Terraform_Task_1

1 - Define the provider and resource blocks in the providers.tf

2 – Create aws_resources.tf and create vpc with range 10.0.0.0/16 and internet gateway attached to this vpc

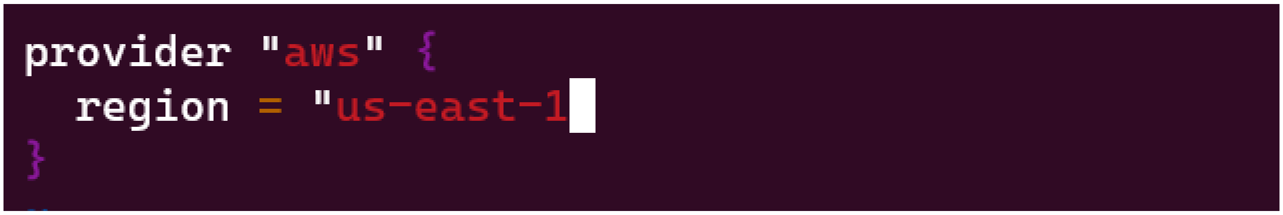
then inside vpc create public subnet with range 10.0.0.0/24 then create routing table for this vpc

3 - Search for the Ubuntu AMI ID then Create a security group to allow HTTP and HTTPS traffic

Finally Create an EC2 instance connected to the previously created resources.

The Solution :

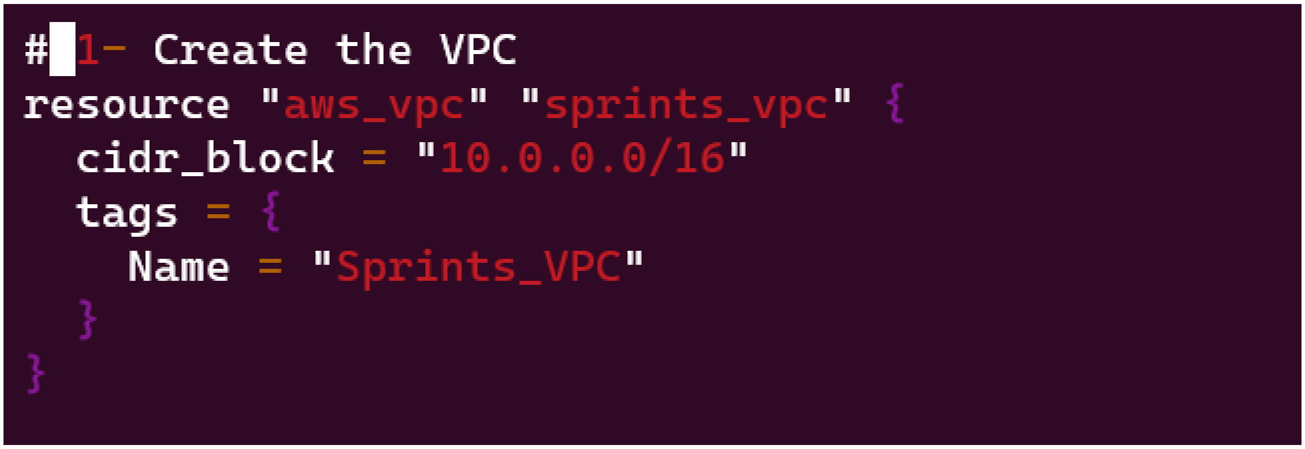
1 –



```
provider "aws" {
  region = "us-east-1
}
```

