distributed-systems-work

Assignments Repo for CS6650 Building Scalable Distributed Systems 23fall@neu

Ayan Mao 2023 Nov

https://gortonator.github.io/bsds-6650/assignments-2022/Assignment-1

https://gortonator.github.io/bsds-6650/assignments-2022/Assignment-2

Github link: https://github.com/98may/distributed-systems-work/

1: Overview

1.1: milestones

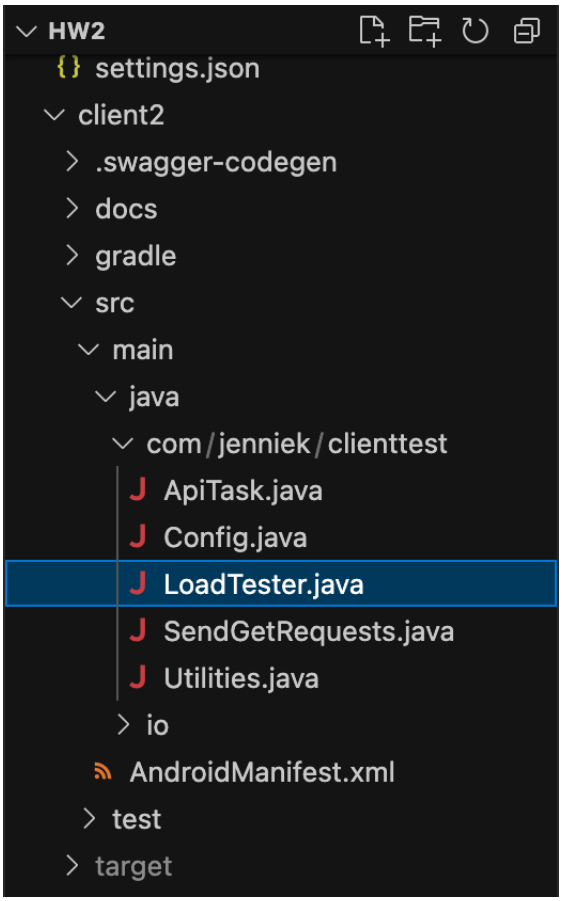
1) add analytics

2) add persist database - AWS RDS + MySQL

3) use multiple servers with load balancers - AWS EC2 + ALB

4) fine tune AWS args

1.2: code design



loadTester.java

loadTester is the main java class to handle tests,
`parse` to set up args from `Usage: LoadTester <threadGroupSize> <numThreadGroups> <delay> [java|go]`,

`startUp` to make sure the server on aws works,

and then `mainLoad` send out the major requests.

```
    public static void main(String[] args) {
        LoadTester loadTester = new LoadTester();
        loadTester.parse(args);
        loadTester.startUp();
        loadTester.mainLoad();
    }
```

ApiTask.java

implements `runnable`

as a single thread unit to send one pair of post and get requests

SendGetRequests.java

implements `runnable`

as a single thread unit to send one get request for main's `startUp`, like half of the `ApiTask.java`

Utilities.java

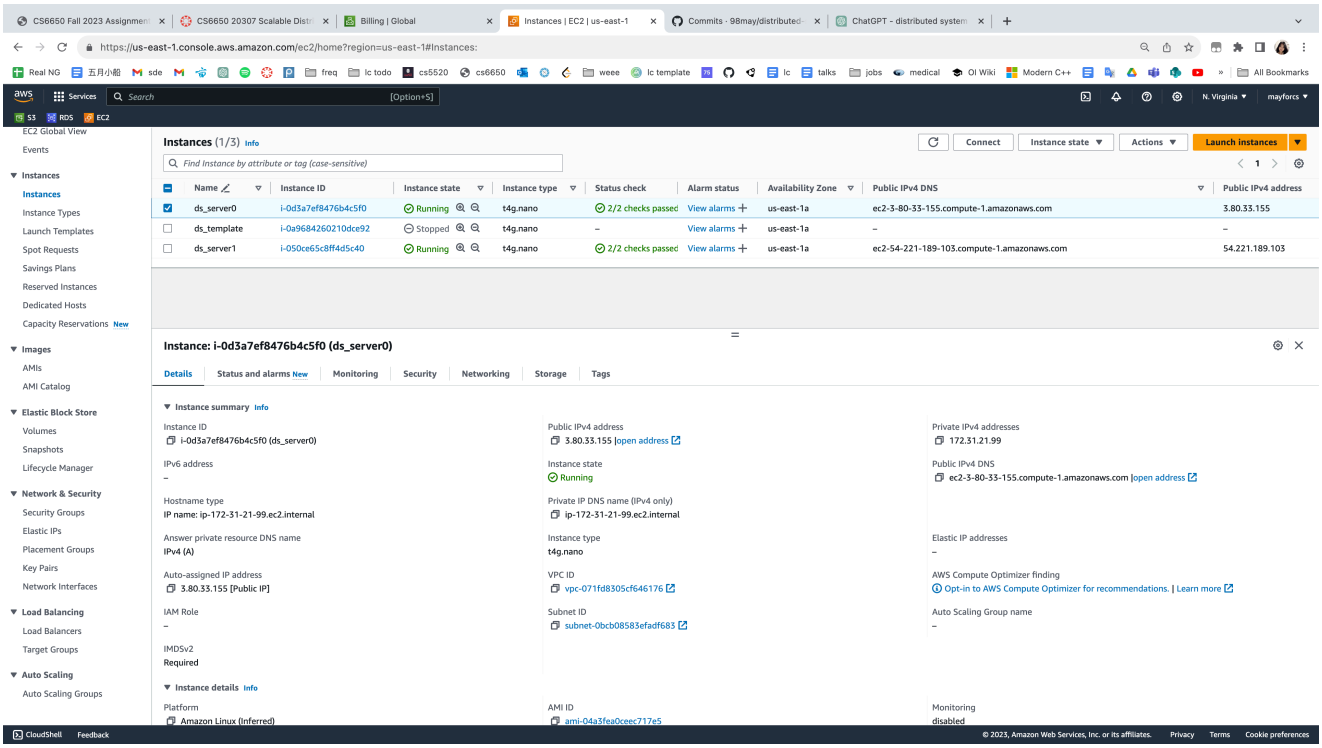
mainly the Utilitiy functions for logging and numbers

Config.java

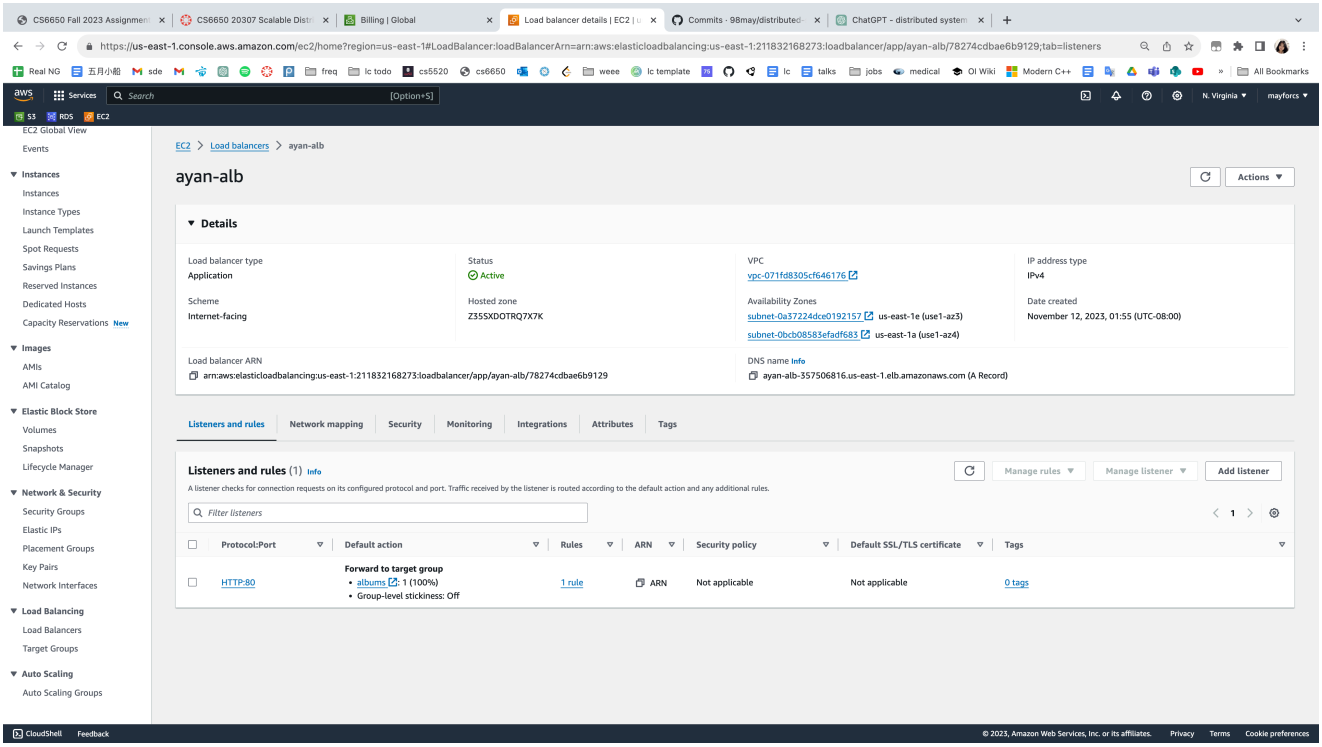
set up the public static final arguments like `INITIAL_THREAD_COUNT`, `CLIENT_LOG_PATH` and `javaServletAddress`

