Terraform_Task_2

Use variables for all arguments, output public_ip , private_ip , screenshot of the public ip from the browser, private ip logs also use count or for each for creating subnets

The Solution

First The Steps Should Be Like This

1 - Create VPC and use Sprints as the name and 10.0.0.0/16 as the CIDR block then click create vpc.

2 - use for each to create 2 subnet on this vpc (1 public subnet with "Public_Sub_Sprints" as name and "10.0.0.0/24" as the CIDR block) (1 private subnet with r "Private_Sub_Sprints" as the name, and "10.0.1.0/24" as the CIDR block)

3 - Create Internet Gateway and use "Sprints-IGW" as the name and attach it for this vpc

4 - Create NAT Gateway And use“Sprints-NAT “ as the name and Choose Sprints-IGW as the gateway with Public_Sub_Sprints as the subnet\

5 - Create Route Table And use"Public-RT" as the name, select Sprints VPC and destination 0.0.0.0./0 and target Sprints-IGW as the gateway

6 - Create Route Table And use"Private-RT" as the name, select Sprints VPC and destination 0.0.0.0./0 and target Sprints-NAT gateway

7 - Create 1 Security Group with “Security_Sprints_Public” as name and allow HTTP and HTTPS and ssh traffic

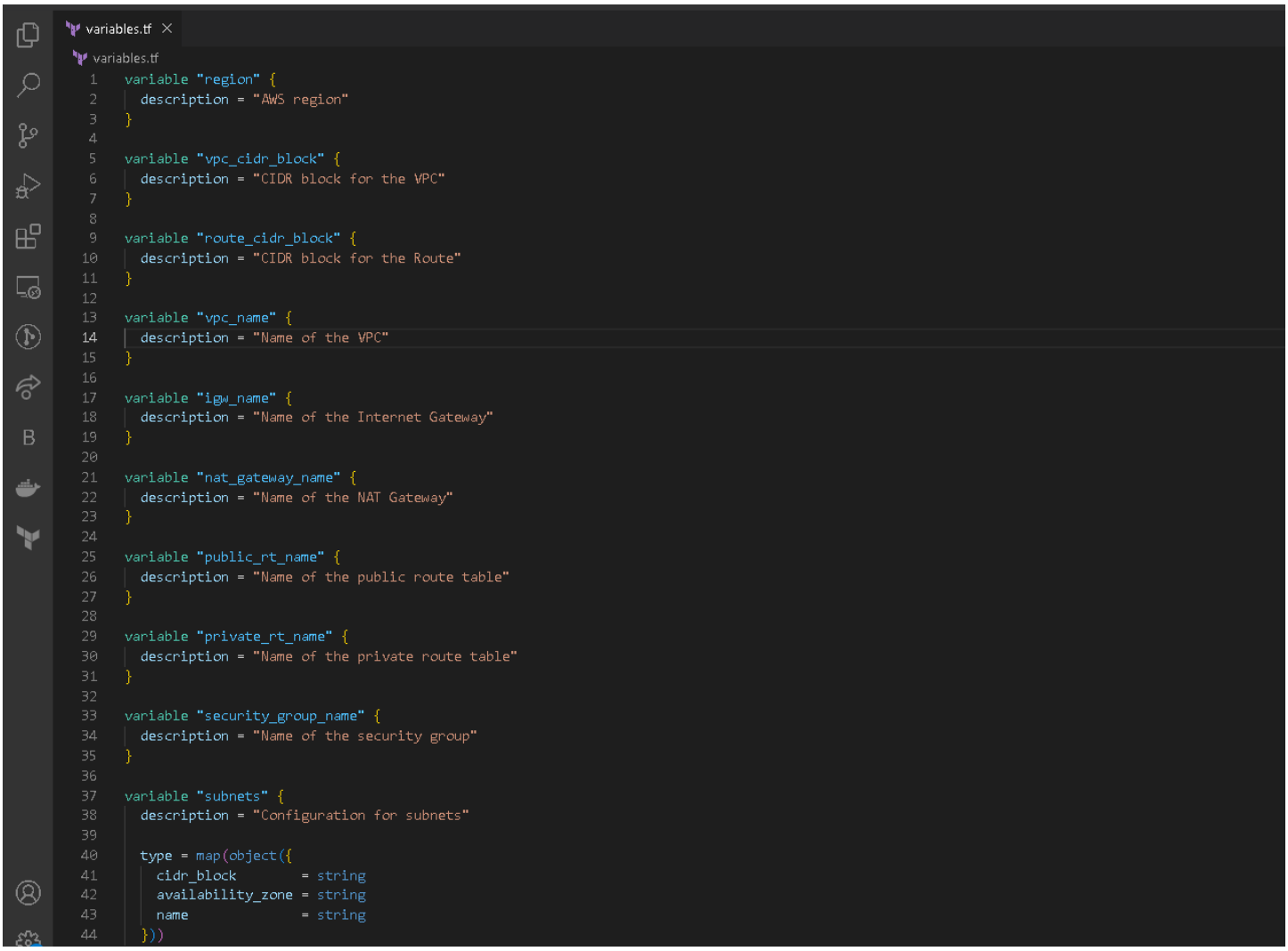
8 - Search for the Ubuntu AMI ID.

