

Experiment 8

1. Create a Terraform Directory:

```
singhthatcodes@JAPJEETs-MacBook-Pro spcm % mkdir terraform-vpc && cd terraform-vpc
singhthatcodes@JAPJEETs-MacBook-Pro terraform-vpc %
```

2. Create Config files :

```
provider "aws" {
  region = "ap-south-1"
  access_key = "AKIAVU2WWA23XD2BCN6J"
  secret_key = "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"
}
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true
  tags = {
    Name = "MyVPC"
  }
}
resource "aws_subnet" "my_subnet" {
  count = 2
  vpc_id = aws_vpc.my_vpc.id
  cidr_block = "10.0.${count.index + 1}.0/24"
  availability_zone = "ap-south-1a"
  map_public_ip_on_launch = true
  tags = {
    Name = "MySubnet-${count.index + 1}"
  }
}
```

3. terraform init

```
singhthatcodes@JAPJEETs-MacBook-Pro terraform-vpc % 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-MacBook-Pro terraform-vpc %
```

4. terraform apply

```
singhatcodes@JAPJEETs-MBP terraform-vpc % terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-08ecd654fbc2c8937]

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_subnet.my_subnet[0] will be created
+ resource "aws_subnet" "my_subnet" {
  arn                                     = (known after apply)
  assign_ipv6_address_on_creation        = false
  availability_zone                      = "ap-south-1a"
  availability_zone_id                   = (known after apply)
  cidr_block                             = "10.0.1.0/24"
  enable_dns64                           = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id                                     = (known after apply)
  ipv6_cidr_block_association_id         = (known after apply)
  ipv6_native                            = false
  map_public_ip_on_launch                = true
  owner_id                               = (known after apply)
  private_dns_hostnames_type_on_launch   = (known after apply)
  tags                                   = {
    "Name" = "MySubnet-1"
  }
  tags_all                               = {
    "Name" = "MySubnet-1"
  }
  vpc_id                                 = "vpc-08ecd654fbc2c8937"
}

# aws_subnet.my_subnet[1] will be created
+ resource "aws_subnet" "my_subnet" {
  arn                                     = (known after apply)
  assign_ipv6_address_on_creation        = false
  availability_zone                      = "ap-south-1a"
  availability_zone_id                   = (known after apply)
  cidr_block                             = "10.0.2.0/24"
  enable_dns64                           = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id                                     = (known after apply)
  ipv6_cidr_block_association_id         = (known after apply)
  ipv6_native                            = false
  map_public_ip_on_launch                = true
  owner_id                               = (known after apply)
  private_dns_hostnames_type_on_launch   = (known after apply)
  tags                                   = {
    "Name" = "MySubnet-2"
  }
  tags_all                               = {
    "Name" = "MySubnet-2"
  }
  vpc_id                                 = "vpc-08ecd654fbc2c8937"
}

Plan: 2 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_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Still creating... [18s elapsed]
aws_subnet.my_subnet[1]: Creation complete after 11s [id=subnet-0b330b422ade9a3a]
aws_subnet.my_subnet[0]: Creation complete after 12s [id=subnet-09086f051651bfba7]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
singhatcodes@JAPJEETs-MBP terraform-vpc %
```

5. Terraform destroy :

```
Plan: 0 to add, 0 to change, 3 to 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_subnet.my_subnet[1]: Destroying... [id=subnet-0b330b422ade9a3a]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-09086f051651bfba7]
aws_subnet.my_subnet[0]: Destruction complete after 1s
aws_subnet.my_subnet[1]: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-08ecd654fbc2c8937]
aws_vpc.my_vpc: Destruction complete after 0s

Destroy complete! Resources: 3 destroyed.
singhatcodes@JAPJEETs-MBP terraform-vpc %
```