Running MongoDB in Docker Containers

The following instructions can be used to install MongoDB in a Docker container and run database queries in IBM LinuxONE Community Cloud instances. This MongoDB example consists of a database collection of restaurant documents. The example will demonstrate a query and an insertion of a document in the collection.

Prerequisites

Sign up for a LinuxONE Community Cloud trial account

If you have not done so already, register at http://www.ibm.com/linuxone/try for a 120-day trial account. You will receive an email containing credentials to access the LinuxONE Community Cloud self-service portal. This is where you can start exploring all our available services.

Deploy a virtual server instance

If you have not deployed a virtual server already, please follow these instructions http://developer.ibm.com/linuxone/wp-content/uploads/sites/57/virtual-servers-quick-start.pdf to create one before proceeding. Make sure you select a flavor (resource definition) with 4 GB of memory (Medium).

This quick-start guide has been tested with the following Linux distributions:

- Red Hat Enterprise Linux (RHEL) 7.2
- SUSE Linux Enterprise Server (SLES) 12
- Ubuntu 16.04

Part A: Install Docker

1. Log on to your virtual server with the ‘linux1’ user for RHEL and SLES, or the ‘ubuntu’ user for Ubuntu.

   ```
   ssh -i <ssh key> linux1@<server IP address>
   ```

   or

   ```
   ssh -i <ssh key> ubuntu@<server IP address>
   ```

   or use an SSH client like PuTTY.

2. Switch to root user.

   ```
   sudo su -
   ```

3. Download and install Docker files.

   RHEL:

```
a. wget ftp://ftp.unicamp.br/pub/linuxpatch/s390x/redhat/rhel7.2/docker-1.11.2-rhel7.2-20160623.tar.gz

b. tar -xvzf docker-1.11.2-rhel7.2-20160623.tar.gz

c. mv docker-1.11.2-rhel7.2-20160623/docker* /bin/
```

SLES:

`Docker already installed.  Proceed to Step 4.`

Ubuntu:

```
a. apt-get -y install docker.io=1.10.3-0ubuntu6
```

4. Start the Docker daemon.

   RHEL, SLES:

   ```
   a. docker daemon -g /local/docker/lib &
   b. Hit "enter" or "return"
   ```

   Ubuntu:

   `Daemon already running.  Proceed to Part B.`

Part B: Download and Install MongoDB

1. Create a local directory to store MongoDB data.

   ```
   mkdir -p /local/docker/mongo-data
   ```

2. Download and run the MongoDB image in a Docker container.

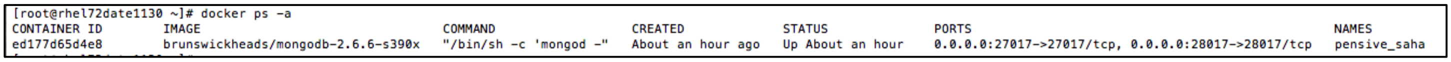
   ```
   docker run -v /local/docker/mongo-data:/mongodb/data -p 27017:27017 -p 28017:28017 -d sinenomine/mongodb-s390x
   ```

Part C: Import an Example Database

1. Note the Docker Container ID running MongoDB.

   ```
   docker ps –a
   ```

   For example,



```
[root@rhel72date1130 ~]# docker ps -a
CONTAINER ID        IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                              NAMES
ed177d65d4e8        brunswickheads/mongodb-2.6.6-s390x   "/bin/sh -c 'mongod -"   About an hour ago   Up About an hour    0.0.0.0:27017->27017/tcp, 0.0.0.0:28017->28017/tcp   pensive_saha
```

2. Start a Bash session within the Docker container.

   ```
   docker exec -it <CONTAINER ID> bash
   ```

3. Download the example restaurant collection.

```
curl -O https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-
dataset.json
```

4. Import the collection into the test database.

```
mongoimport --db test --collection restaurants --drop --file ./primer-dataset.json
```

Part D: Interact with the Example Database in MongoDB

1. Start the MongoDB service.

```
mongo
```

2. Connect to the test database to access the restaurant collection.

```
use test
```

3. Query the database for all ‘Bakeries’.

```
db.restaurants.count( { "cuisine": /Bakery/ } )
```

   The result should show 691.

4. Insert a new ‘Bakery’ document.

```
db.restaurants.insert({"address" : { "building" : "123", "coord" : [ -73.9434351, 40.6075879 ], "street"
: "First Street", "zipcode" : "12345" }, "borough" : "Borough", "cuisine" : "Bakery", "grades" : [ {
"date" : ISODate("2016-01-01T00:00:00Z"), "grade" : "A", "score" : 10 } ], "name" : "New Bakery",
"restaurant_id" : "12345678" })
```

5. Query the database for all ‘Bakeries’.

```
db.restaurants.count( { "cuisine": /Bakery/ } )
```

   The result should show 692.

Extras

Handy Commands

To stop all Docker containers:

```
docker stop $(docker ps -a -q)
```

To remove all Docker instances:

```
docker rm -f $(docker ps -a -q)
```

To remove all Docker images:

```
docker rmi -f $(docker images -q)
```

Reference

Links

https://docs.docker.com/engine/userguide/intro/

https://docs.docker.com/engine/installation/

https://docs.mongodb.com/getting-started/shell/