

## Lab Exercise 9– Creating Multiple EC2 Instances with for\_each in Terraform

### Objective:

Learn how to use for\_each in Terraform to create multiple AWS EC2 instances with specific settings for each instance.

### Prerequisites:

- Terraform installed on your machine.
- AWS CLI configured with the necessary credentials.

### Steps:

#### 1.Create a Terraform Directory:

```
mkdir terraform-ec2-for-each
```

```
cd terraform-ec2-for-each
```

- Create Terraform Configuration Files:
- Create a file named main.tf:

# main.tf

```
terraform {  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
      version = "5.68.0"  
    }  
  }  
}
```

~~provider "aws" {~~

~~access\_key = ""~~

~~secret\_key = ""~~

~~region = "ap-south-1"~~

~~}~~

#Var.tf

```
Var.tf  
1  variable "instances" {  
2    description = "Map of EC2 instances with settings"  
3    default = {  
4      "instance1" = {  
5        ami          = "ami-0ddfba243cbee3768"  
6        instance_type = "t2.micro"  
7      },  
8      "instance2" = {  
9        ami          = "ami-0ddfba243cbee3768"  
10       instance_type = "t2.small"  
11      },  
12     "instance3" = {  
13       ami          = "ami-0ddfba243cbee3768"  
14       instance_type = "t2.large"  
15     }  
16   }  
17 }  
18
```

### #Instance.tf

```
Instance.tf
1  ✓ resource "aws_instance" "ec2_instances" {
2      for_each = var.instances
3      ami      = var.instances[each.key].ami
4      instance_type = var.instances[each.key].instance_type
5  ✓  tags = {
6      Name = "EC2-Instance-${each.key}"
7  }
8  }
9
```

- Replace "your-key-pair-name" and "your-subnet-id" with your actual key pair name and subnet ID.
- In this configuration, we define a variable instances as a map containing settings for each EC2 instance. The aws\_instance resource is then used with for\_each to create instances based on the map.

## 2.Initialize and Apply:

- Run the following Terraform commands to initialize and apply the configuration:

```
terraform init
```

```
terraform apply
```

- Terraform will prompt you to confirm the creation of EC2 instances. Type yes and press Enter.

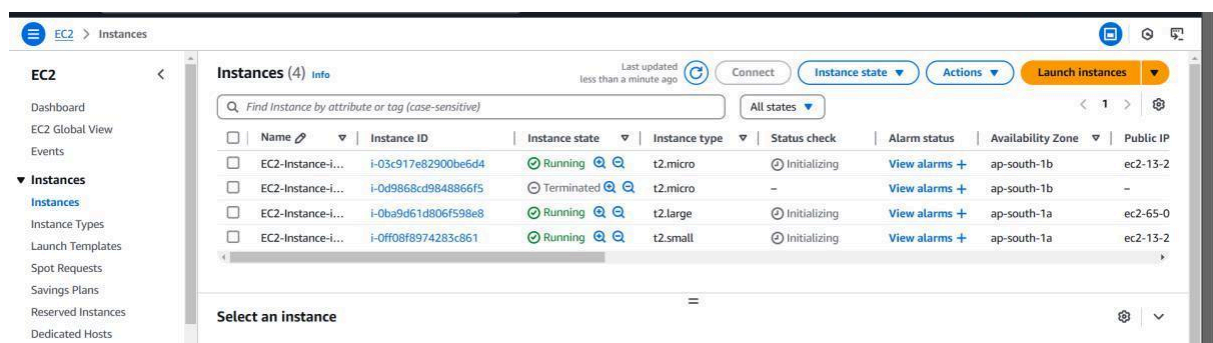
### 3. Verify Instances in AWS Console:

- Log in to the AWS Management Console and navigate to the EC2 service.
- Verify that the specified EC2 instances with the specified names and settings have been created.

### 4. Update Instance Configuration:

- If you want to modify the EC2 instance configuration, update the main.tf file with the desired changes.
- Rerun the terraform apply command to apply the changes:

```
terraform apply
```



### 5. Clean Up:

- After testing, you can clean up the EC2 instances:

```
terraform destroy
```

- Confirm the destruction by typing yes.

### 6. Conclusion:

**This lab exercise demonstrates how to use the `for_each` construct in Terraform to create multiple AWS EC2 instances with specific settings for each instance.**

**The use of a map allows you to define and manage settings for each instance individually. Experiment with different instance types, AMIs, and settings in the main.tf file to observe how Terraform provisions resources based on your configuration.**