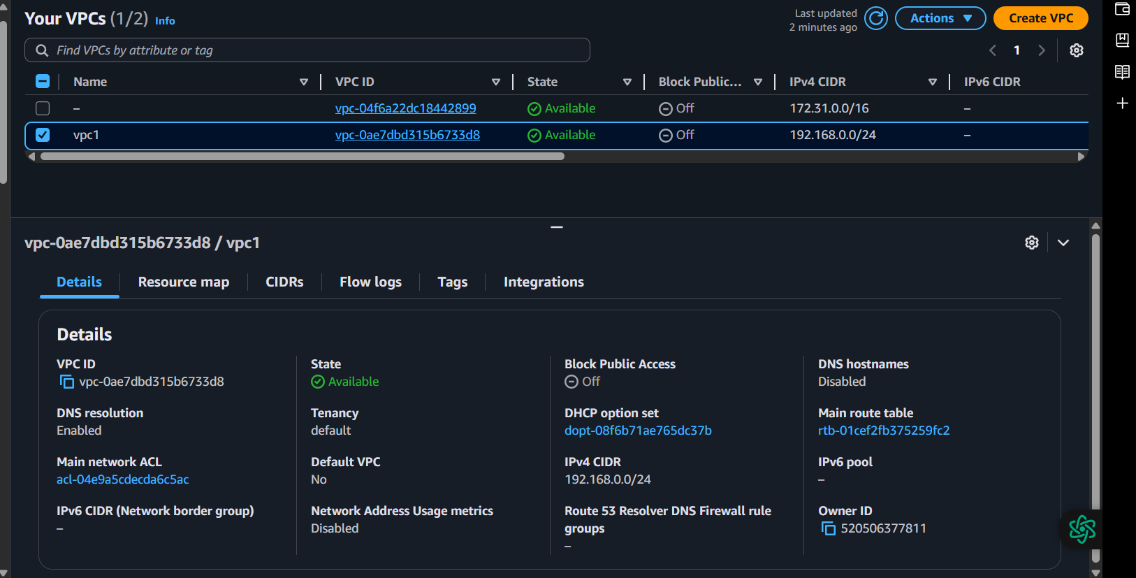
1. Create VPC with 2 private and 2 public subnets.

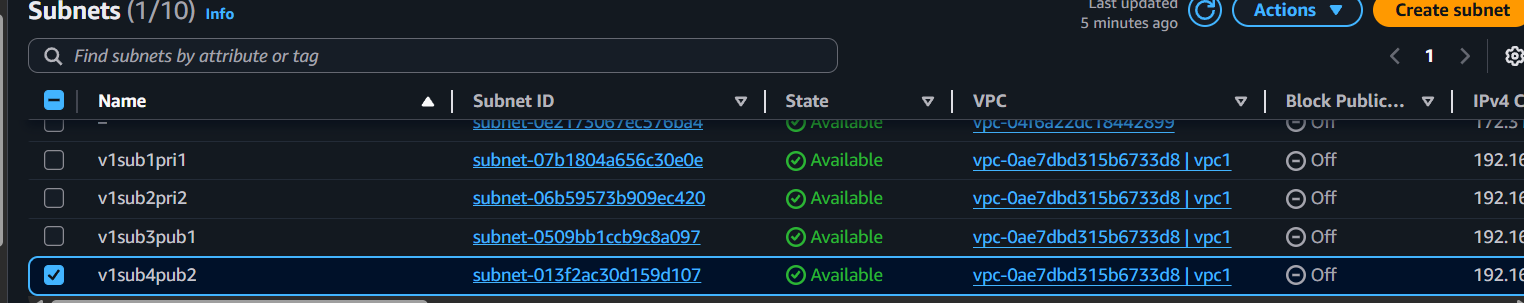
 Go to **AWS Management Console → VPC → Your VPCs → Create VPC**.

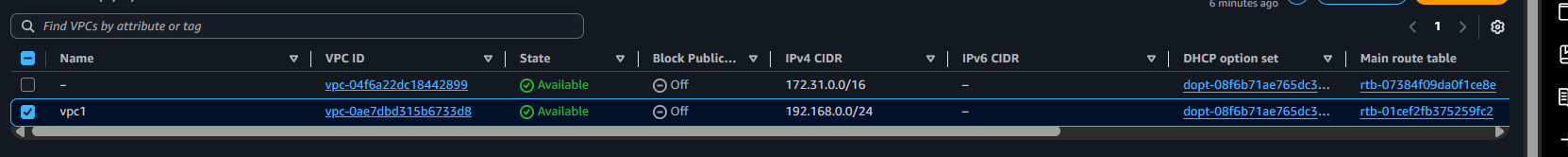
* **Name tag**: MyVPC
* **IPv4 CIDR block**: 10.0.0.0/16
* Enable **DNS Resolution** and **DNS Hostnames** (or do in step 2).
* Click **Create VPC**.

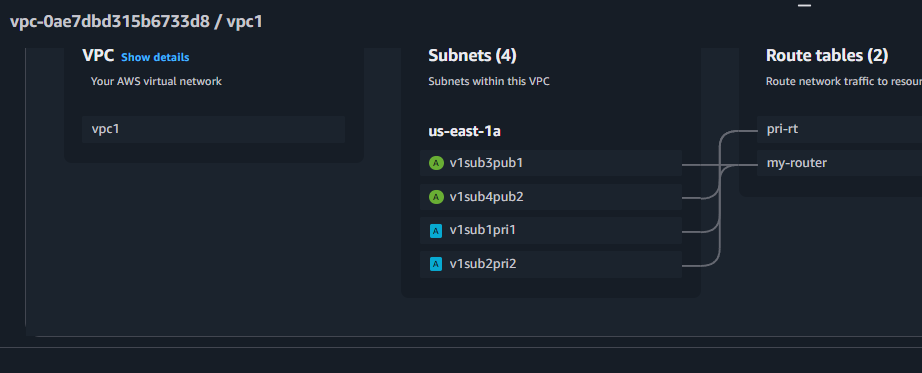
 Create **subnets**:

* **Subnet 1 (Public)**: 10.0.1.0/24
* **Subnet 2 (Public)**: 10.0.2.0/24
* **Subnet 3 (Private)**: 10.0.3.0/24
* **Subnet 4 (Private)**: 10.0.4.0/24
* Assign subnets to **2 different AZs** for high availability

