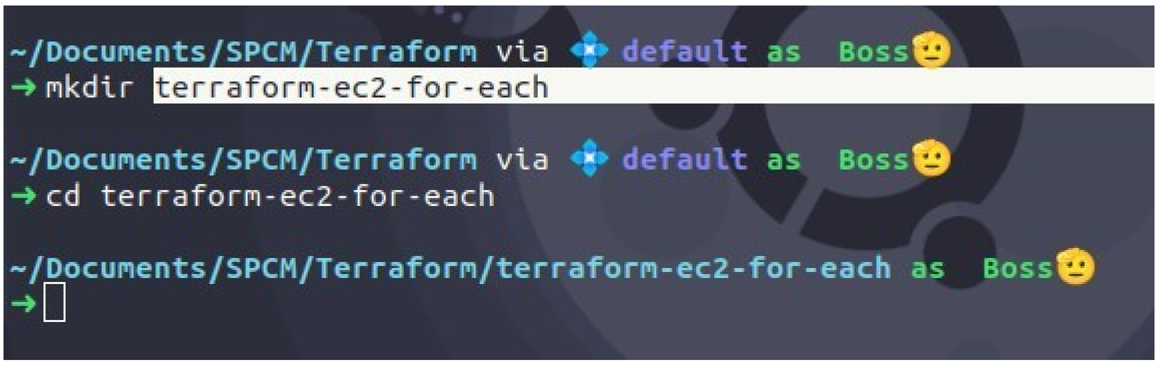
LAB-9

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:



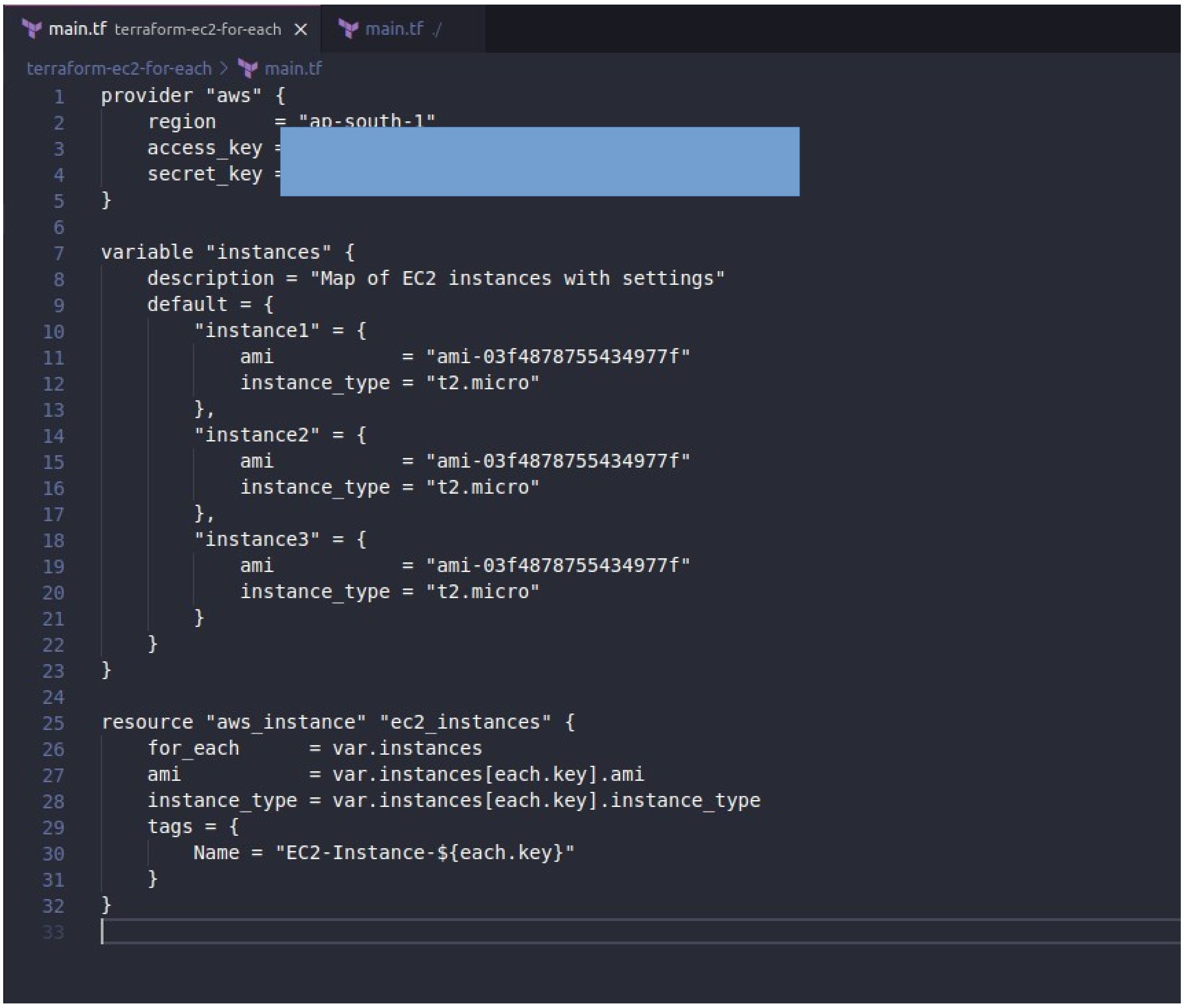
```
~/Documents/SPCM/Terraform via default as  Boss
→ mkdir terraform-ec2-for-each

~/Documents/SPCM/Terraform via default as  Boss
→ cd terraform-ec2-for-each

~/Documents/SPCM/Terraform/terraform-ec2-for-each as  Boss
→
```