1.3: key info

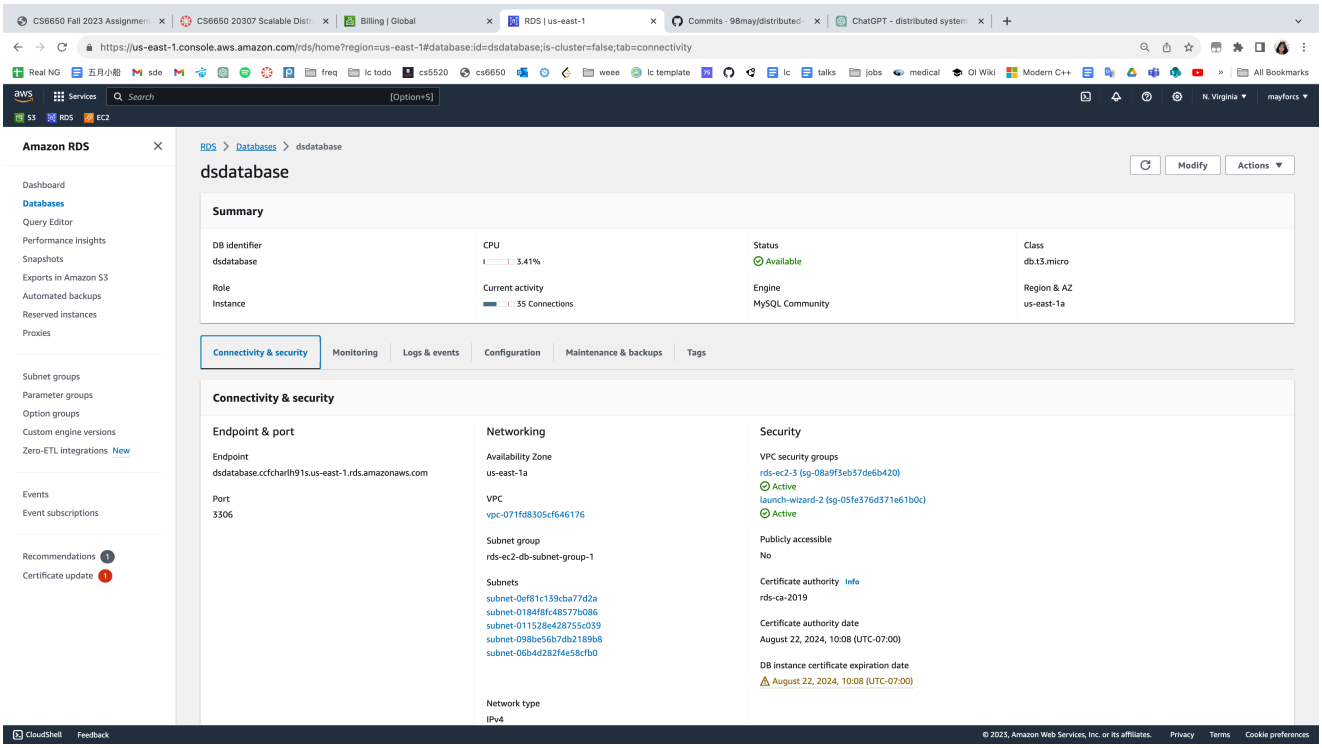
AWS EC2



AWS ALB



AWS RDS



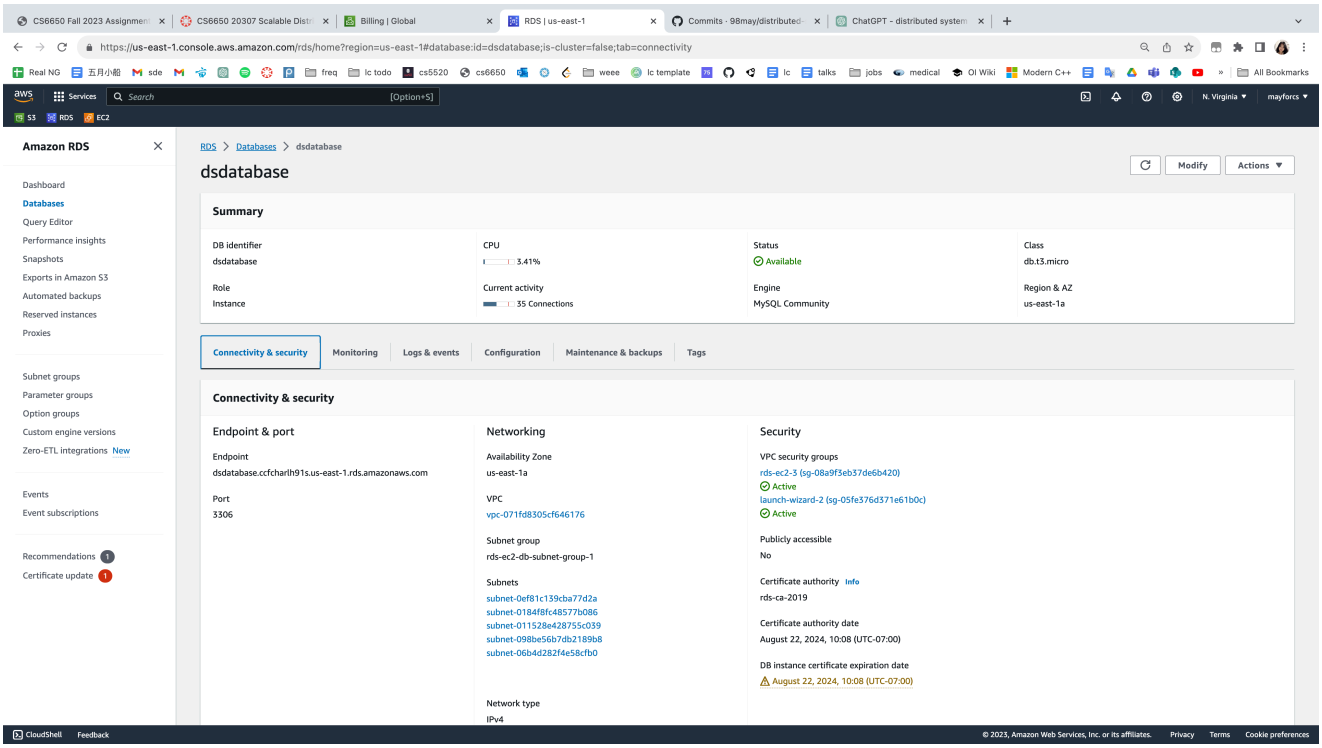
server address

```
public static final String javaServletAddress =
"http://3.80.33.155:8080/AlbumApp";
public static final String javaServletAddress2 =
"http://54.221.189.103:8080/AlbumApp";
public static final String goServerAddress =
"http://3.80.33.155:8081/IGORTON/AlbumStore/1.0.0";
```

