**Terraform Assignment : 01**

**Q. Create a vpc, subnet, internet gateway, route table and then launch an ec2 instance inside a subnet then install apache httpd and access it through port no.80 with the help of terraform**

Step-1: Launch ec2 instance and install terraform

Step-2: Create a directory

* mkdir velocity
* cd velocity
* vi main.tf

provider "aws"{

region = "ap-south-1"

access\_key = ""

secret\_key = ""

}

###########Creating vpc#############

resource "aws\_vpc" "test"{

cidr\_block = "10.10.0.0/16"

tags = {

name = "vpc-a"

}

}

###########Creating subnet##########

resource "aws\_subnet" "subnet1"{

vpc\_id = "${aws\_vpc.test.id}"

cidr\_block = "10.10.1.0/24"

availability\_zone = "ap-south-1a"

tags = {

Name = "Public\_subnet"

}

}

resource "aws\_subnet" "subnet2"{

vpc\_id = "${aws\_vpc.test.id}"

cidr\_block = "10.10.2.0/24"

availability\_zone = "ap-south-1b"

tags = {

Name = "Private\_subnet"

}

}

###########Creating Igw##############

resource "aws\_internet\_gateway" "test\_igw"{

vpc\_id = "${aws\_vpc.test.id}"

tags = {

Name = "igw"

}

}

#########creating route table###########

resource "aws\_route\_table" "publicrt"{

vpc\_id = "${aws\_vpc.test.id}"

route{

cidr\_block = "0.0.0.0/0"

gateway\_id = "${aws\_internet\_gateway.test\_igw.id}"

}

tags = {

Name = "public rt"

}

}

#########route table association#########

resource "aws\_route\_table\_association" "public-1"{

route\_table\_id = "${aws\_route\_table.publicrt.id}"

subnet\_id = "${aws\_subnet.subnet1.id}"

}

resource "aws\_route\_table\_association" "public-2"{

route\_table\_id = "${aws\_route\_table.publicrt.id}"

subnet\_id = "${aws\_subnet.subnet2.id}"

}

#########Security Group###########

resource "aws\_security\_group" "sg" {

vpc\_id = "${aws\_vpc.test.id}"

ingress {

description = "SSH from VPC"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "HTTP from VPC"

from\_port = 80

to\_port = 80

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\_ssh\_http"

}

}

###########aws instance############

resource "aws\_instance" "web" {

ami = "ami-0d1e92463a5acf79d"

instance\_type = "t2.micro"

subnet\_id = "${aws\_subnet.subnet1.id}"

key\_name = "linux-kp"

security\_groups = [aws\_security\_group.sg.id]

user\_data = <<-EOF

#!/bin/bash

sudo yum install httpd -y

sudo systemctl httpd start

echo "<h1>Learning Terraform</h1>" /var/www/html/index.html

EOF

tags = {

Name = "web\_instance"

}

}

######Elastic-ip########

resource "aws\_eip" "my\_eip"{

instance = aws\_instance.web.id

}

* vi output.tf

output "public\_ip"