9 - use for each for Create 2 EC2 instance (1 instance for public subnet and install nginx on it with enable public ip) and (1 instance for private subnet and install apache on it with disable public ip)

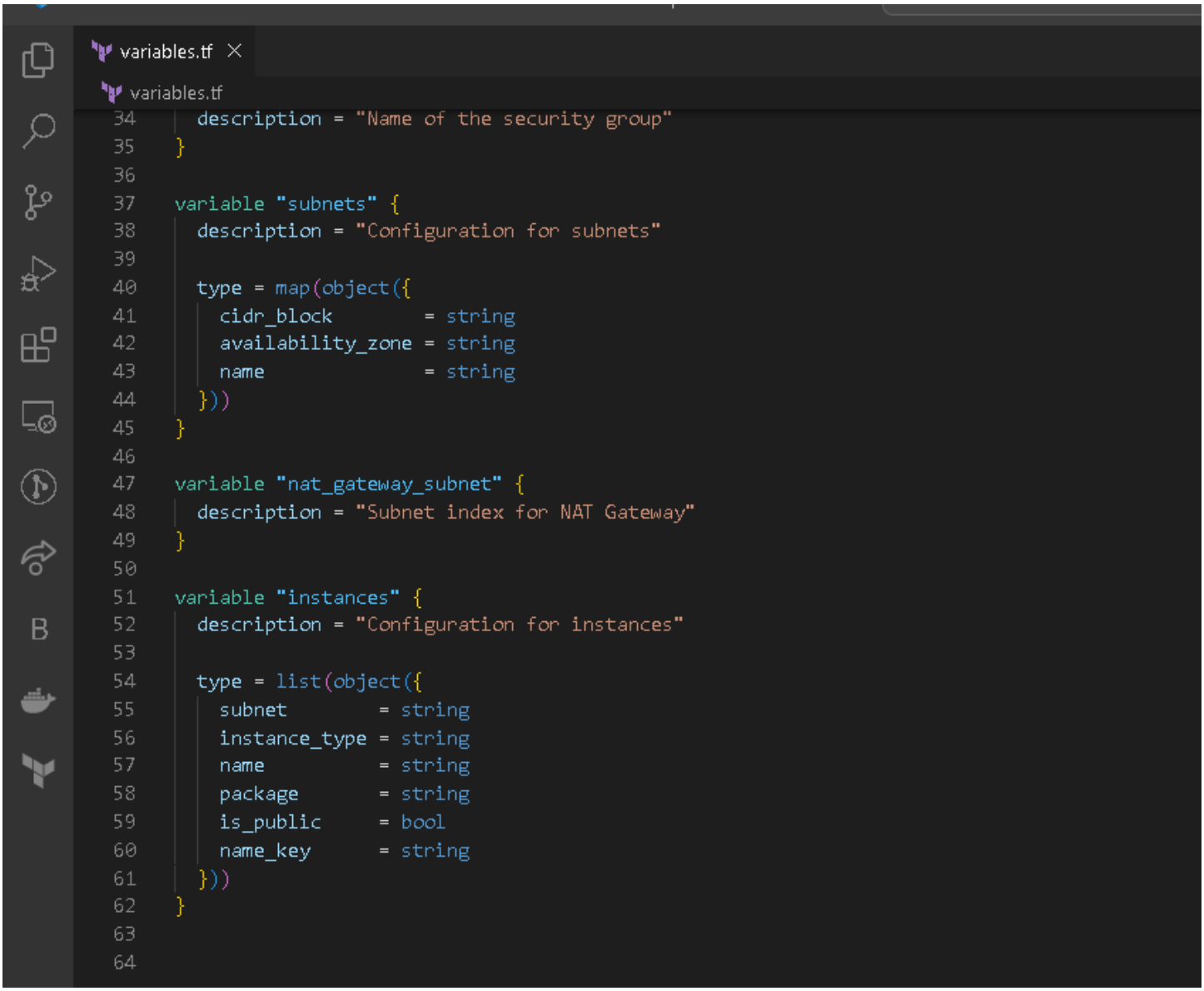
10 - generate output logs of both instance

The Final Code is:

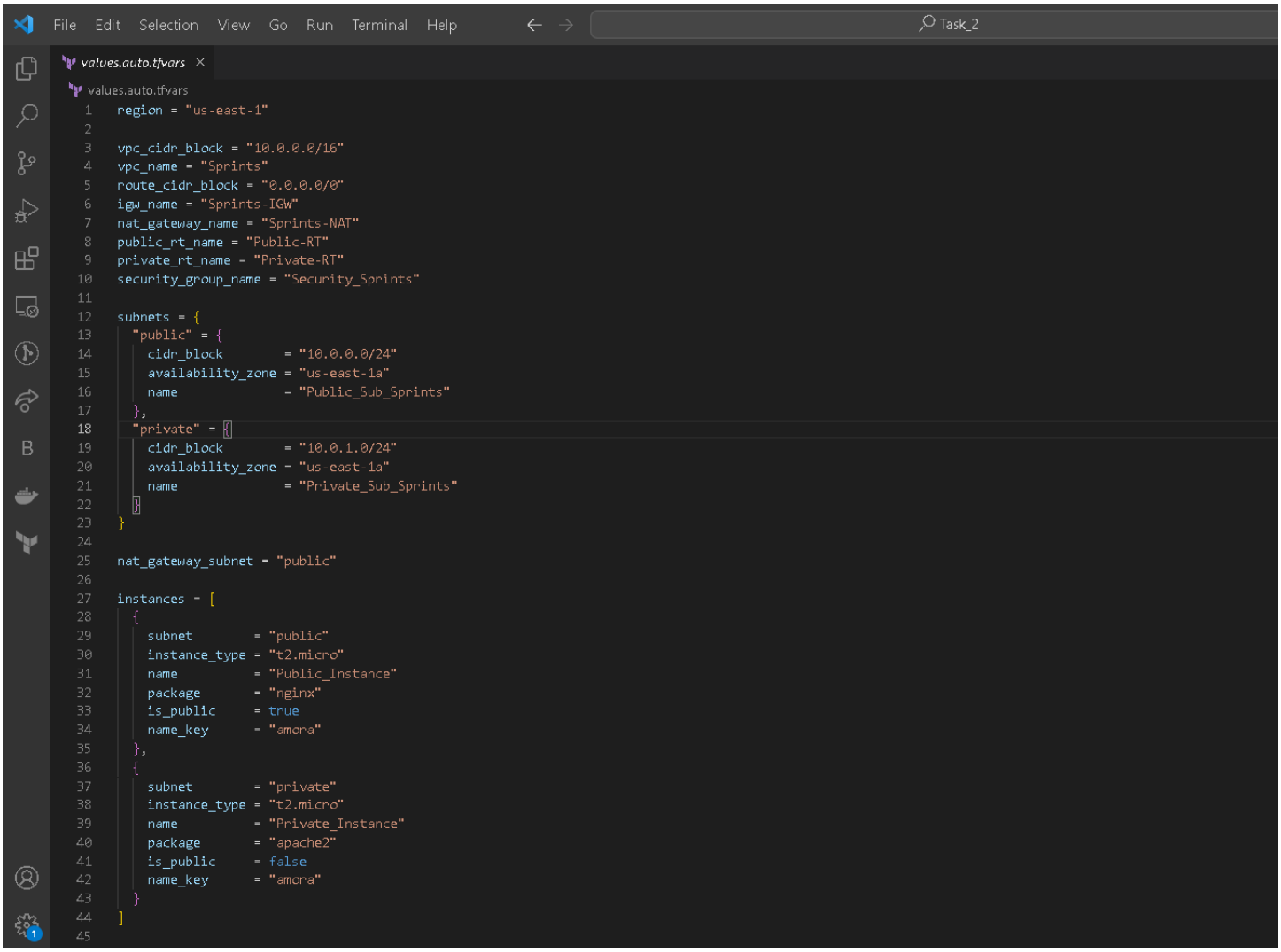
Variables.tf



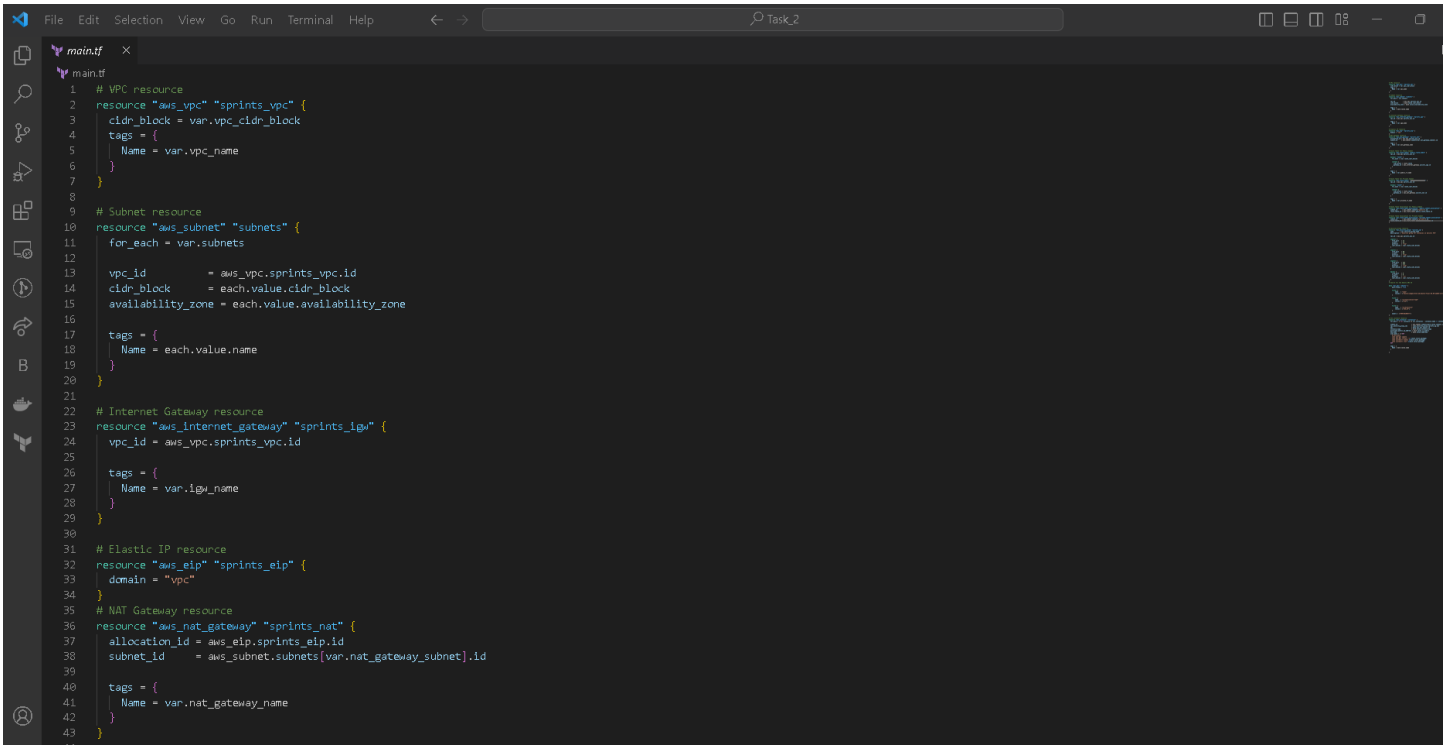
```hcl
variable "region" {
  description = "AWS region"
}

variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
}

variable "route_cidr_block" {
  description = "CIDR block for the Route"
}

variable "vpc_name" {
  description = "Name of the VPC"
}

variable "igw_name" {
  description = "Name of the Internet Gateway"
}

variable "nat_gateway_name" {
  description = "Name of the NAT Gateway"
}

variable "public_rt_name" {
  description = "Name of the public route table"
}

variable "private_rt_name" {
  description = "Name of the private route table"
}

variable "security_group_name" {
  description = "Name of the security group"
}

variable "subnets" {
  description = "Configuration for subnets"

  type = map(object({
    cidr_block        = string
    availability_zone = string
    name              = string
  }))
```

