git-repo-URL: https://github.com/lyz-sys/neu-cs6650-projects-v2

Description of Database Designs and Deployment Topologies on AWS

Projects 3's code is located at src/main/java/project3. The major changes are in project3/server/consumers/Driver.java and project3/server/db/DynamoDBController.java file. In these file, I add dynamoDB update function after the information is stored at concurrent hash map.

The database has one table. The primary key is skierId, and other attributes are verticals, time, liftID, dayID, seasonID, resortID. This should be sufficient enough to handle later queries in assignment 4.

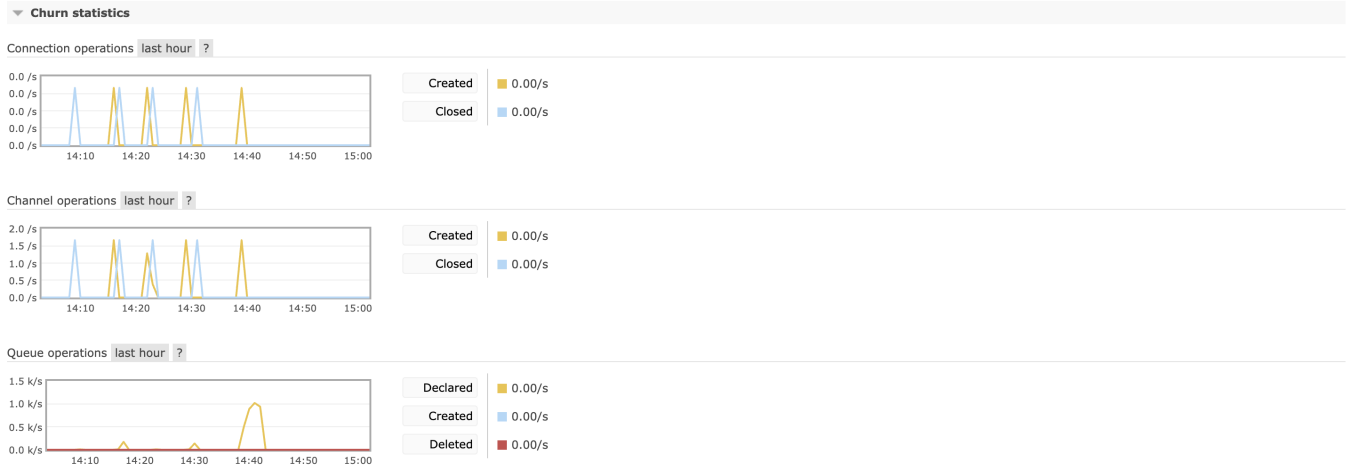
The deployment topology is same as assignment 2, which is like:
client -> aws load balencer -> (2 * EC2 hosting Tomcat server) -> rmq -> consumer -> dynamo db

