Myron Zhang
Dr. Dancik
CSC 480 Independent Study
October 7, 2018

Docker and Shiny Proxy Write Up

System OS: Ubuntu 16.04 LTS

Table of Contents:

1. Build the database image

a. Change directory to the folder location of the dockerfile and my.cnf

Files required in this directory:

- dockerfile
- my.cnf
- A folder called data and within this folder is the .sql dump

Dockerfile explanation:

```
FROM mysql:8.0

ENV MYSQL_ROOT_PASSWORD=password
ENV MYSQL_DATABASE=dcast

RUN apt-get update && apt-get install -y vim

ADD my.cnf /etc/mysql/my.cnf
ADD ./data /docker-entrypoint-initdb.d
```

- FROM mysql:8.0 mean that it is pulling the pre-built image of mysql version 8.0
- Line 2 and 3 is defining the environment variable for mysql. The root password and the database we will be using. Source: Docker MySQL ENV
- Line 4 is to install vim in this mysql container. Useful for checking files in the container when testing. Source: Docker MySQL CNF
- Line 5 is to add my custom my.cnf file into the mysql image. Because the my.cnf file is located in the current directory of dockerfile, I do not need to set a path so I only need to call out the file name. /etc/mysql/my.cnf is the file location within the mysql image that I want to place my.cnf.
- Line 6 is to import the .sql dump file into the mysql container. When the container gets initialized, whatever .sql files that are found in /docker-entrypoint-initdb.d will be import to the database. This ADD instruction will add the files found within data folder into /docker-entrypoint-initdb.d. Source: Docker MySQL Ref

my.cnf explanation:

```
[mysqld]
lower_case_table_names=1
secure_file_priv=""
```

- Line 1 is the group for mysql server
- Line 2 is to change the table named stored in lowercase and name comparisons are not case-sensitive. Source: Link
- Line 3 is to remove the secure setting for data import and export operation because I want Docker to manage the storage of my database. Source: MySQL Ref and Docker MySQL Ref

b. Run in terminal: `$ docker build -t mz/mysql:1.0 .`

Command explanation:

- `docker build` is the command to build a Docker image from a Dockerfile
- `-t` is to name and optionally a tag in the ‘name:tag’ format of the image. I have named my image as mz/mysql with version number 1.0
- the `.` (dot) is to specify that the dockerfile is in the current directory

source: https://takacsmark.com/dockerfile-tutorial-by-example-dockerfile-best-practices-2018/

END OF SECTION

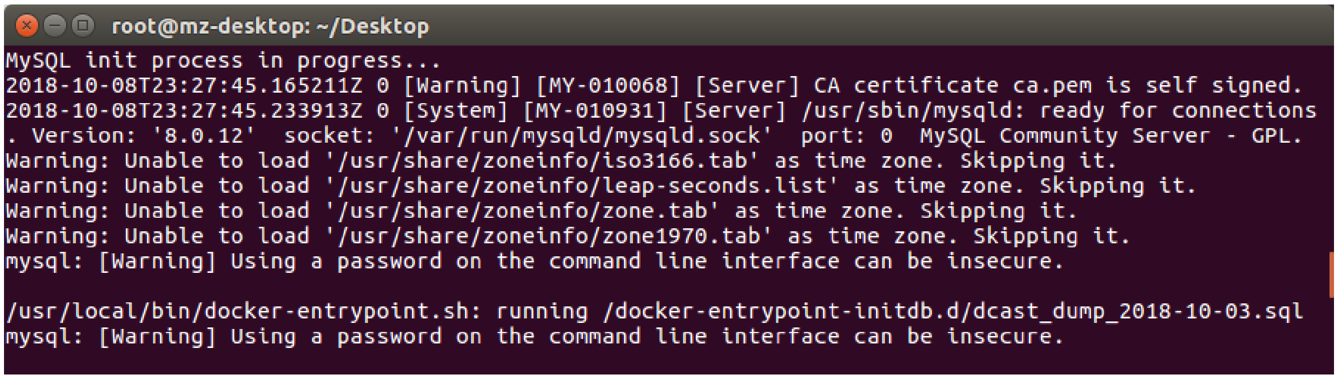
2. Run the database image

a. Run in terminal: $ docker run -it mz/mysql:1.0

Command explanation:

- docker run is the command to run the image
- -it is the combination of -i -t. -i is to keep STDIN open even if not attached. -t is to allocate a pseudo-TTY.
- Because --name was not included, docker will randomly generate a name for the container