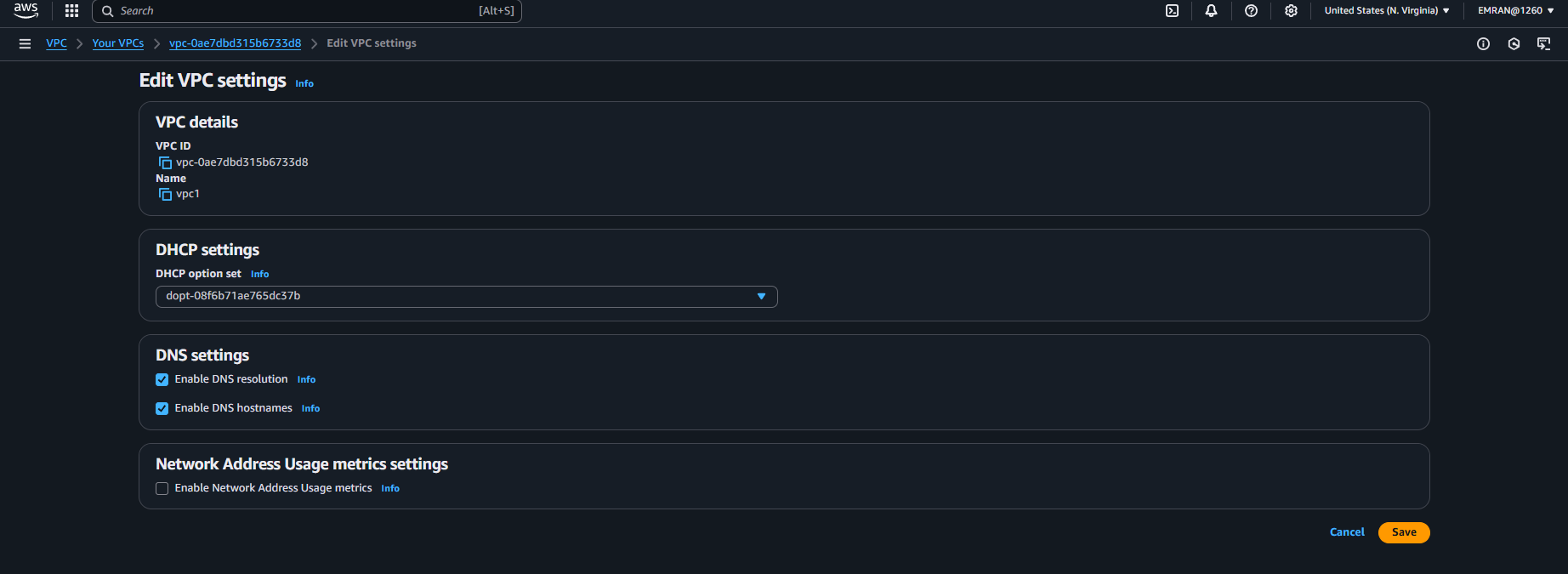








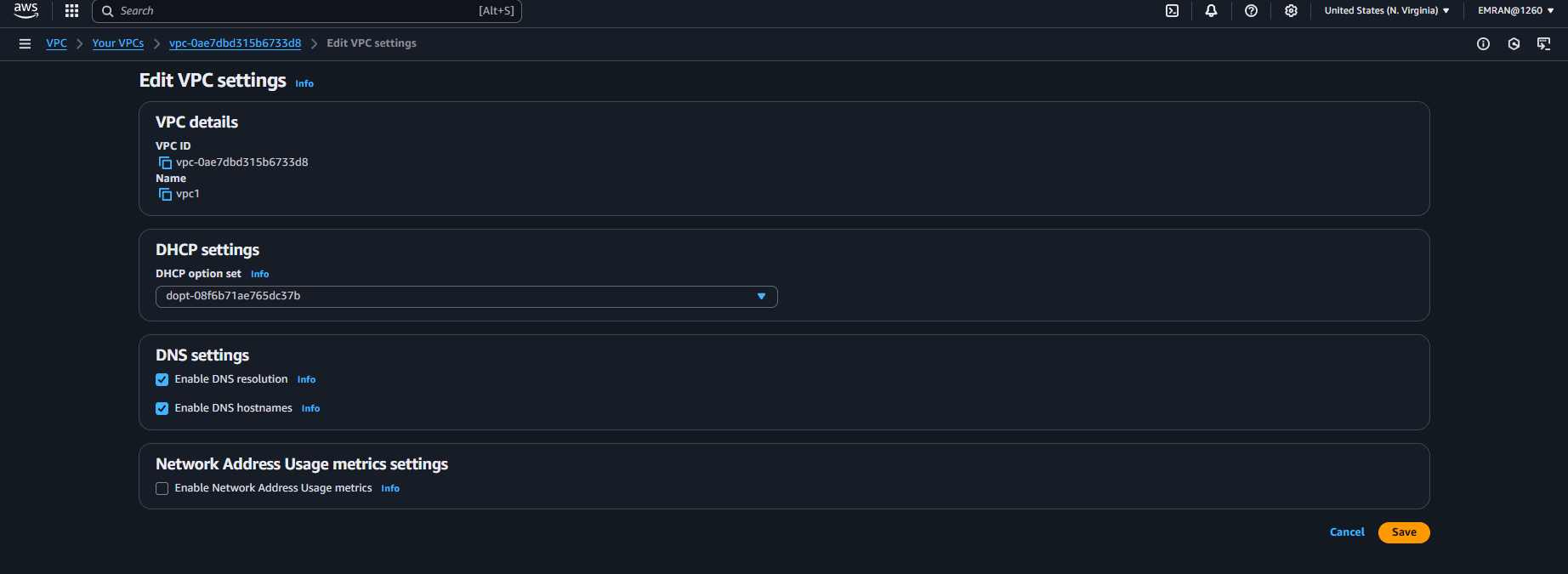
* 2) Enable DNS Hostname in VPC  
  Go to **VPC → Your VPCs → Select VPC → Actions → Edit DNS Hostnames → Enable**.