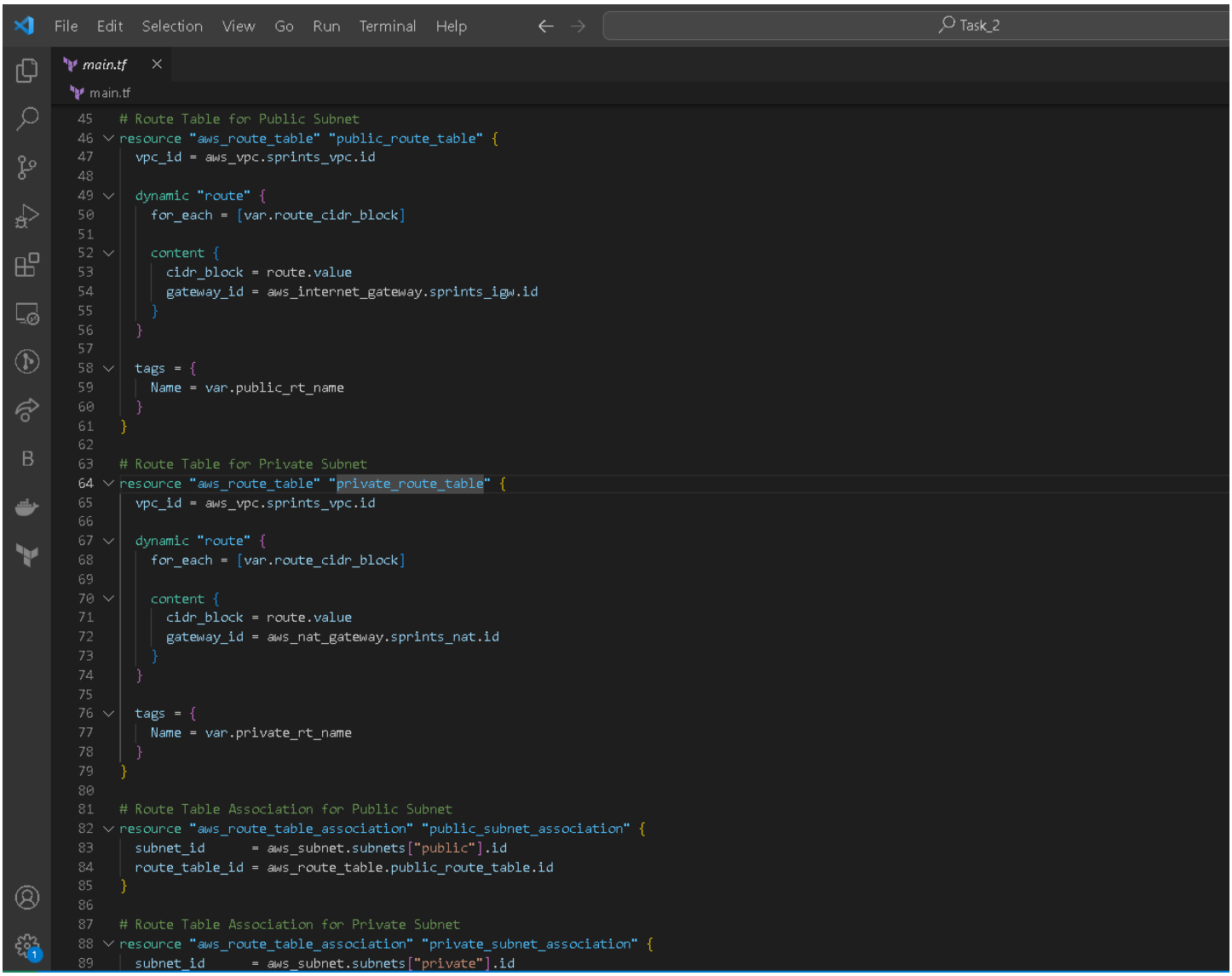

variables.tf
variables.tf
34 description = "Name of the security group"
35 }
36
37 variable "subnets" {
38 description = "Configuration for subnets"
39
40 type = map(object({
41 cidr_block = string
42 availability_zone = string
43 name = string
44 }))
45 }
46
47 variable "nat_gateway_subnet" {
48 description = "Subnet index for NAT Gateway"
49 }
50
51 variable "instances" {
52 description = "Configuration for instances"
53
54 type = list(object({
55 subnet = string
56 instance_type = string
57 name = string
58 package = string
59 is_public = bool
60 name_key = string
61 }))
62 }
63
64