{

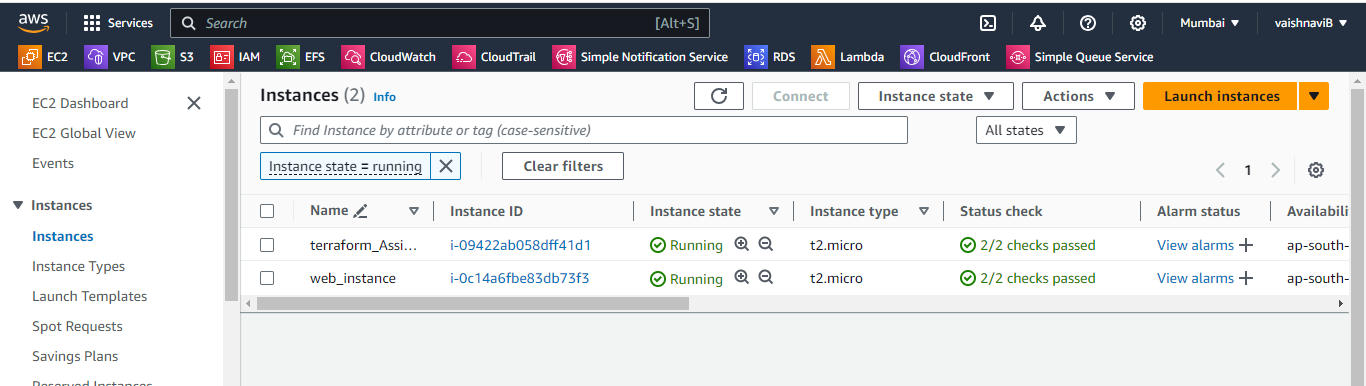
value = aws\_instance.web.public\_ip

}

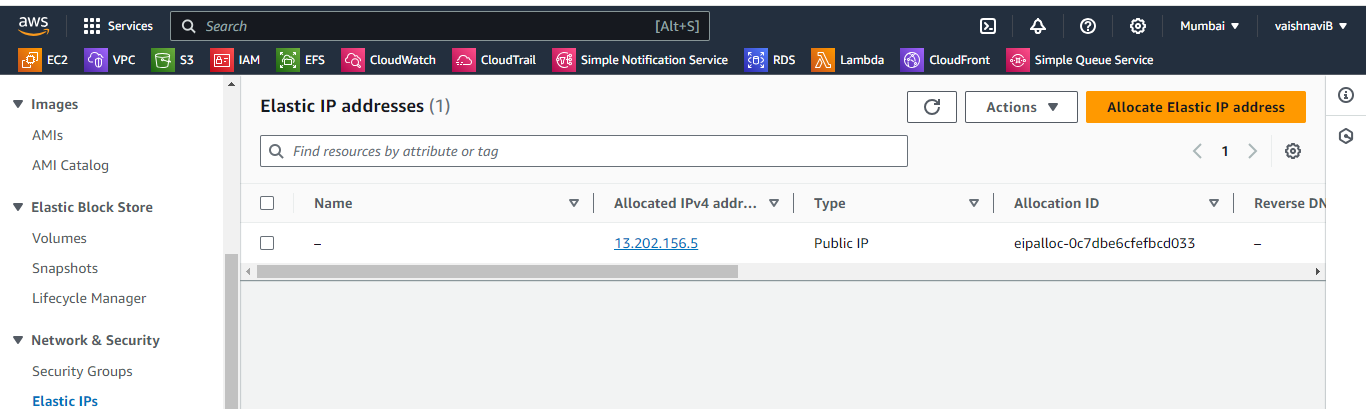
Step-3:

* terraform init
* ls -ltra
* terraform plan
* terraform apply –auto-approve

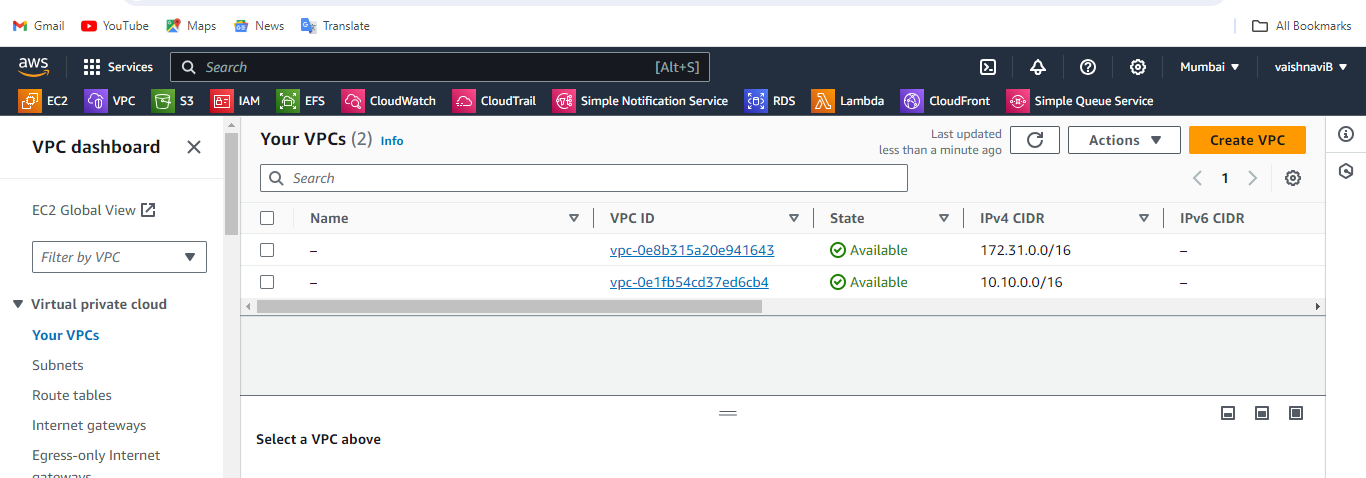
1. ec2 instance created inside a vpc



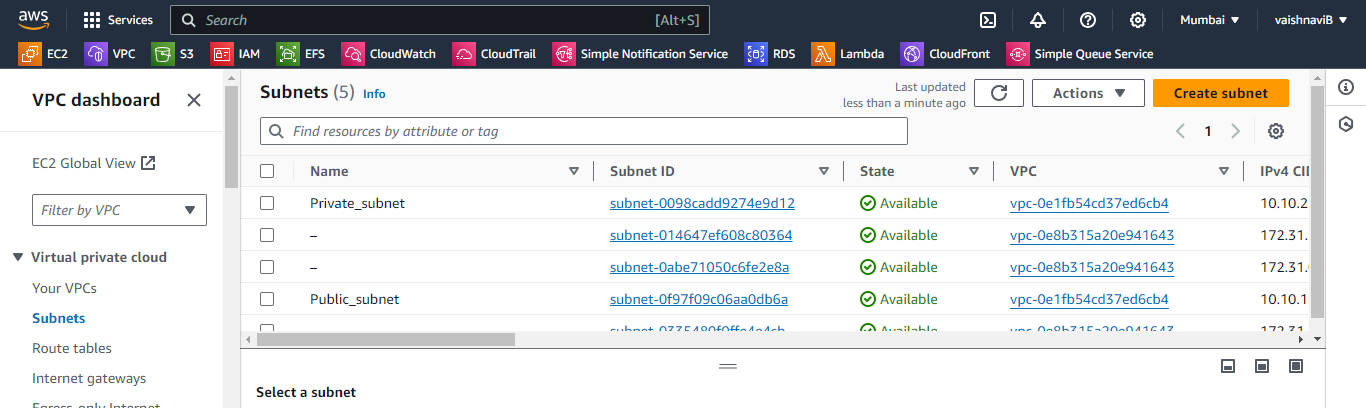
1. eip created



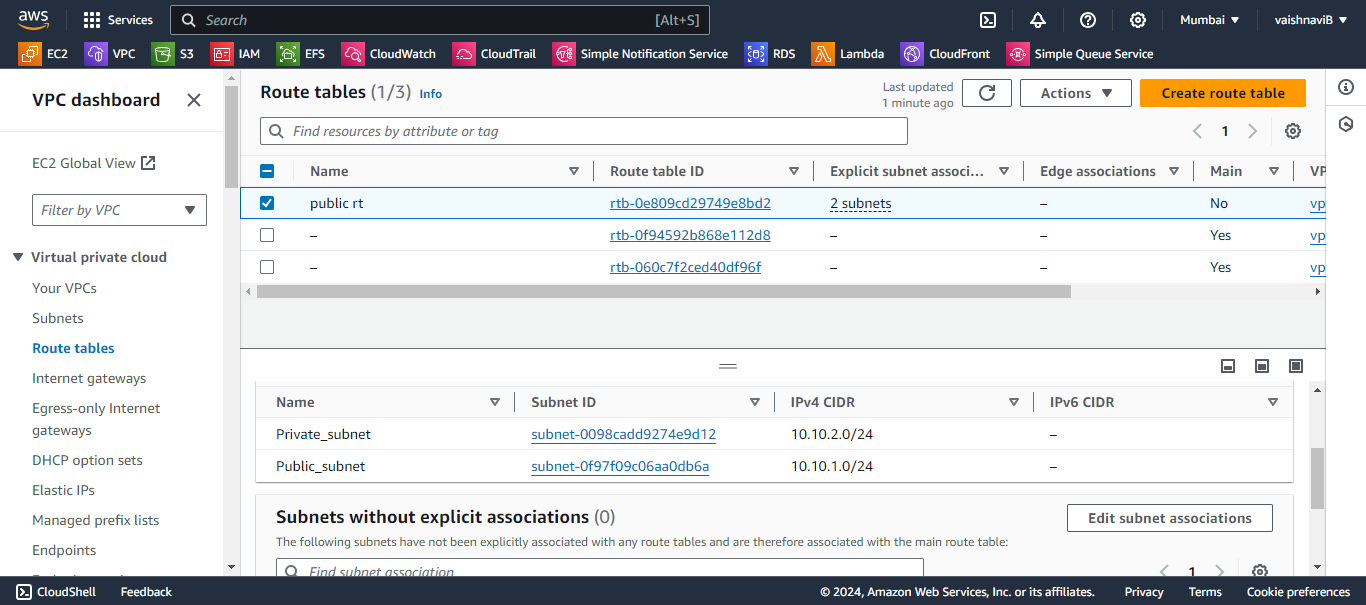
