

# Day 4

---

## Info - How to see the values stored inside etcd database?

```
oc exec -it etcd-master01.ocp4.rps.com -c etcdctl -n openshift-etcd sh
etcdctl get "/kubernetes.io/pods/jegan" --prefix=true
etcdctl get "/kubernetes.io/deployments/jegan/nginx" --prefix=true
exit
```

## Info - What is an Openshift Job?

- any one-time activity we can create them as a Job
- Example
  - delete all Persistent Volume which are unused
  - taking backup of etcd database

## Info - What is an Openshift CronJob?

- any recurring activity but that will run for few minutes and terminate we can run them as a CronJob
- Example
  - taking backup of etcd database every Sunday midnight

## Lab - Create a one-time job

```
cd ~/openshift-27may-2024
git pull
cd Day4/job
oc apply -f job.yml
oc get jobs
oc get pods
oc logs job/hello-job
```

Once you are done with the exercise, you may cleanup the resources

```
cd ~/openshift-sep-2024
cd Day4/job
oc delete -f job.yml
```

## Lab - Create a recurring job using Cronjob

```
cd ~/openshift-sep-2024
git pull
```

```
cd Day4/cronjob
oc apply -f cronjob.yml
oc get cronjobs
oc get po -w
oc logs cronjobs/cron-job
```

Once you are done with this exercise, you may delete the cronjob

```
cd ~/openshift-sep-2024
cd Day4/cronjob

oc delete -f cronjob.yml
```

## Info - What is DeploymentConfig?

- In older versions of Kubernetes to deploy stateless application we had to use ReplicationController
- In Red Hat they wanted to support declarative style while scaling and while performing rolling update, hence they add a new type of custom resource in OpenShift called DeploymentConfig
- ReplicationController supports both Scaling and rolling update, which is not a good design as it does more than one thing ( against SRP SOLID Design Principle )
- DeploymentConfig helps us deploy stateless applications
- Meanwhile, Google refactored the ReplicationController into two resources
  - 1. Deployment - which takes care of Rolling update
  - 2. ReplicaSet - which takes care of scaling up/down
- As per SOLID Design Principles
  - S - Single Responsibility Principle (SRP)
  - O - Open Closed Principle (OCP)
  - L - Liskov Substitution Principle
  - I - Interface Segregation
  - D - Dependency Injection or Dependency Inversion or Inversion of Control (IOC)
- By the Kubernetes added Deployment & ReplicaSet as an alternate to ReplicationController, the OpenShift team already added Deployment Config
- In new versions of OpenShift we would see
  - Deployment & ReplicaSet
  - DeploymentConfig ( this was introduced in openshift when there was no Deployment & ReplicaSet, hence we should avoid using DeploymentConfig instead we should use Deployment )
  - ReplicationController ( old kubernetes features now ideally we should use Deployment )

## Lab - Deploying Angular application from OpenShift Webconsole using Developer context

+Add - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco We... x Gmail x Login to your... x Login to your... x Login to your... x +Add - Red Hat OpenShift - Google Chrome x openshift-27... x 24MAN0571... x redhat-devel... x

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/add/ns/jegan

Red Hat OpenShift

Developer

Project: jegan

Add

Getting started resources ③

**Create applications using samples**  
Choose a code sample to get started creating an application with:  
Basic Quarkus →  
Basic Spring Boot →

**Build with guided documentation**  
Follow guided documentation to build applications and familiarize yourself with key features.  
Exploring Serverless applications →  
Get started with Quarkus using s2i →

**Explore new developer features**  
Discover certified Helm Charts →  
Start building your application quickly in topology →

**What's new in OpenShift 4.15**

**Developer Catalog**

**All services**  
Browse the catalog to discover, deploy and connect to services

**Database**  
Browse the catalog to discover database services to add to your application

**Eventing**

**Event Source**  
Create an Event source to register interest in a class of events from a particular system

**Broker**  
Create a Broker to define an event mesh for collecting a range of events and routes

**Serverless function**

**Import from git**  
Create and deploy stateless, Serverless functions

**Samples**  
Build from pre-built function sample code

**Git Repository**

**Import from Git**  
Import code from your Git repository to be built and deployed

**Container images**

Details on

Project: jegan

Import from Git - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco We... x Gmail x Login to your... x Login to your... x Login to your... x Import from... x openshift-27... x 24MAN0571... x redhat-devel... x

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/import/ns/jegan

Red Hat OpenShift

Developer

Project: jegan Application: All applications

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

### Import from Git

**Git**

**Git Repo URL** \*

https://github.com/tektutor/openshift-27may-2024.git

Validated

**Hide advanced Git options**

**Git reference**

main

Optional branch, tag, or commit.

**Context dir**

Day4/angular/Angular-openshift-example

Optional subdirectory for the source code, used as a context directory for build.

**Source Secret**

Select Secret name

Create Cancel

Import from Git - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco Well... x Gmail x Login to your... x Login to your... x Login to your... x Import from... x React App x openshift-27... x Digital Learn... x Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/import/ns/jegan kube:admin

### Red Hat OpenShift

Developer

Project: jegan Application: All applications

Context dir: Day4/angular/Angular-openshift-example

Optional subdirectory for the source code, used as a context directory for build.

Source Secret: Select Secret name

Secret with credentials for pulling your source code.

Multiple import strategies detected: The Dockerfile at Dockerfile is recommended.

Import Strategy: Devfile (selected), Dockerfile, Builder Image

Dockerfile path: Dockerfile

Allows the builds to use a different path to locate your Dockerfile, relative to the Context Dir field.

Dockerfile icon

Create Cancel

Import from Git - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco Well... x Gmail x Login to your... x Login to your... x Login to your... x Import from... x openshift-27... x 24MAN0571... x redhat-devel... x Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/import/ns/jegan kube:admin

### Red Hat OpenShift

Developer

Project: jegan Application: All applications

Source Secret: Select Secret name

Secret with credentials for pulling your source code.

Multiple import strategies detected: The Dockerfile at Dockerfile is recommended.

Dockerfile icon

General

Application name: tektutor

A unique name given to the application grouping to label your resources.

Name: hello-angular

A unique name given to the component that will be used to name associated resources.

Build

Create Cancel

Import from Git - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/import/ns/jegan

Red Hat OpenShift

Developer

Project: jegan Application: All applications

A unique name given to the component that will be used to name associated resources.

**Build**

Build Option

Builds

Build Option to use for building the application.

▶ Show advanced Build option

**Deploy**

Resource type

Deployment

Resource type to generate. The default can be set in [User Preferences](#).

▶ Show advanced Deployment option

**Advanced options**

Target port

8080

Create Cancel

Import from Git - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/import/ns/jegan

Red Hat OpenShift

Developer

Project: jegan Application: All applications

▶ Show advanced Build option

**Deploy**

Resource type

Deployment

Resource type to generate. The default can be set in [User Preferences](#).

▶ Show advanced Deployment option

**Advanced options**

Target port

8080

Target port for traffic.

Create a route

Exposes your component at a public URL.

▶ Show advanced Routing options

Click on the names to access advanced options for [Health checks](#), [Scaling](#), [Resource limits](#) and [Labels](#).

Create Cancel

Topology - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco Well... x Gmail x Login to your... x Login to your... x Login to your... x Topology - Re... x openshift-27... x 24MAN0571... x redhat-devel... x

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/topology/ns/jegan?view=graph

Red Hat OpenShift

Developer

Project: jegan Application: All applications

Topology

Observe

Search

Functions

Builds

Helm

Project

ConfigMaps

Secrets

hello-angular

tekutor

hello-angular-1 - Build - Logs - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco Well... x Gmail x Login to your... x Login to your... x Login to your... x hello-angular-1 x openshift-27... x 24MAN0571... x redhat-devel... x

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/jegan/builds/hello-angular-1/logs

Red Hat OpenShift

Developer

Project: jegan

Builds > Build details

hello-angular-1 Running

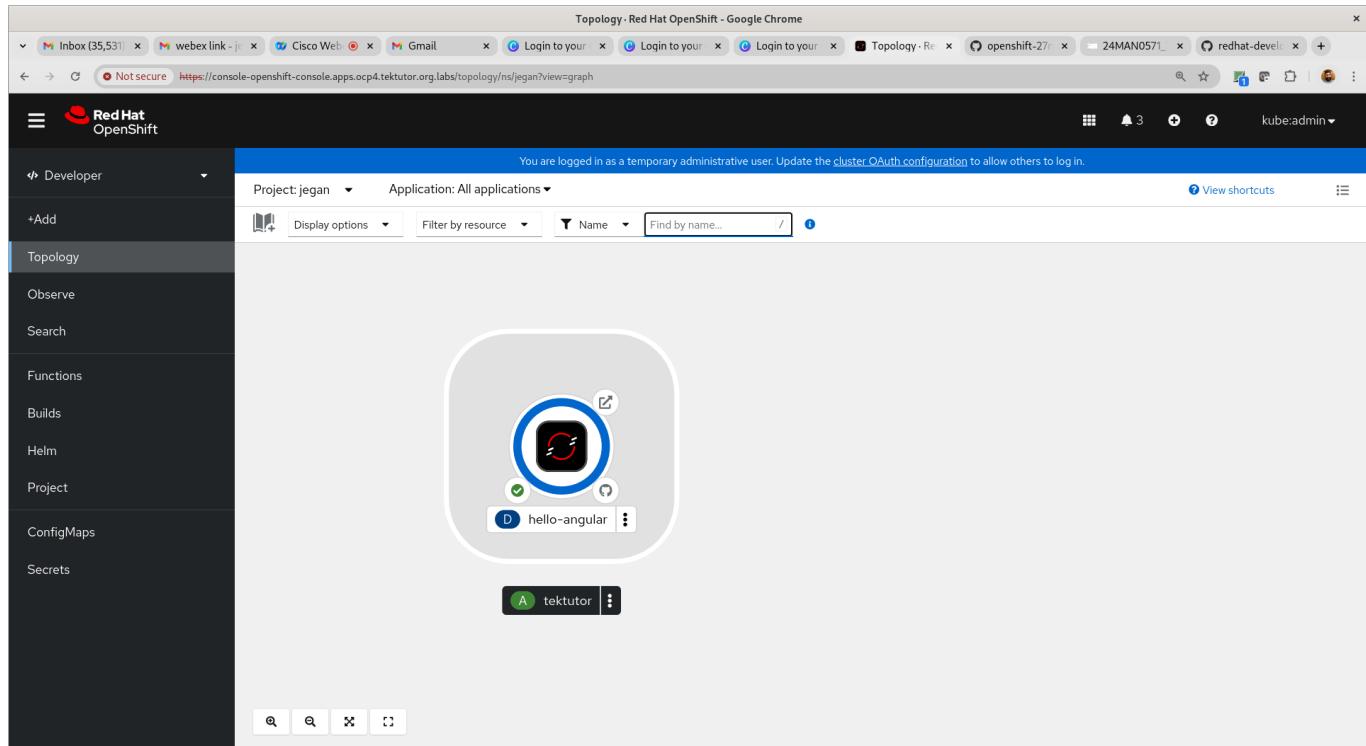
Actions

Details Metrics YAML Environment Logs Events

Log streaming... Search

14 lines

```
1 Commit: d1b5f4d1f6248def56cd5f70a3882bf3de142cad (Added angular sample project)
2 Author: Jegannathan Swaminathan <mailto:2bjegan@gmail.com>
3 Date: Thu May 30 15:06:33 2024 +0530
4 time="2024-05-30T09:47:45Z" level=info msg="Not using native diff for overlay, this may cause degraded performance for building images: kernel has CONFIG_OVERLAY_FS_RELATIVES=1"
5 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
6 I0530 09:47:45.280553 1 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
7 Caching blobs under "/var/cache/blobs".
8
9 Pulling image registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533...
10 Trying to pull registry.access.redhat.com/ubi8/nodejs-18:1-71.1695741533...
11 Getting image source signatures
12 Copying blob sha256:dd3d4615d388dbdeab85be1040d50e0fcfa768900e2ea889d721e7d520a6dad
13 Copying blob sha256:1d359a4146e4fec6c81a3025174d73437ac8fa1038f8d60735b0859b0959980d
14 Copying blob sha256:36270d048bc746f67dd912baf6ff18894c44b04ada5631ed834bf9f38ee909dc
```