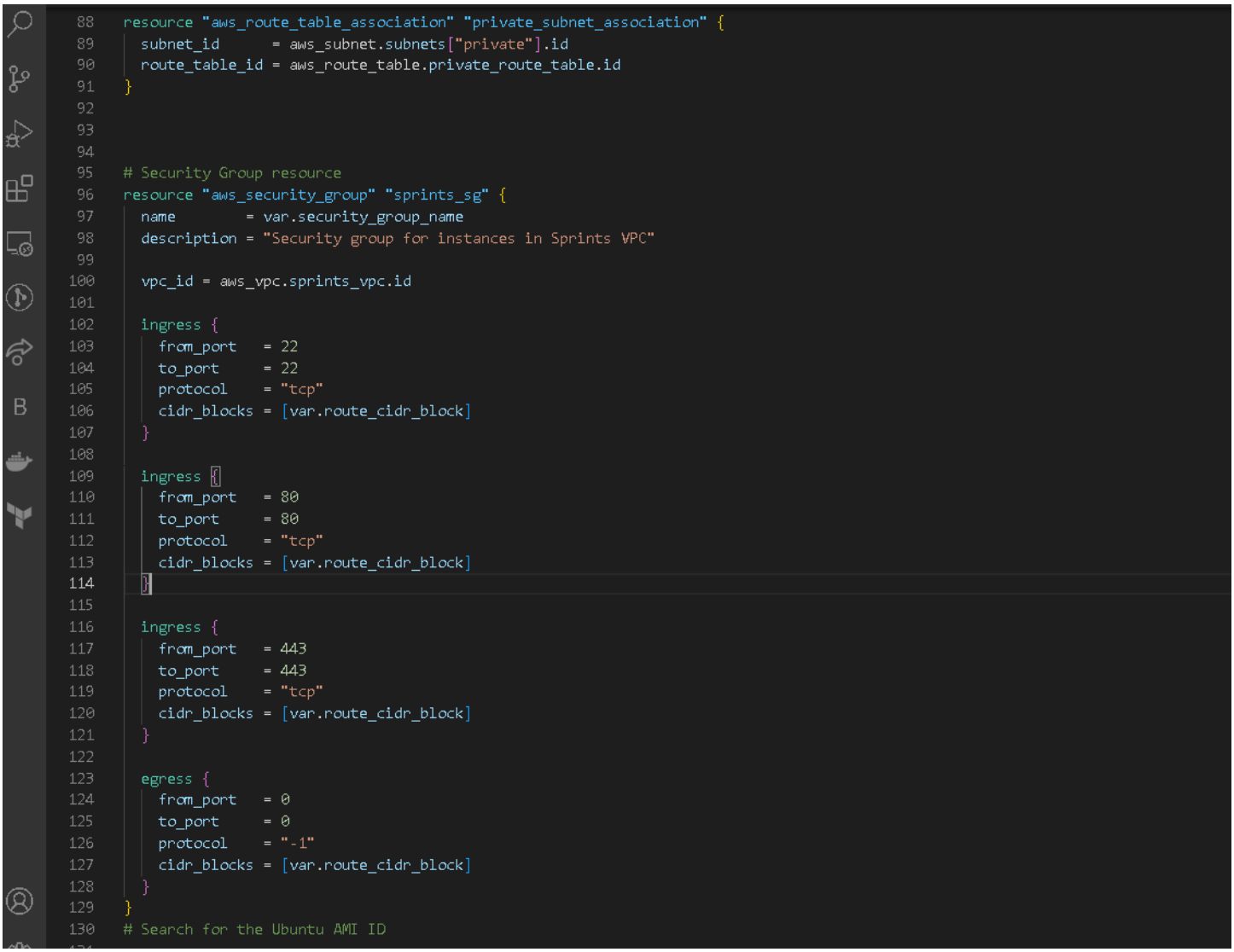
values.auto.tfvars



Main.tf


File Edit Selection View Go Run Terminal Help
Task_2
main.tf
main.tf
# VPC resource
resource "aws_vpc" "sprints_vpc" {
cidr_block = var.vpc_cidr_block
tags = {
Name = var.vpc_name
}
}
# Subnet resource
resource "aws_subnet" "subnets" {
for_each = var.subnets
vpc_id = aws_vpc.sprints_vpc.id
cidr_block = each.value.cidr_block
availability_zone = each.value.availability_zone
tags = {
Name = each.value.name
}
}
# Internet Gateway resource
resource "aws_internet_gateway" "sprints_igw" {
vpc_id = aws_vpc.sprints_vpc.id
tags = {
Name = var.igw_name
}
}
# Elastic IP resource
resource "aws_eip" "sprints_eip" {
domain = "vpc"
}
# NAT Gateway resource
resource "aws_nat_gateway" "sprints_nat" {
allocation_id = aws_eip.sprints_eip.id
subnet_id = aws_subnet.subnets[var.nat_gateway_subnet].id
tags = {
Name = var.nat_gateway_name
}
}


File Edit Selection View Go Run Terminal Help
Task_2
main.tf
main.tf
# Route Table for Public Subnet
resource "aws_route_table" "public_route_table" {
vpc_id = aws_vpc.sprints_vpc.id
dynamic "route" {
for_each = [var.route_cidr_block]
content {
cidr_block = route.value
gateway_id = aws_internet_gateway.sprints_igw.id
}
}
tags = {
Name = var.public_rt_name
}
}
# Route Table for Private Subnet
resource "aws_route_table" "private_route_table" {
vpc_id = aws_vpc.sprints_vpc.id
dynamic "route" {
for_each = [var.route_cidr_block]
content {
cidr_block = route.value
gateway_id = aws_nat_gateway.sprints_nat.id
}
}
tags = {
Name = var.private_rt_name
}
}
# Route Table Association for Public Subnet
resource "aws_route_table_association" "public_subnet_association" {
subnet_id = aws_subnet.subnets["public"].id
route_table_id = aws_route_table.public_route_table.id
}
# Route Table Association for Private Subnet
resource "aws_route_table_association" "private_subnet_association" {
subnet_id = aws_subnet.subnets["private"].id



```
resource "aws_route_table_association" "private_subnet_association" {
  subnet_id      = aws_subnet.subnets["private"].id
  route_table_id = aws_route_table.private_route_table.id
}



# Security Group resource
resource "aws_security_group" "sprints_sg" {
  name        = var.security_group_name
  description = "Security group for instances in Sprints VPC"

  vpc_id = aws_vpc.sprints_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.route_cidr_block]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = [var.route_cidr_block]
  }

  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = [var.route_cidr_block]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = [var.route_cidr_block]
  }
}
# Search for the Ubuntu AMI ID
```

