1. URL for repo:
   https://github.com/edellhou/CS6650/tree/main/Assignment3

2:

(a) Database Design:
Database chosen is Redis. It is a key-value based nonrelational database.
The data models I design can query following request.

- "For skier N, how many days have they skied this season?"
- "For skier N, what are the vertical totals for each ski day?" (calculate vertical as liftID*10)
- "For skier N, show me the lifts they rode on each ski day"
- "How many unique skiers visited resort X on day N?"

There are 2 data models. The first model can query request 1-3, and the second data model can query request 4.

First Model: Redis Sets are used. Key: SkierID:dayID, Value: liftID

Assumption made: The season in our project is fixed, which is 2022, so I didn't use season as part of the Key. I know dayID is also fixed, which is 1. But request 2-3 explicitly wanna know the skier info by each ski day, so I made dayId as part of the key.

Second Model: Redis HyperLoglog is used.

HyperLogLog is good at this since you get great performance at low computation cost, and a small amount of memory. HyperLogLog can be used in various situations like;

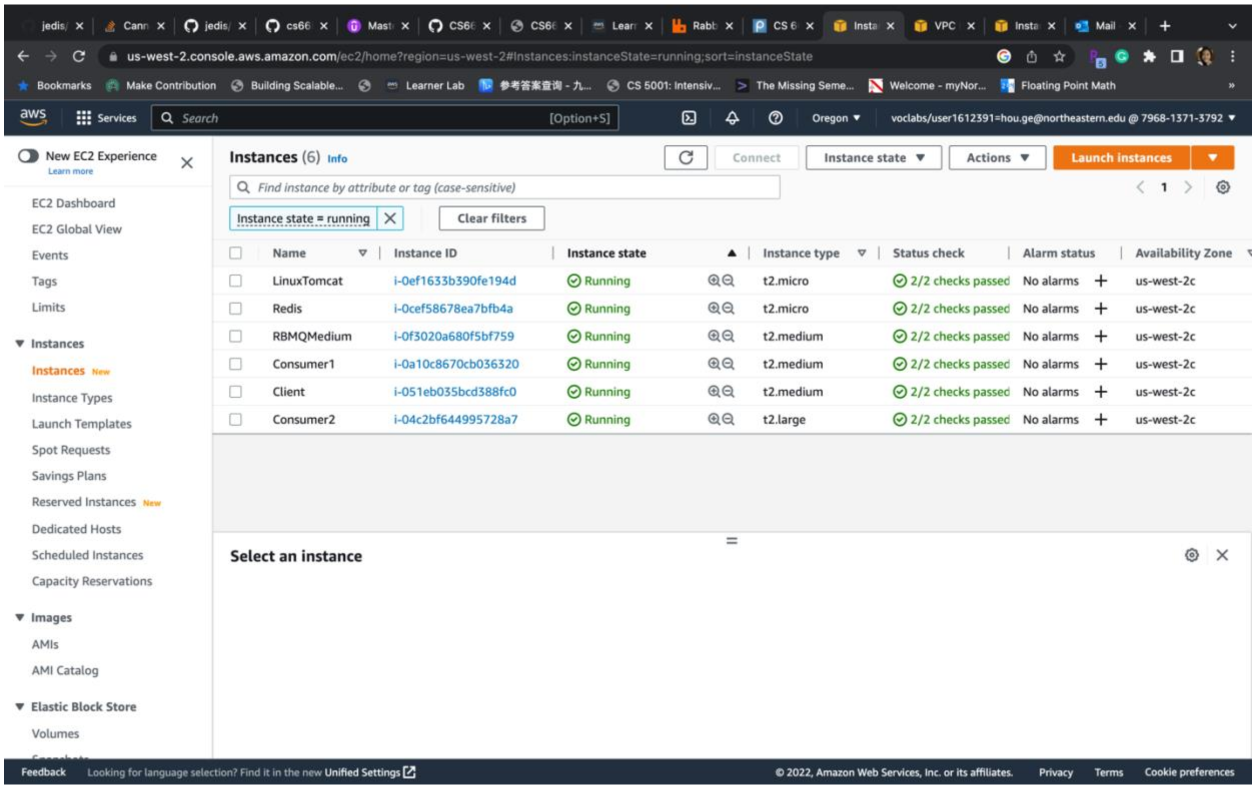
- Counting unique visitors
- Creating unique in a book or books
- Keeping best stocks of all times
- Create unique names of a products, services, category
- A situation where you are not worries about the elements of the datasets, but its counts
- Create unique names of students in a class and many more.