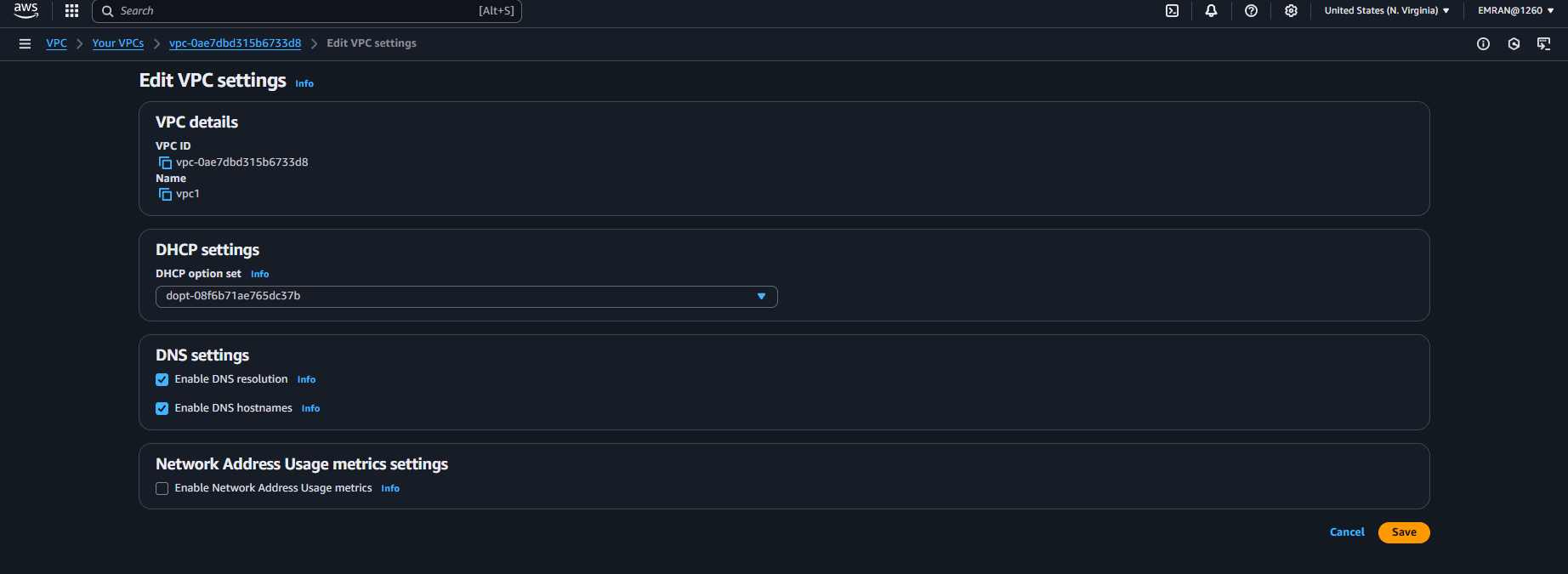


1. Enable Auto Assign Public ip in 2 public subnets  
   Go to **Subnets → Select Public Subnet → Actions → Modify auto-assign IP settings**.

Check **Enable auto-assign public IPv4 address**.

Do this for both public subnets.

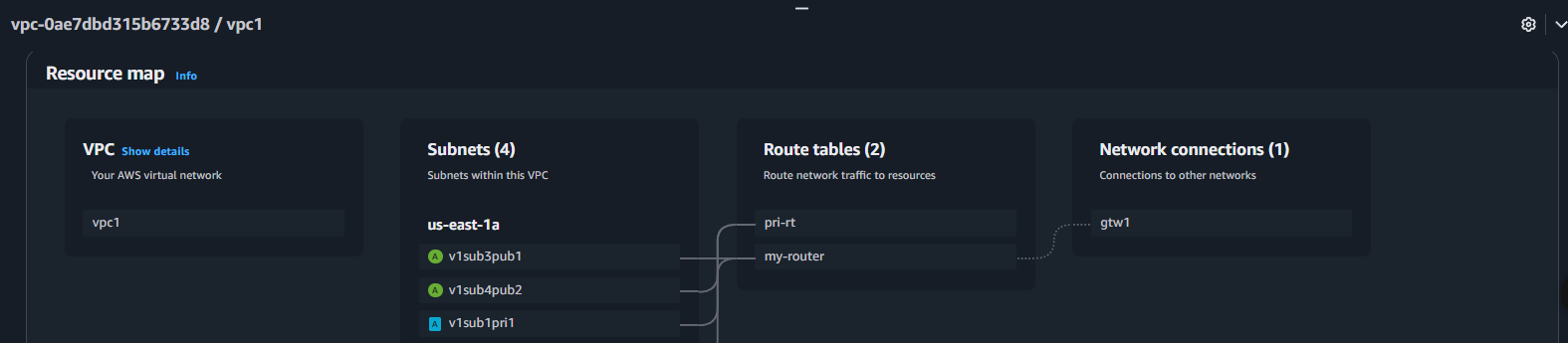




1. Add 2 private subnets in private route table

 Create **Route Table**: Private-RT.

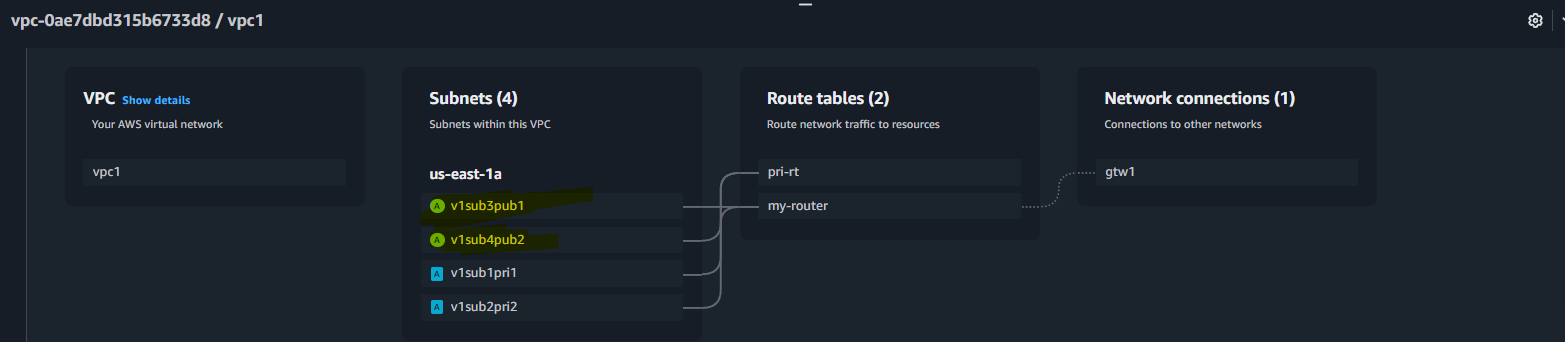
 **Associate** private subnets (10.0.3.0/24 and 10.0.4.0/24) with Private-RT.



