1.OpenShift is a Kubernetes-based platform with added functions

**prerequisites**

**Docker:** You need a containerization software for building containers.

**OpenShift CLI:** You need the OpenShift command-line tool oc to interact with your Kubernetes cluster.

To verify that the OpenShift CLI is installed correctly, run the following command:

***oc version***

**Sample OP :**

Client Version: version.Info{Major:"4", Minor:"1+", GitVersion:"v4.1.14", ... }

Before you can deploy your microservices, you must gain access to a cluster on OpenShift.

To login to OpenShift by using the CLI, navigate to the online web console by following the [username] > Copy Login Command > Display Token > Log in with this token path.

***oc login --token=[your-token] --server=https://api.[region].online-starter.openshift.com:[port]***

To create the project

***oc new-project [project-name]***

After building the application build the docker image for the application.

***docker build -t [jar-name] system/.***

***docker build -t [jar-name] inventory/.***

The -t flag in the docker build command allows the Docker image to be labeled (tagged) in the name[:tag] format. The tag for an image describes the specific image version. If the optional [:tag] tag is not specified, the latest tag is created by default.

To list out the docker images:

***docker images***

In order to run the microservices on the cluster, you need to push the microservice images into a container image registry.

First, you must authenticate your Docker client to your OCR. Start by running the login command:

***oc registry login***

You can store your Docker credentials in a custom external credential store, which is more secure than using a Docker configuration file.  If you are unsure where your credentials are stored, use the following command:

***oc whoami***

***oc whoami -t***

***oc registry info***

Replace the square brackets in the following docker login command with the results from the previous commands:

***docker login -u [oc whoami] -p [oc whoami -t] [oc registry info]***

The command authenticates your credentials against the internal registry so that you are able to push and pull images.

You can also view the registry address by running the following command:

***oc registry info***

Ensure that you are logged in to OpenShift and the registry, and run the following commands to tag your applications:

***oc registry info***

***oc project -q***

Replace the square brackets in the following docker tag commands with the results from the previous commands:

***docker tag system:1.0-SNAPSHOT [oc registry info]/[oc project -q]/system:1.0-SNAPSHOT***

***docker tag inventory:1.0-SNAPSHOT [oc registry info]/[oc project -q]/inventory:1.0-SNAPSHOT***

Finally, push your images to the registry:

***oc registry info***

***oc project -q***

Replace the square brackets in the following docker push commands with the results from the previous commands:

***docker push [oc registry info]/[oc project -q]/system:1.0-SNAPSHOT***

***docker push [oc registry info]/[oc project -q]/inventory:1.0-SNAPSHOT***

After you push the images, run the following command to list the images that you pushed to the internal OCR:

***oc get imagestream***

### **Deploying the microservices**

Now that your container images are built, deploy them by using a Kubernetes object configuration file.

Kubernetes objects can be configured in a YAML file that contains a description of all your deployments, services, or any other objects that you want to deploy. All objects can also be deleted from the cluster by using the same YAML file that you used to deploy them.

Run the following commands to deploy the objects as defined in kubernetes.yaml file:

***oc apply -f kubernetes.yaml***

When the apps are deployed, run the following command to check the status of your pods:

***oc get pods***

## Making requests to the microservices

To access the services and the application, use a route. A route in OpenShift exposes a service at a hostname such as www.your-web-app.com so external users can access the application.

Your microservices can now be accessed through the hostnames that you can find by running the following command:

***oc get routes***

They can also be found in the web console by following the Networking > Routes > Location path. Hostnames are in the inventory-route-[project-name].apps.[region].starter.openshift-online.com format.

## Tearing down the environment

When you no longer need your deployed microservices, you can delete the Kubernetes deployments, services, and routes by running the following command:

***oc delete -f kubernetes.yaml***

***oc delete imagestream/inventory***

***oc delete imagestream/system***

Finally, you can delete the project by running the following command: