

Day 5

Post Test Link

```
https://rpsconsulting116.examly.io/contest/public?  
U2FsdGVkX1%20lj2JvRF3%20bex%2Fua1NkTg4RWPsa3kSjTgsunMnwNZHAvmgEYl90cBn70z9g  
rc0QNwV933IJwQubQ=%3D
```

Feedback link

```
https://survey.zohopublic.com/zs/M50MeI
```

Creating a Docker Registry in your JFrog Artifactory Cloud environment

Login to your JFrog Artifactory Cloud account

The screenshot shows the JFrog Artifactory Cloud interface. The left sidebar has a dark theme with white text. It includes sections for Get Started, Artifactory, Packages, Builds, Artifacts, Release Lifecycle, Xray, Distribution, Pipelines, Integrations, and MyJFrog Portal. The main content area shows a package named "exposures" with a timestamp of 26-06-24 20:03:35 +0530 and the latest version. It also shows 1 version and 0 downloads. The top navigation bar shows the date as Jun 28 10:18 and includes links for Activities, Google Chrome, and various system icons.

Switch to Administration Tab

A screenshot of a Google Chrome browser window displaying the JFrog Platform interface. The title bar shows 'JFrog - Google Chrome'. The address bar contains the URL 'https://tektutorjegan74.jfrog.io/ui/admin/projects/list'. The top navigation bar includes tabs for 'All Projects' (selected), 'Application', and 'Administration'. A search bar for 'Admin Resources' is also present. On the left, a sidebar menu lists various administrative sections: Projects, Environments, Repositories, User Management, User Authentication, Platform Security, General Management, Platform Monitoring, Topology, Artifactory Settings, Xray Settings, and Workers (Beta). The main content area displays a 'Welcome to Projects' dialog box. This dialog features a central image of three stacked green cubes, a title 'Welcome to Projects', a descriptive text about setting up projects for resources, and a 'Next' button at the bottom.

Click on Repositories on the left side menu

A screenshot of a Google Chrome browser window displaying the JFrog Platform interface. The title bar shows 'JFrog - Google Chrome'. The address bar contains the URL 'https://tektutorjegan74.jfrog.io/ui/admin/repositories/virtual'. The top navigation bar includes tabs for 'All Projects' (selected), 'Application', and 'Administration'. A search bar for 'Admin Resources' is also present. On the left, a sidebar menu lists various administrative sections, with 'Repositories' highlighted in green. The main content area displays a 'Repositories' section. It includes tabs for 'Virtual' (selected), Local, Remote, and Federated. Below this, a message states '0 Repositories' and 'No results were found'. A note says 'Try to change your search'. At the bottom right, there are navigation arrows and a link to 'https://tektutorjegan74.jfrog.io/ui/admin/repositories/virtual'.

Click on Create Repository menu on the top right corner

The screenshot shows the JFrog Platform interface in Google Chrome. The main navigation bar includes 'JFrog Platform', 'All Projects', 'Application', and 'Administration'. The 'Administration' tab is selected. On the left, a sidebar lists various management options like 'Projects', 'Environments', 'Repositories', and 'Workers'. The main content area is titled 'Repositories' and shows a table with columns: Repository Key, Type, Project, Environment, Selected Repositories, Shared With, and Actions. A message 'No results were found' is displayed. In the top right corner of the main content area, there is a green button labeled 'Create a Repository'.

Click on Pre-built Setup

The screenshot shows the JFrog Platform interface in Google Chrome, similar to the previous one but with a modal window open. The modal is titled 'Create Repositories' and contains the sub-instruction 'Select the package type you want - we'll create the default repositories for you!'. Below this, there is a search bar labeled 'Search by package type' and a grid of icons representing different repository types. The grid includes: Alpine, Bower, Cargo, Chef, CocoaPods, Composer, Conan, Conda, CRAN, Debian, Docker, Gems, Generic, GitLfs, Go, Gradle, Helm, HelmOCI, HuggingFace ML, Ivy, Maven, Npm, NuGet, OCI, Opkg, Pub, Puppet, PyPi, SBT, Swift, Terraform, and Vagrant. The background of the main interface is dimmed to indicate it is not active while the modal is open.

