Terraform Basic

$ brew install terraform
$ brew upgrade
$ brew install awscli
$ aws configure

1. Exercise 1 : Instances

1st-instance.tf

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

resource "aws_instance" "intro" {
  ami                    = "ami-0cff7528ff583bf9a"
  instance_type          = "t2.micro"
  availability_zone      = "us-east-1a"
  key_name               = "teraform-basic-key"
  vpc_security_group_ids = ["sg-0141abaf6a6038247"]
  tags = {
    Name = "Terraform-Instance"
  }
}
```

$ terraform init
$ terraform validate
$ terraform fmt
$ terraform plan
$ terraform apply

$ terraform destroy

To check current state: $ cat terraform.tfstate

2. Exercise 2: Variables

Instance.tf

```
resource "aws_instance" "terraform-inst" {
  ami                    = var.AMIA[var.REGION]
  instance_type          = "t2.micro"
  availability_zone      = var.ZONE1
  key_name               = "teraform-basic-key"
  vpc_security_group_ids = ["sg-0141abaf6a6038247"]
  tags = {
    Name = "Terraform-Instance"
  }
}
```

provider.tf

```
provider "aws" {
  region = var.REGION
}
```

vars.tf

```
variable "REGION" {
  default = "us-east-1"
}
variable "ZONE1" {
  default = "us-east-1a"
}

variable "AMIA" {
  type = map(any)
  default = {
    us-east-1 = "ami-0cff7528ff583bf9a"
    us-east-2 = "ami-02d1e544b84bf7502"
  }
}
```

3. Exercise 3: Provisioners

```
$ ssh-keygen
      terrakey
```

var.tf

```
variable "REGION" {
  default = "us-east-1"
}
variable "ZONE1" {
  default = "us-east-1a"
}

variable "AMIA" {
  type = map(any)
  default = {
    us-east-1 = "ami-0cff7528ff583bf9a"
    us-east-2 = "ami-02d1e544b84bf7502"
  }
}
variable "USER" {
  default = "ec2-user"
}
```

provider.tf

```
provider "aws" {
  region = var.REGION
}
```

instance.tf

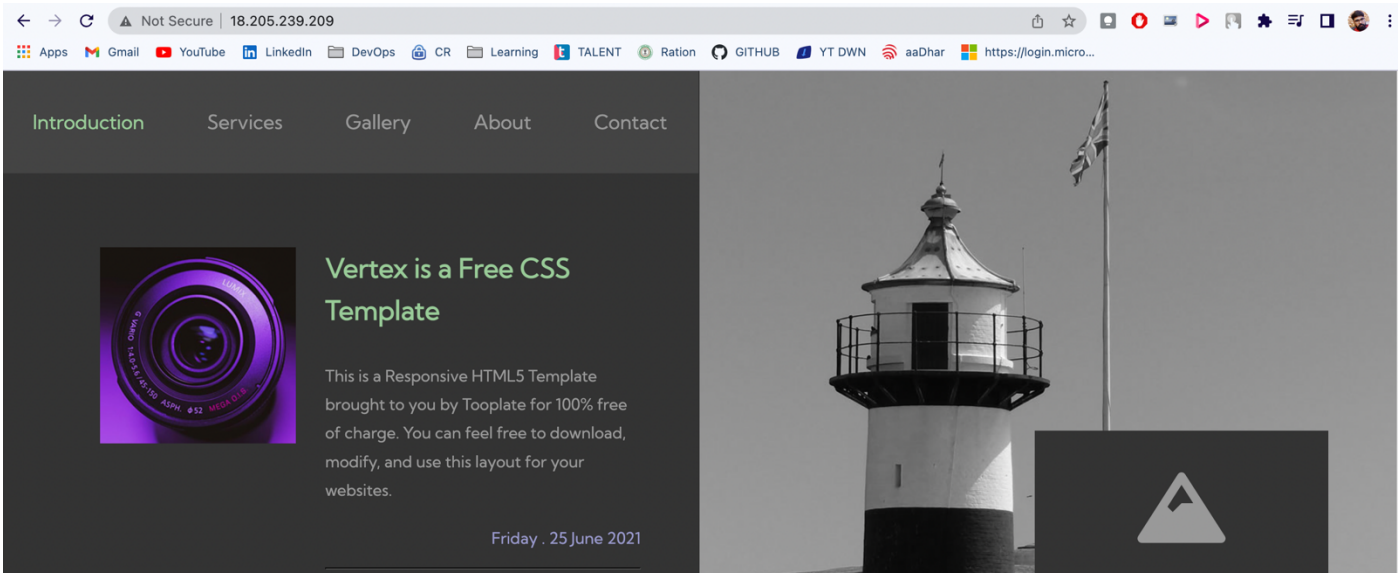
```
resource "aws_key_pair" "terra-key" {
  key_name   = "terrakey"
  public_key = file("terrakey.pub")
}

resource "aws_instance" "terraform-inst" {
  ami                    = var.AMIA[var.REGION]
  instance_type          = "t2.micro"
  availability_zone      = var.ZONE1
  key_name               = aws_key_pair.terra-key.key_name
  vpc_security_group_ids = ["sg-0141abaf6a6038247"]
  tags = {
    Name    = "Terraform-Instance"
    Project = "Terraform"
  }
  provisioner "file" {
    source      = "web.sh"
    destination = "/tmp/web.sh"
  }
  provisioner "remote-exec" {
    inline = [
      "chmod u+x /tmp/web.sh",
      "sudo /tmp/web.sh"
    ]
  }
  connection {
```

