

Lab Exercise 5– Terraform Variables with Command

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 Terraform Directory,

```
PS C:\terraform-scripts> pwd

Path
----
C:\terraform-scripts
```

2. Create Terraform Configuration Files

Main.tf

```
main.tf
2
3   provider "aws" {
4     region = var.region
5     access_key = "AKIAVRUVTDBUGK3JSJB4"
6     secret_key = "7cCnN0wzsabhZZI1y6inat9s4bjmsG/EpHopyrlw"
7   }
8
9   resource "aws_instance" "example" {
10     ami = var.ami
11     instance_type = var.instance_type
12   }
```

Variables.tf

```
variables.tf
1  variable "region" {
2      description = "AWS region"
3      default = "ap-south-1"
4  }
5  variable "ami" {
6      description = "AMI ID"
7      default = "ami-025680a74fd48deb7"
8  }
9  variable "instance_type" {
10     description = "EC2 Instance Type"
11     default = "t2.micro"
12 }
```

3. Use Command Line arguments

terraform init

```
PS C:\terraform-scripts> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.34.0

Terraform has been successfully initialized!
```

Make changes

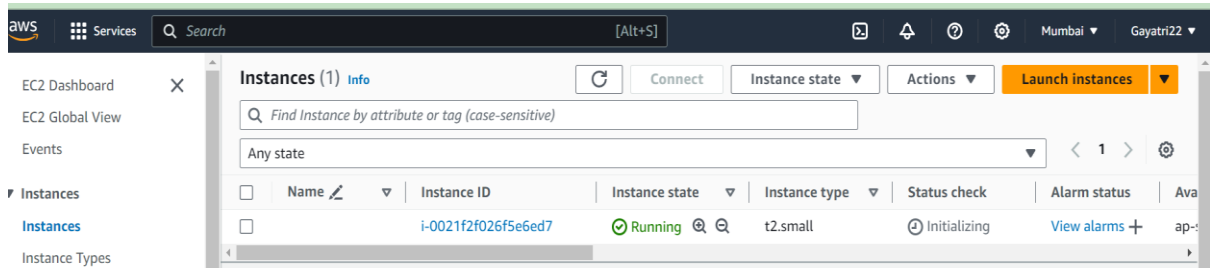
```
PS C:\terraform-scripts> terraform apply -var 'region=ap-south-1' -var 'ami=ami-025680a74fd48deb7' -var 'instance_type=t2.small'
```

```
Enter a value: yes

aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Creation complete after 22s [id=i-0021f2f026f5e6ed7]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

4. Test and apply



5. Clean Up

```
aws_instance.example: Destroying... [id=i-0021f2f026f5e6ed7]
aws_instance.example: Still destroying... [id=i-0021f2f026f5e6ed7, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0021f2f026f5e6ed7, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0021f2f026f5e6ed7, 30s elapsed]
aws_instance.example: Destruction complete after 33s

Destroy complete! Resources: 1 destroyed.
```

Lab Exercise 6– Terraform Multiple tfvars Files

Objective:

Learn how to use multiple tfvars files in Terraform for different environments.

Prerequisites:

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

Steps:

1.Create Terraform Directory,

```
PS C:\terraform-scripts> pwd
```

```
Path
```

```
----
```

```
C:\terraform-scripts
```

2. Create Terraform Configuration Files

Main.tf

```
main.tf
2
3 provider "aws" {
4     region = var.region
5     access_key = "AKIAVRUVTDBUGK3JSJB4"
6     secret_key = "7cCnN0wzsabhZZI1y6inat9s4bjmsG/EpHopyr1w"
7 }
8
9 resource "aws_instance" "example" {
10     ami = var.ami
11     instance_type = var.instance_type
12 }
```

Variables.tf

```
variables.tf
1  variable "region" {
2      description = "AWS region"
3      default = "ap-south-1"
4  }
5  variable "ami" {
6      description = "AMI ID"
7      default = "ami-025680a74fd48deb7"
8  }
9  variable "instance_type" {
10     description = "EC2 Instance Type"
11     default = "t2.micro"
12 }
```

3. Create multiple tfvars files.

4. Initialize and Apply for Dev Environment:

```
dev.tfvars
1
2  ami="ami-025680a74fd48deb7"
3  instance_type="t2.micro"
```

```
terraform plan -help
PS C:\terraform-scripts> terraform init

Initializing the backend...
```

```
commands will detect it and remind you to do so if necessary.
PS C:\terraform-scripts> terraform apply --var-file=dev.tfvars

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create
```

```
aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Still creating... [30s elapsed]
aws_instance.example: Creation complete after 33s [id=i-074d223a082068100]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

5. Initialize and Apply for Prod Environment:

```
prod.tfvars
1
2   ami="ami-025680a74fd48deb7"
3   instance_type="t2.large"
4
```

```
Commands will detect it and remind you to do so if necessary.
PS C:\terraform-scripts> terraform apply --var-file=prod.tfvars
aws_instance.example: Refreshing state... [id=i-074d223a082068100]
```

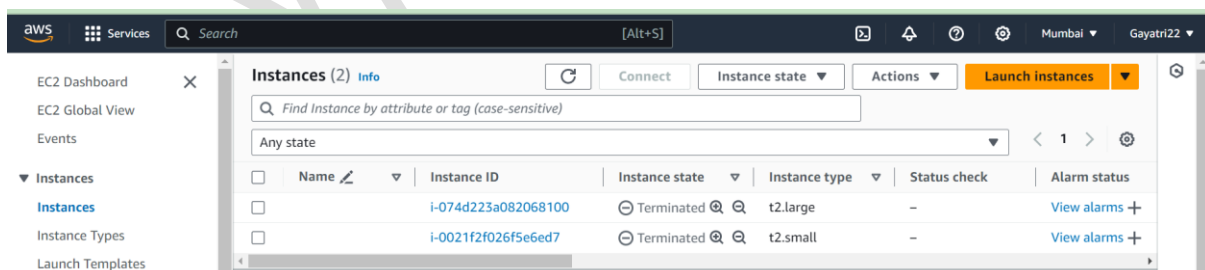
```
aws_instance.example: Modifying... [id=i-074d223a082068100]
aws_instance.example: Still modifying... [id=i-074d223a082068100, 10s elapsed]
aws_instance.example: Still modifying... [id=i-074d223a082068100, 20s elapsed]
aws_instance.example: Still modifying... [id=i-074d223a082068100, 30s elapsed]
aws_instance.example: Still modifying... [id=i-074d223a082068100, 40s elapsed]
aws_instance.example: Still modifying... [id=i-074d223a082068100, 50s elapsed]
aws_instance.example: Modifications complete after 54s [id=i-074d223a082068100]
```

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

```
PS C:\terraform-scripts>
```

6. Clean Up

```
aws_instance.example: Destruction complete after 31s
Destroy complete! Resources: 1 destroyed.
```



The screenshot shows the AWS Management Console interface. On the left, the navigation menu includes 'EC2 Dashboard', 'EC2 Global View', 'Events', and 'Instances'. The 'Instances' section is expanded, showing 'Instances', 'Instance Types', and 'Launch Templates'. The main content area is titled 'Instances (2)' and includes a search bar and a table of instances.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>		i-074d223a082068100	Terminated	t2.large	-	View alarms
<input type="checkbox"/>		i-0021f2f026f5e6ed7	Terminated	t2.small	-	View alarms