SJSU CMPE 282 HW Docker FALL 2017

REMINDER: Each homework is individual. "Every single byte must come from you." Cut&paste from others is not allowed.

Theme: Deploy your Java-based REST server to two Docker containers in a vCenter server based VM

Description

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

Environment

Host1: On CMPE vCenter server, deploy a Linux VM as host1. Install docker container to host1. Both containers can be deployed to either the default bridge network, or a user-created network.

Please follow "CMPE vCenter Server Lab Rules" (see slide).

Host2: use your local laptop as host2. Any platform (Linux, Windows, Mac, etc.) is fine.

Note host1 and host2 must be two separate machines.

Requirements

- Please utilize docker engine directly; you are not allowed to utilize docker compose.
- 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 client running from host2, you must expose necessary ports of the webapp container on host1. You are not allowed to expose any port of the db container on host1. i.e.,
    - host2 can connect to only the webapp container on host1, not to the db container on host1
    - the db container on host1 can be connected only from webapp container on host1, but not from host2
- Keep your VM up until the instructor finishes grading (i.e., unmute HW) – deduction if your VM is not up!
- Inherit requirements from HW1, such as
    - URI must be /cmpe282<YourName><L3SID>/rest/…
    - The database name (i.e., keyspace in Cassandra) of NoSQL must be cmpe282<YourName><L3SID>
    - Etc.

Questions

Q1.

a. List technologies, softwares (including version), and platform (include version) for the REST client, REST server, and NoSQL.

b. For host1 and host2, list their OS (and version), and IP address.

c. A sample entire HTTP URL (including actual IP address of host1), URI, and request body for POST to create a new `employee` based on the XML format or JSON format (depending on your implementation). Also indicate if there is any additional setup (e.g., HTTP header, etc.).

For example,

a.

REST Client: soapUI 5.3.0 on Windows 10
REST Server: direct Servlet, JAXB, Tomcat 8.5.16 on (vCenter server) Ubuntu 16.04
NoSQL database: Cassandra 3.9.0 container

b.

Host1: Ubuntu 16.04, 130.65.159.171
Host2: Windows 10, 192.168.1.1 (NAT, home network)

c.

entire URL: `http://130.65.159.171/cmpe282Demo123/rest/employee`
URI: `POST /cmpe282Demo123/rest/employee`
request body:

```
<employee>
  <id>1</id>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
</employee>
```

no additional setup (no additional request header) is needed.

Q2. Describe detailed steps to **build** webapp and db docker containers with screenshots. (You can pull docker images directly, utilize Dockerfile to build one, or build on your own, or combination of the above.)

- If you pull the docker image(s) directly
  - Specify the name of the images
  - Any modifications made to the image
- If you utilize Dockerfile
  - Include Dockerfile, and any supporting files to build these two containers
  - Describe each step in Dockerfile
- If you build your own
  - List steps (including exact command lines) to build these two containers
- List known issues if any
- Any additional unique design or features you are proud of

Q3. Describe detailed steps to **deploy** webapp and db docker containers with screenshots.

- Specify the docker command line to deploy webapp and db containers, respectively. Capture its screenshot.
- Capture screenshot of "docker ps" after each container is deployed
- Describe the steps (and command lines) to deploy the REST server (e.g., .war) to containers
- Describe if you need to modify any source 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 version

- 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) - success cases only:

- issue a "POST /…/rest/employee" request to create two employees with id 10 and 20
  - screenshots: request URI, request body, and response header (status 201 and Location header)
- issue a "GET /…/rest/employee" request to retrieve all employees
  - screenshots: request URI, response body, response header (status 200)
- issue a "PUT /…/rest/employee/10" request to update employee 10's first name only
  - screenshots: request URI, request body (only firstname), and response header (status 200)
- issue a "DELETE /…/rest/employee/20" request to delete employee 20
  - screenshots: request URI, response header (status 200)
- issue a "GET /…/rest/employee" request to retrieve all employees
  - screenshots: request URI, response body (employee 10 with new first name), response header (status 200)

Submission

Submit the followings as *separate* files to Canvas

- CMPE282_HW2_<YourName>_<L3SID>.zip: any supporting files such as Dockerfile 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.
  - Keep your REST server/VM on until the instructor finishes grading (i.e., unmute HW) – deduction if site not up!

The ISA and/or instructor leave feedback to your homework as comments and/or annotated comment. To access annotated comment, 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 (up to additional 15 points)

Q6. In addition to the original homework, use docker compose to build and deploy containers in Q2 and Q3.

- Include docker-compose.yml file, and detailed steps, docker commands, and screenshots
  - If there is any manual steps (copy .war to webapp container), discuss possible "docker" ways (i.e., using docker commands only) to fully automate the entire build/deploy process without any manual steps and without any ad-hoc scripts. Assume any configuration change to web.xml in .war is done already.
- After docker compose is completed, capture screenshots of "docker ps"
- Re-run the first two cases of Q5, and capture its screenshots