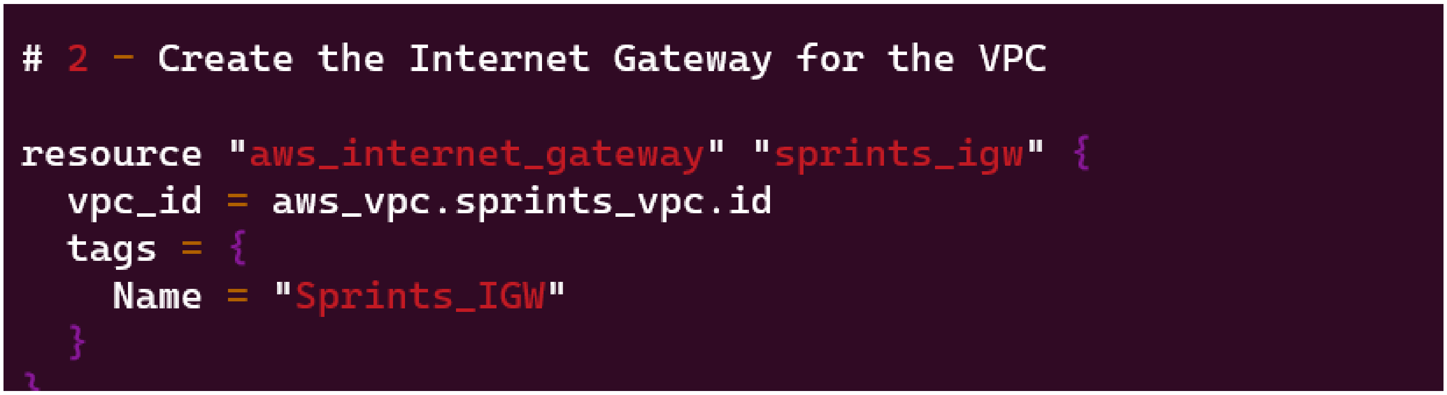
2 –

A – Create the VPC



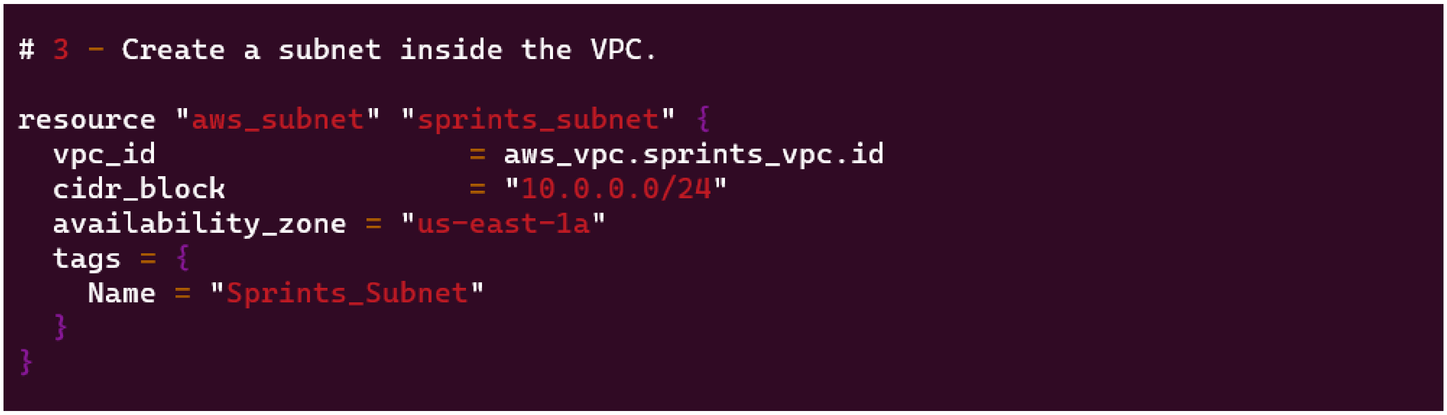
```
# 1- Create the VPC
resource "aws_vpc" "sprints_vpc" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "Sprints_VPC"
  }
}
```

B - Create the Internet Gateway for the VPC.



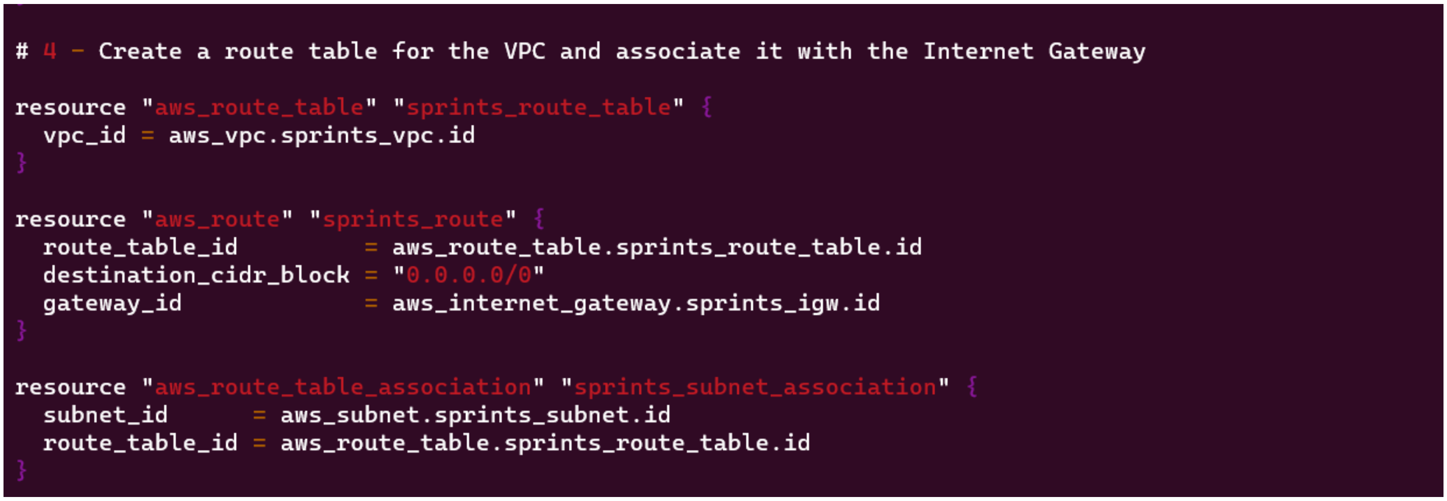
```
# 2 - Create the Internet Gateway for the VPC

resource "aws_internet_gateway" "sprints_igw" {
  vpc_id = aws_vpc.sprints_vpc.id
  tags = {
    Name = "Sprints_IGW"
  }
}
```

