Module 8: Terraform Assignment - 5

1. Destroy the previous deployments
2. Create a script to install Apache2
3. Run this script on a newly created EC2 instance
4. Print the IP address of the instance in a file on the local once deployed.

Solution:-

$ sudo terraform destroy

```
ubuntu@terraform-server:~/tcode/assignment4$ sudo terraform destroy
aws_vpc.assignment-4-vpc: Refreshing state... [id=vpc-0babd84c5ce683037]
aws_subnet.assignment-4-subnet: Refreshing state... [id=subnet-08e6bca2520f9a372]
aws_instance.assignment-4: Refreshing state... [id=i-0e1e4c192737f1f79]

Terraform used the selected providers to generate the following execution plan. Reso
urce actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.assignment-4 will be destroyed
  - resource "aws_instance" "assignment-4" {
      - ami                                  = "ami-024e6efaf93d85776" -> null
      - arn                                  = "arn:aws:ec2:us-east-2:806224870762:i
nstance/i-0e1e4c192737f1f79" -> null
      - associate_public_ip_address          = false -> null
      - availability_zone                    = "us-east-2a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0e1e4c192737f1f79" -> null
```

```
  Enter a value: yes

aws_instance.assignment-4: Destroying... [id=i-0e1e4c192737f1f79]
aws_instance.assignment-4: Still destroying... [id=i-0e1e4c192737f1f79, 10s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-0e1e4c192737f1f79, 20s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-0e1e4c192737f1f79, 30s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-0e1e4c192737f1f79, 40s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-0e1e4c192737f1f79, 50s elapsed]
aws_instance.assignment-4: Destruction complete after 50s
aws_subnet.assignment-4-subnet: Destroying... [id=subnet-08e6bca2520f9a372]
aws_subnet.assignment-4-subnet: Destruction complete after 1s
aws_vpc.assignment-4-vpc: Destroying... [id=vpc-0babd84c5ce683037]
aws_vpc.assignment-4-vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
ubuntu@terraform-server:~/tcode/assignment4$
```

$ cd .. && sudo mkdir assignment5 && cd assignment5

```
ubuntu@terraform-server:~/tcode/assignment4$ cd .. && sudo mkdir assignment5 && cd assignment5
ubuntu@terraform-server:~/tcode/assignment5$
```

$ sudo vi provider.tf

```
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key = "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"
}
```

```
ubuntu@terraform-server:~/tcode/assignment5$ cat provider.tf
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key = "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"
}
```

$ main.tf

```
resource "aws_instance" "assignment-5" {
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        key_name = ""
        user_data = "${file("install-apache2.sh")}"
        tags = {
                Name = "assignment-5"
        }
```

}

output "public_ip" {
value = aws_instance.assignment-5.public_ip
}

```
ubuntu@terraform-server:~/tcode/assignment5$ cat main.tf
resource "aws_instance" "assignment-5" {
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        key_name = ""
        user_data = "${file("install-apache2.sh")}"
        tags = {
                Name = "assignment-5"
        }
}

output "public_ip" {
        value = aws_instance.assignment-5.public_ip
}
```

$ install-apache2.sh

#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo su
echo "Custom html page" > /var/www/html/index.html

# Capture the public IP address
instance_ip=$(terraform output public_ip)

Store the IP address in a local file
echo "$instance_ip" > instance_ip.txt

cat instance_ip.txt

```
ubuntu@terraform-server:~/tcode/assignment5$ cat install-apache2.sh
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo su
echo "Custom html page" > /var/www/html/index.html

# Capture the public IP address
instance_ip=$(terraform output public_ip)

# Store the IP address in a local file
echo "$instance_ip" > instance_ip.txt

cat instance_ip.txt
```

$ sudo terraform init

```
ubuntu@terraform-server:~/tcode/assignment5$ sudo terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.14.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

$ sudo terraform plan

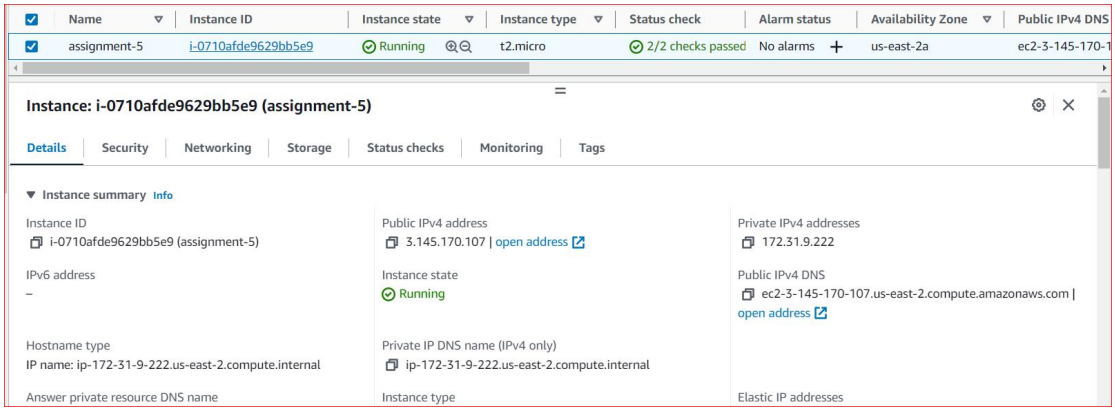
```
ubuntu@terraform-server:~/tcode/assignment5$ sudo terraform plan

Terraform used the selected providers to generate the following execution plan.
  + create

Terraform will perform the following actions:

  # aws_instance.assignment-5 will be created
  + resource "aws_instance" "assignment-5" {
      + ami                                  = "ami-024e6efaf93d85776"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
```

$ sudo terraform apply

```
ubuntu@terraform-server:~/tcode/assignment5$ sudo terraform apply

Terraform used the selected providers to generate the following execution
  + create

Terraform will perform the following actions:

  # aws_instance.assignment-5 will be created
  + resource "aws_instance" "assignment-5" {
      + ami                                  = "ami-024e6efaf93d85776"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
```

```
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_instance.assignment-5: Creating...
aws_instance.assignment-5: Still creating... [10s elapsed]
aws_instance.assignment-5: Still creating... [20s elapsed]
aws_instance.assignment-5: Still creating... [30s elapsed]
aws_instance.assignment-5: Creation complete after 32s [id=i-0710afde9629bb5e9]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

public_ip = "3.145.170.107"
```

* Instance created present in AWS console:-



* Apache2 output with :-