Exercise 4 - Kubernetes Advanced

Assigments

1. Create two new yaml files with following definitions: First yaml file: Object definition for node-RED Service for node-RED Route for node-RED

Second yaml file: PersistentVolumeClaim

Object definition for node-RED should include the following: Use deployment or deploymentConfig Use image: nodered/node-red Mount directory /data with volumeMount Create a volume for the mounted directory Declare a persistentVolumeClaim for the volume and give it a name

Service for node-RED should include the following Use port: 1880 Type: LoadBalancer

Route for node-RED Use spec.host: nodered..rahtiapp.fi

PersistentVolumeClaim (Separate yaml file!) Use spec. accessModes: - ReadWriteOnce

2. Use “oc create” command to create the objects defined in your yaml files in Rahti

Inspect your deployment with "oc get all" and “oc status” Use "oc get pod" to get node-RED pod name, should be something like nodered-75883903-hwk9y Use "oc exec -i -t -- sh" (e.g. "oc exec nodered-75883903-hwk9y -i -t -- sh") to get access inside the container. Once inside use “cd /data” to get into /data folder. Check with "ls" Use “echo hello > hello.txt” to create hello.txt file with text hello. Check that file was created with “cat hello.txt” Exit pod with “exit” command. Destroy pod with “oc delete pod nodered-pod”. Use “oc replace –force -f “.yaml””. Use “oc get pods” and wait until pod is ready. Repeat steps 3 and 4. Pod names should be different! Do not create a new hello.txt file. Check that persistent volume works with "cat hello.txt" 3. Read https://nodered.org/docs/tutorials/

4. Edit flow.json file. Change the text inside the comparison operators <>. e.g. "topic": "/#", change into "topic": "student123/#",

5. With your browser go to nodered..rahtiapp.fi. Add node-red-dashboard, by clicking the menu (upper right corner) and selecting "manage palette". Select install tab and type "node-red-dashboard". Install it. Import flow.json by copypasting it into import prompt. Select menu> import. Upload certificates. Click Deploy.

6. Use image from exercise 1 to run an MQTT publisher on your own machine. Use MQTT_URL=mqtt..rahtiapp.fi and MQTT_PORT=443 in the docker run command.

7. Go to nodered..rahtiapp.fi/ui. Check that speedometer is moving. Take a screenshot.

USE ONLY CHARACTERS a-z AND minus sign (-) IN SERVICE NAMES

Deliverables

1. Two yaml files. First defining at least a Deployment, a Service, an Route and second yaml with PersistentVolumeClaim Object definition-

2. Short text report with screenshots from the inputs and output of all of the commands in tasks 2 including commands in subtasks.

3. Screenshots of working node-red speedometer and modified flow.json file

Possibly useful commands:

oc –help

oc create -f .yaml

oc status

oc replace –force -f .yaml

oc delete –all service (OR route OR configmap OR deployment OR etc..)

oc get pods (OR routes OR deployments OR etc..)

My Text Report

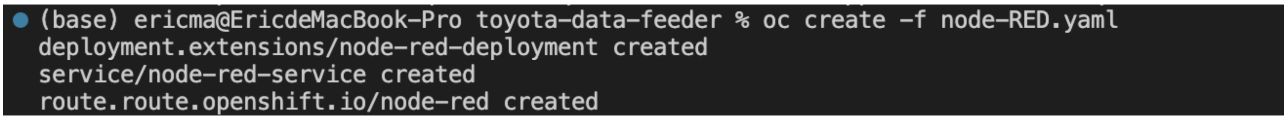
INFO:

My name: Xinyuan Ma
CSC Rahti account: student297

STEPS:

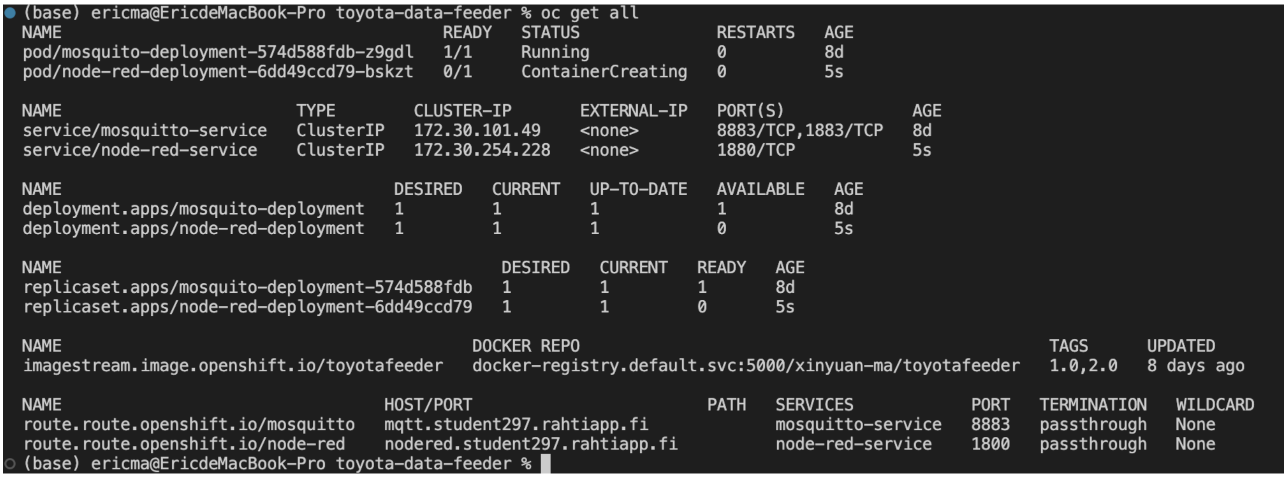
1. Create two new yaml files with following definitions node-RED and PersistentVolumeClaim

2. Use “oc create” command to create the objects defined in your yaml files in Rahti

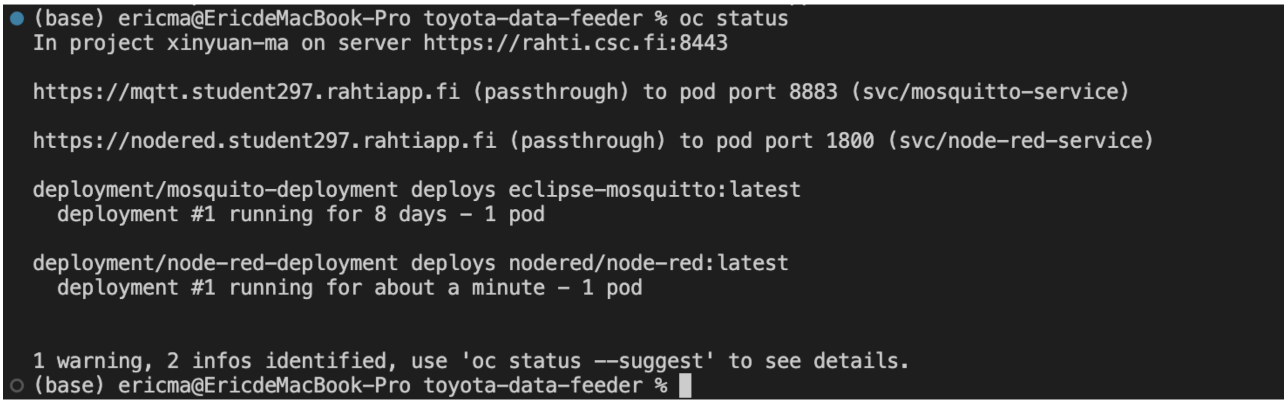


```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc create -f node-RED.yaml
  deployment.extensions/node-red-deployment created
  service/node-red-service created
  route.route.openshift.io/node-red created
```

