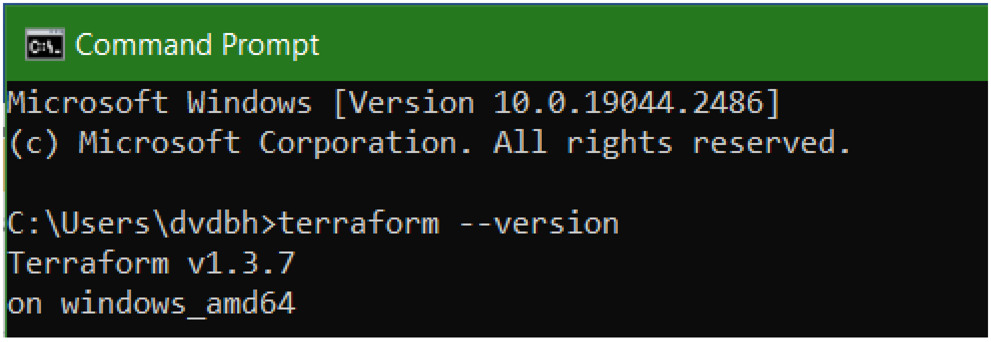
CLOUD DEVOPS

LAB-2

ID:2000031248

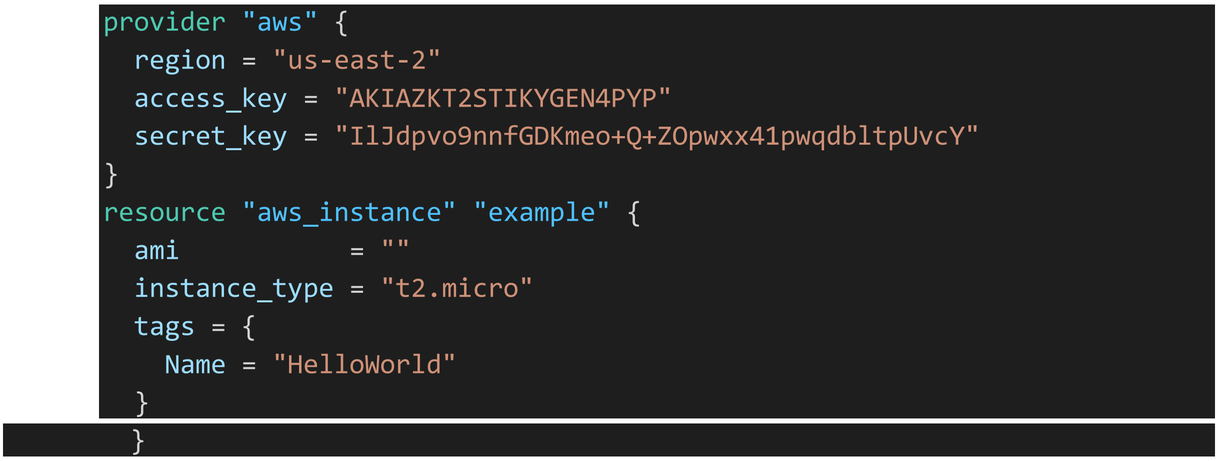
Name: D V D Bhargava

1. **Install Terraform**
   - Go to https://developer.hashicorp.com/terraform/downloads?product_intent=terraform Select Windows
   - Now click on AMD64 and download the zip file
   - Extract the Zip file
   - Now go to environment variables in your System and past the path where you have downloaded the zip file.



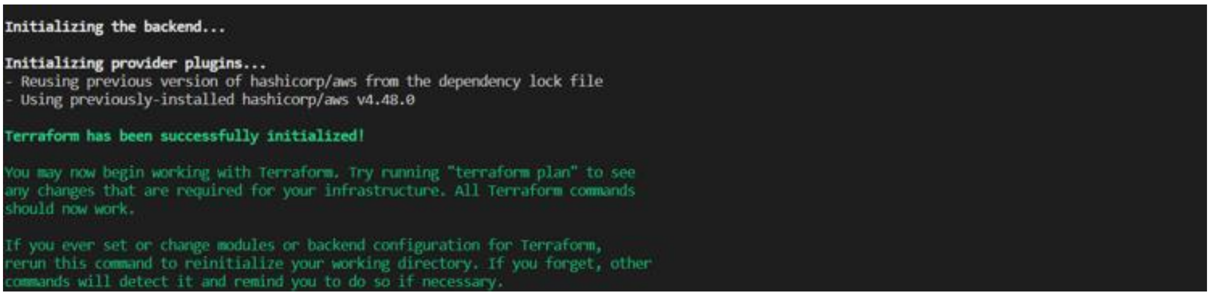
```
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dvdbh>terraform --version
Terraform v1.3.7
on windows_amd64
```

2. **Build and test basic terraform module**
   - Now let us build a basic EC2 instance using Terraform
   - Create a file named main.tf and type the sample code given below.