Topology - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco We... x Gmail x Login to your... x Login to your... x Login to your... x Topology - Re... x openshift-27... x 24MAN0571... x redhat-devel... x

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/topology/ns/jegan?view=graph

Red Hat OpenShift

Developer

Project: jegan Application: All applications

+Add Display options Filter by resource Name Find by name... View shortcuts

Topology

Observe

Search

Functions

Builds

Helm

Project

ConfigMaps

Secrets

hello-angular

tekutor

Topology - Red Hat OpenShift - Google Chrome

Inbox (35,531) x webex link - j... x Cisco We... x Gmail x Login to your... x Login to your... x Login to your... x Topology - Re... x HelloWorld x openshift-2... x 24MAN0571... x redhat-devel... x

Not secure https://hello-angular-jegan.apps.ocp4.tektutor.org.labs

Welcome

Red Hat Developers are Awesome!

<https://developers.redhat.com>

Here are some links to help you get started:

RHEL > OpenShift > Ansible Automation Platform >

## Lab - Deploying ReactJS application in Openshift from webconsole

Topology - Red Hat OpenShift - Google Chrome

Inbox (35,531) web link - jegan Cisco Webex Gmail Login to your Cluster Login to your Cluster Login to your Cluster Topology - Red Hat OpenShift - Google Chrome Editing openshift Digital Learning

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/topology/ns/jegan?view=graph

Red Hat OpenShift

Developer

Project: jegan Application: All applications

+Add Display options Filter by resource Name Find by name... View shortcuts

Topology

Observe

Search

Functions

Builds

Helm

Project

ConfigMaps

Secrets

No resources found

Start building your application or visit the [Add page](#) for more details.

Search

Topology

Observe

Search

Functions

Builds

Helm

Project

ConfigMaps

Secrets

Start building your application or visit the [Add page](#) for more details.

Search

+Add - Red Hat OpenShift - Google Chrome

Inbox (35,531) web link - jegan Cisco Webex Gmail Login to your Cluster Login to your Cluster Login to your Cluster +Add - Red Hat OpenShift - Google Chrome Editing openshift Digital Learning

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/add/ns/jegan

Red Hat OpenShift

Developer

Project: jegan

Add

Getting started resources

**Create applications using samples**  
Choose a code sample to get started creating an application with.  
Basic Quarkus →  
Basic Spring Boot →  
[View all samples](#)

**Build with guided documentation**  
Follow guided documentation to build applications and familiarize yourself with key features.  
Exploring Serverless applications →  
Get started with Quarkus using s2i →  
[View all quick starts](#)

**Explore new developer features**  
Explore new features and resources within the developer perspective.  
Discover certified Helm Charts →  
Start building your application quickly in topology →  
[What's new in OpenShift 4.15](#)

Developer Catalog

**All services**  
Browse the catalog to discover, deploy and connect to services

**Database**  
Browse the catalog to discover database services to add to your application

Eventing

**Event Source**  
Create an Event source to register interest in a class of events from a particular system

**Broker**  
Create a Broker to define an event mesh for collecting a series of events and routes

Serverless function

**Import from git**  
Create and deploy stateless, Serverless functions

**Samples**  
Build from pre-built function sample code

Git Repository

**Import from Git**  
Import code from your Git repository to be built and deployed

Container images

Developer Catalog - Red Hat OpenShift - Google Chrome

Inbox (35,531) webex link - jegan Cisco Webex Gmail Login to your Cluster Login to your Cluster Login to your Cluster Developer Catalog Editing openshift Digital Learning Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/catalog/ns/jegan

Red Hat OpenShift

Developer

Project: jegan

Developer Catalog

Add shared applications, services, event sources, or source-to-image builders to your Project from the developer catalog. Cluster administrators can customize the content made available in the catalog.

All items All items 170 items

Filter by keyword... A-Z

CI/CD Databases Languages Middleware Other

Type .NET Application Helm Charts a10networks-a10tck akeyless-api-gateway

Builder Images (12) Devfiles (6) Event Sources (4) Helm Charts (102) Operator Backed (5) Templates (41)

**.NET** Provided by Red Hat Build and run .NET 8 applications on UBI 8. For more information about using this builder image,...

**.NET Application** Provided by Red Hat A Helm chart to build and deploy .NET applications

**a10networks-a10tck** A Helm chart for A10 Thunder Kubernetes Connector

**akeyless-api-gateway** Provided by Akeyless A Helm chart for Kubernetes that deploys akeyless-api-gateway

**Alaz** Alaz is an open-source Ddosify eBPF agent that can inspect and collect Kubernetes (K8s) service traffic without the need for code instrumentation

**Apache HTTP Server** Provided by Red Hat, Inc. An example Apache HTTP Server (httpd) application that serves

**Apache HTTP Server (httpd)** Build and serve static content via Apache HTTP Server (httpd) 2.4 on RHEL 7. For more information...

**ApiServerSource** Provided by Red Hat This object can be used to connect an event sink, such as a...

<https://console-openshift-console.apps.ocp4.tektutor.org.labs/catalog/ns/jegan?category=...>

Developer Catalog - Red Hat OpenShift - Google Chrome

Inbox (35,531) webex link - jegan Cisco Webex Gmail Login to your Cluster Login to your Cluster Login to your Cluster Developer Catalog Editing openshift Digital Learning Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/catalog/ns/jegan?category=languages&keyword=react

Red Hat OpenShift

Developer

Project: jegan

Developer Catalog

Add shared applications, services, event sources, or source-to-image builders to your Project from the developer catalog. Cluster administrators can customize the content made available in the catalog.

All items Languages 1 items

Filter by keyword... A-Z

CI/CD Databases Languages

React

JavaScript

Templates

**React Web Application** Build a basic React Web Application

**Templates**

**React Web Application** Build a basic React Web Application

Type Templates (1)

<https://console-openshift-console.apps.ocp4.tektutor.org.labs/catalog/ns/jegan?category=languages&keyword=react>

Instantiate Template - Red Hat OpenShift - Google Chrome

Inbox (35,531) - x webex link - jeg x Cisco Webex x Gmail x Login to your Ca x Login to your Ca x Login to your Ca x Instantiate Temp x Editing openshi x Digital Learning x +

Not secure https://console.openshift-console.apps.ocp4.tektutor.org.labs/catalog/instantiate-template?template=react-web-app-example&template-ns=openshift&preselected-ns=jegan

Red Hat OpenShift

Developer

+Add

Topology

Observe

Search

Functions

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.

## Instantiate Template

**Namespace \***  
PR jegan

**Name \***  
react-web-app

The name assigned to all of the frontend objects defined in this template.

**Namespace \***  
openshift

The OpenShift Namespace where the ImageStream resides.

**Version of NodeJS Image \***  
16-ubi8

Version of NodeJS image to be used (14-ubi8, 16-ubi8, or latest).

**Memory Limit \***  
512Mi

Maximum amount of memory the container can use.

**Source URL \***  
https://github.com/nodeshift-blog-examples/react-web-app

The source URL for the application

**Source Branch \***  
main

**React Web Application**  
NODEJS REACT WEB APP

Build a basic React Web Application

The following resources will be created:

- BuildConfig
- DeploymentConfig
- ImageStream
- Route
- Service

Instantiate Template - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/catalog/instantiate-template?template=react-web-app-example&template-ns=openshift&preselected-ns=jegan

Red Hat OpenShift

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Developer

Version of NodeJS image to be used (14-ubi8, 16-ubi8, or latest).

Memory Limit \*

512Mi

Maximum amount of memory the container can use.

Source URL \*

<https://github.com/nodeshift-blog-examples/react-web-app>

The source URL for the application

Source Branch \*

main

The branch name for the application

Source Directory \*

.

The location within the source repo of the application

Custom NPM mirror URL

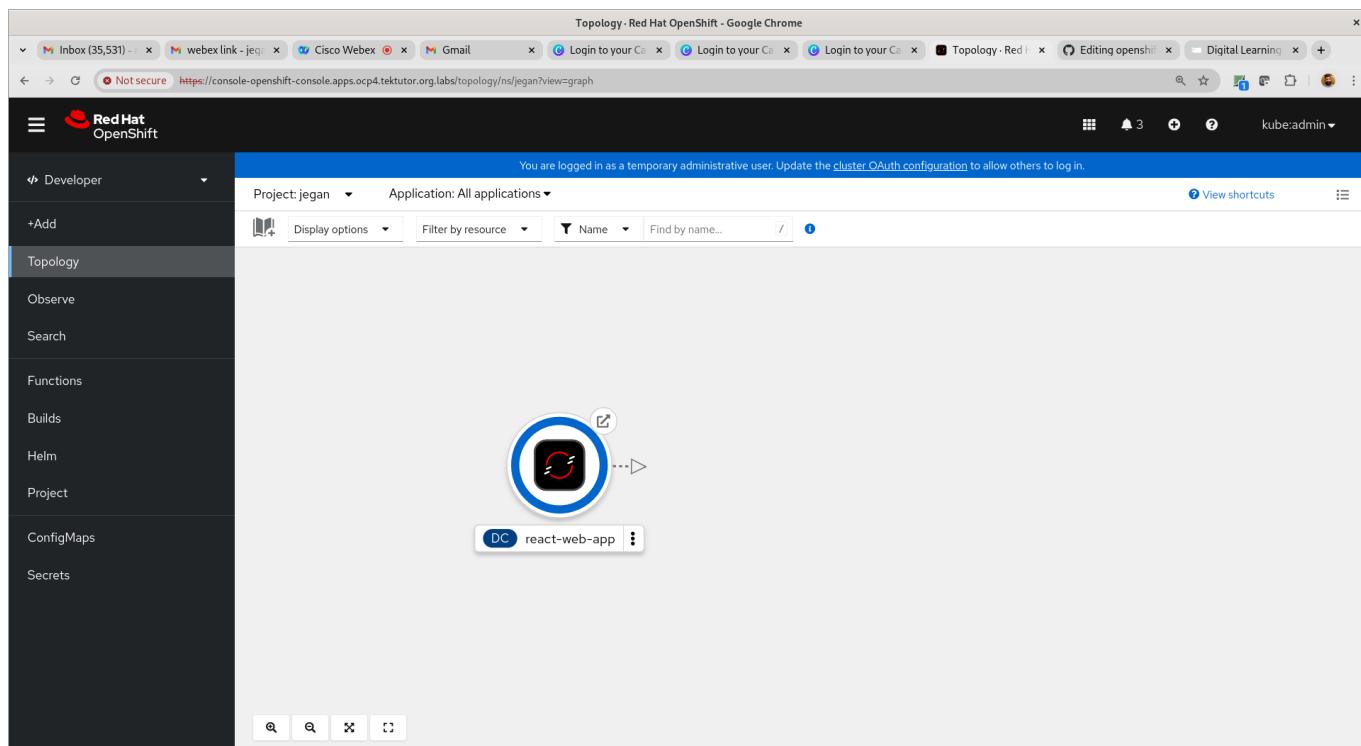
The custom NPM mirror URL.

