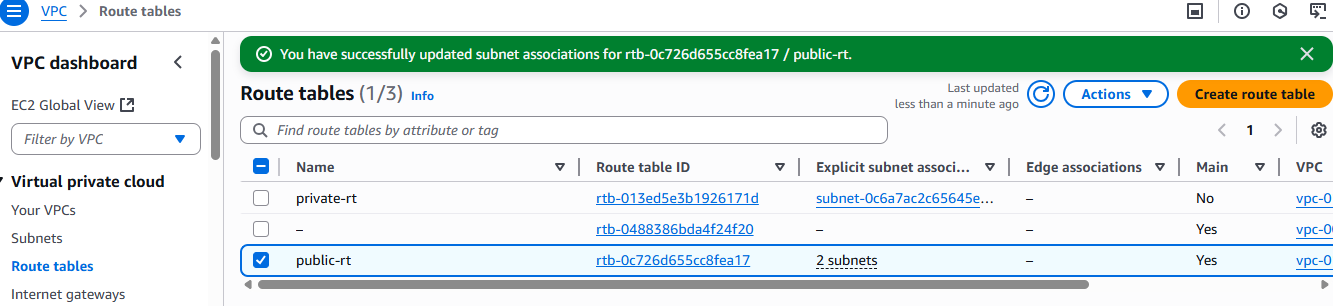
1. **Add 2 private subnets in private route table.**

****

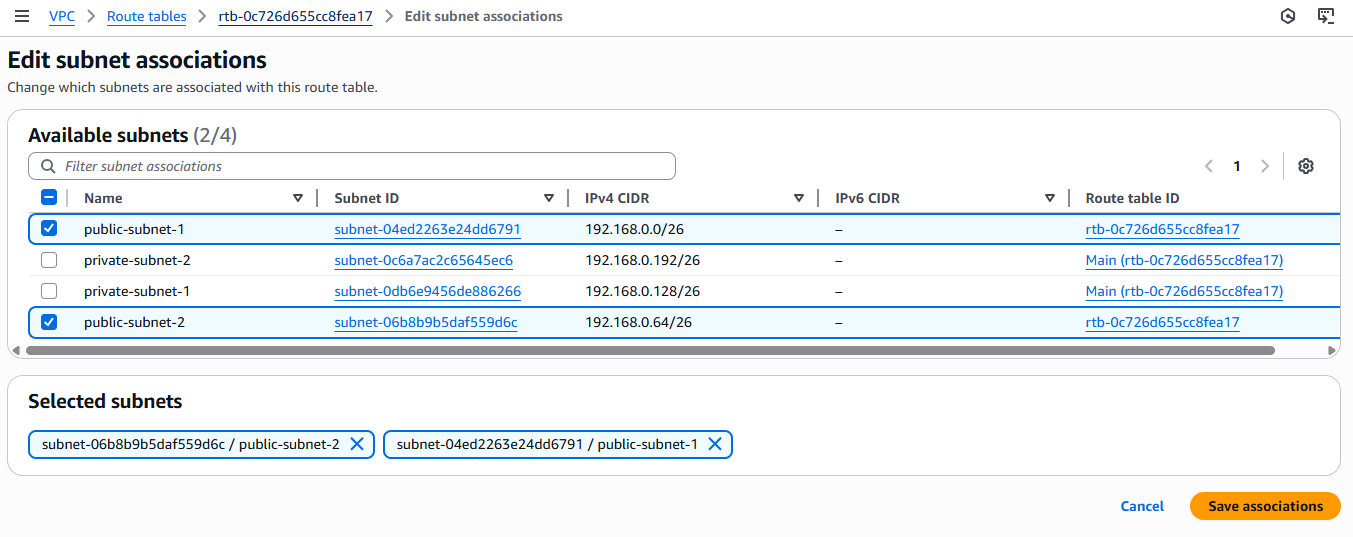
****

1. **Add 2 public subnets in public route table.**

VPC ---> Route Table ---> Actions ---> Edit subnet associations --->

****

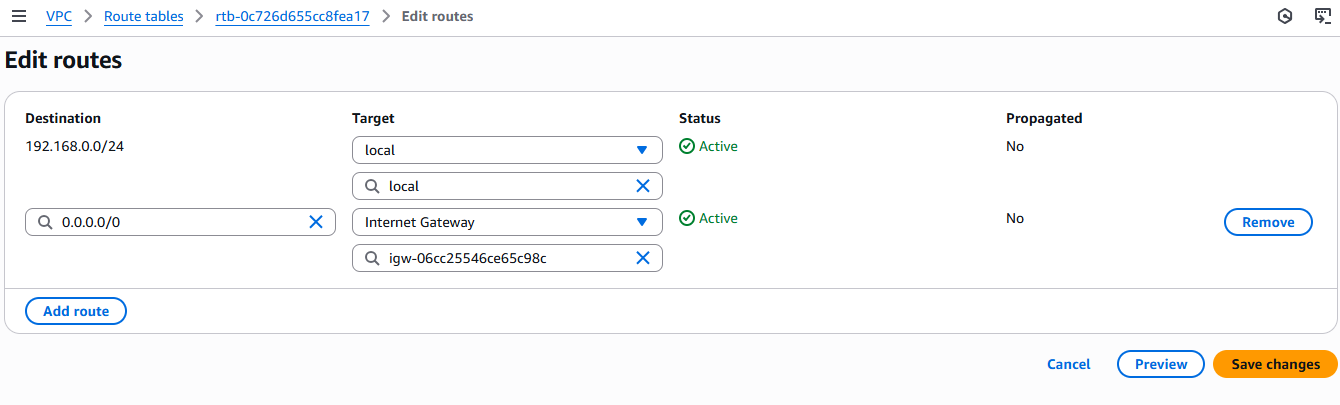
Select Public Subnets ---> click Save associatons

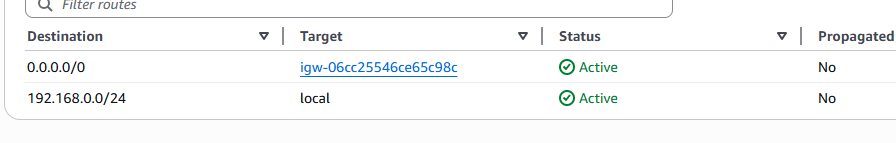
****

1. **Public route table will have the routes to internet and local.**

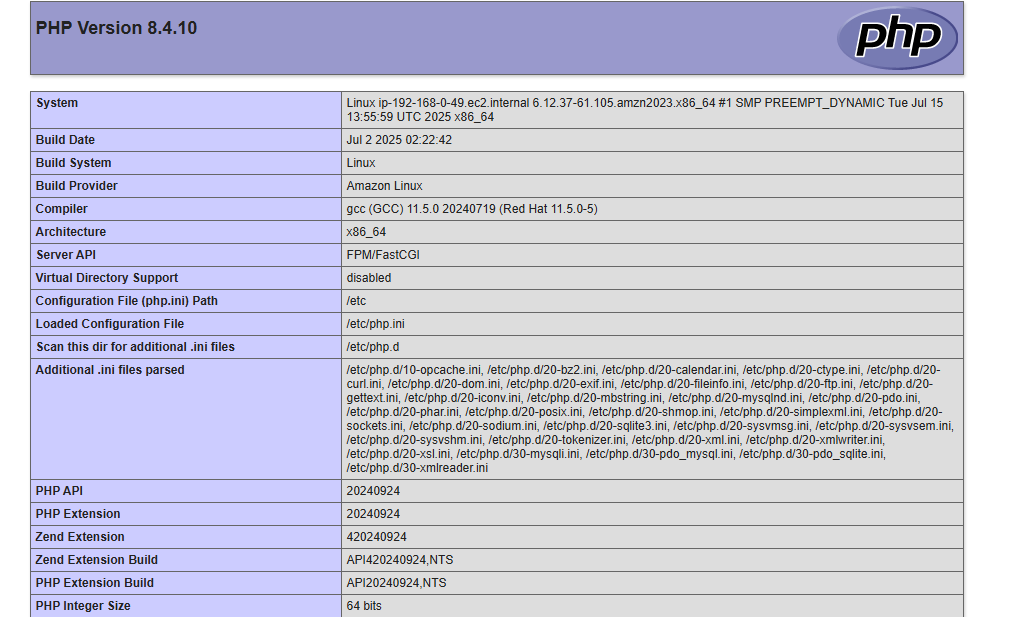
VPC ---> Route tables ---> Select your Route table --->Edit Routes

---> Add Route ---> Click Save changes





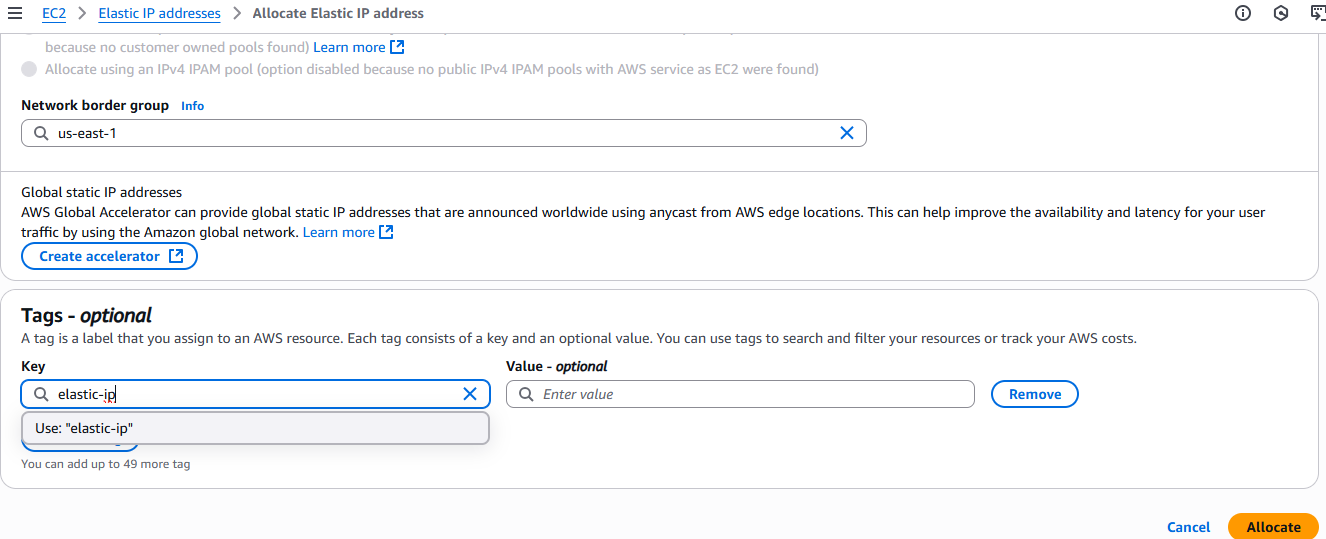
1. **Create Ec2 in public subnet with t2micro and install php.**

