Day 4

References - You may find some interesting openshift examples here

```
https://github.com/redhat-developer-demos
```

Installing Helm package manager in Linux

```
curl -fsSL -o get_helm.sh
https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
```

Helm Overview

- a package manager for Kubernetes and Openshift orchestration platforms
- using helm we can package our application as Charts
- our end-users will be able to use the charts we packaged to install into openshift
- helm tool is aware of Kubernetes and Openshift

Lab - Creating a custom helm package for wordpress and mariadb multi-pod application

```
cd ~/openshift-may-2024
git pull
cd Day4/helm
helm create wordpress-chart
rm -rf wordpress-char/templates/*
#cp yaml-files/* wordpress-chart/templates
ls -l
helm package wordpress-chart
helm install wordpress wordpress-chart-0.1.0.tgz

helm list
```

You may verify if the wordpress and mysql is deployed properly.

Lab - Deploying redis database with persistent volume

```
oc new-app --name=redis -e REDIS_PASSWORD=pass@123 bitnami/redis:latest --
dry-run -o yaml
oc new-app --name=redis -e REDIS_PASSWORD=pass@123 bitnami/redis:latest --
```

```
dry-run -o yaml > deploy-redis.yml

cd ~/openshift-may-2024
git pull
cd Day4/redis
oc apply -f redis-pv.yml
oc apply -f redis-pvc.yml
oc apply -f deploy-redis.yml
oc apply -f redis-route.yml

oc rsh deploy/redis

redis-cli -h localhost -p 6379 -a pass@123
set msg 'Hello Redis'
get msg
```

Expected output

```
[jegan@tektutor.org redis]$ oc new-app --name=redis -e
REDIS_PASSWORD=pass@123 bitnami/redis:latest --dry-run -o yaml
apiVersion: v1
items:
- apiVersion: image.openshift.io/v1
  kind: ImageStream
  metadata:
    annotations:
      openshift.io/generated-by: OpenShiftNewApp
    creationTimestamp: null
    labels:
      app: redis
      app.kubernetes.io/component: redis
      app.kubernetes.io/instance: redis
    name: redis
  spec:
    lookupPolicy:
      local: false
    tags:
    - annotations:
        openshift.io/imported-from: bitnami/redis:latest
      from:
        kind: DockerImage
        name: bitnami/redis:latest
      generation: null
      importPolicy:
        importMode: Legacy
      name: latest
      referencePolicy:
        type: ""
  status:
    dockerImageRepository: ""
- apiVersion: apps/v1
  kind: Deployment
```

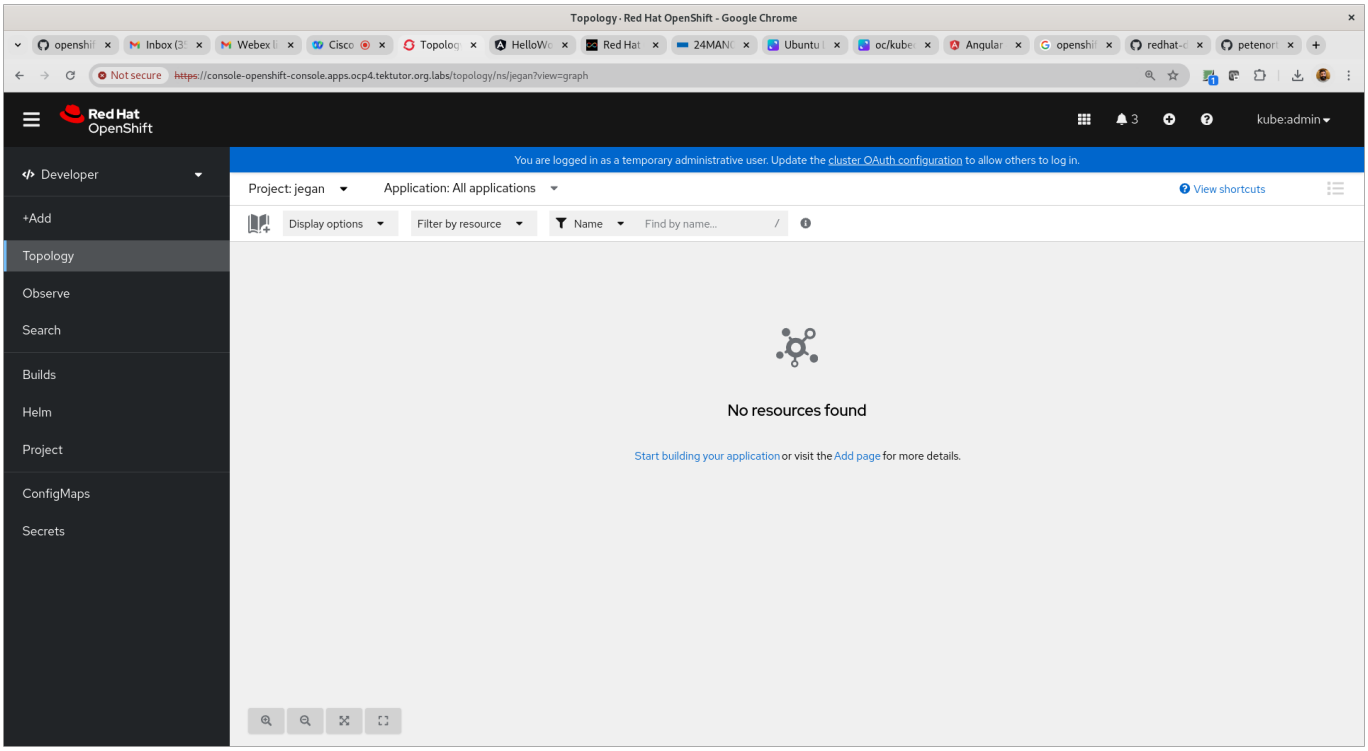
```
    metadata:
      annotations:
        image.openshift.io/triggers: '[{"from":
  {"kind":"ImageStreamTag","name":"redis:latest"},"fieldPath":"spec.template.
  spec.containers[?(@.name==\"redis\")].image"}]'
        openshift.io/generated-by: OpenShiftNewApp
      creationTimestamp: null
      labels:
        app: redis
        app.kubernetes.io/component: redis
        app.kubernetes.io/instance: redis
      name: redis
    spec:
      replicas: 1
      selector:
        matchLabels:
          deployment: redis
      strategy: {}
      template:
        metadata:
          annotations:
            openshift.io/generated-by: OpenShiftNewApp
          creationTimestamp: null
          labels:
            deployment: redis
        spec:
          containers:
          - env:
            - name: REDIS_PASSWORD
              value: pass@123
            image: ' '
            name: redis
            ports:
            - containerPort: 6379
              protocol: TCP
            resources: {}
    status: {}
  - apiVersion: v1
    kind: Service
    metadata:
      annotations:
        openshift.io/generated-by: OpenShiftNewApp
      creationTimestamp: null
      labels:
        app: redis
        app.kubernetes.io/component: redis
        app.kubernetes.io/instance: redis
      name: redis
    spec:
      ports:
      - name: 6379-tcp
        port: 6379
        protocol: TCP
        targetPort: 6379
```

```
    selector:
      deployment: redis
  status:
    loadBalancer: {}
kind: List
metadata: {}

[jegan@tektutor.org redis]$ oc apply -f redis-pv.yml
persistentvolume/redis-pv-jegan created
[jegan@tektutor.org redis]$ oc apply -f redis-pvc.yml
persistentvolumeclaim/redis-pvc-jegan created
[jegan@tektutor.org redis]$ oc apply -f deploy-redis.yml
imagestream.image.openshift.io/redis created
deployment.apps/redis created
service/redis created

[jegan@tektutor.org redis]$ oc get svc
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
redis   ClusterIP   172.30.29.107           6379/TCP   20s
[jegan@tektutor.org redis]$ oc expose svc/redis --dry-run=client -o yaml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  creationTimestamp: null
  labels:
    app: redis
    app.kubernetes.io/component: redis
    app.kubernetes.io/instance: redis
  name: redis
spec:
  port:
    targetPort: 6379-tcp
  to:
    kind: ""
    name: redis
    weight: null
status: {}
[jegan@tektutor.org redis]$ oc expose svc/redis --dry-run=client -o yaml >
redis-route.yml
[jegan@tektutor.org redis]$ oc apply -f redis-route.yml
route.route.openshift.io/redis created

jegan@tektutor.org redis]$ oc rsh deploy/redis
$ ls
bin  bitnami  boot  dev  entrypoint.sh  etc  home  lib  lib64  media  mnt
opt     proc  root  run  run.sh  sbin  srv  sys  tmp  usr  var

$ redis-cli -h localhost -p 6379 -a
Unrecognized option or bad number of args for: '-a'
$ redis-cli -h localhost -p 6379 -a pass@123
Warning: Using a password with '-a' or '-u' option on the command line
interface may not be safe.
localhost:6379> set msg 'Hello Redis'
OK
```

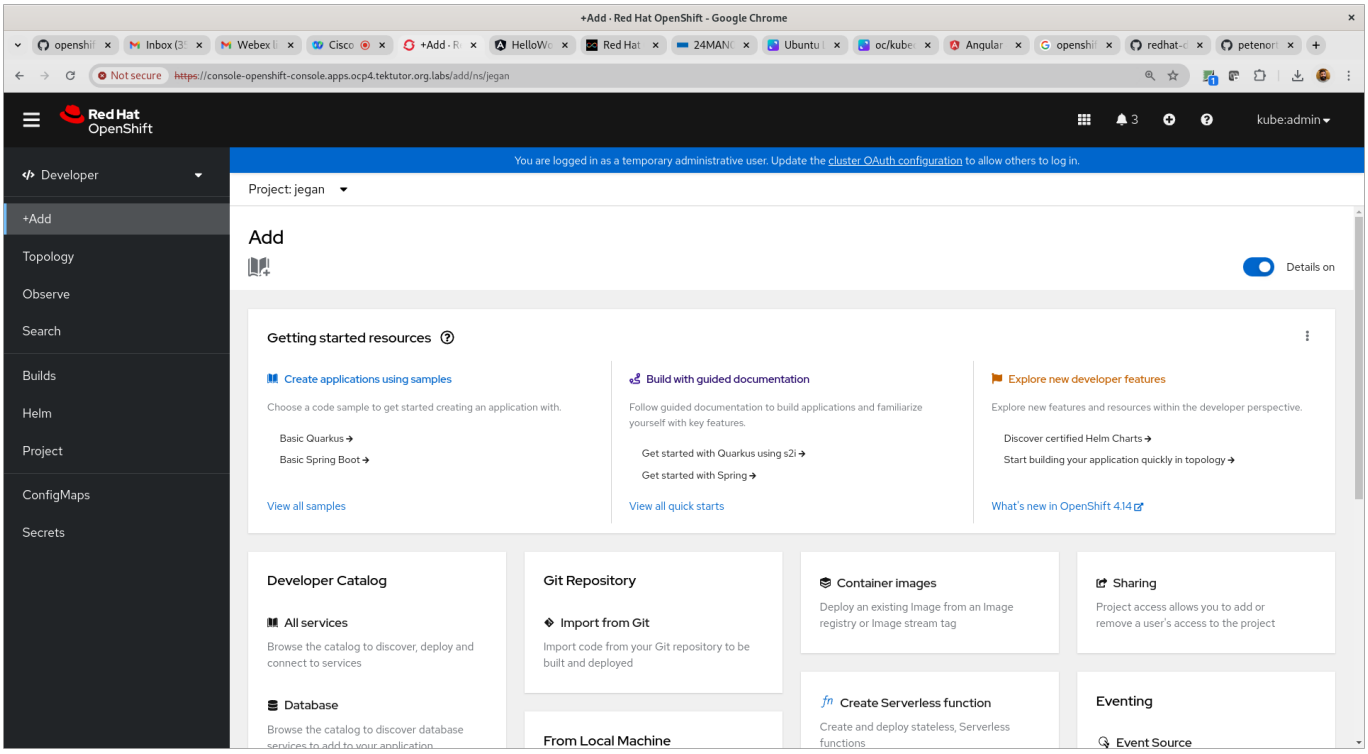
```
localhost:6379> get msg
"Hello Redis"
localhost:6379> exit
$ exit
```

Lab - Deploying an Angularjs application into openshift

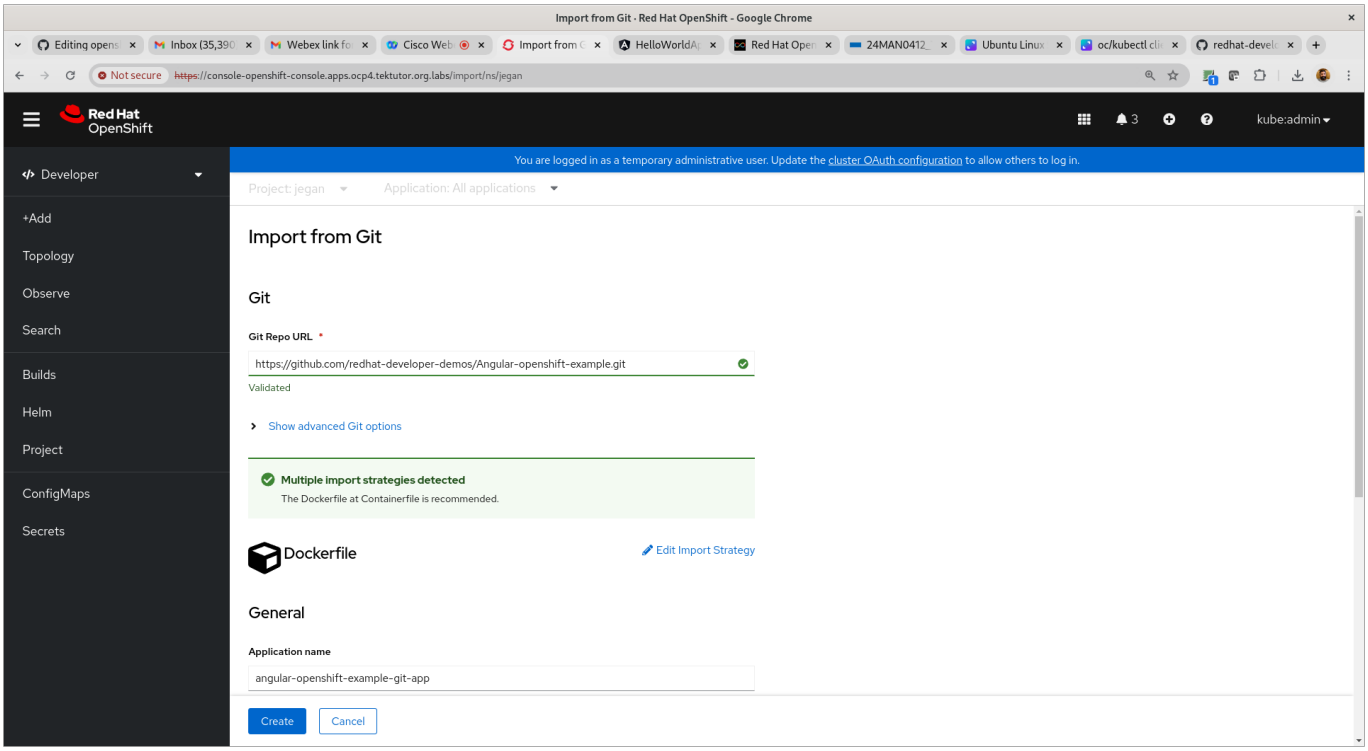
Navigate to Developer Context



Click on Add menu



Click on "Import From Git"

