Docker-Kubernetes Case study Exercise

Instructor: Prakash Badhe.

There are two applications interconnected and defined here.

1. The user-service.war run as war in Tomcat 8.0 container and connecting to mysql database server ver. 5.6 with host name as localhost.
2. MySQL 5.6 as backend database for the above war application with shared volume from host system.

Define initial table and records of users to be added in database with the table.sql and records.sql script (optional with mysql_init folder as shared volume.)

1. Build and publish the docker image for user-service.war with tomcat dependency and named as tomcat-user-service image to run. Define the in compose YML file to include this image.

2. Define the directory from the host file system as shared volume for the MYSQL Database server 5.6.

3. Define the docker-compose yml file to create the Database container by specifying container name to be linked to tomcat-user-service container image.

Define Docker-compose.yml for combining both the above containers.

Run with docker-compose and verify add/get/remove records with CURL to the REST API.

4. Verify the initial records by running the user-service URL from browser as Get Method URL : http://localhost:8090:user-service/users

   Post Method URL : http://localhost:8090:user-service/user

5. Verify by scaling up the container instances.

10. Implement Docker Swarm cluster for the above docker compose config and verify the load balancing by increasing and reducing the replicas.

11. Implement the above application containers with Kubernetes Minicube cluster and verify with browser and monitor with minikube dashboard.

12. Deploy the above docker compose in cloud containers and verify from client side.

13. Deploy the above application with Kubernetes cluster in cloud and verify from client side.

****