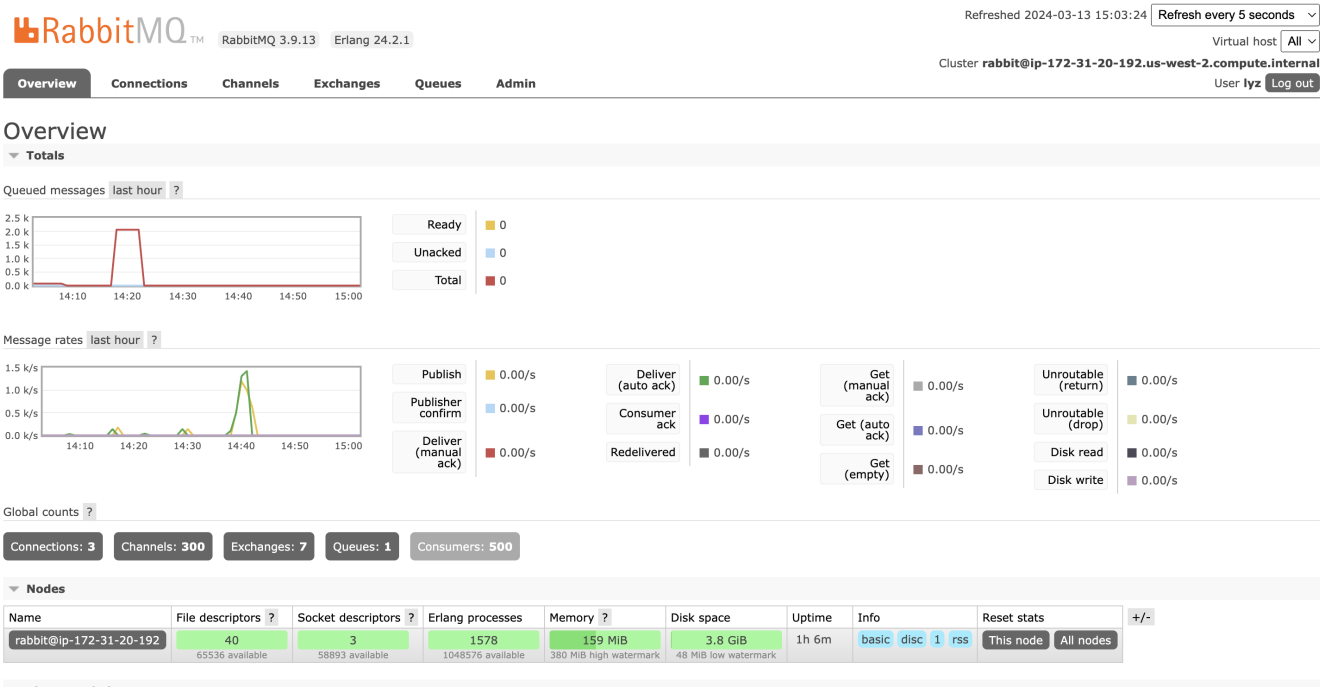
Metric Results

client metrics

```
14:41:18.503 [main] INFO  project3.client.Util - Starting to calculate statistics.
14:41:18.586 [main] INFO  project3.client.Util - Successful requests: 200000
14:41:18.587 [main] INFO  project3.client.Util - Mean response time: 160ms
14:41:18.587 [main] INFO  project3.client.Util - Median response time: 23ms
14:41:18.587 [main] INFO  project3.client.Util - Throughput: 1123 requests/sec
14:41:18.587 [main] INFO  project3.client.Util - p99 response time: 706ms
14:41:18.587 [main] INFO  project3.client.Util - Min response time: 10ms
14:41:18.587 [main] INFO  project3.client.Util - Max response time: 52420ms
14:41:18.587 [main] INFO  project3.client.Util - Writing to CSV

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.5/userguide/command_line_interface.html#sec:command_line_warnings in the Grad
le documentation.

BUILD SUCCESSFUL in 2m 59s
```

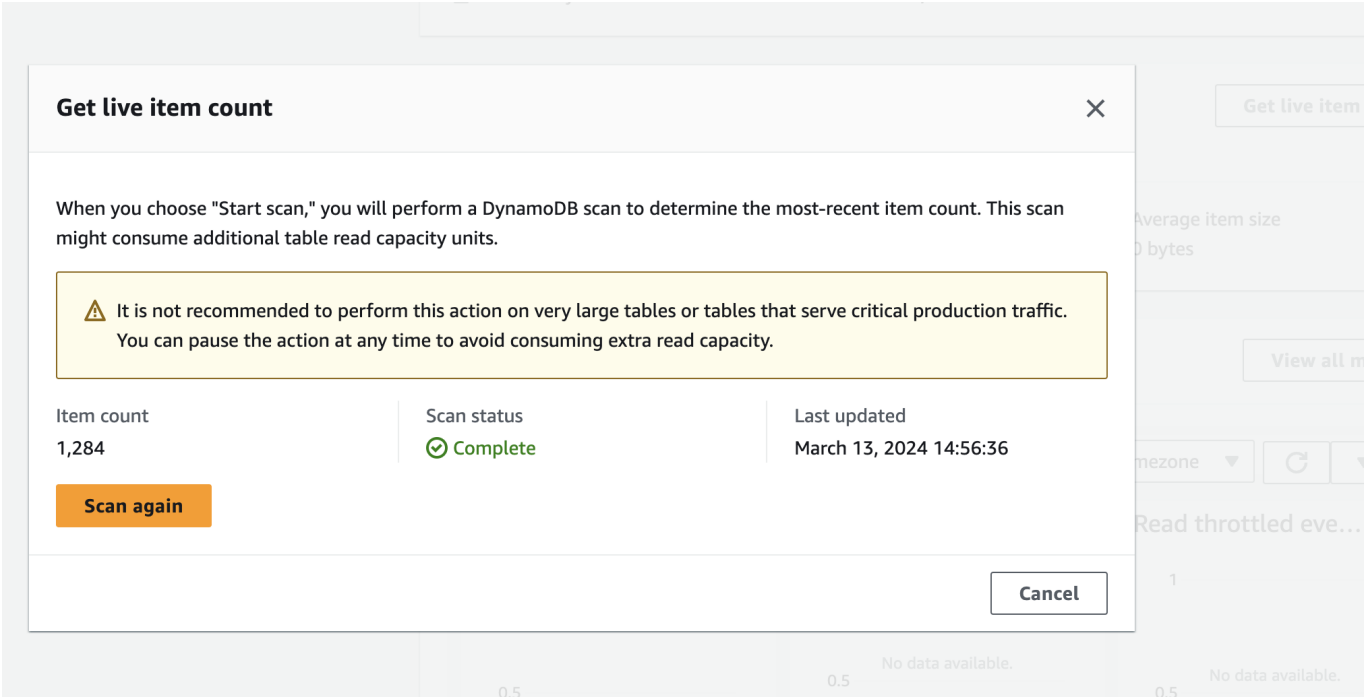
rmq console profile





dynamo db profile

the current item count as the program running:



the example db items:

Items returned (827)

Actions ▼ Create item

1 2 3

skierId *(String)*	dayID	liftID	resortID	seasonID	time	verticals
11981	1	26	8	2024	52	260
29337	1	13	1	2024	93	130
75673	1	15	8	2024	195	150
36043	1	13	9	2024	24	130
73891	1	35	2	2024	237	350
3087	1	17	9	2024	296	170
19905	1	33	1	2024	124	330
49672	1	11	8	2024	347	110
98533	1	24	7	2024	191	240
82908	1	11	4	2024	53	110
83595	1	39	2	2024	267	390
98786	1	5	5	2024	213	50
63877	1	21	3	2024	239	210
65757	1	25	5	2024	110	250