2: Database

2.1: choice

AWS RDS + MySQL 8.0.33



2.2: data model

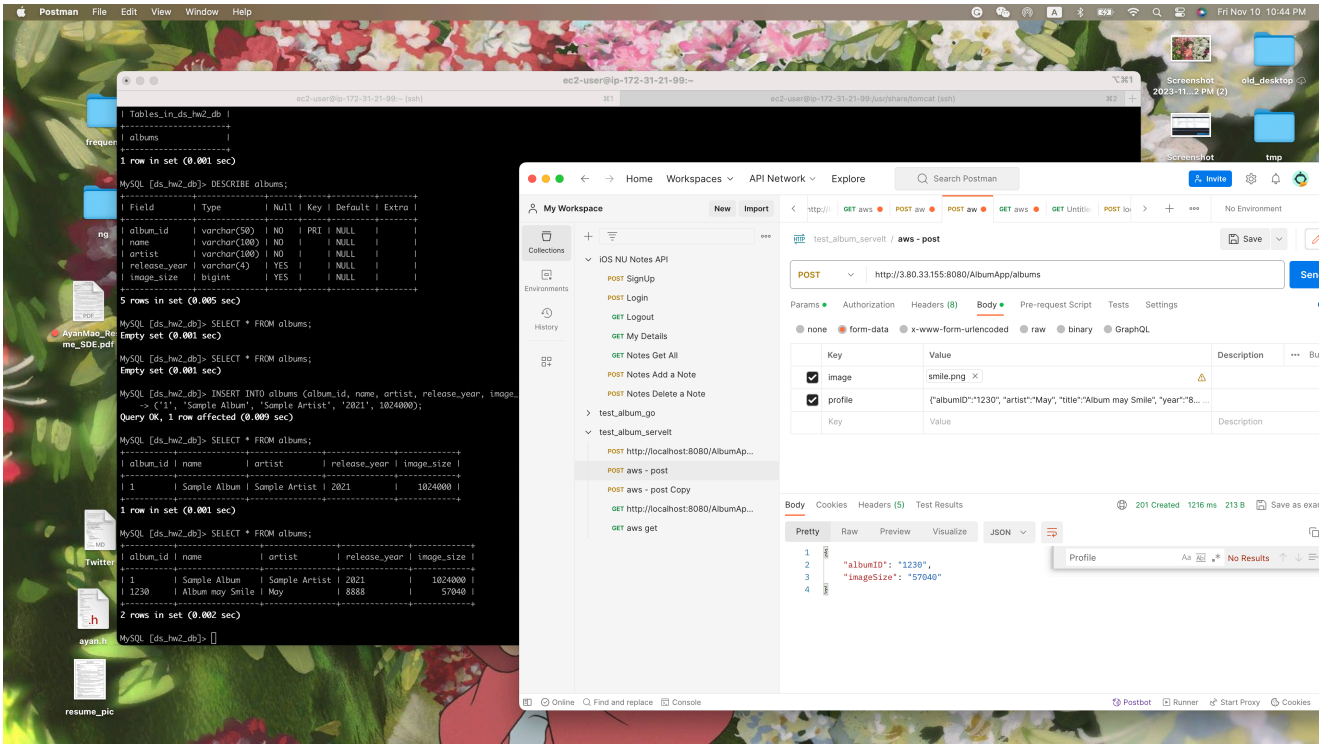
table: albums

field: album_i d(PK), name, artist, release_year, image

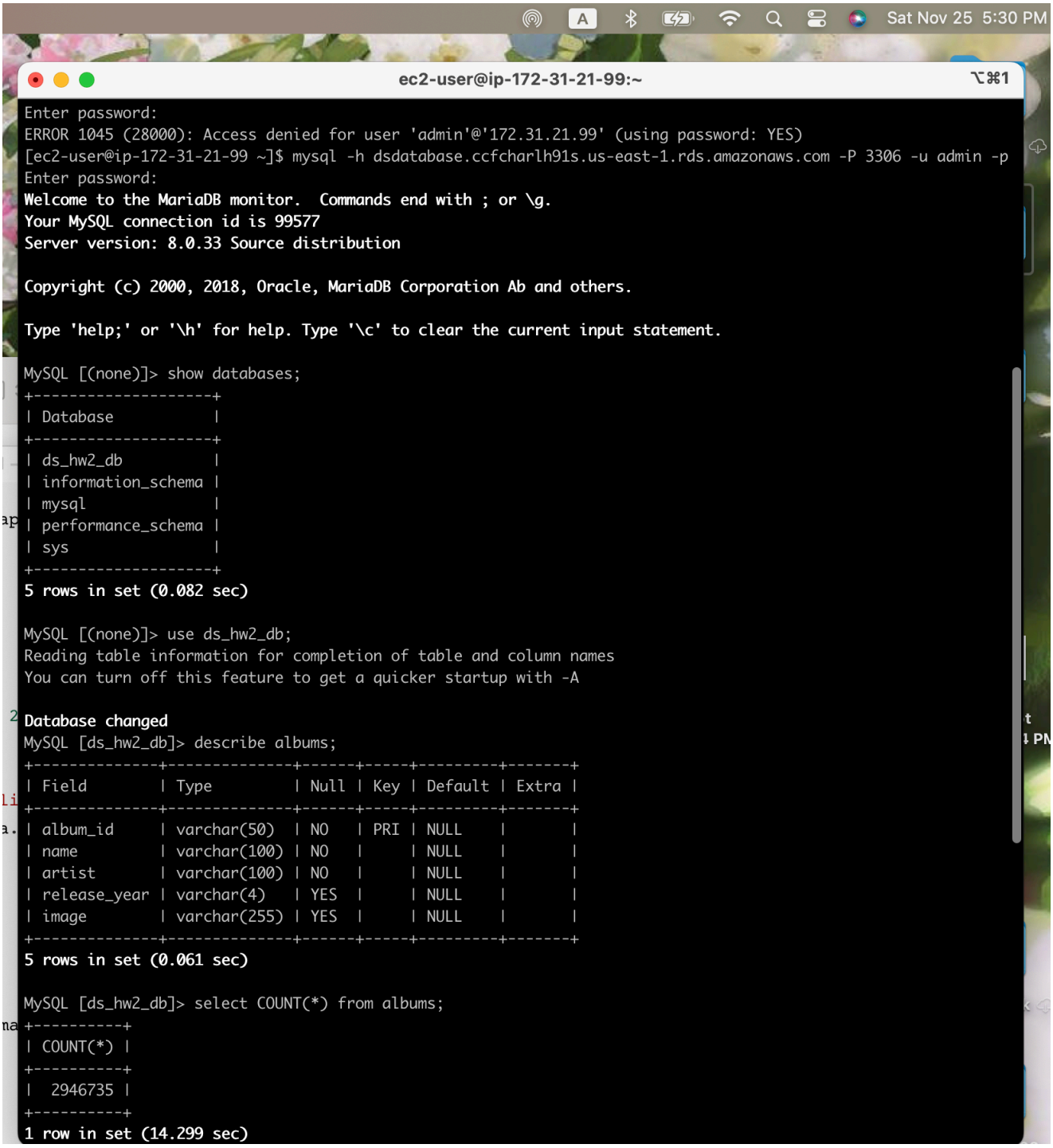
```
Database changed
MySQL [ds_hw2_db]> describe albums;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| album_id     | varchar(50)  | NO   | PRI | NULL    |       |
| name         | varchar(100) | NO   |     | NULL    |       |
| artist       | varchar(100) | NO   |     | NULL    |       |
| release_year | varchar(4)   | YES  |     | NULL    |       |
| image        | varchar(255) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

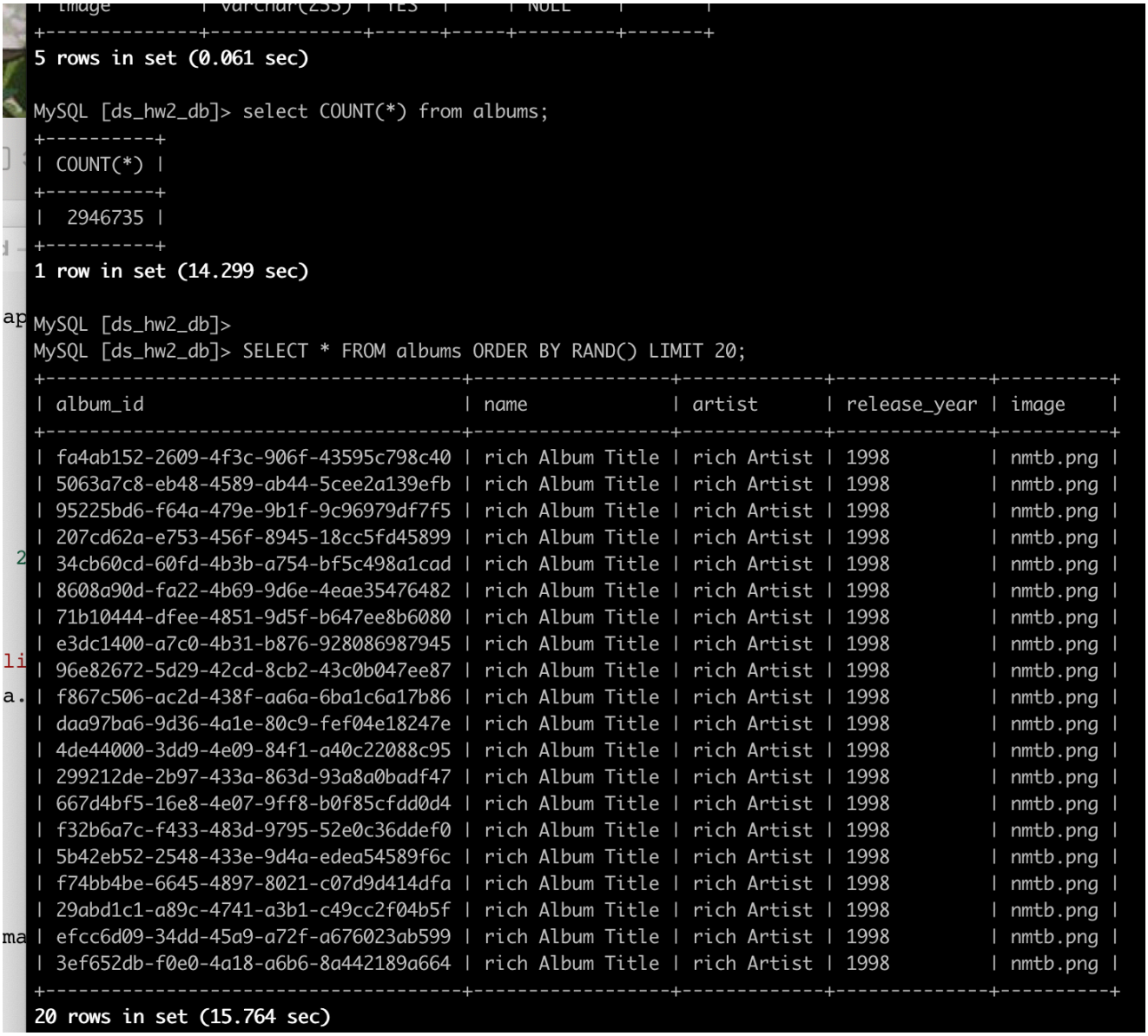
2.3: results

simple postman test succeeded:



As you can see in the graph, it could have **millions** of records in the albums table.


Sat Nov 25 5:30 PM
ec2-user@ip-172-31-21-99:~
⌥⌘1
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'172.31.21.99' (using password: YES)
[ec2-user@ip-172-31-21-99 ~]$ mysql -h dsdatabase.ccfcharlh91s.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 99577
