Module 8: Terraform Assignment - 1

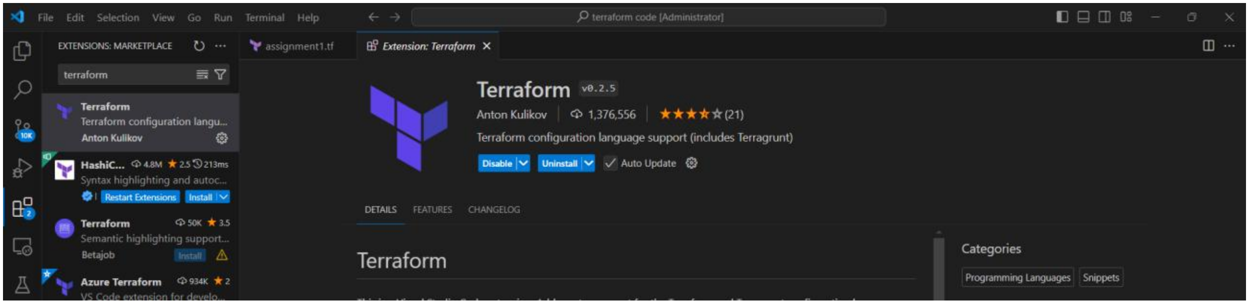
Assignment Submitted By:-Hitesh Chauhan
Course Offered: -Advanced Cloud Computing and Devops
Assignment By: -Intellipaat
Trainer: -Kumar
Date Of Submission: -02/03/2025

Tasks To Be Performed:

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

SOLUTION

First, We need to prerequisites for terraform then require visual studio code with extension terraform for terraform coding.



Terraform
Terraform configuration langu...
Anton Kulikov

HashiC... 4.8M ★ 2.5 213ms
Syntax highlighting and autoc...

Terraform 50K ★ 3.5
Semantic highlighting support...
Betajob

Azure Terraform 934K ★ 2
VS Code extension for develo...

Terraform v0.2.5
Anton Kulikov | 1,376,556 | ★★★★☆ (21)
Terraform configuration language support (includes Terragrunt)

Terraform

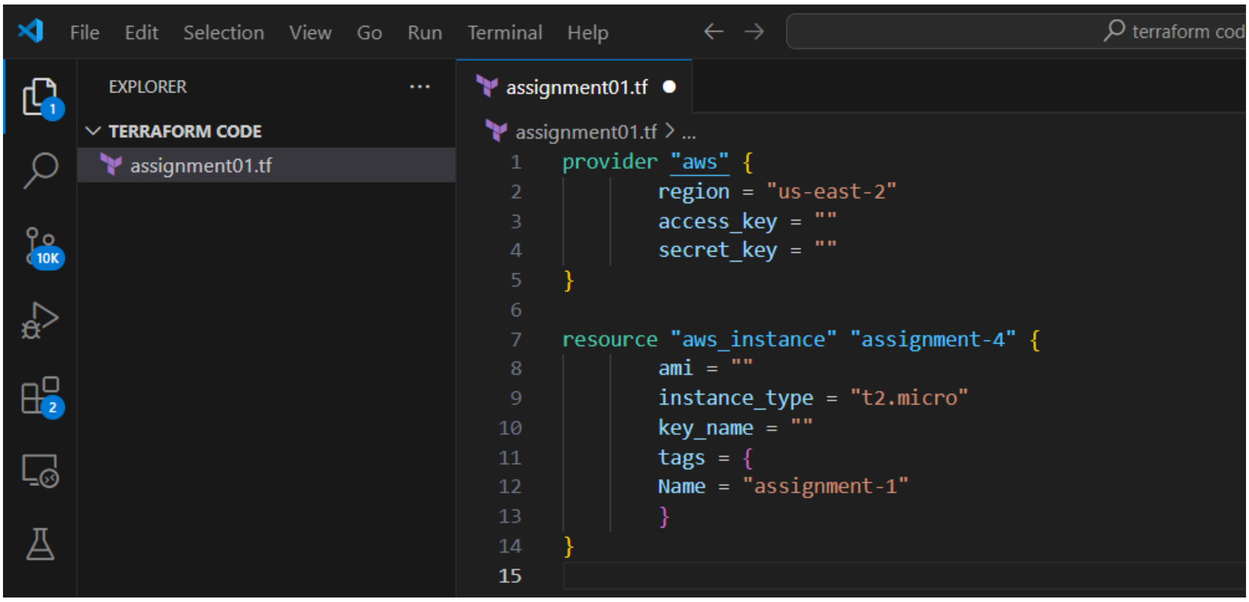
Categories
Programming Languages
Snippets

We need to install the terraform in our system.
In my case I have wsl (windows subsystem linux featured enabled).
You can install terraform any system like cloud and windows pc and linux.