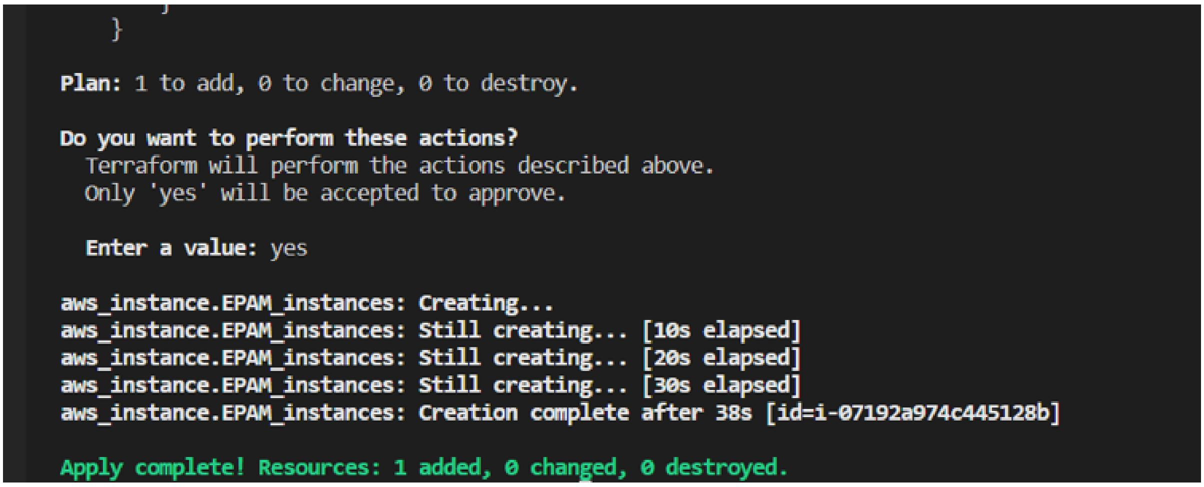
```
provider "aws" {
  region = "us-east-2"
  access_key = "AKIAZKT2STIKYGEN4PYP"
  secret_key = "IlJdpvo9nnfGDKmeo+Q+ZOpwxx41pwqdbltpUvcY"
}
resource "aws_instance" "example" {
  ami           = ""
  instance_type = "t2.micro"
  tags = {
    Name = "HelloWorld"
  }
  }
```

- Now in command prompt type the given commands.
- Terraform init
- Terraform plan
- Terraform appl

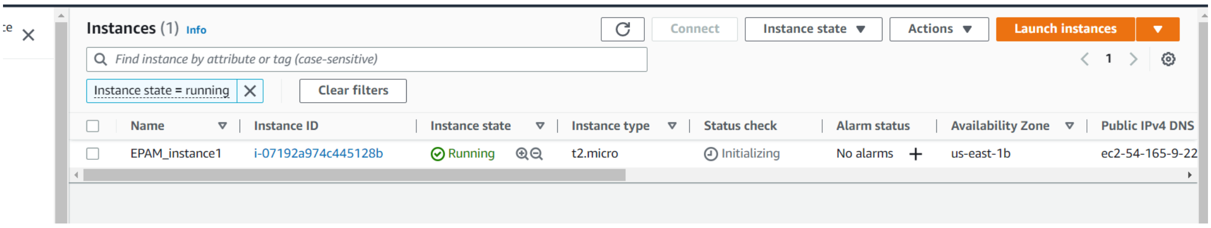


```
Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.48.0

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.
```

```
Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.48.0

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.
```



```
    }

Plan: 1 to add, 0 to change, 0 to destroy.

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.EPAM_instances: Creating...
aws_instance.EPAM_instances: Still creating... [10s elapsed]
aws_instance.EPAM_instances: Still creating... [20s elapsed]
aws_instance.EPAM_instances: Still creating... [30s elapsed]
aws_instance.EPAM_instances: Creation complete after 38s [id=i-07192a974c445128b]

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



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	EPAM_instance1	i-07192a974c445128b	✓ Running	t2.micro	Initializing	No alarms +	us-east-1b	ec2-54-165-9-22