*VOLUMES ATTACHING WITHOUT DOWNTIME(VOLUME SIZE INCREASING)

Ebs volume attaching without downtime 8gb to 13 gb

volumes

1.create ec2 instance

default 8gb

2.went storage options
volume id

3.open that volume id

4.then go to actions modify volume when it is okay state

5.(8g getting randomly for every volume)it will modify how much you want(13g) but not fully

6. now connect to the instance

7.use the commands

df -h details of volumes(only mounted volumes)

lsblk (full list of volume)
growpart /Device name
after that
you need to find the volume which file extension means xfs,ext4.
after u need use commands
xfs -growfs /devicename/volumename
ext4 resize2fs /device name/volume name.
now it will be mounted.
use command to show details
df -hT.

*CONNECTING GIT BASH TO AWS

create an ec2 instance
after creating an instance(health check 2/2) it will be created.
click on instance
click on connect
copy ssh client
past it in the git bash(if you don't have a git bash download it on a google)
asking permissions yes/no
type yes
it will be connected to git bash.

*AWS COST MANAGEMENT CALCULATION.

GO the webpage
type a url ========aws pricing calculation
click on official link (aws pricing calculator)
create estimate
add services
choose a location type
choose region
find service like ec2,elastic ip address
i taking ec2
click on amazon ec2 configuration
amazon ec2 configuration dashboard opening
description
localtype & choose a region
ec2 specifications like tenancy ,operating system,number of ec2 instances.
search an instance type like t2 micro, t2 nano,t2 medium, t3 medium.
instance family ==any instance family
    Vcpu's
    memory(gib)
    networking performance.
payment options.
click on payments options (what you want)

save and view summary (or) save and add service
my estimate summary
monthly, yearly/12 months.

***Create a backup and restore the instance and check if the data is available or not.**

create a instance

put some data on that ec2 machine

go to services and search a aws backup

go to dashboard
create on demand backup
settings
resource type like ec2, dynamodb etc like that
total retention period
create a backup vault or use default backup vault
if u have iam role use it otherwise default role use it
click on create on demand backup

create a backup plan
backup plan options is coming whatever you want use it
create a backup plan
backup rule configuration like schedule,backup vault backup frequency,backup window
choose the region

backup vaults

finally go to the ec2 instance and terminate that instance again without losing data.

*Bastion host

create two ec2 machines
one with public ip
second one is private ip
after completed ec2 machines creation
go to public ip machine and connect to instance & ec2 instance connect
and connect with private ip ssh-client on public ec2 machine
chmod 400 pem.file
it will be not connected means
you must add pem.file on that public ec2 machine
after you use a this command'
ssh -i pem.file user@private
It will be connected.

***Create a cloud watch alarm for data transfer once it starts the billing it should give notification**.

create a ec2 instance add some data on it

go to the services and search a cloudwatch and create a alarm
select matric like ec2 autoscaling whatever you're using for alarm.
but using an ec2 instance you add an ec2 instance id to make it easy to find your instance.
then specify metric and conditions. like metric name, instance id ,statistics,period.
conditions:threshold type
cpu utilization is greater or lower etc(whatever you cross a threshold time)
threshold value(mail getting when the cross the threshold time what you give value)

NOTIFICATION:
send a notification to an already existing sns topic or if you want a new sns topic you add it.

After that, put the alarm name whatever you want.
alarm will be created.

***Create an ec2 instance and transfer the file from local machine to instance**.

create ec2 instance

open a local terminal

connect this run command with local terminal (chmod 400 prasad.pem)

after connect you can create file or folder (already existing file or folder)
then use this command

```
scp -i /path/to/your/key.pem /path/to/local/file.txt ec2-user@your-ec2-ip:/path/on/ec2
```

then open your ec2 instance
enter into your path what you gave to to scp command
enter ls.
your file or folder is coming

*cross region replication(crr)

first create a two buckets with different regions
click on 1stbucket
go to management
click on replication rule
opening replication rule configuration
assigning replication rule name
status enabled
source bucket
choose the rule scope means apply which type of files
destination path
give the IAM role and save
go back to dashboard
click on the bucket which is asign a management access
upload a file
and next open the 2nd bucket then you find a data or file which is uploaded in 1stbucket.
this is the cross region replication

