Advance JAVA Project – Microservices

Aharon & Gil own a very successful restaurant, but this restaurant doesn’t have on-line food order system, because of the Covid-19 (corona virus ☺ ) all the restaurants are close, so we thought it is a good time to be able to have an on-line site that **customers will be able to order food**

Your mission is to build a simple restaurant service where customer can order food on-line

Pre – requirements:

1. Plan a simple design diagram for these services
2. Using **microservice** architectural is a must (can use spring-boot)
3. This should be able to run inside Kubernetes cluster

Assumptions:

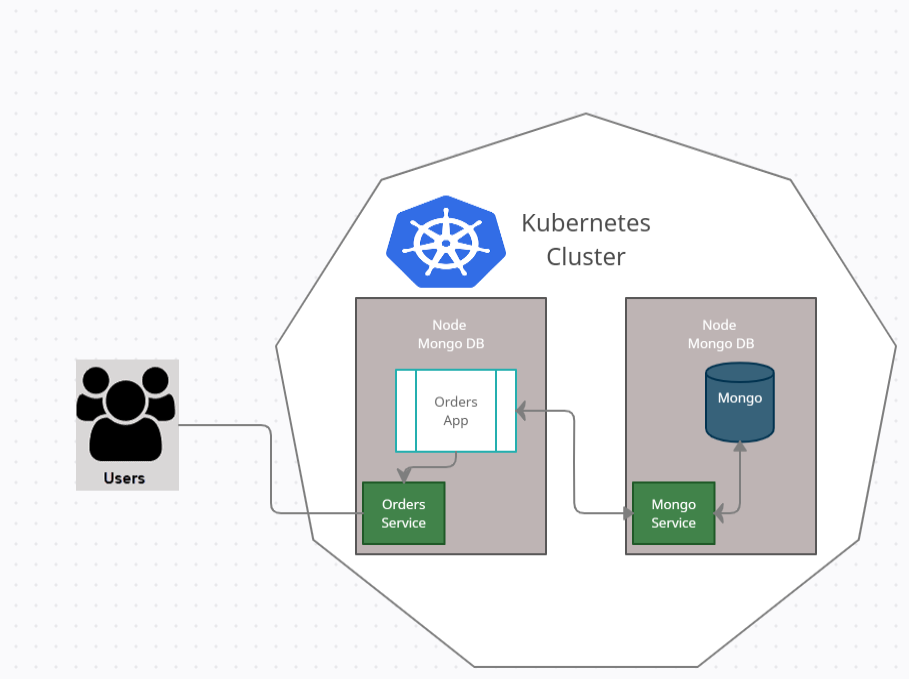
1. There is **no need to build a UI for this**.
2. You will need to have a Docker file image - (you can use docker desktop <https://docs.docker.com/desktop/>
3. For Kubernetes installation you can use minikube <https://kubernetes.io/docs/tasks/tools/>

Please read and understand how this can help you

**Think how to show us this is working**

Good Luck

Simple design diagram



* User order from the restaurant
* Orders app store the orders in the Mongo database
* Kitchen get all pending orders
* Kitchen change order status to In progress
* etc

Installation

1. minikube
2. Kubernetes
3. VS code add-on's to java and more spring-boot
4. minikube add-ons enable e ingress
5. helm for public yaml's
   1. template engine
   2. for version
6. persists storage and storage class

compilation and running commands

|  |  |  |
| --- | --- | --- |
| # | Command | Description |
|  | mvn compile | Compile |
|  | mvn package | Set package |
|  | mvn clean install spring-boot:run | Compile and run on local machine |
|  | docker container inspect --format '{{.NetworkSettings.Networks.nat.IPAddress}}' acb298e7f7f3 | Get container ip, get container id with  docker ps -a |
|  | docker login | Login to docker hub, interactive |
|  | docker build -t itziksela/orders:1.0.5 . | Build image with docker hub username  And tag version |
|  | docker push itziksela/orders:1.0.5 | Push image to docker hub |
|  | kubectl delete deployment orders | Delete deployment before create new |
|  | kubectl create deployment orders --image=itziksela/orders:1.0.5 | Create deployment on minikube |
|  | kubectl apply -f .\configuration\orders.yml | Apply config yaml to deploymnet |
|  | minikube service <service name>  like: minikube service orders | Open app service in browser, get service info   * Get service name with kubectl get service * Open the console with admin rights |

command for mongo and mongo express

|  |  |  |
| --- | --- | --- |
| # | Command | Description |
|  | kubectl apply -f .\configuration\mongo-secret.yml | Apply secret |
|  | minikube service mongodb-express-service | Open the service on browser – need admin on console. note that the service kubectl get services is in pending status, in order to give it IP, run this comman! |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

kubectl basic command (CRUD)

|  |  |  |
| --- | --- | --- |
| # | Command | Description |
|  | kubectl get nodes | List of nodes – only minikube |
|  | kubectl get pod |  |
|  | kubectl get services | List of open services (actual interface of apps) |
|  | kubectl get deployment | List of deployment |
|  | kubectl get all | Get all data |
|  | kubectl edit deployment | Edit deployment |
|  | kubectl get replicaset | List of replicaset of pod |
|  | kubectl create deployment nginx-depl --image=nginx | Create deployment (abstract of pod) |
|  | kubectl logs <id> | Get the logs of pod id |
|  | kubectl exec –it <pod id> -- bin/bash | Get into the pod machine, get the id by kubectl get pod |
|  | kubectl delete deployment <id> | delete pod/ deployment/ replicaset with id |
|  | kubectl apply –f filename.yml | Create deployment with configuration file |
|  | kubectl describe service <name> -o wide | Get service details |
|  | minikube service list | Get service list |

TASK: reload the mongo environment and connect it to app

current status:

* app running without mongo
* able to build app
* able to build the docker
* unable to run the docker – failed because missing mongo

command for app docker

|  |  |  |
| --- | --- | --- |
| # | Command | Description |
|  | mvn clean install spring-boot:run | Build and run the code |
|  | docker build -t myorders:1.0.5 . | Build docker with myorders tag |
|  | docker images | List of active images |
|  | docker run myorders:1.0.5 | Run the docker |
|  | docker stop <id> | Stop docker by container id, get the id by docker ps –a |
|  | docker rm <id> | Delete docker by container id, get the id by docker ps –a |
|  | docker ps –a | List of containers |
|  | docker rmi c0665711ef25 | Remove image by id, get id by docker images |
|  | Docker logs <id> | Docker logs by container id |
|  | Docker tag <repository tag> <new repository tag> | Rename repository |
|  | Docker push repository tag | Push repository to remote repository |

Note

Stop container, Remove container then remove image