Key:resortID:dayID, value:skierID

Eg: say at resort 3 on day 1 which is the key, skier no.10 skied 5 times, skier no.100 skied 2 times, the hyperloglog datatype will return 2

(b) Deployment topology.
    6 servers are deployed.
    1 for client
    1 for Tomcat
    1 for RBMQ
    2 for consumer
    1 for Redis Database

Picture below shows the instance type



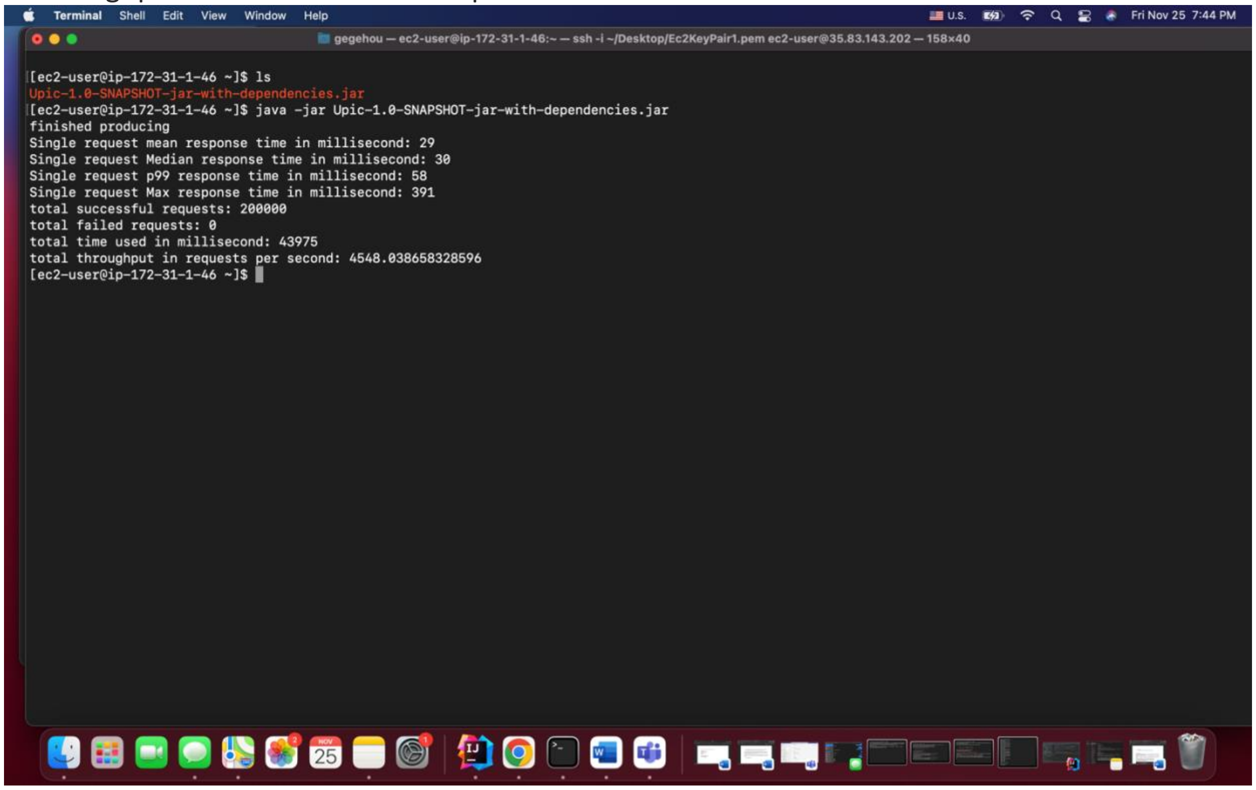
Instances (6) Info

Instance state = running

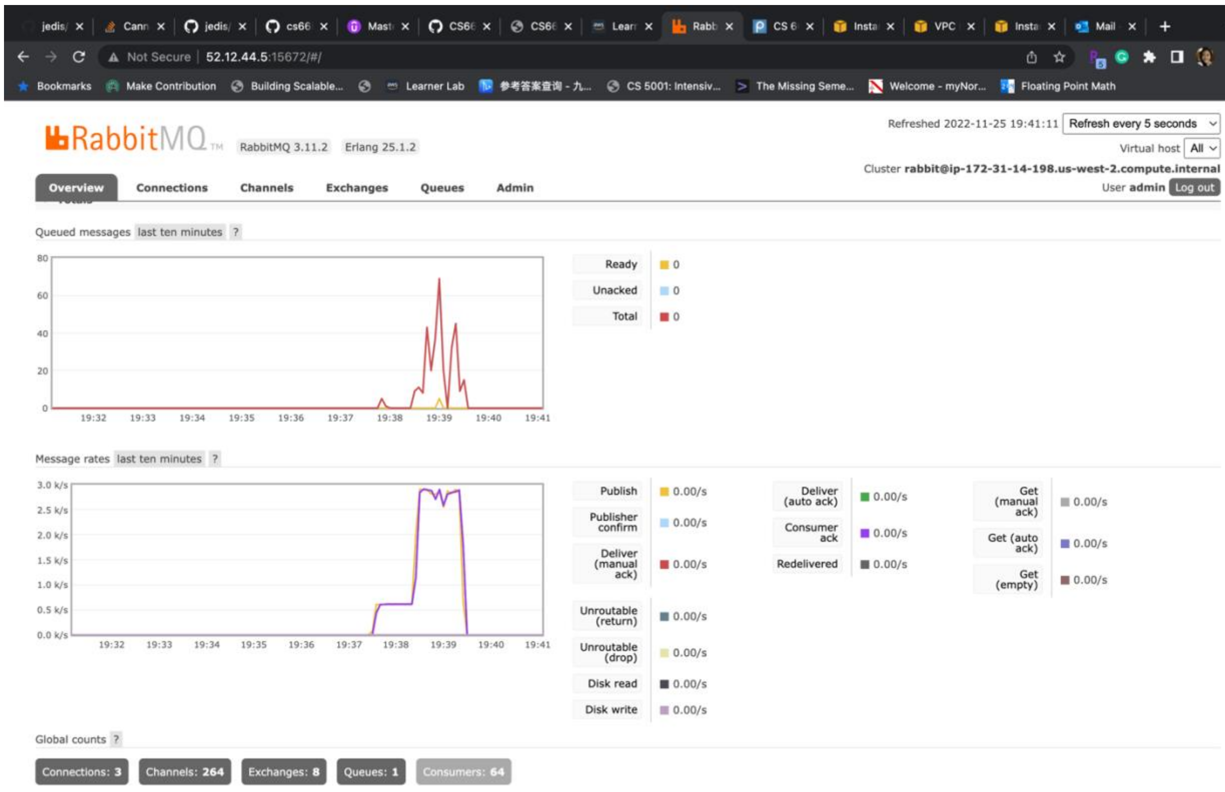
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
LinuxTomcat	i-0ef1633b390fe194d	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
Redis	i-0cef58678ea7bfb4a	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
RBMQMedium	i-0f3020a680f5bf759	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Consumer1	i-0a10c8670cb036320	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Client	i-051eb035bcd388fc0	Running	t2.medium	2/2 checks passed	No alarms	us-west-2c
Consumer2	i-04c2bf644995728a7	Running	t2.large	2/2 checks passed	No alarms	us-west-2c

Select an instance

3:Throughput and RBMQ console clips


Terminal Shell Edit View Window Help
Fri Nov 25 7:44 PM
gegehou — ec2-user@ip-172-31-1-46:~ — ssh -i ~/Desktop/Ec2KeyPair1.pem ec2-user@35.83.143.202 — 158×40