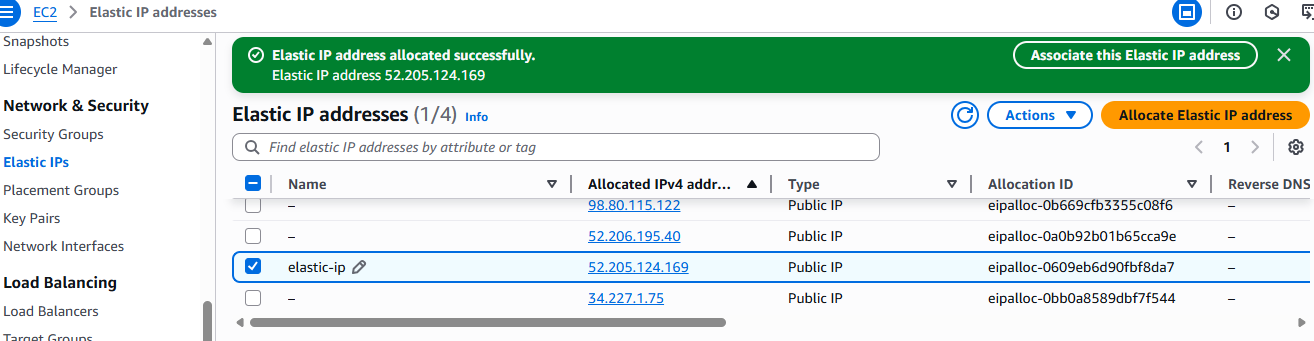
****

1. **Configure Nat gateway in public subnet and connect to private Instance.**

i)Create or Use existing Elastic Ip

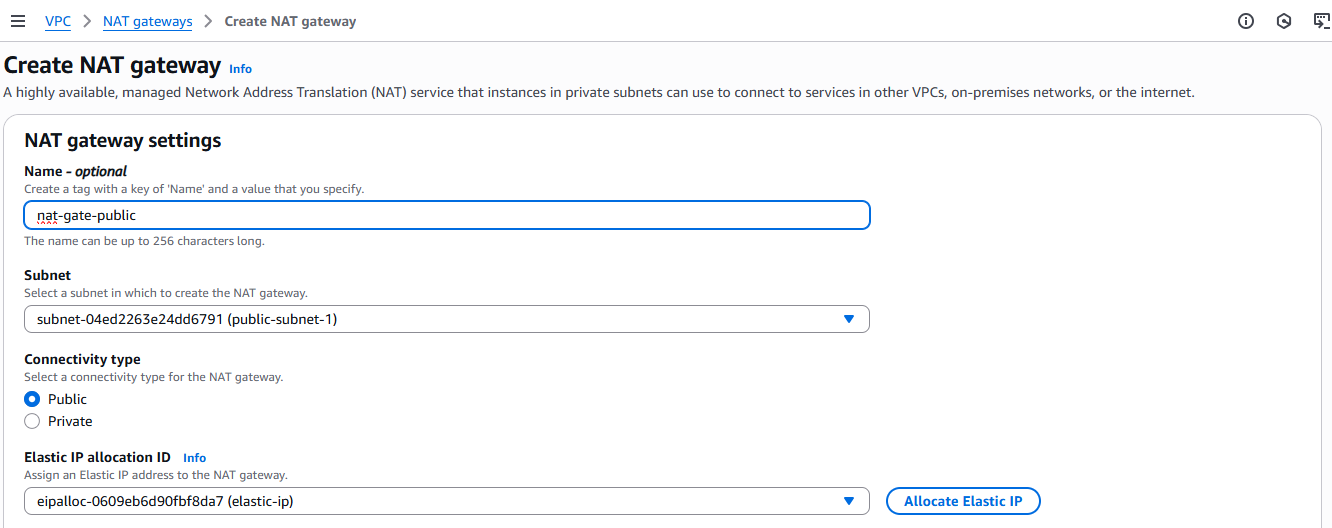
Ec2 --->Elastic Ip address ---> Allocate Elastic IP address ---> Click save

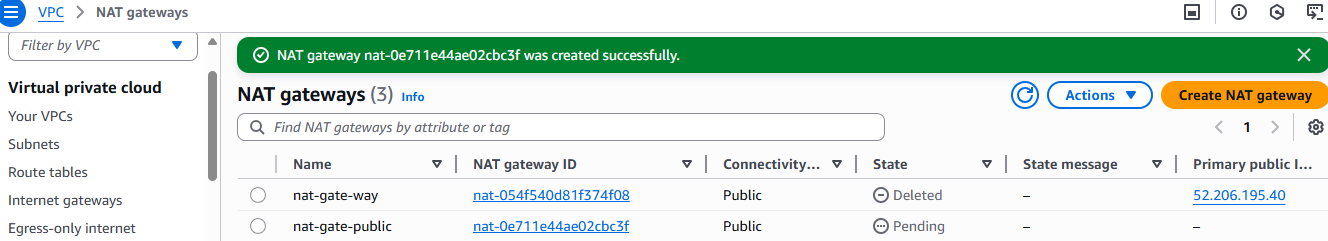
****

****

Ii)Create the NAT Gateway

VPC ---> NAT gateway ---> Create NAT gateway ---> Add created public subnet

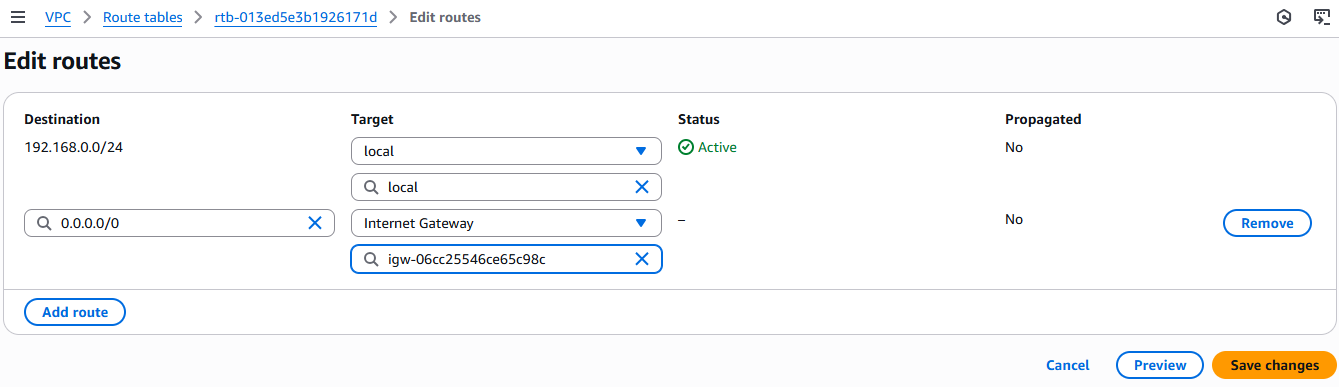
****

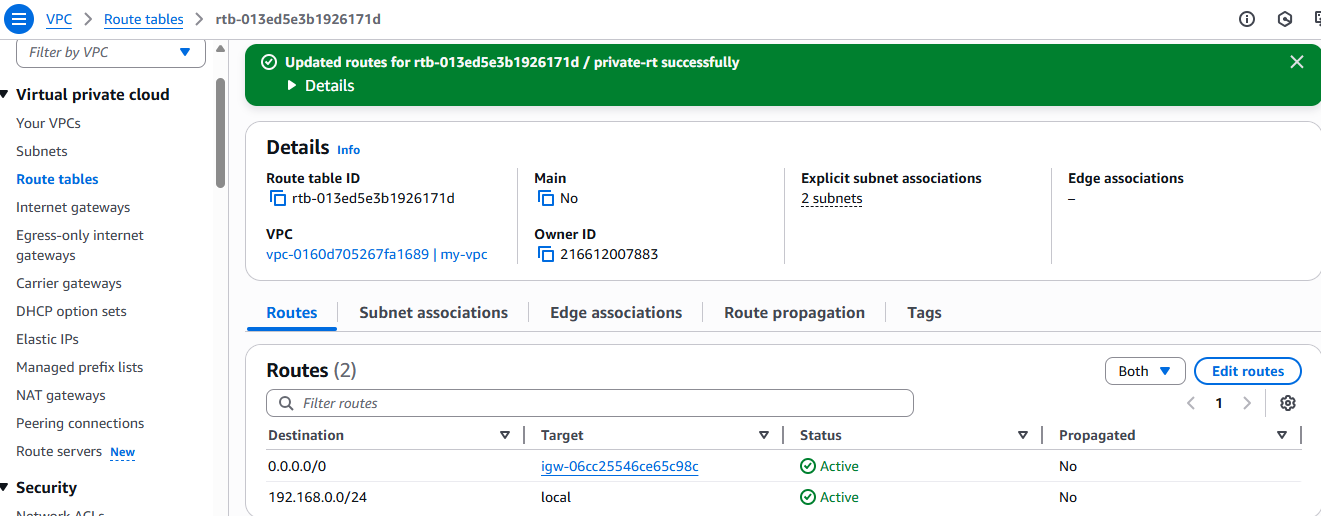
****

Iii)Update Private Route Table

VPC ---> Route Table ---> Select your Private Route ---> Click Edit Route

---> Click Save changes.

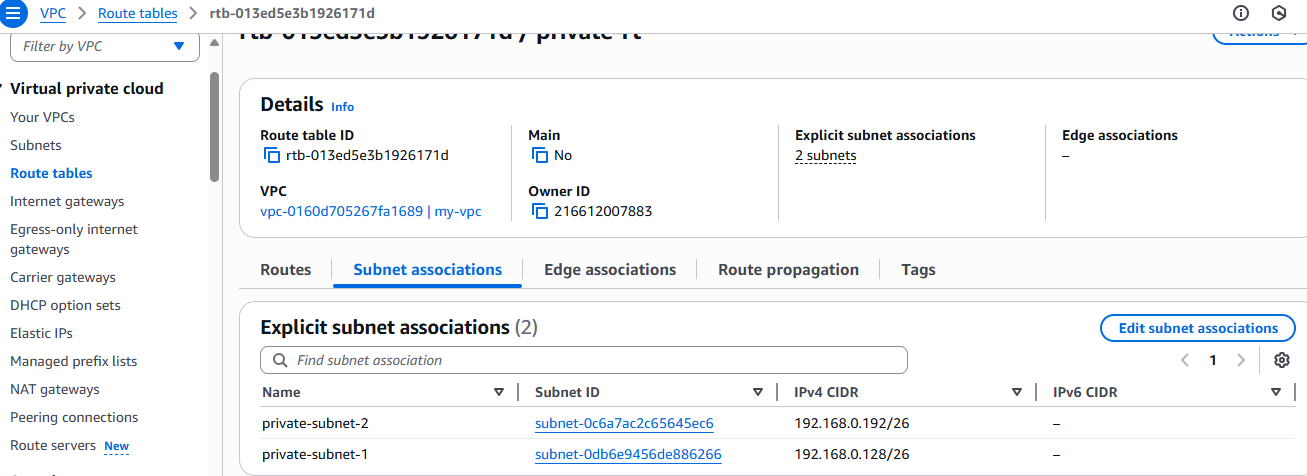


****

Iv)Confirm Subnet association.