main.tf

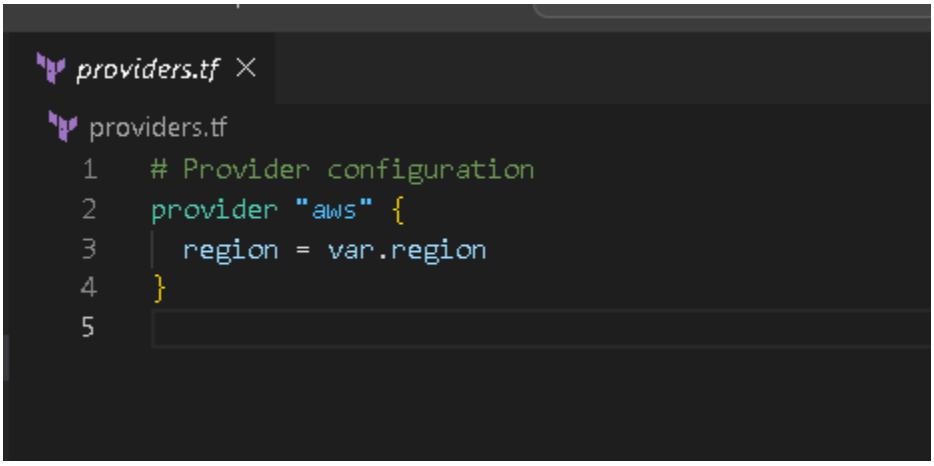
```
data "aws_ami" "ubuntu" {
    most_recent = true

    filter {
        name   = "name"
        values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
    }

    filter {
        name   = "virtualization-type"
        values = ["hvm"]
    }

    filter {
        name   = "architecture"
        values = ["x86_64"]
    }

    owners = ["099720109477"]
}

# EC2 instance resource
resource "aws_instance" "instances" {
  for_each = { for instance in var.instances : instance.name => instance }

  subnet_id                   = aws_subnet.subnets[each.value.subnet].id
  vpc_security_group_ids      = [aws_security_group.sprints_sg.id]
  ami                         = data.aws_ami.ubuntu.id
  instance_type               = each.value.instance_type
  associate_public_ip_address = each.value.is_public
  key_name                    = each.value.name_key
  user_data = <<-EOF
    #!/bin/bash
    sudo apt-get update
    sudo apt-get install -y ${each.value.package}
    sudo systemctl enable ${each.value.package}
    sudo systemctl start ${each.value.package}
  EOF

  tags = {
    Name = each.value.name
  }

}
```

