SYSTEM PROVISIONING AND CONFIGURATION MANAGEMENT LAB

Lab File
(2023-2024)

for

6th Semester

Submitted To

Dr. Hitesh Kumar Sharma
Cluster Head (Cybernetics)
School of Computer Science

Submitted By:

Arpit Goyal
B. Tech. CSE DevOps [6th Semester]
500094790
R2142210148
B-3

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 a Terraform Directory:
• Create Terraform Configuration Files:
• Create a file named main.tf and Create a file named variables.tf:

main.tf

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

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

variables.tf

```
variable region {
  type        = string
  default     = "ap-south-1"
  description = "AWS Region"
}

variable "ami"{
    type = string
    default = "ami-03f4878755434977f"
    description = "AMI ID"
}

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

2. Create Multiple tfvars Files:
• Create a file named dev.tfvars:

```
dev.tfvars
region= "ap-south-1"
ami= "ami-03f4878755434977f"
instance_type = "t2.micro"
```

Create a file named prod.tfvars:

```
prod.tfvars
region= "us-east-1"
ami= "ami-0c7217cdde317cfec"
instance_type = "t2.micro"
```

3. Initialize and Apply for Dev Environment:

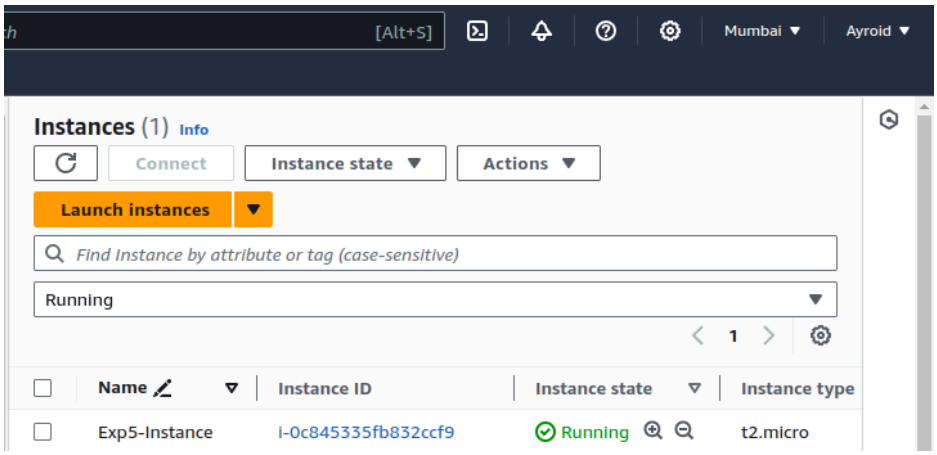
```
→ Exp2  terraform apply -var-file=dev.tfvars

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

Terraform will perform the following actions:

  # aws_instance.Ayroid-ec2[0] will be created
  + resource "aws_instance" "Ayroid-ec2" {
```

4. Initialize and Apply for Prod Environment:

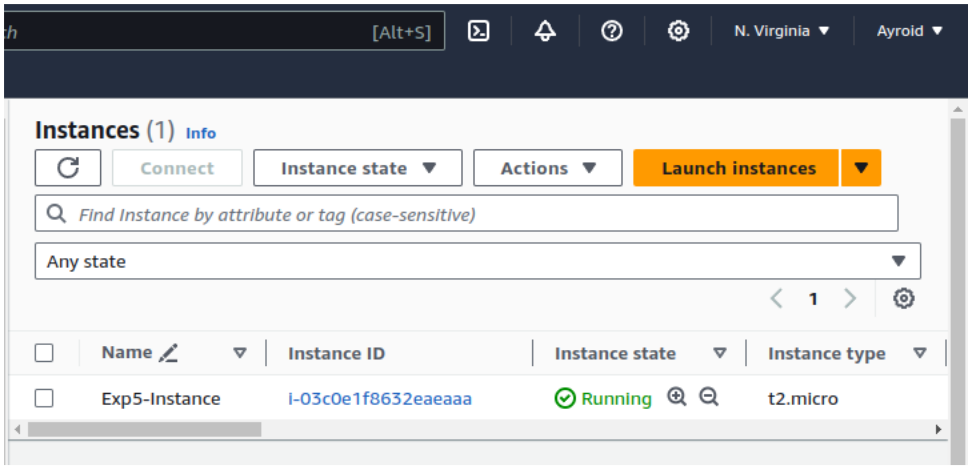
```
→ Exp2  terraform apply -var-file=prod.tfvars
aws_instance.Ayroid-ec2[0]: Refreshing state... [id=i-0c845335fb832ccf9]

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

Terraform will perform the following actions:

  # aws_instance.Ayroid-ec2[0] will be created
  + resource "aws_instance" "Ayroid-ec2" {
```

5. Test and Verify:

a. dev



b. prod



6. Clean up resources in both environments

a. dev

```
→  Exp2  terraform destroy -var-file=dev.tfvars
aws_instance.Ayroid-ec2[0]: Refreshing state... [id=i-0ea2577c5a7fa799a]

Terraform used the selected providers to generate the
following execution plan. Resource actions are indicated
with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.Ayroid-ec2[0] will be destroyed
```

b. prod

```
→  Exp2  terraform destroy -var-file=prod.tfvars
aws_instance.Ayroid-ec2[0]: Refreshing state... [id=i-03c0e1f8632eaeaaa]

Terraform used the selected providers to generate the following execution plan. Resource a
  - destroy

Terraform will perform the following actions:

  # aws_instance.Ayroid-ec2[0] will be destroyed
  - resource "aws_instance" "Ayroid-ec2" {
```