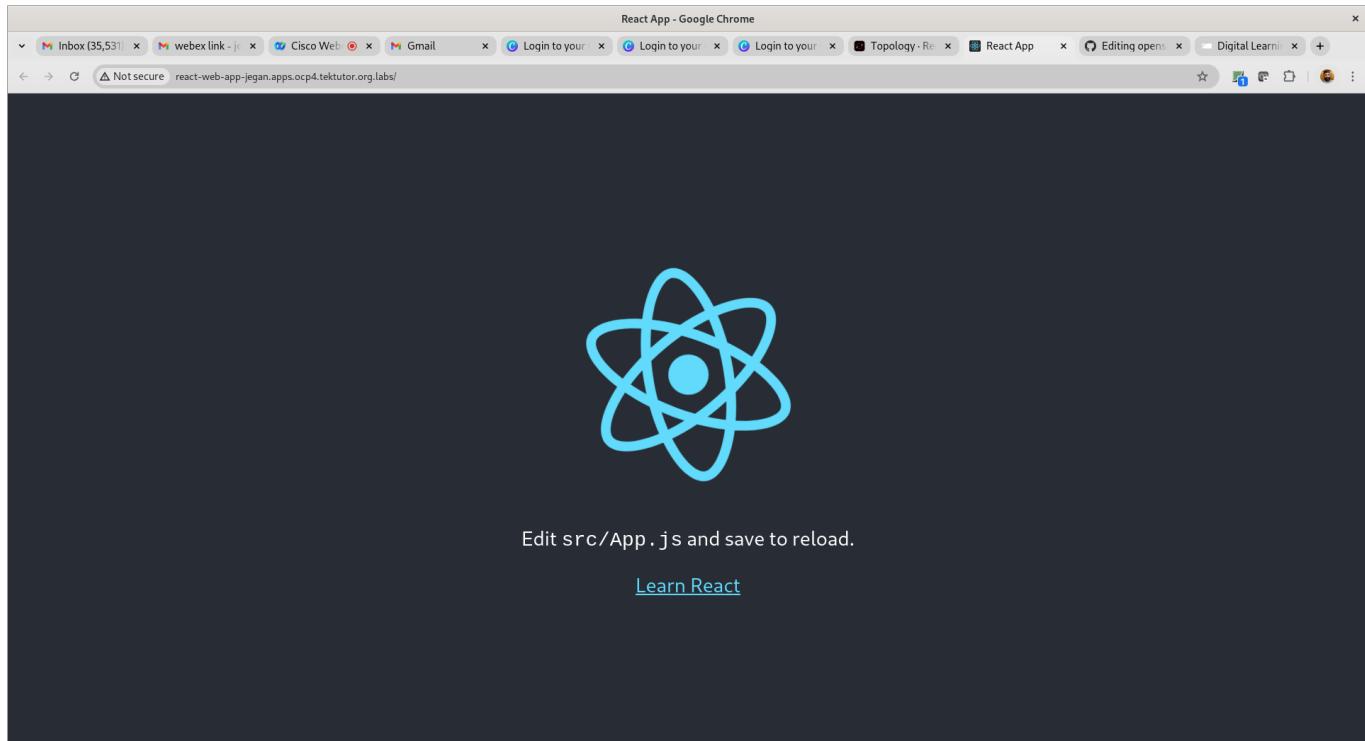
GitHub Webhook Secret

(generated if empty)

A secret string used to configure the GitHub webhook.

Create Cancel





## Lab - Deploying a Java springboot application from GitHub source code into Openshift

```
oc new-app https://github.com/tektutor/spring-ms.git --strategy=docker
oc expose svc/spring-ms
oc get bc
oc logs -f bc/spring-ms
```