Outputs.tf

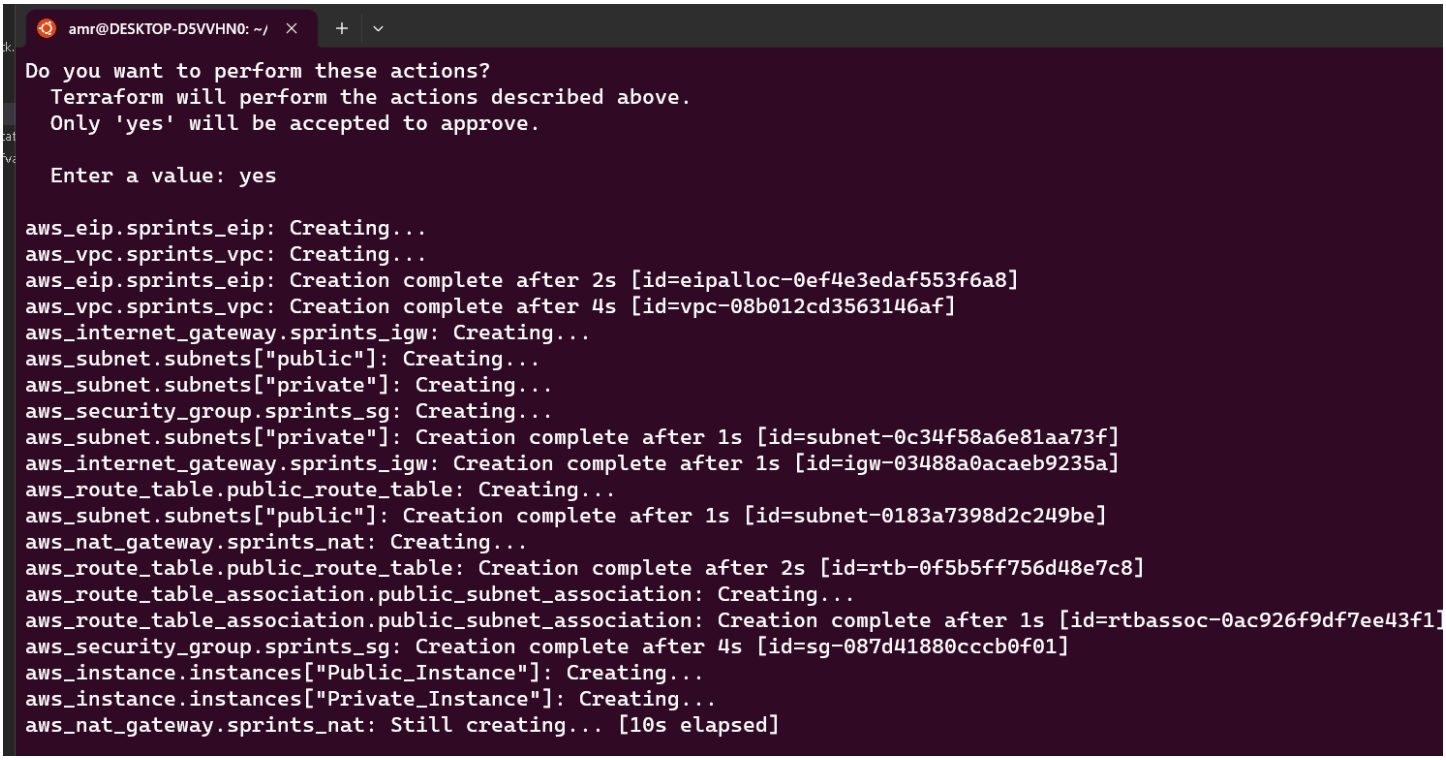
outputs.tf

```
output "public_ip" {
  value = [for instance in aws_instance.instances : instance.public_ip]
}

output "private_ip" {
  value = [for instance in aws_instance.instances : instance.private_ip]
}
```

Provider.tf

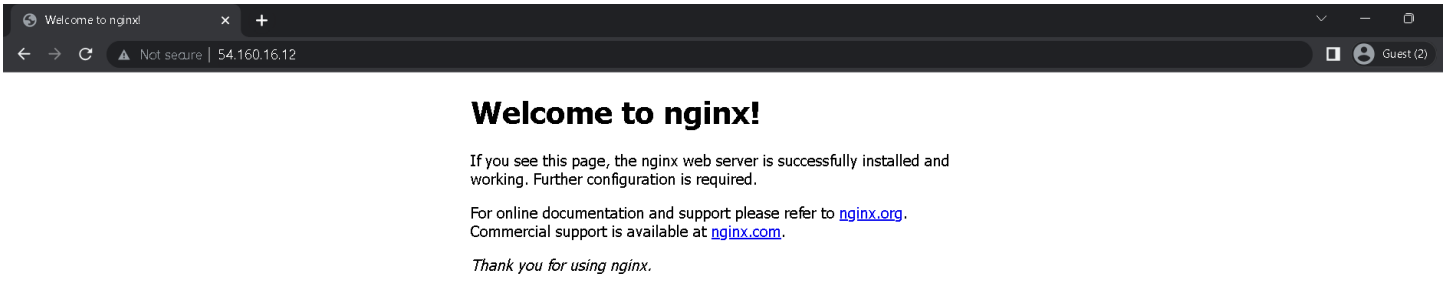


The final result in terminal



```
amr@DESKTOP-D5VVHN0: ~/
aws_nat_gateway.sprints_nat: Still creating... [40s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [50s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m0s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m10s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m20s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m30s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m40s elapsed]
aws_nat_gateway.sprints_nat: Creation complete after 1m46s [id=nat-06b8c7498d9e027c6]
aws_route_table.private_route_table: Creating...
aws_route_table.private_route_table: Creation complete after 3s [id=rtb-0b7e97ebb4cb02292]
aws_route_table_association.private_subnet_association: Creating...
aws_route_table_association.private_subnet_association: Creation complete after 1s [id=rtbassoc-03411c08fc5c1cbc0]

Apply complete! Resources: 13 added, 0 changed, 0 destroyed.

Outputs:

private_ip = [
  "10.0.1.64",
  "10.0.0.187",
]
public_ip = [
  "",
  "54.160.16.12",
]
amr@DESKTOP-D5VVHN0:~/terraform_tasks/Task_2$
```

The Public Ip Results In Browser



The Private Ip from logs