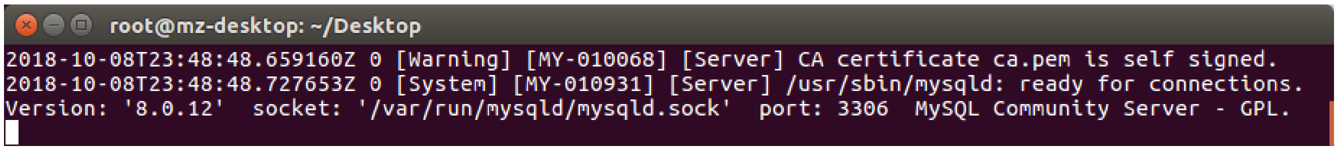
Note: While the terminal is still running /docker-entrypoint-initdb.d/dcast_dump_2018-10-03.sql, mysql is still busy importing all the data into dcast. It will take about 25 mins.



```
MySQL init process in progress...
2018-10-08T23:27:45.165211Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2018-10-08T23:27:45.233913Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections
. Version: '8.0.12'  socket: '/var/run/mysqld/mysqld.sock'  port: 0  MySQL Community Server - GPL.
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
mysql: [Warning] Using a password on the command line interface can be insecure.

/usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/dcast_dump_2018-10-03.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
```

Once done with importing, the terminal will say “[Server] /usr/sbin/mysqld: ready for connections.”



```
2018-10-08T23:48:48.659160Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2018-10-08T23:48:48.727653Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections.
Version: '8.0.12'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
```

END OF SECTION

3. Set up ShinyProxy on ubuntu 16.04 LTS
While https://www.shinyproxy.io/getting-started/ was adequate with installing ShinyProxy on ubuntu, I found this guide https://lukesingham.com/shiny-containers-with-shinyproxy/ a lot more thorough so I followed lukesingnham.com for installing Shiny Proxy.

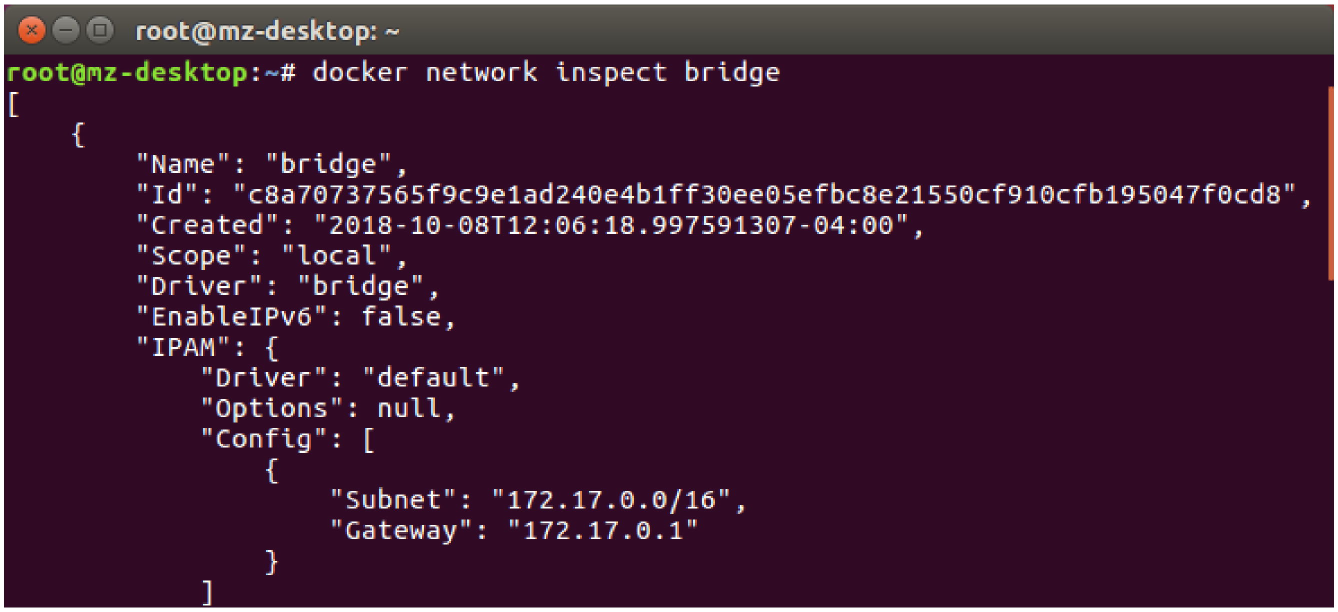
END OF SECTION

4. Inspect docker network bridge

Bridge is docker’s default network driver. Because we do not specify a driver when running the containers, it will default to bridge. By running the mysql container first, we will know that IP address assigned to it will be increment of the subnet. Gateway takes 172.17.0.1 so the next container will be assigned 172.17.0.2.