3. Inspect your deployment with "oc get all" and “oc status”

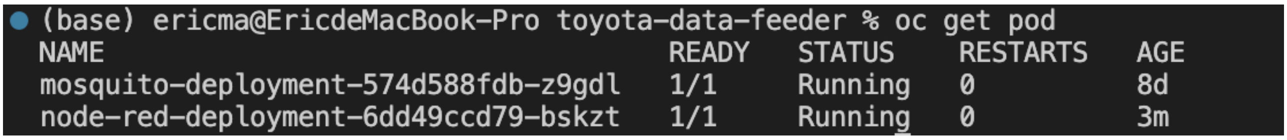


```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc get all
  NAME                                        READY   STATUS              RESTARTS   AGE
  pod/mosquito-deployment-574d588fdb-z9gdl   1/1     Running             0          8d
  pod/node-red-deployment-6dd49ccd79-bskzt   0/1     ContainerCreating   0          5s

  NAME                         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE
  service/mosquitto-service   ClusterIP   172.30.101.49    <none>        8883/TCP,1883/TCP   8d
  service/node-red-service    ClusterIP   172.30.254.228   <none>        1880/TCP            5s

  NAME                                   DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
  deployment.apps/mosquito-deployment   1         1         1            1           8d
  deployment.apps/node-red-deployment   1         1         1            0           5s

  NAME                                              DESIRED   CURRENT   READY   AGE
  replicaset.apps/mosquito-deployment-574d588fdb   1         1         1       8d
  replicaset.apps/node-red-deployment-6dd49ccd79   1         1         0       5s

  NAME                                          DOCKER REPO                                                TAGS      UPDATED
  imagestream.image.openshift.io/toyotafeeder   docker-registry.default.svc:5000/xinyuan-ma/toyotafeeder   1.0,2.0   8 days ago

  NAME                                 HOST/PORT                          PATH   SERVICES            PORT   TERMINATION   WILDCARD
  route.route.openshift.io/mosquitto   mqtt.student297.rahtiapp.fi               mosquitto-service   8883   passthrough   None
  route.route.openshift.io/node-red    nodered.student297.rahtiapp.fi            node-red-service    1800   passthrough   None
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % 
```



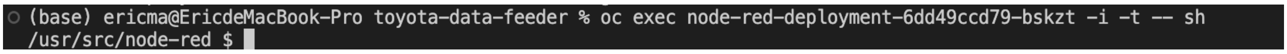
```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc status
  In project xinyuan-ma on server https://rahti.csc.fi:8443

  https://mqtt.student297.rahtiapp.fi (passthrough) to pod port 8883 (svc/mosquitto-service)

  https://nodered.student297.rahtiapp.fi (passthrough) to pod port 1800 (svc/node-red-service)

  deployment/mosquito-deployment deploys eclipse-mosquitto:latest
    deployment #1 running for 8 days - 1 pod

  deployment/node-red-deployment deploys nodered/node-red:latest
    deployment #1 running for about a minute - 1 pod


  1 warning, 2 infos identified, use 'oc status --suggest' to see details.
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % 
```

4. Use "oc get pod" to get node-RED pod name, should be something like nodered-75883903-hwk9y



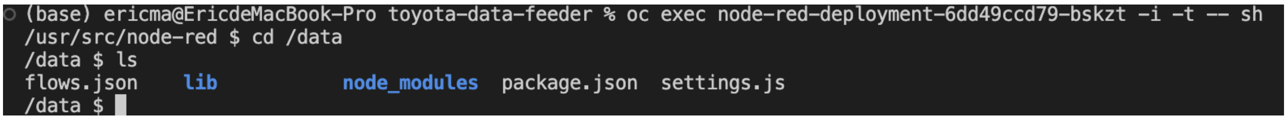
```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc get pod
  NAME                                 READY   STATUS    RESTARTS   AGE
  mosquito-deployment-574d588fdb-z9gdl   1/1     Running   0          8d
  node-red-deployment-6dd49ccd79-bskzt   1/1     Running   0          3m
```

5. Use "oc exec -i -t -- sh" (e.g. "oc exec nodered-75883903-hwk9y -i -t -- sh") to get access inside the container.



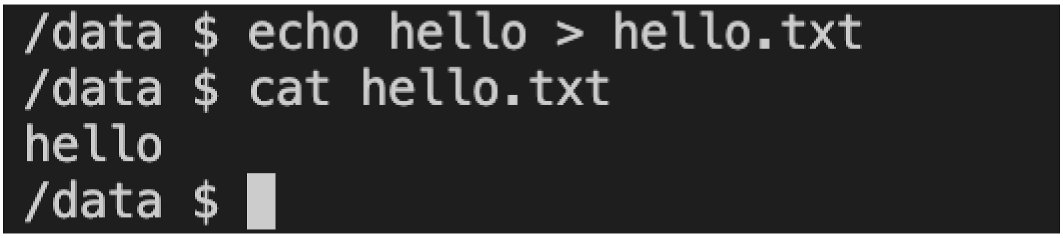
```
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc exec node-red-deployment-6dd49ccd79-bskzt -i -t -- sh
  /usr/src/node-red $ 
```

