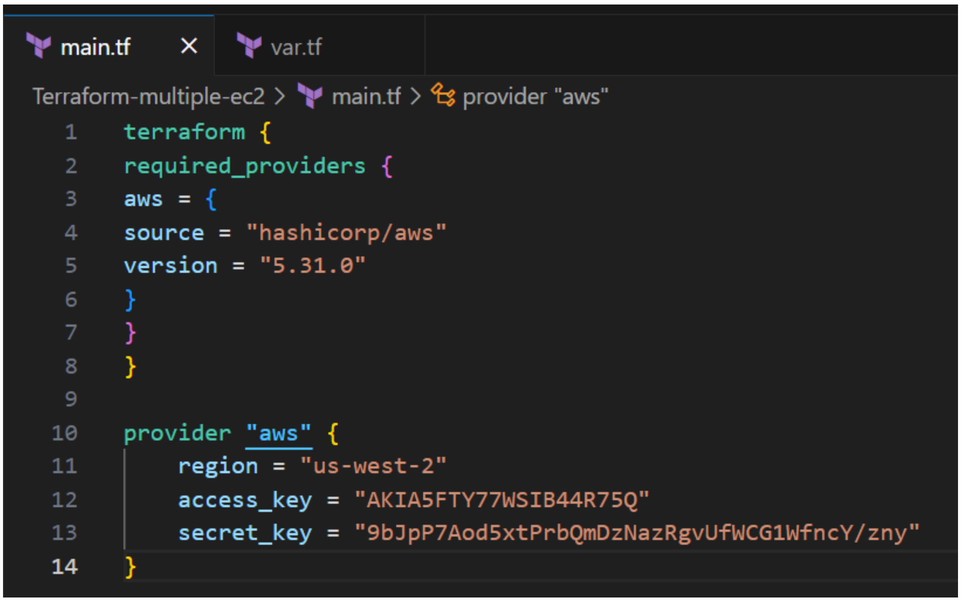
Lab Exercise 9– Creating Multiple EC2 Instances with for_each in Terraform

1. Create a Terraform Directory:

```
mkdir terraform-multiple-ec2
cd terraform-multiple-ec2
```

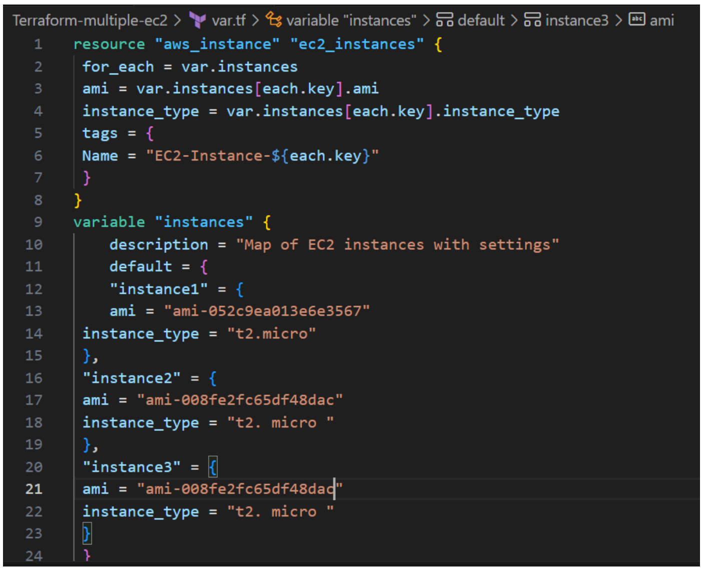
- Create Terraform Configuration Files:
- Create a file named main.tf:

# main.tf



```
terraform {
required_providers {
aws = {
source = "hashicorp/aws"
version = "5.31.0"
}
}
}

provider "aws" {
    region = "us-west-2"
    access_key = "AKIA5FTY77WSIB44R75Q"
    secret_key = "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"
}
```

