- **Terraform Installation – Ubuntu**
  Website to Refer: https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

- **Credentials file for Terraform AWS Access**

  Get your Access token for AWS
  Create a folder

  ```
  mkdir ~/.aws
  nano ~/.aws/credentials
  ```

  Paste

  ```
  [default]
  aws_access_key_id = XXXXX
  aws_secret_access_key = XXXXX/XXXX
  ```

- **S3 - Central State File**

  Create an S3 bucket in the same location as the EC2

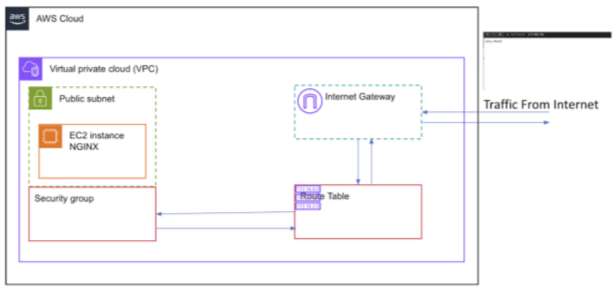
  ```
  terraform {
    backend "s3" {
      bucket = "terraform-project-new"
      key    = "terraform.tfstate"
      region = "us-east-2"
    }
  }
  ```

- **Terraform Commands**

  ```
  terraform validate
  terraform plan
  terraform apply
  terraform apply -auto-approve
  terraform destroy
  terraform fmt
  ```

- **Terraform AWS main.tf file for the below Architecture**

AWS Architecture



```
provider "aws" {
```

```
  region                   = "us-east-2"
  shared_credentials_files = ["~/.aws/credentials"]
  profile                  = "default"
}

# Central State File
terraform {
  backend "s3" {
    bucket = "terraform-project-new"
    key    = "terraform.tfstate"
    region = "us-east-2"
  }
}

# Define the VPC
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

# Define the subnet within the above VPC
resource "aws_subnet" "main" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"
}

# Define the internet gateway attached to the VPC
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
}

# Define the route table associated with the VPC
resource "aws_route_table" "r" {
  vpc_id = aws_vpc.main.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
}

# Associate the route table to the subnet
resource "aws_route_table_association" "a" {
  subnet_id      = aws_subnet.main.id
  route_table_id = aws_route_table.r.id
}

# Define the security group with rules for SSH (port
22) and HTTP (port 80)
resource "aws_security_group" "allow_web" {
  name        = "allow_web"
  description = "Allow all inbound traffic on ports
80 and 22"
  vpc_id      = aws_vpc.main.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
    # above cidr_block allows any IP to SSH.
  }

  ingress {
    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"]
  }
}

# Define the EC2 instance that will run NGINX
resource "aws_instance" "web" {
  ami                         = "ami-
024e6efaf93d85776"
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.main.id
  associate_public_ip_address = true
  vpc_security_group_ids      =
["${aws_security_group.allow_web.id}"]

  # Script to install NGINX and create a custom
index.html
  user_data = <<-EOF
#!/bin/bash
              sudo apt-get update
              sudo apt-get install -y nginx
              echo 'This is My Page - Krishna' >
/var/www/html/index.html
              systemctl start nginx
              systemctl enable nginx
              EOF
  tags = {
    Name = "nginx-webserver"
  }
```

```
}

output "instance_public_ip" {
  description = "The public IP address of the web
server"
  value       = aws_instance.web.public_ip
}
```