

CMPE 281 - MIDTERM EXAM (PRACTICE)

⚠ This is a preview of the published version of the quiz

Started: Oct 10 at 11:46am

Quiz Instructions

CMPE 281 - MIDTERM PRACTICE

Hands-on Lab Problem Set

***** THIS IS A PRACTICE EXAM AND WILL NOT BE GRADED *****

- You have **120 minutes** on this part of the exam
- This part of the exam is worth **125 points**
- This is an open book, open notes exam and use of internet is allowed
- Do not communicate with any other person in real-time via chat or messaging
- All work must be your own, with the exception of code copied from documentation, forums or blogs.
- All Source Code Used from other sources other than your own must be referenced.
- Any violations of this will result in an immediate ZERO score on this exam

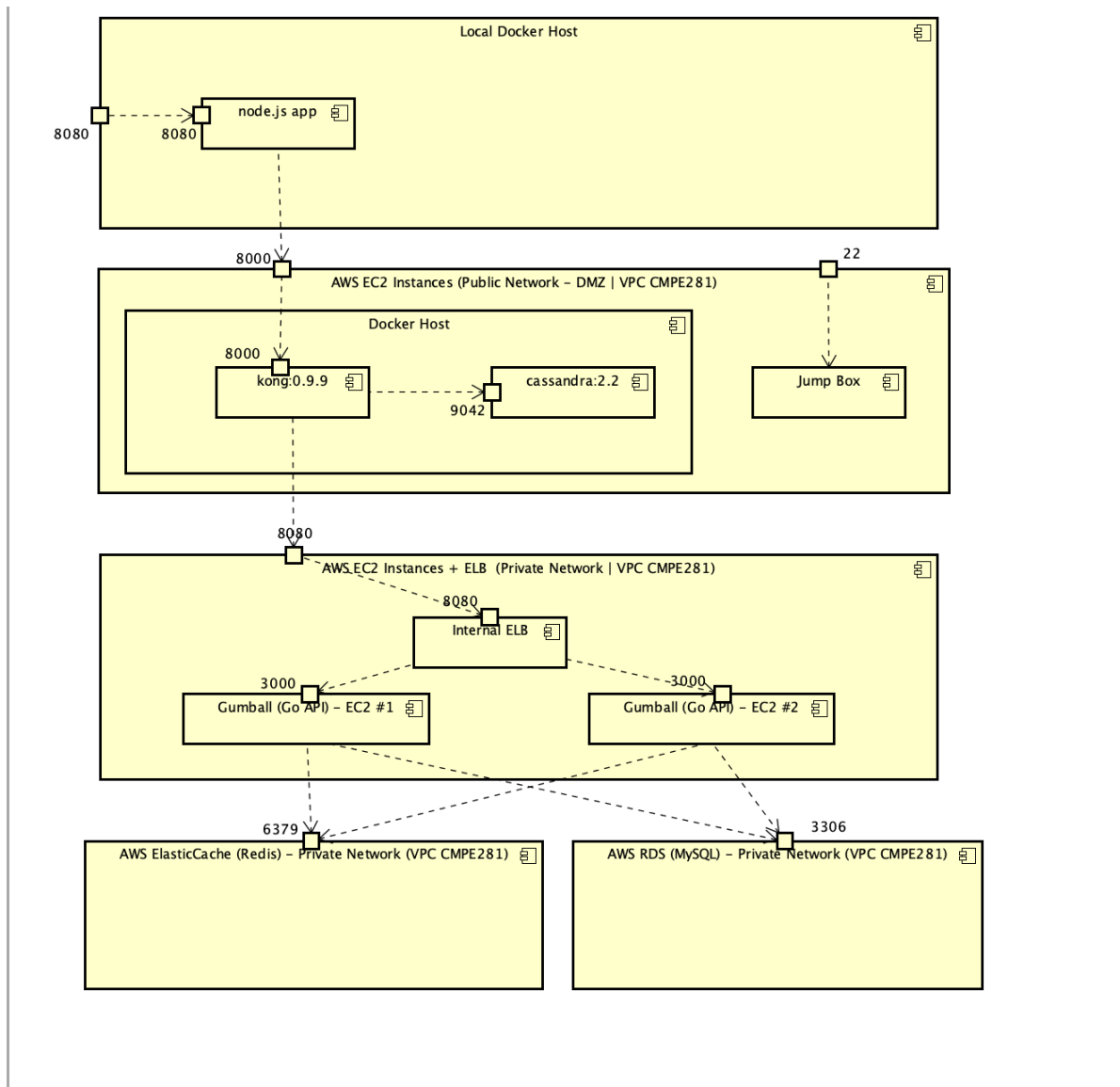
**** Please note that you will be asked to upload screenshots and provide evidence of your work. Please make sure to take "full screenshots" of your desktop and browsers, urls, etc... showing your account id, name and other identifying data from the screen. Not doing so will risk losing points ****