***Ebs volume attaching without downtime 8gb to 100 gb**

volumes

1.create ec2 instance

default 8gb

2.went storage options
volume id

3.open that volume id

4.then go to actions modify volume when it is okey state

5.it will modify but not fully

6. now connect to the instance

7.use the commands

df -h details of volumes(only mounted volumes)

lsblk (full list of volume)
growpart /Device name
after that
you need to find the volume which file extension means xfs,ext4.
after u need use commands
xfs -growfs /devicename/volumename
ext4 resize2fs /device name/volumename.
now it will be mounted.
use command to show details
df -hT.

.

*EBS VOLUMES ATTACH AND DETACHED.

create an ec2 instance
select that instance and open that instance
click on storage
volume id
device name
volume size
attachment time
etc
if you want to add or attach a volume
go to ec2 dashboard
click on volumes
create a volume
volume type=gp2,gp3,gp1,etc whatever you want you select it
how much volume you attach(gib)based on volume type min and max is there.
size is between the volume type min and maximum.
must and should carefully check the availability zone
tags are optional
then click on create volume
now your created volume is in which state
when you get a availability zone
go to actions
click on attach
volume will be attached as soon as possible
using some commands to mount the volumes
after mount you don't want a that attached volume
use some commands
click on that you had created volume and go actions
click on attach the volume will be detached as soon as possible.
if you delete that volume
go to actions delete the volume.

***ELASTIC IP ALLOCATE AND DEALLOCATE**

ELASTIC IP ALLOCATION
create an ec2 instance
go to ec2 dashboard
click elastic ip
click on allocate elastic ip address
tags is optional (if you want add it otherwise no need)(if you add tags any difficult times you find easily based on tag name)
click on next or allocate
elastic ip address allocated successfully
showing a elastic ip address
now select that elastic ip address
go to actions
now click on associate elastic ip address
now click on instance
choose instance and choose id address
click on associate
elastic ip address associated successfully

DIS-ALLOCATE

go to ec2 dashboard
click ON elastic ip
Click on actions then now click on disalocate ip address
successfully dislocated then again go to actions and released elastic ip address
finally click on release
elastic ip address released.

* gunicorn configuration

login aws console
create an ec2 instance
Connect using the EC2 Instance Connect (or) connect with ssh client.

sudo apt-get update (or)apt update
sudo apt-get install gunicorn (or)apt install gunicorn

you want to open a /etc/systemd/system/--------path of configuration
cd /etc/systemd/system.
open that path
/etc/systemd/system.----sudo vim /etc/systemd/system/gunicorn.service

```
[Unit]
Description=Gunicorn service for My Flask App(or)name
After=network.target

[Service]
User=ubuntu or any thing
Group=www-data
WorkingDirectory=/path/to/your/app (or)present working area
ExecStart=gunicorn --bind port number (name of the gunicorn project (or)description what you given).wsgi

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl start gunicorn
sudo systemctl status gunicorn
sudo systemctl stop gunicorn.
```

*

HOST A WEBSITE USING A GITHUB

first create a github account

click on NEW option
create a new repository
enter repository name

public
private
based on project

initialization this repository
ADD A README FILE
add .gitignore
choose the license
click on next
opening what the repository is your created
click on files
upload files
create a new file

creating a new file with filename(htmlfile.name with .html,python is name with .py)

write code of html(optional)

click on commit changes

now, open settings on the your repository

in the general settings

put a branch where is your created a file like main ,master
click on save
finally getting website
click on that website
getting data output that you can create on that file.

*HOST THE WEBPAGE OR WEBSITE USING EC2 INSTANCE

sign up for aws
launch an ec2 instances
connect to the ec2 instance
install a web server
linux=sudo yum update -y
      sudo yum install httpd -y
ubuntu=sudo apt update
       sudo apt install nginx -y

upload your webpage
configure your web server
start your webserver
httpd=sudo service httpd start
nginx=sudo service nginx start

access your website
domain name (optional)

commands

```
sudo -i
yum update -y
yum install httpd -y
mkdir temp(optional)
cd temp(optional)
wget (downloaded link)
ls -lrt
downloaded file is zip file means
unzip (zip file name)
cd unzip file name
mv * /var/www/html
cd /var/www/html
ls -lrt
systemctl enabled httpd
systemctl start httpd
```

this is a web page that's why when you create an instance you must be given a port number 80.