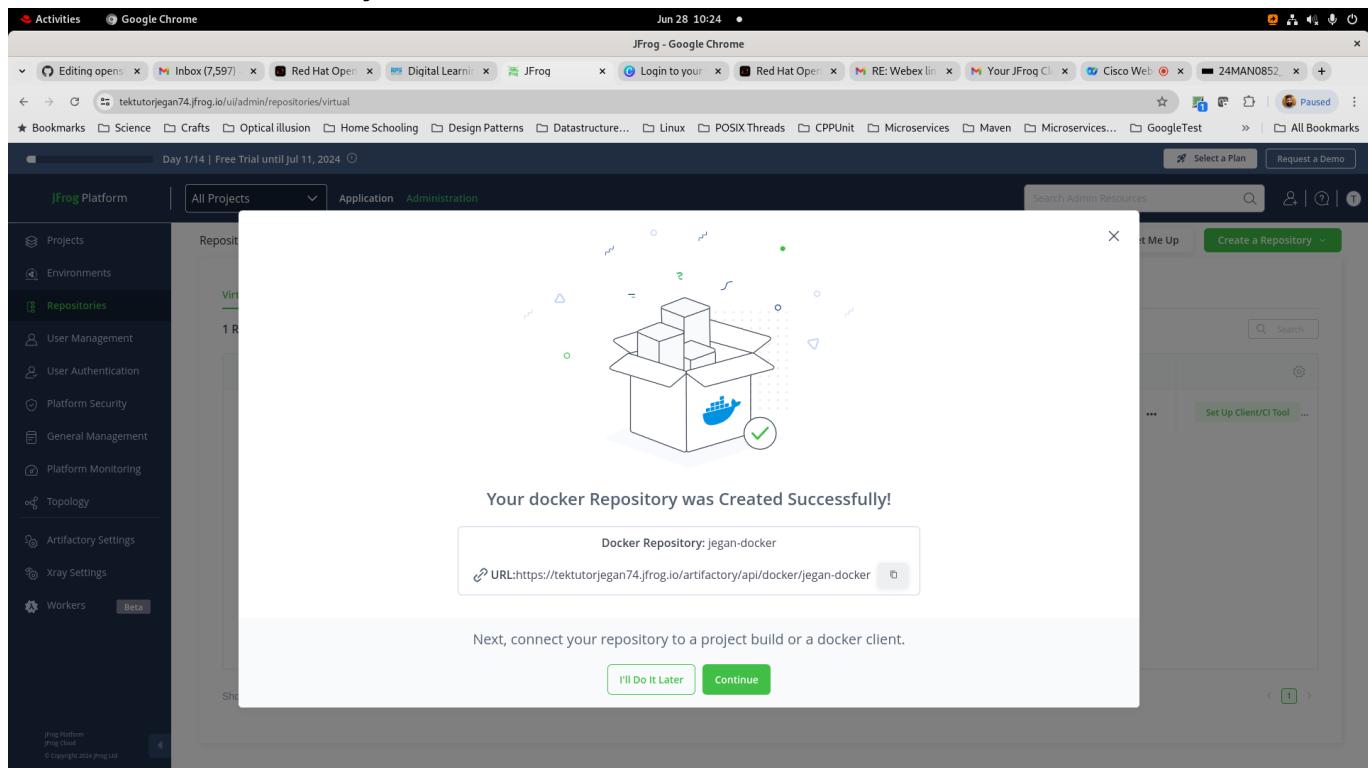
Select Docker

The screenshot shows the JFrog Platform interface. On the left, there's a sidebar with various navigation options like Projects, Environments, and Repositories. The main area is titled 'Repositories' and shows a 'Virtual' tab selected. A modal window titled 'Create Repositories' is open, prompting the user to 'Assign a name to your new repositories by adding a meaningful prefix identifier'. In the 'Repositories prefix' input field, the user has typed 'jegan'. Below this, there are three repository types listed: 'Local Repository' (labeled 'docker-local'), 'Remote Repository' (labeled 'https://registry-1.docker.io/docker-remote'), and 'Virtual Repository' (labeled 'docker'). At the bottom right of the modal is a 'Create' button.

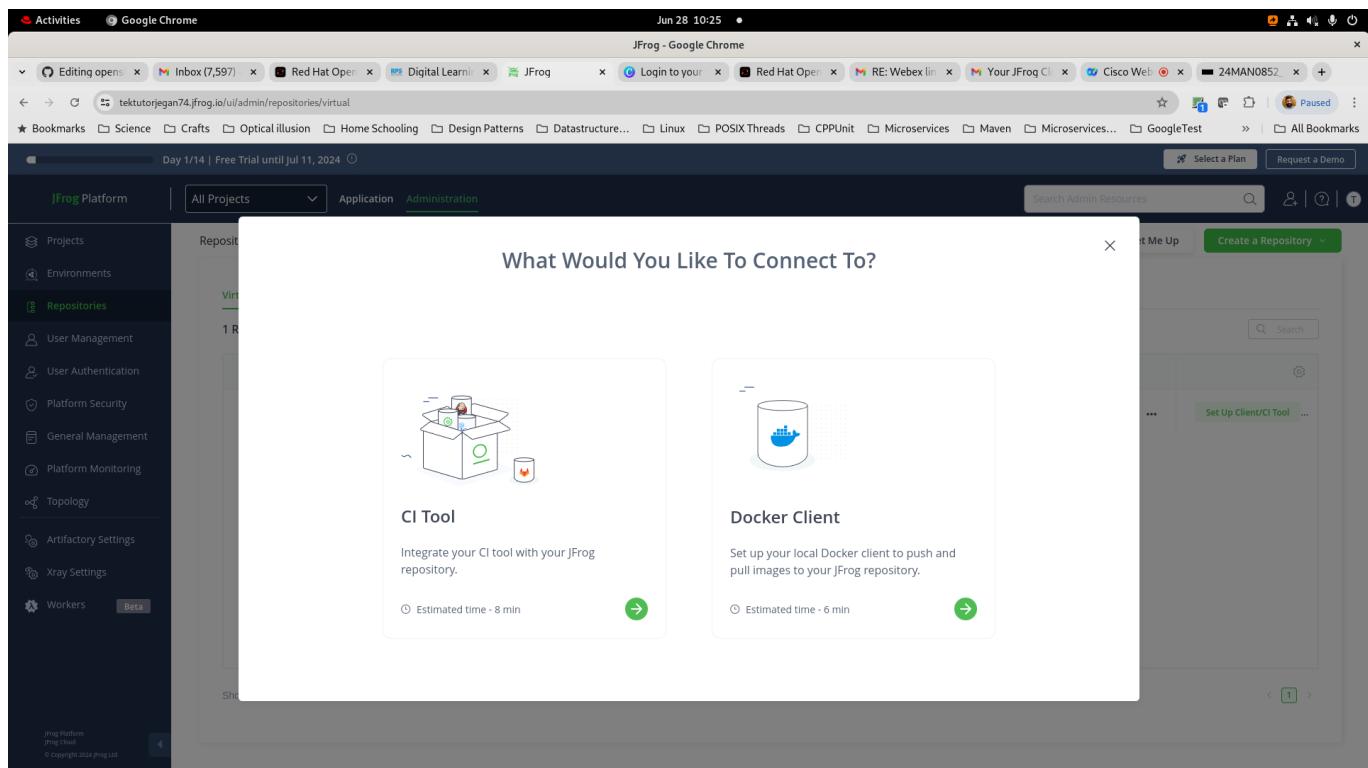
Type your name and click on create button