C - Create a subnet inside the VPC.



```
# 3 - Create a subnet inside the VPC.

resource "aws_subnet" "sprints_subnet" {
  vpc_id                  = aws_vpc.sprints_vpc.id
  cidr_block              = "10.0.0.0/24"
  availability_zone = "us-east-1a"
  tags = {
    Name = "Sprints_Subnet"
  }
}
```

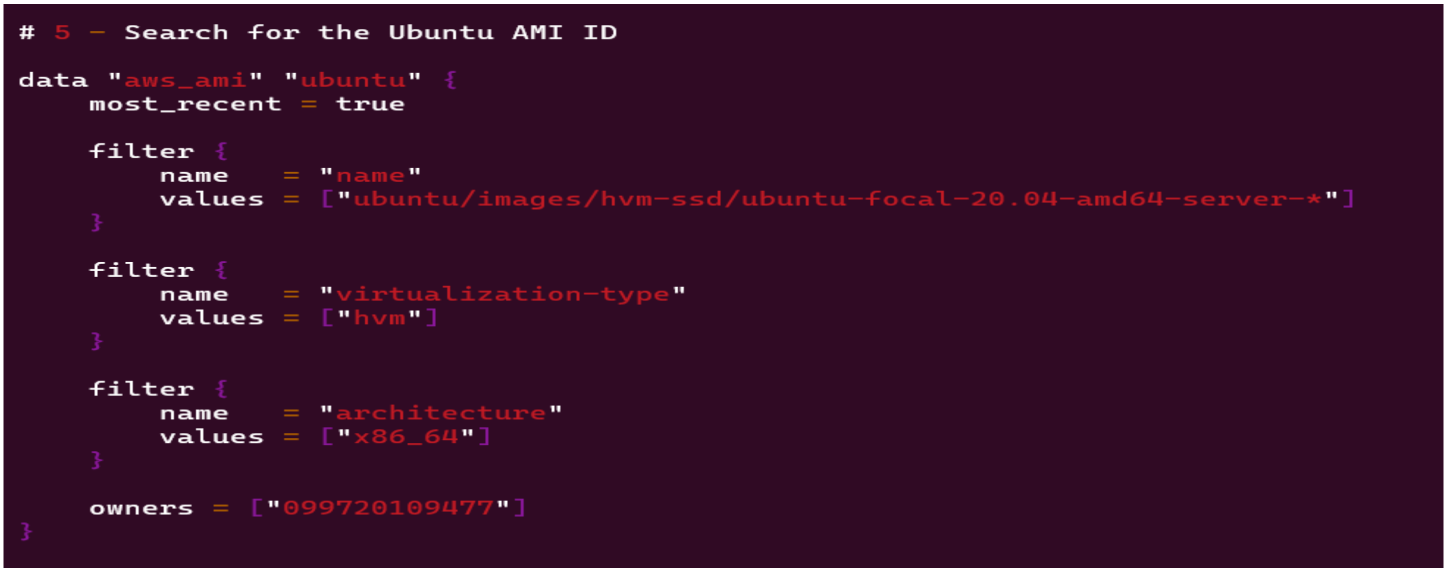
D - Create a route table for the VPC and associate it with the Internet Gateway.



```
# 4 - Create a route table for the VPC and associate it with the Internet Gateway

resource "aws_route_table" "sprints_route_table" {
  vpc_id = aws_vpc.sprints_vpc.id
}

resource "aws_route" "sprints_route" {
  route_table_id         = aws_route_table.sprints_route_table.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.sprints_igw.id
}

resource "aws_route_table_association" "sprints_subnet_association" {
  subnet_id      = aws_subnet.sprints_subnet.id
  route_table_id = aws_route_table.sprints_route_table.id
}
```