Server version: 8.0.33 Source distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [(none)]> show databases;
+--------------------+
| Database |
+--------------------+
| ds_hw2_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+--------------------+
5 rows in set (0.082 sec)
MySQL [(none)]> use ds_hw2_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MySQL [ds_hw2_db]> describe albums;
+--------------+--------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
album_id	varchar(50)	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
artist	varchar(100)	NO		NULL	
release_year	varchar(4)	YES		NULL	
image	varchar(255)	YES		NULL	
+--------------+--------------+------+-----+---------+-------+
5 rows in set (0.061 sec)
MySQL [ds_hw2_db]> select COUNT(*) from albums;
+----------+
| COUNT(*) |
+----------+
| 2946735 |
+----------+
1 row in set (14.299 sec)

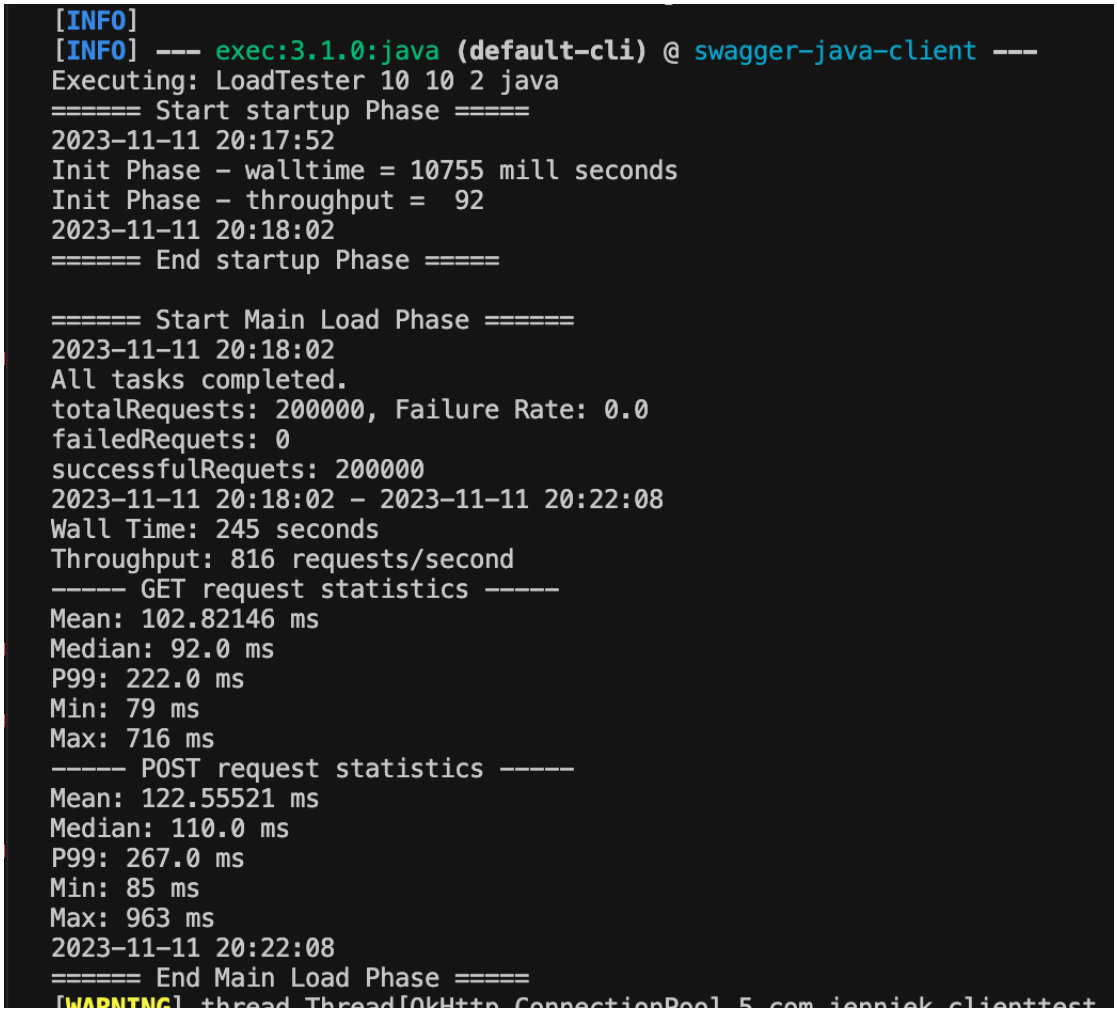


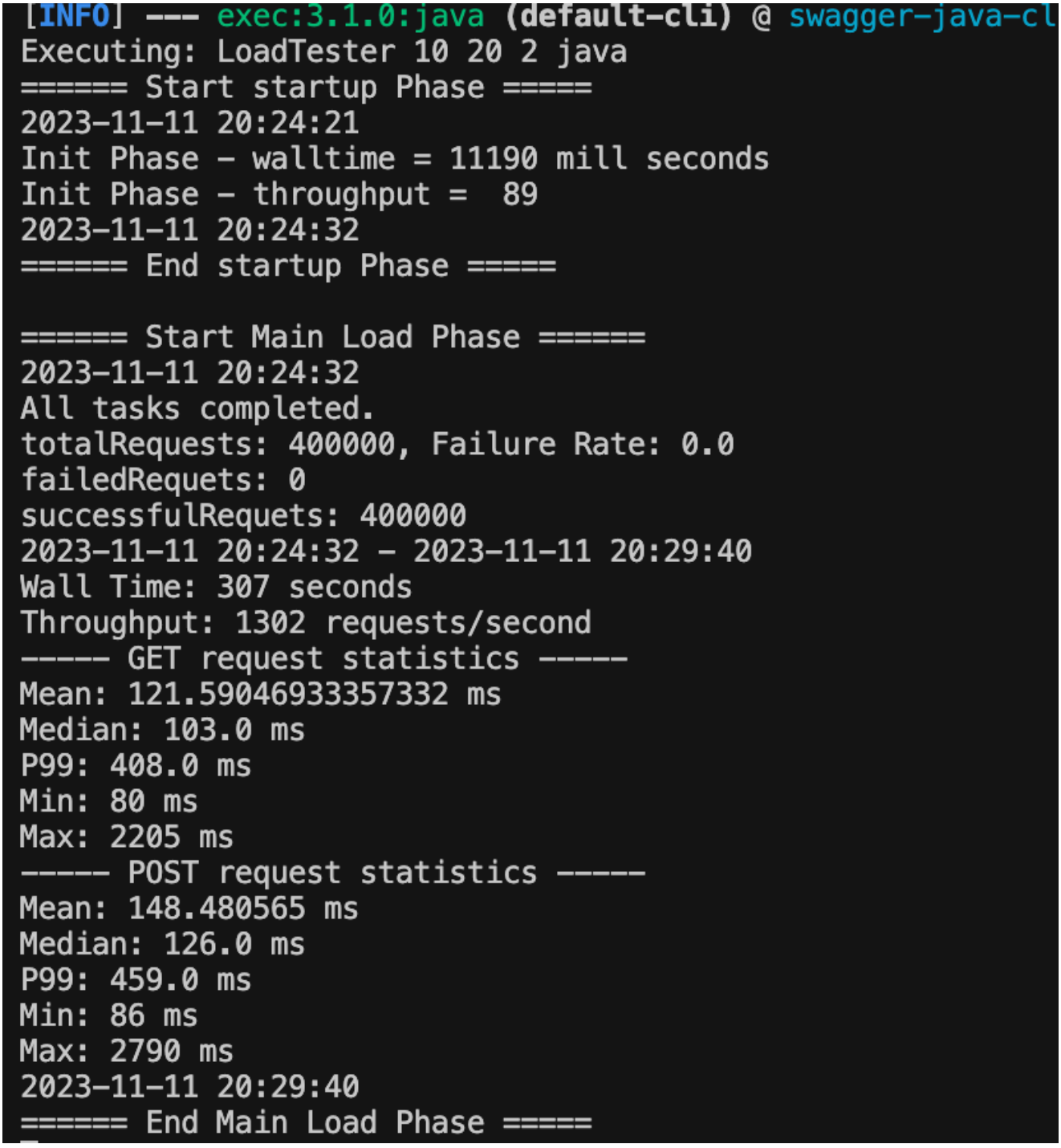
```
| image        | varchar(255) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
5 rows in set (0.061 sec)