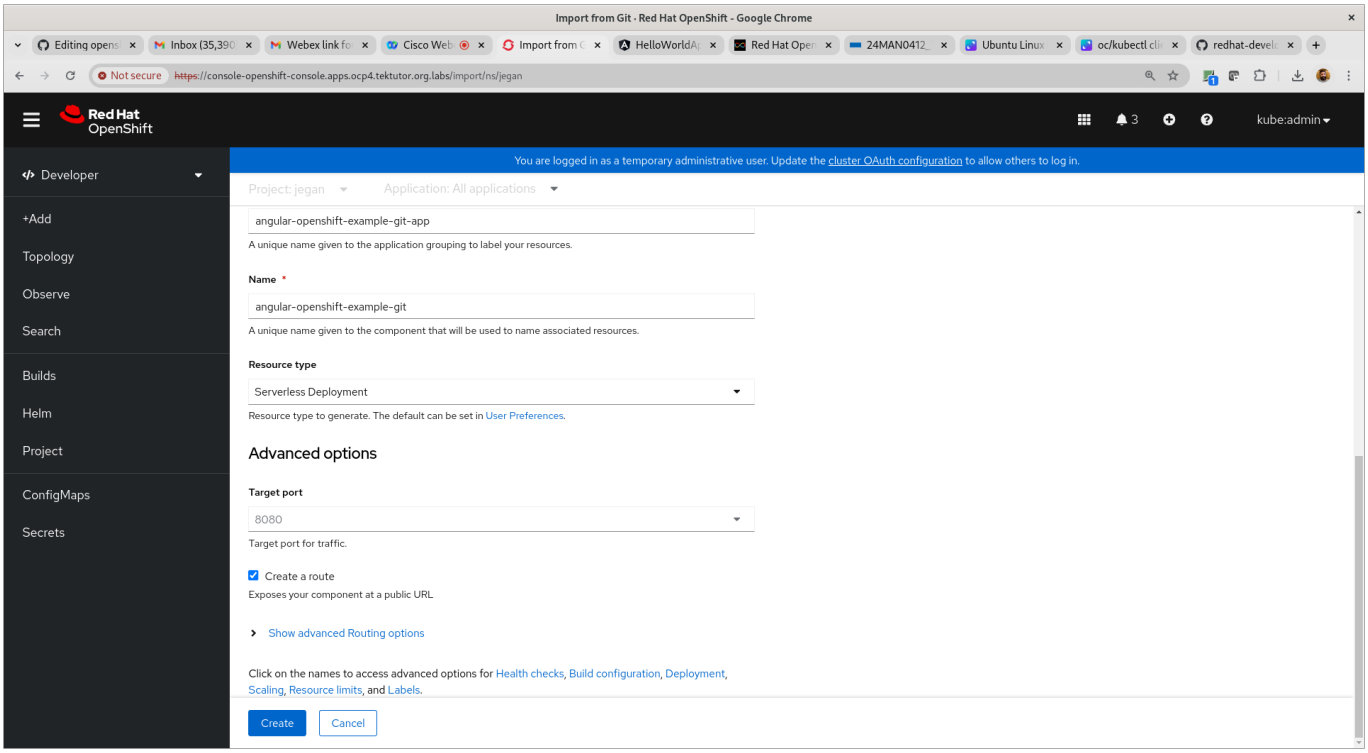


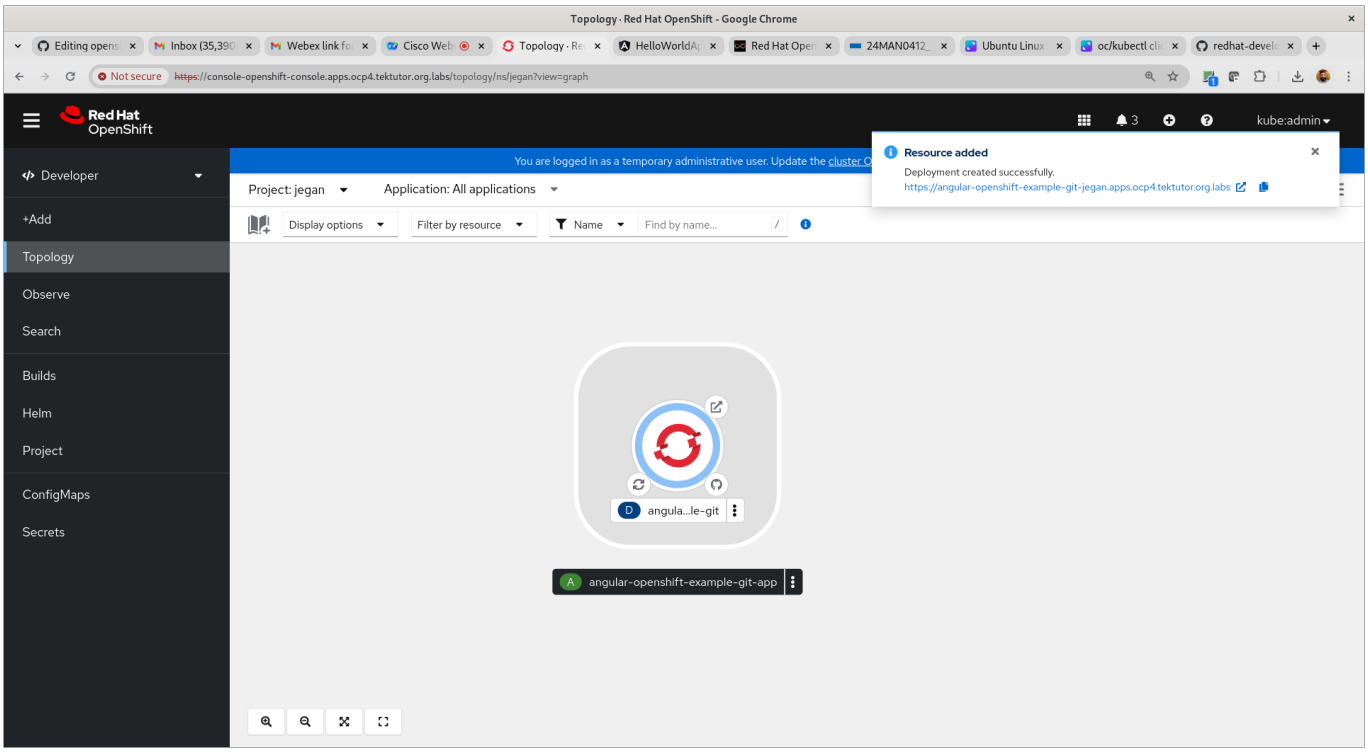
Type this as GitHub url

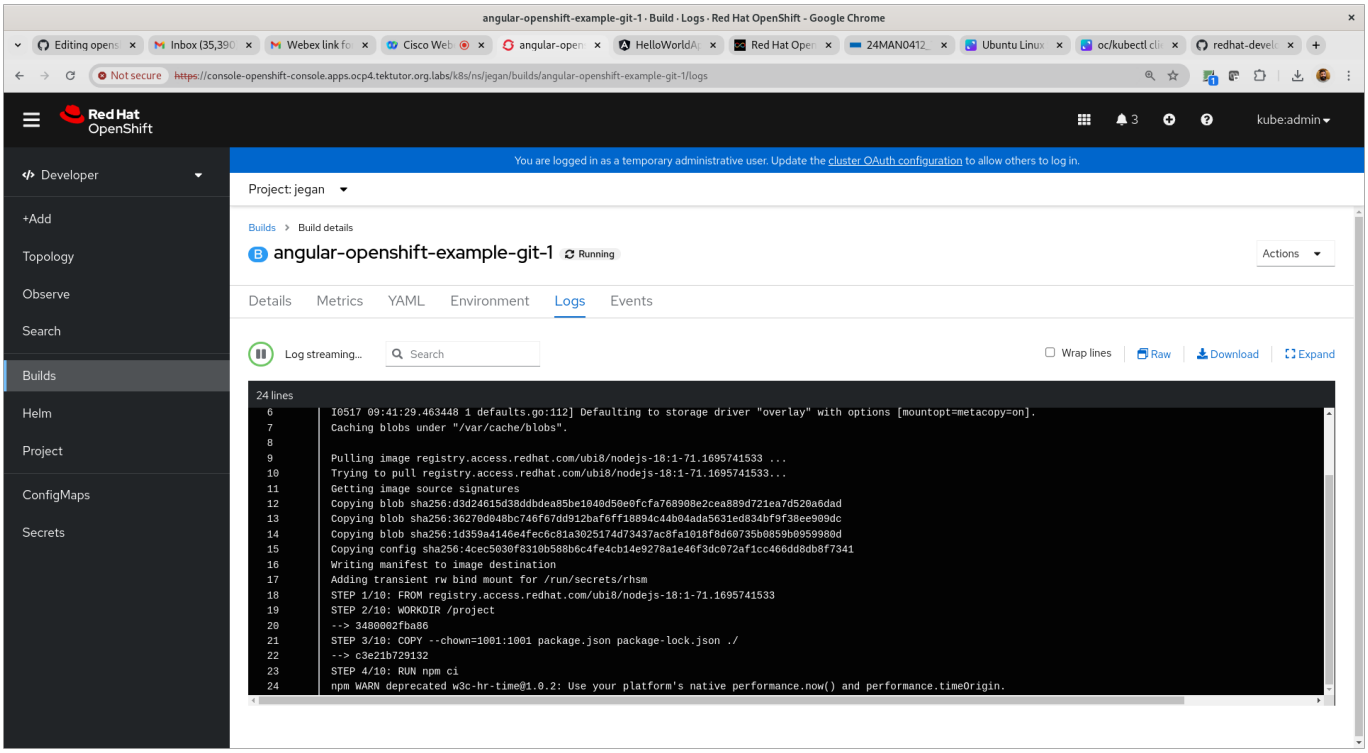
```
https://github.com/redhat-developer-demos/Angular-openshift-example.git
```

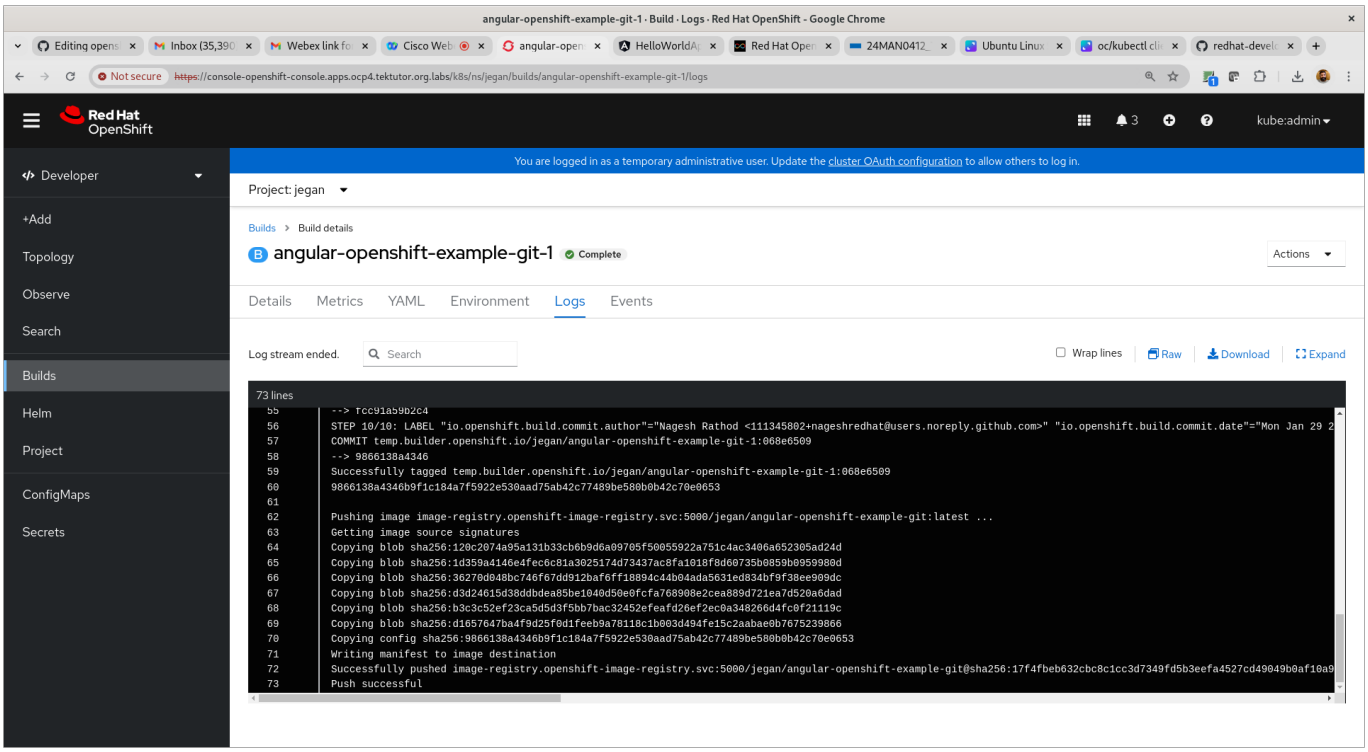
Under "Resource Type" make sure you selected "Deployment".

