

Project 2: Building a multi-tier website on using AWS components

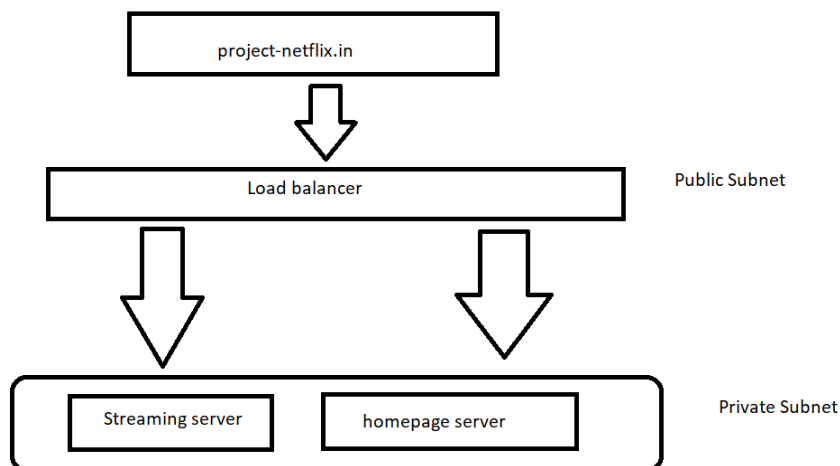
Components used:

- 1)VPC
- 2)EC2
- 3)Load balancer
- 4)Route 53

Summary:

This project has a pseudo architecture and it is inspired from Netflix, In Netflix when we are on a main home page the URL comes up like netflix/browse and we are watching some content over the Netflix the URL comes up like Netflix/watch. We have used a similar architecture where we can hit the homepage and the streaming page based on our preference. This is done by creating two different target groups for the streaming servers and homepage servers, load balancer was used to hit these target groups. We created a proxy server where we coded the website (simple text for our project), later on we modified the website based on our need and simultaneously created AMIs from them, so that in case we need more server, we can create from them. In the final stage we used Route 53 to map the domain that we purchased from Freenom and after the mapping is complete, we were able to launch the website with the domain registered. Also, when we typed domain/home the system would get us the home page server and when domain/stream was launched streaming content from the streaming server would come up.

Diagram:



Steps:

1) Create the network using VPC service in AWS console, following were steps we followed:

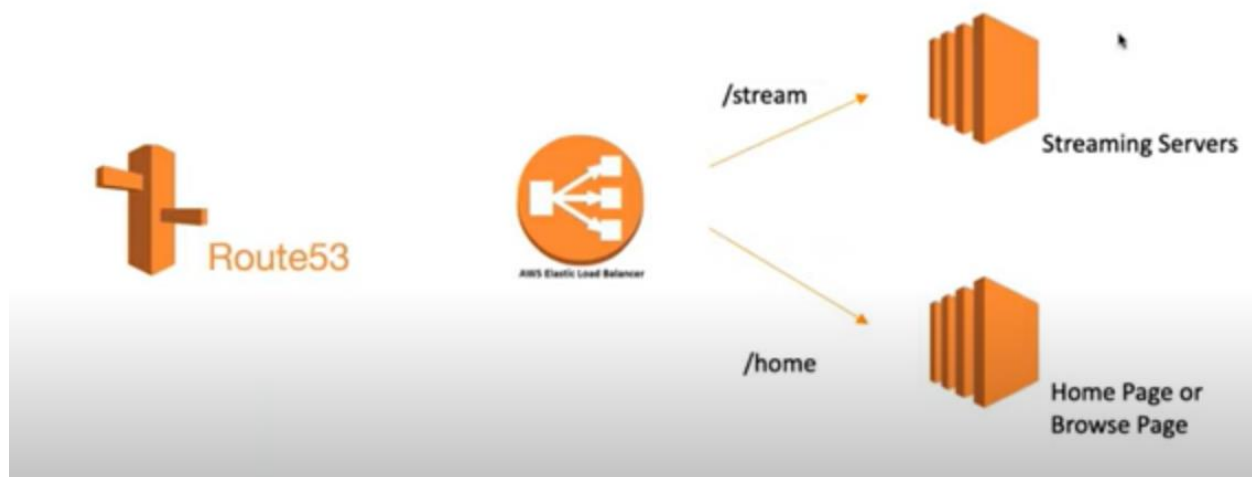
- Navigate to services and find VPC service, create new VPC, fill the details as range (10.1.0.0/16) and name of the network
- Create four subnets a) Two public subnet (with internet access, range 10.1.0.0/24, 10.1.2.0/24) b) Private subnet (without internet access, range 10.1.3.0/24, 10.1.4.0/24)
- Create an internet gateway to give internet access to the public subnets
- Once IG is created, attach it to VPC
- The public subnet will still not be able to access internet because there is no route created for the internet, in order to resolve this, we will have to create a route table and associate with subnets and internet gateway
- Go to route tables, there is a default route table is created automatically when a VPC is created, rename this route table to be attached to the public subnets
- Create a new route table for the non-internet accessing servers
- On the default route table, add internet gateway route (under edit route), Also, connect the public subnets to it (under edit subnet association)
- On the newly created route table, attach the private subnet to this route table, these will not have access to internet (edit subnet association and attach the two private subnets)
- Enable the auto assign IP address to the public subnets, this is to provide a public IP to the machines that are deployed in these subnets, by doing this we will be able to connect to these machines via internet

