=> static website hosting => user data

<u>Load balancer</u>, monolith, microservices }

Website :
 -> Collection of web pages (html pages)
 ->static website --> gives same response to every user
-> dynamic website --> gives response based on user

--> Webserver is used to host/run our website

for static websites --> httpd, apache2, ...
for dynamic websites --> tomcat , IIS..........

Hosting website using httpd
==================
$ sudo yum update -y

$ sudo yum install httpd

$ sudo systemctl start httpd
Note : Enable HTTP : 80 in Security group inbound rules

Access our website using EC2 vm public ip

to modify the content we can navigate
$ cd /var/www/html

sudo vi index.html

insert : <h1> bbbbbb</h1>

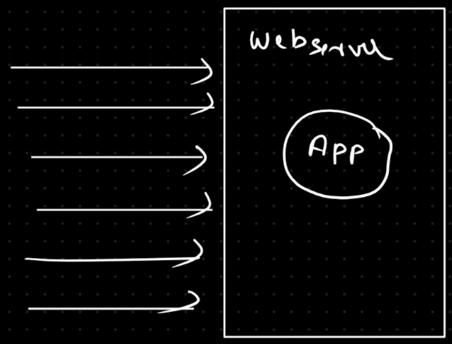
Again access our website using ec2 instance public ip

user-data in EC2 VM
====================
--> used to execute script while launching machine

-->User data will execute only once

Create EC2 VM with below user data

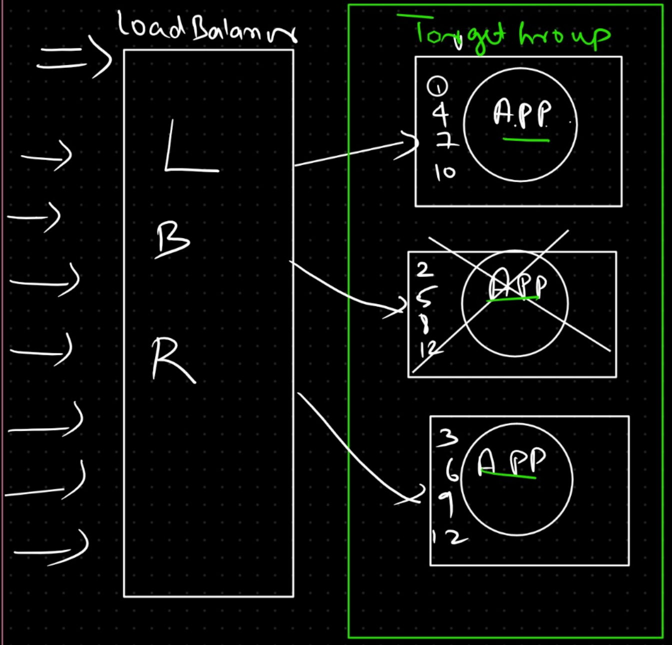
<u>Disadvantage of having one server</u>



=> One server must handle all the incoming request
-> high burden on server which might result in delay in responses

-> can lead to server crash (single point of failure)

<u>Business loss</u>



Round Robin

=> App will run on multiple servers

-> Load will be distributed

-> Fast Performance

-> High availability

Load Balancer LBR --> used to distribute incoming load to multiple servers in round robbin technique

There diff types of Load Balancers in AWS :
1) Application Load Balancer (http & https)
2) Network Load Balancer
3) Gateway Load Balancer

Classic Load Balancer (outdated / old gen)

Practical Task on Load Balancer
===========================

--->Create EC2 VM1

```
#! /bin/bash

sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Telusko Banking App Server -1 </h1></html>" > index.html
service httpd start
```

---> Create EC2 VM2

```
#! /bin/bash

sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Telusko Banking App Server -2 </h1></html>" > index.html
service httpd start
```

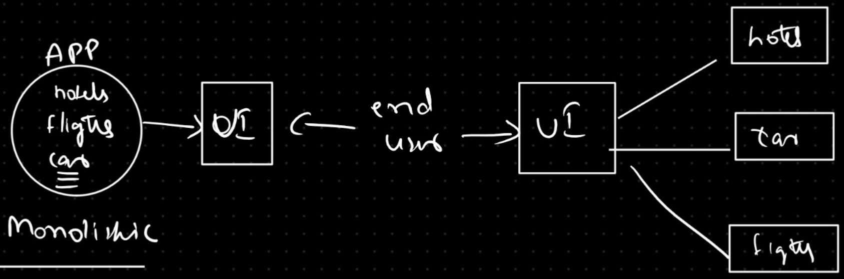
-->Add these instances to one Target Group -->(TG - List of servers running our app)