Source: https://docs.docker.com/network/

a. To get IP address of bridge, enter in terminal: $ docker network inspect bridge

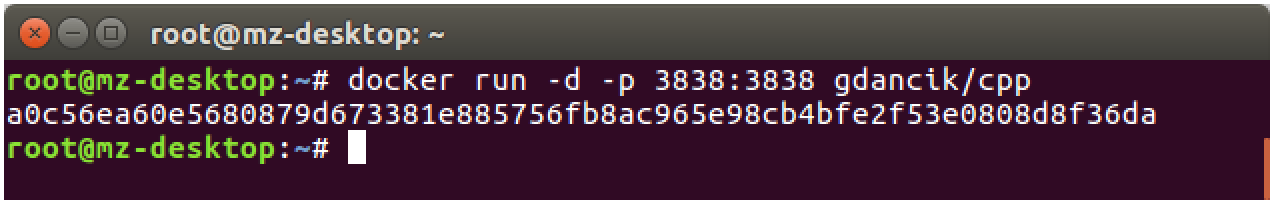


```
root@mz-desktop: ~
root@mz-desktop:~# docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "c8a70737565f9c9e1ad240e4b1ff30ee05efbc8e21550cf910cfb195047f0cd8",
        "Created": "2018-10-08T12:06:18.997591307-04:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
```

END OF SECTION

5. Editing gdancik/cpp image

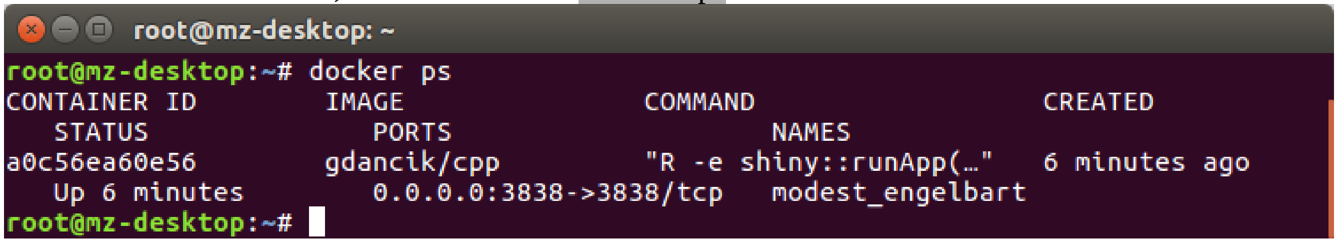
a. To run gdancik/cpp image in background, enter in terminal: $ docker run -d -p 3838:3838 gdancik/cpp



root@mz-desktop: ~

```
root@mz-desktop:~# docker run -d -p 3838:3838 gdancik/cpp
a0c56ea60e5680879d673381e885756fb8ac965e98cb4bfe2f53e0808d8f36da
root@mz-desktop:~#
```

b. Get the container name, enter in terminal: $ docker ps

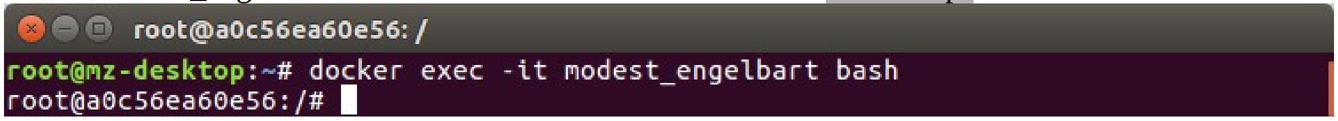


root@mz-desktop: ~

```
root@mz-desktop:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
   STATUS              PORTS                    NAMES
a0c56ea60e56        gdancik/cpp         "R -e shiny::runApp(…"   6 minutes ago
   Up 6 minutes        0.0.0.0:3838->3838/tcp   modest_engelbart
root@mz-desktop:~#
```

c. Execute an interactive bash shell on the container, enter in terminal: $ docker exec -it modest_engelbart bash

