Module 8: Terraform Assignment-3

Tasks To Be Performed:

1. Destroy the previous deployment

2. Create 2 EC2 instances in Ohio and N.Virginia respectively

3. Rename Ohio's instance to 'hello-ohio' and Virginia's instance to

'hello-virginia'

Solution:

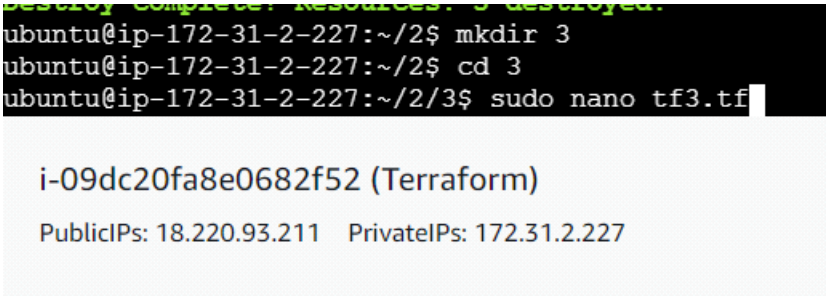
Step 1: Destroy the Previous Deployment

Navigate to the directory containing your Terraform configuration files and run the following command to destroy the previous deployment:

terraform destroy

Terraform will show you the resources that will be destroyed and prompt you for confirmation. Type yes and press Enter.

Step 2: Write the Terraform Configuration




```
provider "aws" {
        alias = "NV"
        region = "us-east-1"
        access_key = " "
        secret_key = " "
}
provider "aws" {
        alias = "Ohio"
        region = "us-east-2"
        access_key = "AKIASLJH4JT2OBYRVB7D"
        secret_key = "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"
}
resource "aws_instance" "hello-ohio" {
        provider = aws.Ohio
        ami = "ami-07d7e3e669718ab45"
        instance_type = "t2.micro"
        tags = {
        name = "hello-ohio"
        }
```

```
}
resource "aws_instance" "hello-virginia" {
        provider = aws.NV
        ami = "ami-01b799c439fd5516a"
        instance_type = "t2.micro"
        tags = {
        name = "hello-virginia"
        }
```

Step 3: Initialize Terraform

Run the following command to initialize Terraform. This will download the necessary provider plugins**:**

terraform init

```
ubuntu@ip-172-31-2-227:~/2/3$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.55.0...
- Installed hashicorp/aws v5.55.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
ubuntu@ip-172-31-2-227:~/2/3$
```

i-09dc20fa8e0682f52 (Terraform)

PublicIPs: 18.220.93.211 PrivateIPs: 172.31.2.227

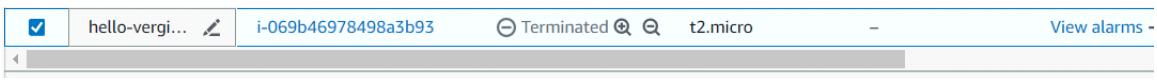
Step 4: Apply the Terraform Configuration

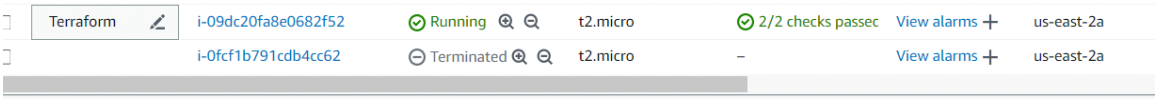
Run the following command to create the EC2 instance

terraform apply

Step 5: Verify the EC2 Instance

Once the Terraform apply command completes, you can verify that the EC2 instance has been created by logging into the AWS Management Console and navigating to the EC2 dashboard in the Ohio region.




Terraform
i-09dc20fa8e0682f52
Running
t2.micro
2/2 checks passec
View alarms
us-east-2a
i-0fcf1b791cdb4cc62
Terminated
t2.micro
–
View alarms
us-east-2a