1) Creating Instances using Terraform

```
provider "aws" {
   access_key = "xxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxx"
   region = "us-east-2"
}

resource "aws_instance" "Instance-name" {
 count        = 3
 ami          = "ami-019f9b3318b7155c5"
 instance_type = "t2.micro"

 tags = {
  Name = "Instance-name"
 }
}
```

2) Creating vpc and subnets instances

```
provider "aws" {
   access_key = "xxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxx"
   region = "us-east-2"
}

resource "aws_vpc" "myvpc1" {
 cidr_block      = "10.0.0.0/24"

 tags = {
  Name = "myvpc1"
 }
}
resource "aws_internet_gateway" "myigw1" {
 vpc_id = aws_vpc.myvpc1.id

 tags = {
  Name = "myigw"
 }
}
resource "aws_internet_gateway_attachment" "myigwvpc" {
 internet_gateway_id = aws_internet_gateway.myigw1.id
 vpc_id              = aws_vpc.myvpc1.id
}
resource "aws_subnet" "mysn1" {
 vpc_id     = aws_vpc.myvpc1.id
 cidr_block = "10.0.0.0/25"
```

```
  tags = {
    Name = "mysn1terra"
  }
}
resource "aws_subnet" "mysn2" {
  vpc_id     = aws_vpc.myvpc1.id
  cidr_block = "10.0.0.128/25"

  tags = {
    Name = "mysn2terra"
  }
}
resource "aws_eip" "lb" {
  vpc = "true"
}
resource "aws_nat_gateway" "mynat1" {
  allocation_id = aws_eip.lb.id
  subnet_id     = aws_subnet.mysn1.id

  tags = {
    Name = "terraNAT"
  }
}
resource "aws_route_table" "myrt1" {
  vpc_id = aws_vpc.myvpc1.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myigw1.id
  }

}
resource "aws_route_table" "myrt2" {
  vpc_id = aws_vpc.myvpc1.id

  route {
    cidr_block = "0.0.0.0/0"
    nat_gateway_id = aws_nat_gateway.mynat1.id
  }
}

resource "aws_route_table_association" "myrt1sn1" {
  subnet_id      = aws_subnet.mysn1.id
  route_table_id = aws_route_table.myrt1.id
}

resource "aws_route_table_association" "myrt1sn2" {
  subnet_id      = aws_subnet.mysn2.id
  route_table_id = aws_route_table.myrt2.id
```

```
}
resource "aws_security_group" "mysg1" {
  name        = "allow ssh"
  description = "Allow TLS inbound traffic and all outbound traffic"
  vpc_id      = aws_vpc.myvpc1.id
  ingress {
    from_port       = 22
    to_port         = 22
    protocol        = "tcp"
    cidr_blocks     = ["0.0.0.0/0"]
   }
  tags = {
    Name = "mysg1terra"
  }
}


resource "aws_instance" "myec2" {
  ami           = "ami-019f9b3318b7155c5"
  instance_type = "t2.micro"
  subnet_id = aws_subnet.mysn1.id
  key_name = "kops"
  vpc_security_group_ids = aws_security_group.mysg1.id
  tags = {
    Name = "myterraec2"
  }
}
resource "aws_eip" "lb1" {
  instance = aws_instance.myec2.id
  vpc = "true"
}
resource "aws_ebs_volume" "myvol1" {
  availability_zone = "us-east-2a"
  size              = 7

  tags = {
    Name = "terravol"
  }
}
resource "aws_volume_attachment" "myec2vol1" {
device_name = "/dev/sdh"
volume_id = aws_ebs_volume.myvol1.id
instance_id = aws_instance.myec2.id

}
```

Variable's

3) Creating instances with For_each Variables

```
provider "aws" {
   access_key = "xxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxx"
   region = "us-east-2"
}



      resource "aws_instance" "vm" {
       for_each = {
         "vm1" = { t2.micro }
         "vm2" = { t2.small }
         "vm3" = {t3.micro}



       }
       ami              = "ami-019f9b3318b7155c5"
       instance_type      = each.value.
       availability_zone  = us-east-2
       tags = {
         Name = each.key
       }
      }
```