1. Add 2 public subnets in public route table

 Create **Route Table**: Public-RT.

 **Associate** public subnets (10.0.1.0/24 and 10.0.2.0/24) with Public-RT.



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

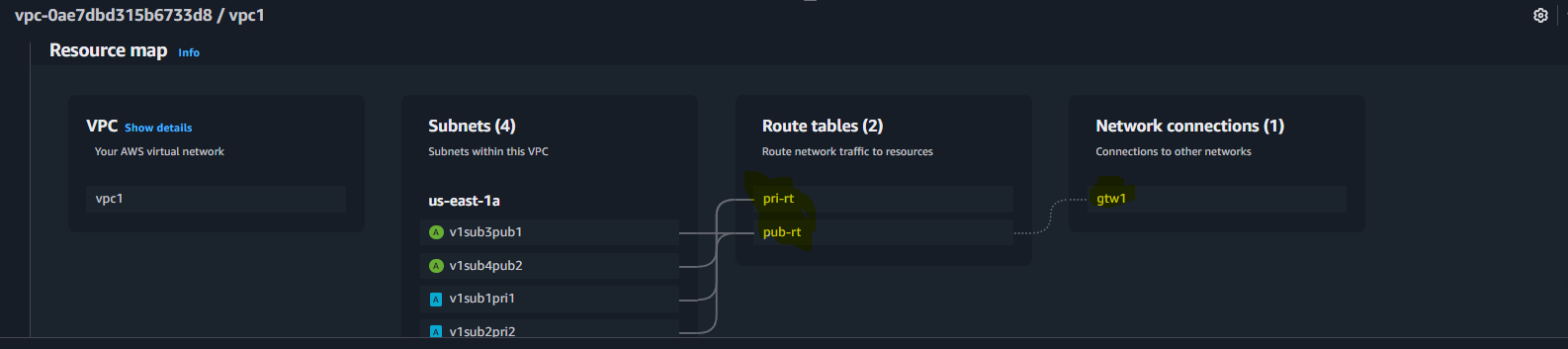
Create **Internet Gateway**:

* Go to **Internet Gateways → Create IGW → Attach to VPC**.

Add route in **Public-RT**:

* **Destination**: 0.0.0.0/0
* **Target**: Internet Gateway

Local route is added by default (10.0.0.0/16 → local).



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

 Go to **EC2 → Launch Instance**:

* Name: Public-EC2
* AMI: **Amazon Linux 2**
* Instance type: t2.micro
* Network: **MyVPC**
* Subnet: **Public Subnet (10.0.1.0/24)**
* Enable Auto-assign Public IP.

 Add **Security Group**:

* Allow SSH (port 22) from your IP.
* Allow HTTP (port 80) from anywhere.

