provider "aws" {

access\_key = "-"

secret\_key ="-"

region = "us-east-1" # You can change this to your desired region

}

# Create VPC

resource "aws\_vpc" "my\_vpc" {

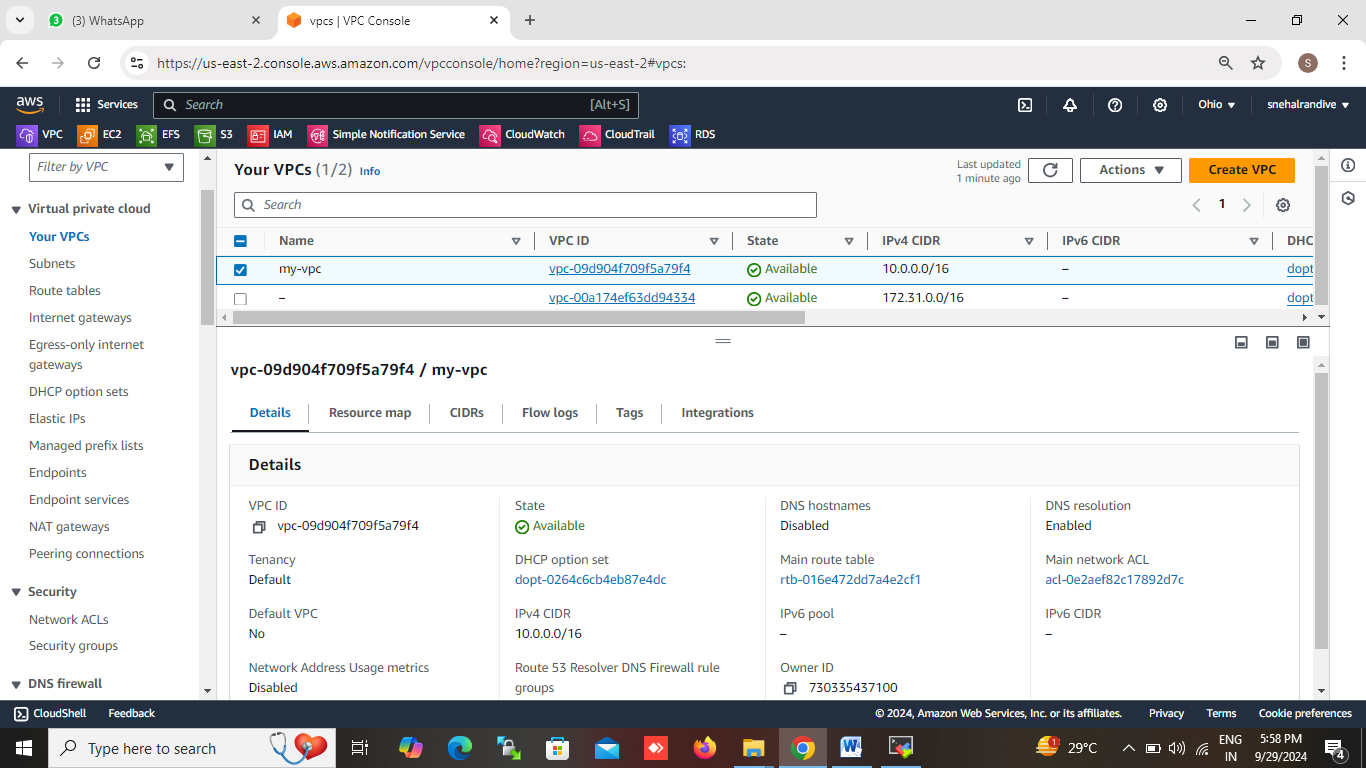
cidr\_block = "10.0.0.0/16"

tags = {

Name = "my-vpc"

}

}



# Create two subnets

resource "aws\_subnet" "subnet\_1" {

vpc\_id = aws\_vpc.my\_vpc.id

cidr\_block = "10.0.1.0/24"

availability\_zone = "us-east-1a"

tags = {

Name = "subnet-1"

