ANSHIKA SRIVASTAVA

ROLL NUMBER – R2142220907

SAP ID – 500107049

LAB EXERCISE 9

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.

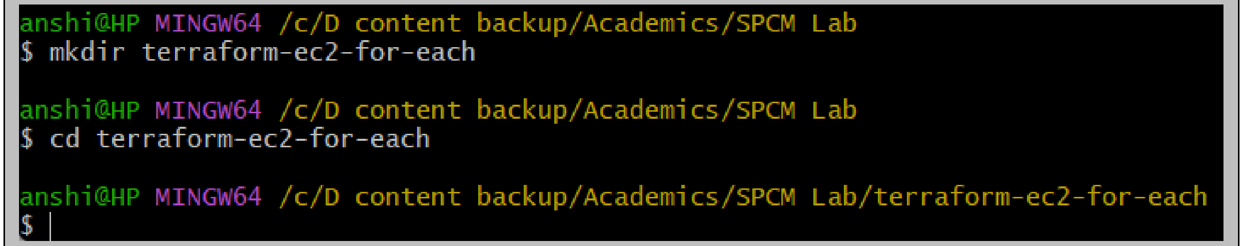
Prerequisites:

- 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
```

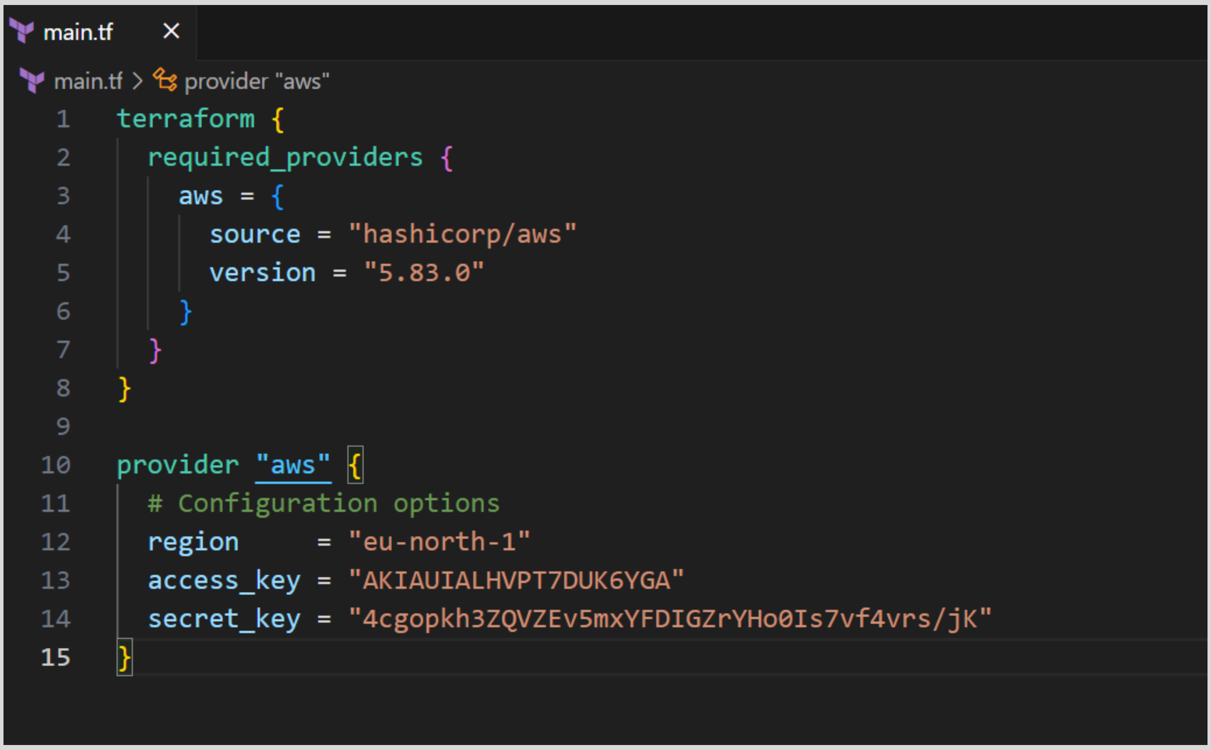


