# Get the aws cli if you don't already have it

https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

# Configure your aws cli

https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html

# Test to make sure you have version 2 of the aws cli installed

aws elbv2 help

# Get the AMI id of the free tier eligible AMI: ami-0742b4e673072066f

# List your VPCs

aws ec2 describe-vpcs

# Create an environment variable for your VPC

export VPC=vpc-492aea2e

# Use a key pair from your existing key pairs

MyALBKeyPair.pem

# Create your security group

aws ec2 create-security-group --group-name MyALBSecurityGroup --description "My ALB security group"

# Set environment variable for your security group

export SGID=sg-09477f9083e9de4f8

# Get your local IP Address

curl https://checkip.amazonaws.com

# Set an environment variable for you ip address

export IPADD=151.199.248.31

# Add ssh and HTTP rules to you inbound rules

aws ec2 authorize-security-group-ingress --group-id $SGID --protocol tcp --port 22 --cidr $IPADD/32

aws ec2 authorize-security-group-ingress --group-id $SGID --protocol tcp --port 80 --cidr 0.0.0.0/0

# Create your subnets

aws ec2 create-subnet --vpc-id $VPC --availability-zone-id use1-az1 --cidr-block 172.31.128.0/20

aws ec2 create-subnet --vpc-id $VPC --availability-zone-id use1-az2 --cidr-block 172.31.192.0/20

# Create environment variables for your subnets

export AZ1SUB=subnet-05360d7c4734dc693

export AZ2SUB=subnet-0ffd9fcd47685e987

# Create your EC2 instances using the AMI id of ami-0742b4e673072066f (the free tier eligible AMI), two in each subnet; one for the video server and the other for the web server

aws ec2 run-instances --image-id ami-0742b4e673072066f \

--instance-type t2.micro --count 1 --subnet-id $AZ1SUB \

--key-name MyALBKeyPair2 --security-group-ids $SGID \

--associate-public-ip-address --user-data file://userdata-video-server-1.txt

aws ec2 run-instances --image-id ami-0742b4e673072066f \

--instance-type t2.micro --count 1 --subnet-id $AZ2SUB \

--key-name MyALBKeyPair2 --security-group-ids $SGID \

--associate-public-ip-address --user-data file://userdata-video-server-2.txt

aws ec2 run-instances --image-id ami-0742b4e673072066f \

--instance-type t2.micro --count 1 --subnet-id $AZ1SUB \

--key-name MyALBKeyPair2 --security-group-ids $SGID \

--associate-public-ip-address --user-data file://userdata-web-server-1.txt

aws ec2 run-instances --image-id ami-0742b4e673072066f \

--instance-type t2.micro --count 1 --subnet-id $AZ2SUB \

--key-name MyALBKeyPair2 --security-group-ids $SGID \

--associate-public-ip-address --user-data file://userdata-web-server-2.txt

# Create environment variables for your EC2 instances

export VIDSERV1=i-06bff504f600d141c

export VIDSERV2=i-043121ab7943f4d71

export WEBSERV1=i-0af79b8d0cde97923

export WEBSERV2=i-06c70da990ed10f1b

# Tage your instances with names

aws ec2 create-tags --resources $VIDSERV1 --tags Key="Name",Value="Video Server #1"

aws ec2 create-tags --resources $VIDSERV2 --tags Key="Name",Value="Video Server #2"

aws ec2 create-tags --resources $WEBSERV1 --tags Key="Name",Value="Web Server #1"

aws ec2 create-tags --resources $WEBSERV2 --tags Key="Name",Value="Web Server #2"

#Create your Video Load Balancer

aws elbv2 create-load-balancer --name MyALB --subnets $AZ1SUB $AZ2SUB --security-groups $SGID

# Create environment variables for your ALB ARN and DNS name

export ALBARN=arn:aws:elasticloadbalancing:us-east-1:############:loadbalancer/app/MyALB/a6f121e8d8e04021

export ALBDNS=MyALB-949638453.us-east-1.elb.amazonaws.com

# Create your Target Groups

aws elbv2 create-target-group --name VideoTargets --protocol HTTP --port 80 --vpc-id $VPC

aws elbv2 create-target-group --name WebTargets --protocol HTTP --port 80 --vpc-id $VPC

# Create environment variables for your target group ARNs

export VIDTGARN=arn:aws:elasticloadbalancing:us-east-1:############:targetgroup/VideoTargets/ecf67383b371d68e

export WEBTGARN=arn:aws:elasticloadbalancing:us-east-1:############:targetgroup/WebTargets/c62e5325012e7d51

# Register your EC2 instances with your Target Groups

aws elbv2 register-targets --target-group-arn $VIDTGARN --targets Id=$VIDSERV1 Id=$VIDSERV2

aws elbv2 register-targets --target-group-arn $WEBTGARN --targets Id=$WEBSERV1 Id=$WEBSERV2

# Create a listener for your ALB and give it a default Target Group of the web target group

aws elbv2 create-listener --load-balancer-arn $ALBARN --protocol HTTP \

--port 80 --default-actions Type=forward,TargetGroupArn=$WEBTGARN

# Create an environment variable for your listener ARN

export LISTARN=arn:aws:elasticloadbalancing:us-east-1:############:listener/app/MyALB/a6f121e8d8e04021/b233b9957d512d9a

# Verify the health of your targets in each Target Group

aws elbv2 describe-target-health --target-group-arn $VIDTGARN

aws elbv2 describe-target-health --target-group-arn $WEBTGARN

# Add path-based routing

aws elbv2 create-rule \

--listener-arn $LISTARN \

--priority 5 \

--conditions file://conditions-pattern.json \

--actions Type=forward,TargetGroupArn=$VIDTGARN

# Get your listener arns

aws elbv2 describe-rules --listener-arn $LISTARN

# Create environment variables for your rule ARNs

export VIDRULEARN=arn:aws:elasticloadbalancing:us-east-1:############:listener-rule/app/MyALB/a6f121e8d8e04021/b233b9957d512d9a/5ac15ce0c4755a35

export WEBRULEARN=arn:aws:elasticloadbalancing:us-east-1:############:listener-rule/app/MyALB/a6f121e8d8e04021/b233b9957d512d9a/d40b830a3ddef854

# Delete your listener rules

aws elbv2 delete-rule --rule-arn $VIDRULEARN

aws elbv2 delete-rule --rule-arn $WEBRULEARN

# Delete your listener

aws elbv2 delete-listener --listener-arn $LISTARN

# Delete your Target Groups

aws elbv2 delete-target-group --target-group-arn $VIDTGARN

aws elbv2 delete-target-group --target-group-arn $WEBTGARN

# Delete your ALB

aws elbv2 delete-load-balancer --load-balancer-arn $ALBARN

# Terminate your EC2 instances

aws ec2 terminate-instances --instance-ids $VIDSERV1 $VIDSERV2 $WEBSERV1 $WEBSERV2

# Delete your subnets

aws ec2 delete-subnet --subnet-id $AZ1SUB

aws ec2 delete-subnet --subnet-id $AZ2SUB

# Delete your security group

aws ec2 delete-security-group --group-id $SGID