This screenshot is identical to the one above, but it shows the 'Repositories prefix' input field containing the value 'jegan'. The rest of the interface, including the modal window and the list of repository types, remains the same.

Click Continue button when you see a screen as shown in the screenshot below



Need to click on "Docker Client" Green arrow



Copy the instructions shown below in some text file and save it for your reference, click on Generate

The screenshot shows the JFrog Platform interface with the 'Administration' tab selected. On the left, a sidebar lists various management options like User Management, Artifactory Settings, and Xray Settings. The main area is titled 'Repositories' and shows a 'Virtual' tab selected, displaying one repository named 'jegan-docker'. A modal window titled 'Set Up Your Docker Client' provides step-by-step instructions for setting up a Docker client. Step 1: 'Login' - It says 'Run this Docker command in your terminal to authenticate' followed by a command: `docker login -u tektutorjegan74@gmail.com tektutorjegan74.jfrog.io`. There is a 'Copy' button next to the command. Step 2: 'Pull' - It says 'When asked for a password, enter your identity token' followed by a 'Generate' button. Step 3: 'Push' - This part is partially visible. At the bottom of the modal, there is a 'Waiting for your action' message and a 'Next' button.

Copy the token and save it in the same text file where you saved the previous instructions. Now, we need to execute the docker login instructions shown in the screen on your linux terminal using the token as the password

```
jegan@tektutor.org ➜ docker login -u jegan74@gmail.com tektutorjegan74.jfrog.io
Password:
WARNING! Your password will be stored unencrypted in /home/jegan/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
jegan@tektutor.org ➜
```

Activities Google Chrome Jun 28 10:32 • JFrog - Google Chrome

Editing opens... Inbox (7,597) Red Hat Open... Digital Learn... JFrog Login to your... Red Hat Open... RE: Webex lin... Your JFrog Cl... Cisco Web... 24MAN0852...

All Projects Application Administration

Search Admin Resources

Set Me Up Create a Repository

Projects Environments Repositories User Management User Authentication Platform Security General Management Platform Monitoring Topology Artifactory Settings Xray Settings Workers Beta

