**LOADBALANCER WITH NETWORK LOAD WITH EXTERNAL**

**🡪**by using a network load balancer, we need to perform port-based routing. -->application load balancer, we need to perform path-based routing

1. create 1 vpc

2. create 3 subnets

3. create 1 router

4. create an internet gateway and attach vpc

5. create 3 ec2 instances for each subnet

6. lunch 3 mobs

7. sudo su,apt-get update, apt-get upgrade

8. apt-get install nginx

9. service nginx status

10. systemctl start nginx. Service

11.copy ip public address of 3(you need to gry nginx on chrome)

12. cd /var/www/html/ 🡪ls (you will get an index.html file)

13. vi (copy that index file and keep the h1 tag in the style above)

14.copy ip public(changes will get in nginx)

15. next we need to create a load balancer for that first we need to create target groups

16. create target group with instances🡪tcp🡪2🡪30🡪create🡪select all🡪include below🡪create

17. create load balancer 🡪with subnets🡪tcp

18. and check target should get healthy.

19. copy DNS name in chrome

Ec2-Webserver1

Webserver2

Webserver3

**Network load balancer for each instance of each port-based target. (80,8000,8080)for length decrease of DNS**

🡪Now we need 3 instances for 3 target groups.

🡪target1, target2, target3

🡪for target groups we need to keep port TCP must and should, including below only 1 target for each.

🡪next we need to click load balancer🡪down go to add listener🡪tcp 80,8000,8080🡪and select target group target1, target2, target3.

🡪we should get individual targets healthy.

🡪and go to load balancer copy DNS with ports (80,8000,8080)

A screenshot of a computer

Description automatically generated

🡪go to management certification🡪next🡪validation🡪DNS validation🡪key🡪RSA🡪request.

🡪Route 53 is the mediator for your domain service provider and your load balancer provider (or any other service on AWS)

🡪go to Route53🡪dashboard🡪Hosted zone🡪copy domain name

🡪create record 🡪record name(www)🡪record type(C name)🡪value(go to load balancer DNS copy🡪create record

🡪go to certificate manager🡪click🡪copy Cname(1st)🡪 type(C name)🡪value(cname value)

🡪go to the certificate manager and check the status.

🡪go to the domain click the domain edit namespaces and copy(4)

🡪go to route53🡪copy 4 namespaces.

🡪ro go route 53 delete www one and create one more.

🡪create record one more 🡪www🡪ipv4🡪alias enable🡪network load balancer🡪region🡪network load balancer.

🡪www.navyar.com copy in google.

A screenshot of a computer

Description automatically generated

**Create records for failover, weighted, latency, and geolocation.**

**Crete 3 VPC and 3 subnets 3 routers 3 ec2 instances with different regions like Virginia, Ohio, Mumbai**

🡪Create 3 ec2 instances in 3 different regions.

🡪first create VPC, subnet, router, ig, and ec2 in 3 regions

🡪open mobs

🡪 sudo su

🡪apt-get update.

🡪apt-get upgrade.

🡪apt-get install nginx.

🡪service nginx status.

🡪cd /var/www/html/

🡪ls (index.html)

🡪vi

🡪in that add an h1 tag like the east 1 server

🡪same thing in 3 instances (Virginia, Ohio, Mumbai)

🡪and check it 3 mobs instances.

🡪next go to VPC enable settings and enable DNS.

🡪go to ec2 and refresh you will get public IPV4 DNS

🡪go to route 53 🡪and go to health check give a name like Virginia east1🡪monitor endpoint🡪protocol HTTP 🡪and keep domain name (copy that ec2 ipv4 DNS)🡪next🡪

🡪and again, create a health check for Ohio🡪first go to VPC enable DNS🡪and to go to ec2 instance and refresh you will get Public IPV4 DNS and go to health check give a name like OHIO east2🡪monitor endpoint🡪protocol HTTP 🡪and keep domain name (copy that ec2 ipv4 DNS)🡪next🡪

🡪 and again, create a health check for Mumbai🡪first go to VPC enable DNS🡪and to go to ec2 instance and refresh you will get Public IPV4 DNS and go to health check give a name like OHIO east2🡪monitor endpoint🡪protocol HTTP 🡪and keep domain name (copy that ec2 ipv4 DNS)🡪next🡪

🡪

🡪Go to route53🡪host zones🡪create first for failover🡪domain name(failover)

Record type (a type IPV4)🡪60🡪and click define failover record🡪 choose endpoint (IP address or another value)🡪and copy Virginia public IP address and paste it🡪failover record type(primary)🡪health check (Virginia east1)🡪record ID (Virgina east1 server)🡪create failover record🡪 domain name(failover)

Record type (a type IPV4)🡪60🡪and click define failover record🡪 choose endpoint (IP address or another value)🡪and copy OHIO public IP address and paste it🡪failover record type(primary)🡪health check (Virginia east1)🡪record ID (Virgina east1 server)🡪

🡪now go to hosted zones🡪failover.navyar.com🡪copy in google you need to get failover one.

A screenshot of a computer

Description automatically generated

🡪go to Virginia mobs.

🡪curl http://failover.navyar.com/ (this is the above one) (you will get an HTML file)

🡪curl http://failover.navyar.com/ | grep 'EAST1’(this east1 comes in above HTML file)

🡪curl -sL http://failover.navyar.com/ | grep 'EAST1’(will get like this <h1>US-EAST1-SERVICE</h1>)

🡪while true

🡪do

🡪 curl -sL http://failover.navyar.com/ | grep 'EAST1’

🡪sleep1

🡪done (you will get a lot)

🡪stop ec2 instance of Virginia. (In route 53 check healthy checks (unhealthy in Virginia)

🡪go to the ec2 instance of Ohio and copy it to Google failovernavyar.com.

A screenshot of a computer

Description automatically generated

🡪again, start ec2 instance of Virginia.

🡪go to route53 🡪healthy checks of Virginia 🡪edit health check and paste again ec2 instance of started Virginia updated one🡪create.

🡪this everything failover

**The weighted instance weight is 255 have to share into 3 instances because we have 3 different instances in different regions, we have to share the load balancer to all 3 instances like 85.**

**🡪**go to hosted zone 🡪go to the weighted 🡪name weighted🡪type(A type IPV4)🡪60🡪and define record create 3 instances for 3 define records each weight 85🡪create

🡪and copy with weighted. navyar.com you need to get this

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**3. Latency**

**🡪**create a record of latency.

**🡪**create record🡪latency🡪name(lat)🡪type (A IPV4)🡪60🡪create define latency record 🡪create 3 🡪3 instances for locations 3

A screenshot of a computer

Description automatically generated

Will get the nearest one mine is in Virginia.

**4. GEOLOCATION**

**🡪create a record in geolocation.**

**🡪**create 3 defined records with Virginia-Georgia

Ohio-California

Mumbai-India

A screenshot of a computer screen

Description automatically generated

The nearest one will get

-