```
    user        = var.USER
    private_key = file("terrakey")
    host        = self.public_ip
  }
}

output "PublicIP" {
  value = aws_instance.terraform-inst.public_ip
}

output "PrivateIP" {
  value = aws_instance.terraform-inst.private_ip
}
```

web.sh

```
#!/bin/bash
yum install wget unzip httpd -y
systemctl start httpd
systemctl enable httpd
wget https://www.tooplate.com/zip-templates/2124_vertex.zip
unzip -o 2124_vertex.zip
cp -r 2124_vertex/* /var/www/html/
systemctl restart httpd
```

Copy the public IP of EC2 instance : http://18.205.239.209/



4. Exercise 4: output

var.tf

```
variable "REGION" {
  default = "us-east-1"
}
variable "ZONE1" {
  default = "us-east-1a"
}

variable "AMIA" {
  type = map(any)
  default = {
    us-east-1 = "ami-0cff7528ff583bf9a"
    us-east-2 = "ami-02d1e544b84bf7502"
  }
}
variable "USER" {
  default = "ec2-user"
}
```

provider.tf

```
provider "aws" {
  region = var.REGION
}
```

instance.tf

```
resource "aws_key_pair" "terra-key" {
  key_name   = "terrakey"
  public_key = file("terrakey.pub")
}

resource "aws_instance" "terraform-inst" {
  ami                    = var.AMIA[var.REGION]
  instance_type          = "t2.micro"
  availability_zone      = var.ZONE1
  key_name               = aws_key_pair.terra-key.key_name
  vpc_security_group_ids = ["sg-0141abaf6a6038247"]
  tags = {
    Name    = "Terraform-Instance"
    Project = "Terraform"
  }
  provisioner "file" {
    source      = "web.sh"
    destination = "/tmp/web.sh"
  }
  provisioner "remote-exec" {
    inline = [
      "chmod u+x /tmp/web.sh",
      "sudo /tmp/web.sh"
    ]
  }
  connection {
    user        = var.USER
    private_key = file("terrakey")
    host        = self.public_ip
```

```
  }
}

output "PublicIP" {
  value = aws_instance.terraform-inst.public_ip
}

output "PrivateIP" {
  value = aws_instance.terraform-inst.private_ip
}
```

web.sh

```
#!/bin/bash
yum install wget unzip httpd -y
systemctl start httpd
systemctl enable httpd
wget https://www.tooplate.com/zip-templates/2124_vertex.zip
unzip -o 2124_vertex.zip
cp -r 2124_vertex/* /var/www/html/
systemctl restart httpd
```

Outputs:

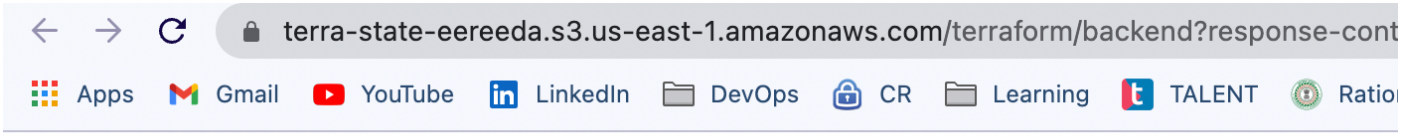
```
PrivateIP = "172.31.22.106"
PublicIP = "54.158.89.194"
```

5. Exercise 5: output

backend.tf

```
terraform {
  backend "s3" {
    bucket = "terra-state-eereeda"
    key = "terraform/backend"
    region = "us-east-1"
  }
}
```

Amazon S3 > Buckets > terra-state-eereeda > terraform > backend > open




```
{
  "version": 4,
  "terraform_version": "1.2.3",
  "serial": 33,
  "lineage": "7b609612-9024-b75c-321d-c0b7afdabb93",
  "outputs": {
    "PrivateIP": {
      "value": "172.31.22.106",
      "type": "string"
    },
    "PublicIP": {
      "value": "54.158.89.194",
      "type": "string"
    }
  },
  "resources": [
    {
      "mode": "managed",
      "type": "aws_instance",
      "name": "terraform-inst",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 1,
          "attributes": {
            "ami": "ami-0cff7528ff583bf9a",
            "arn": "arn:aws:ec2:us-east-1:162857390402:instance/i-0e1f114110807829c",
            "associate_public_ip_address": true,
            "availability_zone": "us-east-1a",
            "capacity_reservation_specification": [
              {
                "capacity_reservation_preference": "open",
                "capacity_reservation_target": []
              }
            ],
            "cpu_core_count": 1,
            "cpu_threads_per_core": 1,
            "credit_specification": [
              {
                "cpu_credits": "standard"
              }
            ],
            "disable_api_termination": false,
            "ebs_block_device": [],
            "ebs_optimized": false,
```