### Expected output

```
jegan@tektutor.org ┶ oc new-app https://github.com/tektutor/spring-ms.git --strategy=docker
--> Found container image 41ecfe9 (3 weeks old) from registry.access.redhat.com for "registry.access.redhat.com/ubi8/openjdk-11"
Java Applications
-----
Platform for building and running plain Java applications (fat-jar and flat classpath)

Tags: builder, java
* An image stream tag will be created as "openjdk-11:latest" that will track the source image
* A Docker build using source code from https://github.com/tektutor/spring-ms.git will be created
  * The resulting image will be pushed to image stream tag "spring-ms:latest"
  * Every time "openjdk-11:latest" changes a new build will be triggered

--> Creating resources ...
imagestream.image.openshift.io "openjdk-11" created
imagestream.image.openshift.io "spring-ms" created
buildconfig.build.openshift.io "spring-ms" created
deployment.apps "spring-ms" created
service "spring-ms" created
--> Success
Build scheduled, use 'oc logs -f buildconfig/spring-ms' to track its progress.
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
  "oc expose service/spring-ms"
Run 'oc status' to view your app.
jegan@tektutor.org ┶ oc expose svc/spring-ms
route/spring-ms exposed
jegan@tektutor.org ┶ oc get bc
NAME          TYPE      FROM      LATEST
java-springboot-basic  Docker   Git      2
spring-ms      Docker   Git      1
jegan@tektutor.org ┶ oc logs -f bc/spring-ms
Cloning "https://github.com/tektutor/spring-ms.git" ...
  Commit: 82552fb8a8eb3a7cc7e8165b8878dc5e47e50db3 (Renamed deploy.yml to deploy.yaml)
  Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
  Date:  Wed Feb 15 15:11:17 2023 +0530
```

```
jegan@tektutor.org x jegan@tektutor.org x jegan@tektutor.org x
NAME          TYPE   FROM     LATEST
java-springboot-basic  Docker  Git    2
spring-ms        Docker  Git    1
jegan@tektutor.org -> oc logs -f bc/spring-ms
Cloning "https://github.com/tektutor/spring-ms.git" ...
  Commit: 82552fb8a8eb3a7c7e8165b8878dc5e47e50db3 (Renamed deploy.yml to deploy.yaml)
  Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
  Date:  Wed Feb 15 15:11:17 2023 +0530
Replaced Dockerfile FROM image registry.access.redhat.com/ubi8/openjdk-11
time="2024-05-30T10:42:44Z" level=info msg="Not using native diff for overlay, this may cause degraded performance for building images:
kernel has CONFIG_OVERLAY_FS_REDIRECT_DIR enabled"
I0530 10:42:44.174344      1 defaults.go:112] Defaulting to storage driver "overlay" with options [mountopt=metacopy=on].
Caching blobs under "/var/cache/blobs".

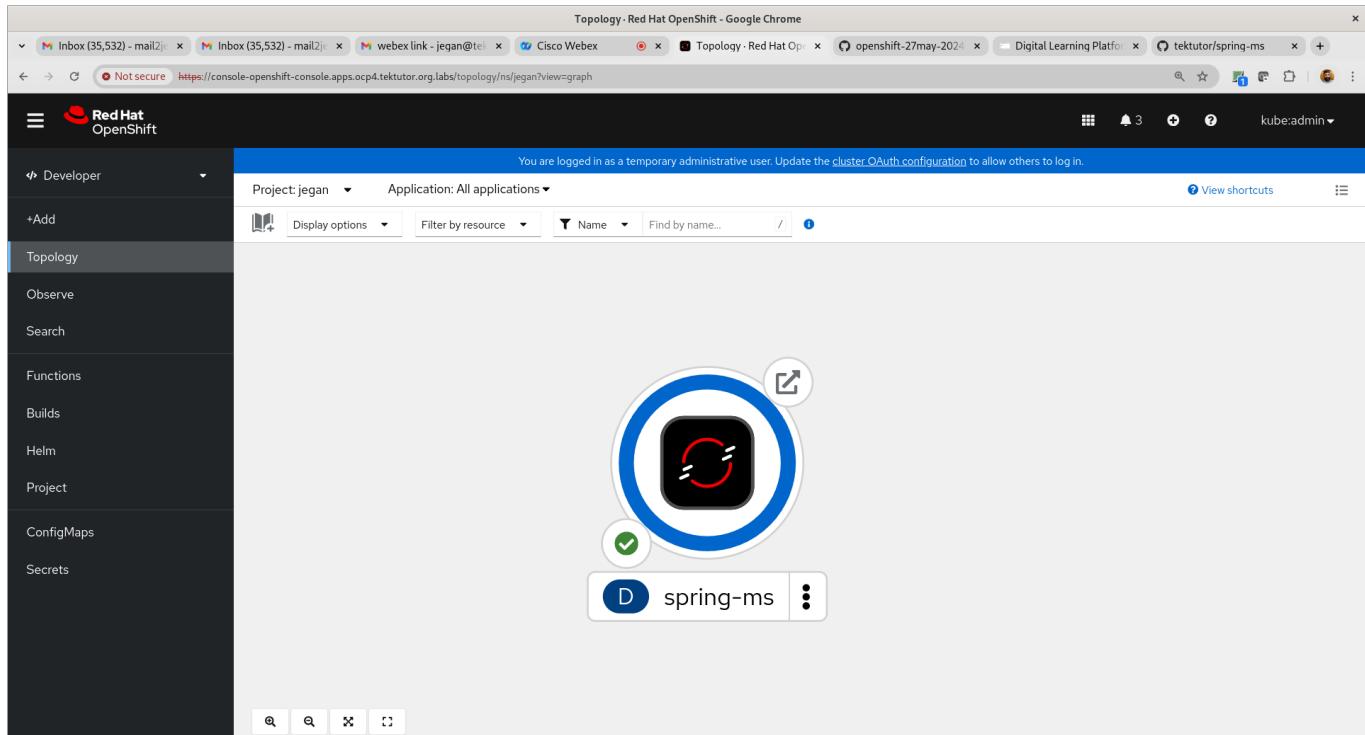
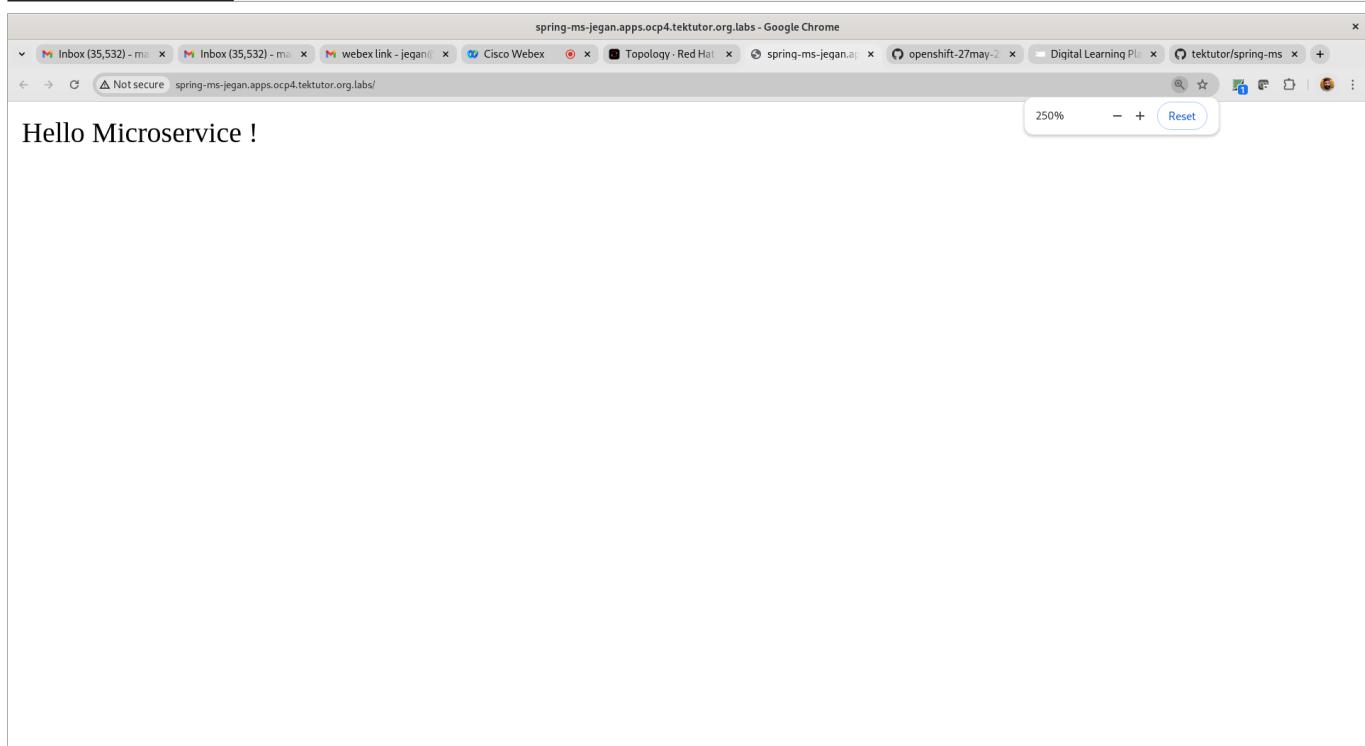
Pulling image docker.io/maven:3.6.3-jdk-11 ...
Trying to pull docker.io/library/maven:3.6.3-jdk-11...
Getting image source signatures
Copying blob sha256:5d6f1e8117dbb1c6a57603cb4f321a861a08105a81bcc6b01b0ec2b78c8523a5
Copying blob sha256:2347b0d479d7f16d7ee8d04e4ffdacc57d7d1431fa59d332f18b2e9418743
Copying blob sha256:48c2faf66abec3dce9f54d6722ff592fce6dd4fb58a0d0b72282936c6598a3b3
Copying blob sha256:004f1eed87d3f75f5e2a1a649fa7edd7f713d1300532fd0909bb39cd48437d7
Copying blob sha256:c215424f70bd949a62e8092549943905e2d4f9c874af532d7740ae8647d3a
Copying blob sha256:d7eb6c022a4e61219b32a8e07c8c22c8964ff440ebac1506121794bc15ccc
Copying blob sha256:355e8215390faee903502a9fddfc65cd823f1606f053376ba2575adce66974a1
Copying blob sha256:c5eb43522f68d7e2347e19ad70dadfc1594d25b792ede0464c2936ff902c4c6
Copying blob sha256:4fe0489a65b6405f6181358639bf85fd87776630830fd02ce8c15e34928bf9c
Copying blob sha256:413646e6fa5d7bcd9722d3e400f080a77deb505baed79afa5fedae23583af25
Copying config sha256:e23b595c92ada5c9f20a27d547ed980a445f644eb1cbde7cfb27478fa38c4691
Writing manifest to image destination

Pulling image registry.access.redhat.com/ubi8/openjdk-11@sha256:3f8b96e45b83c6170641f387331b49d690f85fa92f625057aa2ab7f2bfd41671 ...
Trying to pull registry.access.redhat.com/ubi8/openjdk-11@sha256:3f8b96e45b83c6170641f387331b49d690f85fa92f625057aa2ab7f2bfd41671...
Getting image source signatures
Copying blob sha256:50973ec5afdbfa48c719a37a132e9a827dalad121015a22a9420e05800137a28
Copying blob sha256:c9cc9ee30899ce07641f8ba17831ffd074240384f32c2b0
Copying config sha256:41ccf02a065800a58d961239b9222611116689a4d42905fd0c8c12f45729a0c7c
```

```
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org

[INFO] -----
[INFO] Total time: 15.710 s
[INFO] Finished at: 2024-05-30T10:43:43Z
[INFO] -----
--> ea2206257555
[2/2] STEP 1/6: FROM registry.access.redhat.com/ubi8/openjdk-11@sha256:3f8b96e45b83c6170641f387331b49d690f85fa92f625057aa2ab7f2bfd41671
[2/2] STEP 2/6: COPY --from=stage1 target/*.jar app.jar
--> 7f429ab58553
[2/2] STEP 3/6: EXPOSE 8080
--> 4e47026abb15
[2/2] STEP 4/6: ENTRYPOINT ["java","-jar","app.jar"]
--> c5fdc4d4c295
[2/2] STEP 5/6: ENV "OPENSHIFT_BUILD_NAME"="spring-ms-1" "OPENSHIFT_BUILD_NAMESPACE"="jegan" "OPENSHIFT_BUILD_SOURCE"="https://github.com/tektutor/spring-ms.git" "OPENSHIFT_BUILD_COMMIT"="82552fb8a8eb3a7cc7e8165b8878dc5e47e50db3"
--> 1e5f03cf6355
[2/2] STEP 6/6: LABEL "io.openshift.build.commit.author"="Jeganathan Swaminathan <mail2jegan@gmail.com>" "io.openshift.build.commit.date"="Wed Feb 15 15:11:17 2023 +0530" "io.openshift.build.commit.id"="82552fb8a8eb3a7cc7e8165b8878dc5e47e50db3" "io.openshift.build.commit.message"="Renamed deploy.yml to deploy.yaml" "io.openshift.build.commit.ref"="master" "io.openshift.build.name"="spring-ms-1" "io.openshift.build.namespace"="jegan" "io.openshift.build.source-location"="https://github.com/tektutor/spring-ms.git"
[2/2] COMMIT temp.builder.openshift.io/jegan/spring-ms-1:9a9dab49
--> 4c926dc4eb0d
Successfully tagged temp.builder.openshift.io/jegan/spring-ms-1:9a9dab49
4c926dc4eb0d455a9ef347131166c05995ef61c7849ab5ed7c333388047bbd2f

Pushing image image-registry.openshift-image-registry.svc:5000/jegan/spring-ms:latest ...
Getting image source signatures
Copying blob sha256:8e3b289656e83e3efc76f9e917e1e0e610dc039bd26c41924cc28060f4e3f3d0
Copying blob sha256:ca19c1d8b6a56d82b4d9cc9ee30899ce07641f8ba17831ffd074240384f32cb0
Copying blob sha256:50973ec5afdbaf48c719a37a132e9a827dalad121015a22a9420e05800137a28
Copying config sha256:4c926dc4eb0d455a9ef347131166c05995ef61c7849ab5ed7c333388047bbd2f
Writing manifest to image destination
Successfully pushed image-registry.openshift-image-registry.svc:5000/jegan/spring-ms@sha256:22aef16073f9e45ec5db513193a29b504172664e9560
6f5bbbe6e6d8780c29d1
Push successful
jegan@tektutor.org ➤
```

Info - Installing openssl ( is already installed in our lab - just for your future reference )

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_LIBRARY
_PATH
EOF

which openssl
openssl version
```

## Lab - Create an edge route ( https url )

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.ocp4.rps.com
```

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
```

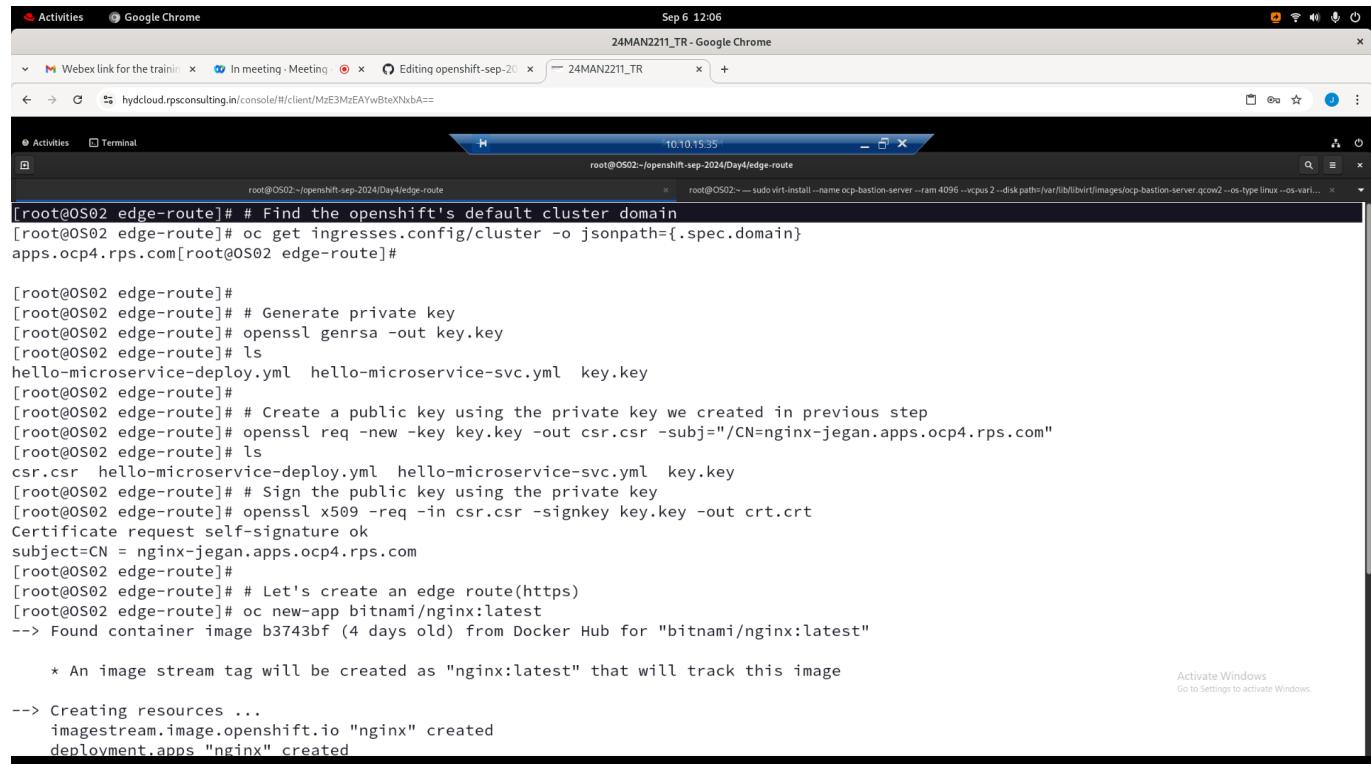
We need to create a public key using the private key with specific with your organization domain