MySQL [ds_hw2_db]> select COUNT(*) from albums;
+----------+
| COUNT(*) |
+----------+
|  2946735 |
+----------+
1 row in set (14.299 sec)

MySQL [ds_hw2_db]>
MySQL [ds_hw2_db]> SELECT * FROM albums ORDER BY RAND() LIMIT 20;
+--------------------------------------+------------------+-------------+--------------+----------+
| album_id                             | name             | artist      | release_year | image    |
+--------------------------------------+------------------+-------------+--------------+----------+
| fa4ab152-2609-4f3c-906f-43595c798c40 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 5063a7c8-eb48-4589-ab44-5cee2a139efb | rich Album Title | rich Artist | 1998         | nmtb.png |
| 95225bd6-f64a-479e-9b1f-9c96979df7f5 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 207cd62a-e753-456f-8945-18cc5fd45899 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 34cb60cd-60fd-4b3b-a754-bf5c498a1cad | rich Album Title | rich Artist | 1998         | nmtb.png |
| 8608a90d-fa22-4b69-9d6e-4eae35476482 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 71b10444-dfee-4851-9d5f-b647ee8b6080 | rich Album Title | rich Artist | 1998         | nmtb.png |
| e3dc1400-a7c0-4b31-b876-928086987945 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 96e82672-5d29-42cd-8cb2-43c0b047ee87 | rich Album Title | rich Artist | 1998         | nmtb.png |
| f867c506-ac2d-438f-aa6a-6ba1c6a17b86 | rich Album Title | rich Artist | 1998         | nmtb.png |
| daa97ba6-9d36-4a1e-80c9-fef04e18247e | rich Album Title | rich Artist | 1998         | nmtb.png |
| 4de44000-3dd9-4e09-84f1-a40c22088c95 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 299212de-2b97-433a-863d-93a8a0badf47 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 667d4bf5-16e8-4e07-9ff8-b0f85cfdd0d4 | rich Album Title | rich Artist | 1998         | nmtb.png |
| f32b6a7c-f433-483d-9795-52e0c36ddef0 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 5b42eb52-2548-433e-9d4a-edea54589f6c | rich Album Title | rich Artist | 1998         | nmtb.png |
| f74bb4be-6645-4897-8021-c07d9d414dfa | rich Album Title | rich Artist | 1998         | nmtb.png |
| 29abd1c1-a89c-4741-a3b1-c49cc2f04b5f | rich Album Title | rich Artist | 1998         | nmtb.png |