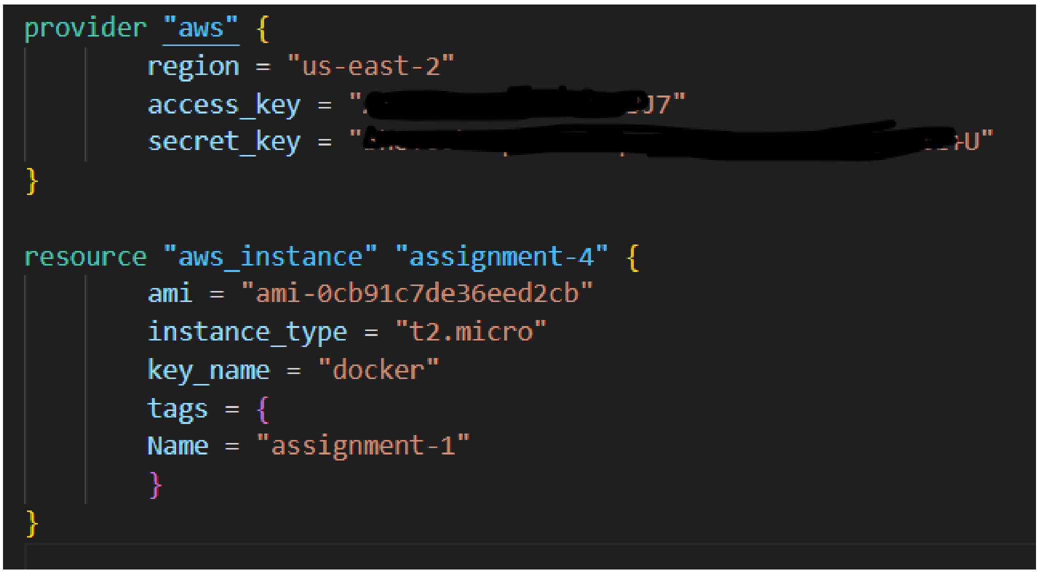
This Is the official documents
https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

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

   Create folder for terraform code and save this code like below.

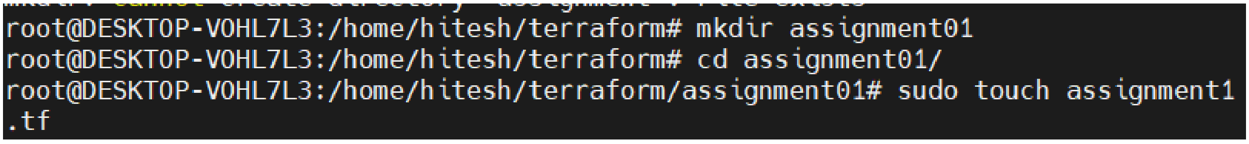


```
provider "aws" {
        region = "us-east-2"
        access_key = ""
        secret_key = ""
}

resource "aws_instance" "assignment-4" {
        ami = ""
        instance_type = "t2.micro"
        key_name = ""
        tags = {
        Name = "assignment-1"
        }
}
```



```
provider "aws" {
        region = "us-east-2"
        access_key = "[redacted]"
        secret_key = "[redacted]"
}

resource "aws_instance" "assignment-4" {
        ami = "ami-0cb91c7de36eed2cb"
        instance_type = "t2.micro"
        key_name = "docker"
        tags = {
        Name = "assignment-1"
        }
}
```

2. Create Directory.
3. In my case I created assignment 01directory.in under directory there is one more file name is assignment1.tf file.



```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform# mkdir assignment01
root@DESKTOP-VOHL7L3:/home/hitesh/terraform# cd assignment01/
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# sudo touch assignment1
.tf
```

In this file mentioned below code .