}

}

resource "aws\_subnet" "subnet\_2" {

vpc\_id = aws\_vpc.my\_vpc.id

cidr\_block = "10.0.2.0/24"

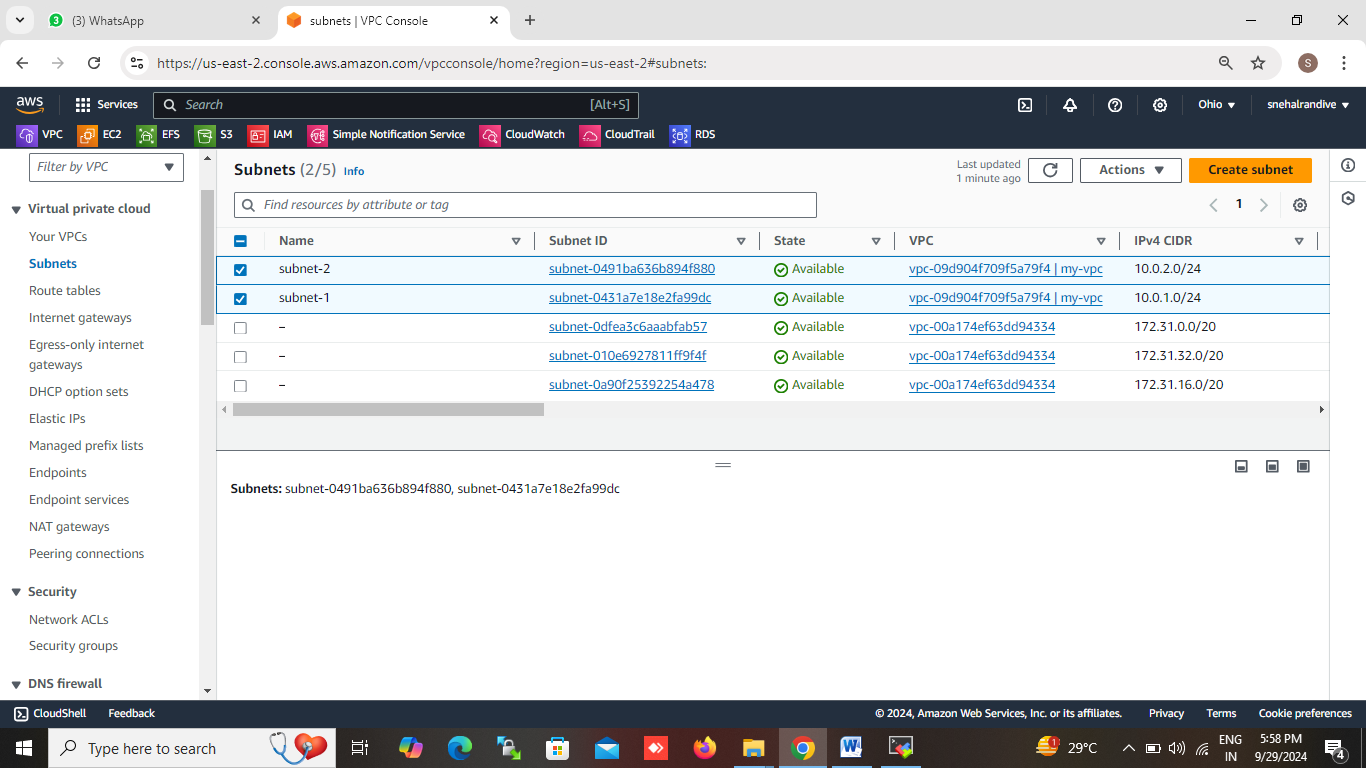
availability\_zone = "us-east-1b"

tags = {

Name = "subnet-2"