*IAM ROLES AND ACTIONS

login in aws account
go to services
search a IAM(identity management access)
click on IAM
IAM dashboard are opening
USER GROUPS
USERS
ROLES
POLICIES

USER GROUPS:
create a groups
name of the group
add users to group(optional)
attach permission policies to that group
there is only 10 policies to this user group(optional)
the users in this group will have permissions that are defined in the selected policies.

USERS
create user
create a user name
provide a user access to the aws management console

user types like
specify a user in identity center
i want to create a IAM user

want you want you selected
you want a own credentials you want to select
i want to create a IAM user
asking console password
its means
auto-generated password
custom password
what you want click on that option.
then click next
set permissions
permissions options like
add user to group
copy permissions

attach policies directly
permission policies to user
add some policies what your required for that user
tags are optional
click on createuser
user successfully created
retrieve password
console sign in details
when you download .csv file
now open that download .csv file
there are credentials of that user to login user aws account.
you enter the autogenerated credentials
After that you retrieve the new credentials that you want.
After login you only access the policies that you are getting.

ROLES
click on create a role
Trusted entity type
what you want click on that entity type
use case ore services
now click on next
in that role we have to add permission and click on next
assign a rolename how you want
tags are optional
now click on create roles
finally role is created

policies
click on create policies
select a service (which service you want)

now click actions allowed (allow / deny)
manual actions like read,list,write etc
resources must needed
click on next
create a policies name
tags are optional
click on create policy
you want to see that policy by using filters in custom management.

*INSTALL JENKINS AND DOCKER

create an ec2 instances

click on connect when instance is 2/2.

connect using ec2 instance connect
(or)
connect through the ssh -client in git bash
first went to root through
sudo -i
#apt update
# apt install openjdk-11-jre-headless (you want particular version means mention version also)
(or) type java and press enter (showing some commands with versions)

jenkins commands
use commands for official website of jenkins

```
#sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

#echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null

#sudo apt-get update

#sudo apt-get install jenkins
```

jenkins will be installed
give the security port number 8080
copy ip address and search in web browser with ip address:8080
Unlock Jenkins
use this command in terminal
cat /var/lib/jenkins/secrets/initialAdminPassword
press enter getting this credentials
c328d6806eec49df842701cbe71d48f4 (this is enter into jenkins unlock administration page)
after asking install plugins
you will selected means started some basic plugins
after that create first admin user
username=jenkins(optional)
password=jenkins(optional)
confirm password=jenkins(optional)

full name=jenkins(optional)
email address=your mail
after that instance configuration
getting jenkins urls (http://3.144.138.205:8080/)
Jenkins is ready.
use this commands

docker
apt update
apt install docker.io (optional)based on what you want
docker install completed.

systemctl start jenkins/docker
systemctl stop jenkins/docker
systemctl status jenkins/docker.

*ROUTE 53

login aws console
go to services
search route53
click on route53
go to dashboard
registered domains
do you any registered domains you can use it
you don't have any register domains
click on hosted zones
create hosted zone
opening hosted zone configuration
create a domain name with .com
type is public or private
tags(keys and values)
click on create hosted zone
hosted zone is created with domain name
now create a record
click on records
there is some route policies
like simple routing
weighted route
geolocation
latency
failover
Multivalue answer
i click on simple routing
next click on define simple record
subdomain name
record type (A ,AAAA,CNAME)ETC
value/tute traffic to which one you want like ip address etc
assign a ip number
ttl seconds(time to alive) its means how much time do you want to alive that record
click on next
click on create a record
coming this name servers
ns-175.awsdns-21.com.
ns-1986.awsdns-56.co.uk.
ns-1237.awsdns-26.org.
ns-619.awsdns-13.net.

its wills be connected with godaddy or other domain registrar
hosted domain is created.....

