Docker : Docker MasterClass for DevOps

Lab : Assignment Solution - Data Volumes in Docker Container

Docker : Service Containers

- Create mysql container with some specific version with Volume named **mysql-db**

- Start Mysql Container

  ```
  docker container run - -name=mysqltest mysql
  ```

- Stop/remove MySQL Container and Start with user Defined Volumes

  ```
  docker run --name=test-mysql --
  env="MYSQL_ROOT_PASSWORD=mypassword" - -mount source=mysql-db,
  target=/var/lib/mysql mysql
  ```

- Verify MySQL Containers

- Go to MySQL DataBase and Create Data

Docker : Service Containers

➢

Docker : Service Containers

- Go to MySQL DataBase and Create Data

- Inspect container to find the IP

- Get the Running Port

- Install MySQL client package.
  apt-get install mysql-client

- Execute Command to login MySQL DB

- mysql -u root -p <password> -h <hostIP> -P <MySQL_Port>
  mysql -u root -p mypassword -h 172.17.0.20 -P 3306

Docker : Service Containers

- Create DataBase

  ```
  CREATE DATABASE databasename;
  ```

- Create Table in DataBase

  ```
  CREATE TABLE Persons ( PersonID int, LastName varchar(255), FirstName
  varchar(255), Address varchar(255), City varchar(255) );
  ```

- Insert Some Data into the Table

  ```
  INSERT INTO Persons (PersonID, LastName, FirstName, Address, City)VALUES (14,
  'B. Erichsen', ’Tom’, ‘Skagen 216’, 'Norway');
  INSERT INTO Persons (PersonID, LastName, FirstName, Address, City)VALUES (17,
  'Zbyszek', ’Wolski’, ‘Keskuskatu 45’, ‘Finland');
  ```

- Verify DataBase

  ```
  Select * From Persons;
  ```

Docker : Service Containers

- Stop and Remove the Running Container.

- Start a New MySQL Container with earlier Data Volumes.

  ```
  docker run - -name=test-mysql-secound --
  env="MYSQL_ROOT_PASSWORD=mypassword" - -mount source=mysql-db,
  target=/var/lib/mysql mysql
  ```

- Go to the DataBase

- Verify the Data that user have created in earlier Container.

Thank You…

Don’t be the Same! Be Better!!!