```
openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-x-jegan.apps.ocp4.rps.com"
```

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 nginx --hostname nginx-x-jegan.apps.ocp4.rps.com --key key.key --cert crt.crt
```

Expected output



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is 'root@OS02:~/.openshift/sep-2024/Day4/edge-route'. The terminal content shows the following command sequence:

```
[root@OS02 edge-route]# # Find the openshift's default cluster domain
[root@OS02 edge-route]# oc get ingresses.config/cluster -o jsonpath={.spec.domain}
apps.ocp4.rps.com[root@OS02 edge-route]#
```

[root@OS02 edge-route]# # Generate private key
[root@OS02 edge-route]# openssl genrsa -out key.key
[root@OS02 edge-route]# ls
hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]#

[root@OS02 edge-route]# # Create a public key using the private key we created in previous step
[root@OS02 edge-route]# openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-x-jegan.apps.ocp4.rps.com"
[root@OS02 edge-route]# ls
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-x-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

\* An image stream tag will be created as "nginx:latest" that will track this image
--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created

At the bottom right of the terminal window, there is a watermark: 'Activate Windows' and 'Go to Settings to activate Windows.'

Activities Google Chrome Sep 6 12:06 24MAN2211\_TR - Google Chrome

Webex link for the trainin... In meeting - Meeting Editing openshift-sep-20 24MAN2211\_TR

Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/edge-route

```
root@OS02:~/openshift-sep-2024/Day4/edge-route
[root@OS02 edge-route]# # Find the openshift's default cluster domain
[root@OS02 edge-route]# oc get ingresses.config/cluster -o jsonpath='{.spec.domain}
apps.ocp4.rps.com[root@OS02 edge-route]#
```

```
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Generate private key
[root@OS02 edge-route]# openssl genrsa -out key.key
[root@OS02 edge-route]# ls
hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Create a public key using the private key we created in previous step
[root@OS02 edge-route]# openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-jegan.apps.ocp4.rps.com"
[root@OS02 edge-route]# ls
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

* An image stream tag will be created as "nginx:latest" that will track this image
```

Activate Windows  
Go to Settings to activate Windows.

```
--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created
```

Activities Google Chrome Sep 6 12:06 24MAN2211\_TR - Google Chrome

Webex link for the trainin... In meeting - Meeting Editing openshift-sep-20 24MAN2211\_TR

Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/edge-route

```
root@OS02:~/openshift-sep-2024/Day4/edge-route
[root@OS02 edge-route]# # Find the openshift's default cluster domain
[root@OS02 edge-route]# oc get ingresses.config/cluster -o jsonpath='{.spec.domain}
apps.ocp4.rps.com[root@OS02 edge-route]#
```

```
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Generate private key
[root@OS02 edge-route]# openssl genrsa -out key.key
[root@OS02 edge-route]# ls
hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Create a public key using the private key we created in previous step
[root@OS02 edge-route]# openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-jegan.apps.ocp4.rps.com"
[root@OS02 edge-route]# ls
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

* An image stream tag will be created as "nginx:latest" that will track this image
```

Activate Windows  
Go to Settings to activate Windows.

```
--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created
```

Activities Google Chrome Sep 6 12:06 24MAN2211\_TR - Google Chrome

Webex link for the trainin... In meeting - Meeting Editing openshift-sep-20... 24MAN2211\_TR

Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/edge-route

```
[root@OS02 edge-route]# # Find the openshift's default cluster domain
[root@OS02 edge-route]# oc get ingresses.config/cluster -o jsonpath={.spec.domain}
apps.ocp4.rps.com[root@OS02 edge-route]#
```

```
[root@OS02 edge-route]# # Generate private key
[root@OS02 edge-route]# openssl genrsa -out key.key
[root@OS02 edge-route]# ls
hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Create a public key using the private key we created in previous step
[root@OS02 edge-route]# openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-jegan.apps.ocp4.rps.com"
[root@OS02 edge-route]# ls
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

* An image stream tag will be created as "nginx:latest" that will track this image
```

Activate Windows  
Go to Settings to activate Windows.

```
--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created
```

Activities Google Chrome Sep 6 12:06 24MAN2211\_TR - Google Chrome

Webex link for the trainin... In meeting - Meeting Editing openshift-sep-20... 24MAN2211\_TR

Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/edge-route

```
[root@OS02 edge-route]# openssl genrsa -out key.key
[root@OS02 edge-route]# ls
hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Create a public key using the private key we created in previous step
[root@OS02 edge-route]# openssl req -new -key key.key -out csr.csr -subj="/CN=nginx-jegan.apps.ocp4.rps.com"
[root@OS02 edge-route]# ls
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

* An image stream tag will be created as "nginx:latest" that will track this image
```

Activate Windows  
Go to Settings to activate Windows.

```
--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created
service "nginx" created
--> Success
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
'oc expose service/nginx'
Run 'oc status' to view your app.
```

```
root@OS02 edge-route]# oc create route edge --service nginx --hostname nginx-jegan.apps.ocp4.rps.com --key key.key --cert crt.crt
```

```
Activities Google Chrome Sep 6 12:07 24MAN2211_TR - Google Chrome
Webex link for the trainin In meeting - Meeting Editing openshift-sep-20 24MAN2211_TR + hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXnbAa Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/edge-route
root@OS02:~/openshift-sep-2024/Day4/edge-route
root@OS02:~/ -- sudo virt-install --name ocp-bastion-server --ram 4096 --cpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari...
csr.csr hello-microservice-deploy.yml hello-microservice-svc.yml key.key
[root@OS02 edge-route]# # Sign the public key using the private key
[root@OS02 edge-route]# openssl x509 -req -in csr.csr -signkey key.key -out crt.crt
Certificate request self-signature ok
subject=CN = nginx-jegan.apps.ocp4.rps.com
[root@OS02 edge-route]#
[root@OS02 edge-route]# # Let's create an edge route(https)
[root@OS02 edge-route]# oc new-app bitnami/nginx:latest
--> Found container image b3743bf (4 days old) from Docker Hub for "bitnami/nginx:latest"

* An image stream tag will be created as "nginx:latest" that will track this image

--> Creating resources ...
imagestream.image.openshift.io "nginx" created
deployment.apps "nginx" created
service "nginx" created
--> Success
Application is not exposed. You can expose services to the outside world by executing one or more of the commands below:
'oc expose service/nginx'
Run 'oc status' to view your app.
[root@OS02 edge-route]# oc create route edge --service nginx --hostname nginx-jegan.apps.ocp4.rps.com --key key.key --cert crt.crt
route/nginx created
[root@OS02 edge-route]# oc get route
NAME            HOST/PORT          PATH  SERVICES          PORT  TERMINATION  WILDCARD
nginx           nginx-jegan.apps.ocp4.rps.com      nginx          8080-tcp  edge        None
openshift-sep-204  openshift-sep-2024-jegan.apps.ocp4.rps.com  openshift-sep-2024  8080-tcp  edge        None
[root@OS02 edge-route]#
```

## Info - OpenShift Network Model

## What is Flannel?

## What is Calico?

## What is Weave?

## What is edge route?

## Info - Persistent Volume Overview

- any application that stores/retrieves data they could either use the Pod storage or external storage
- as the Pod's life time is short, storing the permanent data in a short lived Pod doesn't sound correct
- hence, it is considered a bad practice to modify a pod once it is created
- as per devops philosophy, we must use containers, Pods like an immutable(read-only) resource, though it is mutable
- hence, we should be using some external storage to persistent our application logs, database, etc
- Openshift Administrators can provision external storage either manually or dynamically
- In case the Administrators prefer provisioning the Persistent Volume(storage) manually, they need to create PV with various size, access permissions, etc as per dev/qa/operation team's requirement
- The Persistent Volume (PV storage) can be provisioned
  - from NFS Server ( we would be using NFS Server )
  - Some storage solution in on-premise environment
  - could use AWS S3 buckets, AWS EBS, Azure Storage, etc.
- the Persistent Volume will have the below parameters
  - disk size
  - access mode
  - storageclass(optional)
  - labels(optional)
- Persistent volumes are created in the cluster scope

## Info - Persistent Volume Claim Overview

- application Pods are created within a project, hence the PVC is also created within a project
- any application (Pod) that needs storage will have to ask Openshift by creating a Persistent Volume Claim
- the PVC will define the below
  - size of the storage required
  - access mode expected
  - storageclass(optional)
  - any label restrictions (optional)
- openshift will search the cluster for a matching Persistent Volume and if it finds a matching PV, it let's your application claim and use it
- in case openshift is not able to find a matching Persistent Volume against a Persistent Volume Claim, the Pod will be kept in Pending state as it won't be able to proceed without storage(PVC--->PV)

## Lab - Deploying multipod application Wordpress with MariaDB database

```
cd ~/openshift-sep-2024
git pull
cd Day3/persistent-volume/wordpress
```

```
ls
./deploy.sh
```

