Experiment 9

1. Create a terraform directory

```
singhthatcodes@JAPJEETs-MBP terraform-vpc % mkdir terraform-ec2-for-each
cd terraform-ec2-for-each
singhthatcodes@JAPJEETs-MBP terraform-ec2-for-each %
```

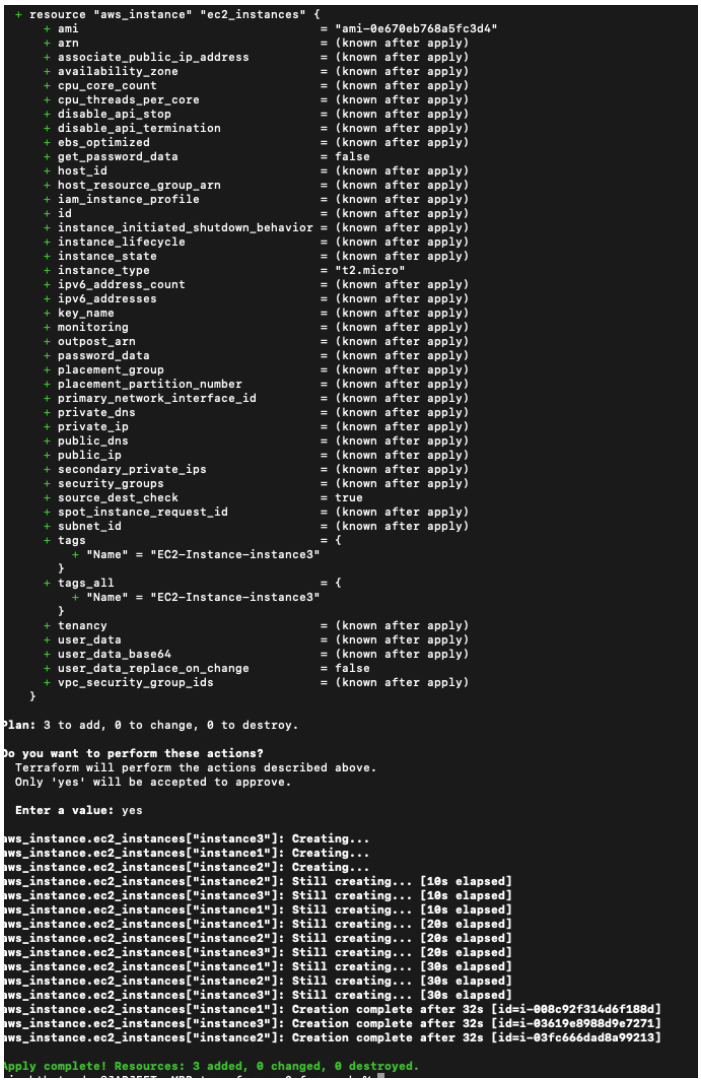
2. Config files :

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

variable "instances" {
  description = "Map of EC2 instances with settings"
  default = {
    "instance1" = {
      ami           = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance2" = {
      ami           = "ami-0e670eb768a5fc3d4"
      instance_type = "t2.micro"
    },
    "instance3" = {
      ami           = "ami-0e670eb768a5fc3d4"
      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}"
  }
}
```

3. Terrafrom init

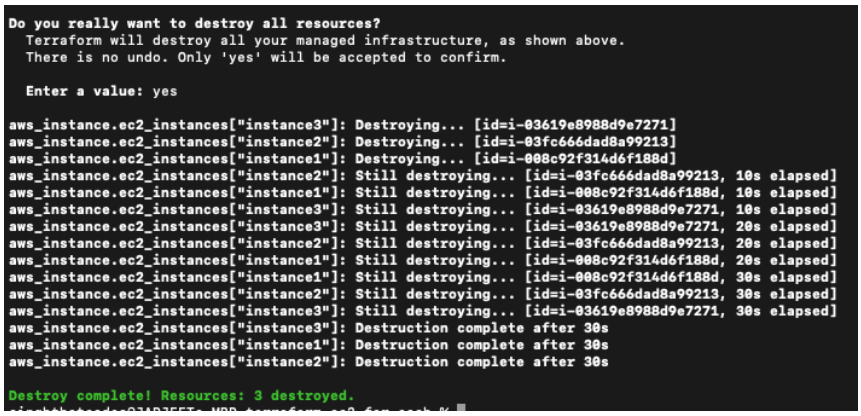
```
singhthatcodes@JAPJEETs-MBP terraform-ec2-for-each % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.38.0...
- Installed hashicorp/aws v5.38.0 (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
you run "terraform init" in the future.

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.
singhthatcodes@JAPJEETs-MBP terraform-ec2-for-each %
```

4. Terraform apply :



```
  + resource "aws_instance" "ec2_instances" {
      + ami                                  = "ami-0e670eb768a5fc3d4"
      + 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)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-Instance-instance3"
        }
      + tags_all                             = {
          + "Name" = "EC2-Instance-instance3"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

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["instance3"]: Creating...
aws_instance.ec2_instances["instance1"]: Creating...
aws_instance.ec2_instances["instance2"]: 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"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance1"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance1"]: Creation complete after 32s [id=i-008c92f314d6f188d]
aws_instance.ec2_instances["instance3"]: Creation complete after 32s [id=i-03619e8988d9e7271]
aws_instance.ec2_instances["instance2"]: Creation complete after 32s [id=i-03fc666dad8a99213]

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

5. Terraform destroy :



```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.ec2_instances["instance3"]: Destroying... [id=i-03619e8988d9e7271]
aws_instance.ec2_instances["instance2"]: Destroying... [id=i-03fc666dad8a99213]
aws_instance.ec2_instances["instance1"]: Destroying... [id=i-008c92f314d6f188d]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-03fc666dad8a99213, 10s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-008c92f314d6f188d, 10s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-03619e8988d9e7271, 10s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-03619e8988d9e7271, 20s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-03fc666dad8a99213, 20s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-008c92f314d6f188d, 20s elapsed]
aws_instance.ec2_instances["instance1"]: Still destroying... [id=i-008c92f314d6f188d, 30s elapsed]
aws_instance.ec2_instances["instance2"]: Still destroying... [id=i-03fc666dad8a99213, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Still destroying... [id=i-03619e8988d9e7271, 30s elapsed]
aws_instance.ec2_instances["instance3"]: Destruction complete after 30s
aws_instance.ec2_instances["instance1"]: Destruction complete after 30s
aws_instance.ec2_instances["instance2"]: Destruction complete after 30s

Destroy complete! Resources: 3 destroyed.
```