}

}



# Create Internet Gateway

resource "aws\_internet\_gateway" "igw" {

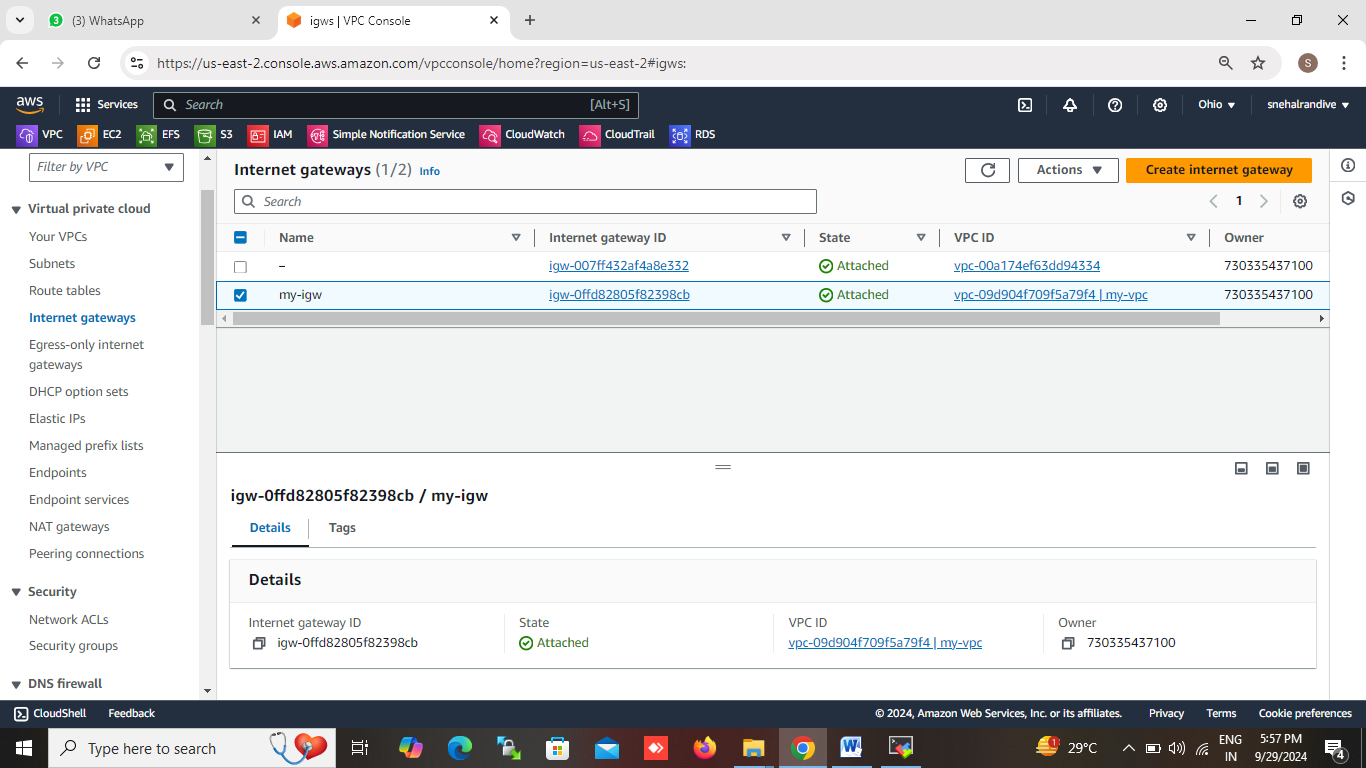
vpc\_id = aws\_vpc.my\_vpc.id

tags = {

Name = "my-igw"

}

}



# Create route table

resource "aws\_route\_table" "route\_table" {

vpc\_id = aws\_vpc.my\_vpc.id

route {

cidr\_block = "0.0.0.0/0"

gateway\_id = aws\_internet\_gateway.igw.id

}

tags = {

Name = "my-route-table"