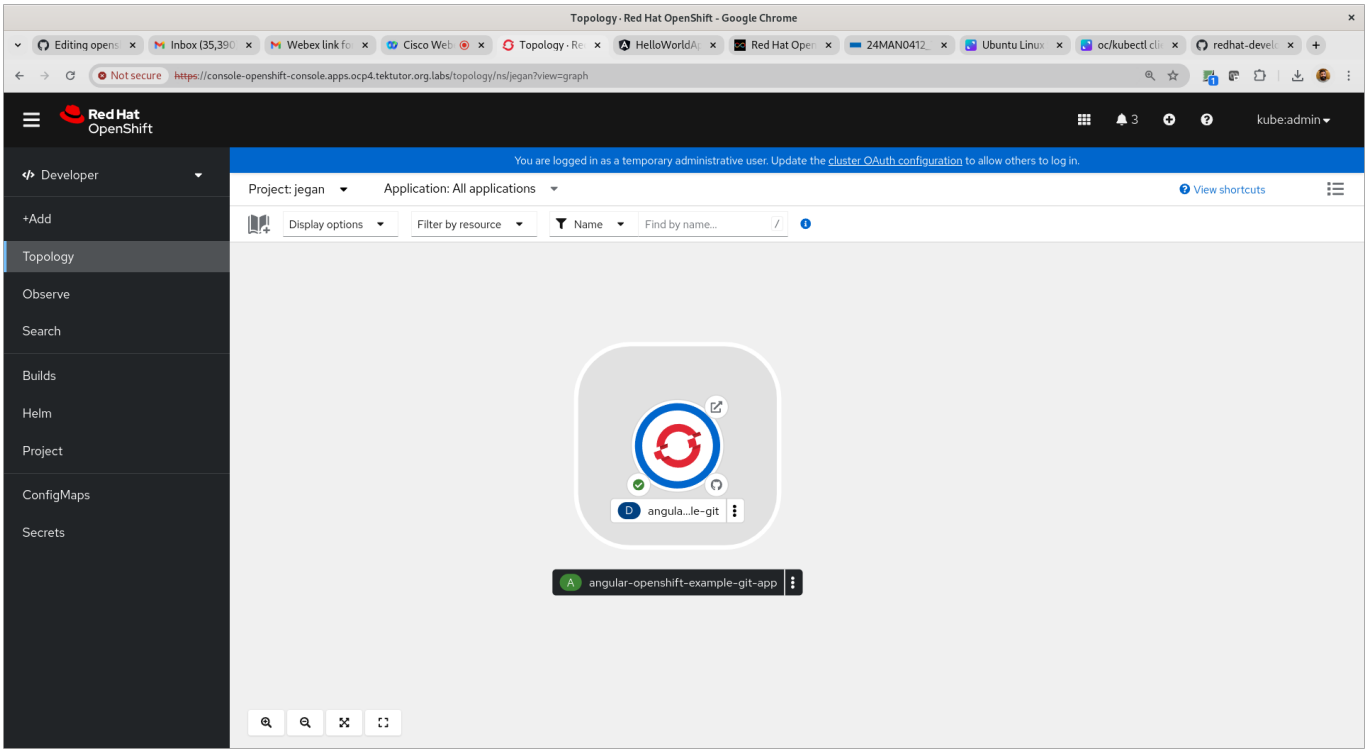
Other options, you may Accept all defaults and scroll down to click the "Create" button.

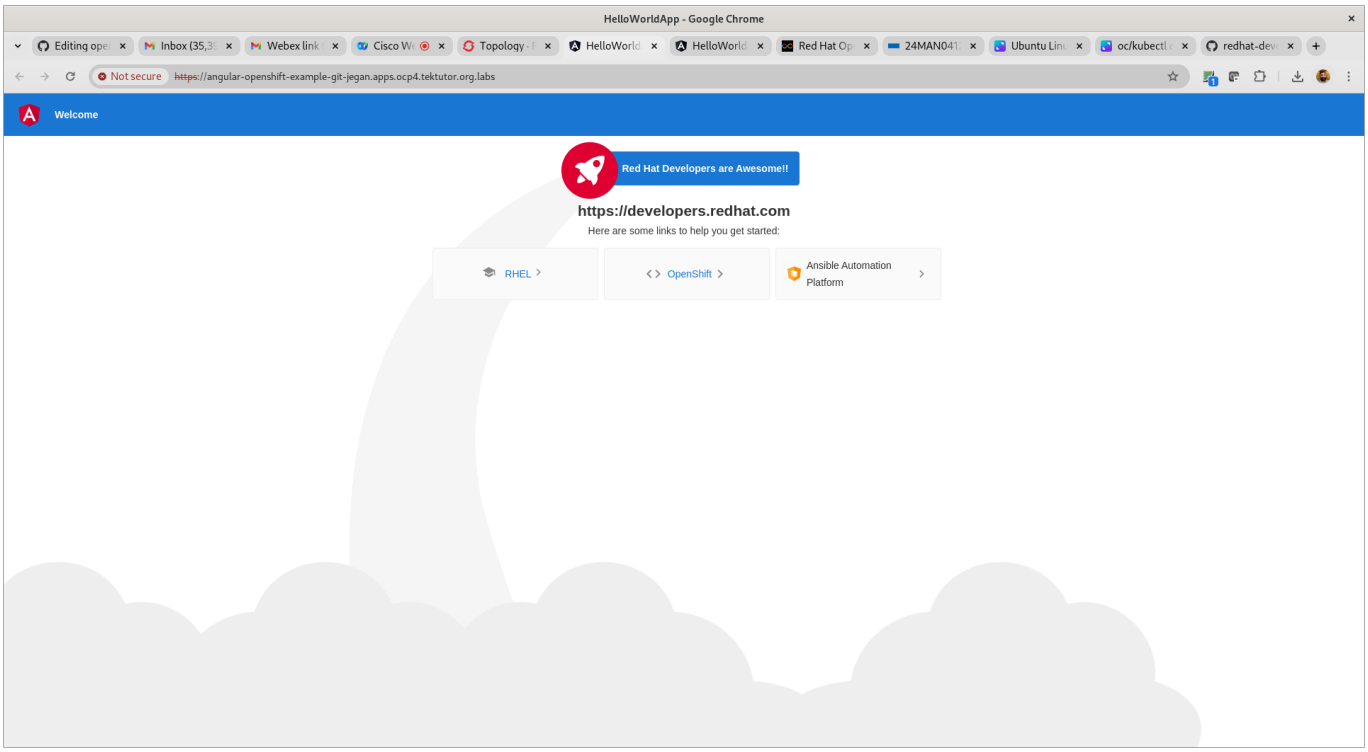



Topology - Red Hat OpenShift - Google Chrome
Red Hat OpenShift
kube:admin
Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Resource added
Deployment created successfully.
https://angular-openshift-example-git-jegan.apps.ocp4.tektutor.org.labs
Project: jegan
Application: All applications
Display options
Filter by resource
Name
Find by name...
angula...le-git
angular-openshift-example-git-app


angular-openshift-example-git-1 · Build · Logs · Red Hat OpenShift - Google Chrome
Red Hat OpenShift
kube:admin
Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Project: jegan
Builds
Build details
angular-openshift-example-git-1
Running
Actions
Details
Metrics
YAML
Environment
Logs
Events
Log streaming...
Search
Wrap lines
Raw
Download
Expand
24 lines
I0517 09:41:29.463448 1 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
Caching blobs under "/var/cache/blobs".
Pulling image registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533 ...
Trying to pull registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533...
Getting image source signatures
Copying blob sha256:d3d24615d38ddbdea85be1040d50e0fcfa768908e2cea889d721ea7d520a6dad
Copying blob sha256:36270d048bc746f67dd912baf6ff18894c44b04ada5631ed834bf9f38ee909dc
Copying blob sha256:1d359a4146e4fec6c81a3025174d73437ac8fa1018f8d60735b0859b0959980d
Copying config sha256:4cec5030f8310b588b6c4fe4cb14e9278a1e46f3dc072af1cc466dd8db8f7341
Writing manifest to image destination
Adding transient rw bind mount for /run/secrets/rhsm
STEP 1/10: FROM registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533
STEP 2/10: WORKDIR /project
--> 3480002fba86
STEP 3/10: COPY --chown=1001:1001 package.json package-lock.json ./
--> c3e21b729132
STEP 4/10: RUN npm ci
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.