6. Once inside use “cd /data” to get into /data folder. Check with "ls"



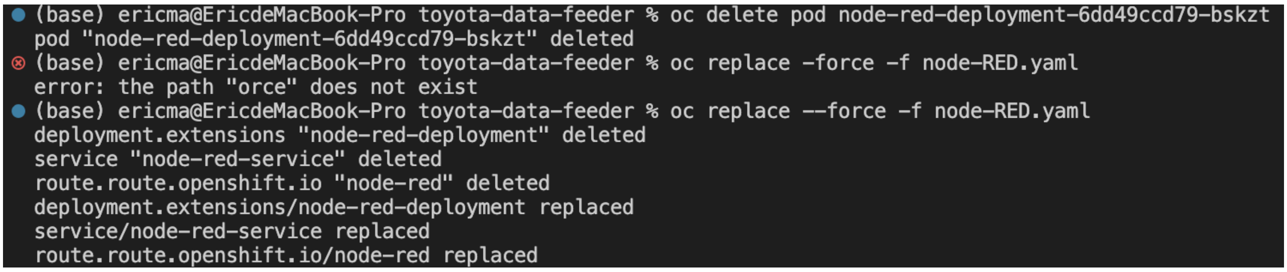
```
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc exec node-red-deployment-6dd49ccd79-bskzt -i -t -- sh
  /usr/src/node-red $ cd /data
  /data $ ls
  flows.json    lib           node_modules  package.json  settings.js
  /data $ 
```

7. Use “echo hello > hello.txt” to create hello.txt file with text hello. Check that file was created with “cat hello.txt”

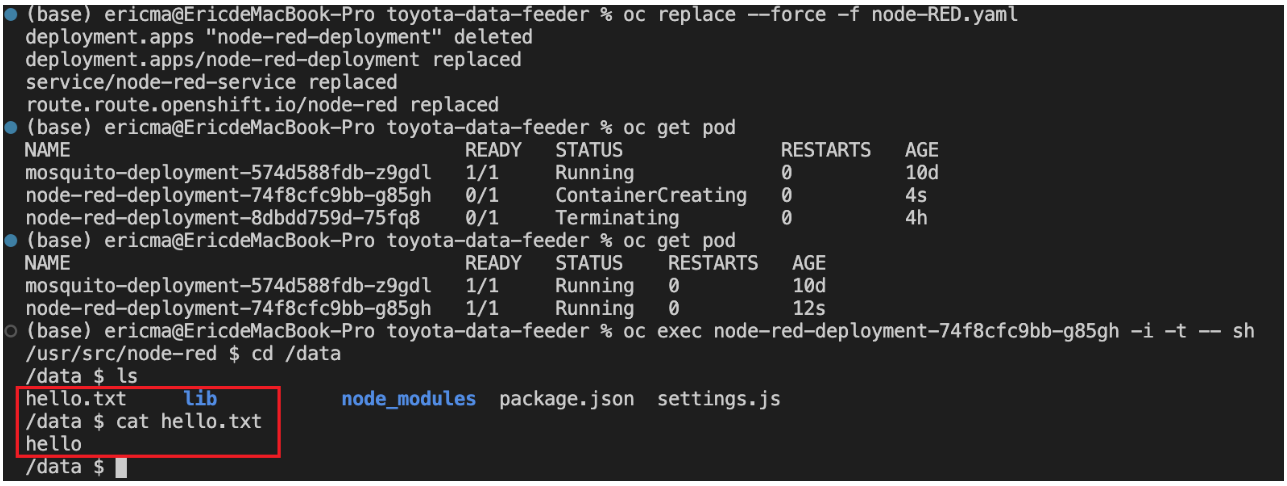


```
/data $ echo hello > hello.txt
/data $ cat hello.txt
hello
/data $ 
```

8. Exit pod with “exit” command. Destroy pod with “oc delete pod nodered-pod”. Use “oc replace –force -f “.yaml””. Use “oc get pods” and wait until pod is ready. Repeat steps 3 and 4. Pod names should be different! Do not create a new hello.txt file. Check that persistent volume works with "cat hello.txt"