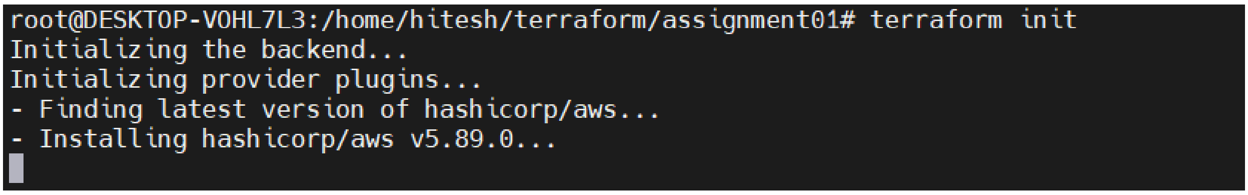
```
provider "aws" {
	region = "us-east-2"
	access_key = ""
	secret_key = ""
}

resource "aws_instance" "assignment-4" {
	ami = ""
	instance_type = "t2.micro"
	key_name = ""
	tags = {
	Name = "assignment-1"
	}
}
```

Now Save this code .tf.in my case I used to **assignment1.tf** file name extension.

Then Deploy the code.

For Deploy code we need to run command like this

terraform init



```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.89.0...
```

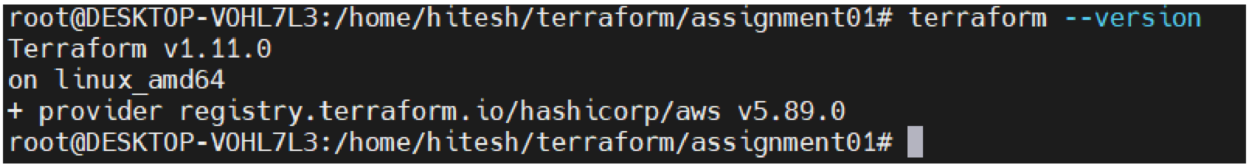
When you run this command download the plugins automatically from terraform.

After Done this you will got information like terraform has been successfully initialized.now you can deploy code.

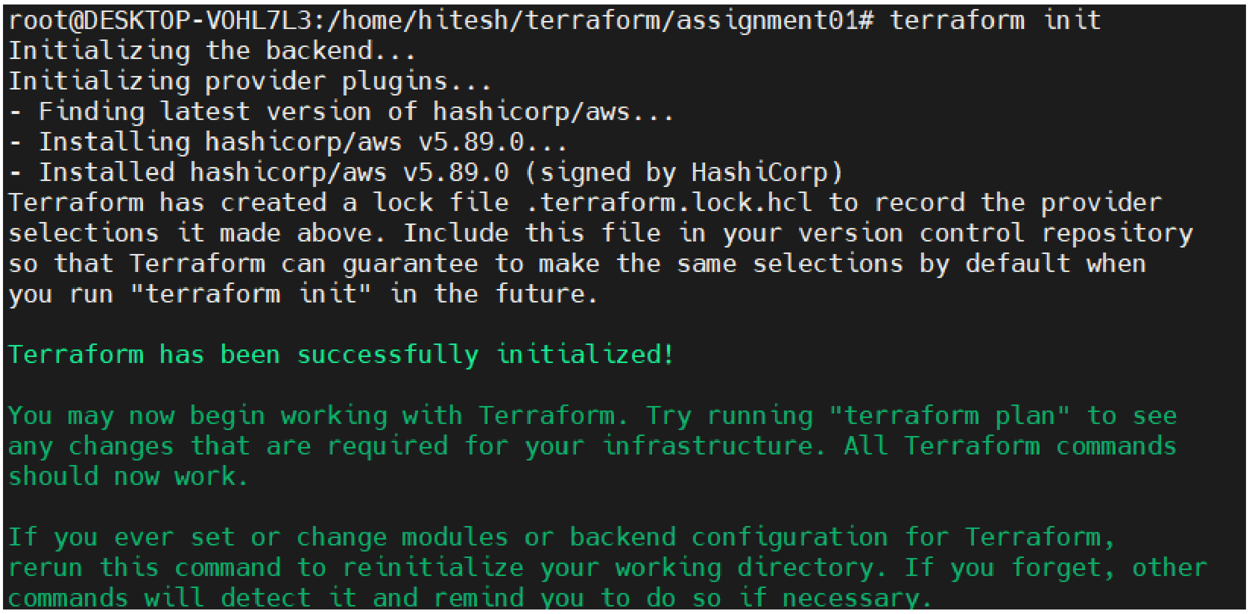
Before we need to check the version.