After launch, **SSH into instance**:  
🡪 Install php  
sudo yum update -y

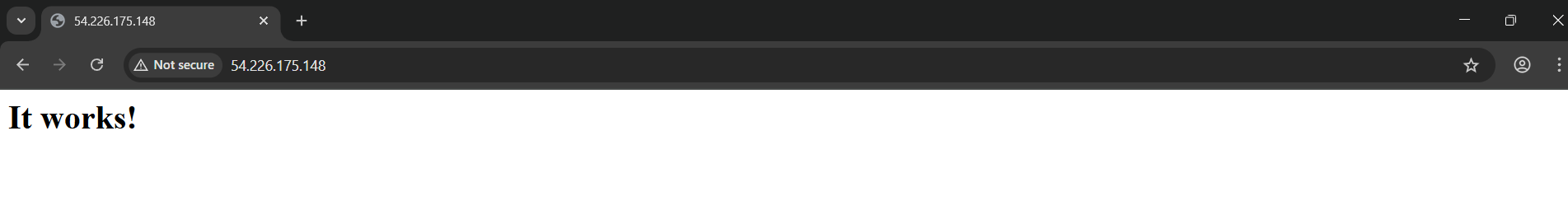
sudo amazon-linux-extras enable php8.0

sudo yum install -y php-cli php-pdo php-fpm php-json php-mysqlnd httpd

sudo systemctl enable httpd

sudo systemctl start httpd

echo "<?php phpinfo(); ?>" > /var/www/html/index.php



8) Configure Nat gateway in public subnet and connect to private Instance

**Create Elastic IP**.

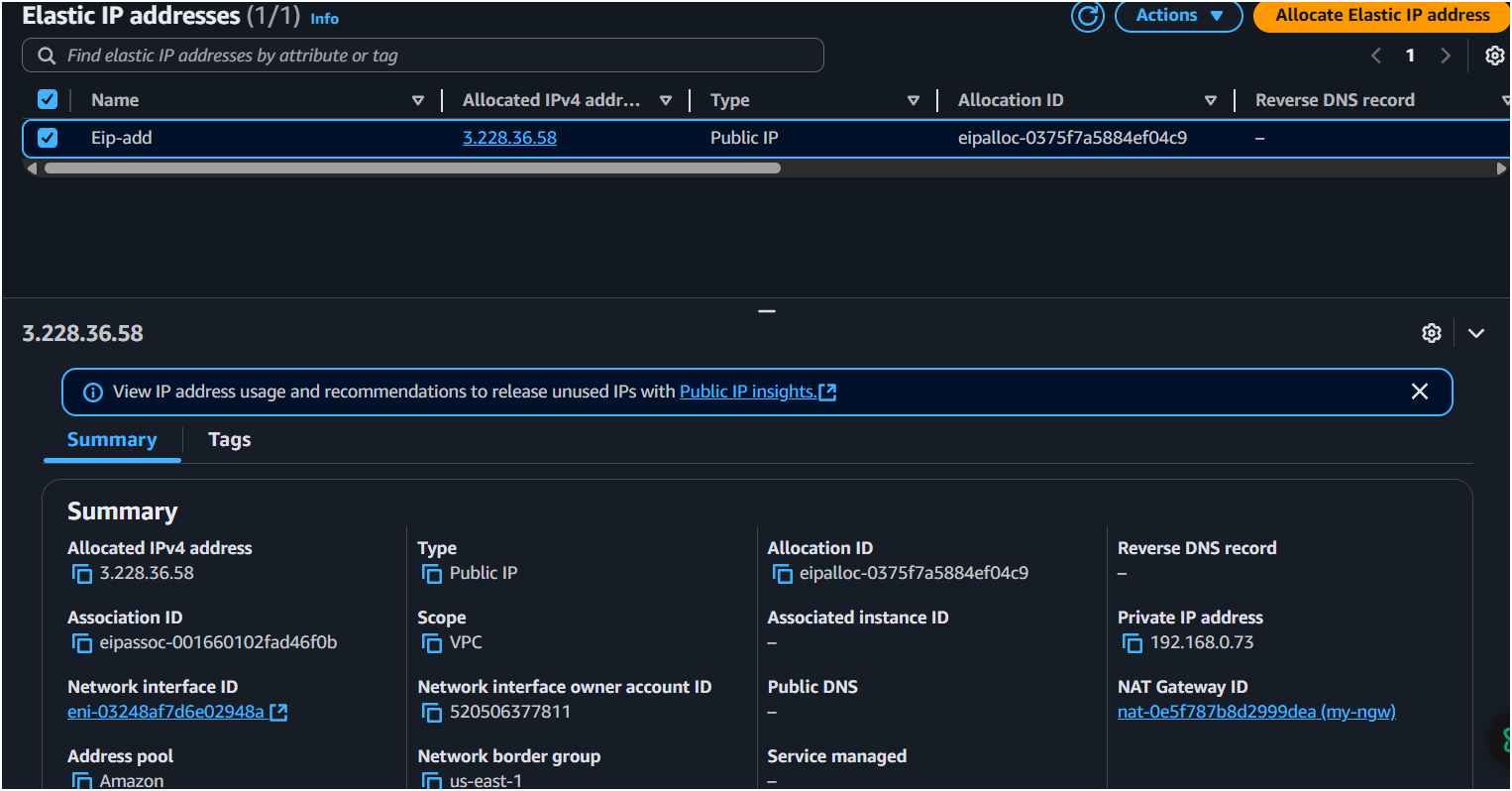
Go to **NAT Gateways → Create**:

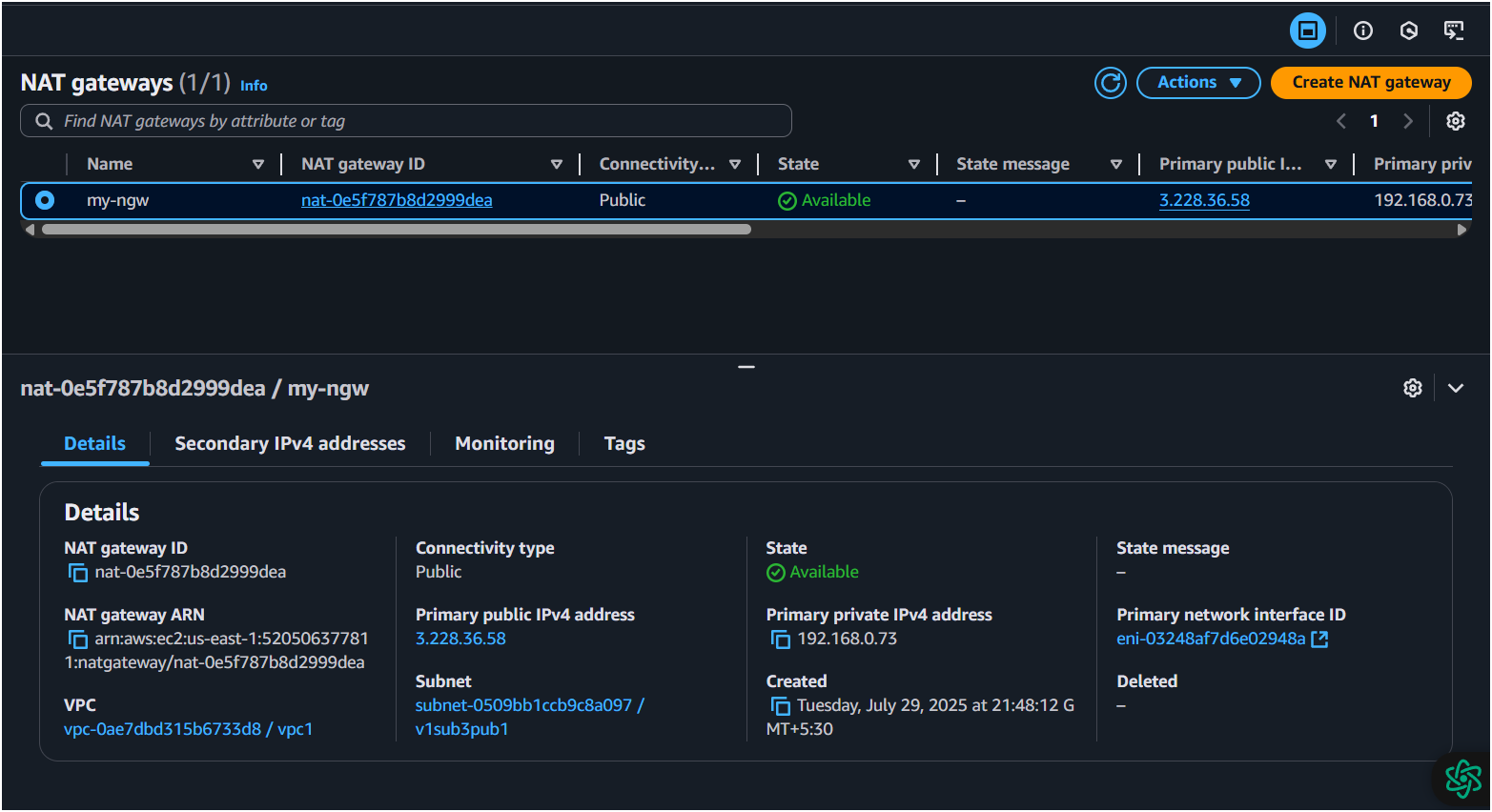
* Subnet: **Public Subnet (10.0.1.0/24)**
* Elastic IP: Select created one.