1. vpc created 10.10.0.0/16



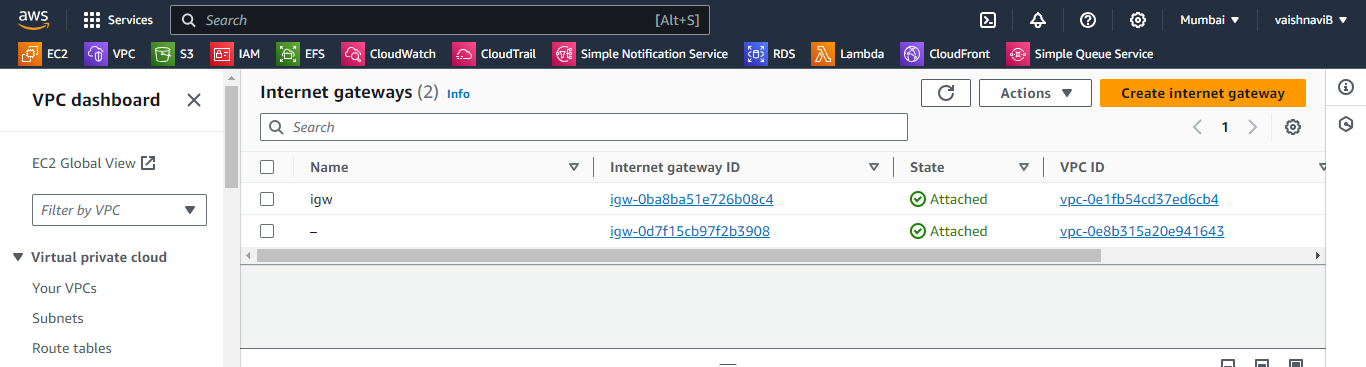
1. subnet created



1. Route table created and subnet association is also done



1. InternetGateway created



1. Apache httpd index file accessed

