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:

main.tf

```
provider "aws" {
  region = var.region
}

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

Create a file named variables.tf:#

variables.tf

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

variable "ami" {
  description = "AMI ID"
  default = "ami-oc55b159cbfafe1fo"
}

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

3. Use Command Line Arguments:

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

terraform init

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

```
terraform apply -var 'region=us-east-1' -var 'ami=ami-12345678' -var 'instance_type=t2.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.

```
× instance.tf
main.tf
                                          💜 var.tf
 main.tf
         terraform {
         required_providers {
         aws = {
         source = "hashicorp/aws"
         version = "5.31.0"
   8
   9
        provider "aws" [
region = "ap-south-1"
access_key = "AKIAWCHJEOQYJV3PQHM2"
sccert_key = "3mbL8074QL9vqQXa2V270l/r9zl1UeHfnAk7KfaE"
  10
  11
  13
         }
```

```
main.tf

instance.tf

resource "aws_instance" "exp-4"{|
    instance_type = var.instance_typ
    ami = var.ami_id
    count = 1
    tags = {
        Name = "exp4-B3"
    }
}
```

```
var.tf

var.tf

variable "instance_typ" {
 type = string
 }

variable "ami_id" {
 type = string
 default = "ami-00952f27cf14db9cd"
 }

var.tf ×

var.tf
```

```
C:\Users\anu39\Terraform-Script1>terraform init

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 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.
```

```
placement_group
placement_partition_number
primary_network_interface_id
          private_dns
          private_ip
          security_groups
source_dest_check
spot_instance_request_id
                                                         = (known after apply)
= (known after apply)
          tags
+ "Name" = "exp4-B3"
          tags_all
+ "Name" = "exp4-B3"
                                                         = (known after apply)
= (known after apply)
= (known after apply)
= false
          user_data_base64
          user_data_replace_on_change
                                                         = (known after apply)
          vpc_security_group_ids
Plan: 1 to add, 0 to change, 0 to destroy.
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
C:\Users\anu39\Terraform-Script1>terraform apply
  ar.instance_typ
Enter a value: t2.micro
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
```

```
Terraform will perform the following actions:
 # aws_instance.exp-4[0] will be created
  + resource "aws_instance" "exp-4" {
     + ami
                                             = "ami-00952f27cf14db9cd"
                                             = (known after apply)
                                             = (known after apply)
      + associate_public_ip_address
      + availability_zone
                                             = (known after apply)
                                             = (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
     + ebs_optimized
                                             = (known after apply)
     + get_password_data
                                             = false
     + host_id
                                             = (known after apply)
     + host_resource_group_arn
                                            = (known after apply)
                                            = (known after apply)
     + iam_instance_profile
                                             = (known after apply)
     + id
     + instance_initiated_shutdown_behavior = (known after apply)
     + instance_lifecycle
                                             = (known after apply)
     + instance_state
                                             = (known after apply)
     + instance_type
                                             = "t2.micro"
                                             = (known after apply)
     + ipv6_address_count
                                             = (known after apply)
     + ipv6_addresses
                                             = (known after apply)
     + key_name
                                             = (known after apply)
     + monitoring
     + outpost_arn
                                             = (known after apply)
     + password_data
                                             = (known after apply)