## Expected output

```
Sep 5 12:18
24MAN2211_TR - Google Chrome
Activities Google Chrome
Webex link fo In meeting 24MAN2211_ Install Multip Red Hat Dev Red Hat Open Create an Ope Dashboard Chapter 2. In Editing opens mysql.default
hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==

Activities Terminal
10.10.15.35
root@OS02:~/openshift-sep-2024/Day3/persistent-volume/wordpress
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari...
[root@OS02 wordpress]# oc project
Using project "jegan" on server "https://api.ocp4.rps.com:6443".
[root@OS02 wordpress]# oc get all
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.10000+
No resources found in jegan namespace.
[root@OS02 wordpress]# ls
deploy.sh mariadb-deploy.yml mariadb-pv.yml undeploy.sh wordpress-deploy.yml wordpress-pv.yml wordpress-svc.yml
deploy.yml mariadb-pvc.yml mariadb-svc.yml undeploy.yml wordpress-pvc.yml wordpress-route.yml
[root@OS02 wordpress]# ./deploy.sh

Deploying mariadb ...
persistentvolume/mariadb-pv-jegan created
persistentvolumeclaim/mariadb-pvc-jegan created
deployment.apps/mariadb created
service/mariadb created

Deploying wordpress ...
persistentvolume/wordpress-pv-jegan created
persistentvolumeclaim/wordpress-pvc-jegan created
deployment.apps/wordpress created
service/wordpress created
route.route.openshift.io/wordpress created
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE
persistentvolume/image-registry-pv 1000Gi RWX Retain Bound openshift-image-registry/image-registry-storage
<unset> 20h

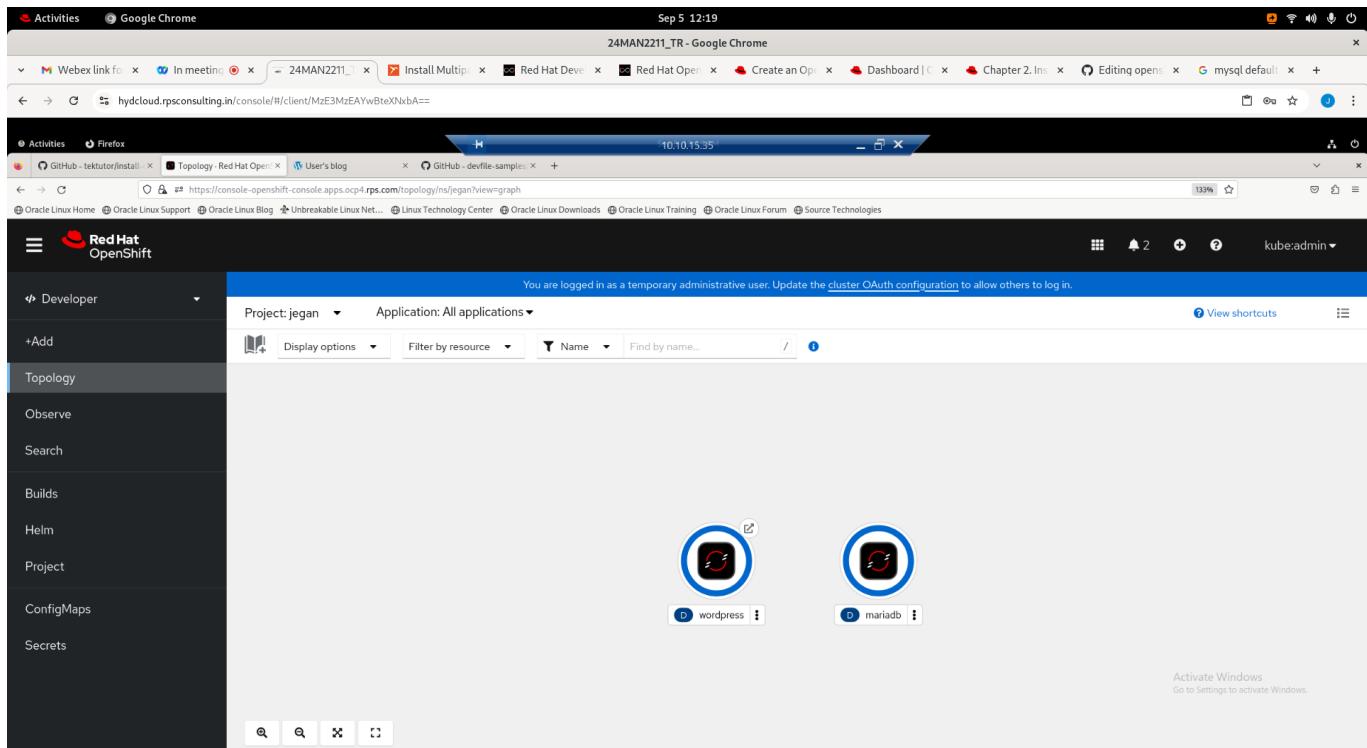
Activities Google Chrome
Sep 5 12:18
24MAN2211_TR - Google Chrome
Activities Terminal
10.10.15.35
root@OS02:~/openshift-sep-2024/Day3/persistent-volume/wordpress
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari...
[root@OS02 wordpress]# oc get persistentvolume
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE
persistentvolume/image-registry-pv 1000Gi RWX Retain Bound openshift-image-registry/image-registry-storage
<unset> 20h
persistentvolume/mariadb-pv-jegan 100Mi RWO Delete Bound jegan/mariadb-pvc-jegan
<unset>
persistentvolume/wordpress-pv-jegan 100Mi RWO Delete Bound jegan/wordpress-pvc-jegan
<unset>

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS
GE
persistentvolumeclaim/mariadb-pvc-jegan Bound mariadb-pv-jegan 100Mi RWO <unset> 1
persistentvolumeclaim/wordpress-pvc-jegan Bound wordpress-pv-jegan 100Mi RWO <unset> 1

NAME READY STATUS RESTARTS AGE
pod/mariadb-5b9895469b-6jbdn 0/1 ContainerCreating 0 1s
pod/wordpress-7f8ffff478-m44p2 0/1 ContainerCreating 0 1s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/mariadb ClusterIP 172.30.131.64 <none> 3306/TCP 1s
service/wordpress ClusterIP 172.30.234.182 <none> 8080/TCP 0s

NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD
route.route.openshift.io/wordpress wordpress-jegan.apps.ocp4.rps.com 8080 wordpress 8080 None
[root@OS02 wordpress]#
```



Activities Google Chrome Sep 5 12:19 24MAN2211 TR - Google Chrome

Webex link fo In meeting 24MAN2211\_ Install Multip... Red Hat Dev... Red Hat Open... Create an Ope... Dashboard | Chapter 2. In... Editing opens... mysql default

hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==

Activities Firefox 10.10.15.35

GitHub - tutkor/install Topology - Red Hat Open... User's blog GitHub - devfile-samples

https://console.openshift-console.apps.ocp4.rps.com/topology?view=graph

Oracle Linux Home Oracle Linux Support Oracle Linux Blog Unbreakable Linux Net... Linux Technology Center Oracle Linux Downloads Oracle Linux Training Oracle Linux Forum Source Technologies

Red Hat OpenShift

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Developer Project: jegan Application: All applications View shortcuts

+Add Display options Filter by resource Name Find by name... /

Topology

Observe

Search

Builds

Helm

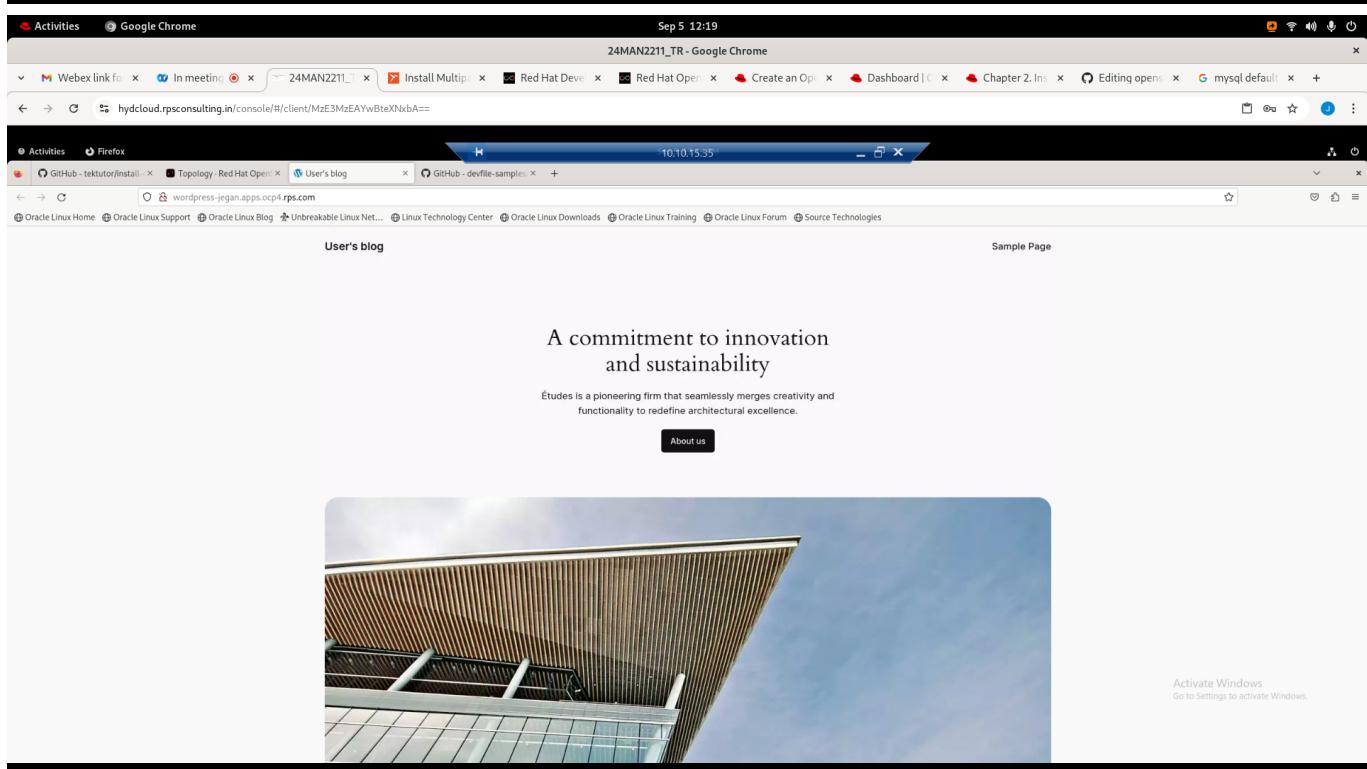
Project

ConfigMaps

Secrets

wordpress mariadb

Activate Windows Go to Settings to activate Windows.

Activities Google Chrome Sep 5 12:19 24MAN2211 TR - Google Chrome

Webex link fo In meeting 24MAN2211\_ Install Multip... Red Hat Dev... Red Hat Open... Create an Ope... Dashboard | Chapter 2. In... Editing opens... mysql default

hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==

Activities Firefox 10.10.15.35

GitHub - tutkor/install Topology - Red Hat Open... User's blog GitHub - devfile-samples

wordpress-jegan.apps.ocp4.rps.com

Activate Windows Go to Settings to activate Windows.

Users blog Sample Page

A commitment to innovation and sustainability

Etudes is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

About us

Activate Windows Go to Settings to activate Windows.