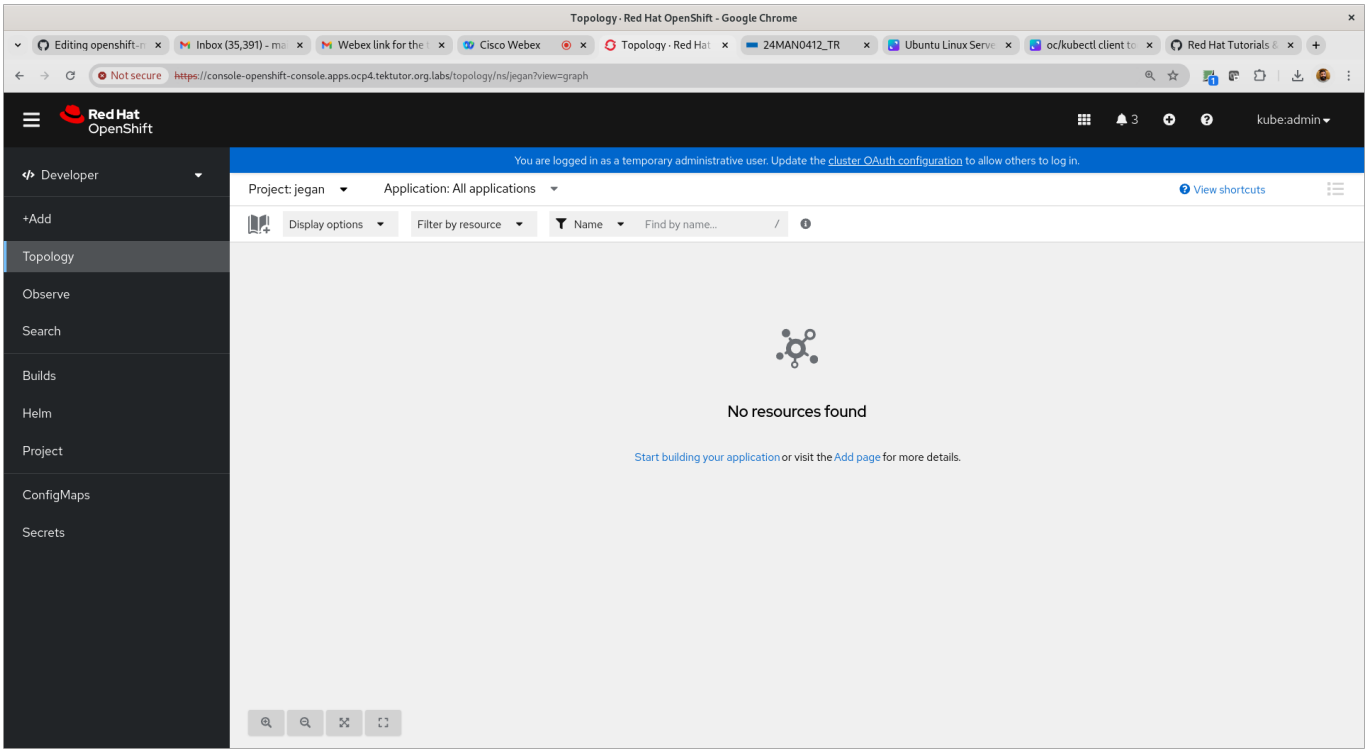

angular-openshift-example-git-1 · Build · Logs · Red Hat OpenShift - Google Chrome
Red Hat OpenShift
kube:admin
Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.
Project: jegan
Builds > Build details
angular-openshift-example-git-1 Complete
Actions
Details Metrics YAML Environment Logs Events
Log stream ended.
Search
Wrap lines Raw Download Expand
73 lines
--> 9866138a4346
Successfully tagged temp.builder.openshift.io/jegan/angular-openshift-example-git-1:068e6509
Pushing image image-registry.openshift-image-registry.svc:5000/jegan/angular-openshift-example-git:latest ...
Getting image source signatures
Copying blob sha256:120c2074a95a131b33cb6b9d6a09705f50055922a751c4ac3406a652305ad24d
Copying blob sha256:1d359a4146e4fec6c81a3025174d73437ac8fa1018f8d60735b0859b0959980d
Copying blob sha256:36270d048bc746f67dd912baf6ff1B894c44b04ada5631ed834bf9f38ee909dc
Copying blob sha256:d3d24615d38ddbdea85be1040d50e0fcfa768908e2cea889d721ea7d520a6dad
Copying blob sha256:b3c3c52ef23ca5d5d3f5bb7bac32452efeafd26ef2ec0a348266d4fc0f21119c
Copying blob sha256:d1657647ba4f9d25f0d1feeb9a78118c1b003d494fe15c2aabae0b7675239866
Copying config sha256:9866138a4346b9f1c184a7f5922e530aad75ab42c77489be580b0b42c70e0653
Writing manifest to image destination
Successfully pushed image-registry.openshift-image-registry.svc:5000/jegan/angular-openshift-example-git@sha256:17f4fbeb632cbc8c1cc3d7349fd5b3eefa4527cd49049b0af10a9
Push successful


Topology · Red Hat OpenShift - Google Chrome
Red Hat OpenShift
kube:admin
Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.
Project: jegan
Application: All applications
View shortcuts
Display options
Filter by resource
Name
Find by name...
angula...le-git
angular-openshift-example-git-app

