Lab Exercise 5– Terraform Variables with Command Line Arguments

Objective:

Learn how to pass values to Terraform variables using command line arguments.

Prerequisites:

- Terraform installed on your machine.
- Basic knowledge of Terraform variables.

Steps:

1. Create a Terraform Directory:

```
mkdir terraform-variablescd

terraform-variables
```

2. Create Terraform Configuration Files:

- Create a file named main.tf:

# main.tf

```
resource "aws_instance" "example" {
    instance_type = var.instance_type
    ami =
var.ami
}

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

- Create a file named variables.tf:

# variables.tf

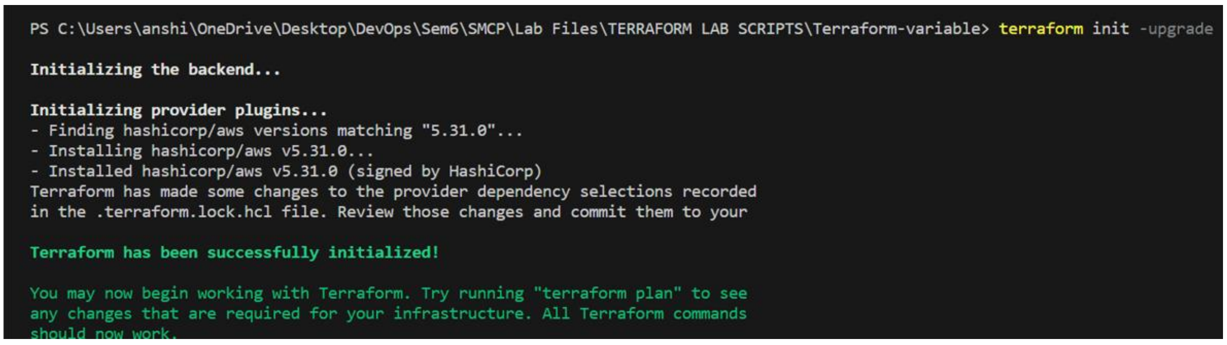
```
variable "region" {
    description = "AWS region"
    default     = "us-west-2"
    }
```

```
variable "ami" {
     description = "AMI ID"
     default = "ami-008fe2fc65df48dac"
}

variable "instance_type" {
     description = "EC2 Instance Type"
     default = "t2.micro"
}
```

3. Use Command Line Arguments:

- Open a terminal and navigate to your Terraform project directory.
- Run the terraform init command:

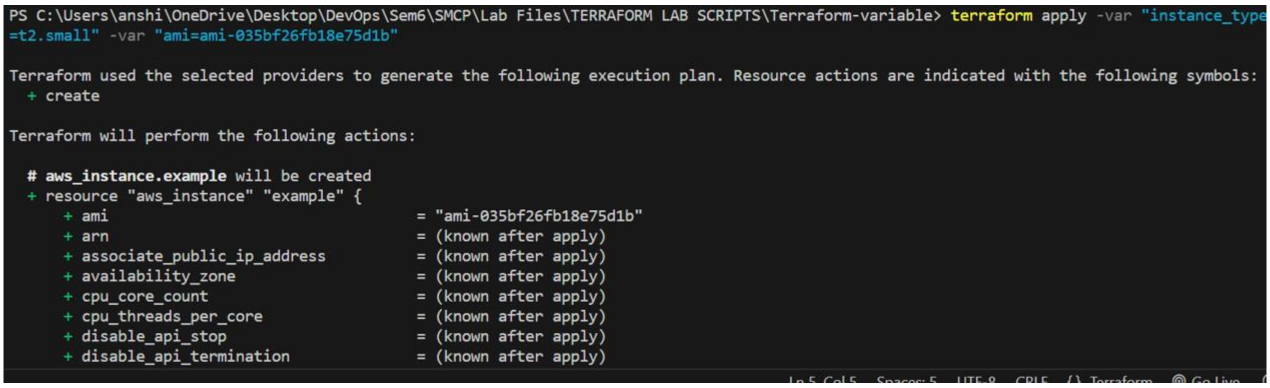
```
terraform init
```



```
PS C:\Users\anshi\OneDrive\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-variable> terraform init -upgrade