To check Private Subnet are associated with Private Route Table

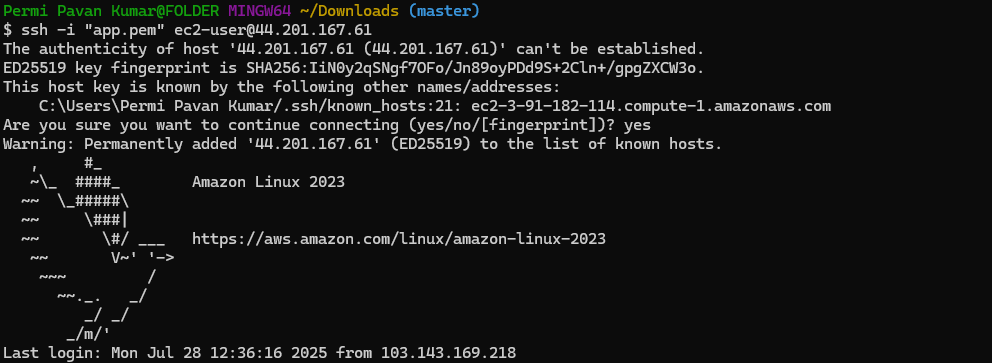
VPC ---> Subnets --->Select Created Private Subnet --->Go to Subnet Association



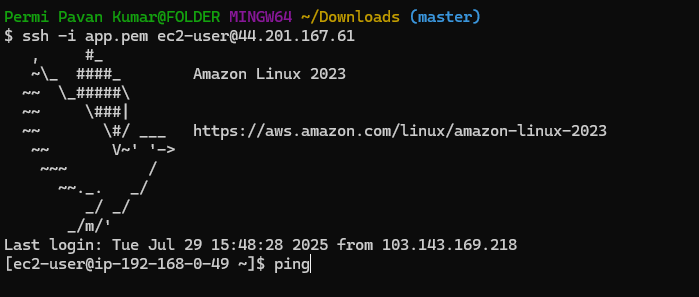
v)First to create EC2 where to configure NAT gateway in public subnet.

Then to create EC2 instance with private subnet.

First to run one ec2 instance with public Ip.



Then run private subnet instance with private Ip.

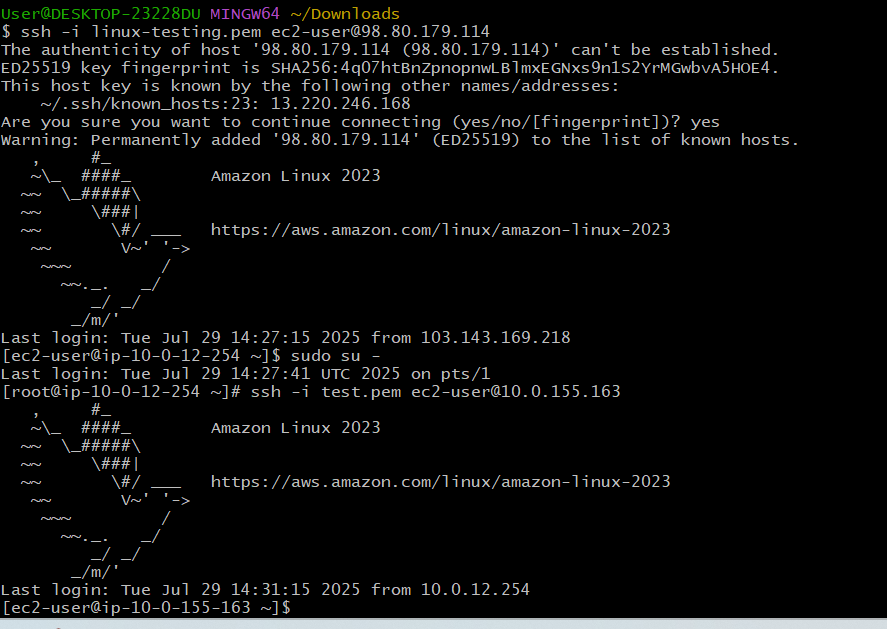


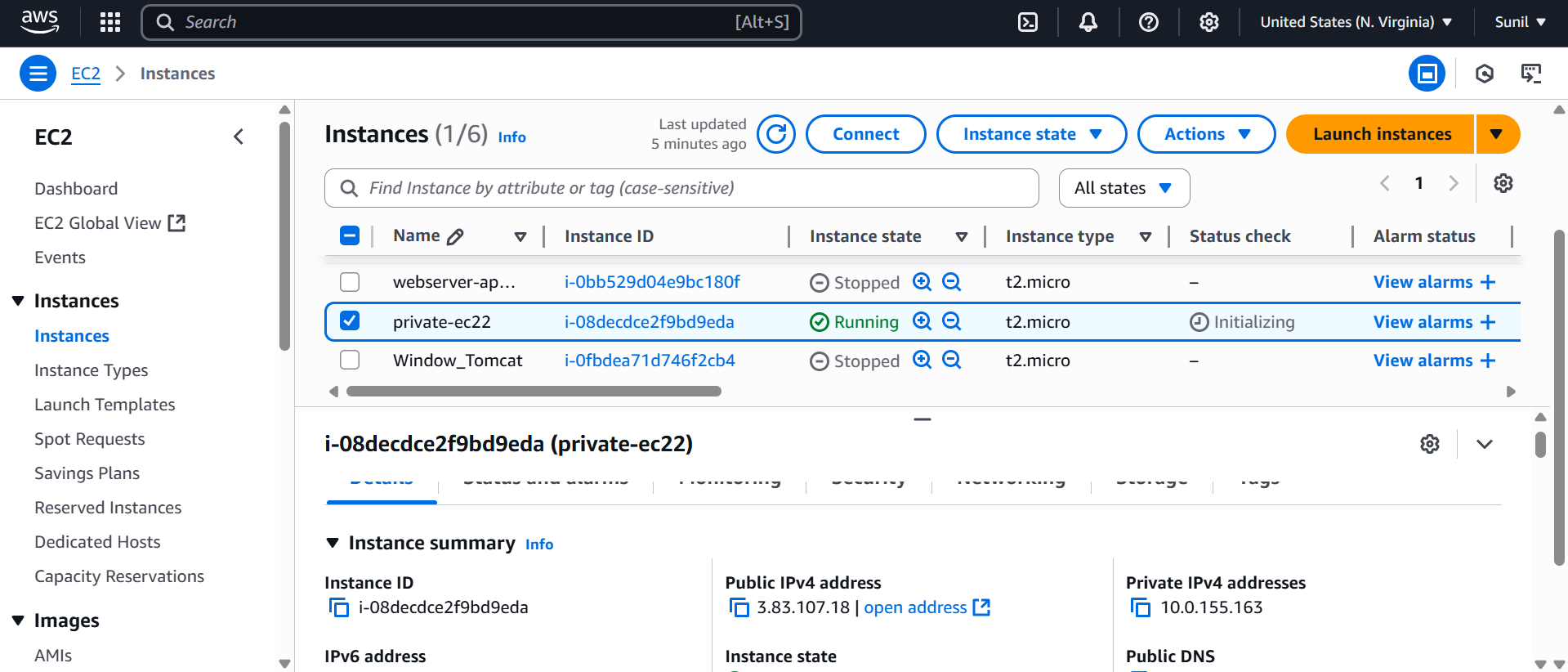
1. **Install Apache Tomcat in private ec2 and deploy a sample app.**

98.80.179.114 → Your **public EC2's public IP**

10.0.155.163 → Your **private EC2's private IP**

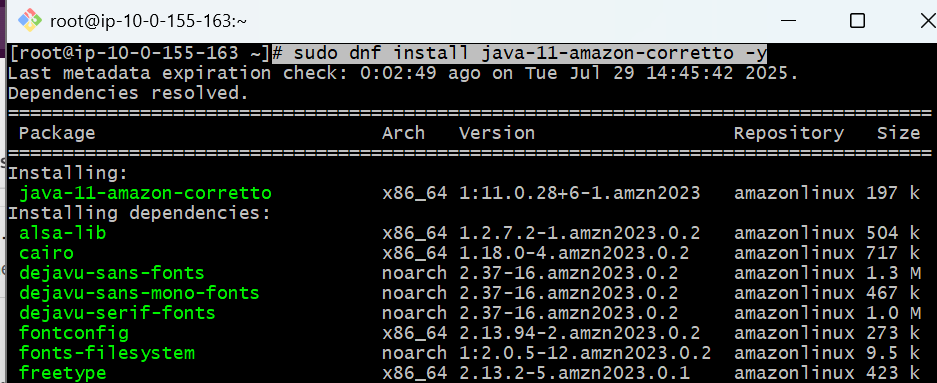
linux-testing.pem → Your **EC2 private key**, stored on your **local machine**





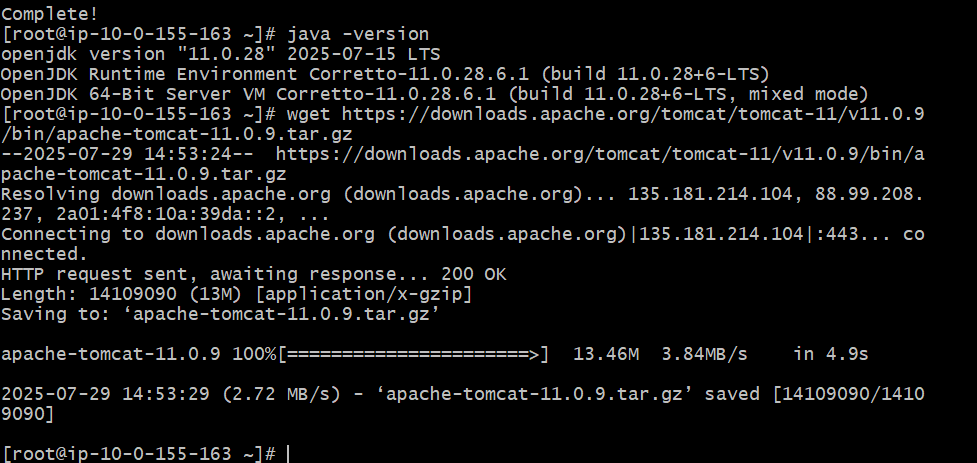
To install java in Gitbash

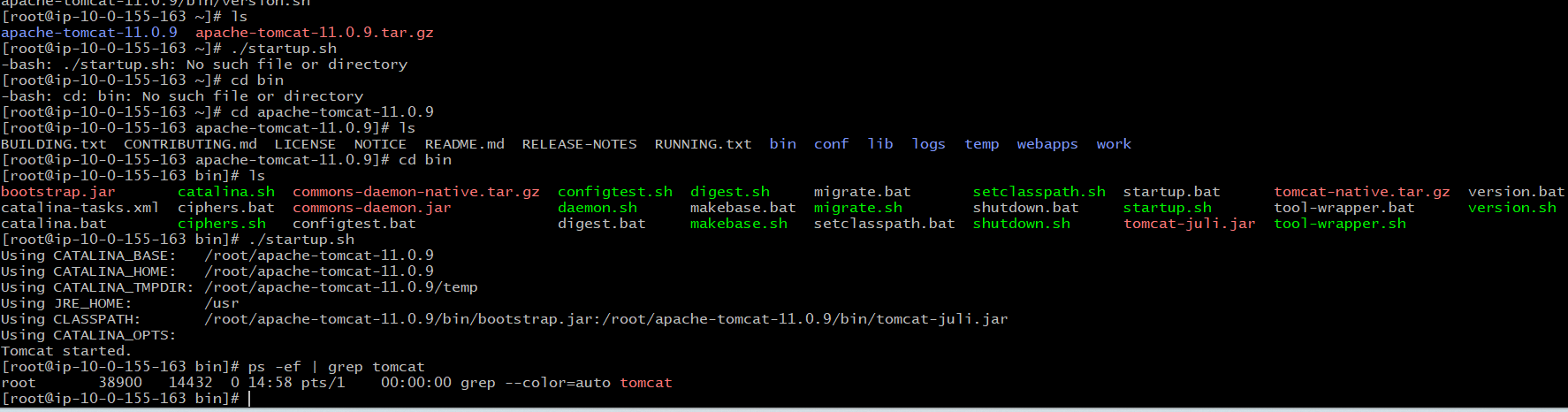
**Sudo dnf install java-11-amazon-corretto -y**