## Info - StatefulSet Overview

- Deployment is an Openshift/Kubernetes component through which we deploy stateless applications
- StatefulSet is an Openshift/kubernetes component through which we deploy stateful applications
- Deployment
  - supports Persistent Volume and Persistent Volume Claims
  - each Pod gets a random name when Deployment is scaled up/down, hence pod names are unpredictable
  - when scale down happens a random pod within the deployment will be

deleted

- when scaled up a new pod with random name gets added
- hence scaling up/down a stateless application is easy and can be achieved with Deployment
- StatefulSet
  - supports Persistent Volume and Persistent Volume Claims
  - each Pod gets an unique and stable identify (name)
  - let's say we create a statefulset named mysql, its first pod will be named mysql-0 which is the sticky pod identify
    - even if the mysql-0 master Pod crashes, the new pod even if runs in a different node it will be created with exact same name mysql-0. This is important otherwise the headless service that refers to mysql-0 Pod will not work
    - if mysql-1 slave Pod crashes, the new pod could be scheduled on the same/different node but its name will be mysql-1 for sure. This is important for the mysql-2 Pod as it will be dependent on mysql-1 Pod to synchronize the data
    - when a statefulset is scaled up
      - it will ensure the first pod mysql-0 is created and running successfully
      - the second pod with an unique name mysql-1 will be created
      - the third pod with an unique name mysql-2 will be created only after ensuring mysql-1 pod is successfully running
    - when a statefulset is scaled down
      - it is guaranteed the last pod i.e mysql-2 will be deleted first followed by mysql-1 and so on
      - allows each Pod using a separate storage(Persistent Volume and Claim)
      - ideal for creating a cluster of database Pods
      - the first pod in statefulset can be configured to act as a master, while the other Pods can be configured to act as a slave
        - while StatefulSet gives all facilities to create a cluster of database/stateful application pods, we still have to manually do the configuration required to setup a db cluster ourselves
        - creating a cluster of legacy db servers requires only the master pod (mysql-0 ) with write access to database
        - while other slave pods ( mysql-1, mysql-2, etc., ) has only read-only access
        - the mysql-1 when created, it clones mysql-0 Pod data and then starts synchronizing in its own copy as new changes happen in mysql-0
        - the mysql-2 when created, it clones mysql-1 Pod and then starts synchronizing new changes as it happens in mysql-1
        - generally statefulset applications get at least two services, one headless service without any IP and other service which load balances between all the replicas of read-only pods

## Lab - Statefulset

## Lab - Using ConfigMap and Secrets and avoid hard coding

```
cd ~/openshift-sep-2024
git pull
cd Day4/configs-and-secrets
./deploy
```

## Info - Helm Overview

- Helm is a package manager to download, install, uninstall, upgrade applications into Kubernetes/OpenShift cluster
- Helm packaged application is called Charts
- It can be download from community maintained Helm repository and deploy the application into your Kubernetes/OpenShift cluster
- We could also create our custom Helm charts and deploy them in Kubernetes/OpenShift cluster

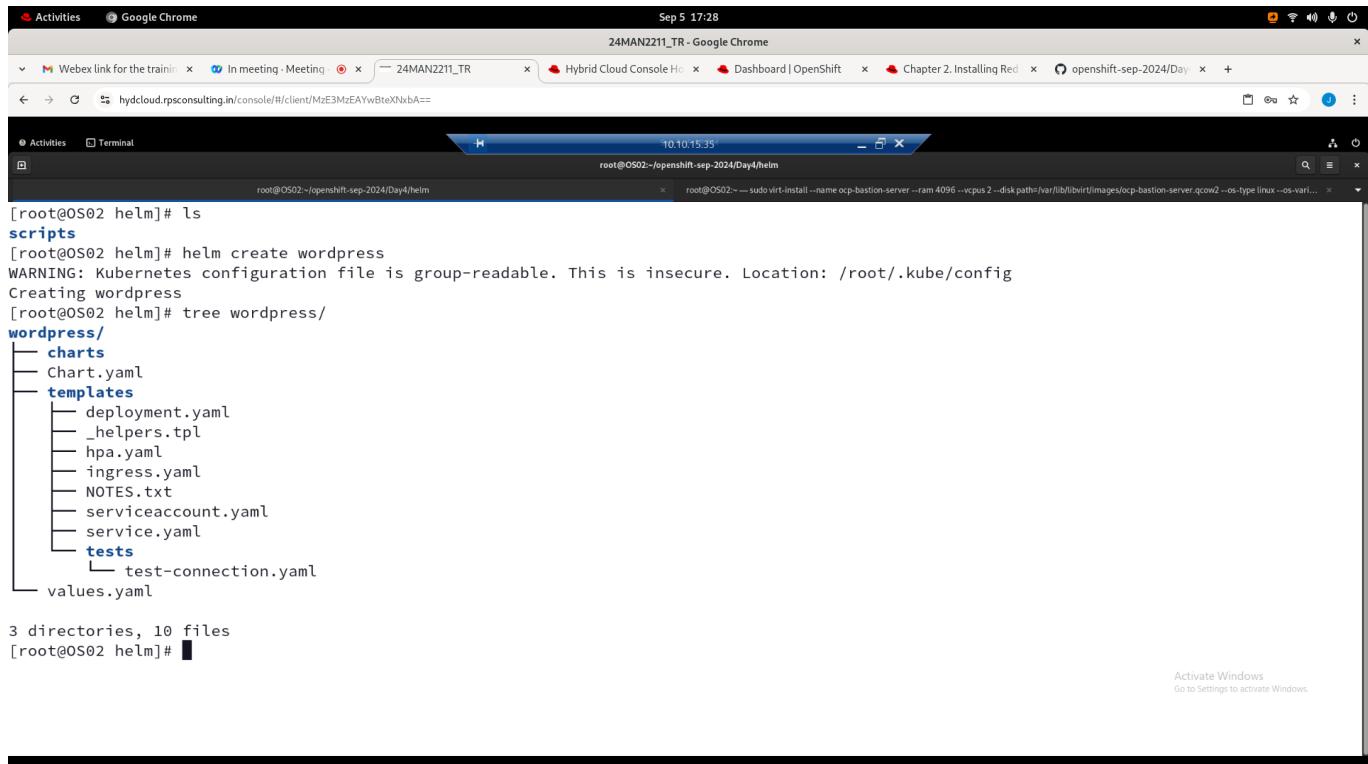
## Demo - Install Helm

```
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
```

## Lab - Creating a Helm chart for WordPress with MariaDB and deploy it in our OpenShift cluster

```
helm version
helm create wordpress
tree wordpress
```

## Expected output



```

Activities Google Chrome Sep 5 17:28
Webex link for the trainin... In meeting - Meeting 24MAN2211_TR - Google Chrome
hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==
Activities Terminal 10:10:15:35
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-variant=redhat-7.9 --cpu host --graphics none --network network:br0 --location /var/www/html/wordpress --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-variant=redhat-7.9 --cpu host --graphics none --network network:br0 --location /var/www/html/wordpress
scripts
[root@OS02 helm]# helm create wordpress
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
Creating wordpress
[root@OS02 helm]# tree wordpress/
wordpress/
├── charts
│   └── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── hpa.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   ├── serviceaccount.yaml
│   ├── service.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml

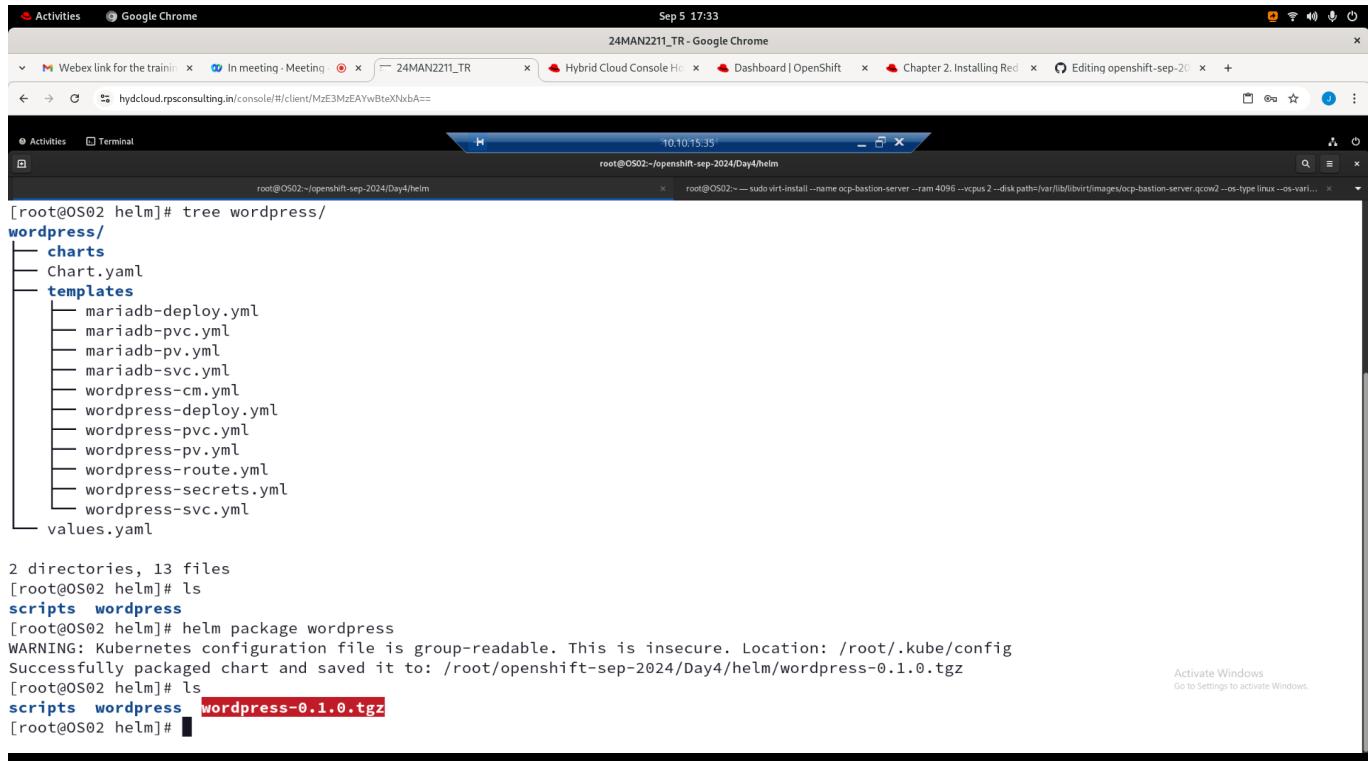
3 directories, 10 files
[root@OS02 helm]#