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

#main.tf

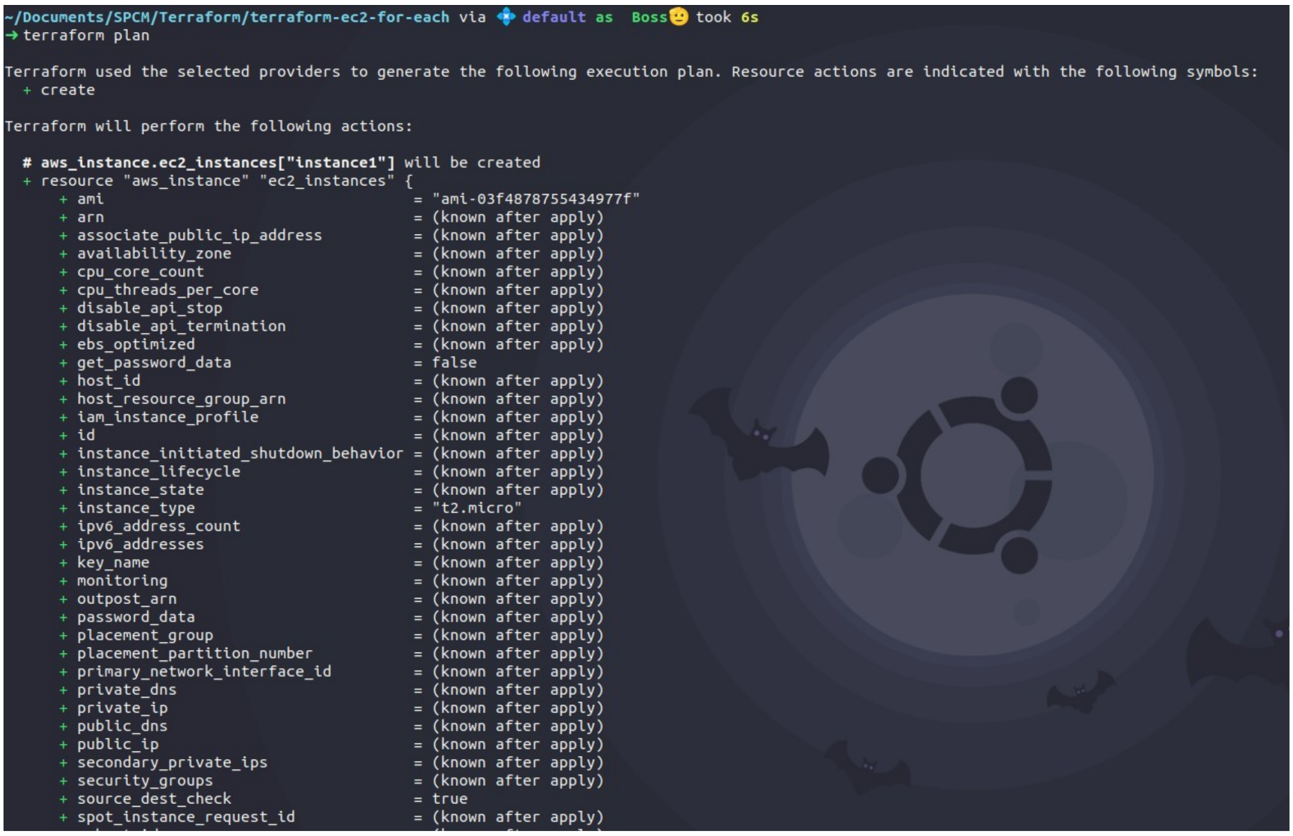

main.tf terraform-ec2-for-each
main.tf ./
terraform-ec2-for-each > main.tf

```
provider "aws" {
    region     = "ap-south-1"
    access_key = [redacted]
    secret_key =
}

variable "instances" {
    description = "Map of EC2 instances with settings"
    default = {
        "instance1" = {
            ami           = "ami-03f4878755434977f"
            instance_type = "t2.micro"
        },
        "instance2" = {
            ami           = "ami-03f4878755434977f"
            instance_type = "t2.micro"
        },
        "instance3" = {
            ami           = "ami-03f4878755434977f"
            instance_type = "t2.micro"
        }
    }
}

resource "aws_instance" "ec2_instances" {
    for_each      = var.instances
    ami           = var.instances[each.key].ami
    instance_type = var.instances[each.key].instance_type
    tags = {
        Name = "EC2-Instance-${each.key}"
    }
}
```

2. Initialize and Apply:
•Run the following Terraform commands to initialize and apply the configuration
`terraform init`
`terraform apply`



```
~/Documents/SPCM/Terraform/terraform-ec2-for-each via 💠 default as  Boss😕 took 6s
➜ terraform plan

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.ec2_instances["instance1"] will be created
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (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              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (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                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
```

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.