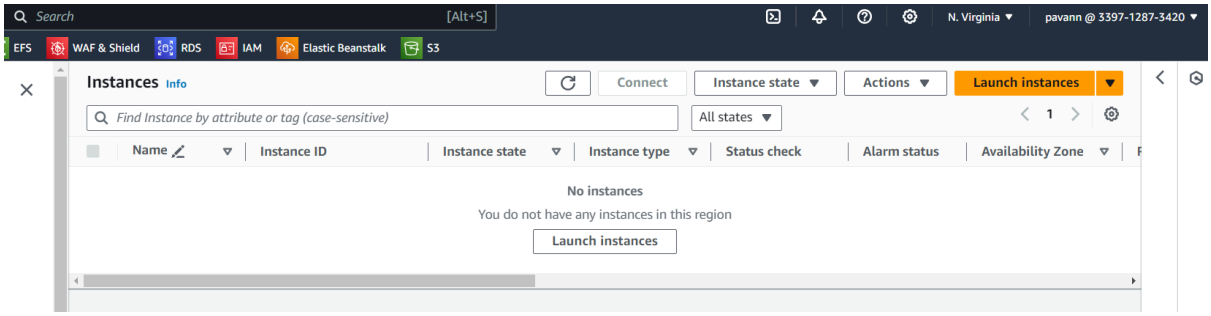
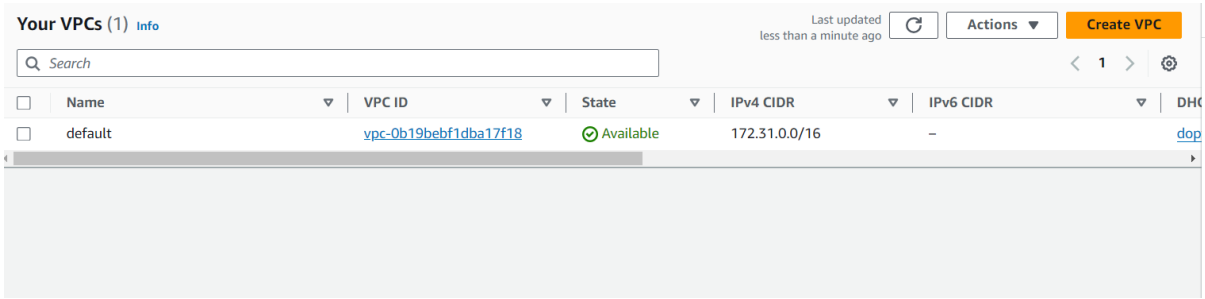
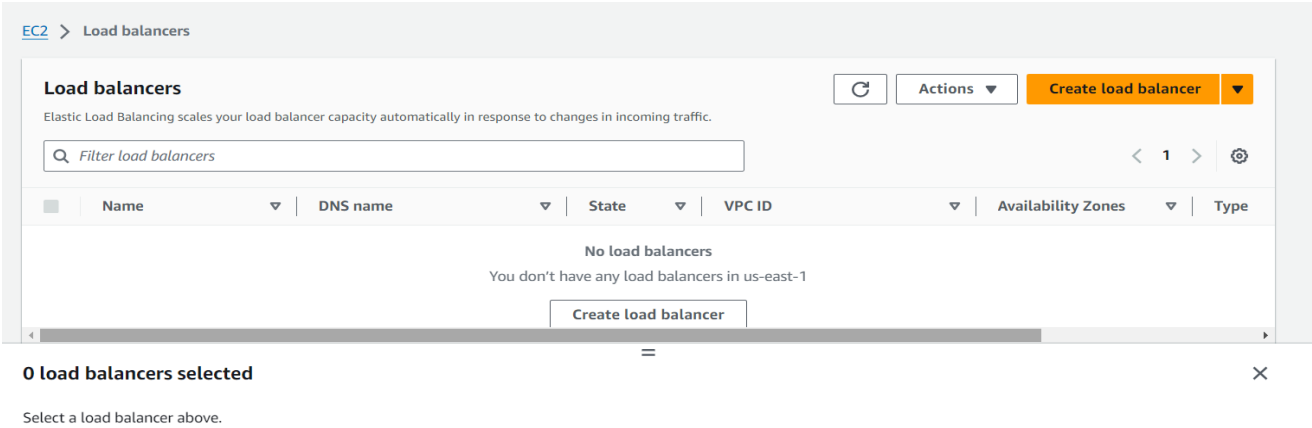
Create load balancer with terraform





Login Aws Account



Open visual studio

Download aws cli and terraform cli

Update in environment variables

Create provider.tf (who are providing access to your aws,gcp,azure)