```

## Create a helm package

```
tree wordpress
```

## Expected output

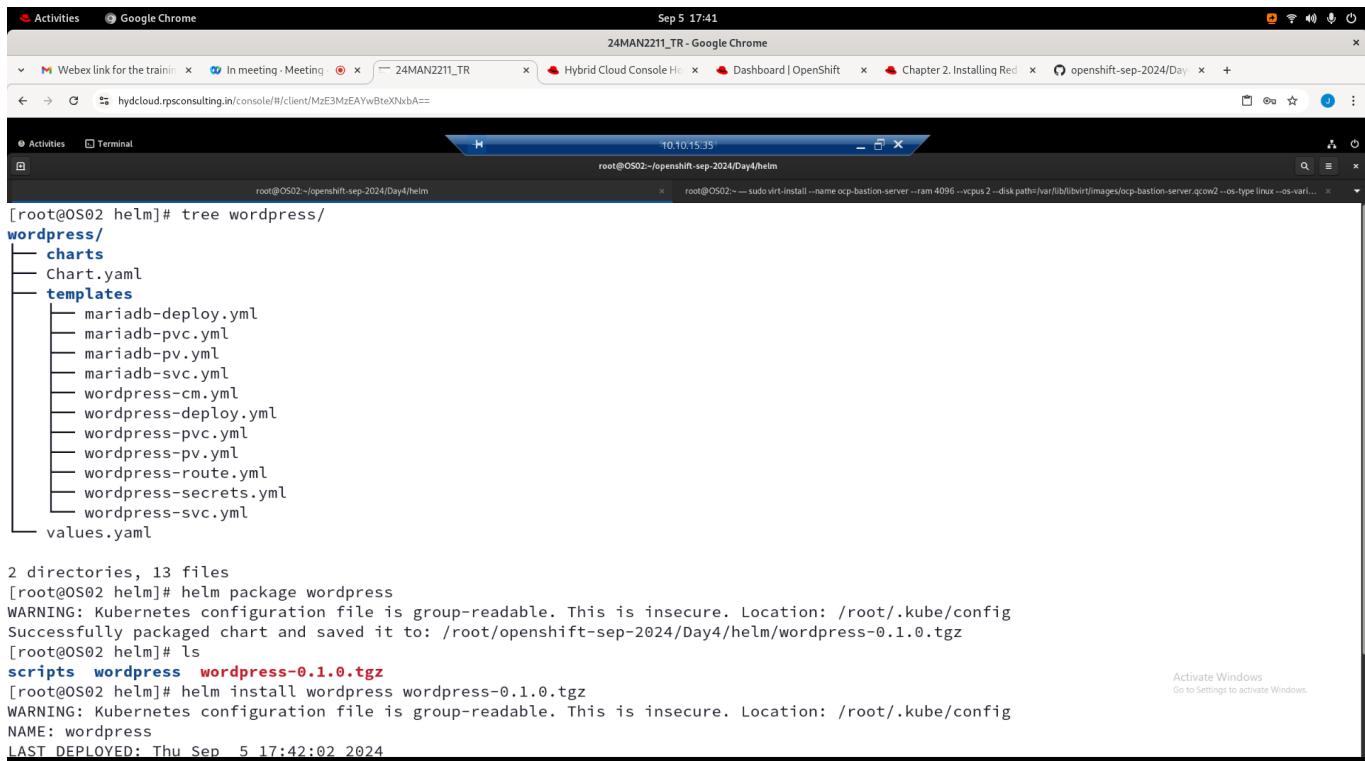


```

Activities Google Chrome Sep 5 17:33
Webex link for the trainin... In meeting - Meeting 24MAN2211_TR - Google Chrome
hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==
Activities Terminal 10:10:15:35
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-variant=redhat-7.9 --cpu host --graphics none --network network:br0 --location /var/www/html/wordpress --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-variant=redhat-7.9 --cpu host --graphics none --network network:br0 --location /var/www/html/wordpress
scripts
[root@OS02 helm]# tree wordpress/
wordpress/
├── charts
│   └── Chart.yaml
├── templates
│   ├── mariadb-deploy.yaml
│   ├── mariadb-pvc.yaml
│   ├── mariadb-pv.yaml
│   ├── mariadb-svc.yaml
│   ├── wordpress-cm.yaml
│   ├── wordpress-deploy.yaml
│   ├── wordpress-pvc.yaml
│   ├── wordpress-pv.yaml
│   ├── wordpress-route.yaml
│   ├── wordpress-secrets.yaml
│   └── wordpress-svc.yaml
└── values.yaml

2 directories, 13 files
[root@OS02 helm]# ls
scripts wordpress
[root@OS02 helm]# helm package wordpress
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
Successfully packaged chart and saved it to: /root/openshift-sep-2024/Day4/helm/wordpress-0.1.0.tgz
[root@OS02 helm]# ls
scripts wordpress wordpress-0.1.0.tgz
[root@OS02 helm]#

```

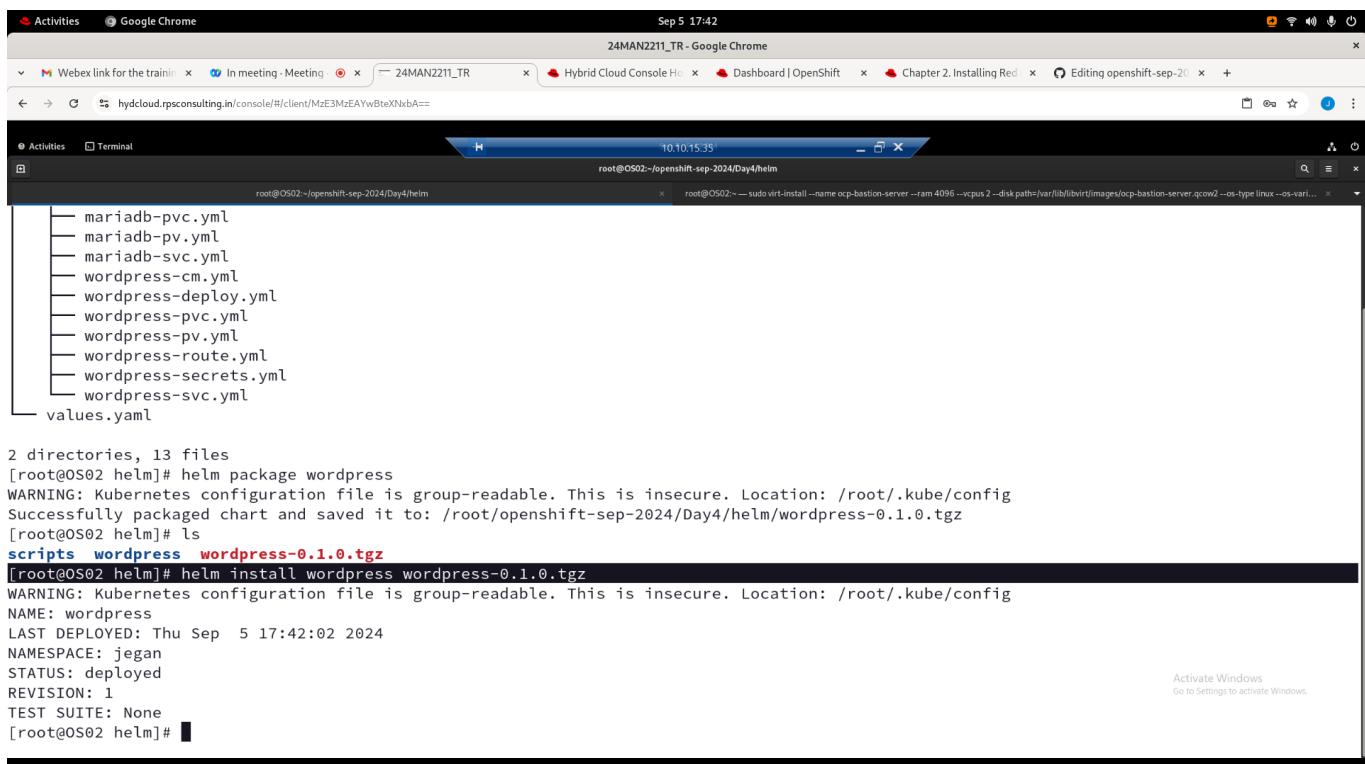


```
Sep 5 17:41
24MAN2211_TR - Google Chrome
Activities Google Chrome
Webex link for the trainin... In meeting - Meeting 24MAN2211_TR - Google Chrome
hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==

Activities Terminal
10.10.15.35
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari...
root@OS02 helm]# tree wordpress/
wordpress/
└── charts
    ├── Chart.yaml
    └── templates
        ├── mariadb-deploy.yml
        ├── mariadb-pvc.yml
        ├── mariadb-pv.yml
        ├── mariadb-svc.yml
        ├── wordpress-cm.yml
        ├── wordpress-deploy.yml
        ├── wordpress-pvc.yml
        ├── wordpress-pv.yml
        ├── wordpress-route.yml
        ├── wordpress-secrets.yml
        └── wordpress-svc.yml
values.yaml

2 directories, 13 files
[root@OS02 helm]# helm package wordpress
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
Successfully packaged chart and saved it to: /root/openshift-sep-2024/Day4/helm/wordpress-0.1.0.tgz
[root@OS02 helm]# ls
scripts wordpress wordpress-0.1.0.tgz
Activate Windows
Go to Settings to activate Windows.

[root@OS02 helm]# helm install wordpress wordpress-0.1.0.tgz
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
NAME: wordpress
LAST DEPLOYED: Thu Sep 5 17:42:02 2024
```



```
Sep 5 17:42
24MAN2211_TR - Google Chrome
Activities Google Chrome
Webex link for the trainin... In meeting - Meeting 24MAN2211_TR - Google Chrome
hydcloud.rpsconsulting.in/console/#/client/MzE3MzEAYwBteXNxbA==

Activities Terminal
10.10.15.35
root@OS02:~/openshift-sep-2024/Day4/helm
root@OS02:~ sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari...
root@OS02 helm]# tree wordpress/
wordpress/
└── charts
    ├── Chart.yaml
    └── templates
        ├── mariadb-pvc.yml
        ├── mariadb-pv.yml
        ├── mariadb-svc.yml
        ├── wordpress-cm.yml
        ├── wordpress-deploy.yml
        ├── wordpress-pvc.yml
        ├── wordpress-pv.yml
        ├── wordpress-route.yml
        ├── wordpress-secrets.yml
        └── wordpress-svc.yml
values.yaml

2 directories, 13 files
[root@OS02 helm]# helm package wordpress
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
Successfully packaged chart and saved it to: /root/openshift-sep-2024/Day4/helm/wordpress-0.1.0.tgz
[root@OS02 helm]# ls
scripts wordpress wordpress-0.1.0.tgz
Activate Windows
Go to Settings to activate Windows.

[root@OS02 helm]# helm install wordpress wordpress-0.1.0.tgz
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
NAME: wordpress
LAST DEPLOYED: Thu Sep 5 17:42:02 2024
NAMESPACE: jegan
STATUS: deployed
REVISION: 1
TEST SUITE: None
[root@OS02 helm]#
```

```
Activities Google Chrome Sep 5 17:42
Webex link for the traini... In meeting - Meeting 24MAN2211_TR Hybrid Cloud Console Ho... Dashboard | OpenShift Chapter 2. Installing Red... openshift-sep-2024/Day4... + Activities Terminal 10.10.15.35 root@OS02:~/openshift-sep-2024/Day4/helm root@OS02:~ root@OS02:~ -- sudo virt-install --name ocp-bastion-server --ram 4096 --vcpus 2 --disk path=/var/lib/libvirt/images/ocp-bastion-server.qcow2 --os-type linux --os-vari... x
wordpress-deploy.yml
wordpress-pvc.yml
wordpress-pv.yml
wordpress-route.yml
wordpress-secrets.yml
wordpress-svc.yml
values.yaml

2 directories, 13 files
[root@OS02 helm]# helm package wordpress
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
Successfully packaged chart and saved it to: /root/openshift-sep-2024/Day4/helm/wordpress-0.1.0.tgz
[root@OS02 helm]# ls
scripts wordpress wordpress-0.1.0.tgz
[root@OS02 helm]# helm install wordpress wordpress-0.1.0.tgz
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
NAME: wordpress
LAST DEPLOYED: Thu Sep 5 17:42:02 2024
NAMESPACE: jegan
STATUS: deployed
REVISION: 1
TEST SUITE: None
[root@OS02 helm]# helm list
WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config
NAME           NAMESPACE      REVISION      UPDATED             STATUS        CHART
wordpress      jegan          1            2024-09-05 17:42:02.124193235 +0530 IST  deployed    wordpress-0.1.0 1.16.0
[root@OS02 helm]#
```