

Module 8: Terraform Assignment - 1

1. Create an EC2 service in the default subnet in the Ohio region.

Solution:-

```
$ curl -O https://releases.hashicorp.com/terraform/1.5.5/terraform_1.5.5_linux_amd64.zip
$ sudo apt-get install unzip
```

```
ubuntu@terraform-server:~$ curl -O https://releases.hashicorp.com/terraform/1.5.5/terraform_1.5.5_linux_amd64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left   Speed
100 20.0M  100 20.0M    0     0  80.8M      0 --:--:-- --:--:-- --:--:-- 81.1M
ubuntu@terraform-server:~$ ls
terraform_1.5.5_linux_amd64.zip
ubuntu@terraform-server:~$ sudo apt-get install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

```
$ unzip terraform_1.5.5_linux_amd64.zip
$ sudo mv terraform /usr/local/bin
```

```
ubuntu@terraform-server:~$ unzip terraform_1.5.5_linux_amd64.zip
Archive:  terraform_1.5.5_linux_amd64.zip
  inflating: terraform
ubuntu@terraform-server:~$ ls
terraform  terraform_1.5.5_linux_amd64.zip
ubuntu@terraform-server:~$ sudo mv terraform /usr/local/bin
```

```
$ terraform version
$ sudo mkdir -p tcode/assignment1 && cd tcode/assignment1
```

```
ubuntu@terraform-server:~$ terraform version
Terraform v1.5.5
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.5.6. You can update by downloading from https://www.terraform.io/downloads.html
ubuntu@terraform-server:~$ sudo mkdir -p tcode/assignment1 && cd tcode/assignment1
ubuntu@terraform-server:~/tcode/assignment1$
```

```
$ sudo vi provider.tf
```

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

```
$ sudo vi main.tf
```

```
resource "aws_instance" "assignment-1" {  
    ami = "ami-024e6efaf93d85776"  
    instance_type = "t2.micro"  
    key_name = "terraform_key"  
    tags = {  
        Name = "assignment-1"  
    }  
}
```

```
ubuntu@terraform-server:~/tcode/assignment1$ cat provider.tf  
provider "aws" {  
    region = "us-east-2"  
    access_key = "AKIA3XNV7HVVOZH64X44"  
    secret_key = "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"  
}  
ubuntu@terraform-server:~/tcode/assignment1$ cat main.tf  
resource "aws_instance" "assignment-1" {  
    ami = "ami-024e6efaf93d85776"  
    instance_type = "t2.micro"  
    key_name = "terraform_key"  
    tags = {  
        Name = "assignment-1"  
    }  
}  
ubuntu@terraform-server:~/tcode/assignment1$
```

```
$ sudo terraform init
```

```
ubuntu@terraform-server:~/tcode/assignment1$ 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/assignment1$ sudo terraform plan

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

Terraform will perform the following actions:

# aws_instance.assignment-1 will be created
+ resource "aws_instance" "assignment-1" {
  + 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)
```

```
$ sudo terraform apply
```

```
ubuntu@terraform-server:~/tcode/assignment1$ sudo terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.assignment-1 will be created
+ resource "aws_instance" "assignment-1" {
  + 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
  + host_id                  = (known after apply)
  + host_resource_group_arn   = (known after apply)
  + iam_instance_profile      = (known after apply)
  + id                      = (known after apply)
  + instance_type             = (known after apply)
  + key_name                  = (known after apply)
  + monitoring                 = (known after apply)
  + placement_group           = (known after apply)
  + primary_interface         = (known after apply)
  + private_ip                = (known after apply)
  + public_ip                 = (known after apply)
  + subnet_id                 = (known after apply)
  + tags                      = {}
  + user_data                  = (known after apply)
  + vpc_security_group_ids    = (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-1: Creating...
aws_instance.assignment-1: Still creating... [10s elapsed]
aws_instance.assignment-1: Still creating... [20s elapsed]
aws_instance.assignment-1: Still creating... [30s elapsed]
aws_instance.assignment-1: Creation complete after 32s [id=i-07f047b84779d654c]

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

* Instance created present in AWS console:-

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input checked="" type="checkbox"/>	assignment-1	i-07f047b84779d654c	Running	t2.micro	Initializing	No alarms	us-east-2a	ec2-3-141-

Instance: i-07f047b84779d654c (assignment-1)

Optional

▼ Instance details Info

Platform

Ubuntu (Inferred)

Platform details

Linux/UNIX

Stop protection

Disabled

Instance auto-recovery

Default

AMI ID

ami-024e6efaf93d85776

AMI name

ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516

Launch time

Sat Aug 26 2023 10:11:51 GMT+0530 (India Standard Time) (7 minutes)

Lifecycle

normal

Monitoring

disabled

Termination protection

Disabled

AMI location

amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230516

Stop-hibernate behavior

disabled