ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat aws\_key\_pair.tf

# creating key on AWS

resource "aws\_key\_pair" "shyam\_aws\_key1" {

key\_name = "shyam\_aws\_key1"

public\_key = file("/home/ec2-user/.ssh/id\_rsa.pub")

}

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat main.tf

# creating instance on AWS

resource "aws\_instance" "s1" {

ami = var.image\_id

instance\_type = var.instance\_type

key\_name = aws\_key\_pair.shyam\_aws\_key1.key\_name

vpc\_security\_group\_ids = ["${aws\_security\_group.allow\_tls.id}"]

tags = {

Name = "Shyam-Server2"

env = "dev"

}

#user\_data = file("/home/ec2-user/environment/app1/scr.sh")

connection {

type = "ssh"

user = "ec2-user"

private\_key = file("/home/ec2-user/.ssh/id\_rsa")

host = self.public\_ip

}

provisioner "file" {

source = "scr.sh"

destination = "/tmp/scr.sh"

}

provisioner "file" {

content = "This is shyam file"

destination = "/tmp/shyam.txt"

}

provisioner "local-exec" {

command = "echo ${self.public\_ip} > /tmp/shyam\_ip.txt "

}

#resource user\_data = "${file("/home/ec2-user/environment/app1/scr.ksh")}"

# user\_data = << EOF

# #!/bin/ksh

# sudo apt-get update

# sudo apt-get install nginx

# sudo echo "Hi Shyam" >/var/www/html/index-nginx-debian.hrml

# EOF

}ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat network.tf

## Defining inbound/outbound rules for security group

resource "aws\_security\_group" "allow\_tls" {

name = "allow\_tls"

description = "Allow TLS inbound traffic"

#vpc\_id = aws\_vpc.main.id

dynamic "ingress" {

for\_each = var.ports

iterator = port

content {

description = "TLS from VPC"

from\_port = port.value

to\_port = port.value

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

ipv6\_cidr\_blocks = ["::/0"]

}

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

ipv6\_cidr\_blocks = ["::/0"]

}

tags = {

Name = "shyam\_sg"

}

}

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat output.tf

output "Server\_details" {

value = aws\_instance.s1.tags\_all

}

output "keyname" {

value = aws\_key\_pair.shyam\_aws\_key1.tags\_all

}

output "security\_group\_details" {

value = aws\_security\_group.allow\_tls.tags\_all

}

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat provider.tf

provider "aws" {

region = "us-east-1"

}

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat scr.sh

#!/bin/sh

#sudo apt-get update

#sudo apt-get install nginx

#sudo echo "Hi Shyam" >/var/www/html/index-nginx-debian.html

touch /tmp/shyam.txt

echo "Shyam Kumar Sharma" > /tmp/shyam.txtec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat security\_group.tf

# resource "aws\_security\_group" "allow\_tls" {

# name = "allow\_tls"

# description = "Allow TLS inbound traffic"

# #vpc\_id = aws\_vpc.main.id

# tags = {

# Name = "shyam\_sg"

# }

# }

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat terraform.tfvars

ports = [22,80,443,3306,27017]

image\_id = "ami-026b57f3c383c2eec"

instance\_type = "t2.micro"

ec2-user:~/environment/app1 $

ec2-user:~/environment/app1 $ cat variables.tf

variable ports {

type = list(number)

}

variable image\_id {

type = string

}

variable instance\_type {

type = string

}

ec2-user:~/environment/app1 $

--------------------------------------------------------------------------------------------------------------------------

ec2-user:~/environment/app1/poc1 $ cat network.tf

## creating VPC

resource "aws\_vpc" "shyamvpc" {

cidr\_block = "192.168.0.0/16"

instance\_tenancy = "default"

tags = {

Name = "shyam-vpc"

}

}

## Route table for Public Subnet ##

# resource "aws\_route\_table" "shyam\_public\_route\_table" {

# vpc\_id = aws\_vpc.shyamvpc.id

# route {

# cidr\_block = "0.0.0.0/0"

# gateway\_id = aws\_internet\_gateway.shyam\_internet\_gateway.id

# }

# tags = {

# Name = "shyam\_public\_route\_table"

# }

# }

# ### Association between Public Subnet and Public Route Table

# resource "aws\_route\_table\_association" "public" {

# subnet\_id = aws\_subnet.shyam\_subnet\_public\_1.id

# route\_table\_id = aws\_route\_table.shyam\_public\_route\_table.id

# }

ec2-user:~/environment/app1/poc1 $ cat provider.tf

provider "aws" {

region = "us-east-1"