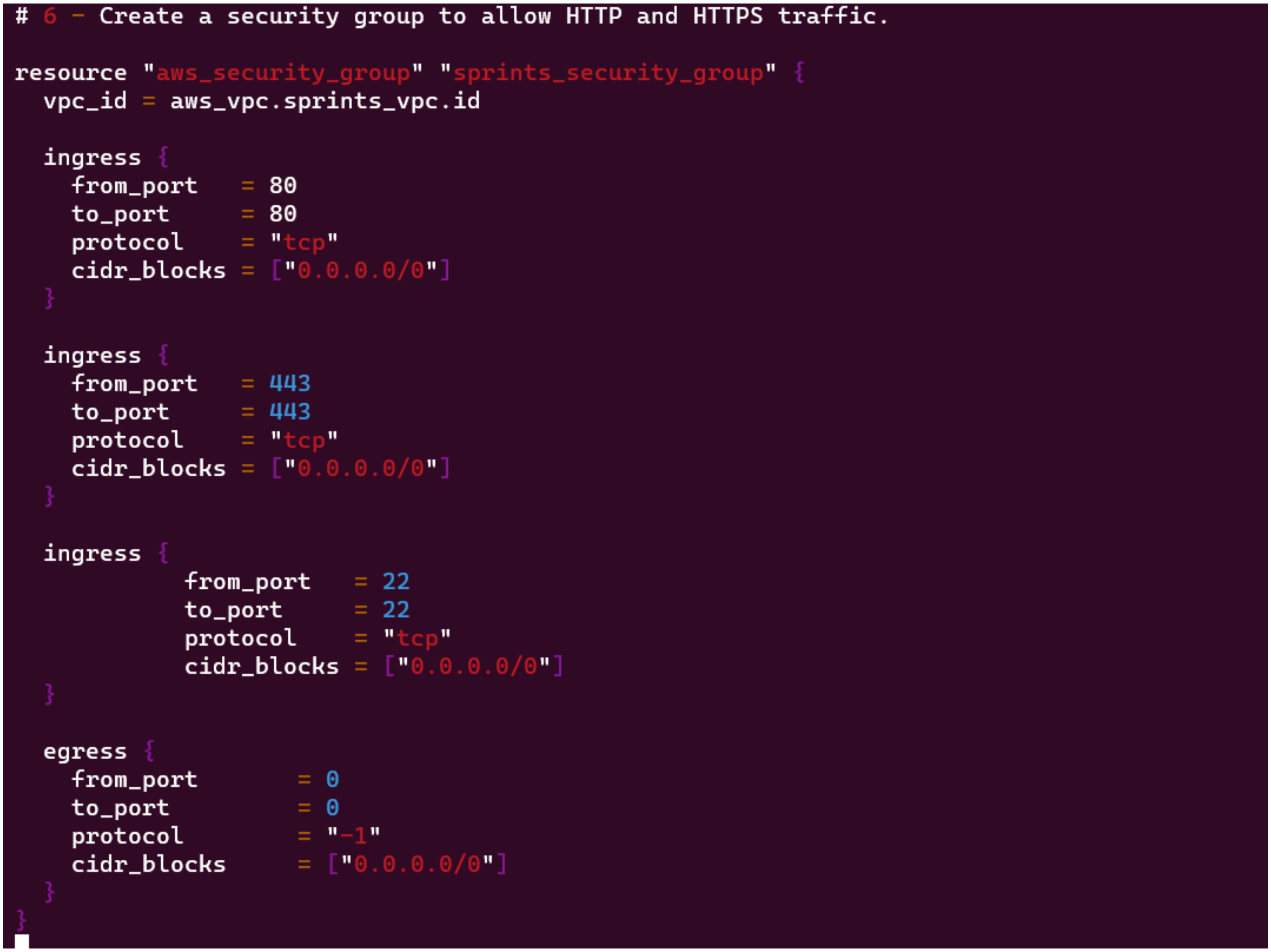
3 -

A - Search for the Ubuntu AMI ID.



