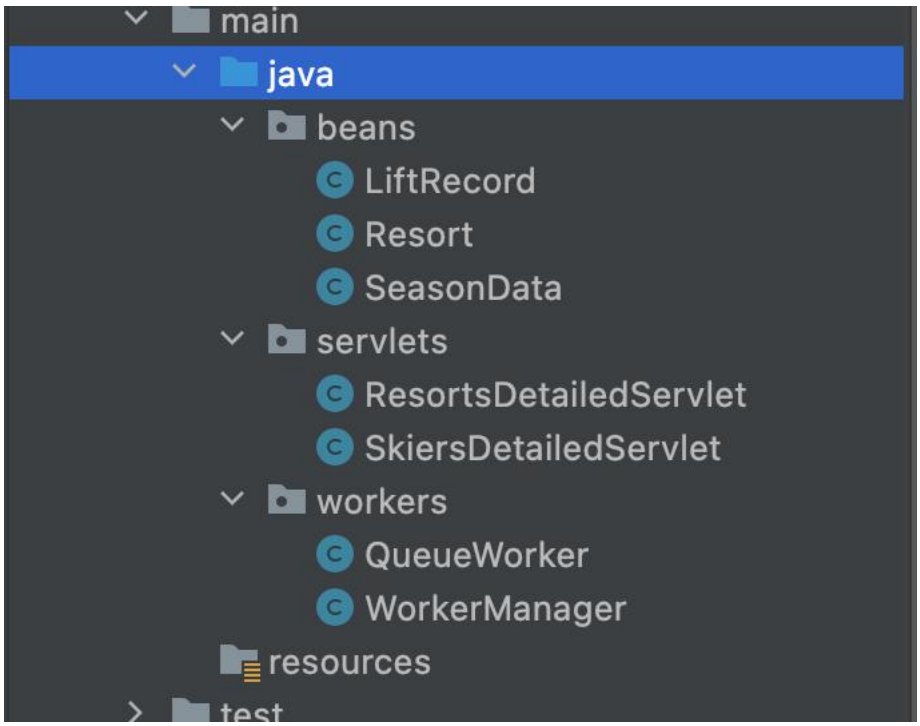
Assignment 2 BSDS

Saiqi He 001096811
URL to github: https://github.com/saiqi1999/BuildingScalableDistributedSystem