}

}

# Associate the subnets with the route table

resource "aws\_route\_table\_association" "subnet\_1\_association" {

subnet\_id = aws\_subnet.subnet\_1.id

route\_table\_id = aws\_route\_table.route\_table.id

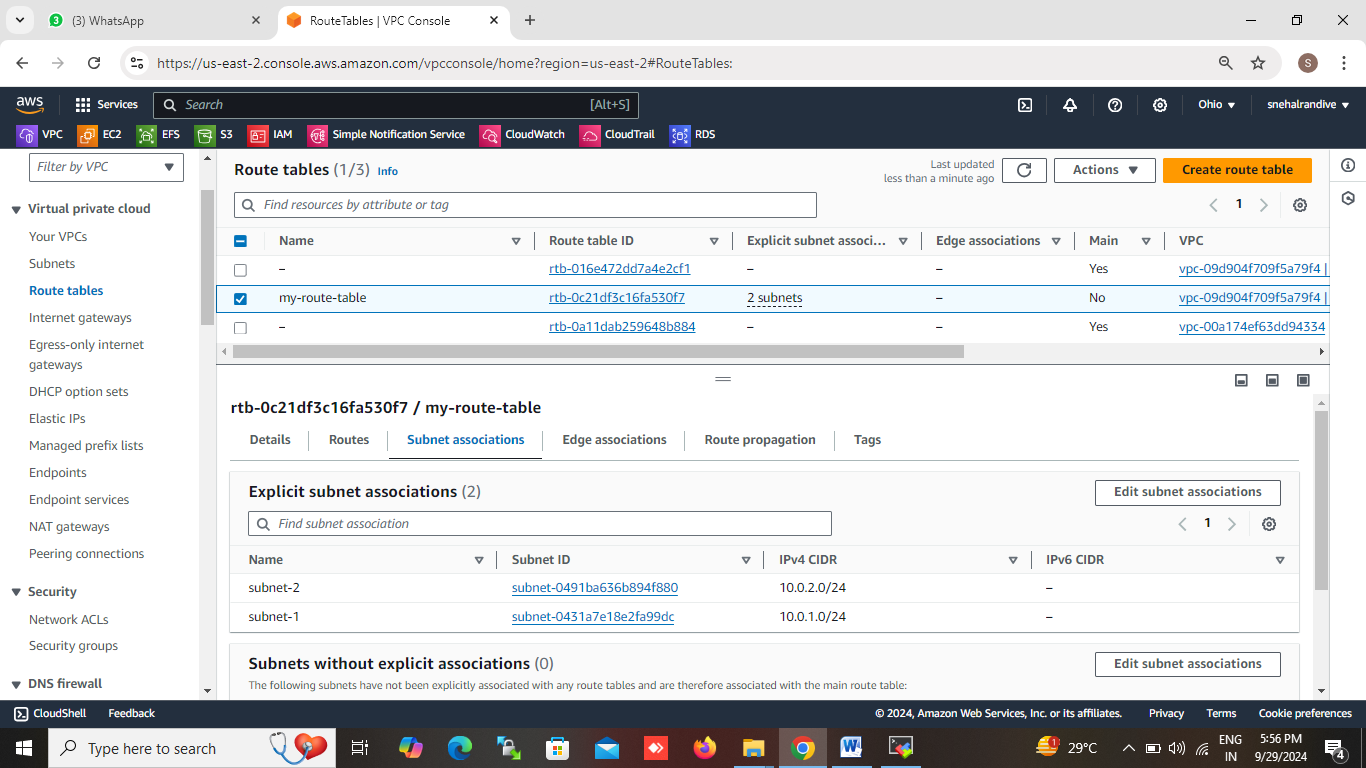
}

resource "aws\_route\_table\_association" "subnet\_2\_association" {

subnet\_id = aws\_subnet.subnet\_2.id