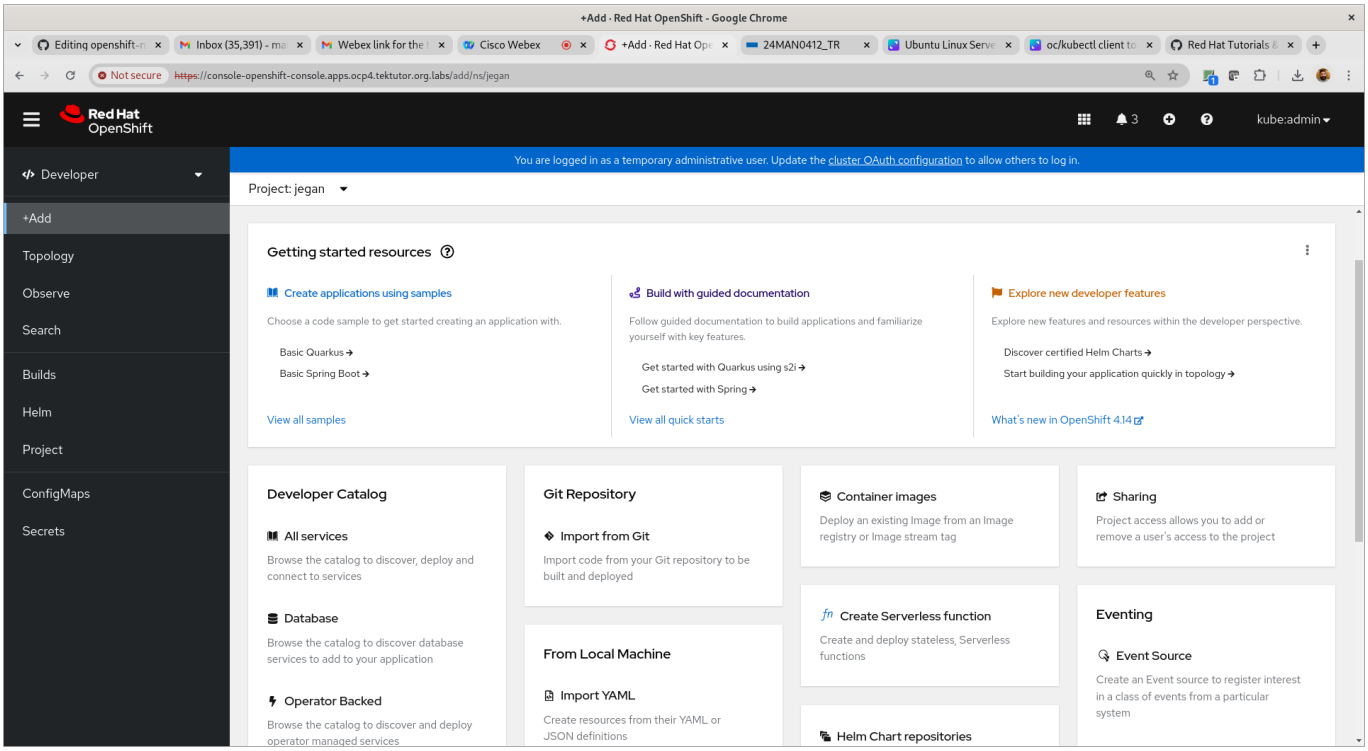


Lab - Deploying a sample reactjs application into openshift

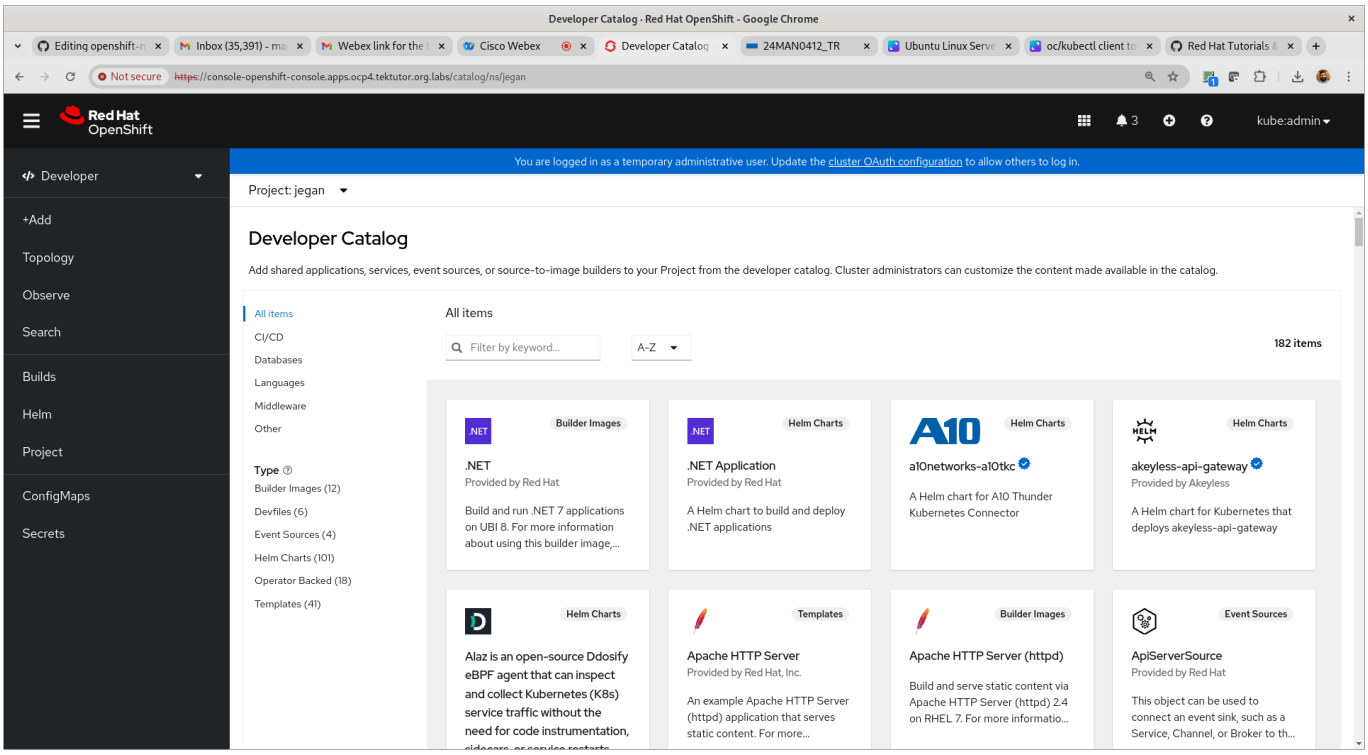
First navigate to Developer context.



Click on "Add" menu

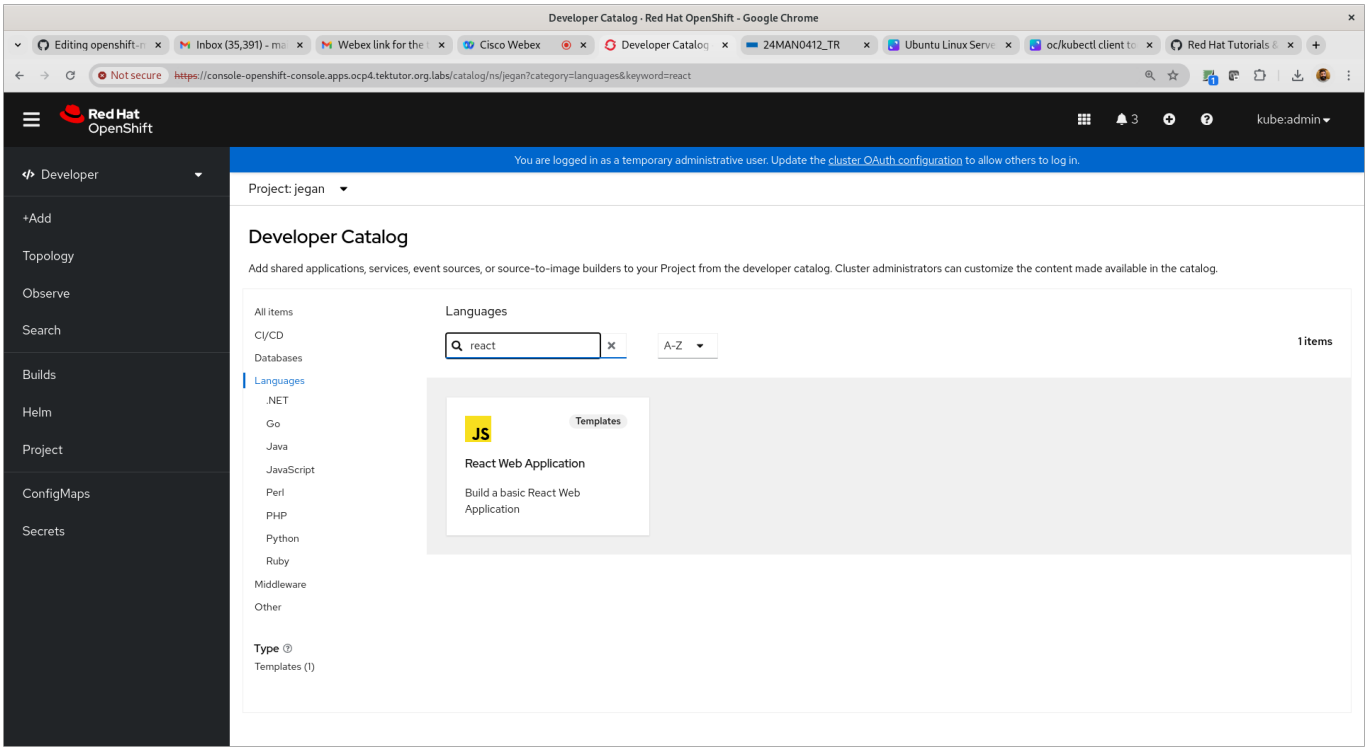


We need to click on "All Services" Under "Developer Catalog"