1. Server design

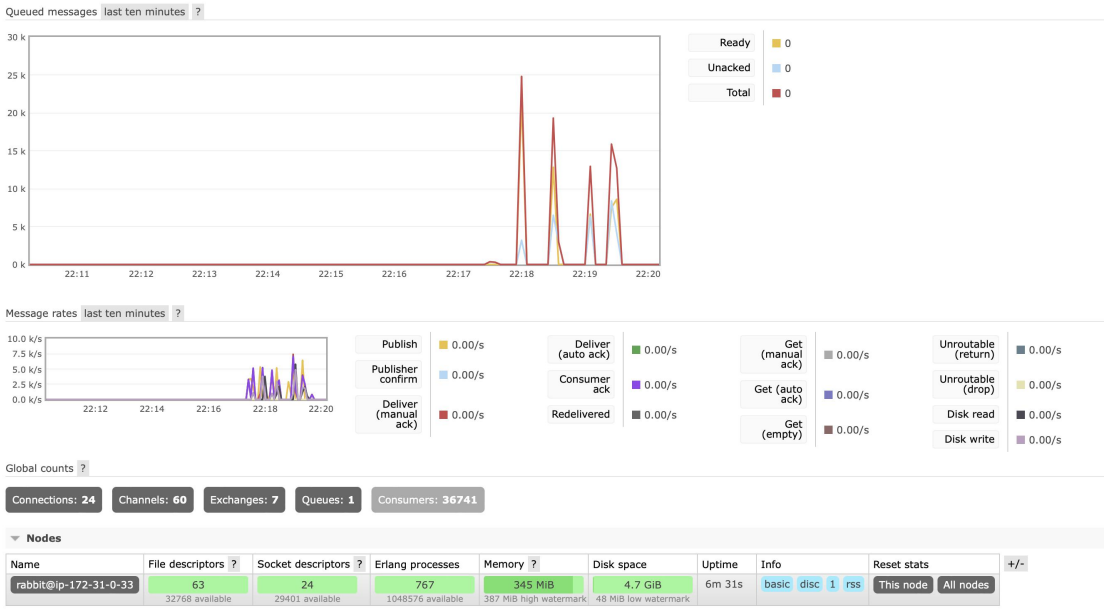


There are some classes in my server, I deleted some of the servlet from HW1 since they are not used.
The classes in “workers” package is important, it’s related to how I send and receive the messages.
Basically I started a cluster with 4 EC2 instances running my server, each has one connection and several channel related to rabbit MQ.
When here comes a request from client, server will have asynchronous process to validate the request, acknowledge the request and try to generate a message which contain all the info with a LiftRecord in to the rabbit MQ queue “From_Workers”
This process is controlled by a singleton “WorkerManager”, which has a list of Runnable “QueueWorker”, WorkerManager will invoke QueueWorker in a round-robin fashion.
Let’s be clear there will be many threads running simultaneously on this server and they will go to different QueueWorker, controlled by the same WorkerManager. WorkerManager will be implemented when it’s first called. Therefore they are using

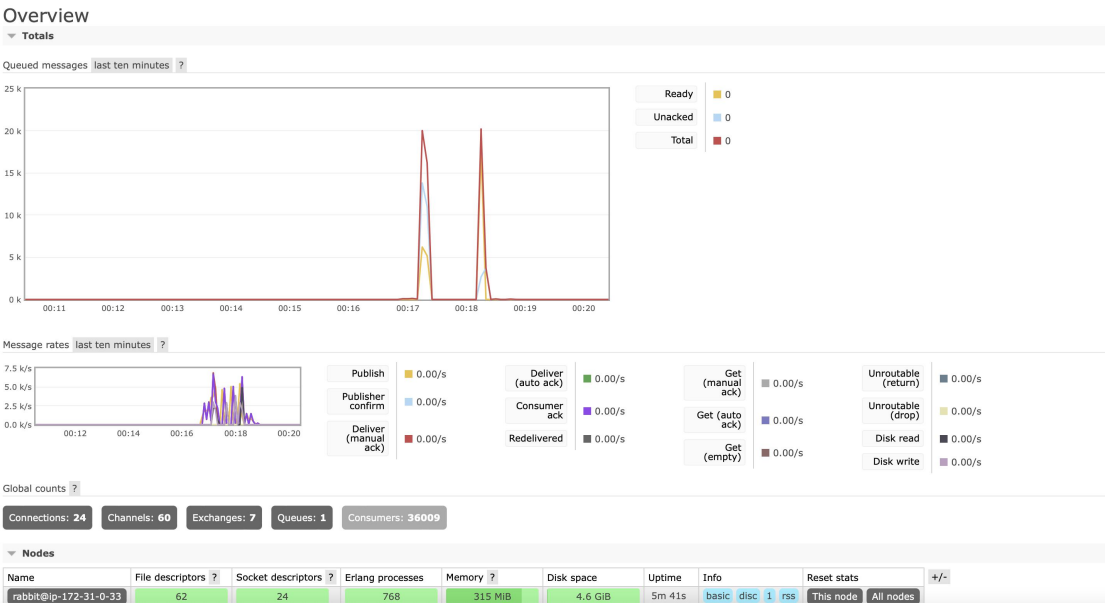
different channels.
Then the multi-thread consumer, which I know it will be deployed on server but now I run it locally, will connect to this broker with some connection, each connection a thread. Each connection will have a channel and consume the messages in “From_Workers” queue. Put them in a concurrent hash map, which is built thread safe.