                                            = (known after apply)
     + placement_group
                                            = (known after apply)
     + placement_partition_number
                                             = (known after apply)
      + primary_network_interface_id
                                             = (known after apply)
      + private_dns
                                             = (known after apply)
      + private_ip
      + public_dns
                                             = (known after apply)
     + public_ip
                                             = (known after apply)
      + secondary_private_ips
                                             = (known after apply)
                                             = (known after apply)
     + security_groups
      + source_dest_check
                                             = true
     + spot_instance_request_id
                                             = (known after apply)
     + subnet_id
                                             = (known after apply)
```

```
secondary_private_ips
security_groups
source_dest_check
spot_instance_request_id
                                                                         = (known after apply)
= (known after apply)
                                                                        = true
= (known after apply)
= (known after apply)
             subnet id
             tags
+ "Name" = "exp4-B3"
             tags_all
+ "Name" = "exp4-B3"
                                                                        = (known after apply)
= (known after apply)
= (known after apply)
= false
          + tenancy
+ user_data
+ user_data_base64
             user_data_replace_on_change
             vpc_security_group_ids
                                                                         = (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.exp-4[0]: Creating...
aws_instance.exp-4[0]: Still creating... [10s elapsed]
aws_instance.exp-4[0]: Still creating... [20s elapsed]
aws_instance.exp-4[0]: Creation complete after 22s [id=i-0ae4852b180b41165]
 C:\Users\anu39\Terraform-Script1>terraform destroy
 var.instance_typ
Enter a value: t2.micro
 aws_instance.exp-4[0]: Refreshing state... [id=i-0ae4852b180b41165]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
```

```
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.exp-4[0]: Destroying... [id=i-0ae4852b180b41165]

aws_instance.exp-4[0]: Still destroying... [id=i-0ae4852b180b41165, 10s elapsed]

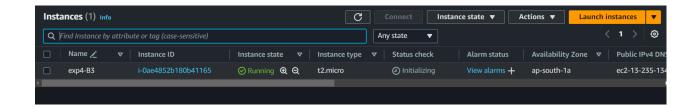
aws_instance.exp-4[0]: Still destroying... [id=i-0ae4852b180b41165, 20s elapsed]

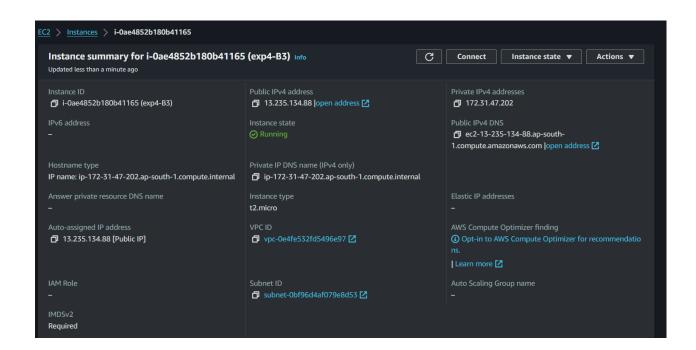
aws_instance.exp-4[0]: Still destroying... [id=i-0ae4852b180b41165, 30s elapsed]

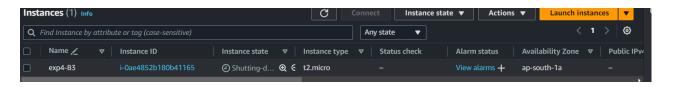
aws_instance.exp-4[0]: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.

C:\Users\anu39\Terraform-Script1>
```







Second Method: -

```
C:\Users\anu39\Terraform-Script1>terraform plan -var "instance_typ=t2.micro"
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
Terraform will perform the following actions:
   # aws_instance.exp-4[0] will be created
+ resource "aws_instance" "exp-4" {
                                                                                                                                "ami-00952f27cf14db9cd"
                                                                                                                           = "ami-00952f27cf14db'
(known after apply)
= false
                     arn
                    arn
associate_public_ip_address
availability_zone
cpu_core_count
cpu_threads_per_core
disable_api_stop
disable_api_termination
ebs_optimized
                                                                                                                                (known after apply)
false
(known after apply)
"t2.micro"
(known after apply)
                     get_password_data
host_id
host_resource_group_arn
iam_instance_profile
                     instance_initiated_shutdown_behavior = instance_lifecycle =
                     instance_state
instance_type
ipv6_address_count
                     ipv6_addresses
key_name
monitoring
                     outpost_arn
                     password data
                     placement_group
placement_partition_number
primary_network_interface_id
private_dns
```