Load balancer > provider.tf

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

Create vpc in terraform

Using vpc.tf

Load balancer > vpc.tf > resource "aws_vpc" "Pavann"

```
resource "aws_vpc" "Pavann"{
    cidr_block = "10.0.0.0/16"
    tags = {
      Name = "Pavann-vpc"
    }
}
```

Create subnets for vpc

Load balancer > subnet.tf > resource "aws_subnet" "two-public-subnet2"

```
resource "aws_subnet" "one-public-subnet1" {
  vpc_id = aws_vpc.Pavann.id
  cidr_block = "10.0.0.0/24"
  map_public_ip_on_launch = "true"
  availability_zone = "us-east-1a"
  tags = {
    Name = "one-public-subnet1"
  }
}

resource "aws_subnet" "two-public-subnet2" {
  vpc_id = aws_vpc.Pavann.id
  cidr_block = "10.0.1.0/24"
  map_public_ip_on_launch = "true"
  availability_zone = "us-east-1b"
  tags = {
    Name = "two-public-subnet2"
  }
}
```

Create IGW

Load balancer > igw.tf > resource "aws_internet_gateway" "pavann-gateway"

```
resource "aws_internet_gateway" "pavann-gateway" {
  vpc_id = aws_vpc.Pavann.id
}
```

Create Route tables

Route 1

Route 2

Load balancer > route.tf > resource "aws_route_table_association" "pavann-route2"

```
resource "aws_route_table" "pavann-route" {
vpc_id = aws_vpc.Pavann.id
route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.pavann-gateway.id

}
tags = {
  Name = "route to internet"
  }
}
#route 1
resource "aws_route_table_association" "pavann-route1" {
subnet_id = aws_subnet.one-public-subnet1.id
route_table_id = aws_route_table.pavann-route.id
}
#route 2
resource "aws_route_table_association" "pavann-route2" {
subnet_id = aws_subnet.two-public-subnet2.id        #aws_subnet.two-public-subnet2.id
route_table_id = aws_route_table.pavann-route.id
}
```

Create security groups

Load balancer > sg.tf > resource "aws_security_group" "pavann-sg"

```
resource "aws_security_group" "pavann-sg" {
vpc_id = aws_vpc.Pavann.id
    ingress {
     from_port = 80
     to_port   = 80
     protocol = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
    }
    ingress {
     from_port = 443
     to_port   = 443
     protocol = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
    }
    ingress {
     from_port = 22
     to_port   = 22
     protocol = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
    }
     egress {
     from_port = 0
     to_port   = 0
     protocol = "-1"
     cidr_blocks = ["0.0.0.0/0"]
    }
    tags = {
      Name = "pavann-sg"
    }
}
```

Create EC2 instances with the name of ocus and ocus1

```
oad balancer > ec2.tf > resource "aws_instance" "ocus1" > tags > Name
resource "aws_instance" "ocus" {
  ami = "ami-04a81a99f5ec58529"
  instance_type = "t2.micro"
  key_name      = "pavan"
  vpc_security_group_ids = [aws_security_group.pavann-sg.id]
  subnet_id              = aws_subnet.one-public-subnet1.id
  associate_public_ip_address = true
     user_data = file("${path.module}/appache1.sh")
   tags = {
     Name = "pavann"
  }

}

resource "aws_instance" "ocus1" {
  ami = "ami-04a81a99f5ec58529"
  instance_type = "t2.micro"
  key_name      = "pavan"
  vpc_security_group_ids = [aws_security_group.pavann-sg.id]
  subnet_id              = aws_subnet.two-public-subnet2.id
  associate_public_ip_address = true
     user_data = file("${path.module}/appache2.sh")
  tags = {
     Name = "pavann"
  }

}
```

Create user data script

Create nginx.sh

```
Load balancer > $ appache1.sh
#!/bin/bash
sudo apt update -y &&
sudo apt install -y nginx
echo "hi this is pavan" > /var/www/html/index.html
sudo systemctl restart nginx.service
```

Create nginx1.sh

Load balancer > $ appache2.sh

```
#!/bin/bash
sudo apt update -y &&
sudo apt install -y nginx
echo "hi this is "khumar" > /var/www/html/index.html
sudo systemctl restart nginx.service
```