}

ec2-user:~/environment/app1/poc1 $

ec2-user:~/environment/app1/poc1 $ cat subnet.tf

## Creating Public Subnet under above VPC

resource "aws\_subnet" "shyam\_subnet\_public\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.0.0/18"

map\_public\_ip\_on\_launch = true

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

tags = {

Name = "shyam-public\_subnet\_us\_east\_1a"

}

}

## Public Subnet-2

resource "aws\_subnet" "shyam\_subnet\_public\_2" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.64.0/18"

map\_public\_ip\_on\_launch = true

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

tags = {

Name = "shyam-public\_subnet\_us\_east\_1b"

}

}

## Private Subnet-1

resource "aws\_subnet" "shyam\_subnet\_private\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.128.0/18"

map\_public\_ip\_on\_launch = false

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

tags = {

Name = "shyam-private\_subnet\_us\_east\_1a"

}

}

## Private Subnet-2

resource "aws\_subnet" "shyam\_subnet\_private\_2" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.192.0/18"

map\_public\_ip\_on\_launch = false

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

tags = {

Name = "shyam-private\_subnet\_us\_east\_1b"

}

}

ec2-user:~/environment/app1/poc1 $

ec2-user:~/environment/app1/poc1 $ cat internet\_gateway.tf

## creating Internet Gateway ##

resource "aws\_internet\_gateway" "shyam\_internet\_gateway" {

vpc\_id = aws\_vpc.shyamvpc.id

tags = {

Name = "shyam\_internet\_gateway"

}

}

ec2-user:~/environment/app1/poc1 $

ec2-user:~/environment/app1/poc1 $ cat elastic\_ips.tf

resource "aws\_eip" "nat1" {

depends\_on = [aws\_internet\_gateway.shyam\_internet\_gateway]

}

resource "aws\_eip" "nat2" {

depends\_on = [aws\_internet\_gateway.shyam\_internet\_gateway]

}

ec2-user:~/environment/app1/poc1 $

ec2-user:~/environment/app1/poc1 $ cat nat\_gateway.tf

resource "aws\_nat\_gateway" "gw\_1" {

allocation\_id = aws\_eip.nat1.id

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

tags = {

Name = "NAT\_gw\_1"

}

# To ensure proper ordering, it is recommended to add an explicit dependency

# on the Internet Gateway for the VPC.

#depends\_on = [aws\_internet\_gateway.example]

}

## Nat gateway-2

resource "aws\_nat\_gateway" "gw\_2" {

allocation\_id = aws\_eip.nat2.id

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

tags = {

Name = "NAT\_gw\_2"

}

# To ensure proper ordering, it is recommended to add an explicit dependency

# on the Internet Gateway for the VPC.

#depends\_on = [aws\_internet\_gateway.example]

}

ec2-user:~/environment/app1/poc1 $

ec2-user:~/environment/app1/poc1 $

------------------------------------------------------------------------------------------------------------------------------------------

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat internet\_gateway.tf

## creating Internet Gateway ##

resource "aws\_internet\_gateway" "shyam\_internet\_gateway" {

vpc\_id = aws\_vpc.shyamvpc.id

tags = {

Name = "shyam\_internet\_gateway"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat nat\_gateway.tf

resource "aws\_nat\_gateway" "gw\_1" {

allocation\_id = aws\_eip.nat1.id

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

tags = {

Name = "NAT\_gw\_1"

}

# To ensure proper ordering, it is recommended to add an explicit dependency

# on the Internet Gateway for the VPC.

#depends\_on = [aws\_internet\_gateway.example]

}

## Nat gateway-2

resource "aws\_nat\_gateway" "gw\_2" {

allocation\_id = aws\_eip.nat2.id

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

tags = {

Name = "NAT\_gw\_2"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat network.tf

## creating VPC

resource "aws\_vpc" "shyamvpc" {

cidr\_block = "192.168.0.0/16"

instance\_tenancy = "default"

tags = {

Name = "shyam-vpc"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat network.tf

## creating VPC

resource "aws\_vpc" "shyamvpc" {

cidr\_block = "192.168.0.0/16"

instance\_tenancy = "default"

tags = {

Name = "shyam-vpc"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat provider.tf

provider "aws" {

region = "us-east-1"

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat subnet.tf

## Creating Public Subnet under above VPC

resource "aws\_subnet" "shyam\_subnet\_public\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.0.0/18"

map\_public\_ip\_on\_launch = true

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

tags = {

Name = "shyam-public\_subnet\_us\_east\_1a"

}

}

## Public Subnet-2

resource "aws\_subnet" "shyam\_subnet\_public\_2" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.64.0/18"

map\_public\_ip\_on\_launch = true

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

tags = {

Name = "shyam-public\_subnet\_us\_east\_1b"