```
[ec2-user@ip-172-31-1-46 ~]$ ls
Upic-1.0-SNAPSHOT-jar-with-dependencies.jar
[ec2-user@ip-172-31-1-46 ~]$ java -jar Upic-1.0-SNAPSHOT-jar-with-dependencies.jar
finished producing
Single request mean response time in millisecond: 29
Single request Median response time in millisecond: 30
Single request p99 response time in millisecond: 58
Single request Max response time in millisecond: 391
total successful requests: 200000
total failed requests: 0
total time used in millisecond: 43975
total throughput in requests per second: 4548.038658328596
[ec2-user@ip-172-31-1-46 ~]$
```



Refreshed 2022-11-25 19:41:11 Refresh every 5 seconds

Virtual host All

Cluster rabbit@ip-172-31-14-198.us-west-2.compute.internal

User admin Log out

Queued messages last ten minutes ?

Ready	0
Unacked	0
Total	0

Message rates last ten minutes ?

Publish	0.00/s	Deliver (auto ack)	0.00/s	Get (manual ack)	0.00/s
Publisher confirm	0.00/s	Consumer ack	0.00/s	Get (auto ack)	0.00/s
Deliver (manual ack)	0.00/s	Redelivered	0.00/s	Get (empty)	0.00/s
Unroutable (return)	0.00/s				
Unroutable (drop)	0.00/s				
Disk read	0.00/s				
Disk write	0.00/s				

Global counts ?

Connections: 3 Channels: 264 Exchanges: 8 Queues: 1 Consumers: 64