On this Exam, You will be building a full stack Node.js Gumball App deployed locally, with Docker and on AWS. The following diagram shows the deployment topology. Please follow the diagram and configure Network Ports, AWS Subnets and Security Groups appropriately. Keep notes all each of the commands you used for each problem along with screenshots showing the proper Network and Security Group settings as required in the diagram.

Also, download the following source code for the exam problem set:

- [gumball-fall18.zip](#)
- [nodejs-fall18.zip](#)

Hint: Your "Jump Box" is open to the internet in the DMZ and has SSH (Port 22) access to all hour Private Network Instances.



Question 1

25 pts

Part 1: AWS Managed Database Instances Setup

In this section, you will configure **MySQL** and **Redis** as AWS Managed Services. Following the configuration steps below:

AWS Database Setup

- <https://aws.amazon.com/rds/> (<https://aws.amazon.com/rds/>)
- <https://aws.amazon.com/elasticache/> (<https://aws.amazon.com/elasticache/>)

RDS MySQL Instance

1. Create DB Subnet Group
2. Name: mysql
3. Description: mysql
4. VPC: cmpe281
5. Add All Subnets Related to this VPC

**** Need to have at least Two Private**

**** Subnets accross two AZ's**

1. Create an RDS MySQL Instance
2. Select "Dev/Test - MySQL" Use Case
3. Select Version 5.5.61
4. Select "Only enable options eligible for RDS Free Usage Tier"
5. Select Instance Type "db.t2.micro"
6. Instance Name: mysql
7. DB User: admin
8. DB Pass: midterm281
9. VPC: cmpe281
10. DB Subnet Group: mysql
11. Public accessibility: No
12. AZ Info: No Preference
13. Security Group: Chose or Create one with MySQL Port Open
14. DB Name: cmpe281
15. Port: 3306
16. IAM DB authentication: Disable
17. Disable auto minor version upgrade
18. Remaining Options: Use Defaults

- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.Connect.html (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.Connect.html)
- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html (https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html)

Install MySQL Client in your "Jump Box"

```
sudo yum update
Install mysql client
sudo yum install mysql
mysql -u admin -p -h <host>
```

ElasticCache Redis Instance

1. Cluster Engine: redis
2. Name: redis
3. Description: redis
4. Version: 4.0.10
5. Port: 6379
6. Parameter group: default.redis4.0
7. Node type: cache.t2.micro
8. Number of Replicas: 2
9. Subnet Group: Create New
10. Subnet Group Name: redis
11. SG Description: redis
12. VPC: cmpe281
13. Subnets: select the two private subnets
14. Security Group: Chose or Create one with Redis Port Open