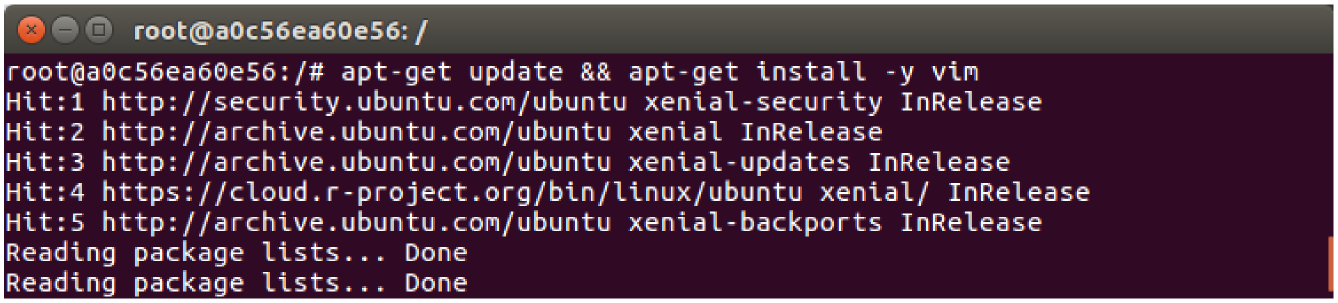
Note: "modest_engelbart" is the container name we found from $ docker ps



root@a0c56ea60e56: /

```
root@mz-desktop:~# docker exec -it modest_engelbart bash
root@a0c56ea60e56:/#
```

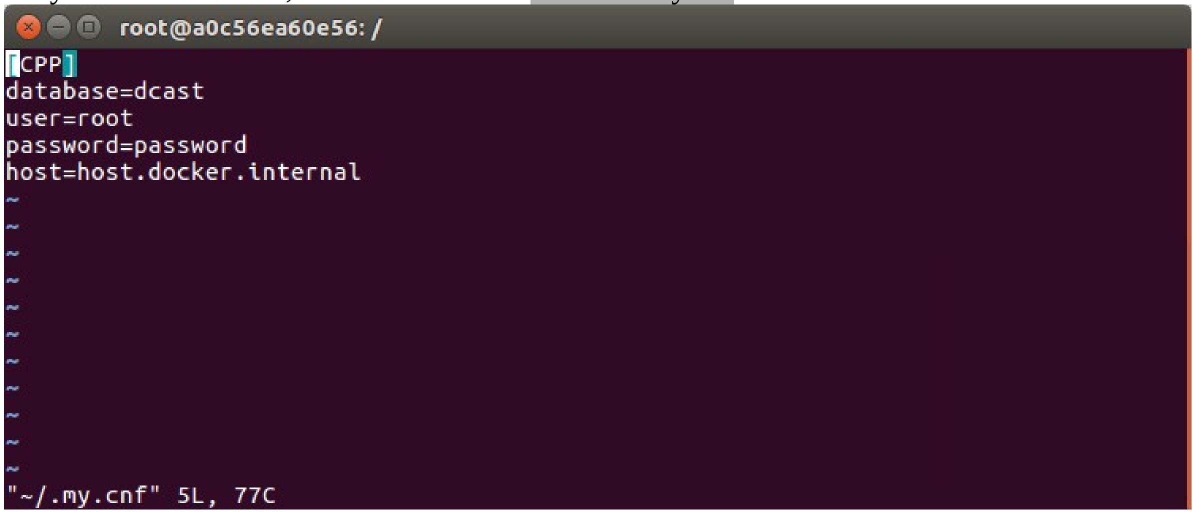
d. Install vim to container so we can edit the my.cnf file, enter in terminal: $ apt-get update && apt-get install -y vim



root@a0c56ea60e56: /

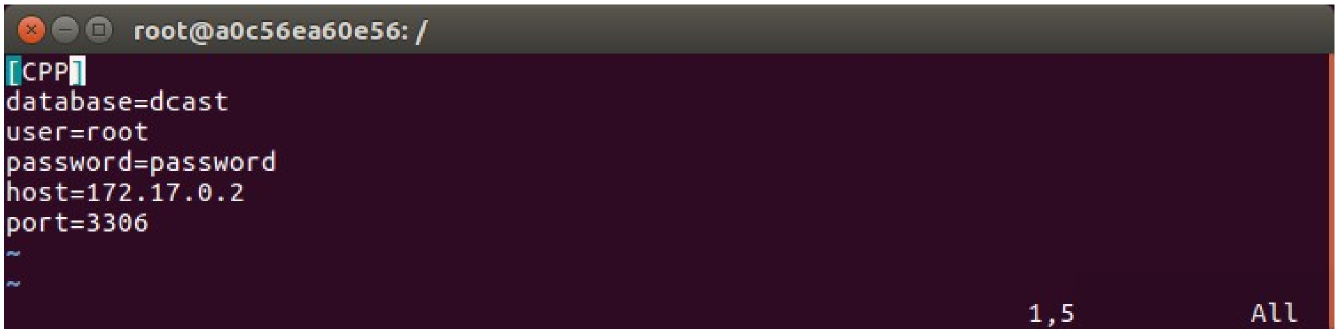
```
root@a0c56ea60e56:/# apt-get update && apt-get install -y vim
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 https://cloud.r-project.org/bin/linux/ubuntu xenial/ InRelease
Hit:5 http://archive.ubuntu.com/ubuntu xenial-backports InRelease
Reading package lists... Done
Reading package lists... Done
```