Repositories

Virtual Local Remote Federated

1 Repositories

Repository Key Type

jegan-docker Docker

Showing 1 - 1 from 1 items

1 Login 2 Pull 3 Push

Set Up Your Docker Client

Let's walk through the steps to push and pull your images

You are logged in
Click on Next to continue

Next

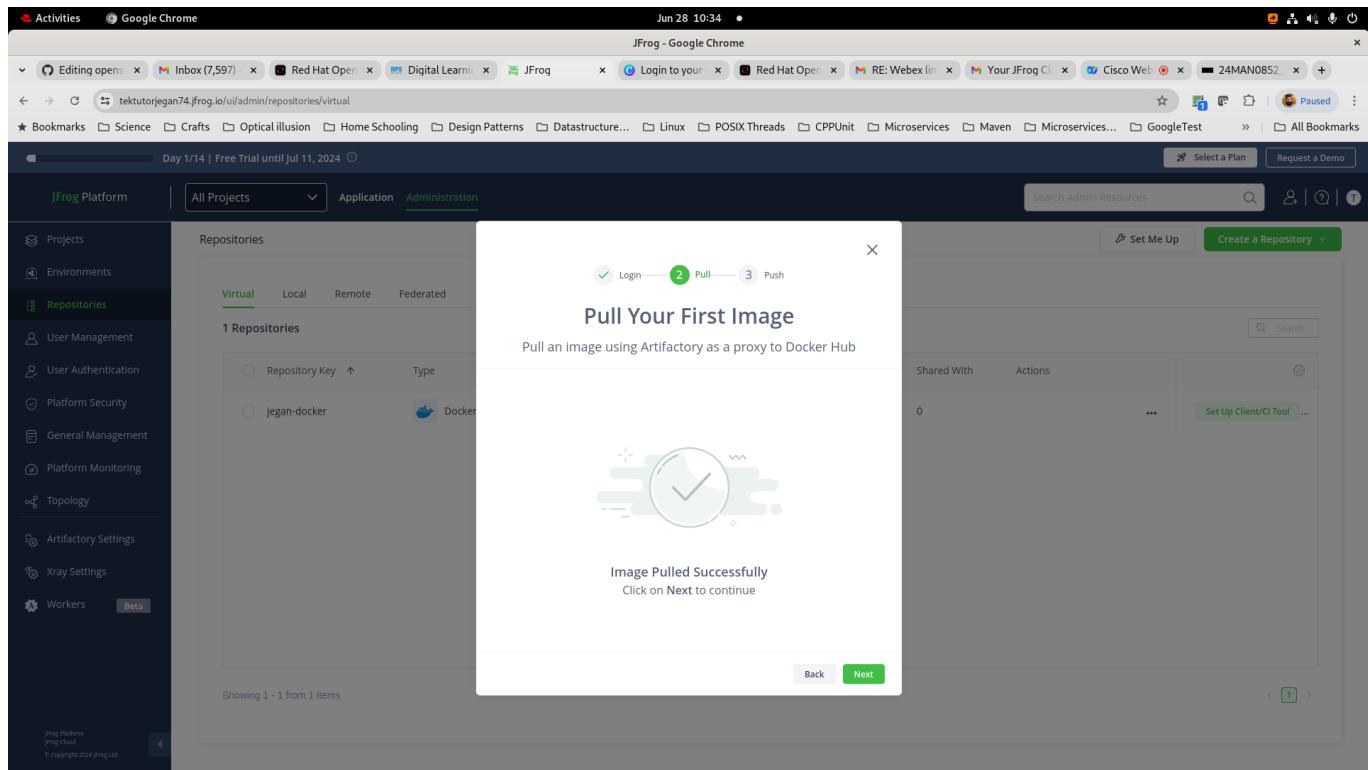
Click on Next button, copy the instruction shown below save it in text file and execute the instruction on your linux terminal

The screenshot shows the JFrog Artifactory interface. A modal window titled "Pull Your First Image" is open, instructing the user to pull an image using Artifactory as a proxy to Docker Hub. It provides a command: `docker pull tektutorjegan74.jfrog.io/jegan-docker/hello-world:latest`. Below the command, there's a "Copy" button and a status message "Waiting for your action". The background shows the "Repositories" section with one virtual repository named "jegan-docker".

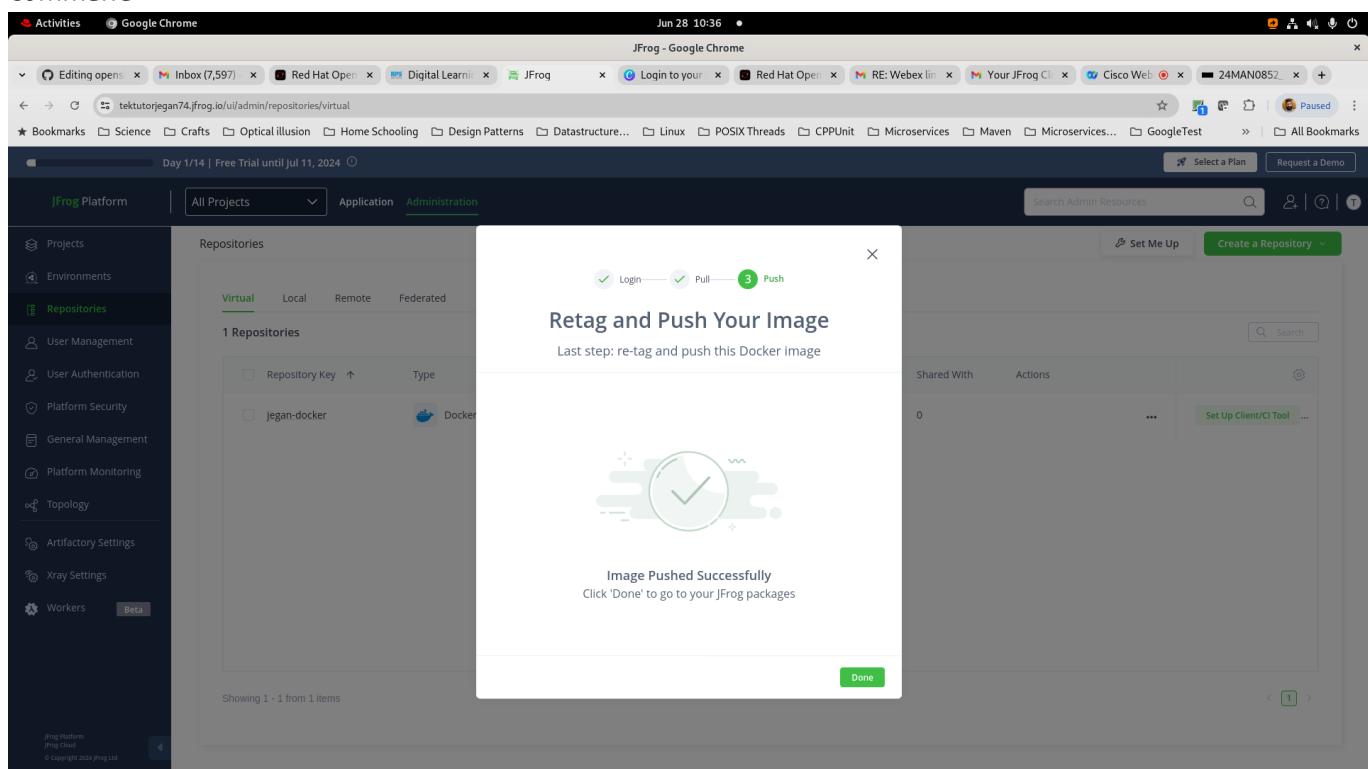
The terminal window below shows the execution of the command:

```
jegan@tektutor.org ➔ docker login -u jegan74@gmail.com tektutorjegan74.jfrog.io
Password:
WARNING! Your password will be stored unencrypted in /home/jegan/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
jegan@tektutor.org ➔ docker pull tektutorjegan74.jfrog.io/jegan-docker/hello-world:latest
latest: Pulling from jegan-docker/hello-world
Digest: sha256:94323f3e5e09a8b9515d74337010375a456c909543e1ff1538f5116d38ab3989
Status: Downloaded newer image for tektutorjegan74.jfrog.io/jegan-docker/hello-world:latest
tektutorjegan74.jfrog.io/jegan-docker/hello-world:latest
jegan@tektutor.org ➔
```