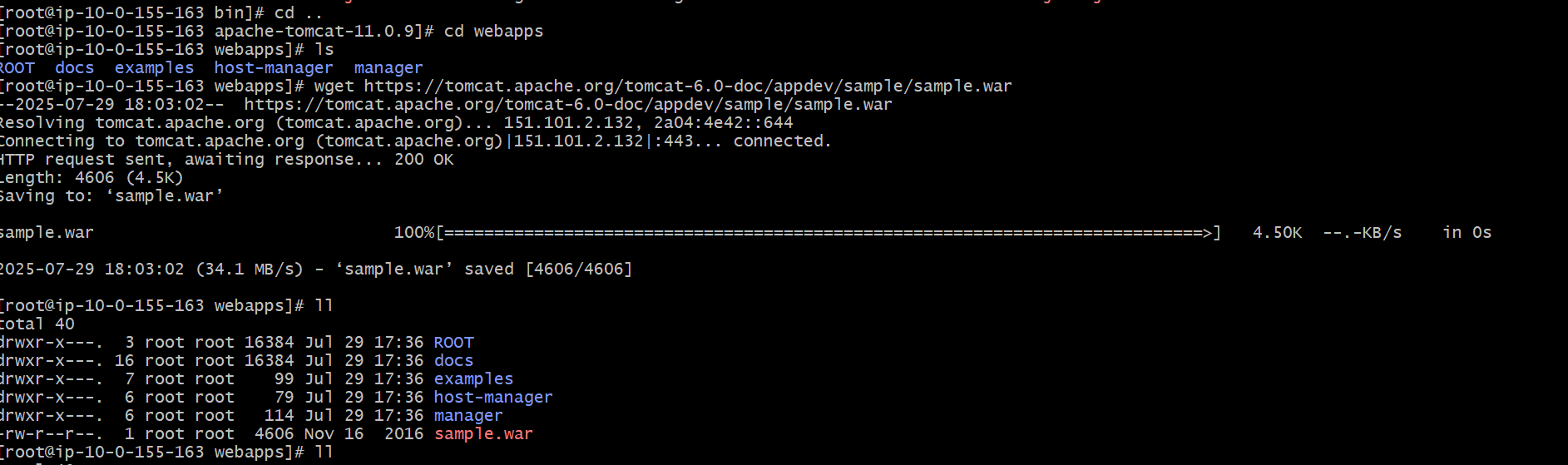


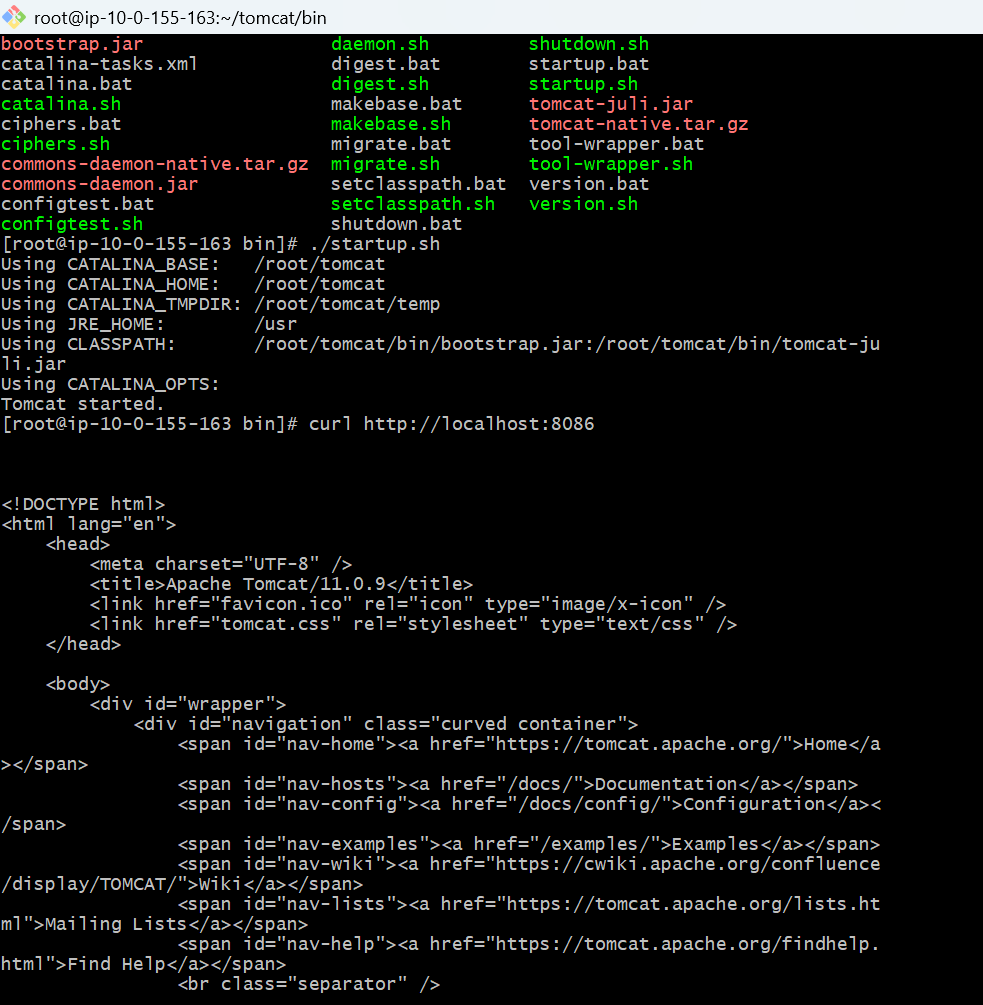
To check java version

**Java -version**









ssh -i linux-testing.pem -L 8086:10.0.155.163:8086 [ec2-user@98.80.179.114](mailto:ec2-user@98.80.179.114)

**Purpose of This Command:**

You're creating a **secure tunnel** from your **local computer** (your laptop or desktop) to a **private EC2 instance**, using a **public EC2 as a jump host**.

This allows you to **open http://localhost:8086 in your browser**, and behind the scenes, it connects to **Tomcat on the private EC2**.

is now creating a secure tunnel from:

* **Your local machine** localhost:8086 →
* **Through the public EC2** (98.80.179.114) →
* **To the private EC2** 10.0.155.163:8086 (where Tomcat is running)

 98.80.179.114 → your **public EC2's public IP**

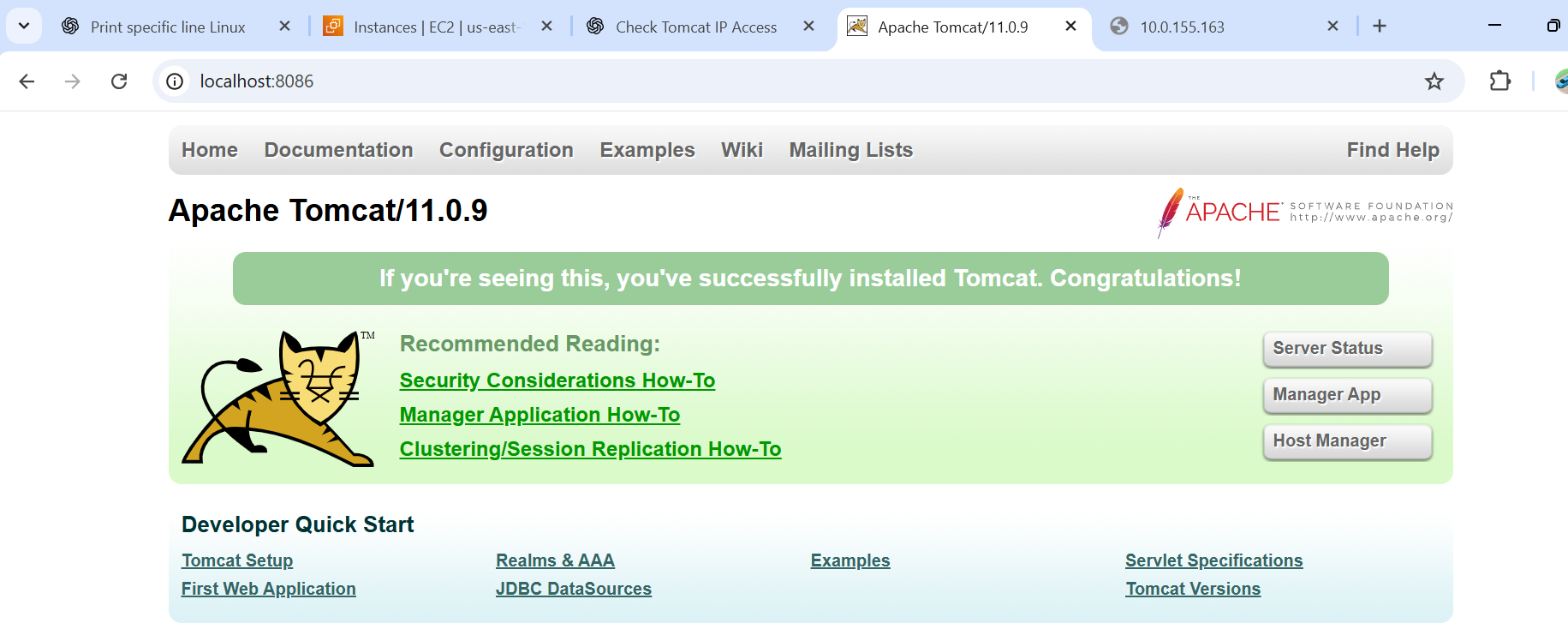
 10.0.155.163 → your **private EC2's private IP**