| efcc6d09-34dd-45a9-a72f-a676023ab599 | rich Album Title | rich Artist | 1998         | nmtb.png |
| 3ef652db-f0e0-4a18-a6b6-8a442189a664 | rich Album Title | rich Artist | 1998         | nmtb.png |
+--------------------------------------+------------------+-------------+--------------+----------+
20 rows in set (15.764 sec)
```

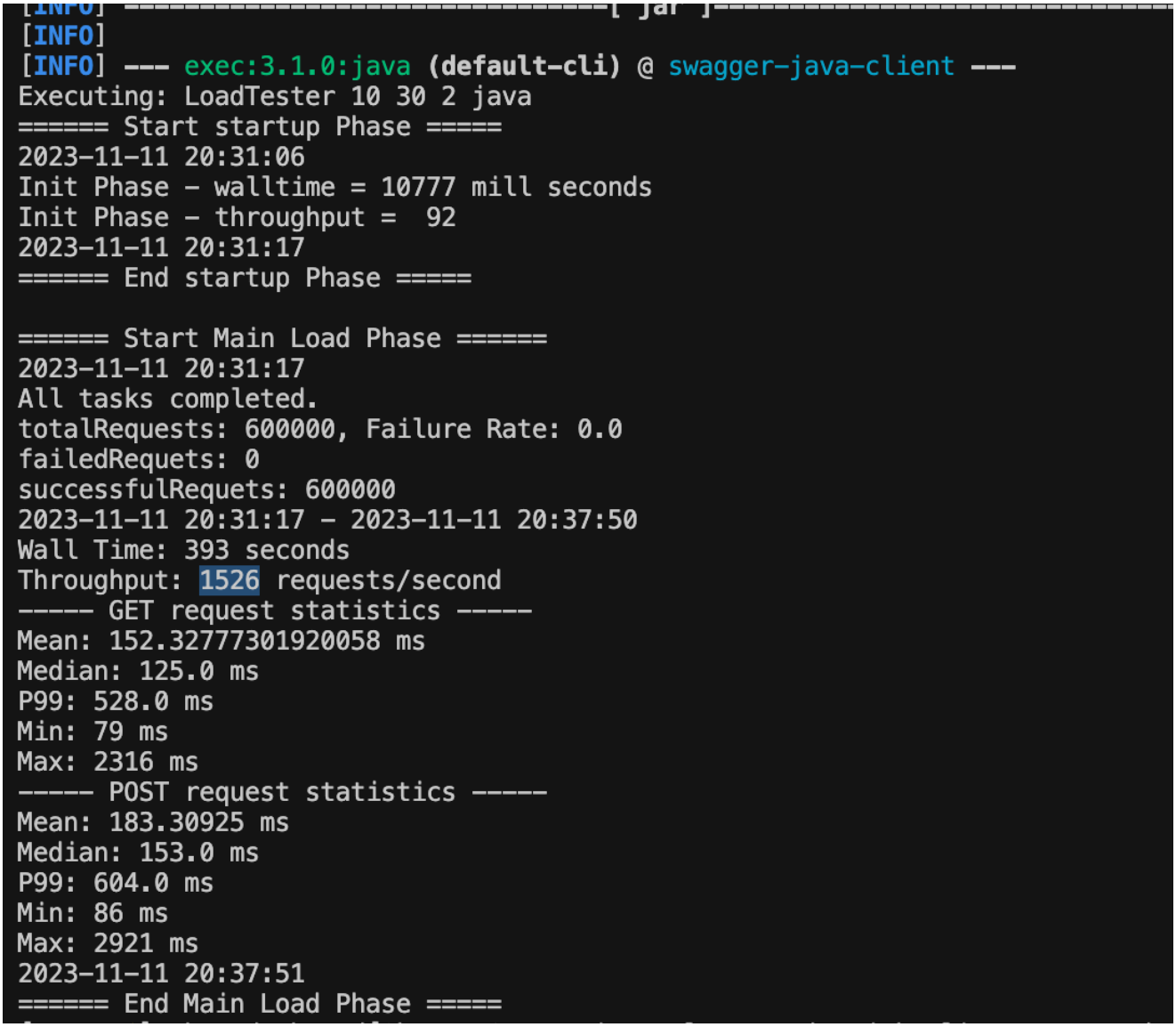
3: single server results

Complete test results under /test_results folder
Here are some highlights:

Java Servlet

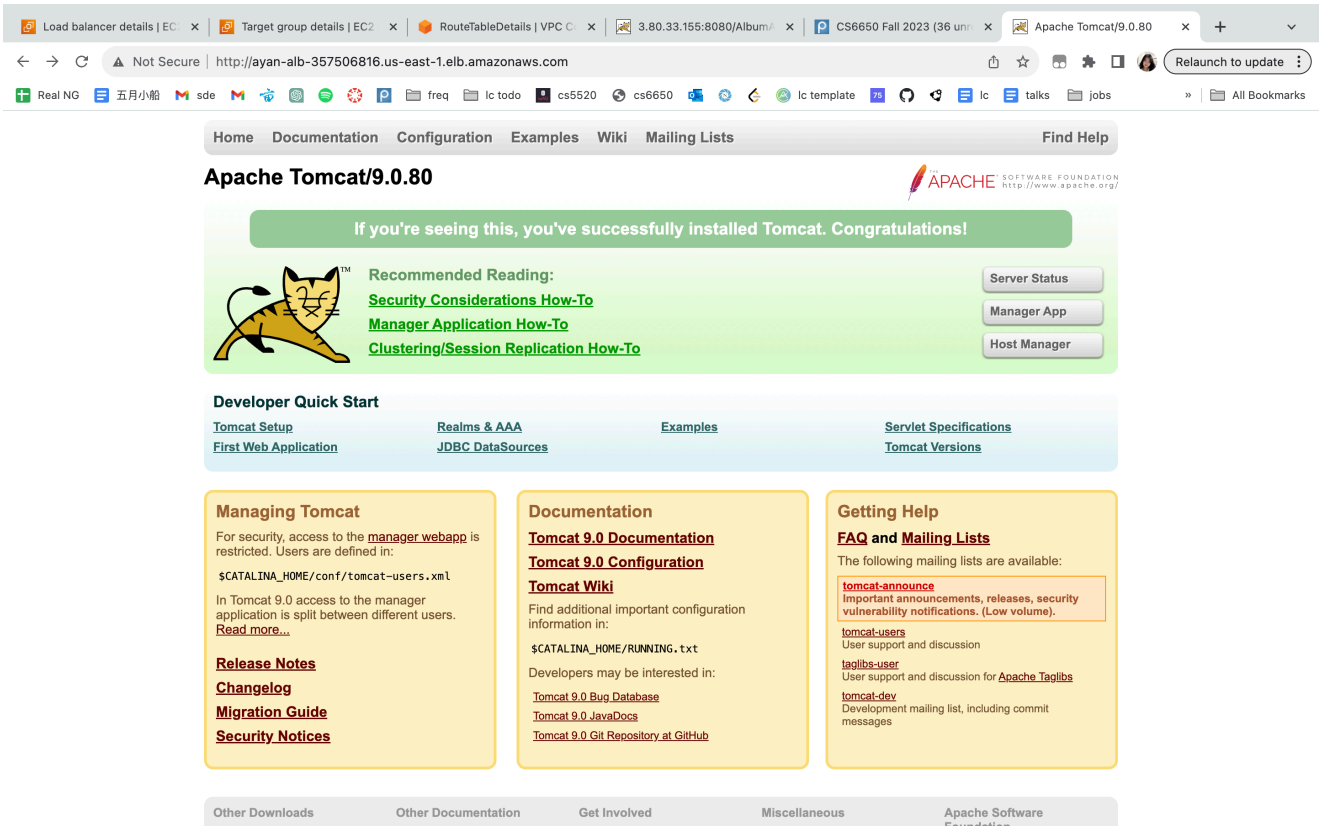

[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ swagger-java-client ---
Executing: LoadTester 10 10 2 java
====== Start startup Phase =====
2023-11-11 20:17:52
Init Phase - walltime = 10755 mill seconds
Init Phase - throughput = 92
2023-11-11 20:18:02
====== End startup Phase =====
====== Start Main Load Phase ======
2023-11-11 20:18:02
All tasks completed.
totalRequests: 200000, Failure Rate: 0.0
failedRequets: 0
successfulRequets: 200000
2023-11-11 20:18:02 - 2023-11-11 20:22:08
Wall Time: 245 seconds
Throughput: 816 requests/second
----- GET request statistics -----
Mean: 102.82146 ms
Median: 92.0 ms
P99: 222.0 ms
Min: 79 ms
Max: 716 ms
----- POST request statistics -----
Mean: 122.55521 ms
Median: 110.0 ms
P99: 267.0 ms
Min: 85 ms
Max: 963 ms
2023-11-11 20:22:08
====== End Main Load Phase =====


[INFO] --- exec:3.1.0:java (default-cli) @ swagger-java-cl
Executing: LoadTester 10 20 2 java
====== Start startup Phase =====
2023-11-11 20:24:21
Init Phase - walltime = 11190 mill seconds
Init Phase - throughput = 89
2023-11-11 20:24:32
====== End startup Phase =====
====== Start Main Load Phase ======
2023-11-11 20:24:32
All tasks completed.
totalRequests: 400000, Failure Rate: 0.0
failedRequets: 0
successfulRequets: 400000
2023-11-11 20:24:32 - 2023-11-11 20:29:40
Wall Time: 307 seconds
Throughput: 1302 requests/second
----- GET request statistics -----
Mean: 121.59046933357332 ms
Median: 103.0 ms
P99: 408.0 ms
Min: 80 ms
Max: 2205 ms
----- POST request statistics -----
Mean: 148.480565 ms
Median: 126.0 ms
P99: 459.0 ms
Min: 86 ms
Max: 2790 ms
2023-11-11 20:29:40
====== End Main Load Phase =====

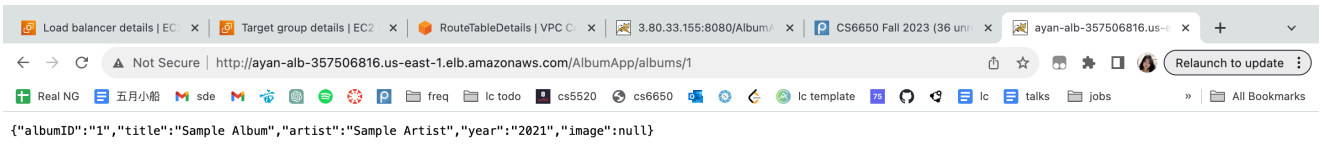


4: load balanced servers results

simple tomcat test passed