}

}

## Private Subnet-1

resource "aws\_subnet" "shyam\_subnet\_private\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.128.0/18"

map\_public\_ip\_on\_launch = false

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

tags = {

Name = "shyam-private\_subnet\_us\_east\_1a"

}

}

## Private Subnet-2

resource "aws\_subnet" "shyam\_subnet\_private\_2" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.192.0/18"

map\_public\_ip\_on\_launch = false

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

tags = {

Name = "shyam-private\_subnet\_us\_east\_1b"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat route\_table

cat: route\_table: No such file or directory

ec2-user:~/environment/poc1 $ cat route\_table.tf

# Route table for Public Subnet ##

resource "aws\_route\_table" "shyam\_public\_route\_table" {

vpc\_id = aws\_vpc.shyamvpc.id

route {

cidr\_block = "0.0.0.0/0"

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

}

tags = {

Name = "shyam\_public\_route\_table"

}

}

# Route table for Private Subnet ##

resource "aws\_route\_table" "shyam\_private\_route\_table\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

route {

cidr\_block = "0.0.0.0/0"

#gateway\_id = aws\_internet\_gateway.shyam\_internet\_gateway.id

nat\_gateway\_id = aws\_nat\_gateway.gw\_1.id

}

tags = {

Name = "shyam\_private\_route\_table\_1"

}

}

# Route table-2 for Private Subnet ##

resource "aws\_route\_table" "shyam\_private\_route\_table\_2" {

vpc\_id = aws\_vpc.shyamvpc.id

route {

cidr\_block = "0.0.0.0/0"

#gateway\_id = aws\_internet\_gateway.shyam\_internet\_gateway.id

nat\_gateway\_id = aws\_nat\_gateway.gw\_2.id

}

tags = {

Name = "shyam\_private\_route\_table\_2"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat route\_table\_association.tf

### Association between Public Subnet and Public Route Table

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

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

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

}

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

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

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

}

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

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

route\_table\_id = aws\_route\_table.shyam\_private\_route\_table\_1.id

}

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

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

route\_table\_id = aws\_route\_table.shyam\_private\_route\_table\_2.id

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $

-----------------------------------------------------

ec2-user:~/environment/poc1 $ cat ec2.tf

## Generating Key

# creating key on AWS

resource "aws\_key\_pair" "shyam\_aws\_key1" {

key\_name = "shyam\_aws\_key1"

public\_key = file("/home/ec2-user/.ssh/id\_rsa.pub")

}

## Creating EC2 in Public Subnet

resource "aws\_instance" "public\_instance\_1" {

ami = "ami-026b57f3c383c2eec"

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.shyam\_aws\_key1.key\_name

vpc\_security\_group\_ids = ["${aws\_security\_group.security\_group1.id}"]

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

count = 1

associate\_public\_ip\_address = true

tags = {

Name = "shyam\_public\_instance\_1"

env = "test"

}

}

## Creating EC2 in Private Subnet

resource "aws\_instance" "private\_instance\_1" {

ami = "ami-026b57f3c383c2eec"

instance\_type = "t2.micro"

key\_name = aws\_key\_pair.shyam\_aws\_key1.key\_name

vpc\_security\_group\_ids = ["${aws\_security\_group.security\_group1.id}"]

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

count = 1

associate\_public\_ip\_address = false

tags = {

Name = "shyam\_private\_instance\_1"

env = "test"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat network.tf

## creating VPC

resource "aws\_vpc" "shyamvpc" {

cidr\_block = "192.168.0.0/16"

instance\_tenancy = "default"

tags = {

Name = "shyam-vpc"

}

}

## creating security group ##

resource "aws\_security\_group" "security\_group1" {

name = "terraform\_ec2\_private\_sg"

description = "Allow limited inbound externale traffic"

vpc\_id = aws\_vpc.shyamvpc.id

dynamic "ingress" {

#for\_each = [22, 80, 443, 3306, 27017]

for\_each = [22, 8080, 443]

iterator = port

content {

description = "TLS from VPC"

from\_port = port.value

to\_port = port.value

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

ipv6\_cidr\_blocks = ["::/0"]

}

}

egress {

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

ipv6\_cidr\_blocks = ["::/0"]

}

tags = {

Name = "shyam\_private\_sg"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat internet\_gateway.tf

## creating Internet Gateway ##

resource "aws\_internet\_gateway" "shyam\_internet\_gateway" {

vpc\_id = aws\_vpc.shyamvpc.id

tags = {

Name = "shyam\_internet\_gateway"