#var.tf

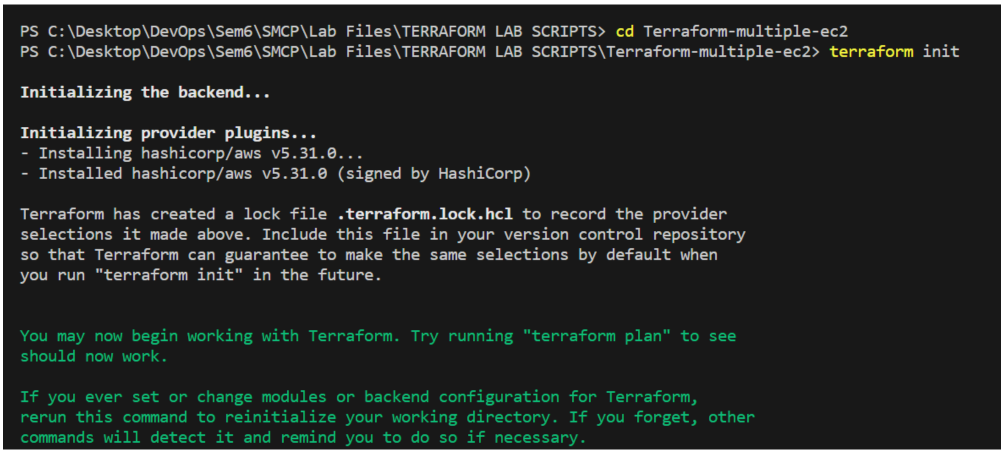


Terraform-multiple-ec2 > var.tf > variable "instances" > default > instance3 > ami

```
resource "aws_instance" "ec2_instances" {
 for_each = var.instances
 ami = var.instances[each.key].ami
 instance_type = var.instances[each.key].instance_type
 tags = {
 Name = "EC2-Instance-${each.key}"
 }
}
variable "instances" {
    description = "Map of EC2 instances with settings"
    default = {
    "instance1" = {
    ami = "ami-052c9ea013e6e3567"
 instance_type = "t2.micro"
 },
 "instance2" = {
 ami = "ami-008fe2fc65df48dac"
 instance_type = "t2. micro "
 },
 "instance3" = {
 ami = "ami-008fe2fc65df48dac"
 instance_type = "t2. micro "
 }
 }
```

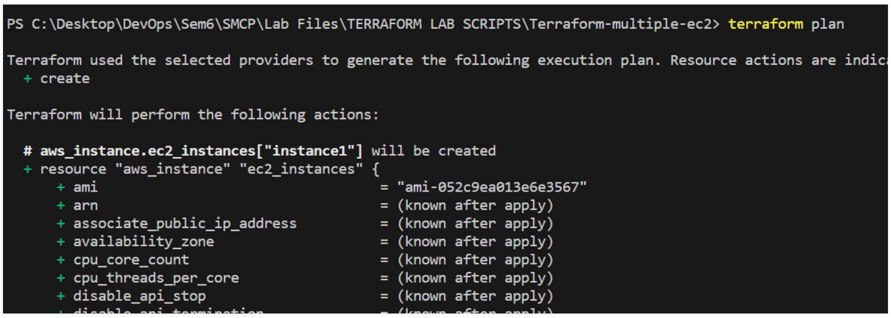
2.Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration:

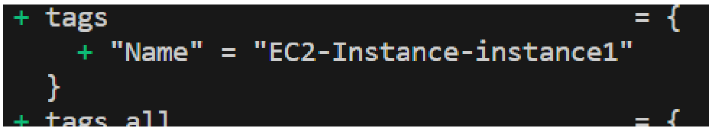


```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS> cd Terraform-multiple-ec2
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-multiple-ec2> terraform init

Initializing the backend...

Initializing provider plugins...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.


You may now begin working with Terraform. Try running "terraform plan" to see
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.
```

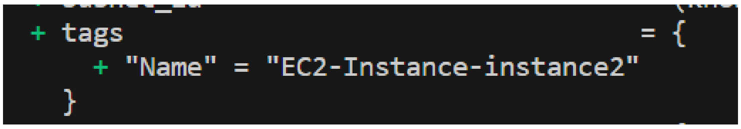
```
terraform init
terraform plan
```



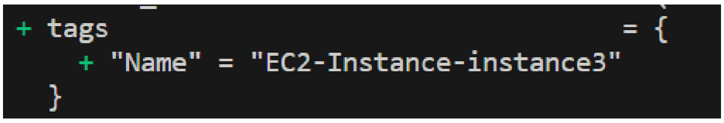
```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-multiple-ec2> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indic
  + create

Terraform will perform the following actions:

  # aws_instance.ec2_instances["instance1"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-052c9ea013e6e3567"
      + 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)
```



```
+ tags                           = {
    + "Name" = "EC2-Instance-instance1"
  }
+ tags_all                       = {
```



```
+ subnet_id                       (kno
+ tags                           = {
    + "Name" = "EC2-Instance-instance2"
  }
```



```
+ tags                           = {
    + "Name" = "EC2-Instance-instance3"
  }
```