Create target group and load balancer

Load balancer > loadbalancer.tf > resource "aws_lb_target_group" "pavank-tg" > # port

```
resource "aws_lb" "pavan-lb" {
name                = "pavan-LB"
 internal           = false
 load_balancer_type = "application"
 security_groups    = [aws_security_group.pavann-sg.id]
 subnets            = [aws_subnet.one-public-subnet1.id,aws_subnet.two-public-subnet2.id]

}
resource "aws_lb_target_group" "pavank-tg" {
  name     = "pavank-TG"
  port     = 80
  protocol = "HTTP"
  vpc_id   = aws_vpc.Pavann.id
  health_check {
    path     = "/health"
    port     = 80
    protocol = "HTTP"
  }
}
resource "aws_lb_target_group_attachment" "pavank" {
  target_group_arn = aws_lb_target_group.pavank-tg.arn
  target_id        = aws_instance.ocus.id
  port             = 80
  depends_on = [
    aws_lb_target_group.pavank-tg,
    aws_instance.ocus
  ]
}
resource "aws_lb_target_group_attachment" "pavan" {
  target_group_arn = aws_lb_target_group.pavank-tg.arn
  target_id        = aws_instance.ocus1.id
  port             = 80
  depends_on = [
    aws_lb_target_group.pavank-tg,
    aws_instance.ocus1
  ]
}
```

```
resource "aws_lb_listener" "listener_elb" {
load_balancer_arn  = aws_lb.pavan-lb.arn   #aws_lb.pavan-lb.arn
port               = 80
protocol           = "HTTP"
default_action {
  type             = "forward"
  target_group_arn = aws_lb_target_group.pavank-tg.arn
}
}
```

Type terraform init

```
PS C:\Users\DELL\Terraform\Load balancer> terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.63.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to se
```

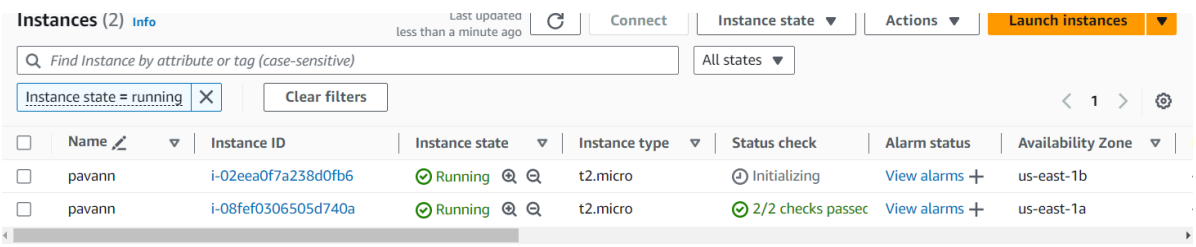
Terraform plan for dry run which you want to creating it shows advance

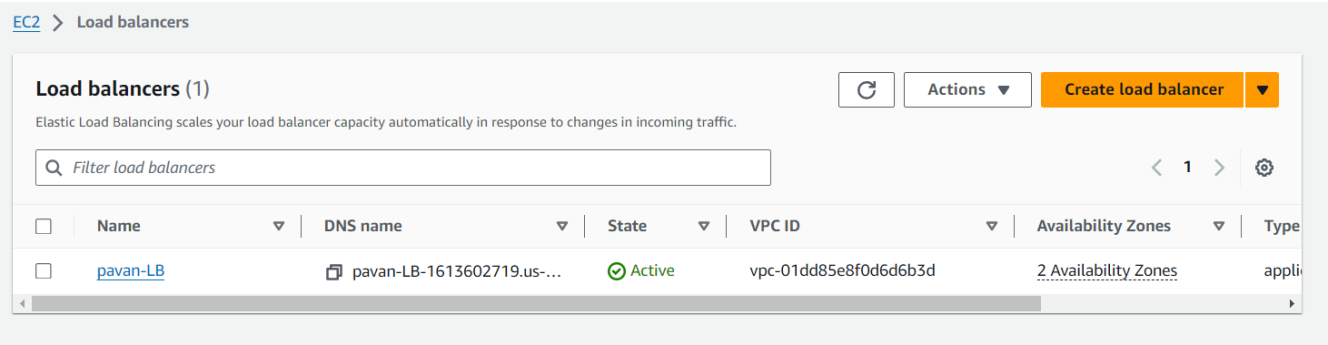
```
PS C:\Users\DELL\Terraform\Load balancer> terraform plan

Terraform used the selected providers to generate the following executi
  + create

Terraform will perform the following actions:

  # aws_instance.ocus will be created
  + resource "aws_instance" "ocus" {
      + ami                                  = "ami-04a81a99f5ec58529"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + 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)
```