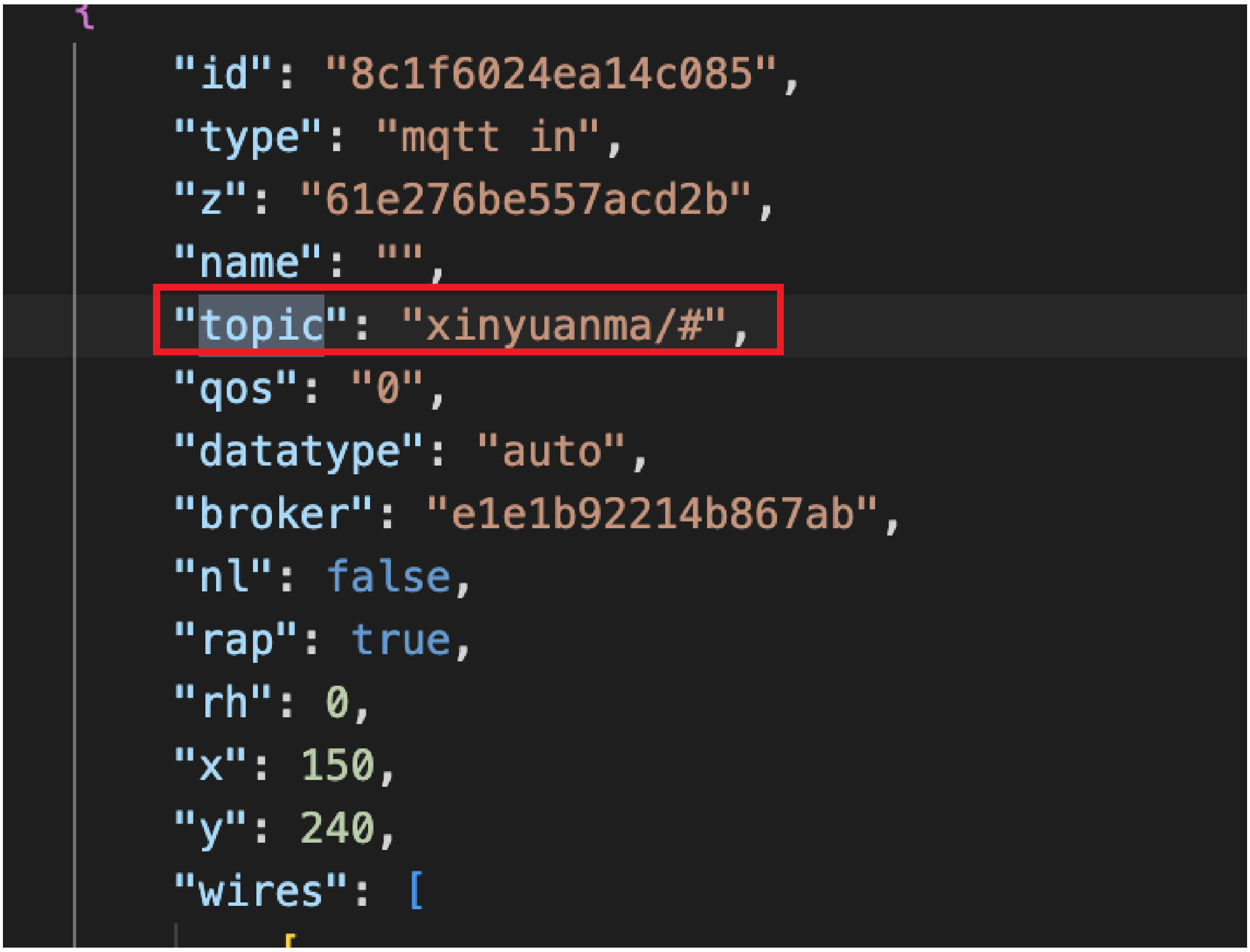


```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc delete pod node-red-deployment-6dd49ccd79-bskzt
  pod "node-red-deployment-6dd49ccd79-bskzt" deleted
⊗ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc replace -force -f node-RED.yaml
  error: the path "orce" does not exist
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc replace --force -f node-RED.yaml
  deployment.extensions "node-red-deployment" deleted
  service "node-red-service" deleted
  route.route.openshift.io "node-red" deleted
  deployment.extensions/node-red-deployment replaced
  service/node-red-service replaced
  route.route.openshift.io/node-red replaced
```