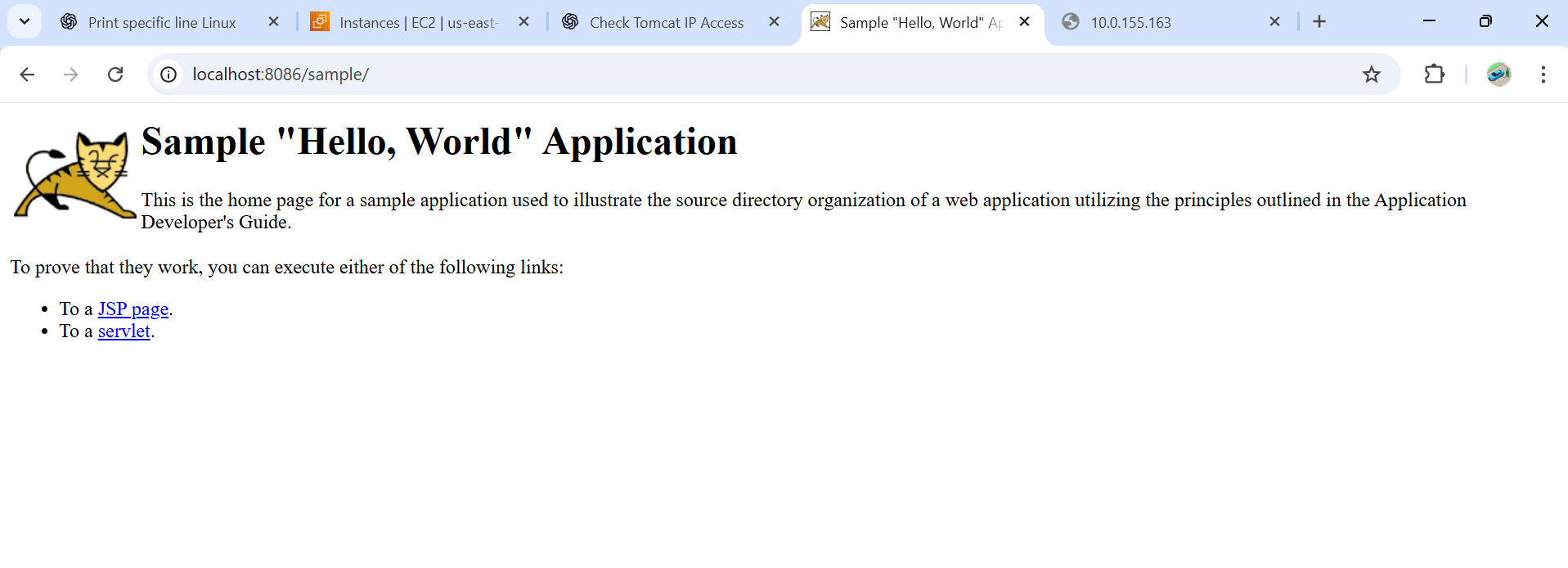
 linux-testing.pem → your **EC2 private key**, stored **on your local machine**

**Purpose of This Command:**

You're creating a **secure tunnel** from your **local computer** (your laptop or desktop) to a **private EC2 instance**, using a **public EC2 as a jump host**.

This allows you to **open http://localhost:8086 in your browser**, and behind the scenes, it connects to **Tomcat on the private EC2**.



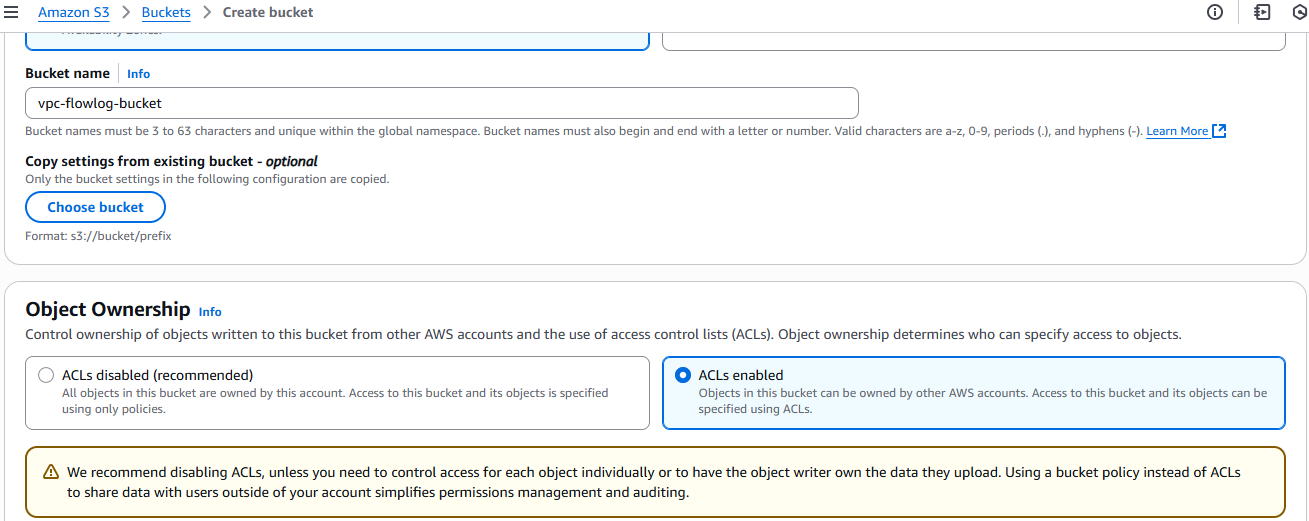


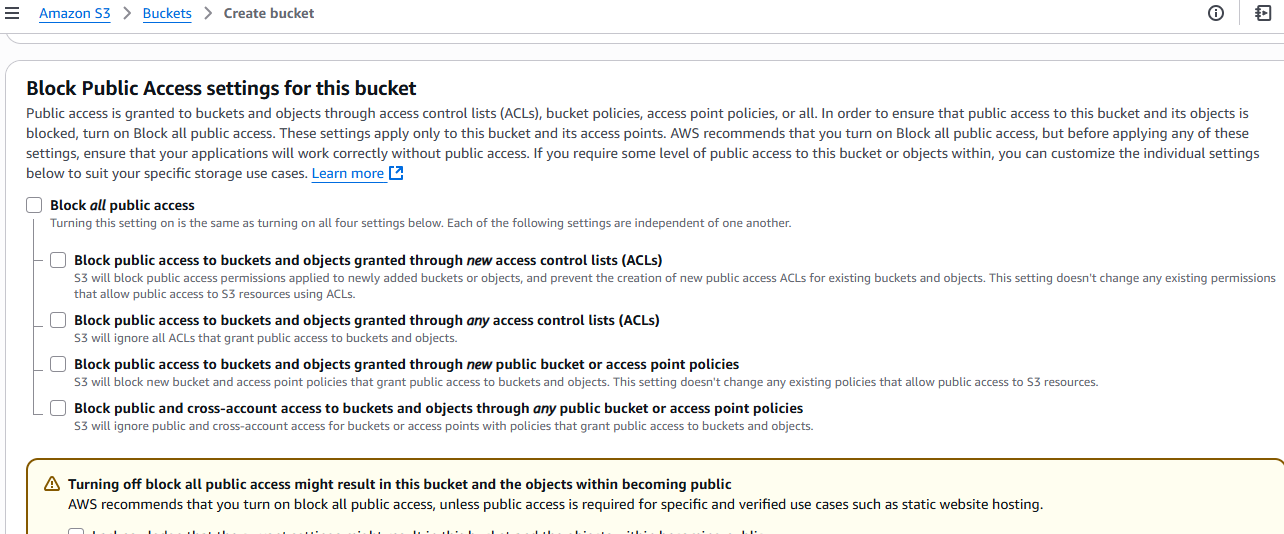
**10)Configure VPC flow logs and store the logs in s3 and CloudWatch.**.

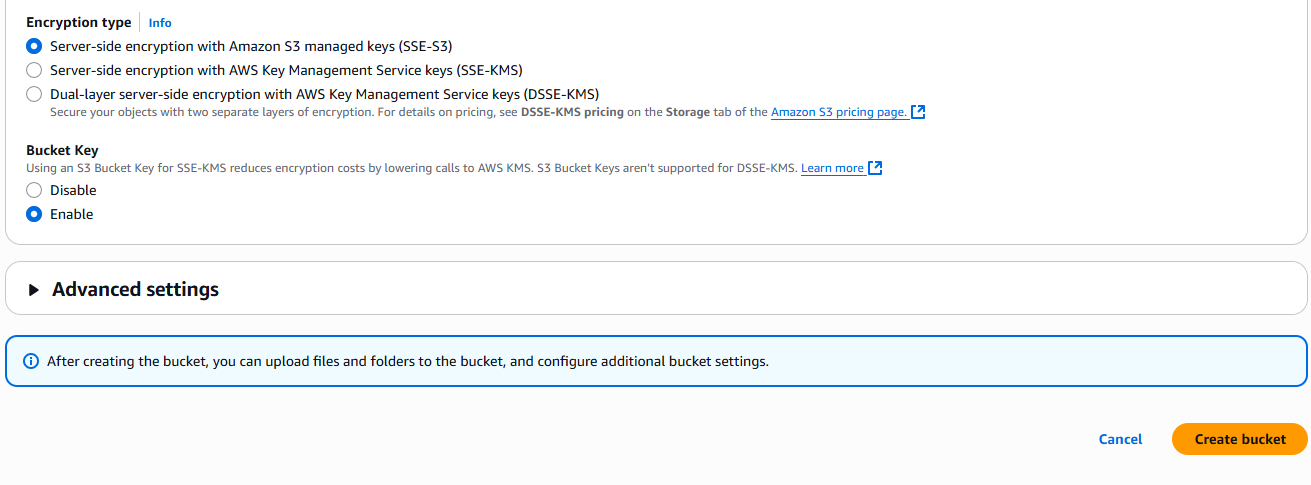
**Configure VPC flow logs and store the logs in s3**