*S3 BUCKET

login aws account
go to the services
search s3 bucket
click on s3 bucket

general configuration
aws region(optional)
bucket name (optional)
choose bucket(optional)already exist bucket

object owned
acls disabled(recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.is specified using only politics.

bucket versioning
disable
enable(is there is any update its is use full)
then click on create bucket a bucket
click on your bucket
click on upload

add a files to that bucket
files will be added successfully after selecting that bucket and going to actions click on share with a pre signed url (it gets a link to saw data on that bucket).

after asking a interval time(mandatory)
minutes(optional)
hours(optional).
copy presigned url.

*SNS

login aws account
go to services
search a sns (simple notification service)
showing sns topic
click on sns
now opening sns dashboard
click on topic
topic dashboard are opening
give some credentials like name,display name.
click on create topic
then create a subscription on that topic (arn)
protocol ===email,http,https,sms,etc.
endpoint (based on your protocol)
click subscription
you do conform it through your protocol type

now you need to get a notification
now create a bucket in s3
upload a files
before uploading files you need to give some credentials
go to properties
click on events
give some credentials like eventname,prefix-suffix(optional)
eventtypes ,object removal,object restore,object acl,and some credential what you want
destination
choose the destination to publish the event.like sns ,lambda functions etc
take sns means asking
choose sns topic
enter sns topic arn
click on save changes
then upload a file
then you get notification of the protocol you took.

*** vpc peering and internet gateway**

login aws account
go to services and search a vpc
create vpc(virtual private cloud)
click on vpc
create vpc
resources to create
name tag
ipv4 CIDR block
ip46 CIDR block
tag
created vpc successfully

internet gateways
click on that internet gateway
create internet gateway
internet gateway settings opening
name tag
tag is optional
click on create internet gateway
internet gateway is created
internet gateway is now attach to vpc
click on attach a vpc
attach to vpc
select a vpc on available vpcs
and attach internet gateway
vpc is successfully attached to the internet gateway.

*FILES TRANSFER FROM LOCAL SYSTEM TO GIT BASH USING GIT

Download git from the internet.
create a github account
create a new repository

open git bash
which file is you want to transfer from local to github repo
change directory(cd) open that path

use this commands
git init-----initializing a new git repository in the current directory
git add (filename)---add a specific file to the staging area
git add .----adds all the data to the staging area
git status ----shows the current state of your repository including tracking files,untracked files,modified files and branch files.
git commit -m—---the commit message line--creates a new commit with the changes in the staging area and specifies
git remote add origin (your github repository http path)
git push
enter.

***IMAGE COMMANDS**

Build an Image from a Dockerfile
docker build -t <image_name>

Build an Image from a Dockerfile without the cache
docker build -t <image_name> . –no-cache

List local images
docker images

Delete an Image
docker rmi <image_name>

Remove all unused images
docker image prune

DOCKER HUB

login into docker
docker login -u <username>

publish an image to dockerhub
docker push <username>/<image_name>

search hub for an image
docker search <image_name>

pull an image from a docker hub
docker pull <image_name>

GENERAL COMMANDS

start the docker daemon

```
docker -d
```

get help with docker can also use - help on all subcommands

```
docker --help
```

display system-wide information-----docker info

CONTAINERS

create and run a container from an image with a customname

```
docker run --name <container_name><image_name>
```

run a container with and publish a containers port to the host

```
docker run --name containername -d -p hostport number:80 <imagename>
```

run a container in the background

```
docker run -d <image_name>
```

start or stop an existing container

```
docker start/stop <container_name>
```

remove a stopped container

```
docker rm <container_name>
```

open a shell inside a running container

```
docker exec -it <container_name>sh.
```

fetch and fellow the logs of a container

```
docker logs -f <container_name>
```

to inspect a running container

```
docker inspect <container_name>
```

to list currently running container

```
docker ps
```

list all docker containers (running and stopped)

```
docker ps --all
```

view resources usage stats

```
docker container stats.
```

*** Calculate the elastic ip charges if it's not running for 12 hours in a month.(approx)**

go to webpage
and search a aws price calculator
click on create estimate
click on find service and search elastic ip
getting amazon elastic ip then click on the configure
configure amazon elastic ip
enter the description for your estimate
choose the location
service settings like number of ec2 instances
number of elastic ips per instance
Number of hours each EIP is attached to an EC2 instance =====unit
Number of hours each EIP is associated with a stopped instance or is unattached
Number of EIP remaps
save and view summary
save and add the service you want to add a service.