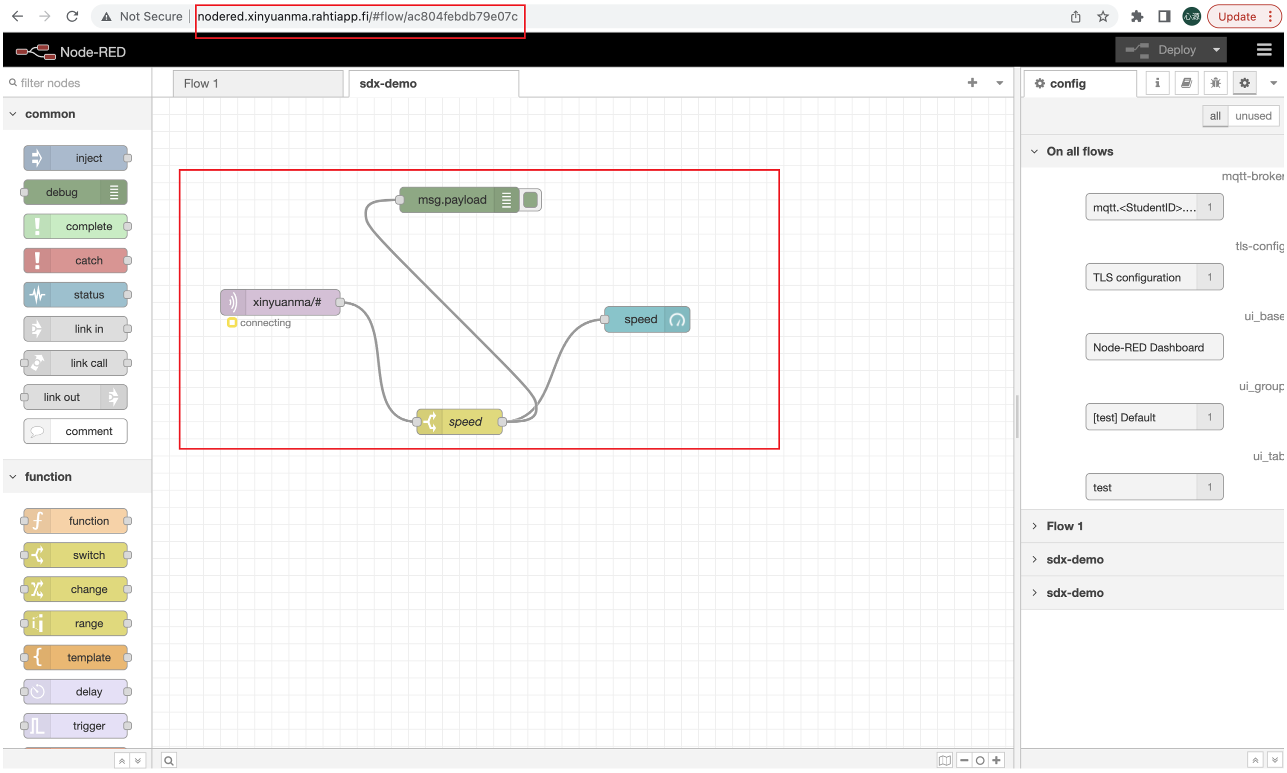
```
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc replace --force -f node-RED.yaml
  deployment.apps "node-red-deployment" deleted
  deployment.apps/node-red-deployment replaced
  service/node-red-service replaced
  route.route.openshift.io/node-red replaced
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc get pod
  NAME                                    READY   STATUS              RESTARTS   AGE
  mosquito-deployment-574d588fdb-z9gdl    1/1     Running             0          10d
  node-red-deployment-74f8cfc9bb-g85gh    0/1     ContainerCreating   0          4s
  node-red-deployment-8dbdd759d-75fq8     0/1     Terminating         0          4h
● (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc get pod
  NAME                                    READY   STATUS    RESTARTS   AGE
  mosquito-deployment-574d588fdb-z9gdl    1/1     Running   0          10d
  node-red-deployment-74f8cfc9bb-g85gh    1/1     Running   0          12s
○ (base) ericma@EricdeMacBook-Pro toyota-data-feeder % oc exec node-red-deployment-74f8cfc9bb-g85gh -i -t -- sh
  /usr/src/node-red $ cd /data
  /data $ ls
  hello.txt     lib           node_modules  package.json  settings.js
  /data $ cat hello.txt
  hello
  /data $ 
```