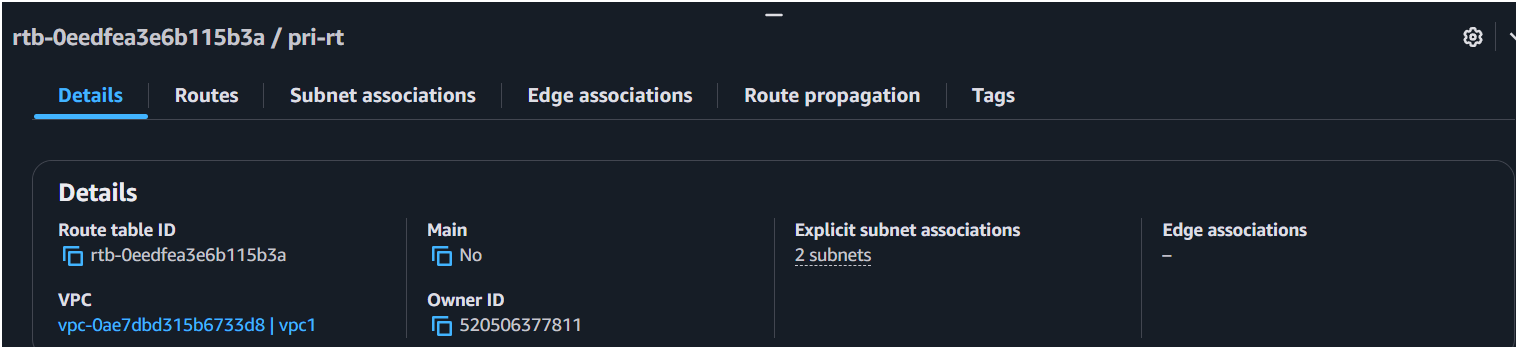
Add route in **Private Route Table**:

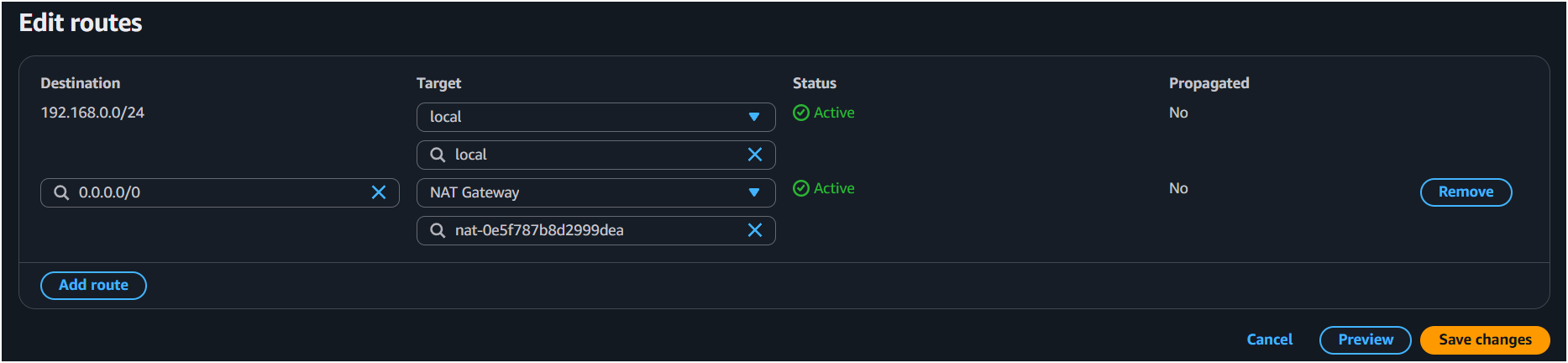
* **Destination**: 0.0.0.0/0
* **Target**: NAT Gateway

Private EC2 will now access internet via NAT.









9) Install Apache Tomcat in private ec2 and deploy a sample app.

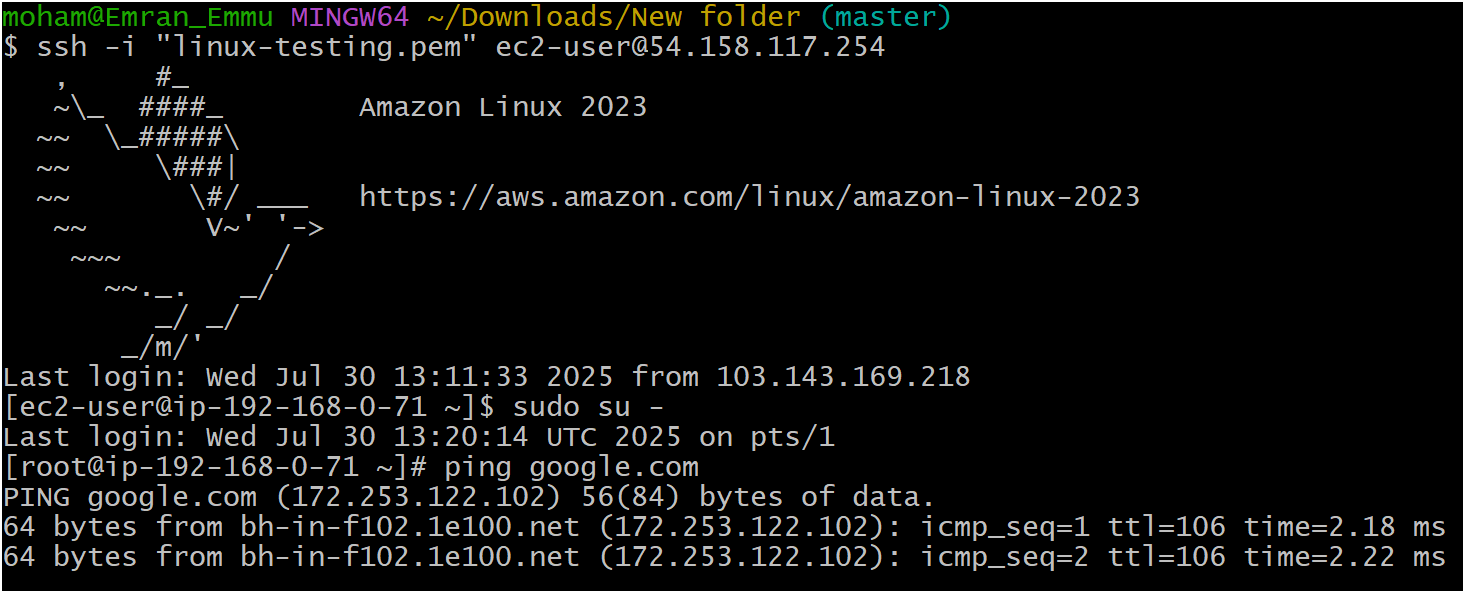
 Launch **EC2 in Private Subnet**:

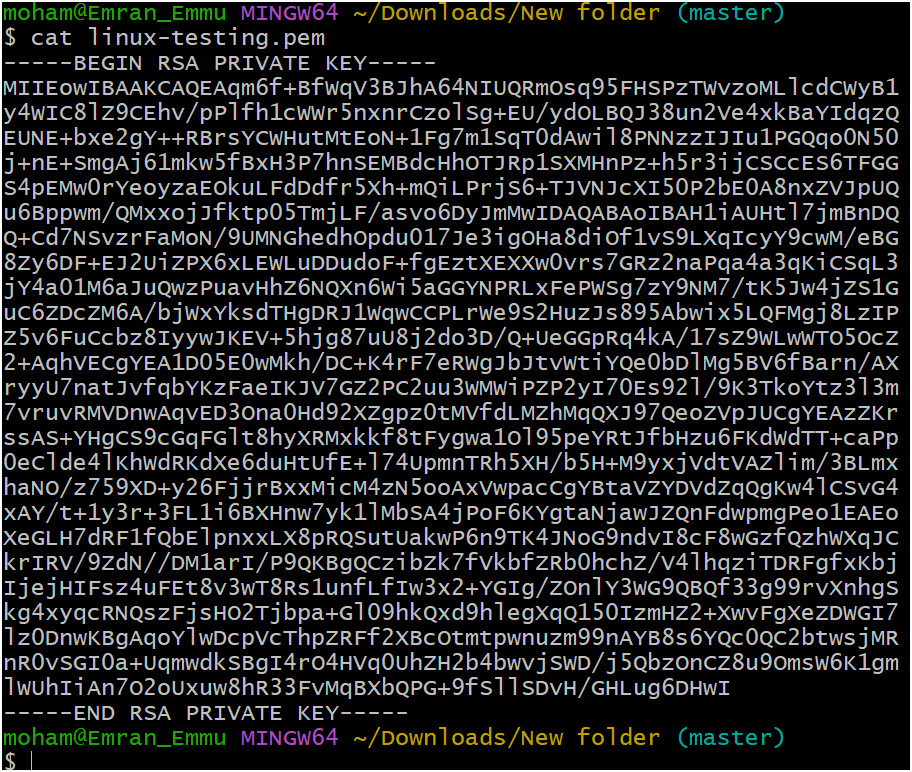
* Subnet: 10.0.3.0/24
* No public IP (because private).

 **SSH via Bastion Host** (Public EC2):

🡪 ssh -i my-key.pem ec2-user@<public-ec2-ip>

ssh ec2-user@<private-ec2-private-ip>





* Install Tomcat:

sudo yum update -y

cd /opt

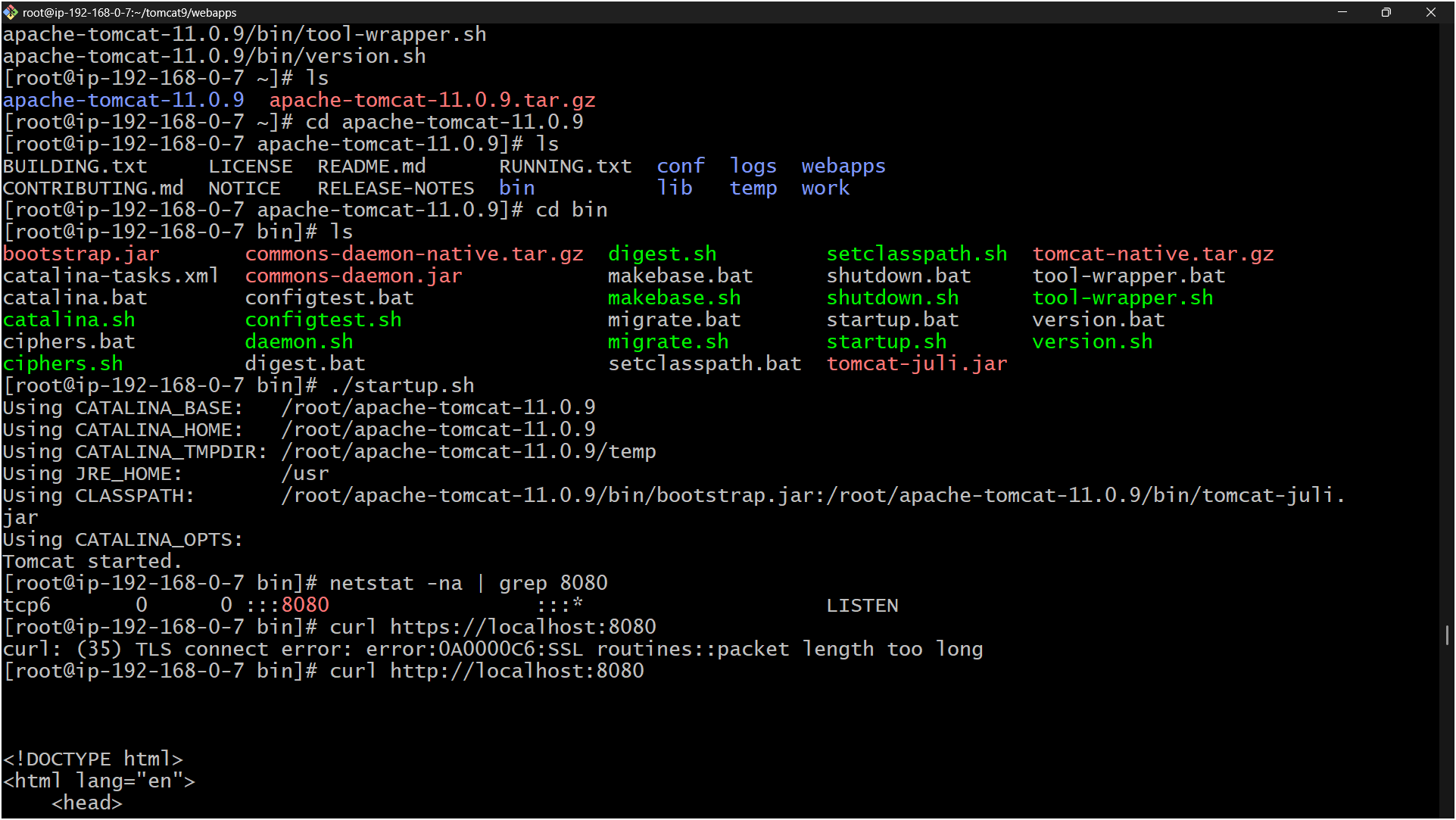
sudo wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.20/bin/apache-tomcat-10.1.20.tar.gz