```
anshi@HP MINGW64 /c/D content backup/Academics/SPCM Lab
$ mkdir terraform-ec2-for-each

anshi@HP MINGW64 /c/D content backup/Academics/SPCM Lab
$ cd terraform-ec2-for-each

anshi@HP MINGW64 /c/D content backup/Academics/SPCM Lab/terraform-ec2-for-each
$ |
```

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

# main.tf

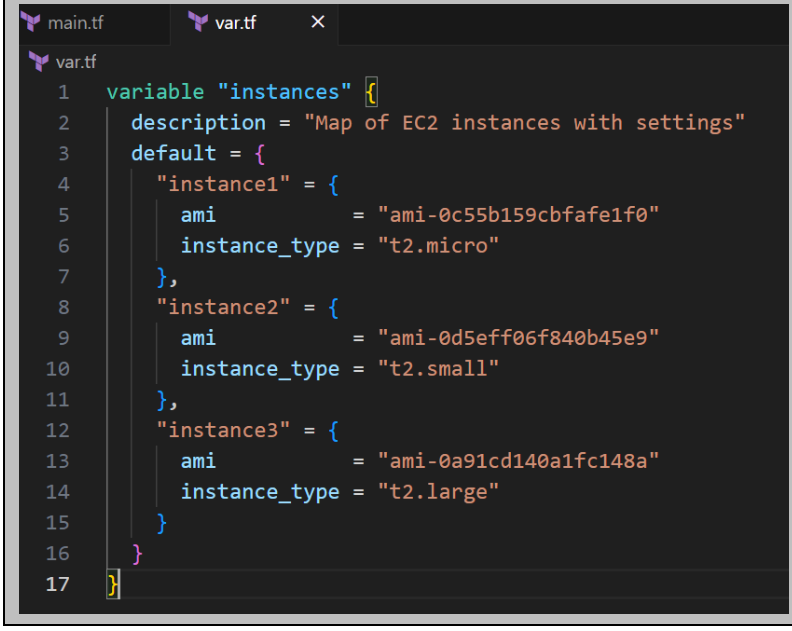
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.68.0"
    }
  }
}

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



```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.83.0"
    }
  }
}

provider "aws" {
  # Configuration options
  region     = "eu-north-1"
  access_key = "AKIAUIALHVPT7DUK6YGA"
  secret_key = "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"
}
```

#Var.tf

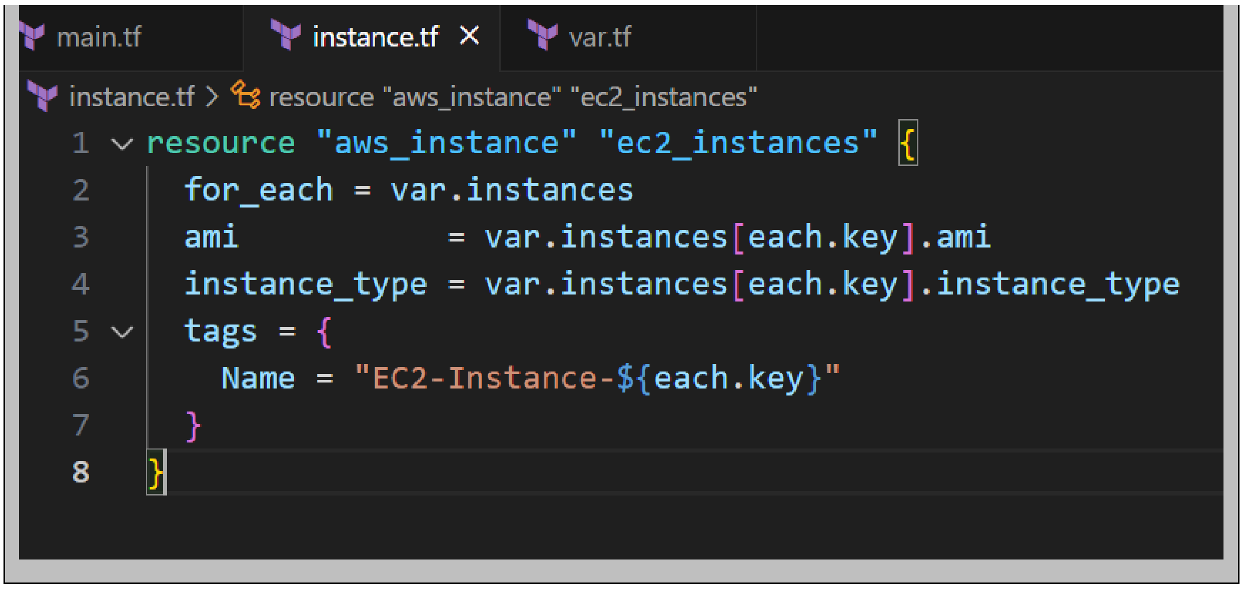
```
variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami         = "ami-0c55b159cbfafe1f0"
    instance_type = "t2.micro"
   },
   "instance2" = {
    ami         = "ami-0123456789abcdef0"
    instance_type = "t2. small "
   },
   "instance3" = {
    ami         = "ami-9876543210fedcba0"
    instance_type = "t2. large "
   }
 }
}
```



