1. Create Ec2 instance and launch
2. Create an folder and put downloaded key in that folder
3. Open gitbash
4. Enter command which will be available on ec2 instance

Ex: ssh -i "new.pem" ubuntu@ec2-34-220-164-93.us-west-2.compute.amazonaws.com

1. Install java : apt-get install openjdk-11-jre openjdk-11-jdk
2. Install tomcat : apt-get install tomcat9
3. To start and stop tomcat – service tomcat9 start/stop
4. Install mysql : apt-get install mysql-server
5. To create password Run command : mysql\_secure\_installation
6. Install ftp client on local system to upload war file on cloud
7. Open ftp client
   1. Click on file
   2. Click on site manager
   3. Click on new site
   4. Click on protocol select sftp
   5. Enter server ip and port 22
   6. select logintype as key file
   7. select file and click on coonect
   8. upload file path: /var/lib/tomcat9/webapps
   9. grant write permission run command : chmod777 /var/lib/tomcat9/webapps

**Cloudfront**: Cloudfront use to reduce latency

**Edge-Location**: edge is kind of server and it has edge cache memory

Ex:

1. User hit url [www.example.com](http://www.example.com)
2. Dns will navigate user to the edge location not origin server
3. If edge server has data it will not hit the main(origin)server and serve the request and if not available then it will hit main server and keep data in memory then it will serve

**Regional Edge-Location**: regional edge location is above edge location if edge location delete it’s

data then it will not directly hit the origin server edge location will ask to the regional edge location

Billing on:

1. Data transfer out
2. http or http request
3. invalidation request “delete data from edge location”
4. SSL certificate

Route53:route53 is global service. route53 is use for dns management and following point.

1. DNS management
2. Traffic management
3. Availability management chek availability all connected resources are working fine ex:(s3bucket, load balancer …)
4. Domain Registration

Aws provide two kind of two domain

1. Generic(.com.in.net)
2. Geographic(.tz,.pk.uk)

**Autoscalling:**

To auto increase and decrease server then we go for Autoscalling

1. Scale up will be horizontal same type instances will be created ex: if you have 2gb server then 2gb server will be created
2. Cost reduce

Component:

1. Launch configuration
   1. Which type of instance you want to create when scale out ex: 2gb ram 4gb gpu etc
2. Autoscalling group
   1. Minimum or maximum instance will be created
   2. There should be max limit because of some error multiple instance will be created that’s why max limit should be add
   3. Auto scale will send ping to check instance
   4. Running instance only can be add in group
   5. Only one instance can be add only in one group.one instance cannot be add in multiple group
3. Scaling policy
   1. When my instance will create ex: my cpu utilization 90% then will create new instance
   2. When my instance will decrease ex: my cpu utilization 50% then will decrease instance
   3. We add availability zone