sudo tar -xvzf apache-tomcat-10.1.20.tar.gz

sudo mv apache-tomcat-10.1.20 tomcat

sudo chmod +x /opt/tomcat/bin/\*.sh

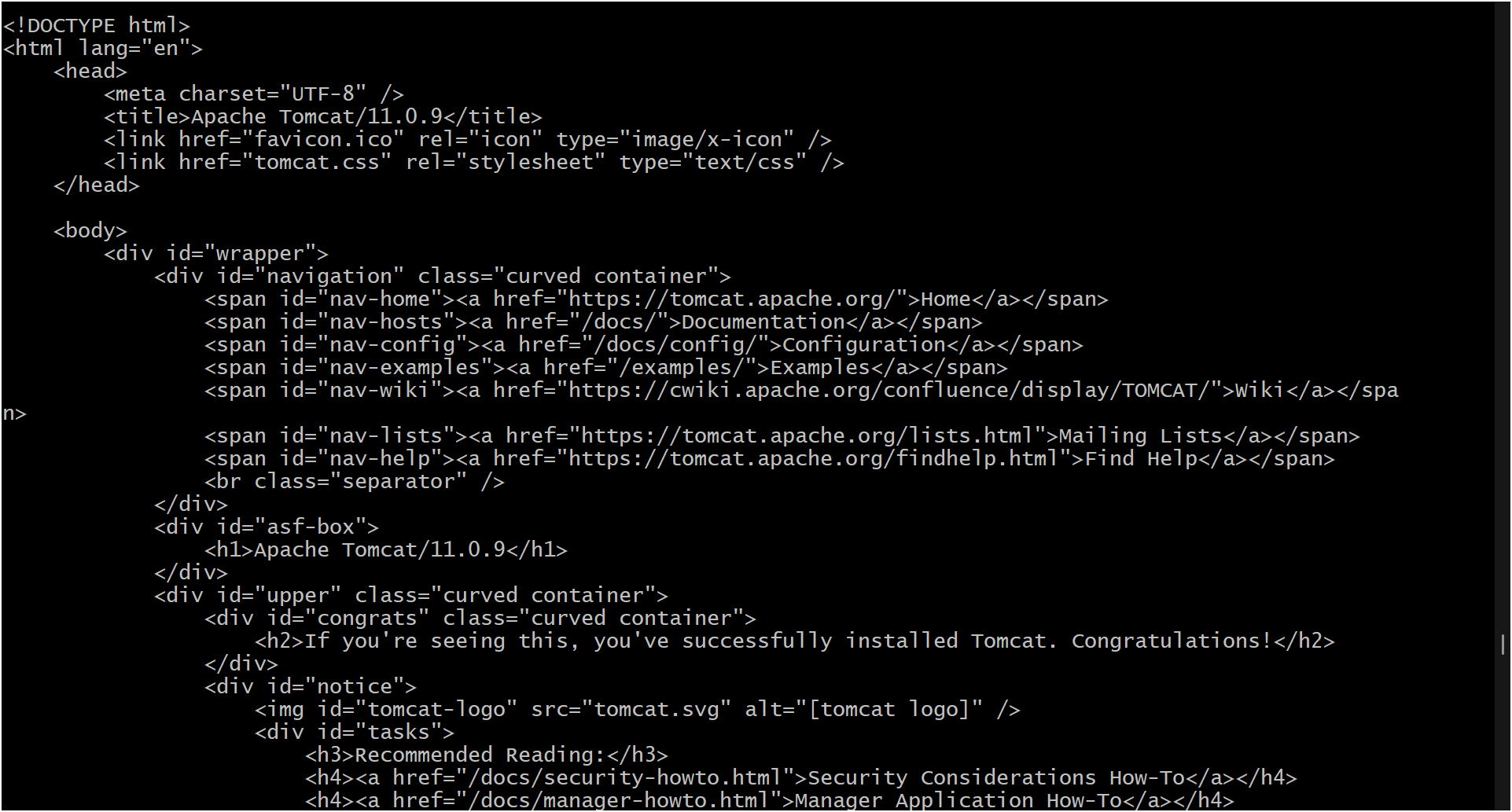
/opt/tomcat/bin/startup.sh



* Deploy Sample WAR:

cd /opt/tomcat/webapps

sudo wget https://tomcat.apache.org/tomcat-10.1-doc/appdev/sample/sample.war



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

* Go to **VPC → Flow Logs → Create Flow Log**:
  + Filter: All
  + Destination: **CloudWatch Logs** or **S3 Bucket**.
* If **S3**:
  + Select/create S3 bucket.
* If **CloudWatch**:
  + Create new Log Group and Role.
* Click **Create**.