e. Edit my.cnf file with vim, enter in terminal: $ vim ~/.my.cnf



```
[CPP]
database=dcast
user=root
password=password
host=host.docker.internal
~
~
~
~
~
~
~
~
~
~
~
"~/.my.cnf" 5L, 77C
```

f. change host=host.docker.internal to host=172.17.0.2 and add port=3306. Then write the file and exit out of the container.

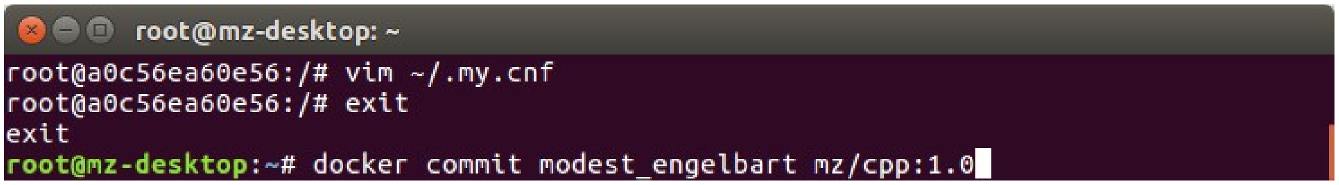


```
[CPP]
database=dcast
user=root
password=password
host=172.17.0.2
port=3306
~
~
                                                        1,5           All
```

g. Commit the change of dgancik/cpp and save it as a new image, enter in terminal: $ docker commit modest_engelbart mz/cpp:1.0

Note:

- “modest_engelbart” is the randomly generated name of the container of gdancik/cpp
- “mz/cpp:1.0” is the new image name and tag



```
root@a0c56ea60e56:/# vim ~/.my.cnf
root@a0c56ea60e56:/# exit
exit
root@mz-desktop:~# docker commit modest_engelbart mz/cpp:1.0
```

Now the new CPP image, mz/cpp:1.0, is ready to connect with the mysql database. The mysql database must be ran first before anything else. Check section “4. Inspect docker network bridge” for reason why.

END OF SECTION

6. ShinyProxy application.yml file

I used example application.yml from https://www.shinyproxy.io/configuration/ as a template.

Changes I made:

- Changed the authentication type to “simple”
- Deleted everything from ldap section.
- Changed user to mzhang
- Deleted the two example application from specs
- Added the CPP shiny application to specs

```
proxy:
  title: CPP Shiny Proxy
  logo-url: http://www.openanalytics.eu/sites/www.openanalytics.eu/themes/oa/logo.png
  landing-page: /
  heartbeat-rate: 10000
  heartbeat-timeout: 60000
  port: 8080
  authentication: simple
  users:
  - name: mzhang
    password: password
  # Docker configuration
  docker:
    cert-path: /home/none
    url: http://localhost:2375
    port-range-start: 20000
  specs:
  - id: 01_cpp
    display-name: CPP Application
    description: mz/cpp:1.0 image
    container-cmd: ["R", "-e", "shiny::runApp('/root/CPP')"]
    container-image: mz/cpp:1.0
logging:
  file:
    shinyproxy.log
```

END OF SECTION

7. Running CPP with ShinyProxy.jar

a. The database must be running first. Follow section “2. Running the database image” if database is not already running.

b. The application.yml file has to be the same directory as shinyproxy.jar file. My jar file was located in `/opt/shinyproxy/` so I moved my application.yml file there.

c. Change your directory to the location of shinyproxy.jar file.

d. To run shiny proxy, enter in terminal: `$ java -jar shinyproxy.jar`

e. In the browser enter http://localhost:8080/ and sign in with “mzhang” and “password” as user name and password respectively.

END OF SECTION