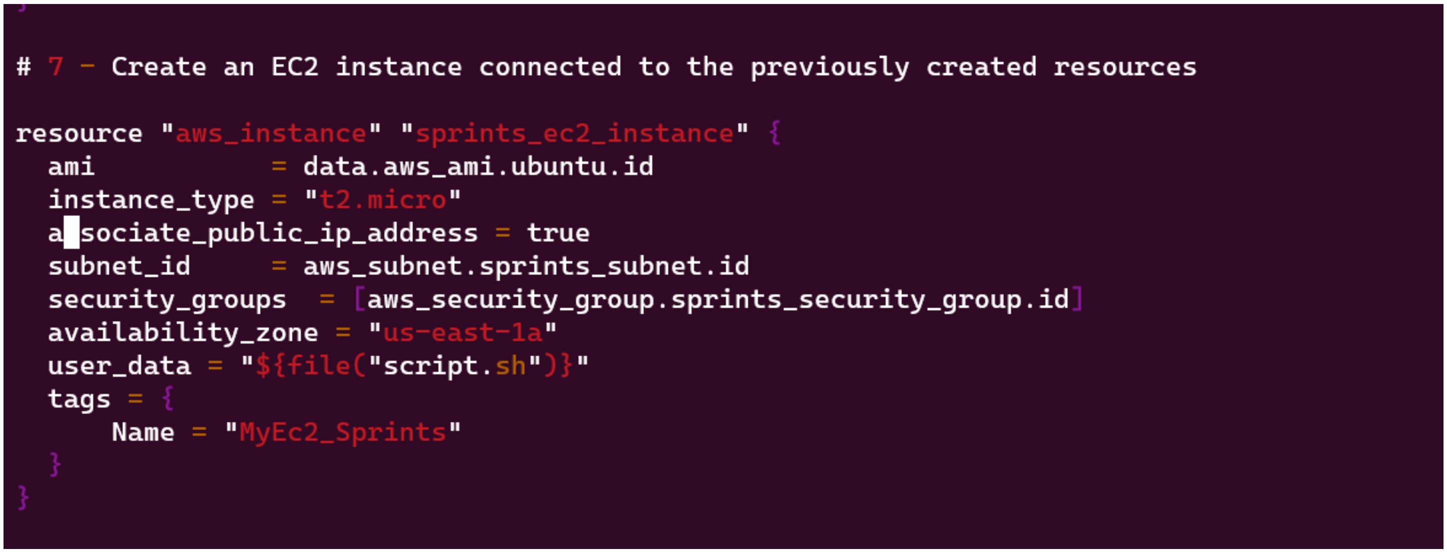
```
# 5 - Search for the Ubuntu AMI ID

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"]
}
```

B - Create a security group to allow HTTP and HTTPS traffic



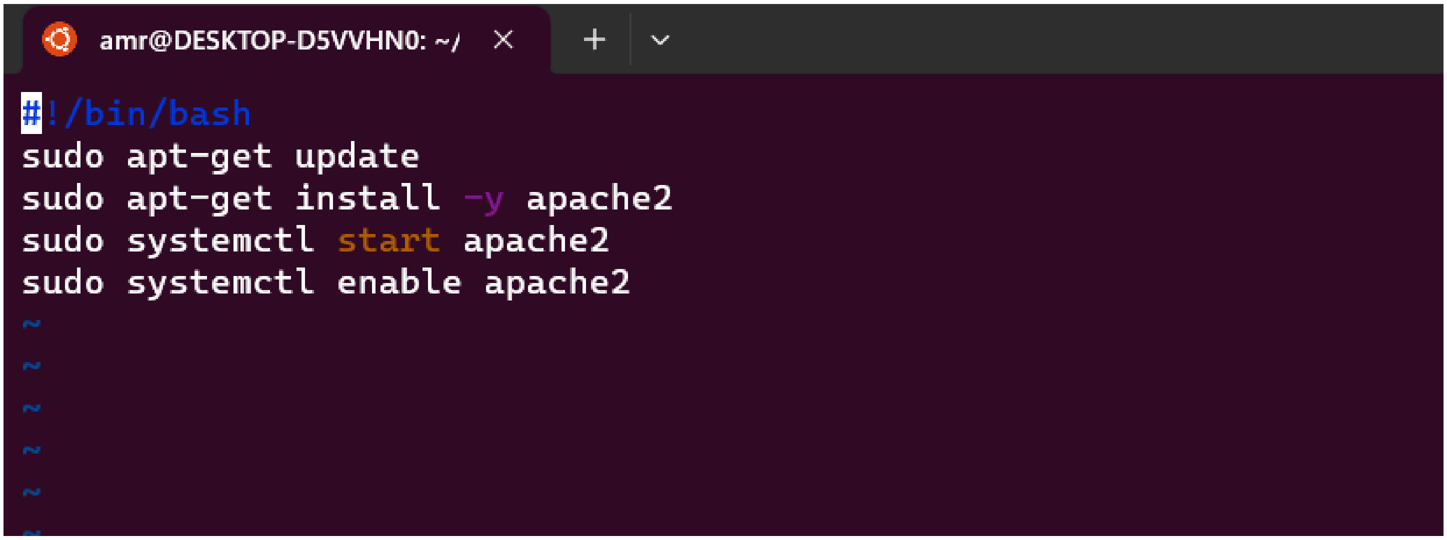
```
# 6 - Create a security group to allow HTTP and HTTPS traffic.

resource "aws_security_group" "sprints_security_group" {
  vpc_id = aws_vpc.sprints_vpc.id

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 443
    to_port     = 443
    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"]
  }
}
```

C - Create an EC2 instance connected to the previously created resources.