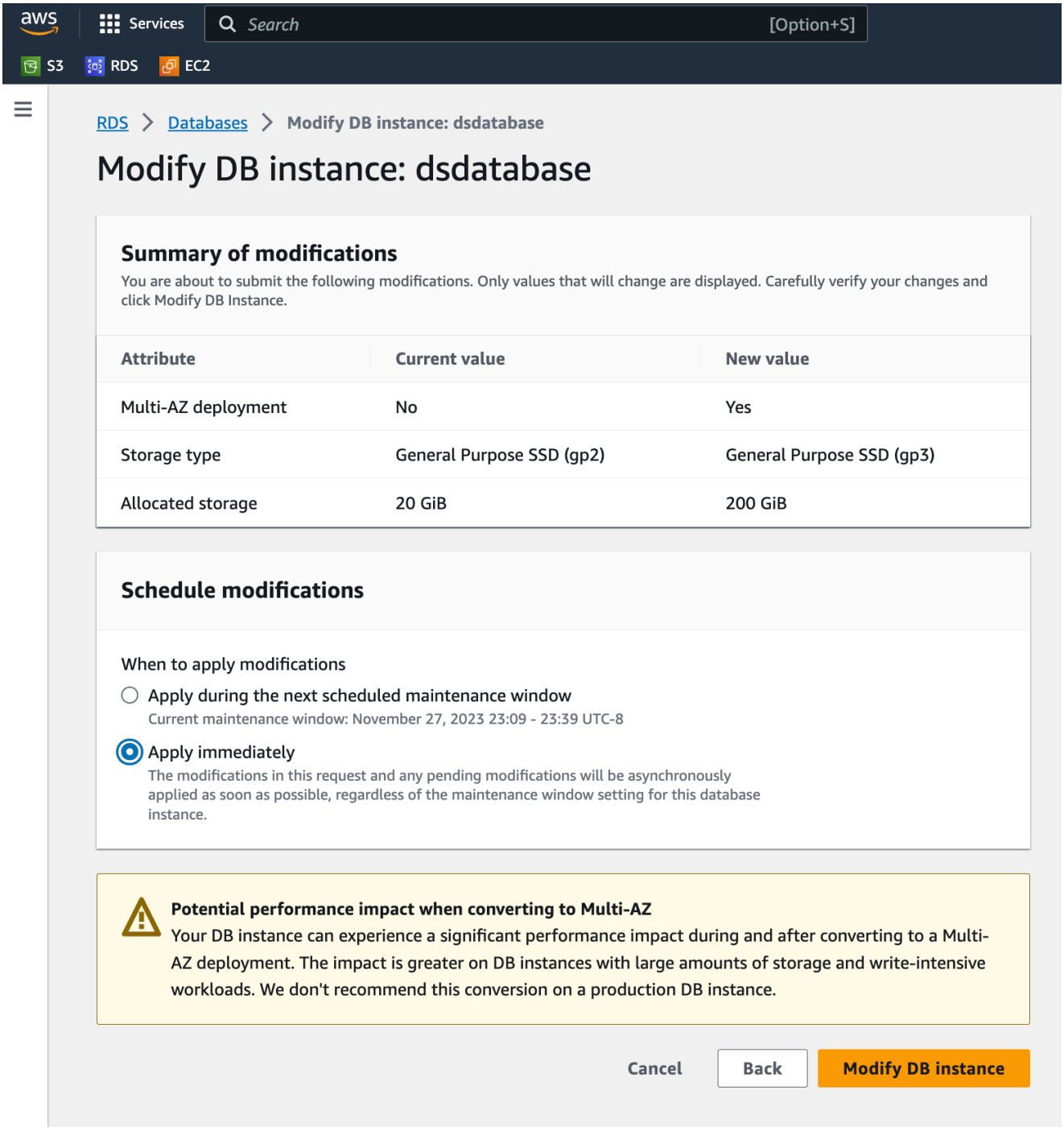
simple alb get test passed



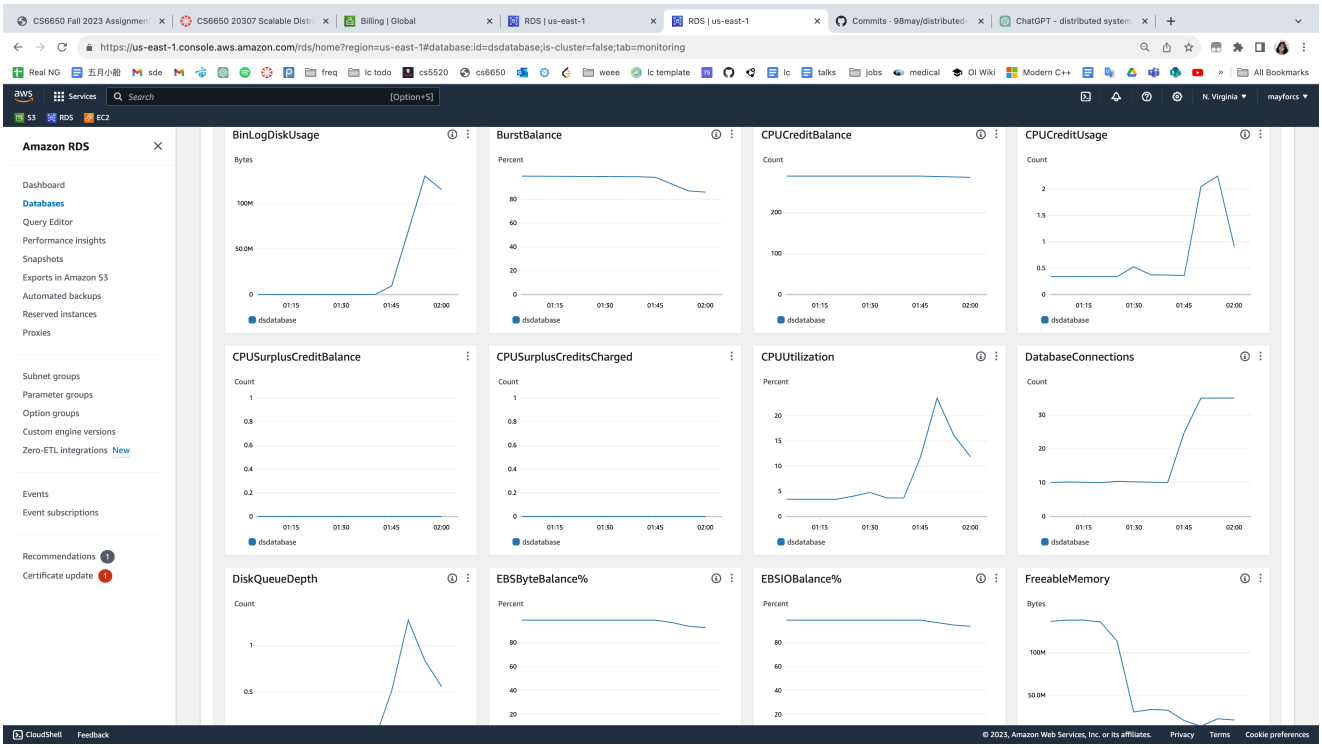
5: tuned results

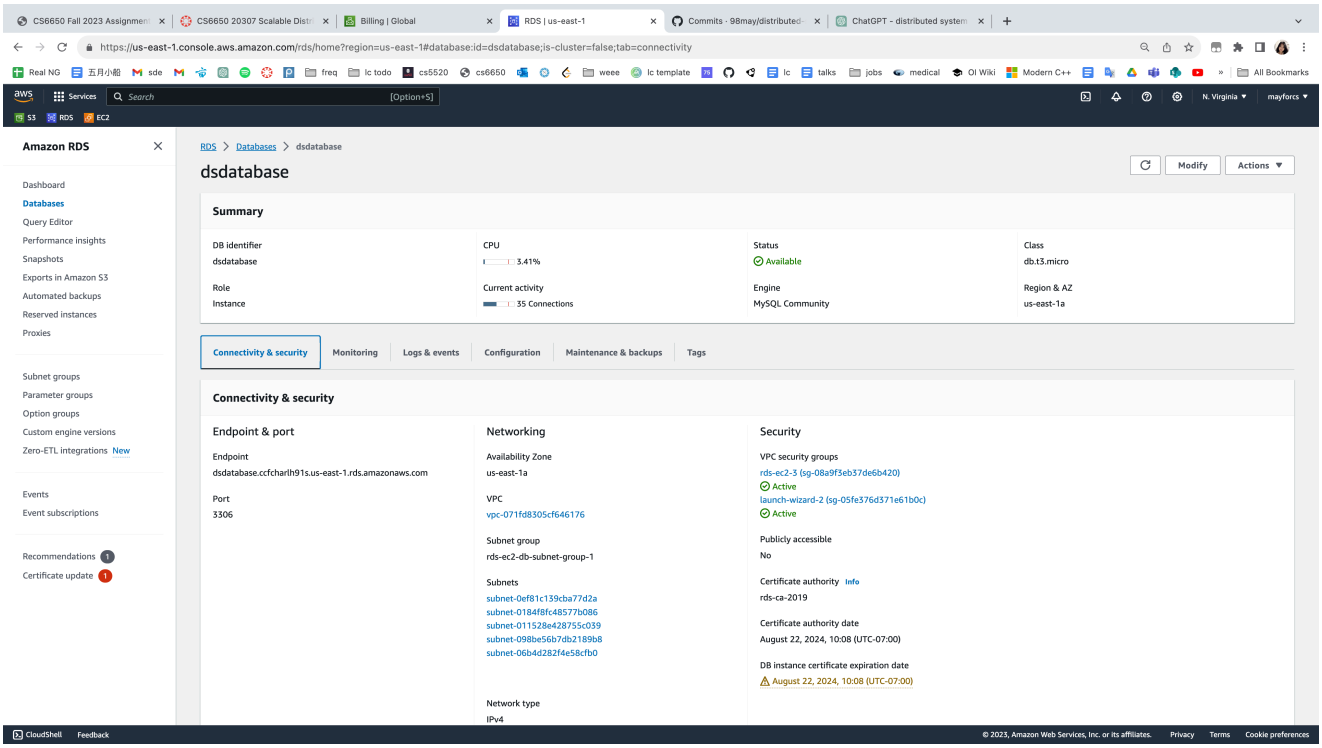
After tune rds, throughput increased from 1117 to 1595, which is 42.8% improvements, great!

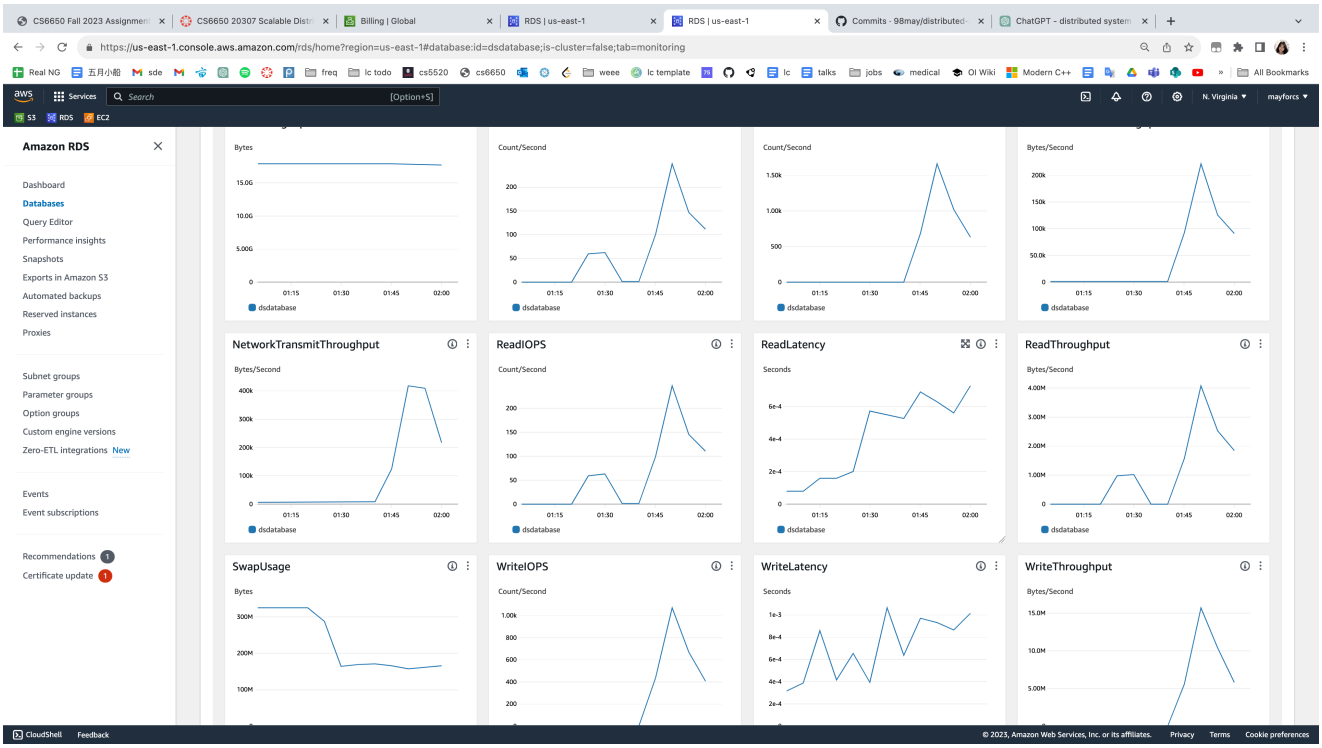
some tune details



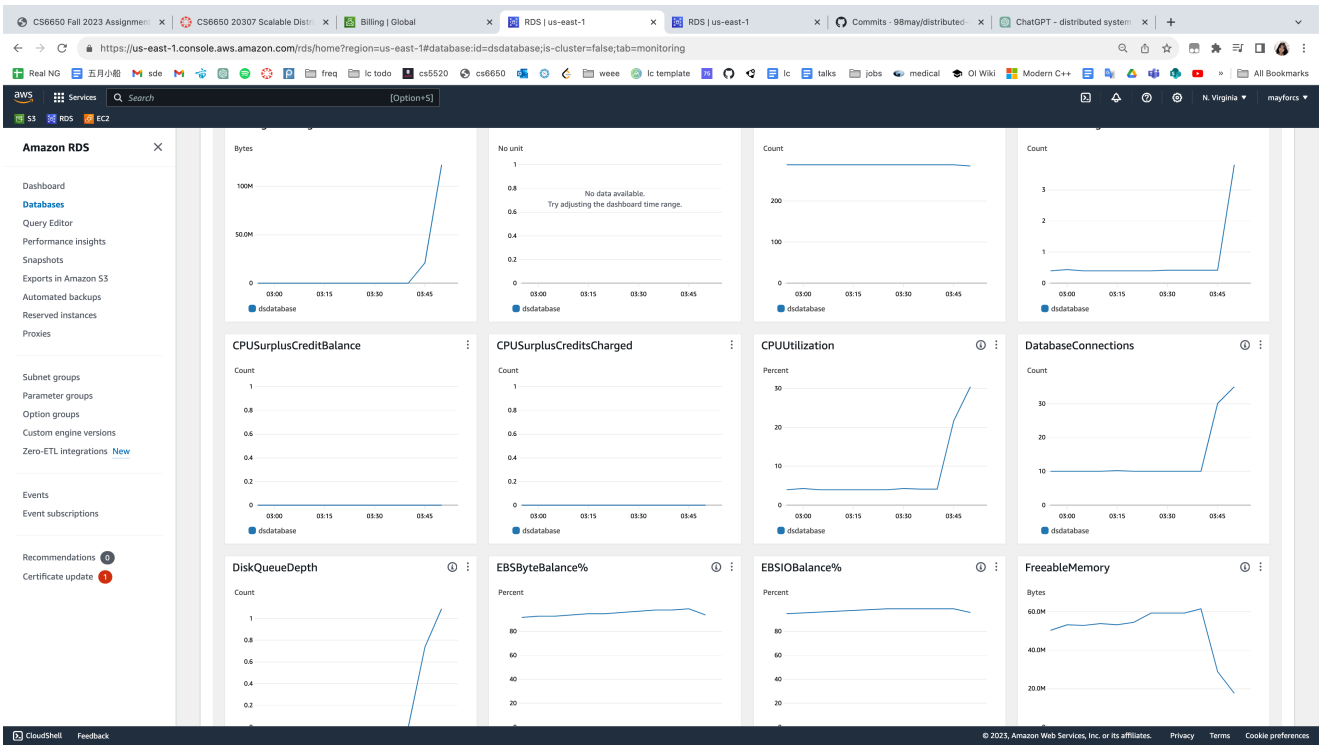
before tune

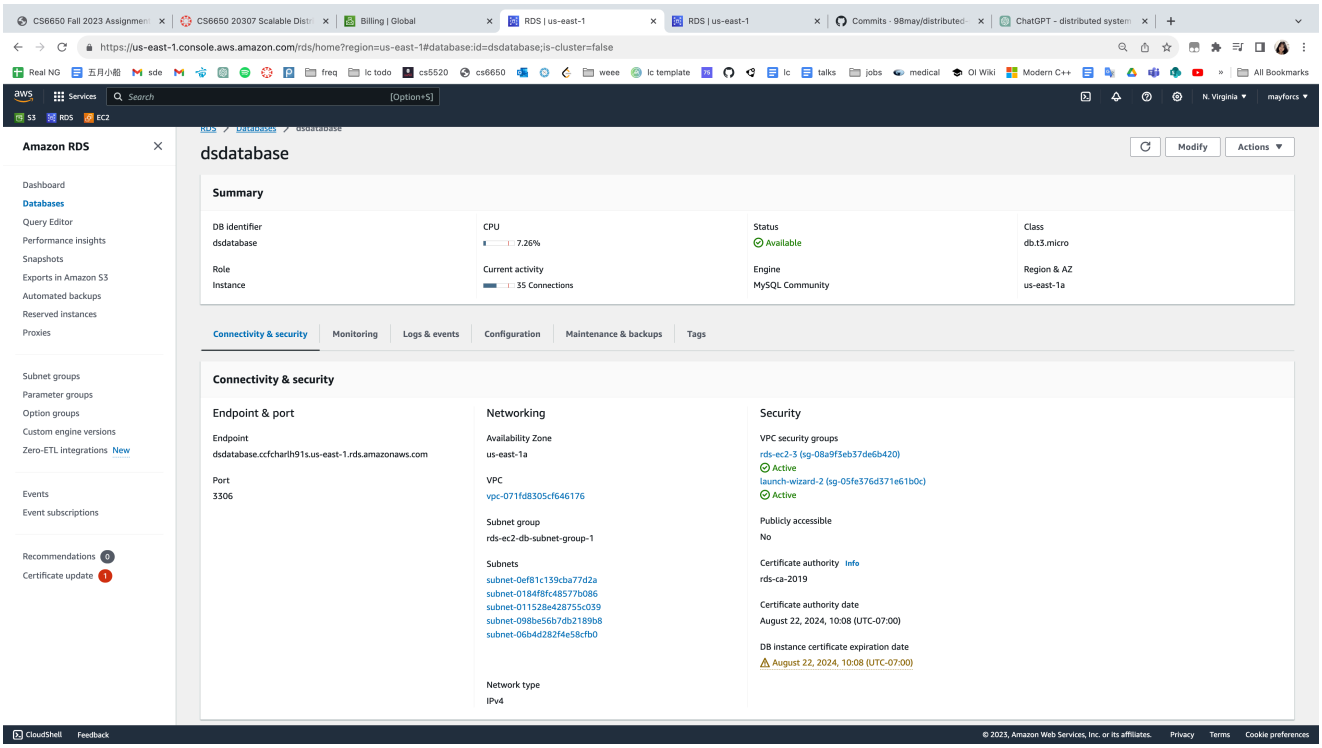

Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 1
Certificate update 1
BinLogDiskUsage
BurstBalance
CPUCreditBalance
CPUCreditUsage
CPUSurplusCreditBalance
CPUSurplusCreditsCharged
CPUUtilization
DatabaseConnections
DiskQueueDepth
EBSByteBalance%
EBSIOBalance%
FreeableMemory
dsdatabase

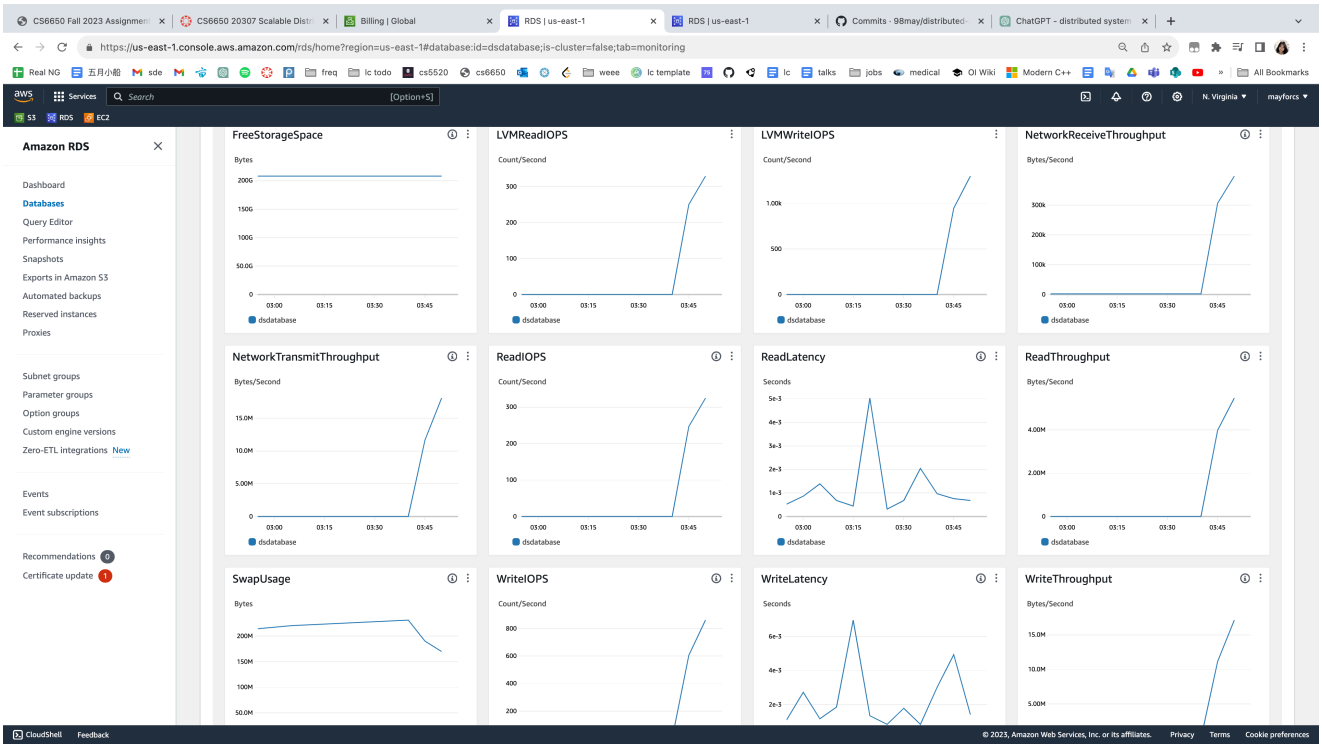

RDS > Databases > dsdatabase
dsdatabase
Modify
Actions
Summary
DB identifier
dsdatabase
CPU
3.41%
Status
Available
Class
db.t3.micro
Role
Instance
Current activity
35 Connections
Engine
MySQL Community
Region & AZ
us-east-1a
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Endpoint & port
Endpoint
dsdatabase.ccfcharlh91s.us-east-1.rds.amazonaws.com
Port
3306
Networking
Availability Zone
us-east-1a
VPC
vpc-071fd8305cf646176
Subnet group
rds-ec2-db-subnet-group-1
Subnets
subnet-0ef81c139cba77d2a
subnet-0184f8fc48577b086
subnet-011528e428755c039
subnet-098be56b7db2189b8
subnet-06b4d282f4e58cfb0
Network type
IPv4
Security
VPC security groups
rds-ec2-3 (sg-08a9f3eb37de6b420)
Active
launch-wizard-2 (sg-05fe376d371e61b0c)
Active
Publicly accessible
No
Certificate authority Info
rds-ca-2019
Certificate authority date
August 22, 2024, 10:08 (UTC-07:00)
DB instance certificate expiration date
August 22, 2024, 10:08 (UTC-07:00)


Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 1
Certificate update 1
NetworkTransmitThroughput
ReadIOPS
ReadLatency
ReadThroughput
SwapUsage
WriteIOPS
WriteLatency
WriteThroughput
dsdatabase

after tune


Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update 1
No data available.
Try adjusting the dashboard time range.
CPUSurplusCreditBalance
CPUSurplusCreditsCharged
CPUUtilization
DatabaseConnections
DiskQueueDepth
EBSByteBalance%
EBSIOBalance%
FreeableMemory
dsdatabase





6: Appendix of Command

Please login as the user "ec2-user" rather than the user "root".

local