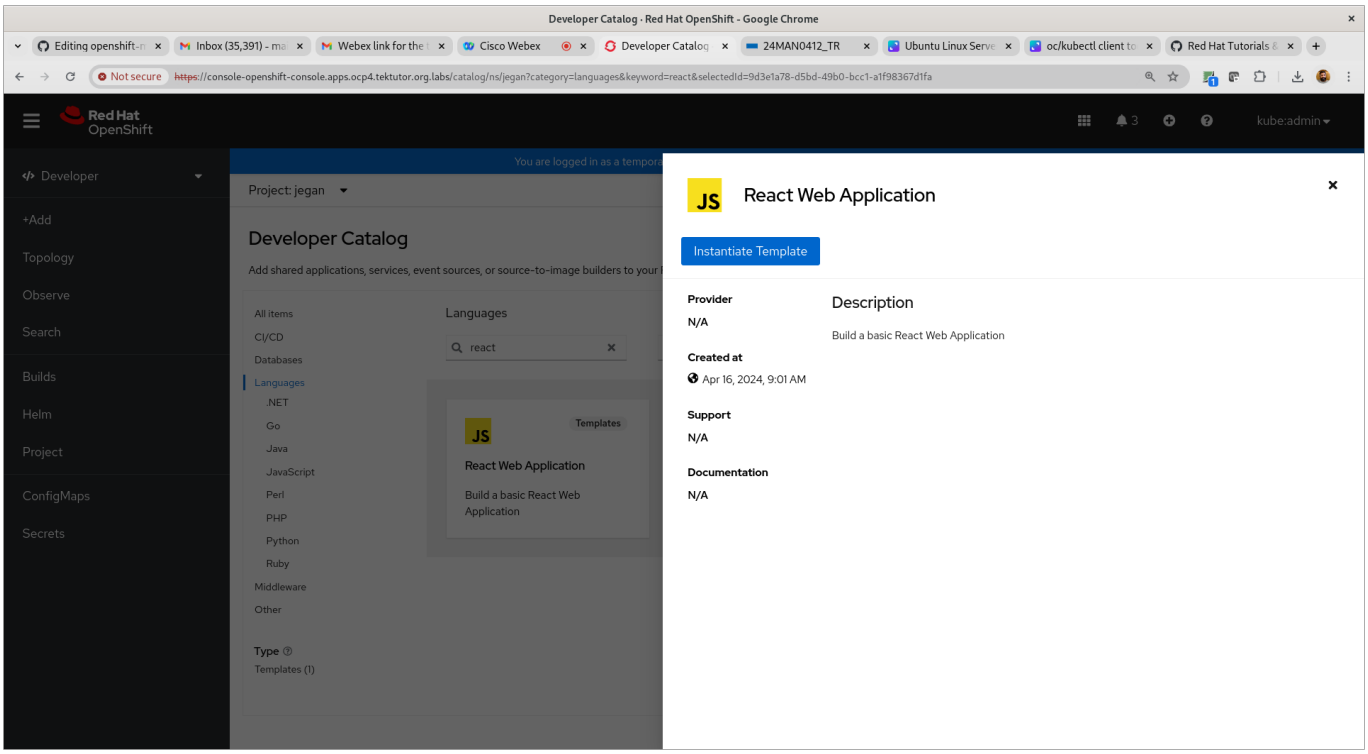


Click on "Languages" Under "All Items" reactjs

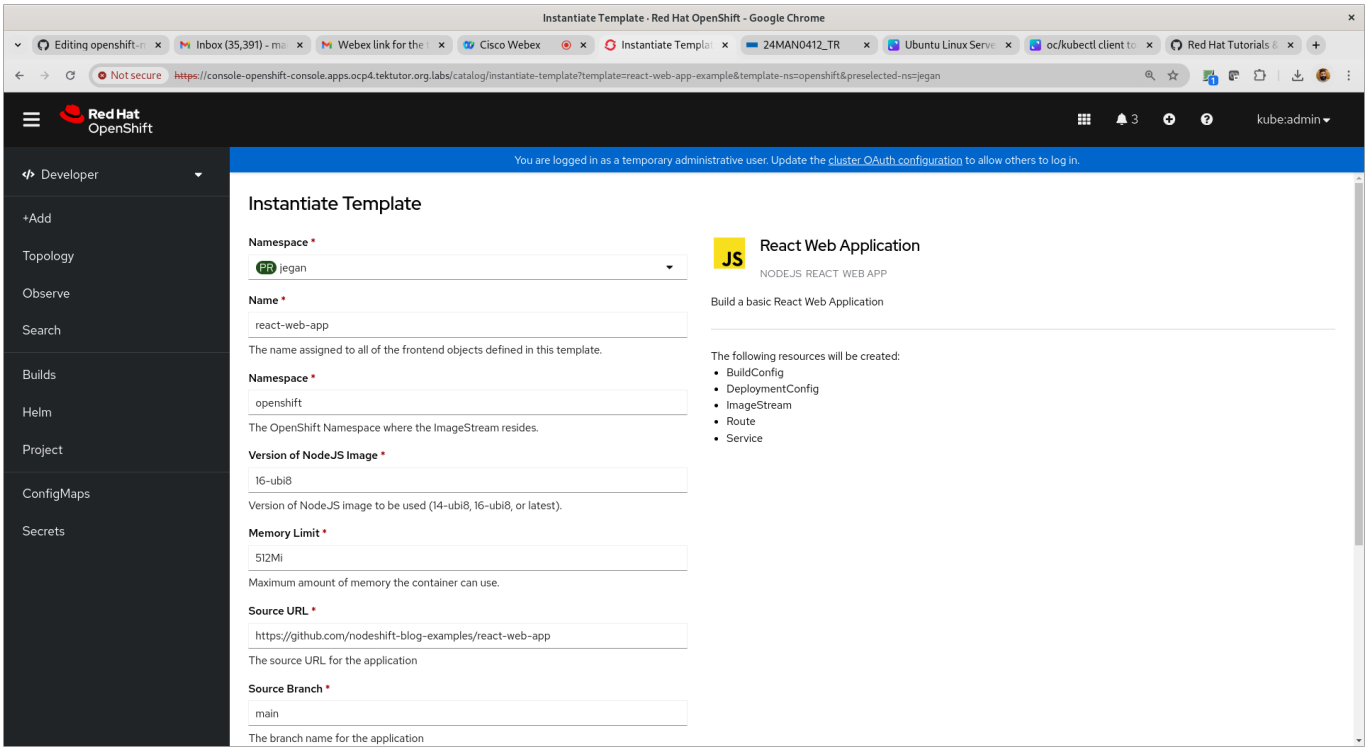
Under the "Language Search option" type "react"



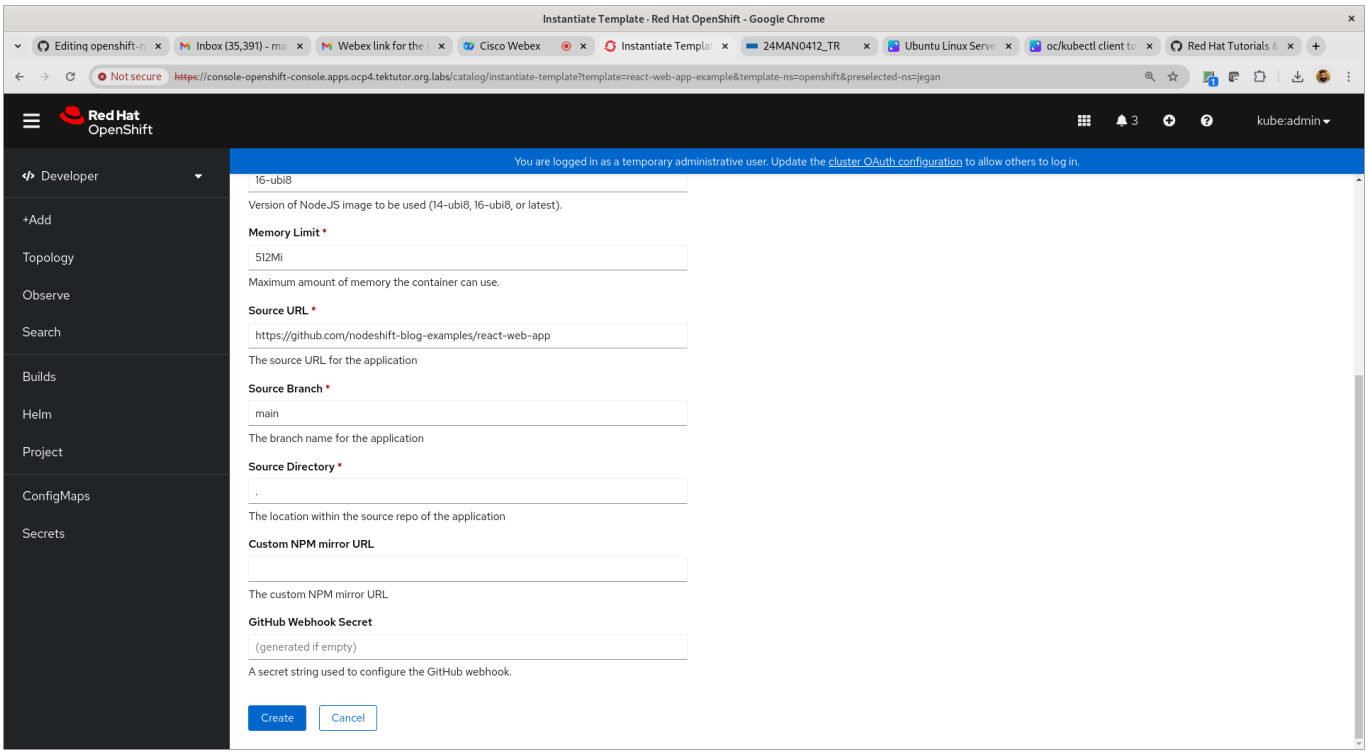
Select the "React Web Application" Template