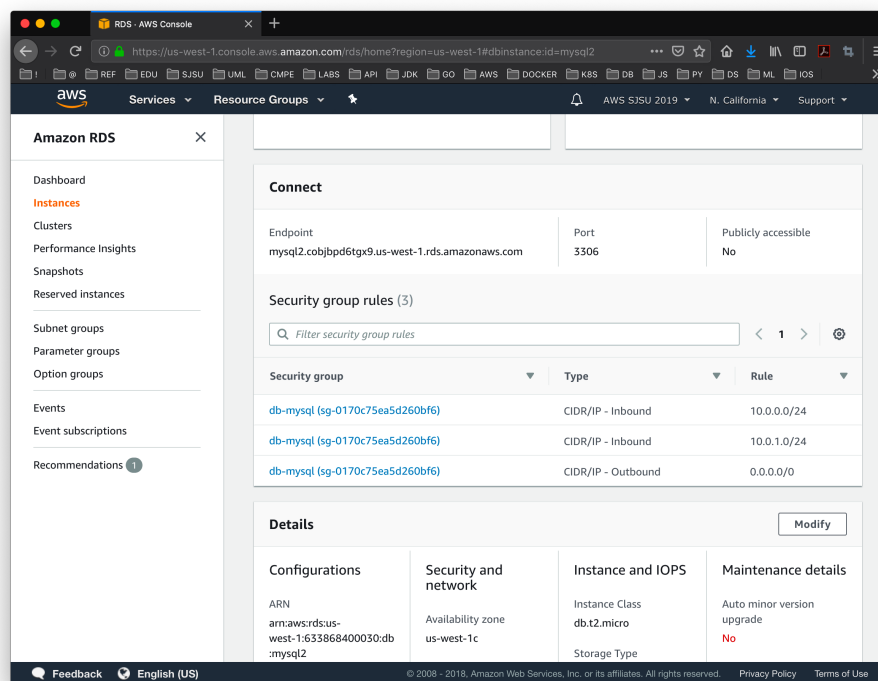
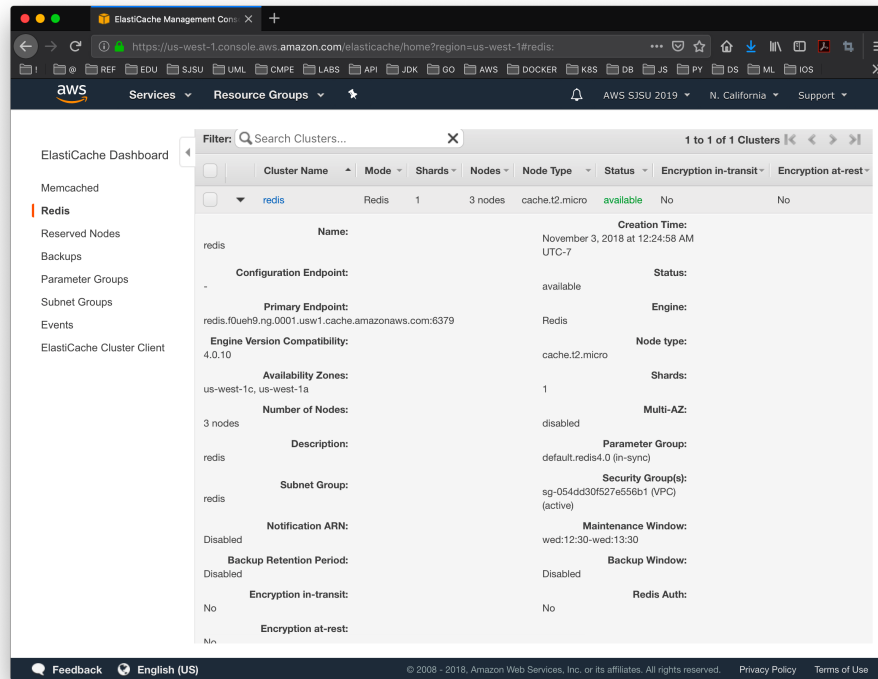
- <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/GettingStarted.ConnectToCacheNode.html> (<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/GettingStarted.ConnectToCacheNode.html>)

Install Redis Client in your "Jump Box"

```
sudo yum install gcc
sudo yum install wget
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make
src/redis-cli -h <host> -p 6379
```

Submission:

- PDF document with screenshots showing that your Redis and MySQL services are running. Make sure to show the Endpoints and Security Groups. See samples.



- From your Jump Box, Configure and Test the Databases using the following Commands

Database Setup

* Create Database Schema (DB User: root, DB Pass: cmpe281)

```
Database Schema: cmpe281
create database cmpe281 ;
use cmpe281 ;
```

* Create Database Table

```
CREATE TABLE gumball (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  version bigint(20) NOT NULL,
  count_gumballs int(11) NOT NULL,
  model_number varchar(255) NOT NULL,
  serial_number varchar(255) NOT NULL,
  PRIMARY KEY (id),
  UNIQUE KEY serial_number (serial_number)
) ;
```

* Load Data

```
insert into gumball ( id, version, count_gumballs, model_number, serial_number )
values ( 1, 0, 1000, 'M102988', '1234998871109' ) ;
```

* Verify Data

```
select * from gumball ;
```

* Check Redis Cache

```
redis-cli // start redis shell
```

```
keys * // find all keys (should return empty list)
```

Upload

Choose a File

Question 2**50 pts**

Part 2: Deploy Gumball Cluster with an Internal ELB

- **Modify the Go Gumball Code to connect to your Redis and MySQL Servers from Part 1.**
- **Build the Code and deploy it to AWS (as two EC2 Instances running as Docker Hosts)**
- **Then put these two EC2 instances behind an Internal Elastic Load Balancer**
- **NOTE: All Instances in this section are in the "Private Network"**

After you have the this up, test the Gumball API with the following CURL commands:

Note: Replace "host" with the DNS name of your ELB.

```
curl http://host:3000/ping \_ \(http://host:3000/ping\)  
curl -X GET http://host:3000/gumball \_ \(http://host:3000/gumball\)  
curl -X POST http://host:3000/order \_ \(http://host:3000/order\)  
curl -X POST http://host:3000/order/<orderid> \_ \(http://host:3000/order/<orderid>\)
```

Then connect to your MySQL and Redis instances and run the following commands:

MySQL:

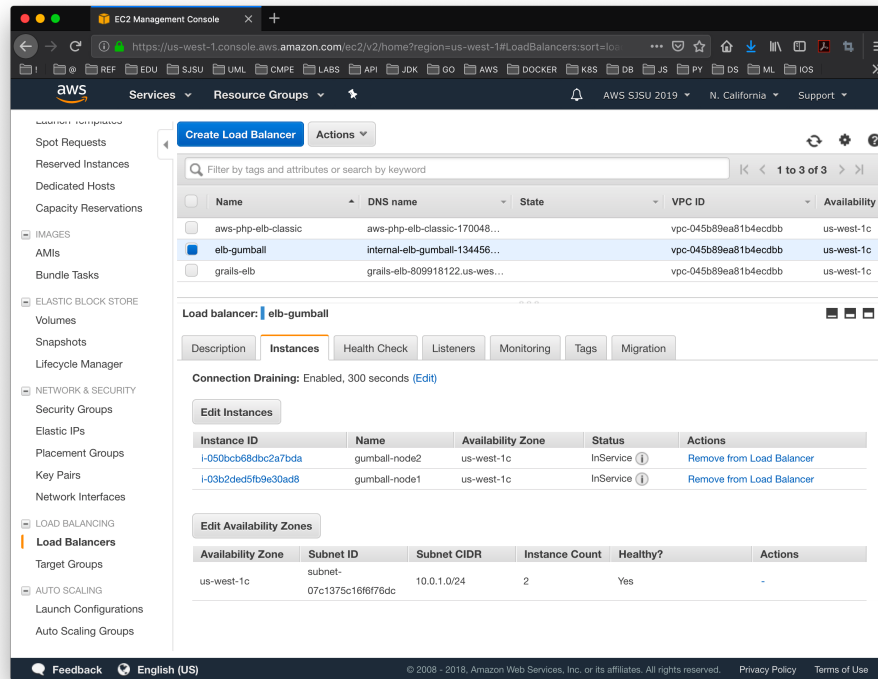
```
select * from gumball
```

Redis:

```
keys *  
get <key>
```

Submission:

- PDF document with a screenshot showing the ELB configuration with the two in-service instances. **For Example:**



- Screenhosts of the output of CURL commands
- Screenshots of the output from MySQL and Redis Queries
- Answer the questions:
 - Was the output from MySQL what you expected? Explain Why or Why Not.
 - Was the output from Redis what you expected? Explain Why or Why Not.

Upload

Choose a File

Question 3

25 pts

Part 3: Deploy Kong API Gateway with Auth Key to Gumball

- Deploy Kong API gateway on your publicly facing Docker Host (from Labs)
- Configure Kong to route to the internal Gumball ELB as an upstream API
- Add an API Key in Kong to the Gumball Load Balanced API

Submission:

- PDF document with:
 - Screenshots of the Docker Commands used to run containers
 - Screenshots showing Curl commands used to configure Kong and API Key
 - Screenshots showing Curl commands with output for the following. Note. replace "host" with the Kong API Gateway Endpoint on Port 8000.

```
curl http://host:3000/ping (http://host:3000/ping)
curl -X GET http://host:3000/gumball (http://host:3000/gumball)
curl -X POST http://host:3000/order (http://host:3000/order)
curl -X POST http://host:3000/order/<orderid> (http://host:3000/order/<orderid>)
```

Upload

Question 4

25 pts

Part 4: Modify & Deploy Node.js on a Local Docker Instance

- Modify the code in Node.js to point at your AWS Kong API Gateway
- Deploy the Node.js App to you local docker host
- Run the Node.js App

Submission:

- Submit a PDF document to include a few screenshots showing:
 - Gumball Page on Initial Startup
 - Gumball Page after inserting a coin
 - Gumball Page after paying
 - Screenshots of the Docker Commands used to build and run container

Upload

Question 5**0 pts****Part 5: Upload your Source Code.**

- Compress the source code for Go Gumball and Node.js Web App
- Upload the Zip here

Upload

Quiz saved at 11:48am