route\_table\_id = aws\_route\_table.route\_table.id

}



# Create a security group to allow HTTP and SSH

resource "aws\_security\_group" "allow\_http\_ssh" {

vpc\_id = aws\_vpc.my\_vpc.id

ingress {

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

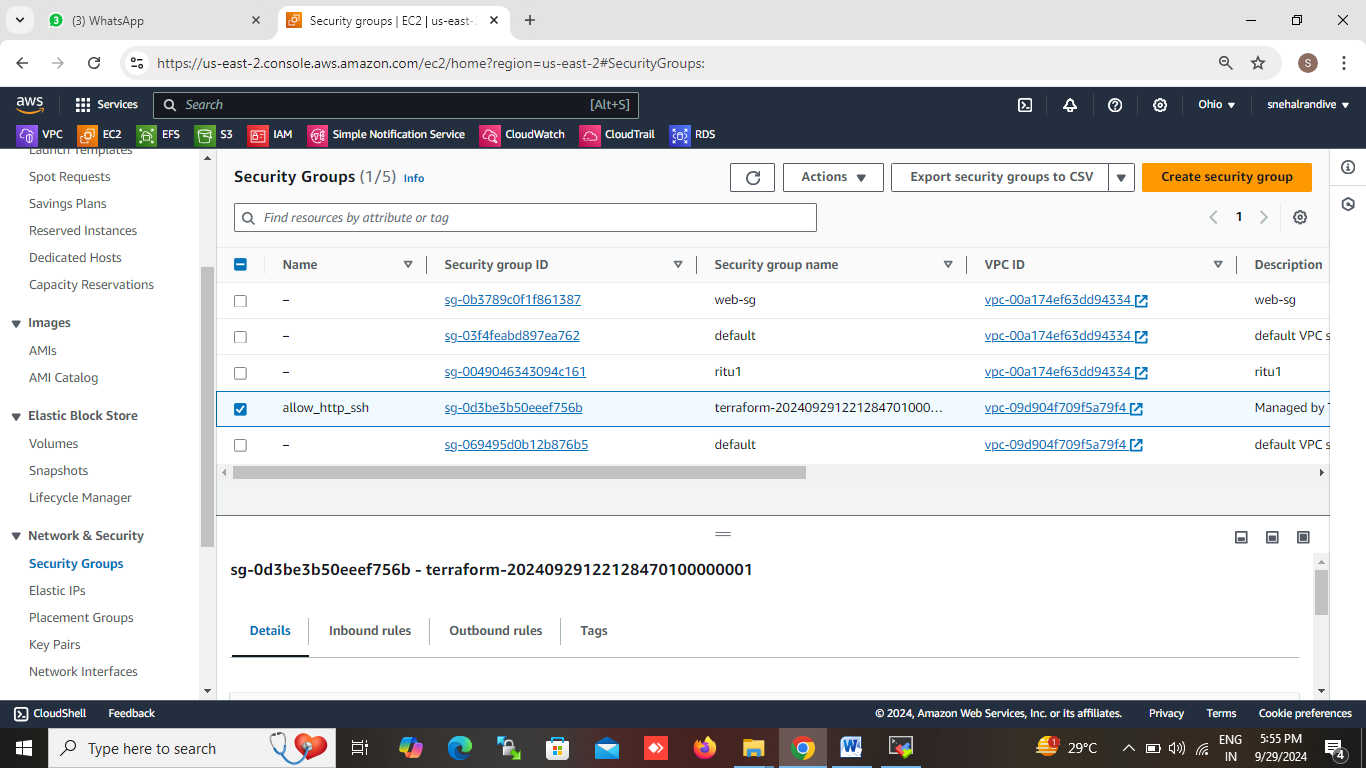
}

tags = {

Name = "allow-http-ssh"