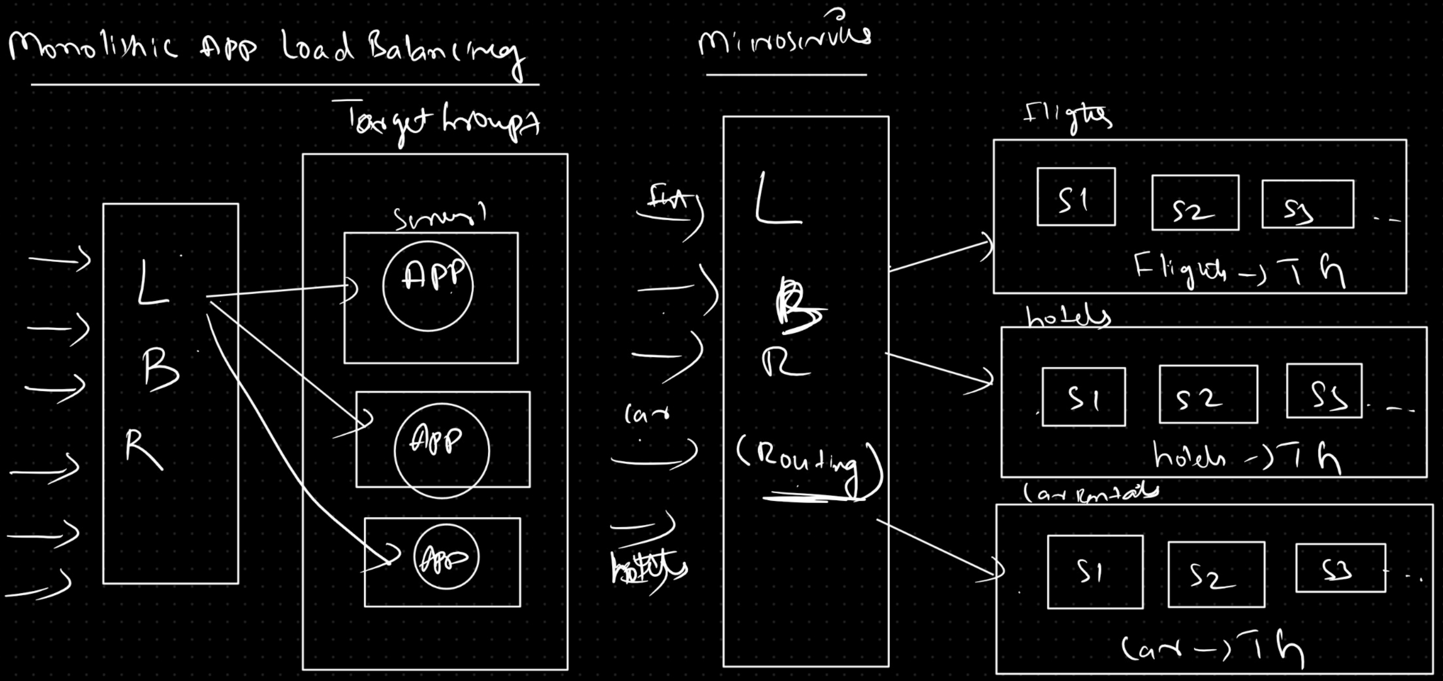
=> Monolithic vs Microservices

Monolithic --> Developing All functionalities in single application

For Monolithic app usually we need One Target Group

Microservices --> Dividing functionalities into Multiple apis (One App is divided into multiple sub / micro app)





=> Types of Load Balancers

-> Application Load Balancer (ALB)

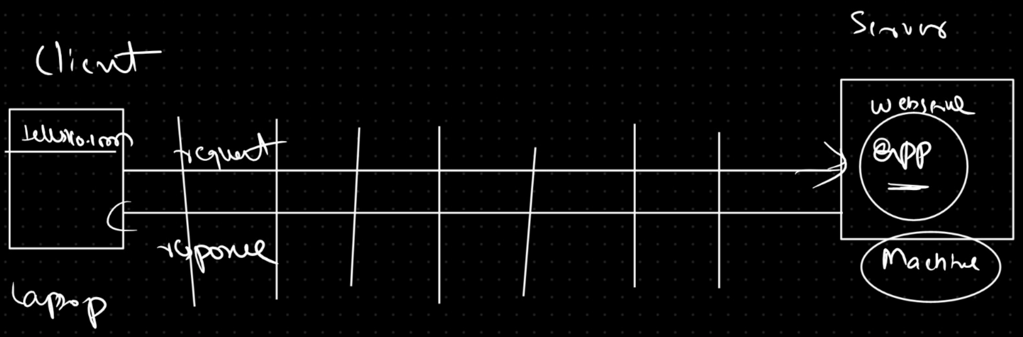
-> Network Load Balancer (NLB)

-> Gateway Load Balancer (GWLB)