How to check the terraform version so this is command

terraform --version



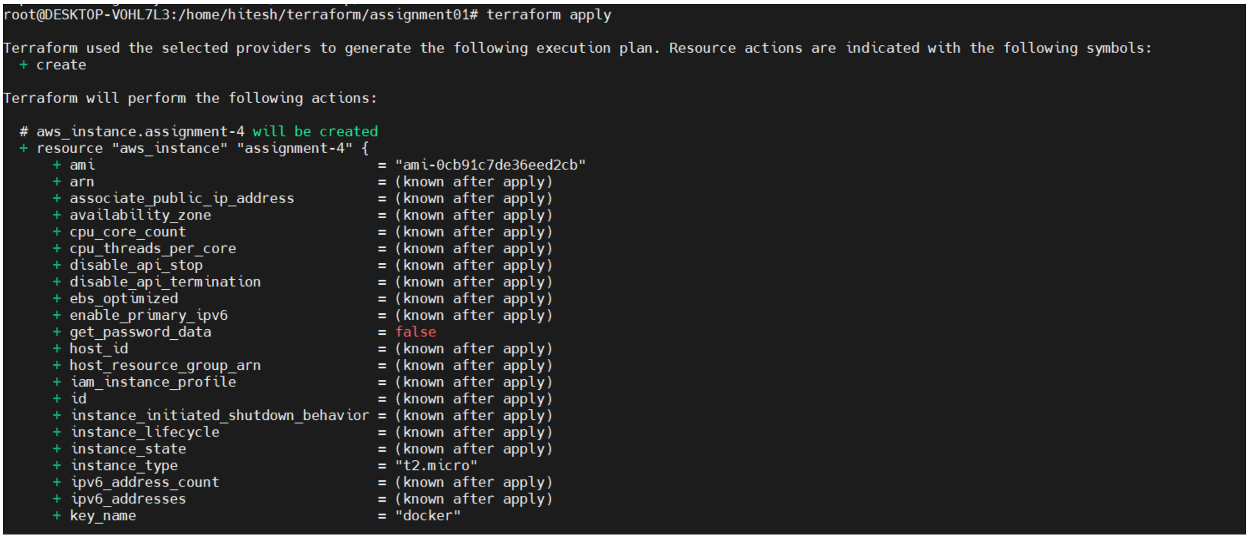
```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# terraform --version
Terraform v1.11.0
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v5.89.0
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01#
```