```
// ssh to aws ec2
/Users/may/Desktop/neu/cs6650_distributed/shortcuts/ssh_ec2.sh

  // chmod 400 ec2_2.pem
  // ssh -i ec2_2.pem ec2-user@ec2-3-80-33-155.compute-1.amazonaws.com
  ssh -i ec2_2.pem ec2-user@ec2-54-221-189-103.compute-1.amazonaws.com


// mysql to aws rdb
mysql -h dsdatabase.ccfcharlh91s.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
  // pwd: 01234567

  /* java servlet */
// local build java servlet in Maven project to generate the AlbumApp.war file
mvn clean install
// scp java servlet - AlbumApp.war to aws ec2
scp -i /Users/may/Downloads/ec2_2.pem
/Users/may/Desktop/neu/cs6650_distributed/distributed-systems-
work/hw2/java_servlet/AlbumApp/target/AlbumApp.war ec2-user@ec2-3-80-33-
155.compute-1.amazonaws.com:~/tmp


  /* Go server */
// local run go server(macOS + m1 chip) for local tests
GOOS=darwin GOARCH=arm64 go run ./main.go
// local build go server for aws ec2(linux + arm64)
GOOS=linux GOARCH=arm64 go build -o server main.go
// scp go server - server to ec2 ~/emp to start
scp -i /Users/may/Downloads/ec2_2.pem
/Users/may/Desktop/neu/cs6650_distributed/distributed-systems-
work/hw1/go_server/go-server-server-generated/server ec2-user@ec2-3-80-33-
155.compute-1.amazonaws.com:~/tmp

  /* client - LoadTester */
`Usage: LoadTester <threadGroupSize> <numThreadGroups> <delay> [java|go]`
// e.g.
mvn clean install && mvn exec:java -
Dexec.mainClass="com.jenniek.clienttest.LoadTester" -Dexec.args="10 30 2 java"

mvn exec:java -Dexec.mainClass="com.jenniek.clienttest.LoadTester" -Dexec.args="10
30 2 java"


  // plot throughput
  conda activate base
  /Users/may/anaconda3/bin/python
```

```
  /Users/may/Desktop/neu/cs6650_distributed/distributed-systems-
work/hw2/test_results/plot.py
```

aws ec2

```
// java servlet
sudo mv ~/tmp/AlbumApp.war /usr/share/tomcat/webapps && sudo systemctl restart
tomcat && systemctl status tomcat

sudo mv ~/tmp/AlbumApp.war /usr/share/tomcat/webapps

sudo systemctl restart tomcat
sudo systemctl stop tomcat
systemctl status tomcat

sudo vim /usr/share/tomcat/logs/catalina.out
sudo tail /usr/share/tomcat/logs/catalina.out -n 200

// see log to debug java servlet
sudo bash -c 'echo > /usr/share/tomcat/logs/catalina.out'
sudo grep "album" /usr/share/tomcat/logs/catalina.out

// go server
~/tmp/server


// mysql to aws rdb
mysql -h dsdatabase.ccfcharlh91s.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
  // 01234567
show databases;
use ds_hw2_db;
describe albums;
select COUNT(*) from albums;
SELECT * FROM albums ORDER BY RAND() LIMIT 20;
```