# 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 MyNLBSecurityGroup \

--description "My NLB security group"

# Set environment variable for your security group

export SGID=sg-0e9072e60af2f20d0

# Get your local IP Address

curl https://checkip.amazonaws.com

# Set en environment variable for you ip address

export IPADD=151.199.248.31

# Add ssh and TCP 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 udp --port 6380 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id $SGID \

--protocol tcp --port 6381 --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-04bc4f621f670e64b

export AZ2SUB=subnet-0a70e9f5ae4684722

# 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://tcp-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://tcp-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://udp-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://udp-server-2.txt

# Create environment variables for your EC2 instances

export TCPSERV1=i-0eac4fba1ea5ee2bd

export TCPSERV2=i-04ed80dd55594dc8e

export UDPSERV1=i-0eb39c518f743f480

export UDPSERV2=i-042de56491fd1dc96

# Tage your instances with names

aws ec2 create-tags --resources $TCPSERV1 --tags Key="Name",Value="TCP Server #1"

aws ec2 create-tags --resources $TCPSERV2 --tags Key="Name",Value="TCP Server #2"

aws ec2 create-tags --resources $UDPSERV1 --tags Key="Name",Value="UDP Server #1"

aws ec2 create-tags --resources $UDPSERV2 --tags Key="Name",Value="UDP Server #2"

#Create your TCP and UDP Load Balancer

aws elbv2 create-load-balancer --name MyNLB \

--type network --subnets $AZ1SUB $AZ2SUB

# Create environment variables for your NLB ARN and DNS name

export NLBARN=arn:aws:elasticloadbalancing:us-east-1:############:loadbalancer/net/MyNLB/67a0933c047467bf

export NLBDNS=MyNLB-fb5a789c4bf10ea1.elb.us-east-1.amazonaws.com

# Create your Target Groups

aws elbv2 create-target-group --name TCPTargets \

--protocol TCP --port 6381 --vpc-id $VPC

aws elbv2 create-target-group --name UDPTargets \

--protocol UDP --port 6380 --vpc-id $VPC

# Create environment variables for your target group ARNs

export TCPTGARN=arn:aws:elasticloadbalancing:us-east-1:############:targetgroup/TCPTargets/3a7f5d97ccdd56b5

export UDPTGARN=arn:aws:elasticloadbalancing:us-east-1:############:targetgroup/UDPTargets/3e6609df53dd72f4

# Register your EC2 instances with your Target Groups

aws elbv2 register-targets --target-group-arn $TCPTGARN \

--targets Id=$TCPSERV1 Id=$TCPSERV2

aws elbv2 register-targets --target-group-arn $UDPTGARN \

--targets Id=$UDPSERV1 Id=$UDPSERV2

# Create listeners for your NLB and give them a default Target Group of the based on protocol

aws elbv2 create-listener --load-balancer-arn $NLBARN --protocol TCP \

--port 6381 --default-actions Type=forward,TargetGroupArn=$TCPTGARN

aws elbv2 create-listener --load-balancer-arn $NLBARN --protocol UDP \

--port 6380 --default-actions Type=forward,TargetGroupArn=$UDPTGARN

# Create an environment variable for your listener ARN

export TCPLISTARN=arn:aws:elasticloadbalancing:us-east-1:############:listener/net/MyNLB/67a0933c047467bf/1b742e080da8cbff

export UDPLISTARN=arn:aws:elasticloadbalancing:us-east-1:############:listener/net/MyNLB/67a0933c047467bf/1537369bbcf6249a

# Verify the health of your targets in each Target Group

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

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

# Get your listener arns

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

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

# Create environment variables for your rule ARNs

export TCPRULEARN=arn:aws:elasticloadbalancing:us-east-1:############:listener-rule/net/MyNLB/67a0933c047467bf/1b742e080da8cbff/e9f08f9d4626ef70

export UDPRULEARN=arn:aws:elasticloadbalancing:us-east-1:############:listener-rule/net/MyNLB/67a0933c047467bf/1537369bbcf6249a/47c2498bfc083953

# Get the UDP output from UDP servers

# then ssh to one of your UDP servers and run these commands

# sudo su

# ps -aef | grep server

# strace -p#### -s9999 -e write # where #### is the pid of the server process you retrieve from the ps command

# then run the udpsender.py with the NLB DNS name or the ip address of the UDP server as its parameter

# you'll see the output in your ssh session with the UDP server

# Delete your listener rules

aws elbv2 delete-rule --rule-arn $TCPRULEARN

# Delete your listeners

aws elbv2 delete-listener --listener-arn $TCPLISTARN

aws elbv2 delete-listener --listener-arn $UDPLISTARN

# Delete your Target Groups

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

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

# Delete your NLB

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

# Terminate your EC2 instances

aws ec2 terminate-instances --instance-ids $TCPSERV1 $TCPSERV2 $UDPSERV1 $UDPSERV2

# 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