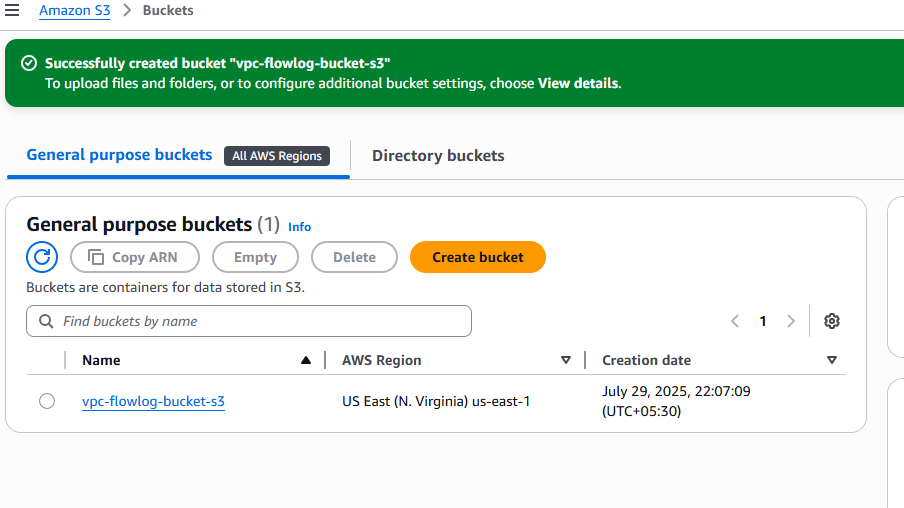
**STEP:1**

Amazon S3 ---> Bucket ---> Create Bucket --->



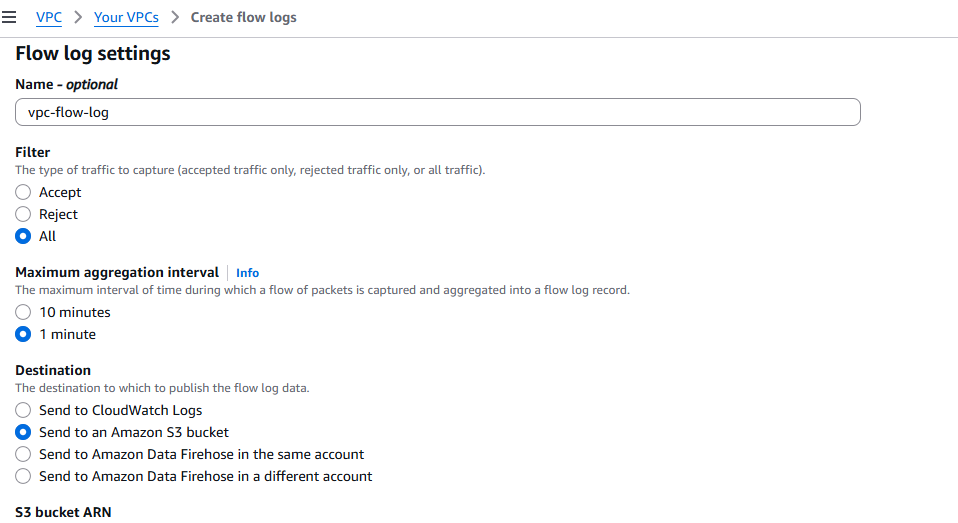




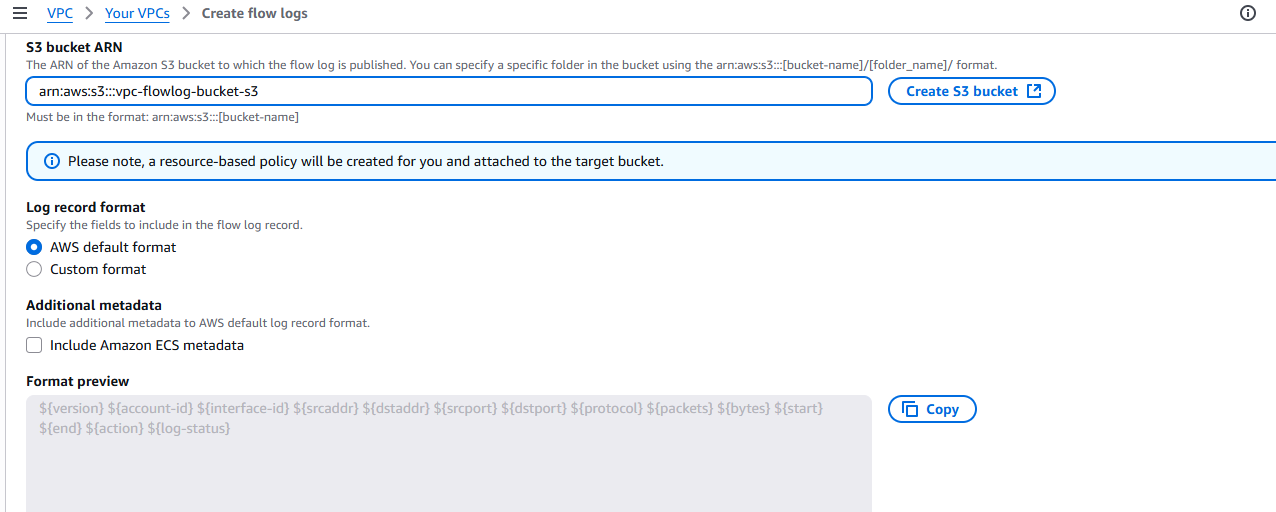


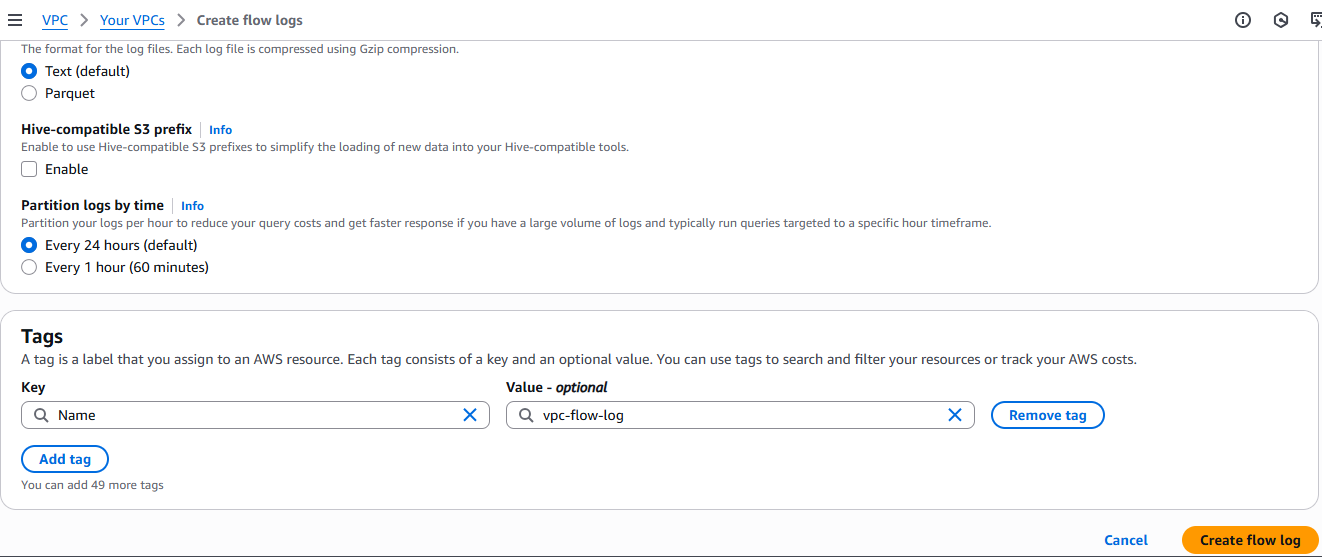
**STEP:2**

VPC ---> Select your VPC ---> Go to Flow Log ---> Click Create Flow Log



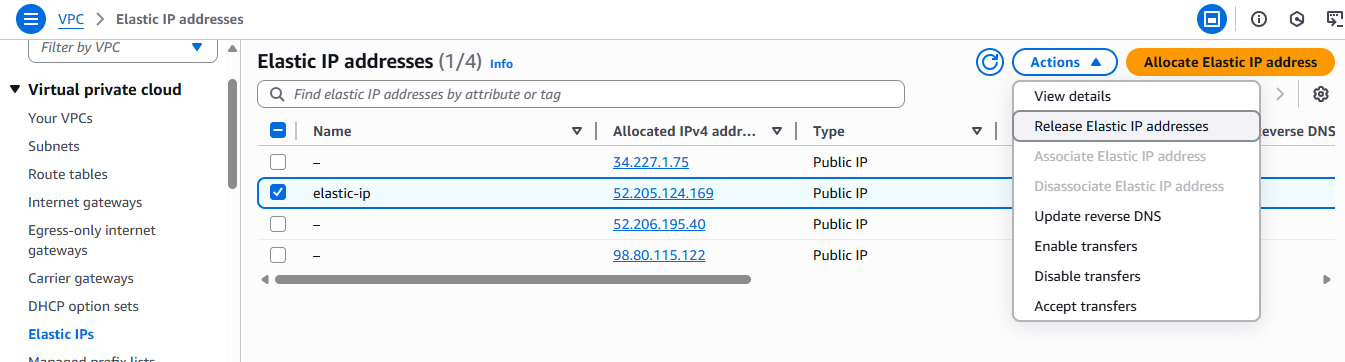
Add created s3 bucket ARN



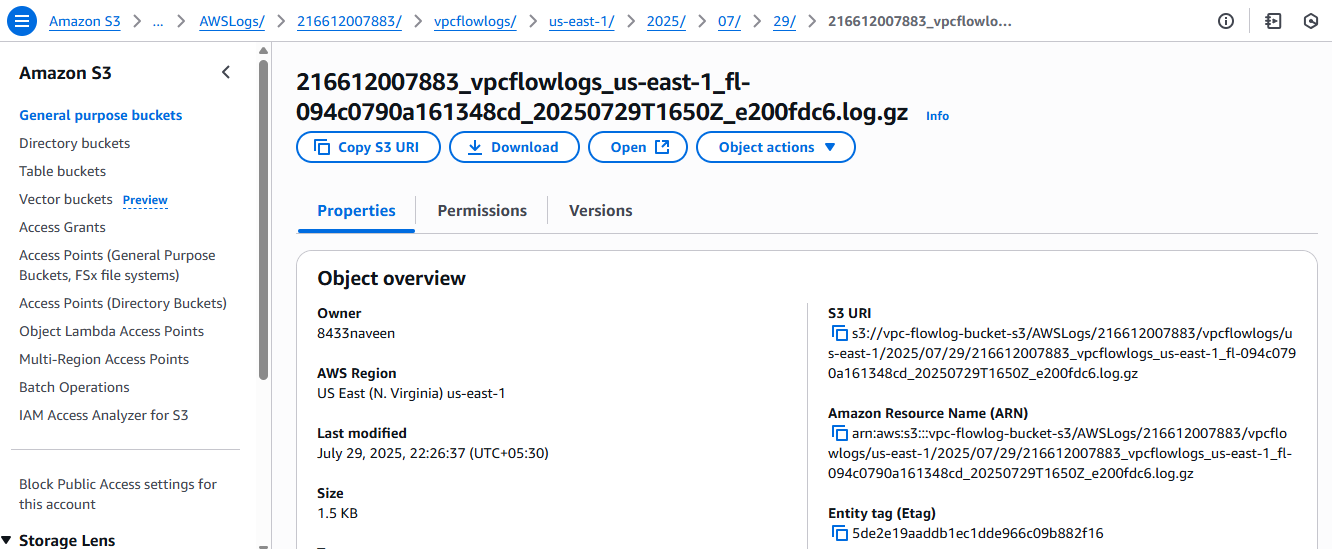


**STEP:3**

VPC ---> Elastic IPs ---> Select Elastic IP ---> Action ---> Release Elastic IP Addresses

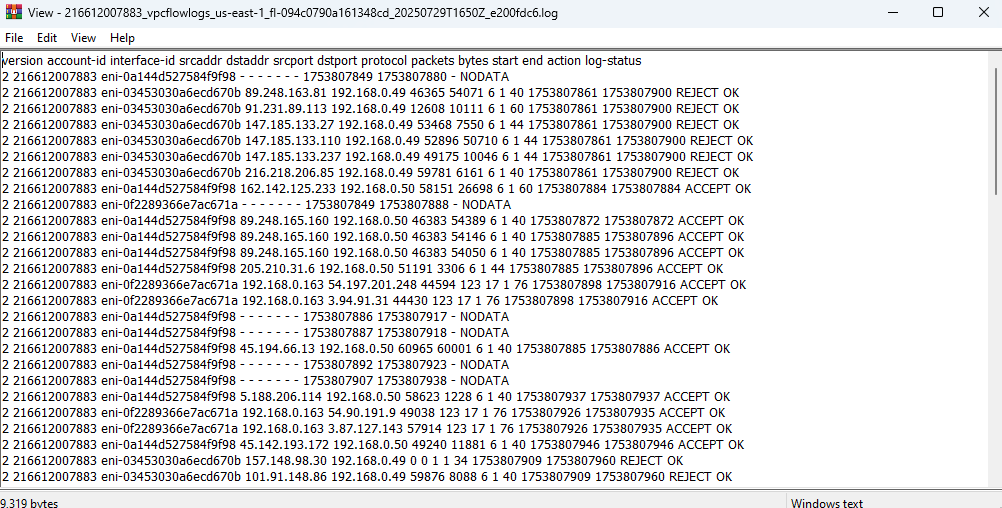


Then go to Amazon S3 where file is saved then download the file to local system.