}

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat nat\_gateway.tf

## Elastic IP for NAR Gateway

resource "aws\_eip" "nat\_eip" {

vpc = true

depends\_on = [aws\_internet\_gateway.shyam\_internet\_gateway]

tags = {

Name = "NAT Gateway EIP"

}

}

## NAT Gateway for VPC ##

resource "aws\_nat\_gateway" "NAT\_gw\_1" {

allocation\_id = aws\_eip.nat\_eip.id

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

tags = {

Name = "NAT\_gw\_1"

}

# To ensure proper ordering, it is recommended to add an explicit dependency

# on the Internet Gateway for the VPC.

#depends\_on = [aws\_internet\_gateway.example]

}

## Nat gateway-2

# resource "aws\_nat\_gateway" "gw\_2" {

# allocation\_id = aws\_eip.nat2.id

# subnet\_id = aws\_subnet.shyam\_subnet\_public\_2.id

# tags = {

# Name = "NAT\_gw\_2"

# }

# }

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat provider.tf

provider "aws" {

region = "us-east-1"

}

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat route\_table

cat: route\_table: No such file or directory

ec2-user:~/environment/poc1 $ cat route\_table.tf

# Route table for Public Subnet ##

resource "aws\_route\_table" "shyam\_public\_route\_table" {

vpc\_id = aws\_vpc.shyamvpc.id

route {

cidr\_block = "0.0.0.0/0"

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

}

tags = {

Name = "shyam\_public\_route\_table"

}

}

# Route table for Private Subnet ##

resource "aws\_route\_table" "shyam\_private\_route\_table\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

route {

cidr\_block = "0.0.0.0/0"

#gateway\_id = aws\_internet\_gateway.shyam\_internet\_gateway.id

nat\_gateway\_id = aws\_nat\_gateway.NAT\_gw\_1.id

}

tags = {

Name = "shyam\_private\_route\_table\_1"

}

}

# # Route table-2 for Private Subnet ##

# resource "aws\_route\_table" "shyam\_private\_route\_table\_2" {

# vpc\_id = aws\_vpc.shyamvpc.id

# route {

# cidr\_block = "0.0.0.0/0"

# #gateway\_id = aws\_internet\_gateway.shyam\_internet\_gateway.id

# nat\_gateway\_id = aws\_nat\_gateway.gw\_2.id

# }

# tags = {

# Name = "shyam\_private\_route\_table\_2"

# }

# }

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat route\_table\_association.tf

### Association between Public Subnet and Public Route Table

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

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

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

}

# resource "aws\_route\_table\_association" "public\_2" {

# subnet\_id = aws\_subnet.shyam\_subnet\_public\_2.id

# route\_table\_id = aws\_route\_table.shyam\_public\_route\_table.id

# }

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

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

route\_table\_id = aws\_route\_table.shyam\_private\_route\_table\_1.id

}

# resource "aws\_route\_table\_association" "private\_2" {

# subnet\_id = aws\_subnet.shyam\_subnet\_private\_2.id

# route\_table\_id = aws\_route\_table.shyam\_private\_route\_table\_2.id

# }

ec2-user:~/environment/poc1 $

ec2-user:~/environment/poc1 $ cat subnet.tf

## Creating Public Subnet under above VPC

resource "aws\_subnet" "shyam\_subnet\_public\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.0.0/18"

map\_public\_ip\_on\_launch = true

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

tags = {

Name = "shyam-public\_subnet\_us\_east\_1a"

}

}

# ## Public Subnet-2

# resource "aws\_subnet" "shyam\_subnet\_public\_2" {

# vpc\_id = aws\_vpc.shyamvpc.id

# cidr\_block = "192.168.64.0/18"

# map\_public\_ip\_on\_launch = true

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

# tags = {

# Name = "shyam-public\_subnet\_us\_east\_1b"

# }

# }

## Private Subnet-1

resource "aws\_subnet" "shyam\_subnet\_private\_1" {

vpc\_id = aws\_vpc.shyamvpc.id

cidr\_block = "192.168.128.0/18"

map\_public\_ip\_on\_launch = false

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

tags = {

Name = "shyam-private\_subnet\_us\_east\_1b"

}

}

# ## Private Subnet-2

# resource "aws\_subnet" "shyam\_subnet\_private\_2" {

# vpc\_id = aws\_vpc.shyamvpc.id

# cidr\_block = "192.168.192.0/18"

# map\_public\_ip\_on\_launch = false

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

# tags = {

# Name = "shyam-private\_subnet\_us\_east\_1b"

# }

# }

ec2-user:~/environment/poc1 $