}

}



# Launch EC2 instance

resource "aws\_instance" "web" {

ami = "ami-id”

instance\_type = "t2.micro"

subnet\_id = aws\_subnet.subnet\_1.id

security\_groups = [aws\_security\_group.allow\_http\_ssh.name]

associate\_public\_ip\_address = true ## show the public ip of the instance

user\_data = <<-EOF

#!/bin/bash

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Hello from Terraform!</h1>" > /var/www/html/index.html

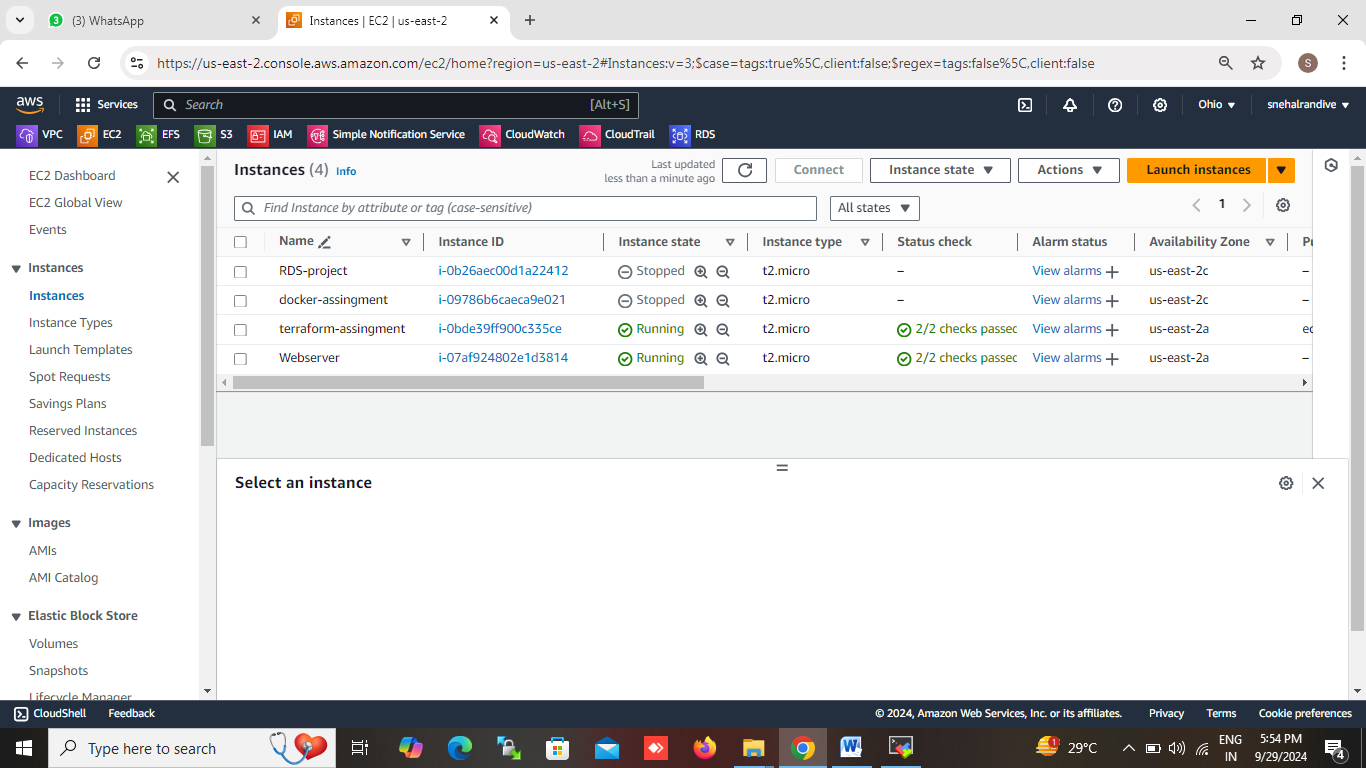
EOF

tags = {

Name = "WebServer"

}

}



# Output the public IP of the instance

output "instance\_public\_ip" {

value = aws\_instance.web.public\_ip

}