```
variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami           = "ami-0c55b159cbfafe1f0"
      instance_type = "t2.micro"
    },
    "instance2" = {
      ami           = "ami-0d5eff06f840b45e9"
      instance_type = "t2.small"
    },
    "instance3" = {
      ami           = "ami-0a91cd140a1fc148a"
      instance_type = "t2.large"
    }
  }
}
```

#Instance.tf

```
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}"
  }
}
```



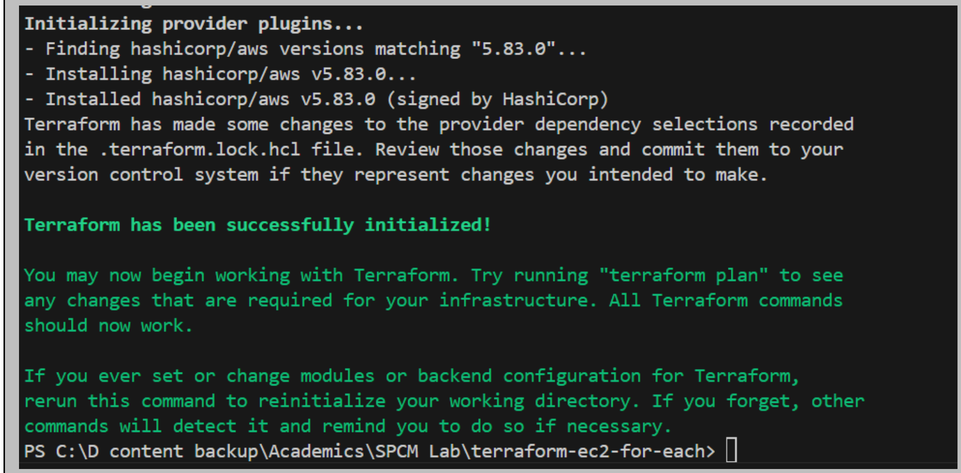
```
resource "aws_instance" "ec2_instances" {
  for_each = var.instances
  ami           = var.instances[each.key].ami
  instance_type = var.instances[each.key].instance_type
  tags = {
    Name = "EC2-Instance-${each.key}"
  }
}
```

- 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:

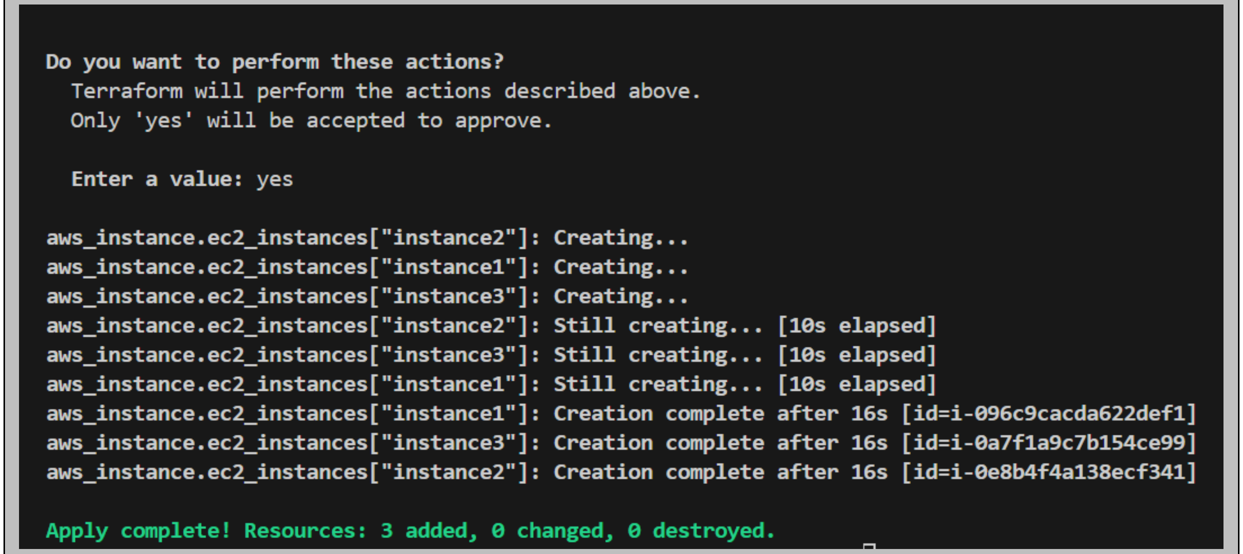
- Run the following Terraform commands to initialize and apply the configuration:

terraform init



```
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.83.0"...
- Installing hashicorp/aws v5.83.0...
- Installed hashicorp/aws v5.83.0 (signed by HashiCorp)
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\D content backup\Academics\SPCM Lab\terraform-ec2-for-each>
```

terraform apply

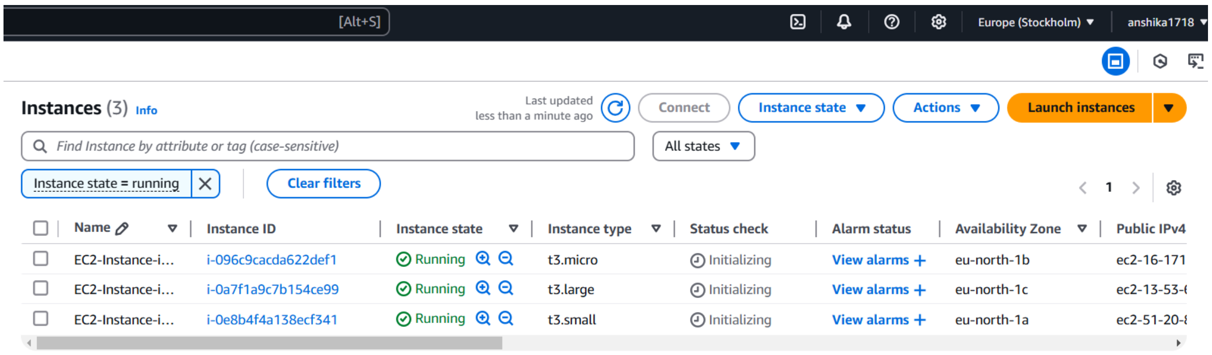


```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.ec2_instances["instance2"]: Creating...
aws_instance.ec2_instances["instance1"]: Creating...
aws_instance.ec2_instances["instance3"]: Creating...
aws_instance.ec2_instances["instance2"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance1"]: Creation complete after 16s [id=i-096c9cacda622def1]
aws_instance.ec2_instances["instance3"]: Creation complete after 16s [id=i-0a7f1a9c7b154ce99]
aws_instance.ec2_instances["instance2"]: Creation complete after 16s [id=i-0e8b4f4a138ecf341]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

- 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.



Instances (3)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
EC2-Instance-i...	i-096c9cacda622def1	Running	t3.micro	Initializing	View alarms +	eu-north-1b	ec2-16-171
EC2-Instance-i...	i-0a7f1a9c7b154ce99	Running	t3.large	Initializing	View alarms +	eu-north-1c	ec2-13-53-
EC2-Instance-i...	i-0e8b4f4a138ecf341	Running	t3.small	Initializing	View alarms +	eu-north-1a	ec2-51-20-

4. Update Instance Configuration:

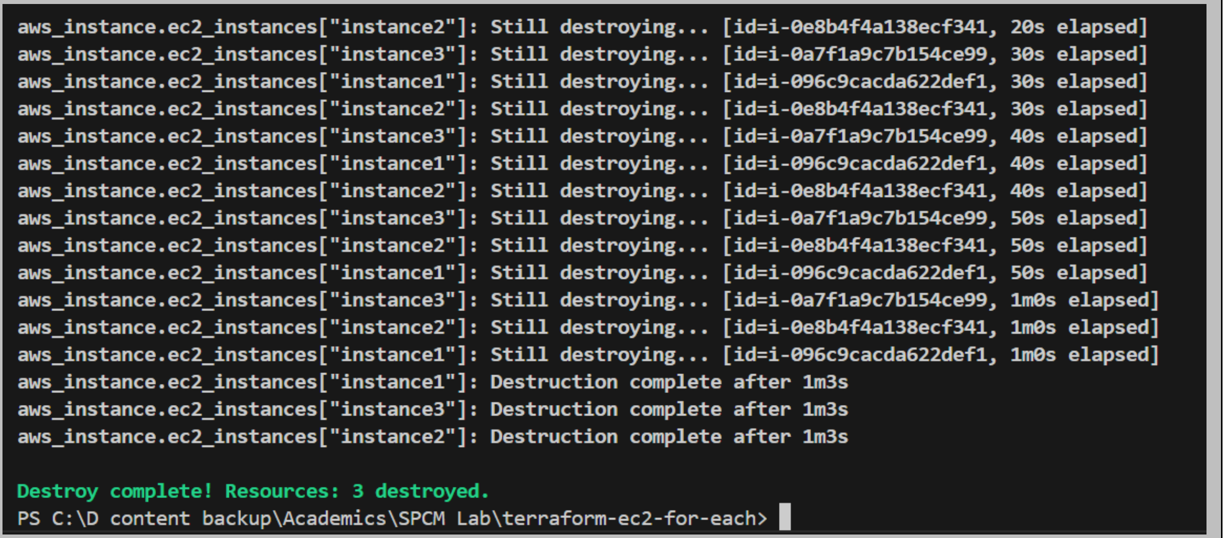
- If you want to modify the EC2 instance configuration, update the main.tf file with the desired changes.

- Rerun the terraform apply command to apply the changes:

5. Clean Up:

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

terraform destroy



```
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0e8b4f4a138ecf341, 20s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0a7f1a9c7b154ce99, 30s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-096c9cacda622def1, 30s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0e8b4f4a138ecf341, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0a7f1a9c7b154ce99, 40s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-096c9cacda622def1, 40s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0e8b4f4a138ecf341, 40s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0a7f1a9c7b154ce99, 50s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0e8b4f4a138ecf341, 50s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-096c9cacda622def1, 50s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-0a7f1a9c7b154ce99, 1m0s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-0e8b4f4a138ecf341, 1m0s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-096c9cacda622def1, 1m0s elapsed]
aws_instance.ec2_instances["instance1"]: Destruction complete after 1m3s
aws_instance.ec2_instances["instance3"]: Destruction complete after 1m3s
aws_instance.ec2_instances["instance2"]: Destruction complete after 1m3s

Destroy complete! Resources: 3 destroyed.
PS C:\D content backup\Academics\SPCM Lab\terraform-ec2-for-each>
```

- 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.