*TOMCAT INSTALLATION

create an ec2 instance
give a port to that instance----8080
8080 is a default port for tomcat
connect to git bash (or)connect with putty ....

1) sudo -i
2) apt-get update
3) apt-get install default-jdk -y
4) java –version
5) cd /opt
6) wget
http://mirrors.fibergrid.in/apache/tomcat/tomcat-8/v8.5.35/bin/apache-tomcat-8.5.75.tar.gz
7) tar -xvzf /opt/apache-tomcat-8.5.75.tar.gz
8) ls
9) mv apache-tomcat-8.5.75 tomcat
10) ls
11) cd tomcat/
12) cd bin
13) ./startup.sh
14) go to google chrome
15) enter the instance ip and port number
16) tomcat will be opening
17) Find / -name context.xml

```
/opt/tomcat/conf/context.xml

/opt/tomcat/webapps/host-manager/META-INF/context.xml

/opt/tomcat/webapps/examples/META-INF/context.xml
vi /opt/tomcat/webapps/examples/META-INF/context.xml
```

```
<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager
sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
```

```
          /opt/tomcat/webapps/manager/META-INF/context.xml
       Vi opt/tomcat/webapps/host-manager/META-INF/context.xml

 Vi opt/tomcat/webapps/host-manager/META-INF/context.xml
```

```
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
       allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
<Manager
sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\
.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap
"/>
</Context>
```

