SJSU CMPE 282 HW Docker SPRING 2016

Theme: Deploy your Java-based REST server to two Docker containers

Description

In HW1, you have a REST server which includes two components: web/app and (NoSQL) database. Please build and deploy each component into a separate docker 1.9.1 (or higher) container. Both containers are running on the same machine (host1). Demonstrate the network connectivity between these two containers with your REST server by using a REST client running on a separate machine (host2).

Environment

Use a Linux VM as host1. Any platform (Linux, Windows, Mac, etc.) is fine for host2. For Linux VM, first install free VMware Player (Windows) or free Oracle VirtualBox (Mac) on your laptop, and then install say Ubuntu 14.04 as guest OS.

On host1, both containers can be deployed to either the default bridge network, or a user-created network.

Requirements

- Please utilize docker engine directly; you are not allowed to utilize docker compose, docker machine, etc.
- In order to build the necessary container, you can pull docker images directly, utilize Dockerfile to build one, build on your own, or combination of all of the above.
- By using the "--name=" option of "docker" command, name these two docker containers as app<YourName><L3SID> and db<YourName><L3SID>, respectively. For example, appDemo123 and dbDemo123. The screenshots (explained later) must include the proper container names.
- To allow REST accessing from host2, you must expose necessary ports of the web/app container on host1. You are not allowed to expose any port of the db container on host1. i.e., only the webapp container can be connected to the db container on host1.

Questions

Q1. List technologies and softwares (including version) for the REST client, web/app and DB components of the REST server. And indicate if there is any change to source code from your HW1.

For example,

REST Client: soapUI 5.2.1
REST Server: web/app component: Tomcat 7.0.67
REST Server: db component: Cassandra 2.2.4

Q2. For host1 and host2, list their machine name, IP address, and OS.

For example,

Host1: OWL, 192.168.1.1, Windows 10
Host2: ubuntu, 192.168.236.131, Ubuntu 14.04

Q3. Describe detailed steps to build and deploy these two docker containers with screenshots. (You can pull docker images directly, utilize Dockerfile to build one, or build on your own, or combination of all of the above.)

- If any, include supporting files such as Dockerfile, etc.

- Describe if you need to modify any code of REST server and why
- List known issues if any
- Any additional unique design or features you are proud of

Q4. While both containers are running on host1, include the screenshots of the followings on host1

- docker ps
- “docker network inspect” for the network to which both containers are connected. This should show the IP of both containers.
- ip addr (or ifconfig -a)

Q5. On host2, use REST client to issue the following requests and include screenshots of REST request and response (method, URL, HTTP headers):

- issue a “POST /rest/employee” request to create two employees with id 10 and 20
- issue a “GET /rest/employee“ request to retrieve all employees
- issue a “PUT /rest/employee/10“ request to update employee 10’s first name
- issue a “DELETE /rest/employee/20“ request to delete employee 20
- issue a “GET /rest/employee“ request to retrieve all employees

Submission

Submit the followings as *separate* files to Canvas

- CMPE282_HW2_<YourName>_<L3SID>.zip: any supporting files such as Dockerfile, .yml file, etc. Include source code only if there is any change from HW1.
- CMPE282_HW2_<YourName>_<L3SID> (.pdf, .doc, or .docx): the report consists of answers and screenshots to questions specified in **Question**.
  - You receive no credit if your report is not .pdf, .doc, or .docx.
  - If a screenshot is unreadable, it will be treated as if you did not turn in that screenshot.
  - If you do not follow requirements (including naming conventions), you will receive no credit.

The ISA and/or instructor leave feedback to your homework as comments and/or in Crocodoc of your submission. To access Crocodoc, click “view feedback” button. For details, see the following URL:
http://guides.instructure.com/m/4212/l/106690-how-do-i-use-the-submission-details-page-for-an-assignment

(Optional) Extra credit

In addition to the original homework, use Dockerfile and docker compose to automate container building and deployment.

- Include Dockerfile and docker-compose.yml file, and detailed steps (and screenshots)
- Demonstrate it is working by re-answering first three REST requests in Q5 (and screenshots)