Experiment 6

Terraform Multiple tfvars Files

Aim

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

Steps

1. Create a main file, instance.tf file & variables.tf file for EC2 Instance.

```
instance.tf
    resource "aws_instance" "Ayroid-ec2" {
    instance_type = var.instance_type
    ami = var.ami
    count = 1

tags = {
    Name = "Exp5-Instance"
}
}
```

- 2. Create two tfvars files for different environments
 - a. dev.tfvars

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

b. prod.tfvars

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

- 3. Initialize and provision resources in both environments
 - a. dev

```
→ 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" {
```

b. prod

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

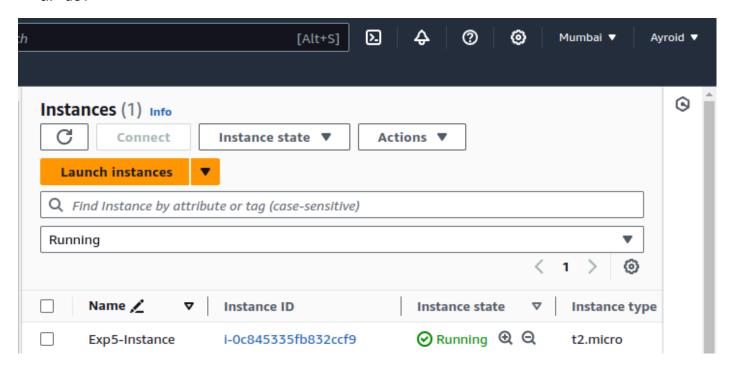
Terraform used the selected providers to generate the following execution plan. Resource actions + create

Terraform will perform the following actions:

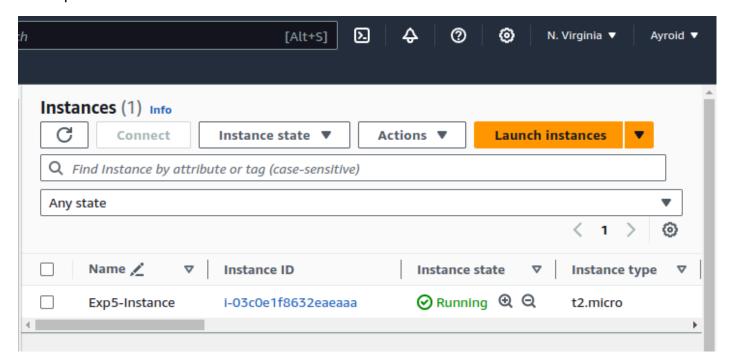
# aws_instance.Ayroid-ec2[0] will be created + resource "aws_instance" "Ayroid-ec2" {
```

4. Verify resources

a. dev



b. prod



5. 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