4) Creating vpc, subnets and instances with variables

```
variable "region" {
type = string
default = "us-east-2"
}

variable "access" {
type = string
default = "xxxxxxxxxxxxx"
}

variable "secret" {
type = string
default = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}

variable "ami" {
type = string
default = "ami-019f9b3318b7155c5"
}
```

```
variable "inst" {
type = string
default = "t2.micro"
}

variable "subent" {
type = list
default = [ "subnet-08107f14984331d0a", "subnet-0e7ae1f819143a821", "subnet-0bdc35d3bd1c53f58" ]
}

variable "sg" {
type = list
default = ["sg-00cd73ddfa32b77e1"]
}
```

➔ Creating instance using instance

```
provider "aws" {
   access_key = var.access
   secret_key = var. secret
   region = var.region

}
resource "aws_vpc" "main" {
 cidr_block = "${var.vpc_cidr}"
 instance_tenancy = "${var.tenancy}"

 tags = {
  Name = "main"
 }
}
resource "aws_subnet" "main" {
 vpc_id = "${aws_vpc.main.id}"
 cidr_block = "${var.subnet_cidr}"

 tags = {
  Name = "main"
 }
}
```

Modules

➔ **Use case**

➔ **Create 3 folders Dev,Prod,Module**

→Dev

→Prod

→Module

→ under Module create Ec2 and Vpc folders

→Ec2

→Vpc

➔ **Creating file with variables creating instance in Module/Ec2 folder**

```
variable "ec2_count" {
   default = "3"
}
variable "ami_id" {}

variable "instance_type" {
   default = "t2.micro"
}
```

➔ **Creating file for creating instance in Module/Ec2 folder using variables**

```
provider "aws" {
   access_key = "xxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxx"
   region = "us-east-2"
}

resource "aws_instance" "web1" {
 count        = var.ec2_count
 ami          = var.ami_id
 instance_type = var.instance_type
 subnet_id     = var.subnet_id
 tags = {
  Name = "myec2-$(count.index+1)"
 }
}
```

➔ **Creating file with variables. creating VPC in Module/VPC folder**

```
provider "aws" {
   region = "us-east-2"
   access_key = "xxxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
resource "aws_vpc" "main" {
 cidr_block = "${var.vpc_cidr}"
 instance_tenancy = "${var.tenancy}"

 tags = {
  Name = "main"
 }
}
resource "aws_subnet" "main" {
 vpc_id = "${aws_vpc.main.id}"
 cidr_block = "${var.subnet_cidr}"

 tags = {
  Name = "main"
 }
}
```

➔ **Creating file using variables. creating VPC in Module/VPC folder**

```
provider "aws" {
   region = "us-east-2"
   access_key = "xxxxxxxxxxxxxxxx"
   secret_key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
resource "aws_vpc" "main" {
 cidr_block = "${var.vpc_cidr}"
 instance_tenancy = "${var.tenancy}"

 tags = {
  Name = "main"
 }
}
resource "aws_subnet" "main" {
 vpc_id = "${aws_vpc.main.id}"
 cidr_block = "${var.subnet_cidr}"

 tags = {
  Name = "main"
 }
}
```

➔ Creating Dev environment using Module file

```
 module "my-vpc" {
   source = "../Modules/vpc"
   vpc_cidr = "192.168.0.0/16"
   tenancy = "default"
   vpc_id = "${module.my_vpc.vpc_id}"
   subnet_cidr = "192.168.1.0/24"

}
module "my-ec2" {
   source = "../Modules/ec2"
   count = 100000
   instance_type = "t2.large"
   sudnet_id = "${module.my_vpc.subnet_id}"
   ami_id = ""

}
```

➔ Creating Prod environment using Module file

```
module "my-vpc" {
   source = "../Modules/vpc"
   vpc_cidr = "192.168.0.0/16"
   tenancy = "default"
   vpc_id = "${module.my_vpc.vpc_id}"
   subnet_cidr = "192.168.1.0/24"

}
module "my-ec2" {
   source = "../Modules/ec2"
   count = 200000
   instance_type = "t3.large"
   sudnet_id = "${module.my_vpc.subnet_id}"
   ami_id = ""

}
```