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your

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.
```

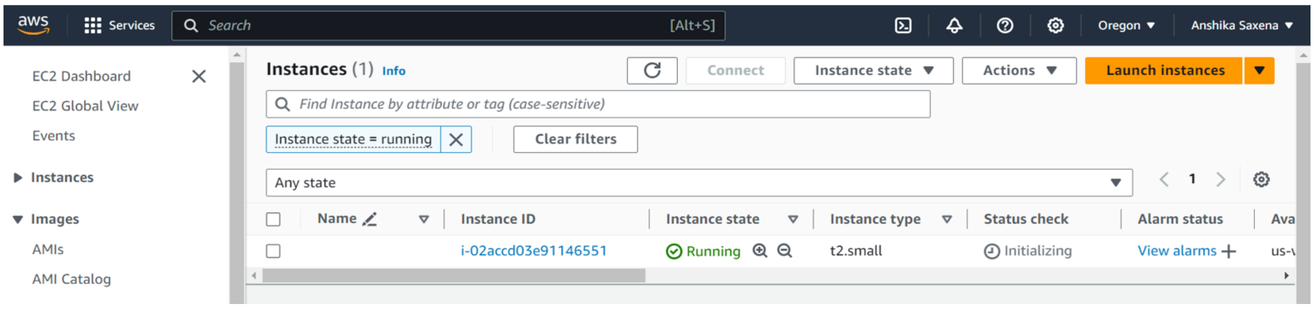
- Run the terraform apply command with command line arguments to set variable values:

```
terraform apply -var 'ami=ami-12345678' -var 'instance_type=t2.small
```



```
PS C:\Users\anshi\OneDrive\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-variable> terraform apply -var "instance_type
=t2.small" -var "ami=ami-035bf26fb18e75d1b"

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.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-035bf26fb18e75d1b"
      + 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)
```

4. Test and Verify:



Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

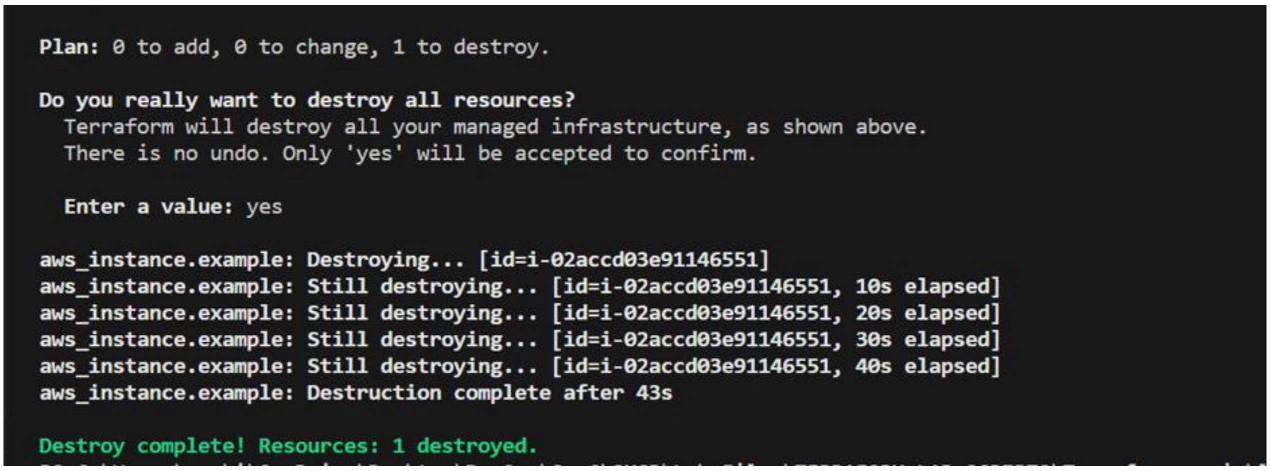
Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
	i-02accd03e91146551	Running	t2.small	Initializing	View alarms +	us-\

5. Clean Up:

After testing, you can clean up resources:

```
terraform destroy
```



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

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.example: Destroying... [id=i-02accd03e91146551]
aws_instance.example: Still destroying... [id=i-02accd03e91146551, 10s elapsed]
aws_instance.example: Still destroying... [id=i-02accd03e91146551, 20s elapsed]
aws_instance.example: Still destroying... [id=i-02accd03e91146551, 30s elapsed]
aws_instance.example: Still destroying... [id=i-02accd03e91146551, 40s elapsed]
aws_instance.example: Destruction complete after 43s

Destroy complete! Resources: 1 destroyed.
```