2. test runs

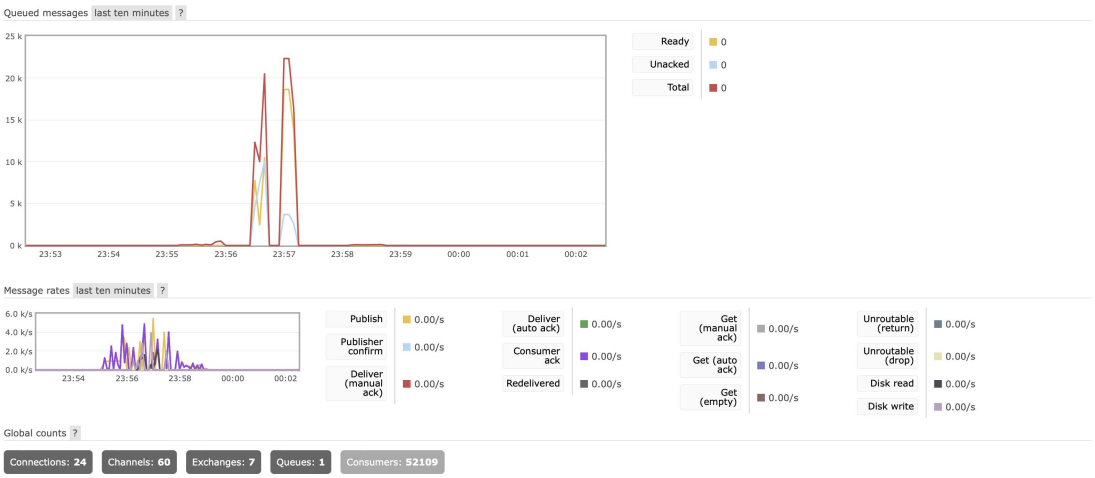
Successful run of 256 threads:



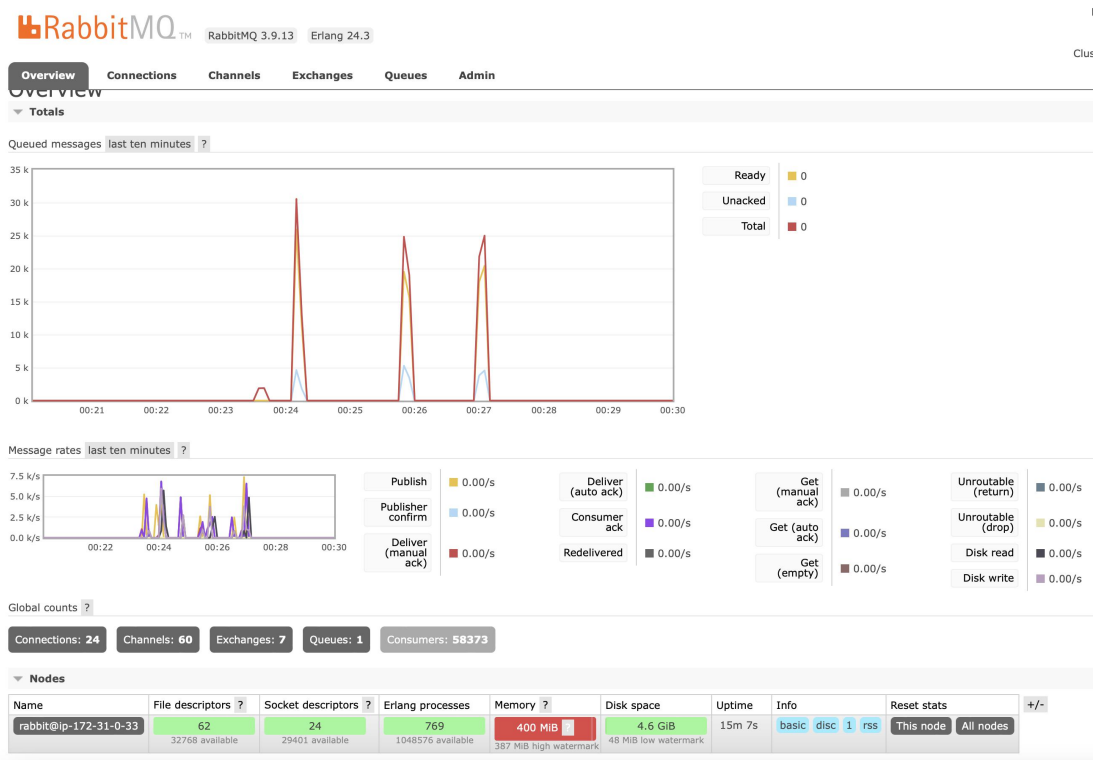
Successful run of 128 threads:



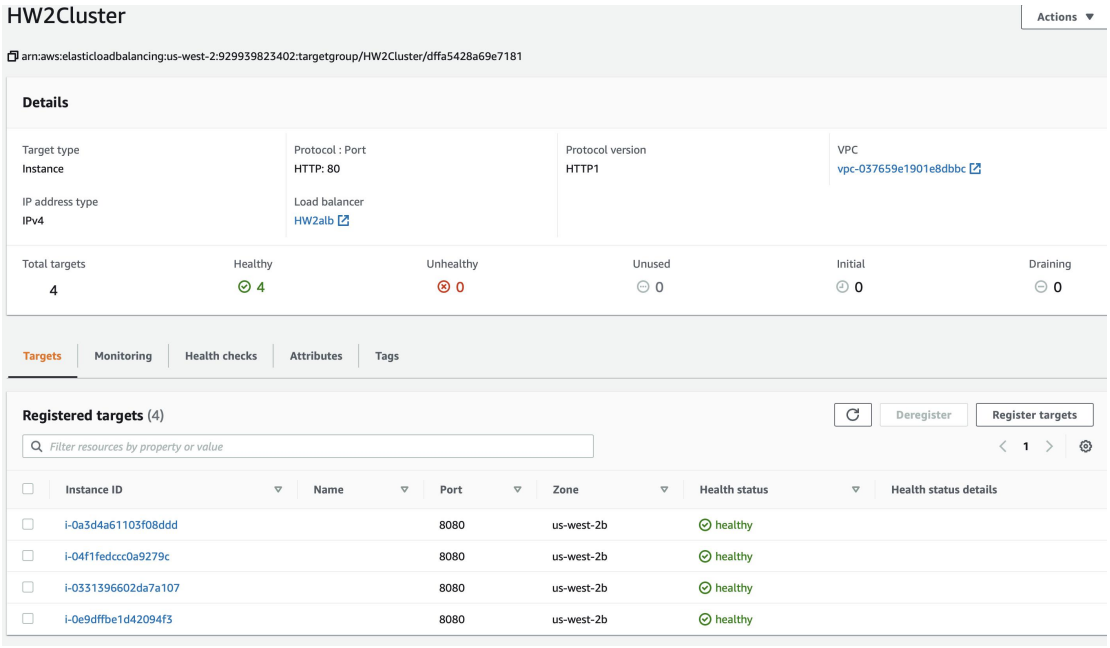
Successful run of 64 threads:



A run of 512 threads



Healthy server



Fast reaction like HW1(256 threads)

```
/Users/alapaka/Library/Java/JavaVirtualMachines/corretto-11.0.14.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=58219
complete phase 1: 14132 milliseconds
complete phase 2: 27608 milliseconds
complete phase 3: 27613 milliseconds
all complete: 65371 milliseconds

data size: 200000
mean: 20 milliseconds
median: 20 milliseconds
p99: 56 milliseconds
min: 11 milliseconds
max: 1047 milliseconds
```

All instances (4 server, 1 RMQ)