**STEP:4**

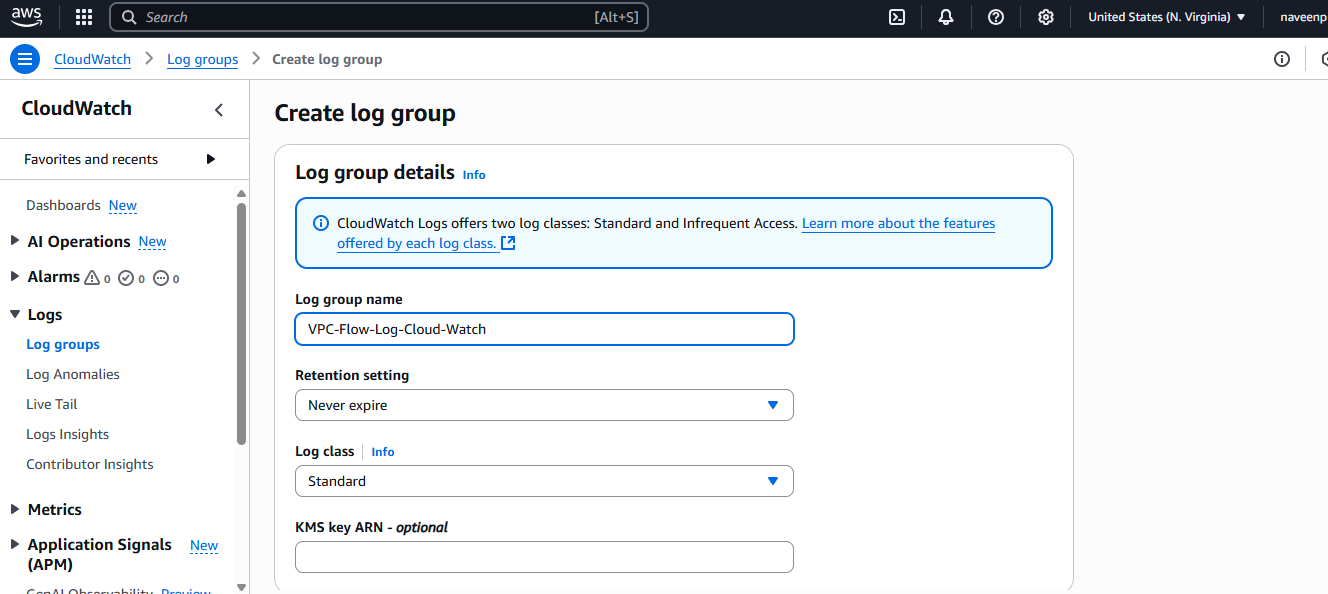
Open the downloaded war file



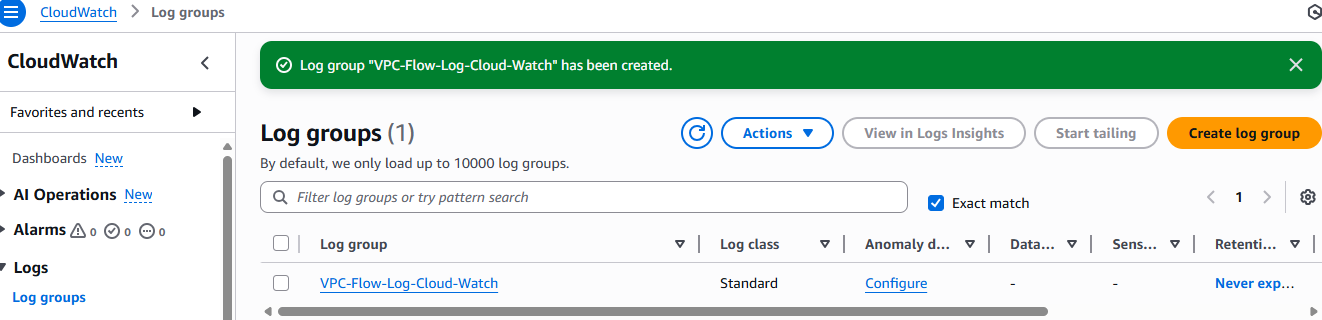
**Configure VPC flow logs and store the logs in CloudWatch.**

**STEP:1**

Cloudwatch ---> Logs ---> Log groups ---> Create Log Group

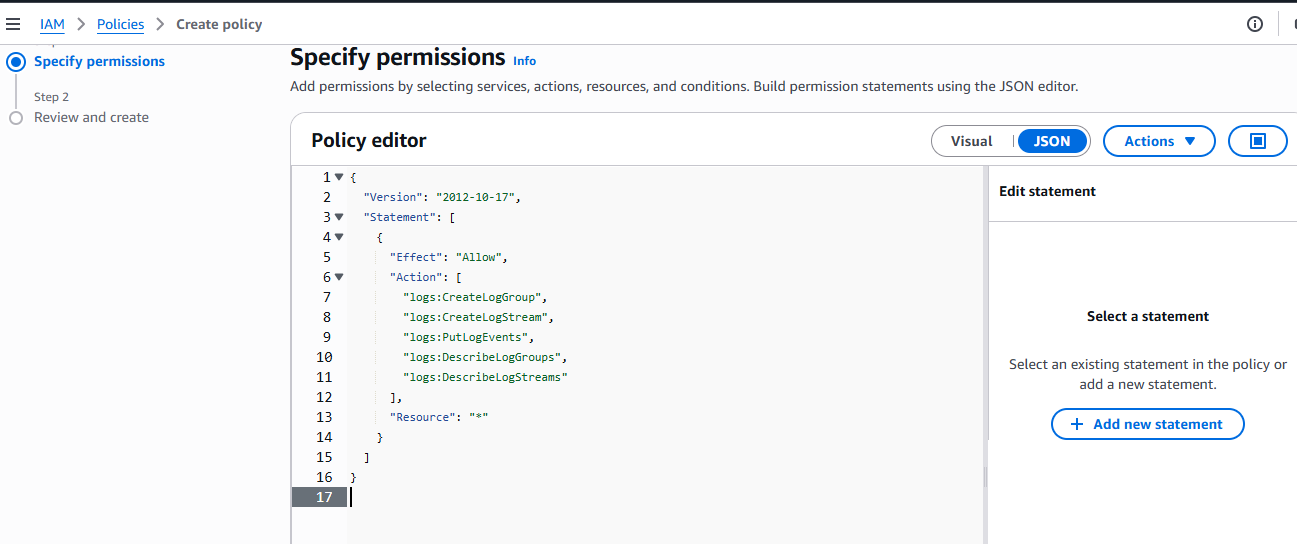


Click Create the Log groups will be created.

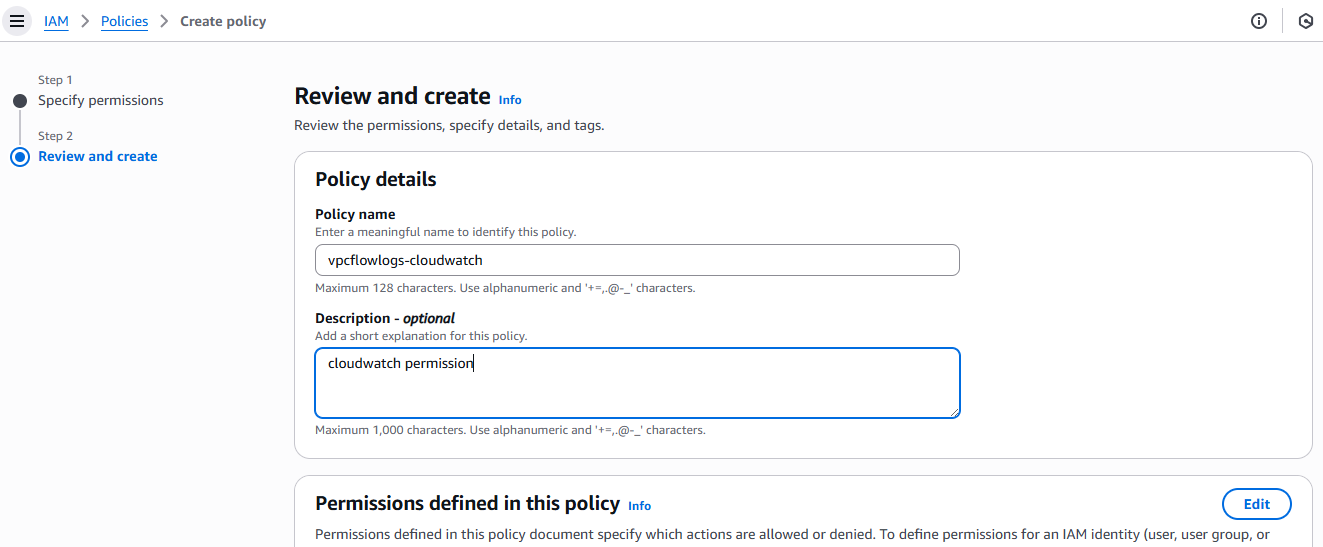


**STEP:2**

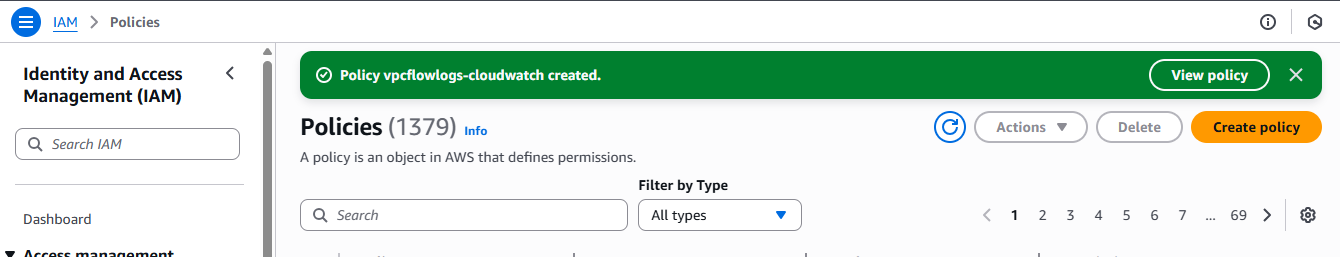
IAM ---> Policies ---> Create policy --->Select polocy editor in JSON format --->



Click Next

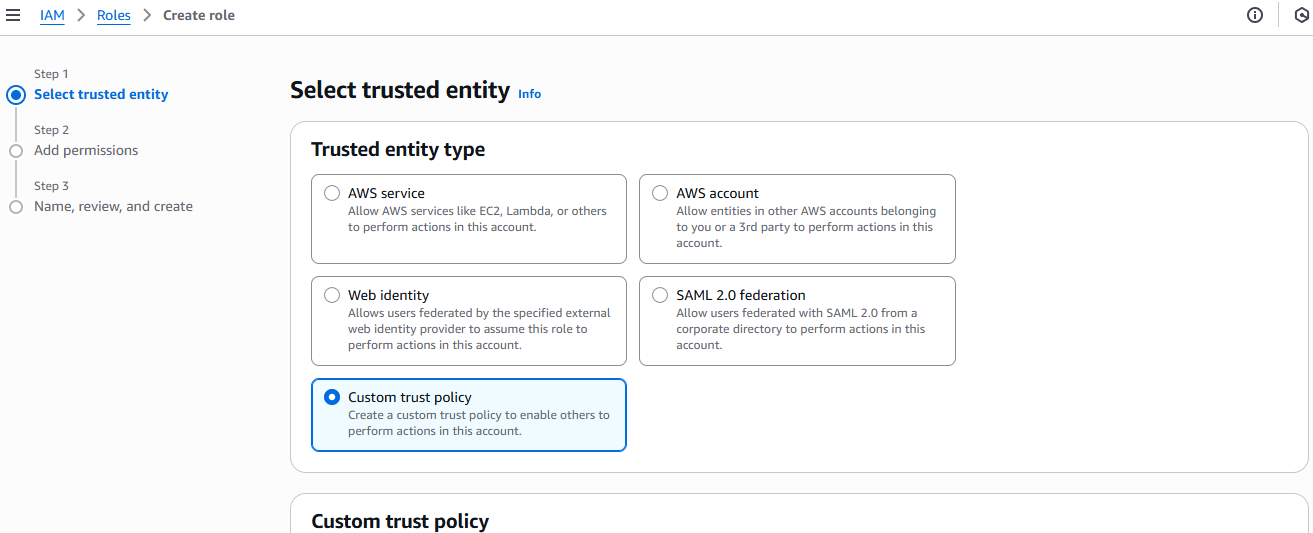


Click create then policy is created.



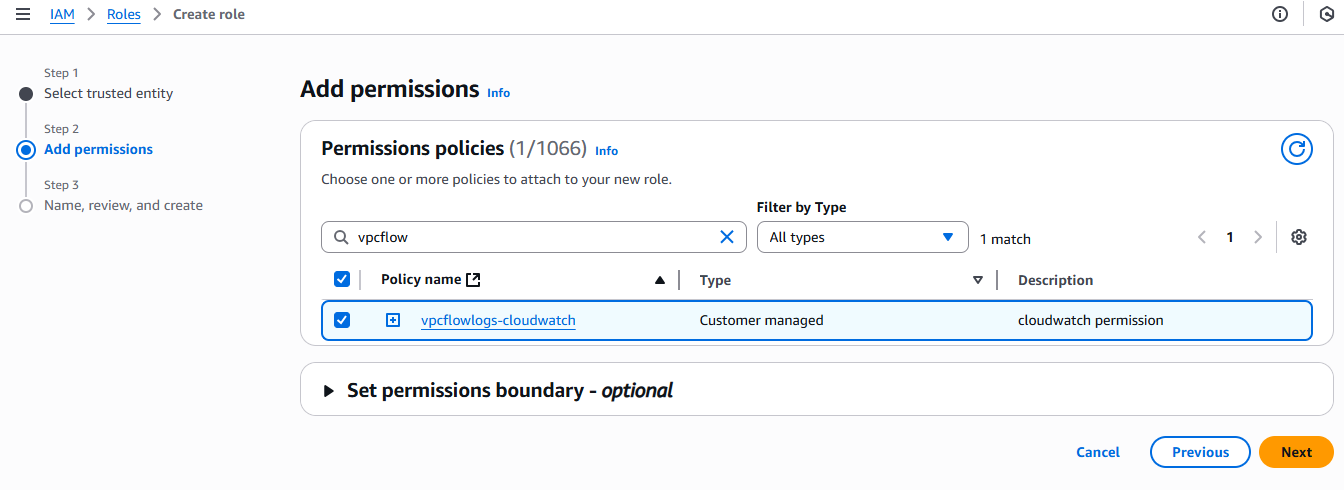
**STEP:3**

IAM ---> Roles ---> Create role ---> select custom trust policy ---> click Next

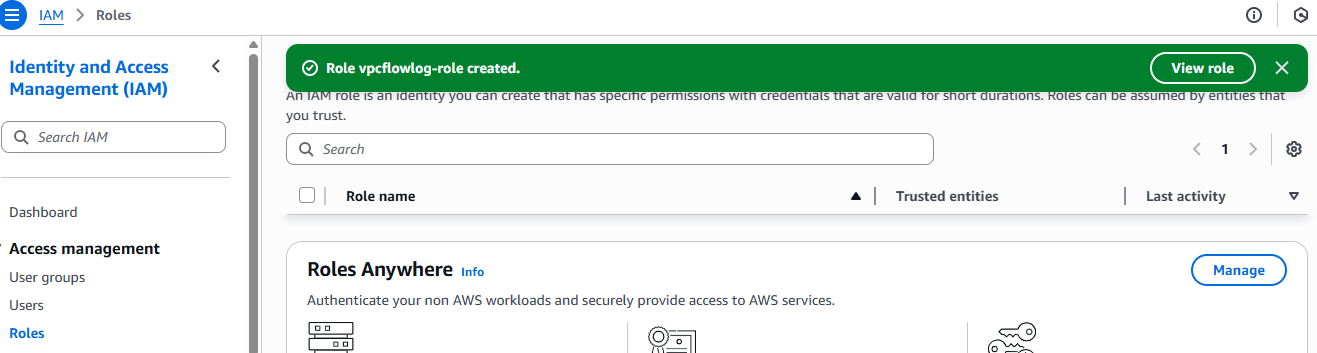




Add created policy in Permission Policy ---> Click Next

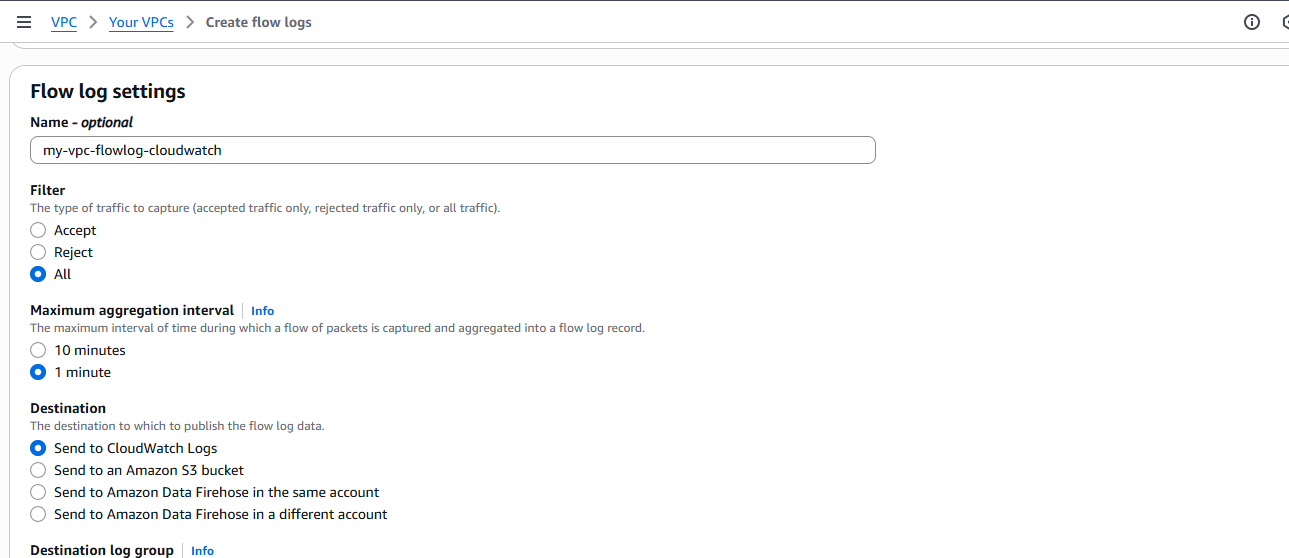


Enter Rolename then review the trusted entities and permission ---> click create Role

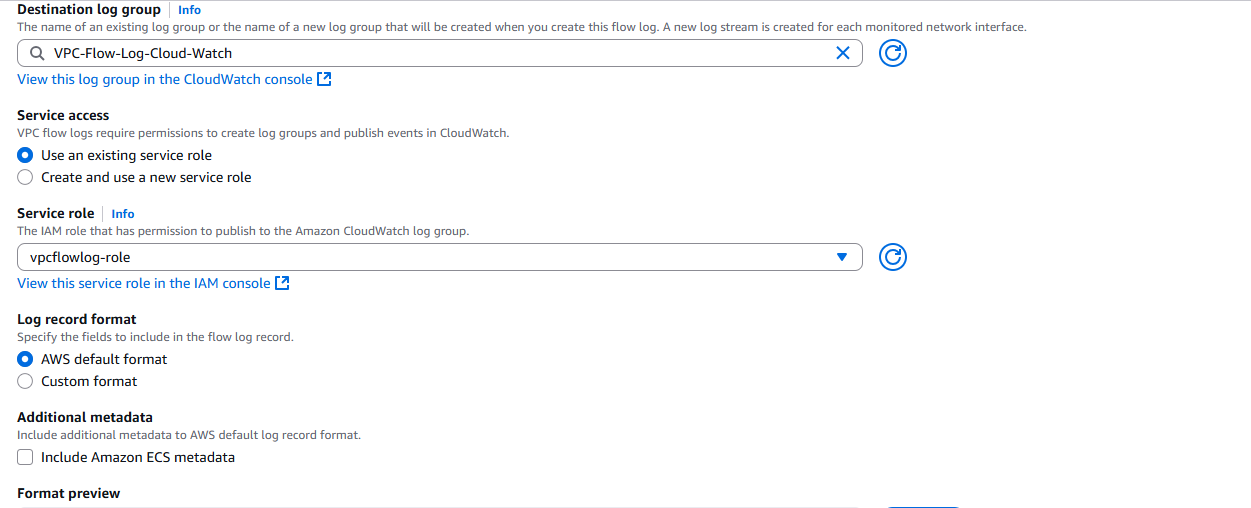


**STEP:4**

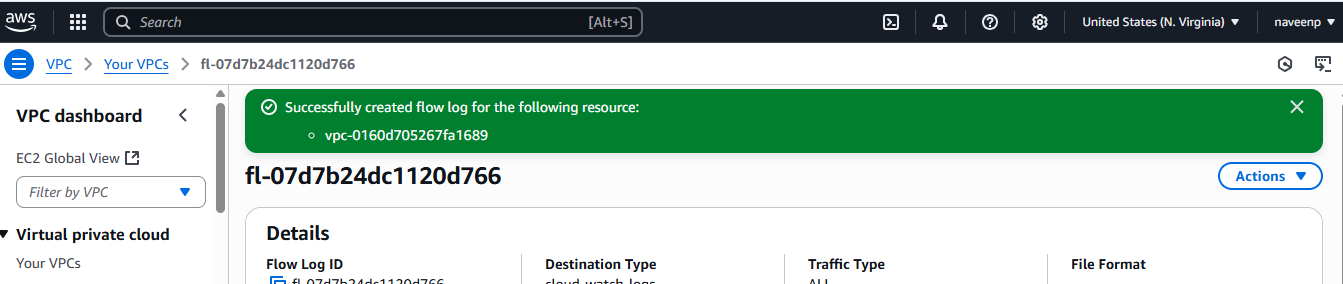
VPC ---> Your VPCs --->select your vpc ---> Flow Logs --->Create Flow log



Add your Cloudwatch(VPC-Flow-Log-Cloud-Watch) in destination log group and Add your IAM Role (vpcflowlog-Role) in IAMRole.



Click create flow log then Flow log is created.



CloudWatch ---> Log Groups ---> Select created log group ---> Select Log streams