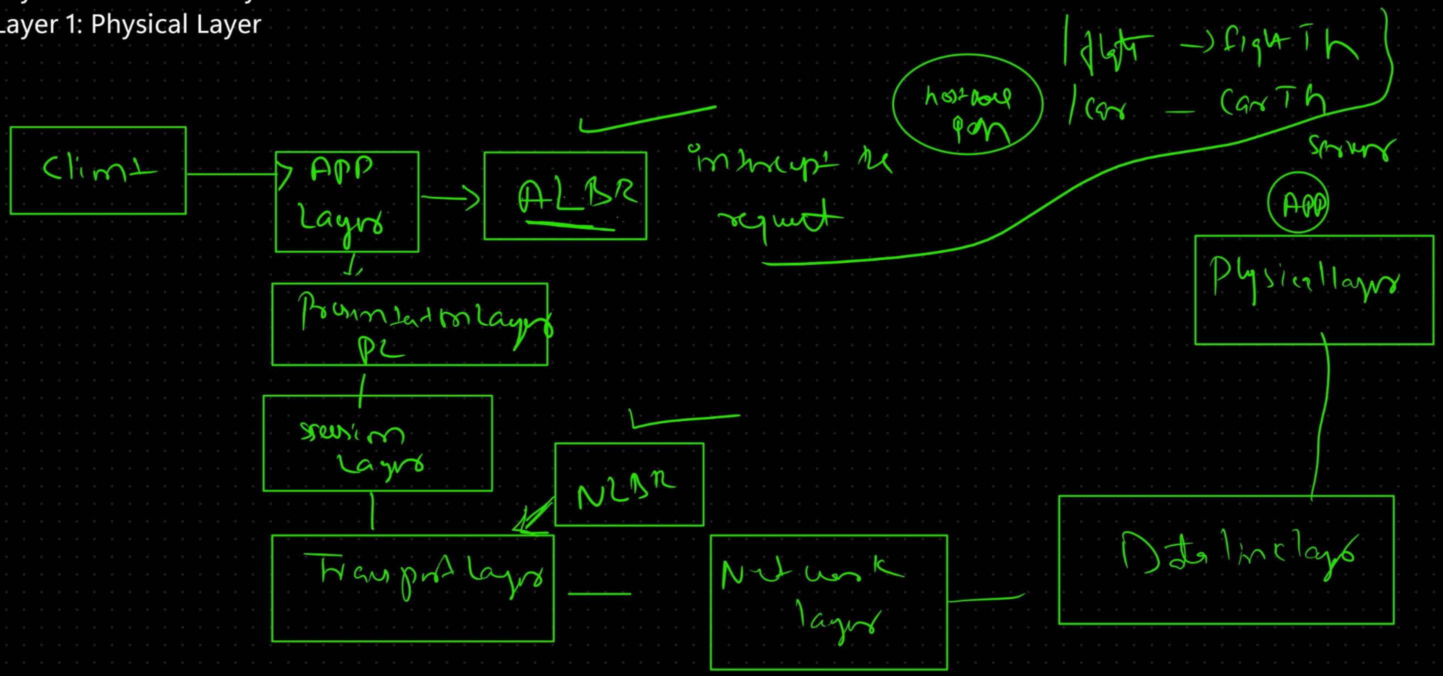
OSI model => Open Systems Interconnection

↓ 7 Layers

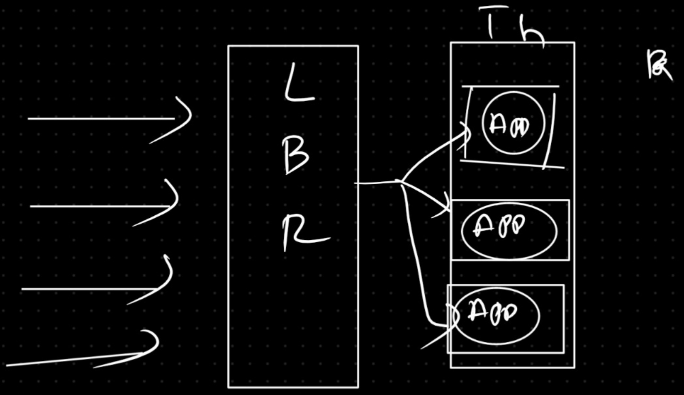
↓ represents how request will transfer from client to server.



Layer - 7 : Application Layer (ALB)
Layer 6 : Presentation Layer
Layer 5 : Session Layer
Layer 4 : Transport Layer(NLB)
Layer 3: Network Layer
Layer 2: Data Link Layer
Layer 1: Physical Layer



Application Load Balancer : Operates at 7(Application layer) of OSI Model
--> Designed to route HTTP and HTTPS traffic based on content (host based and path based routing) with HTTP Headers
--> Ideal for modern web application , Microservices and (Container based application)



Auto Scaling Groups

RPM => 10K => 3 servers

30 servers

1lakh => 30 servers

7 am -> 40K => No problem

9 am -> 80K RPM => No problem

11 am => 2L => Not capable to handle

⇓

Increased the servers 90 server

3 L RPM =>

12 PM => 2.5 L RPM => No problem

1pm =>

2 PM => 25 K RPM