Course › Chapter 3: M... › Guided Exer... › Guided Exer...

Guided Exercise: Managing a MySQL Container

In this exercise, you will create and manage multiple MySQL server containers.

Resources	
Image:	RHSCL MySQL 5.6 container image (**`rhscl/mysql-56-rhel7`**)

Outcomes

You should be able to create and manage multiple MySQL database containers.

Before You Begin

You need to have CDK 3 installed and configured to start this exercise. Check the **Lab Set-up Instructions** to learn how to install and configure the CDK 3.

Check if your Minishift VM is running with the command:

```
$ minishift status
Running
```

If the Minishift VM is not running, start the VM using the command:

```
$ minishift start
```

Steps

1. Open a shell inside the Minishift VM. All Docker commands will be run inside the VM:

   ```
   $ minishift ssh
   [docker@minishift ~]$
   ```

   The shell prompt should change to indicate that you are now inside the Minishift VM.

2. Start the first MySQL server container using the following command:

   ```
   [docker@minishift ~]$ docker run --name mysql-1st rhscl/mysql-56-rhel7
   ```

   This command downloads the MySQL database container image from Red Hat and tries to start it, but it does not start. The reason for this is the image requires a few environment variables to be provided.

   > ### Note
   >
   > If you try to run the container as a daemon (**-d**), the error message about the required variables is not displayed. However, this message is included as part of the container logs, which can be viewed using the following command:
   >
   > ```
   > [docker@minishift ~]$ docker logs mysql-1st
   > ```

3. Check that the container exited:

```
[docker@minishift ~]$  docker ps -a | grep mysql
b2b8a74502f9    rhscl/mysql-56-rhel7   "container-e
ntrypoint"   4 seconds ago   Exited (1) 3 seconds a
go    mysql-1st
```

4. Start a second MySQL server container, providing the required environment variables. Specify each variable using the **-e** option.

```
[docker@minishift ~]$ docker run --name mysql-2nd \
    -e MYSQL_USER=user1 -e MYSQL_PASSWORD=mypa55 \
    -e MYSQL_DATABASE=items -e MYSQL_ROOT_PASSWORD=
r00tpa55 \
    -d rhscl/mysql-56-rhel7
```

5. Verify that the container was started correctly. Run the following command:

```
[docker@minishift ~]$ docker ps | grep mysql
CONTAINER ID    IMAGE                  COMMAND
CREATED             STATUS      PORTS
NAMES
5cd89eca81dd    rhscl/mysql-56-rhel7   "container-e
ntrypoint"   9 seconds ago   Up 8 seconds    3306/t
cp
mysql-2nd
```

6. Inspect the container metadata to obtain the IP address from the MySQL database server container:

```
[docker@minishift ~]$ docker inspect -f '{{ .Networ
kSettings.IPAddress }}' mysql-2nd
172.17.0.2
```

Make a note with the IP address because you will need it for the next steps.

7. Create a third container to run a MySQL client to connect to the database server running on the second container.

   Use the MySQL server container image, but without running its default entry point. Execute the Bash shell instead:

   ```
   [docker@minishift ~]$ docker run --name mysql-3rd -
   it rhscl/mysql-56-rhel7 bash
   bash-4.2$
   ```

   Notice that this container image displays a Bash prompt for a regular user. Different from the MySQL image from the Docker Hub, the image provided by Red Hat does not run as the **root** user.

8. Try to connect to the local MySQL database:

   ```
   bash-4.2$ mysql
   ```

   The following error is displayed:

   ```
   ERROR 2002 (HY000): Can't connect to local MySQL se
   rver through socket '/var/lib/mysql/mysql.sock' (2)
   ```

   This error is expected because the MySQL database server was not started in the third container.

9. Connect to the remote MySQL server in the second container, from the third container. Notice the IP address should be the one you got from Step 6. It might be different from ***172.17.0.2***.

```
bash-4.2$ mysql -uuser1 -h 172.17.0.2 -pmypa55 item
s
```

10. You are connected to the **items** remote database. Create a new table:

```
MySQL [items]> CREATE TABLE Courses (id int NOT NUL
L, name varchar(255) NOT NULL,
    PRIMARY KEY (id));
```

11. Insert a row into the table by running the following command:

```
MySQL [items]> insert into Courses (id, name) value
s (1,'DO081x');
```

12. Exit from the MySQL prompt:

```
MySQL [items]> exit
```

13. Exit from the **bash** shell:

```
bash-4.2$ exit
```

14. When you exit the **bash** shell, the third container was stopped. Verify that the container **mysql-3rd** is not running, but the second container is still up:

```
[docker@minishift ~]$ docker ps | grep mysql
CONTAINER ID        IMAGE                  COMMAND
CREATED             STATUS
PORTS               NAMES
8b2c0ee86419        rhscl/mysql-56-rhel7   "contain
er-entrypoint"   4 minutes ago       Up 4 minutes
3306/tcp            mysql-2nd
```

15. Delete the containers and resources created by this exercise.

    15.1 Stop the second container by running the following command:

    ```
    [docker@minishift ~]$ docker stop mysql-2nd
    ```

    15.2 Remove the containers ephemeral data by running the following commands:

    ```
    [docker@minishift ~]$ docker rm mysql-1st
    [docker@minishift ~]$ docker rm mysql-2nd
    [docker@minishift ~]$ docker rm mysql-3rd
    ```

    15.3 Remove the container image by running the following command:

    ```
    [docker@minishift ~]$ docker rmi rhscl/mysql-
    56-rhel7
    ```

    15.4 Leave the Minishift VM:

```
[docker@minishift ~]$ exit
logout
$
```

Notice you should be back to your operating system prompt.

This concludes the guided exercise.

Discussion

Hide Discussion

Topic: Chapter 3 / Guided Exercise: Managing a MySQL Container

Add a Post

Show all posts ▼ by recent activity ▼

All perfect

All perfect

2

Mr.

helpful

2

? What "container-entrypoint" means in the COMMAND section when you run docker ps command?

What "container-entrypoint" means in the COMMAND section when you run do...

4

Docker run rhscl/httpd-24-rhel7 error

I got after executing this command this **error**: > Unable to find image 'rhscl...

6

Step 9 command displays access denied for user

ERROR 1044 (42000): Access denied for user 'user1'@'%' to database 'items'

2

? It is not possible to use the IP address from a shell in the host machine (outside Docker). Why?

I managed to use the MySQL client when it is in a container but it doesn't work i...

3

Container Hopping!

How wild! I just hopped from one container to another! The reminiscent resem… 1

Syntax in step 9

The syntax in step 9 did not work for me. I used the following syntax; mysql -h <… 2

Stopping other containers

It looks like the other running containers need to be stopped before all contain… 2

? docker-registry-cli command

The above was used in a video. Is this part of the CDK? Part of openshift? It is no… 1

During the exercise issue with creating 1st and 3rd container using the mentioned command

When creating the mysql-1st and mysql-3rd image, i got the error with the speci… 4