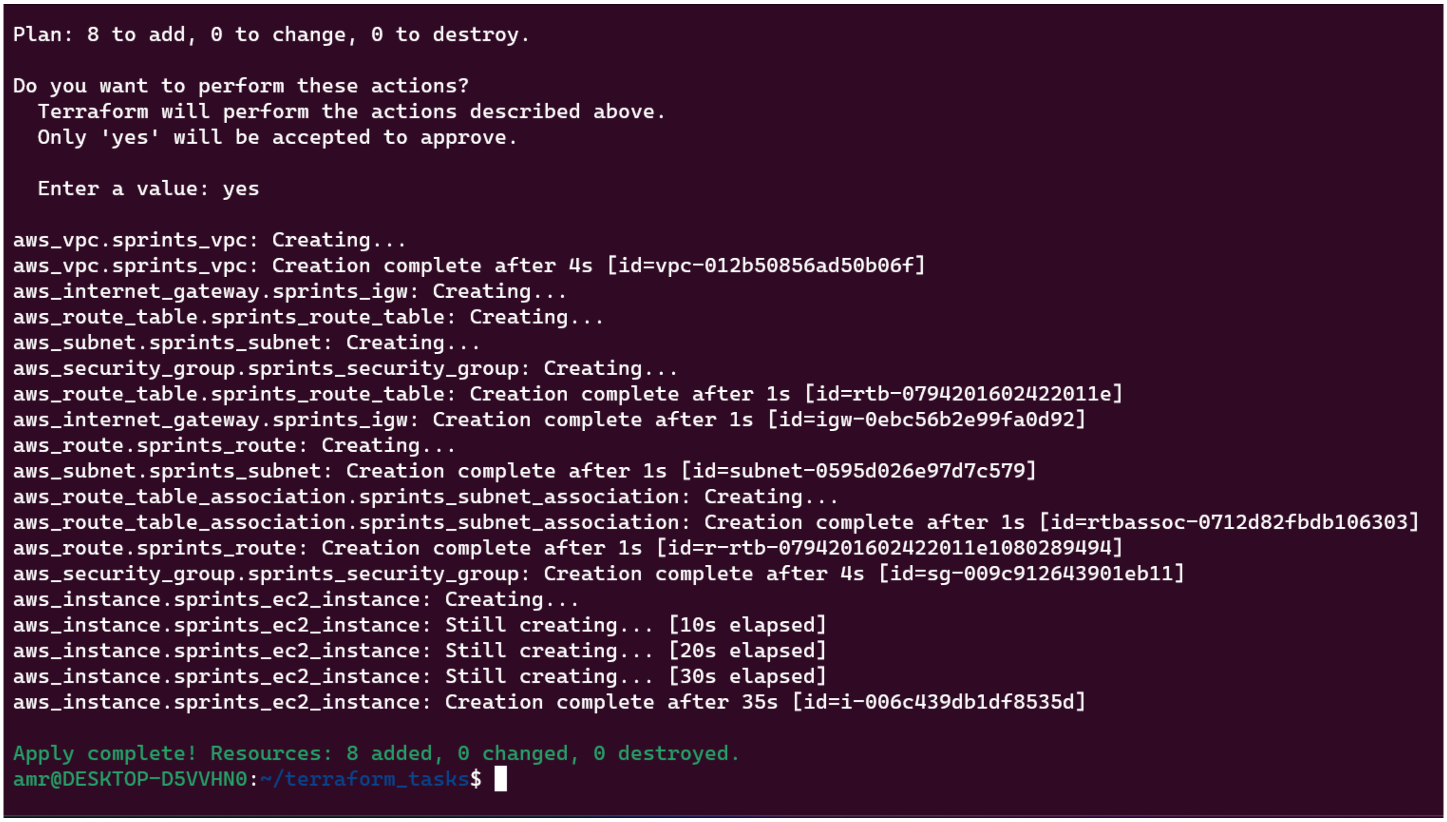
```
# 7 - Create an EC2 instance connected to the previously created resources

resource "aws_instance" "sprints_ec2_instance" {
  ami           = data.aws_ami.ubuntu.id
  instance_type = "t2.micro"
  associate_public_ip_address = true
  subnet_id     = aws_subnet.sprints_subnet.id
  security_groups  = [aws_security_group.sprints_security_group.id]
  availability_zone = "us-east-1a"
  user_data = "${file("script.sh")}"
  tags = {
      Name = "MyEc2_Sprints"
  }
}
```