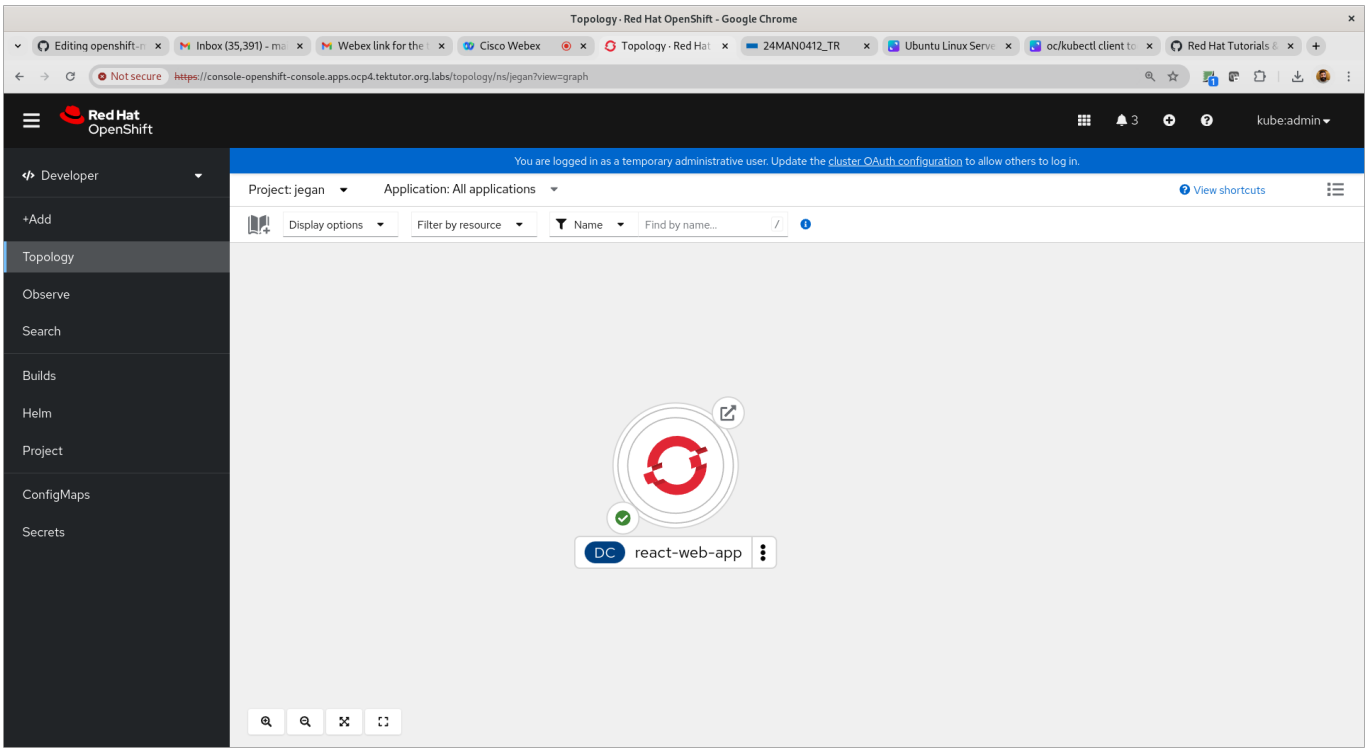


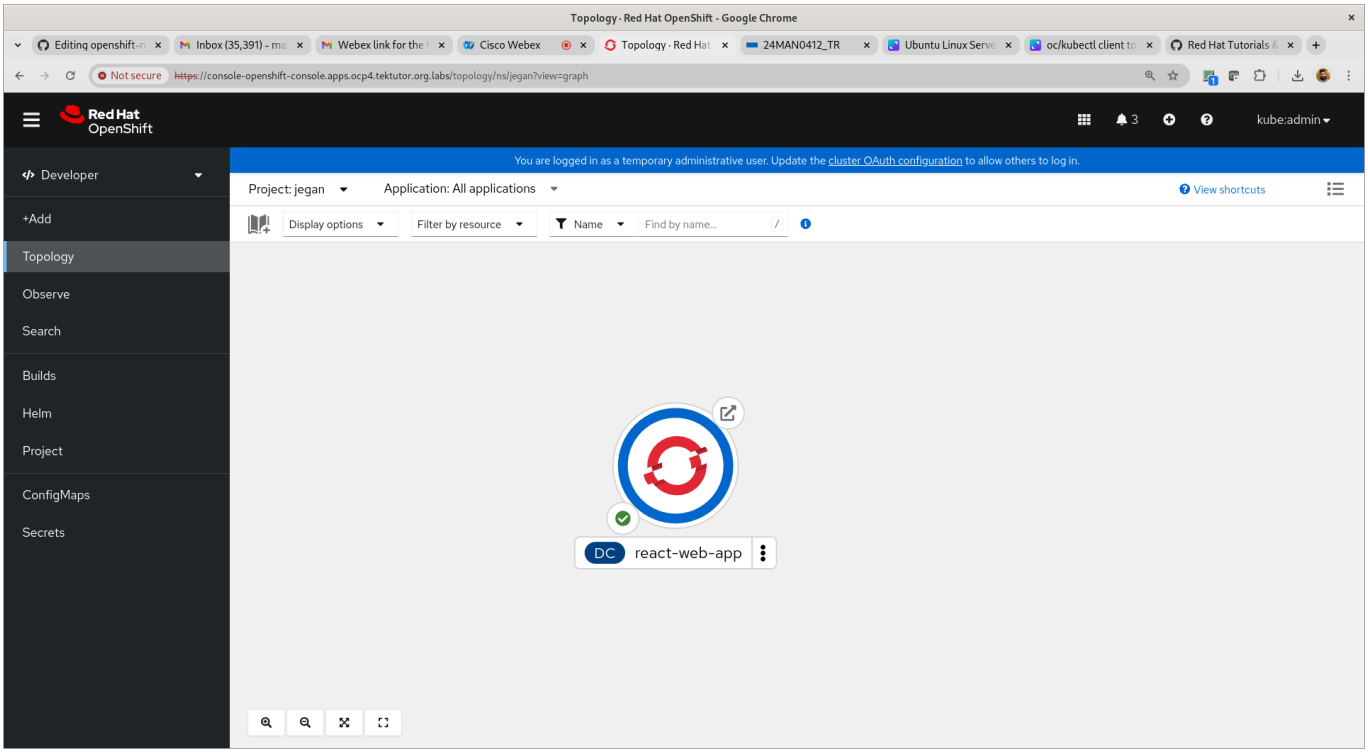
Click on "Instantiate Template"



Scroll down and click on "Create" button







Lab - Edge route

You can secure your routes with https(secured) as url as opposed to http(unsecured).

Lab - Create an edge route (https based public route url)

Find your base domain of your openshift cluster

```
oc get ingresses.config/cluster -o jsonpath={.spec.domain}
```

Expected output

```
[root@tektutor.org auth]# oc get ingresses.config/cluster -o jsonpath=
{.spec.domain}
apps.ocp.tektutor.org.labs
```

Installing openssl from source code (Already installed on Lab machines, so kindly skip this installation)

```
sudo yum -y remove openssl openssl-devel
sudo yum groupinstall 'Development Tools'
sudo yum install perl-IPC-Cmd perl-Test-Simple -y
cd /usr/src
wget https://www.openssl.org/source/openssl-3.0.0.tar.gz
tar -zxf openssl-3.0.0.tar.gz
rm openssl-3.0.0.tar.gz
cd /usr/src/openssl-3.0.0
./config
make
make test
make install

sudo ln -s /usr/local/lib64/libssl.so.3 /usr/lib64/libssl.so.3
sudo ln -s /usr/local/lib64/libcrypto.so.3 /usr/lib64/libcrypto.so.3

sudo ldconfig
sudo tee /etc/profile.d/openssl.sh<<EOF
export PATH=/usr/local/bin:$PATH
export
LD_LIBRARY_PATH=/usr/local/openssl/lib:/usr/local/openssl/lib64:\$LD_LIBRAR
Y_PATH
EOF

which openssl
openssl version
```

Let's deploy a microservice and create an edge route as shown below.

First, let's generate a private key

```
openssl genrsa -out key.key
```

We need to create a public key using the private key with specific with your organization domain

```
openssl req -new -key key.key -out csr.csr -subj="/CN=hello-
jegan.apps.ocp.tektutor.org.labs"
```

Sign the public key using the private key and generate certificate(.crt)

```
openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
oc create route edge --service spring-ms --hostname hello-
jegan.apps.ocp4.tektutor.org.labs --key key.key --cert crt.crt
```

Expected output

```
[jegan@tektutor.org edge-route]$ oc get svc
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
spring-ms   ClusterIP   172.30.208.33           8080/TCP   87m
[jegan@tektutor.org edge-route]$ oc expose deploy/nginx --port=8080
service/nginx exposed

[jegan@tektutor.org edge-route]$ oc get svc
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
nginx       ClusterIP   172.30.16.165           8080/TCP   4s
spring-ms   ClusterIP   172.30.208.33           8080/TCP   87m

[jegan@tektutor.org edge-route]$ oc get ingresses.config/cluster -o
jsonpath={.spec.domain}
apps.ocp4.tektutor.org.labs

[jegan@tektutor.org edge-route]$ oc project
Using project "jegan-devops" on server
"https://api.ocp4.tektutor.org.labs:6443".

[jegan@tektutor.org edge-route]$ openssl req -new -key key.key -out csr.csr
-subj="/CN=nginx-jegan-devops.apps.ocp4.tektutor.org.labs"

[jegan@tektutor.org edge-route]$ openssl x509 -req -in csr.csr -signkey
key.key -out crt.crt

[jegan@tektutor.org edge-route]$ oc create route edge --service nginx --
hostname nginx-jegan-devops.apps.ocp4.tektutor.org.labs --key key.key --
cert crt.crt
route.route.openshift.io/nginx created

[jegan@tektutor.org edge-route]$ oc get route
NAME    HOST/PORT                                          PATH   SERVICES
PORT    TERMINATION   WILDCARD
nginx   nginx-jegan-devops.apps.ocp4.tektutor.org.labs   nginx         edge
None
```

Info - About Flannel Pod Networking in Kubernetes/OpenShift

```
https://github.com/mvallim/kubernetes-under-the-
hood/blob/master/documentation/kube-flannel.md
```