2) Create a proxy server:

- Navigate to the EC2 section in AWS console
- Launch a EC2 instance with Amazon Linux OS, while launching the server, select the VPC that we have created, choose subnet as a public subnet (to have internet access), for now use the default security group for this instance,
- Connect to this instance using "PUTTY" software, install apache on this server to make it a webserver

3) Website infrastructure concept and diagram:

- Two webpages will be created for streaming page and home page
Eg: Netflix/home: this is the homepage
Netflix/stream: this is the page while streaming contents
- We will be following the same scenario, two webserver will have home and stream page, we will be using route 53 to go to home or stream servers based on our preference



Creating the home page and streaming page:

- Go the following path in the proxy Linux server var/www/html and create a new index.html page
- This page can contain text like “welcome to home page”
- Once this is saved, we can see this page when server IP is pasted in the web browser
- Create a copy(image) of this server, we will use this image to deploy new server
- Repeat the same process again to create the streaming page, i.e it will have text like “welcome to streaming page”, this way we will be able to differentiate the request, create the image of this as well
- These images can be found in “My AMI” section in EC2
- Create a new web server (from the home page AMI) that will serve as home page, this server will be launched in private subnet (no internet access) under custom VPC we created, name this as “homepage”
- SSH in this server, create a directory called “/home” and copy the index.html file to it
- Create a new web server (from the streaming page AMI) that will serve as home page, this server will also be launched in private subnet (no internet access) under custom VPC we created, name this as “streampage”
- SSH in this server, create a directory called “/stream” and copy the index.html file to it

Configuring the Load balancer for these server:

- Navigate to load balancer heading in the EC2 section
- Navigate to Application load balancer, we will be using the http protocol
- Load balancer will be the part of the custom VPC that we created
- Deploy the load balancer in the two public subnet
- Default security group to be added
- Create a new target group, name this as “home page tg”

- Register target for this target group as the “homepage” server in the next screen and hit add to register
- Create another target group as “streaming server tg”
- Register a target for this one as the “streamingpage” server, hit add to register
- Specifying the routing rules for LB, select the LB and scroll down to listener, edit rules in listener
- Hit add rule in listener, specify a path = “/home*”, Forward = “homepage” (server name)
- Similarly add a rule for streaming where path = “/stream*”, Forward = “streamingpage” (server name)
- Validate the above configuration by launching the LB URL and using “/stream” and “/home” to toggle on the website pages

Configuring the Route 53:

- Get a domain from the freenom
- Navigate to route 53, create a hosted zone with the same domain that we have brought from freenom
- Copy the name servers from route 53 to freenom nameserver and save them in freenom portal
- In Route 53 set the alias = “yes” , alias target = “load balancer”(points to the newly created LB)
- Create another record set, name as “www”, alias = “yes” , alias target = “load balancer”(points to the newly created LB), hit save

Once this is completed we should be able to browse to the website using the naked domain as well as regular domain from freenom(example anmol.sharma@cf.com and www.anmol.sharma@cf.com)