Script.sh



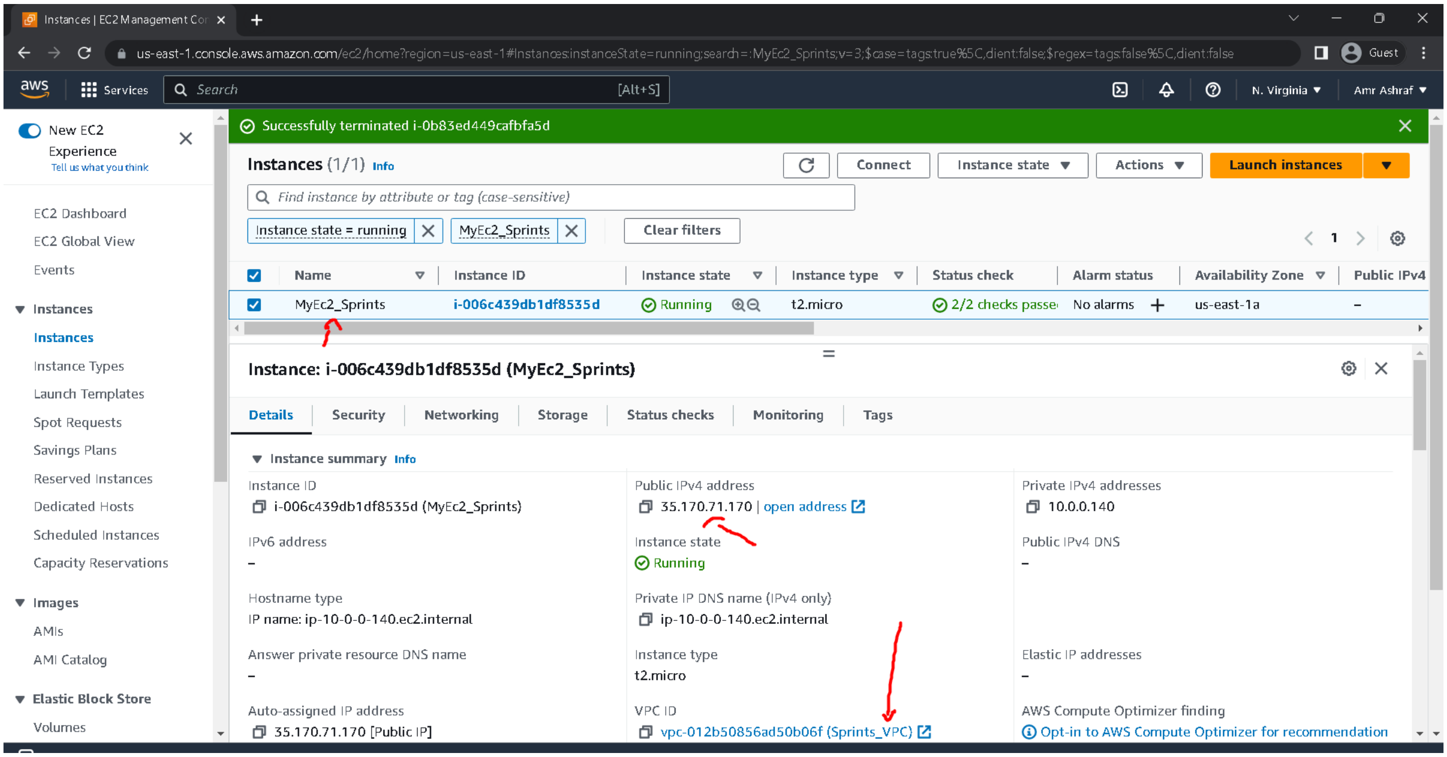
```
#!/bin/bash
sudo apt-get update
sudo apt-get install -y apache2
sudo systemctl start apache2
sudo systemctl enable apache2
~
~
~
~
~
```

The Results in terraform



```
Plan: 8 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.sprints_vpc: Creating...
aws_vpc.sprints_vpc: Creation complete after 4s [id=vpc-012b50856ad50b06f]
aws_internet_gateway.sprints_igw: Creating...
aws_route_table.sprints_route_table: Creating...
aws_subnet.sprints_subnet: Creating...
aws_security_group.sprints_security_group: Creating...
aws_route_table.sprints_route_table: Creation complete after 1s [id=rtb-0794201602422011e]
aws_internet_gateway.sprints_igw: Creation complete after 1s [id=igw-0ebc56b2e99fa0d92]
aws_route.sprints_route: Creating...
aws_subnet.sprints_subnet: Creation complete after 1s [id=subnet-0595d026e97d7c579]
aws_route_table_association.sprints_subnet_association: Creating...
aws_route_table_association.sprints_subnet_association: Creation complete after 1s [id=rtbassoc-0712d82fbdb106303]
aws_route.sprints_route: Creation complete after 1s [id=r-rtb-0794201602422011e1080289494]
aws_security_group.sprints_security_group: Creation complete after 4s [id=sg-009c912643901eb11]
aws_instance.sprints_ec2_instance: Creating...
aws_instance.sprints_ec2_instance: Still creating... [10s elapsed]
aws_instance.sprints_ec2_instance: Still creating... [20s elapsed]
aws_instance.sprints_ec2_instance: Still creating... [30s elapsed]
aws_instance.sprints_ec2_instance: Creation complete after 35s [id=i-006c439db1df8535d]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
amr@DESKTOP-D5VVHN0:~/terraform_tasks$
```