9. modefy json.file

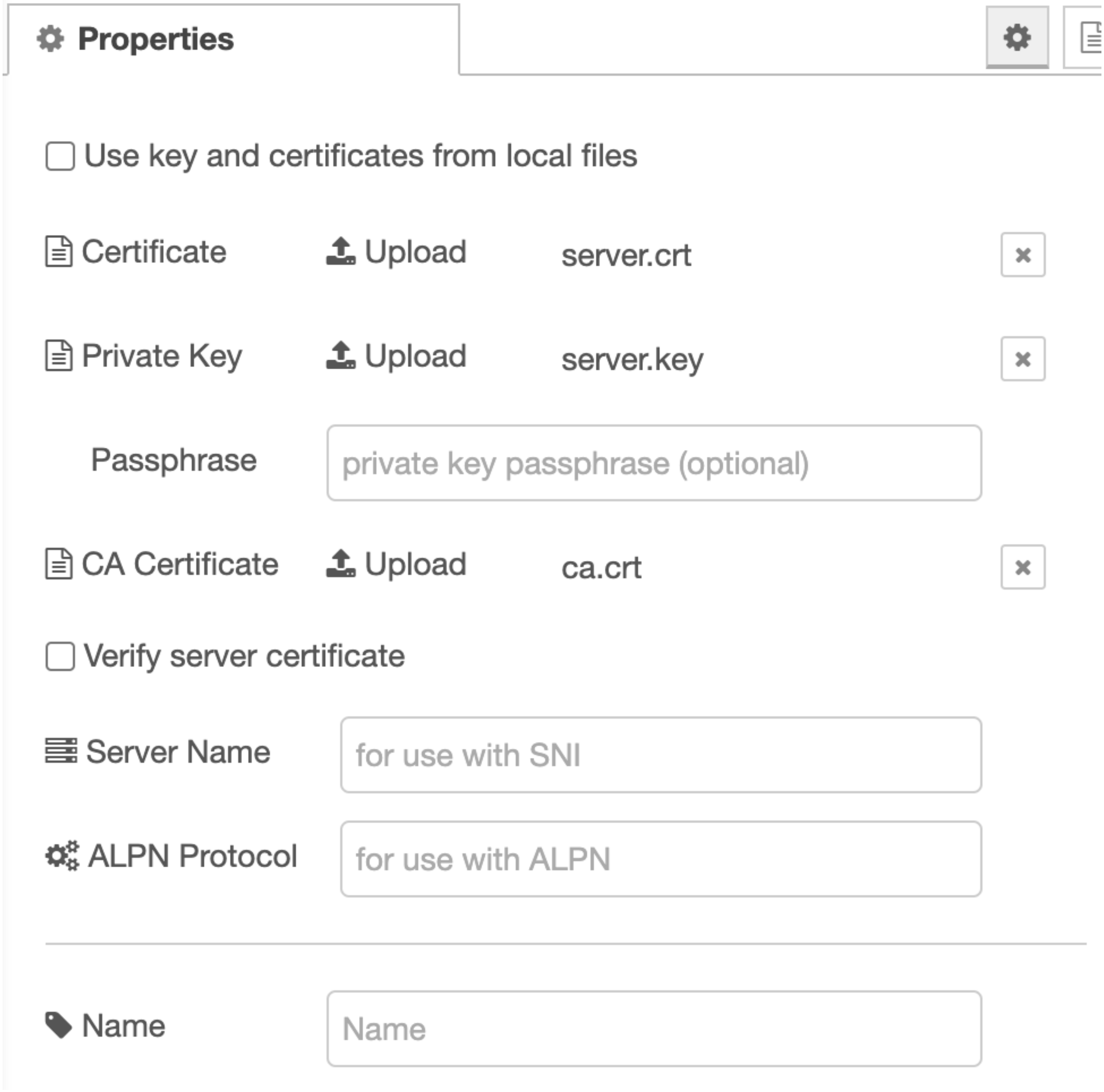


```
{
        "id": "8c1f6024ea14c085",
        "type": "mqtt in",
        "z": "61e276be557acd2b",
        "name": "",
        "topic": "xinyuanma/#",
        "qos": "0",
        "datatype": "auto",
        "broker": "e1e1b92214b867ab",
        "nl": false,
        "rap": true,
        "rh": 0,
        "x": 150,
        "y": 240,
        "wires": [
            [
```

10. nodered web



11. update certs



☐ Use key and certificates from local files

Certificate: server.crt

Private Key: server.key

Passphrase: private key passphrase (optional)

CA Certificate: ca.crt

☐ Verify server certificate

Server Name: for use with SNI

ALPN Protocol: for use with ALPN

Name: Name