18) context.xml files are opening
19) open vi editor with context.xml files
20) edit that files
21) after that open cd conf
22) open tomcat-user.xml

```
    <role rolename="manager-gui"/>
      <role rolename="manager-script"/>
      <role rolename="manager-jmx"/>
      <role rolename="manager-status"/>
<user username="admin" password="admin" roles="manager-gui, manager-script,
manager-jmx, manager-status"/>
<user username="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

22) give the user credentials like (password user id)
24) tomcat are get started (ready to work)

* Nginx configuration

Create an ec2 instance
Install a nginx on that instance
Commands
Apt update
Apt install nginx -y
After install
Check status
Systemctl start nginx
Systemctl status nginx
Systemctl stop nginx

Now start the configuration
Connect to any terminal
After that
You need to create a block files
/var/www/html/—nginx path
Create a directories
After that
Cp /etc/nginx/sites-available/default /etc/nginx/sites-available/directoryname.conf
Cp /etc/nginx/sites-available/default /etc/nginx/sites-available/directoryname2.conf
Any editor open
Vim /etc/nginx/sites-available/directoryname.conf
Edit =server–root—-servername.like that edit what you want
Vim /etc/nginx/sites-available/directoryname2.conf
Edit =server–root—-servername.like that edit what you want
Remove the block files
rm /etc/nginx/sites-enabled/default
Enable the newly created block files
Ln -s /etc/nginx/sites-available/directory1.conf /etc/nginx/sites-enabled/
Ln -s /etc/nginx/sites-available/directory2.conf /etc/nginx/sites-enabled/
Nginx -t
Test is successful after going to next step otherwise work where you get failure
Systemctl restart nginx
 Vim /etc/hosts
Edit what you want that text editor means localhost ,give instance ip address etc….
Save it
Go to webpage give a ip (or)server name [domain name]
Nginx configuration is completed…….

*** linux commands basics**

ls ----list the files
ls -l------long listing of file
ls -l(filename) -----long listing of the particular file
ls -a shows hidden files too
ls -lr----- listing of all files and directories in reverse order
ls -R shows the list of the files in the tree structure

creation of files

cat(concatenate)
touch
vi editor

cat:-
cat > filename
save with ctrl+D

edit a old file on cat (or)add a additional data with
cat >> old filename

touch(file1)(file2)(file3)---- to create empty files only.

vi editor
vi (file1)
to add a text press i(insert)
save
esc:wq
esc:x
esc:se nu ---to give the numbers for the context.

creating directories
mkdir(dir)----single directory
mkdir(d1) (d2) (d3) multiple directories
mkdir -p d1/d2/d3 ------nested directories

navigate
cd--change the directory
cd ..------1 step back
cd ../.. ---two steps back
cd - last working directory
cd (or)cd~ ----jumps to home directory

remove
rm(filename)---to delete a file
rmdir(filename)----to delete a directory
rm -rf to delete whole directory with full of files

copy
cp(file1)(file2) ------file to directory
cp(file1)(file2)-------file to file
cp -r(file1)(file2)------directory to directory

rename
mv(file1)(file2) --------rename
mv(newname)(oldname)-------move

grep commands
grep -v (word) (filename)-------delete that worldline
grep -r (word) (filename)-------output same worldline
grep -i (word) (filename)-------complete common info

find----find(word)*------searching a word

filter commands
head
tail
sort
cut
sed

head(filename)--show default top 10 lines
head -n (filename)--show n=no of lines

tail(filename)--show default bottom 10 lines
tail -n (filename)--show n=no of lines

sort(filename)----show the alphabetic order
sort -r(filename)-----show the reverse order

cut -b 1,2,3(filename)------cutting the letters of the word...

sed 's/word/replace/g'(filename)------replacing the word..

users and group

adduser (filename)
security credentials and personal details

usermod -u (uid) (filename)---to change the userid
groupmod -g (gid) filename-----to change the group id

useradd -u (own user id)------create user with own uid
cat /etc/passwd

rename the user
usermod -i (newname) (oldname)

delete user------userdel (username)

groupadd (group name)
groupadd -g (gid)(group name)-----create group with own gid

groupmod -g (own gid)(group name)----change the gid
groupmod -n (gid)(new groupname)-----rename group

add users to the group
gpasswd<option><arguments>
<options>=M---to add a multiple users
    a---to add a single user
    A---to add admin to the group
    d---to delete user from the group
<arguments>(filename or person name).

*GIT COMMANDS

Git branch(branchname)---to create branch
Git branch—----to show the branch
Git checkout(branchname)--to switch one branch to another branch
Git checkout -b (new branch name) paste head Id.
Git merge(branchname)--merge two branches
Git branch -D(branchname)--to delete the branch
Git branch -m(new branch rename)----rename the branch
Git reflog—-to restore the deleted branch
Git cherry-pick—merging two branches but if you want to merge only particular commit with this cherry pick
Git reset–soft(local -staging)
            –mixed(staging - working directory)
           –-hard(complete reset) commit id is head==delete upper commits.
Git revert(commit id)---it increases the commits and deletes the data
Git stash–it is a temporary area under git.
            –to store temporary files,we will use stash
Git stash save—--to save to commit
Git stash list—--show lists
Git stash apply(id)--its apply the stash and coming to working directory
Git stash pop—it delete stash in the stash list like cut and paste
Git stash drop—it drops all the stashes
Git stash clear—it clears only applied stashes
Git log–history
Git log –n(to show the latest commits)
Git log –oneline(to display all commits in online)

Git configuration and setup commands
Git config –global user.name" github username"
Git config –global user email" github email id"

*Load Balancer:-

create Load balancers
select the loadbalancer type
create on what you want to use

basic configuration
load balancer name
scheme (it cannot be change after loadbalancer create
ip address type

network mapping---The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.
vpc---select vpc(virtual public cloud)
mapping in which zones(zones must be select above 2 zones)
security groups---if you want create a new one if you create other wise use a old security group(existing)(or)default

Listerners and routers
like protocol,port etc
you need to create a target group
create a target group
basic configuration
choose the target like(instances,ip address,lambda functions,application load balancer)
target group name
protocol and port
ip address type
like vpc,ipv4,ipv6
protocol versions like http1 and http2(whatever you want you use it)
click on next
register targets like available instances
select the instances which are you need use
create a target group.
based on your project you need to give some credentials
finally check the summary of your credentials.

1.Log in to the AWS Management Console:
2.Navigate to EC2:
3.Create Instances:
4.Create a Target Group:
5.Create a Load Balancer:
6.Configure Listener Rules:
7.Configure Security Groups:
8.Update DNS (if needed):

9.Test the Setup: