Lab-9 : Creating Multiple EC2 instances with for_each in terraform.

1. Create Terraform directory.

```
➜  ~ mkdir terraform-ec2-for-each
➜  ~ cd terraform-ec2-for-each
➜  terraform-ec2-for-each
```

2. Create terraform configuration file (main.tf) :\

```
provider "aws" {
  region = "us-east-1"
  access_key = "AKIA232UVZYDK5TANG62"
  secret_key = "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"
}

variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1"={
      ami = "ami-0440d3b780d96b29d"
      instance_type = "t2.micro"
    } ,
    "instance2"={
      ami= "ami-0c7217cdde317cfec"
      instance_type="t2.micro"
    },
    "instance3" = {
      ami = "ami-0fe630eb857a6ec83"
      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-${each.key}"
  }

}
```

3. Initialize, validate and Apply :

```
● ➜  Lab-9 terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.37.0...
- Installed hashicorp/aws v5.37.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
```

```
● ➜  Lab-9 terraform validate
Success! The configuration is valid.

○ ➜  Lab-9
```

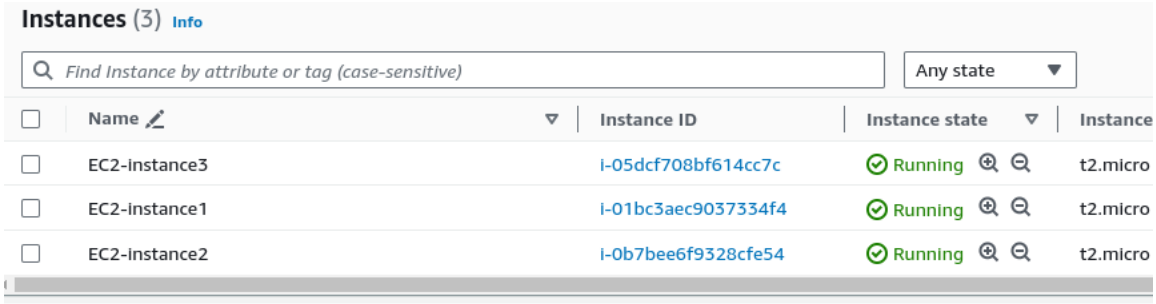
```
→  Lab-9 terraform apply -auto-approve

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-0440d3b780d96b29d"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
```

4. Verify Users in AWS console :



5. Clean up Resources (terraform destroy) :

```
→  Lab-9 terraform destroy
aws_instance.ec2_instances["instance1"]: Refreshing state... [id=i-01bc3aec9037334f4]
aws_instance.ec2_instances["instance3"]: Refreshing state... [id=i-05dcf708bf614cc7c]
aws_instance.ec2_instances["instance2"]: Refreshing state... [id=i-0b7bee6f9328cfe54]

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.ec2_instances["instance1"] will be destroyed
  - resource "aws_instance" "ec2_instances" {
      - ami                                  = "ami-0440d3b780d96b29d" -> null
```