Click on Next button, copy the instruction in the text file and execute the pull and tag followed by push command



Click on Done and you will see the below page. You will notice

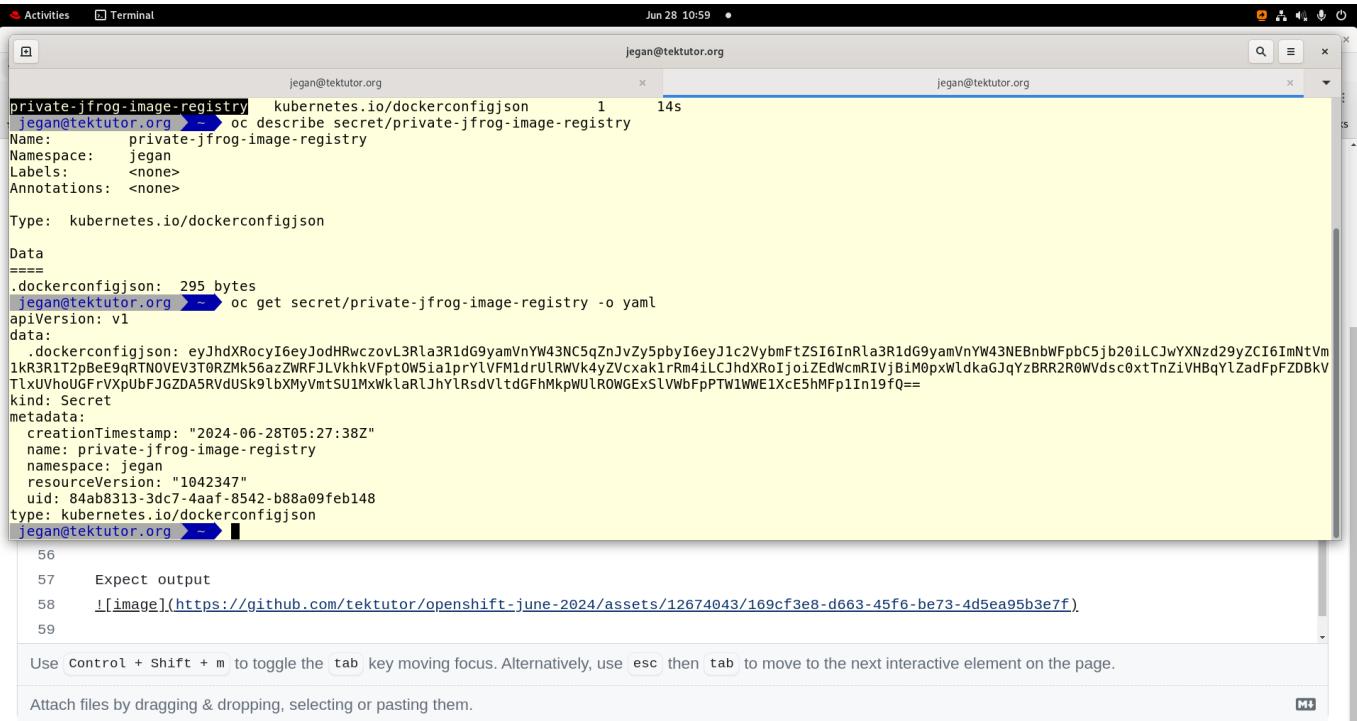
The screenshot shows the JFrog Artifactory web interface. On the left, there's a sidebar with navigation links like 'Get Started', 'Artifactory', 'Builds', 'Artifacts', 'Release Lifecycle', 'Xray', 'Distribution', 'Pipelines', 'Integrations', and 'MyJFrog Portal'. The main area displays a tree view of artifacts under 'jfrog/hello-world/1.0.0'. A specific file, 'manifest.json', is selected. The right panel shows detailed information about this file, including its name, repository path, file URL, source path, size, creation and modification dates, package information, dependency declaration, and virtual repository associations. At the bottom, there are buttons for 'Trash Can' and 'Done'.