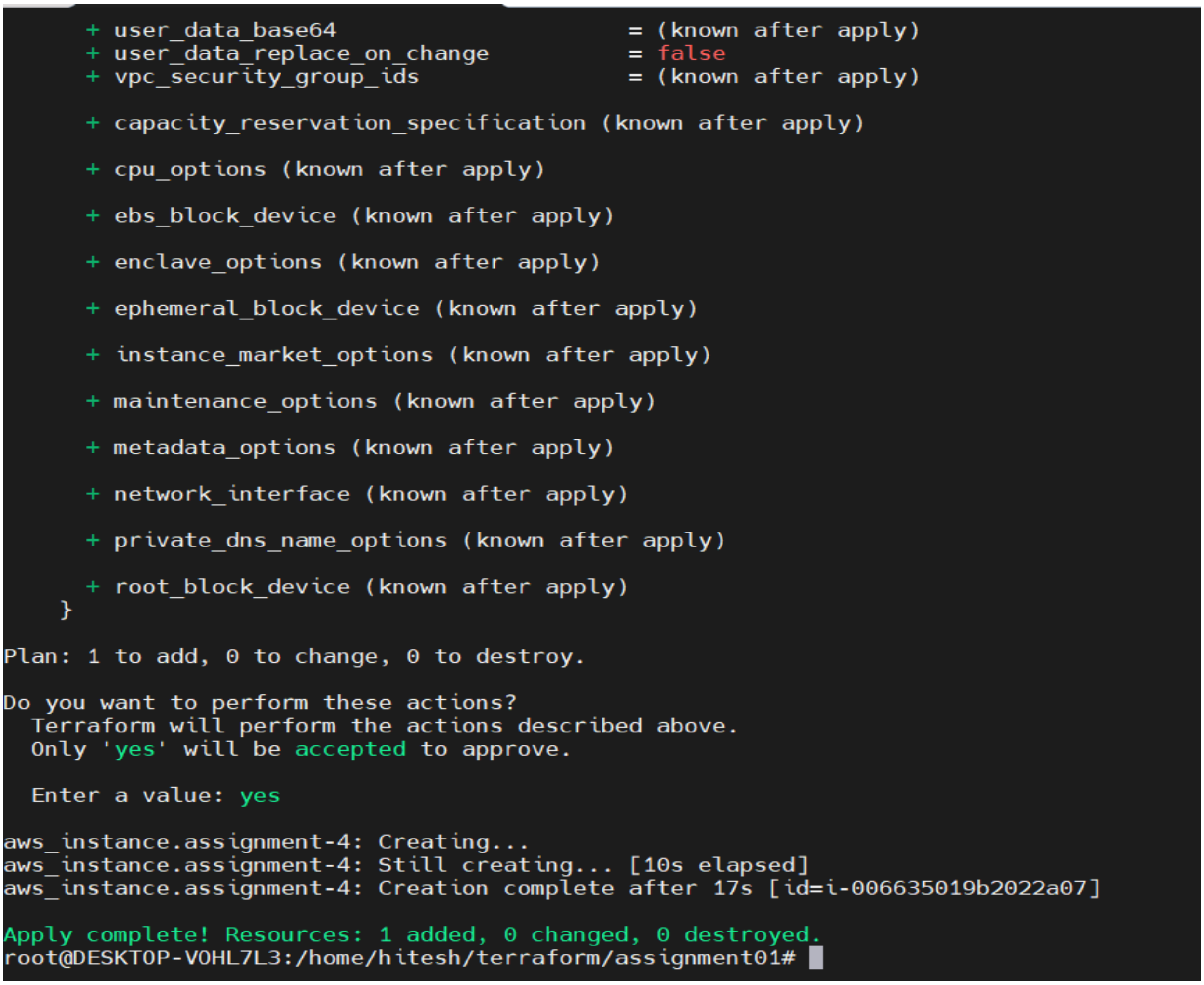
```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.89.0...
- Installed hashicorp/aws v5.89.0 (signed by HashiCorp)
Terraform has created a 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.
```

Now We have to run the command for deployment of ec2 in ohio region.

terraform apply



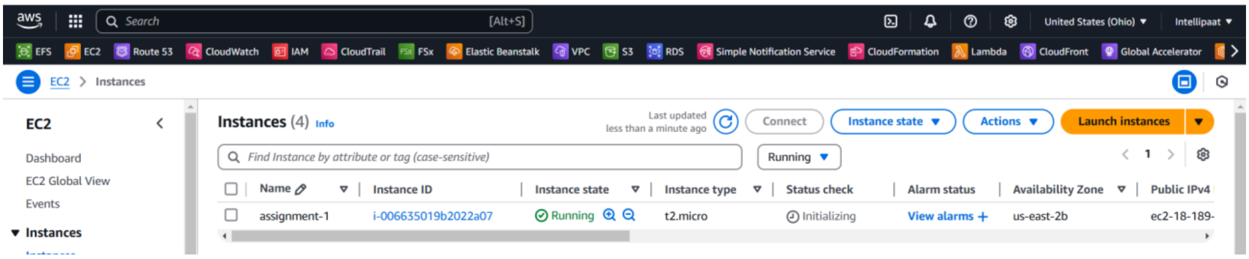
```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# terraform apply

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-4 will be created
  + resource "aws_instance" "assignment-4" {
      + ami                                  = "ami-0cb91c7de36eed2cb"
      + 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)
      + enable_primary_ipv6                  = (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_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "docker"
```



```
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

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.assignment-4: Creating...
aws_instance.assignment-4: Still creating... [10s elapsed]
aws_instance.assignment-4: Creation complete after 17s [id=i-006635019b2022a07]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment01# 
```

Now Our First deployment has been completed successfully.

We need to check the deployment in aws console region ohio.



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	assignment-1	i-006635019b2022a07	✓ Running	t2.micro	Initializing	View alarms +	us-east-2b	ec2-18-189-

The Deployment Has been successfully done.