**Steps to run the application locally (Prerequisite : maven, java to be installed)**

Do these steps only if you want to run the application on your system without deploying to kubernetes

Step 1:

Cd cloud-native

If you want to run the app with mssql server, update the username and password of the datasource in the /src/main/resources/mssql.properties file (Currently they are configured to the instance that will be spun on kubernetes cluster)

If you want to run the app with postgres server, update the username and password of the datasource in the /src/main/resources/pgsql.properties file (Currently they are configured to the instance that will be spun on kubernetes cluster)

Mvn clean install

Step 2:

If you want to run the application with a MS SQL Server use the following command:

Java –jar –Dspring.profiles.active=mssql cloud-native.jar

If you want to run the application with a Postgres SQL Server run the following command:

Java –jar –Dspring.profiles.active=pgsql cloud-native.jar

**Steps to build and run the docker image:**

Cd cloud-native

If you want to run the app with mssql server, update the username and password of the datasource in the /src/main/resources/mssql.properties file (Currently they are configured to the instance that will be spun on kubernetes cluster)

If you want to run the app with postgres server, update the username and password of the datasource in the /src/main/resources/pgsql.properties file (Currently they are configured to the instance that will be spun on kubernetes cluster)

Mvn clean install

Step 2:

Docker build –t hvh208/v2 .

In the above step hvh208/v2 is the name of my docker repository.

Step 3:

If you want to run the application with a MS SQL Server use the following command:

Docker run –e ‘spring.profiles.active=mssql” hvh208/v2

If you want to run the application with a Postgres SQL Server run the following command:

Docker run –e ‘spring.profiles.active=pgsql” hvh208/v2

**Steps to deploy on the kubernetes/minikube cluster**

The manifest files to deploy the app with MS SQL instance can be found in /mssql folder

The manifest files to deploy the app with Postgres SQL instance can be found in /postgressql folder

The manifest files to deploy the application are pulling the application image hvh208/v2 which is my public repository.

Cd cloud-native

./run.sh postgres

To deploy the application with ms sql use the command:

Cd cloud-native

./run.sh

Once the deployment is done, to get the ip address of the application:

Minikube service –url lifoapp

Once we have the ip address access the swagger using ip\_address/swagger-ui.html

**About the Application**

The application basically exposes two endpoints ip\_address/lifo/push and ip\_address/lifo/pop

Push end point is a post endpoint and lets y+ou push the data in the following request format:

{ "data": "string"}

The response of this /push end point will be:

{"id":1,"createdTime":"2020-07-20T04:12:53.886","data":"string"}

Pop end point is a delete endpoint which does not take any input

The response of the pop end point, if data is present :

{"id":1,"createdTime":"2020-07-20T04:12:53.886","data":"string"}

The response of the pop end point if no data is present :

{"timestamp":"2020-07-20T04:13:12.310+0000","status":404,"error":"Not Found","message":"No Data Found In The Persistent Store, Please push some data to pop","path":"/lifo/pop"}

Curl commands:

curl -X POST "http://ip\_address/lifo/push" -H "accept: \*/\*" -H "Content-Type: application/json" -d "{ \"data\": \"string\"}"

curl -X DELETE "http://ip\_Address/lifo/pop" -H "accept: \*/\*"