Lab - Build custom image with our application binary and push the image to JFrog Artifactory Docker Registry

Openshift need to login to JFrog Artifactory server in order to push the custom docker images, hence let's create a secret to store the JFrog Login Credentials in Openshift

```
oc create secret docker-registry private-jfrog-image-registry --docker-server=https://<your-jfrog-id>.jfrog.io --docker-username=<your-jfrog-registered-email> --docker-password=<your-jfrog-token>
secret/private-jfrog-image-registry created
```

Expected output



```
jegan@tektutor.org ~ [1] Jun 28 10:59 •
jegan@tektutor.org ~ [1] Jun 28 10:59 •
jegan@tektutor.org ~ [1] Jun 28 10:59 •

private-jfrog-image-registry kubernetes.io/dockerconfigjson 1 14s
jegan@tektutor.org ~ -> oc describe secret/private-jfrog-image-registry
Name: private-jfrog-image-registry
Namespace: jegan
Labels: <none>
Annotations: <none>

Type: kubernetes.io/dockerconfigjson

Data
====

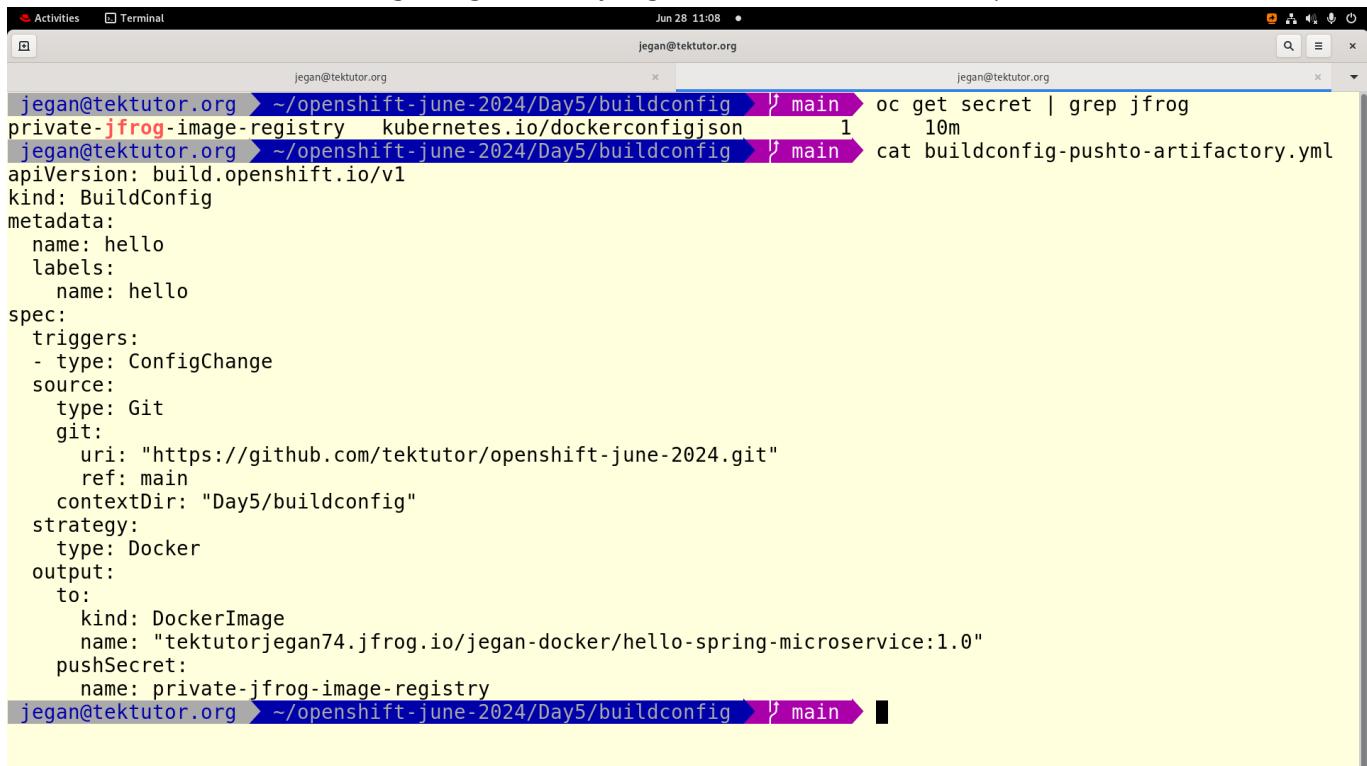
.dockerconfigjson: 295 bytes
jegan@tektutor.org ~ -> oc get secret/private-jfrog-image-registry -o yaml
apiVersion: v1
data:
  .dockerconfigjson: eyJhdXRoYI6eyJodHRwczovL3Rla3R1dG9yamVnYW43NC5qZnJvZy5pbIY6eyJ1c2VybmtZSI6InRla3R1dG9yamVnYW43NEBnbWFpbC5jb20iLCJwYXNzd29yZCI6ImNtVm1kR3R1T2pBeE9qRTNOVEV3T0RZMK56azZWRFJLvhkVFpt0W5ia1prYLVFM1drUlRWk4yZVcxak1rRm4iLCJhdXRoijoizEdWcmRIVjBiM0pxWldkaGJqYzBRR2R0WVdsc0xtTnZiVHBqylZadFpZDBKVTLxUVhuGFrVxpUbFJGZDASRVdUSk9lbXMyVmSU1MxWklaR1jhYlRsdsVldgFhHmkpWULROWGEsS1VwbFpPTW1WWElXcE5hMFplIn19fQ==

kind: Secret
metadata:
  creationTimestamp: "2024-06-28T05:27:38Z"
  name: private-jfrog-image-registry
  namespace: jegan
  resourceVersion: "1042347"
  uid: 84abb313-3dc7-4aae-8542-b88a09feb148
type: kubernetes.io/dockerconfigjson
jegan@tektutor.org ~ -> [1] 56
57   Expect output
58   \[image\]\(https://github.com/tektutor/openshift-june-2024/assets/12674043/169cf3e8-d663-45f6-be73-4d5ea95b3e7f\)
59

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Attach files by dragging & dropping, selecting or pasting them.
```

