Lab Exercise 7– 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-cli-variables

cd terraform-cli-variables
```

2. Create Terraform Configuration Files:

- Create a file named main.tf:

instance.tf

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

- Create a file named variables.tf:

variables.tf

```
variable "ami" {
 description = "AMI ID"
 default     = " ami-08718895af4dfa033"
}
```

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

3. Use Command Line Arguments:

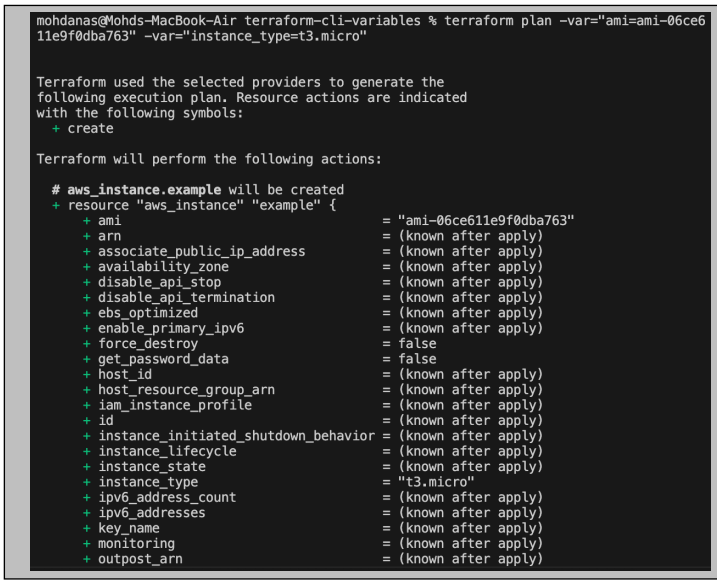
- Open a terminal and navigate to your Terraform project directory.
- Run the terraform init command:

terraform init

```
mohdanas@Mohds-MacBook-Air terraform-cli-variables % terrafor
m init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.11.0...
- Installed hashicorp/aws v6.11.0 (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!
```

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



terraform plan -var=”ami=ami-0522ab6e1ddcc7055” -var=”instance_type=t3.micro”

- Adjust the values based on your preferences.

4. Test and Verify:

- Observe how the command line arguments dynamically set the variable values during the apply process.
- Access the AWS Management Console or use the AWS CLI to verify the creation of resources in the specified region.

5. Clean Up:

After testing, you can clean up resources:

```
terraform destroy
```

Confirm the destruction by typing yes.

6. Conclusion:

This lab exercise demonstrates how to use command line arguments to set variable values dynamically during the terraform apply process. It allows you to customize your Terraform deployments without modifying the configuration files directly. Experiment with different variable values and observe how command line arguments impact the infrastructure provisioning process.