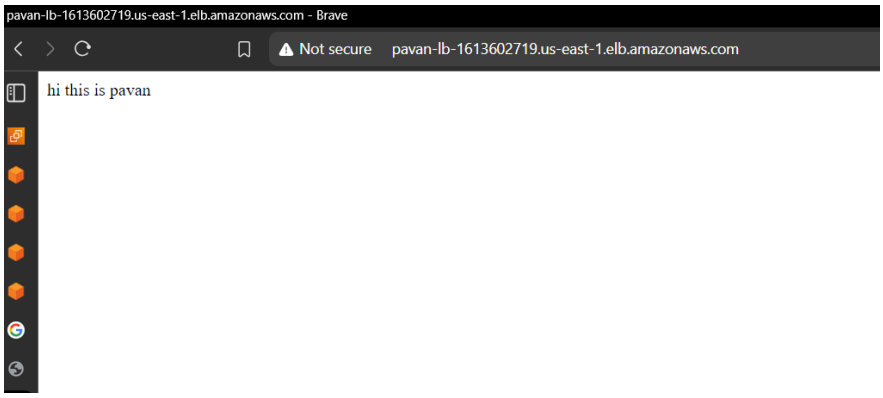
For creating you want to use terraform apply

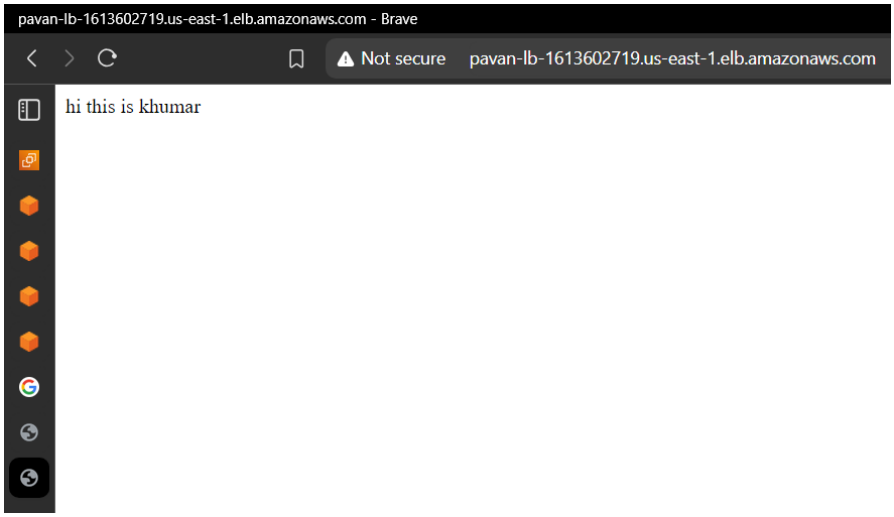
```
aws_vpc.Pavann: Creating...
aws_vpc.Pavann: Creation complete after 5s [id=vpc-01dd85e8f0d6d6b3d]
aws_subnet.two-public-subnet2: Creating...
aws_lb_target_group.pavank-tg: Creating...
aws_internet_gateway.pavann-gateway: Creating...
aws_subnet.one-public-subnet1: Creating...
aws_security_group.pavann-sg: Creating...
aws_internet_gateway.pavann-gateway: Creation complete after 3s [id=igw-0ada8c11869d6236b]
aws_route_table.pavann-route: Creating...
aws_lb_target_group.pavank-tg: Creation complete after 4s [id=arn:aws:elasticloadbalancing:us-east-1:339712873420:targetgroup/pavank-TG/1fa8ca999c6
aws_route_table.pavann-route: Creation complete after 2s [id=rtb-0aa722312b2b81641]
aws_security_group.pavann-sg: Creation complete after 6s [id=sg-021809046c5ba44fd]
aws_subnet.two-public-subnet2: Still creating... [10s elapsed]
aws_subnet.one-public-subnet1: Still creating... [10s elapsed]
aws_subnet.two-public-subnet2: Creation complete after 13s [id=subnet-06c1e137a6632d094]
```

After that is it created or not you can see on aws console




EC2 > Load balancers
Load balancers (1)
Actions
Create load balancer
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Filter load balancers
1
Name
DNS name
State
VPC ID
Availability Zones
Type
pavan-LB
pavan-LB-1613602719.us-...
Active
vpc-01dd85e8f0d6d6b3d
2 Availability Zones


pavan-lb-1613602719.us-east-1.elb.amazonaws.com - Brave
Not secure
pavan-lb-1613602719.us-east-1.elb.amazonaws.com
hi this is pavan


pavan-lb-1613602719.us-east-1.elb.amazonaws.com - Brave
Not secure
pavan-lb-1613602719.us-east-1.elb.amazonaws.com
hi this is khumar