The Results in aws



Successfully terminated i-0b83ed449cafbfa5d

Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running | MyEc2_Sprints

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	MyEc2_Sprints	i-006c439db1df8535d	Running	t2.micro	2/2 checks passe	No alarms	us-east-1a	–

Instance: i-006c439db1df8535d (MyEc2_Sprints)

Instance summary Info

Instance ID
i-006c439db1df8535d (MyEc2_Sprints)

Public IPv4 address
35.170.71.170 | open address

Private IPv4 addresses
10.0.0.140

IPv6 address
–

Instance state
Running

Public IPv4 DNS
–

Hostname type
IP name: ip-10-0-0-140.ec2.internal

Private IP DNS name (IPv4 only)
ip-10-0-0-140.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

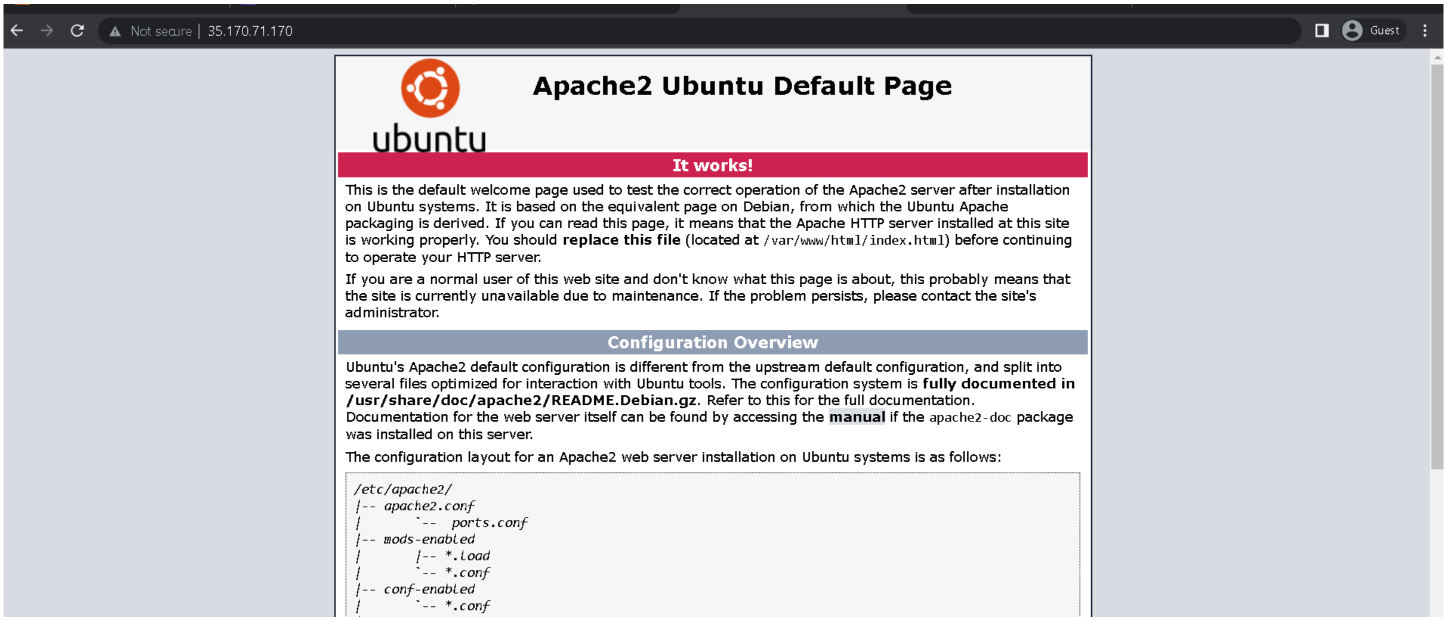
Elastic IP addresses
–

Auto-assigned IP address
35.170.71.170 [Public IP]

VPC ID
vpc-012b50856ad50b06f (Sprints_VPC)

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendation

The Apache result



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
```