

Exercise 3 - Kubernetes basics

Assignments

1. Create a yaml file that has the following Object Definitions:
 1. Object definition for mosquito broker (use deployment or deploymentConfig, not pod!)
 2. Service
 3. ConfigMap for mosquitto.conf
 4. ConfigMap or Secret for server.crt, ca.crt and server.key (MQTT certificates)
 5. Route

tips:

Use the official eclipse-mosquitto:latest image.

Use only one replica.

Configure mosquitto broker with correct configuration - encrypted unauthenticated access. Broker needs to allow anonymous access.

Use ports 1883 and 8883! Use both!

Add ConfigMaps (mosquitto configuration and certificates) to mosquitto Deployment as volumeMounts and volumes.

Use volumeMounts.mountPath: /mosquitto/config/mosquitto.conf for mosquitto broker.

Use ca.crt, server.crt and server.key for enabling encrypted access. For now, it's probably better to use configMap rather than Secret for the certs.

Mount the certs into /mosquitto/config/certs folder and alter mosquitto.conf filepath to match.

In Route object specification name host with mqtt. + your studentid. +rahtiapp.fi (i.e.

mqtt.studentXXX.rahtiapp.fi) and use tls.termination: passthrough USE ONLY CHARACTERS a-z AND minus sign (-) IN SERVICE NAMES!

2. If you have not yet done a Rahti cloud project, navigate to <https://rahti.csc.fi:8443>. Log in and create a project. Important! Add course assistants with admin access to the project. Log into rahti command line interface. In GUI click your name in the upper right corner. Select copy login command. Paste into terminal.

3. Use oc create command to create the objects defined in your yaml file in Rahti cloud.

tips:

Take screenshots from "oc create" command. Inspect your deployment with "oc get all". Take screenshots. Use "oc get pod" to get pod name, should be something like mosquitto-5cdf784bf8-dh729. Use "oc exec -i -t -- sh" (e.g. "oc exec mosquitto-5cdf784bf8-dh729 -i -t -- sh") to get access inside the container. Once inside use "ls mosquitto/config/certs" and "cat mosquitto/config/mosquitto.conf". Take screenshots.

4. Test connection to Rahti cloud

tips:

Use image from exercise 1 to run an MQTT publisher on your own machine. Do not exec into a container!

Take screenshot that shows the input and the command prompt.

(Hint! Use MQTT_URL=mqtt.rahtiapp.fi and MQTT_PORT=443 in the docker run command.)

(Hint! If you get the "do handshake" error: your certs are mounted wrong, they are not in the filepath they need to be or actual filepath and the ones in mosquitto.conf do not match)

Check that messages are received by subscribing to your rahti mqtt server (use mosquitto_sub)

Deliverables

1. Yaml file or several yaml files that define at least a Deployment, a Service, an Route and the needed the ConfigMap objects.
2. Short text report with screenshots from the inputs and output of all of the commands in tasks 2 & 3. Remember to add screenshots. You will fail without them.

Note: You don't need to modify the previous toyota-data-feeder image in any way!

Get MQTT certificates files from bottom of the exercises page.

Reuse mosquitto.conf from exercise 2. You can use the variables in the mosquitto.conf file as example when creating mosquitto.conf ConfigMap object, but you need to alter the filepaths of cert files to match the one in your mosquitto broker deployment definition. Note that mosquitto needs filepaths to all three files.

My Text Report

INFO:

My name: Xinyuan Ma

CSC Rahti account: student297

STEPS:

1. Create a [yaml](#) file
2. Make sure Log in and create a project on Rahti.
3. Screenshot of "oc create"

```
(base) ericma@EricMacBook-Pro toyota-data-feeder % oc create -f mosquitto.yaml
configmap/mosquitto-config created
configmap/mosquitto-certs created
deployment.extensions/mosquito-deployment created
service/mosquitto-service created
route.route.openshift.io/mosquitto created
```

4. Screenshot of "oc get all"

```
(base) ericma@EricMacBook-Pro toyota-data-feeder % oc get all
NAME READY STATUS RESTARTS AGE
pod/mosquito-deployment-574d588fdb-z9gdl 1/1 Running 0 2m

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/mosquitto-service ClusterIP 172.30.101.49 <none> 8883/TCP,1883/TCP 2m

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
deployment.apps/mosquito-deployment 1 1 1 1 2m

NAME DESIRED CURRENT READY AGE
replicaset.apps/mosquito-deployment-574d588fdb 1 1 1 2m

NAME DOCKER REPO TAGS UPDATED
imagestream.image.openshift.io/toyotafeeder docker-registry.default.svc:5000/xinyuan-ma/toyotafeeder 2.0,1.0 5 days ago

NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD
route.route.openshift.io/mosquitto mqtt.student297.rahtiapp.fi mosquitto-service 8883 passthrough None
```

5. Screenshot of "oc get pod"

```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc get pod
NAME                                READY   STATUS    RESTARTS   AGE
mosquito-deployment-574d588fdb-z9gdl 1/1     Running   0           3m
```

6. Screenshot of "oc exec"

```
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc exec mosquito-deployment-574d588fdb-z9gdl -i -t -- sh
/ $
```

7. Screenshot of "ls mosquitto/config/certs" and "cat mosquitto/config/mosquitto.conf"

```
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc exec mosquito-deployment-574d588fdb-z9gdl -i -t -- sh
/ $ ls mosquitto/config/certs
ca.crt  server.crt  server.key
/ $ cat mosquitto/config/mosquitto.conf
allow_anonymous true
listener 1883
listener 8883

certfile /mosquitto/config/certs/server.crt
cafile /mosquitto/config/certs/ca.crt
keyfile /mosquitto/config/certs/server.key
/ $
```