We need to create a buildconfig along with the jfrog credentials in the form of pushsecret as shown below



```
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •

jegan@tektutor.org ~ /~/openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
jegan@tektutor.org ~ /~/openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
jegan@tektutor.org ~ /~/openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •

private-jfrog-image-registry kubernetes.io/dockerconfigjson 1 10m
jegan@tektutor.org ~ /~/openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
cat buildconfig-pushto-artifactory.yml
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: hello
  labels:
    name: hello
spec:
  triggers:
  - type: ConfigChange
  source:
    type: Git
    git:
      uri: "https://github.com/tektutor/openshift-june-2024.git"
      ref: main
      contextDir: "Day5/buildconfig"
  strategy:
    type: Docker
  output:
    to:
      kind: DockerImage
      name: "tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0"
    pushSecret:
      name: private-jfrog-image-registry
jegan@tektutor.org ~ /~/openshift-june-2024/Day5/buildconfig ~ [1] Jun 28 11:08 •
```

```
Activities Terminal Jun 28 11:14 • jegan@tektutor.org
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ pwd
/home/jegan/openshift-june-2024/Day5/buildconfig
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ oc get secret | grep jfrog
private-jfrog-image-registry kubernetes.io/dockerconfigjson 1 2m1s
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ oc apply -f buildconfig-pushto-artifactory.yml
buildconfig.build.openshift.io/hello created
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ oc get buildconfigs
NAME      TYPE    FROM          LATEST
hello     Docker   Git@main:1
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ oc logs -f bc/hello
Cloning "https://github.com/tektutor/openshift-june-2024.git" ...
Commit: e83734fa809cc8efba0404699f06c57c65efb8dd (Merge branch 'main' of https://github.com/tektutor/openshift-june-2024)
Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
Date:   Fri Jun 28 11:13:02 2024 +0530
time="2024-06-28T05:44:04Z" 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"
I0628 05:44:04.175049       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 sha256:6c215442f70bd949a6f2e8092549943905e2d4f9c87a4f532d7740ae8647d33a
Copying blob sha256:48c2faf66abec3dce9f54d6722ff592fce6dd4fb58a0d0b72282936c6598a3b3
Copying blob sha256:234b70d0479d7f16d7ee8d04e4ffdacc57d7d14313faf59d332f18b2e9418743
```

```
Activities Terminal Jun 28 11:14 • jegan@tektutor.org
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ✘ main ➤ 2.4.2/spring-boot-dependencies-2.4.2.pom (108 kB at 1.7 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/datastax/oss/java-driver-bom/4.9.0/java-driver-bom-4.9.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/datastax/oss/java-driver-bom/4.9.0/java-driver-bom-4.9.0.pom (4.1 kB at 99 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-bom/4.1.17/metrics-bom-4.1.17.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-bom/4.1.17/metrics-bom-4.1.17.pom (5.3 kB at 161 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-parent/4.1.17/metrics-parent-4.1.17.pom
Downloaded from central: https://repo.maven.apache.org/maven2/io/dropwizard/metrics/metrics-parent/4.1.17/metrics-parent-4.1.17.pom (17 kB at 487 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy-bom/2.5.14/groovy-bom-2.5.14.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/groovy/groovy-bom/2.5.14/groovy-bom-2.5.14.pom (26 kB at 798 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-bom/11.0.8.Final/infinispan-bom-11.0.8.Final.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-bom/11.0.8.Final/infinispan-bom-11.0.8.Final.pom (19 kB at 538 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-build-configuration-parent/11.0.8.Final/infinispan-build-configuration-parent-11.0.8.Final.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/infinispan/infinispan-build-configuration-parent/11.0.8.Final/infinispan-build-configuration-parent-11.0.8.Final.pom (13 kB at 416 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jboss/jboss-parent/36/jboss-parent-36.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jboss/jboss-parent/36/jboss-parent-36.pom (66 kB at 1.6 MB/s)
```

```
Activities Terminal Jun 28 11:14 • jegan@tektutor.org
jegan@tektutor.org x jegan@tektutor.org x
2.4.2/spring-boot-loader-tools-2.4.2.jar (243 kB at 817 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.inject/0.3.4/org.eclipse.sisu.inject-0.3.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.11/commons-codec-1.1.jar (335 kB at 1.1 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.2.1/plexus-utils-3.2.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/3.6.3/maven-plugin-api-3.6.3.jar (47 kB at 149 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-classworlds/2.6.0/plexus-classworlds-2.6.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.1.0/maven-shared-utils-3.1.0.jar (164 kB at 516 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.plexus/0.3.4/org.eclipse.sisu.plexus-0.3.4.jar (205 kB at 597 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-classworlds/2.6.0/plexus-classworlds-2.6.0.jar (53 kB at 151 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.2.1/plexus-utils-3.2.1.jar (262 kB at 742 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.inject/0.3.4/org.eclipse.sisu.inject-0.3.4.jar (379 kB at 1.1 MB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 16.999 s
[INFO] Finished at: 2024-06-28T05:44:57Z
[INFO] -----
```



```
Activities Terminal Jun 28 11:16 • jegan@tektutor.org
jegan@tektutor.org x jegan@tektutor.org x
--> 4f4b0196bcfa
[2/2] STEP 4/6: ENTRYPOINT ["java", "-jar", "app.jar"]
--> 8b66215a6e28
[2/2] STEP 5/6: ENV "OPENSHIFT_BUILD_NAME"="hello-1" "OPENSHIFT_BUILD_NAMESPACE"="jegan" "OPENSHIFT_BUILD_SOURCE"="https://github.com/tektutor/openshift-june-2024.git" "OPENSHIFT_BUILD_REFERENCE"="main" "OPENSHIFT_BUILD_COMMIT"="e83734fa809cc8efba0404699f06c57c65efb8dd"
--> d66df0a2e8c3
[2/2] STEP 6/6: LABEL "io.openshift.build.commit.author"="Jeganathan Swaminathan <mail2jegan@gmail.com>" "io.openshift.build.commit.date"="Fri Jun 28 11:13:02 2024 +0530" "io.openshift.build.commit.id"="e83734fa809cc8efba0404699f06c57c65efb8dd" "io.openshift.build.commit.message"="Merge branch 'main' of https://github.com/tektutor/openshift-june-2024" "io.openshift.build.commit.ref"="main" "io.openshift.build.name"="hello-1" "io.openshift.build.namespace"="jegan" "io.openshift.build.source-context-dir"="Day5/buildconfig" "io.openshift.build.source-location"="https://github.com/tektutor/openshift-june-2024.git"
[2/2] COMMIT temp.builder.openshift.io/jegan/hello-1:d6c939f1
--> e106e9dbe6a2
Successfully tagged temp.builder.openshift.io/jegan/hello-1:d6c939f1
e106e9dbe6a289e70d4de7a8889d3309c12c5c4f73f3ed2caa649effba4dc6a66

Pushing image tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0 ...
Getting image source signatures
Copying blob sha256:130154cd4ddc70332b820fb247f30649545ba019219a0a2810ddf58743207ed
Copying blob sha256:8393debac2ec316f87eab40af5451d57df2863f844f545d40447e46e025c9663
Copying blob sha256:1846180440d3041e0848b0bcbb735f024f3bc5e2d39f1ab4ad490f630863d903
Copying config sha256:e106e9dbe6a289e70d4de7a8889d3309c12c5c4f73f3ed2caa649effba4dc6a66
Writing manifest to image destination
Successfully pushed tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice@sha256:44d1ea02dbc51cf09ee3dc86815eeecf3432fea06c9fa8a5d4823ae3c34230c
Push successful
jegan@tektutor.org ~ /openshift-june-2024/Day5/buildconfig ➤ main ➤
```

The screenshot shows the JFrog Artifactory web interface. On the left, there's a sidebar with various navigation options like 'Get Started', 'Artifactory', 'Builds', 'Artifacts', 'Release Lifecycle', 'Xray', 'Distribution', 'Pipelines', 'Integrations', and 'MyFrog Portal'. The main area displays a repository structure under 'jegan-docker'. A specific folder 'hello-spring-microservice' is expanded, showing a subfolder '1.0' which contains a file named 'manifest.json'. Other files in '1.0' include several SHA-256 checksums for different versions of the image. To the right of the repository tree, there's a detailed view of the '1.0' version, including tabs for 'General', 'Info', 'Effective Permissions', 'Properties', and 'Followers'. The 'Info' tab shows the name '1.0', repository path 'jegan-docker/hello-spring-microservice/1.0', file URL, creation date (28-06-24 05:46:31 +00:00), package information, dependency declaration, virtual repository associations, included repositories, and checksums.

Lab - Deploying hello microservice using our custom docker image from Private JFrog Docker Registry

```
cd ~/openshift-june-2024
git pull
cd Day5/buildconfig
oc create deployment hello --image=tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0 --replicas=3 -o yaml --dry-run=client

oc create deployment hello --image=tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0 --replicas=3 -o yaml --dry-run=client
> hello-deploy.yml
```

Update the hello-deploy to use your private-jfrog-image-registry secret as JFrog Artifactory will allow only authorized users to download and use the image from it.

```
cd ~/openshift-june-2024
git pull
cd Day5/buildconfig
cat hello-deploy.yml
oc apply -f hello-deploy.yml
oc get deploy,po
oc get po -w
oc get po
```

Expected output

```

Activities Terminal Jun 28 12:02 •
jegan@tektutor.org
jegan@tektutor.org
- image: tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0
  name: hello
imagePullSecrets:
- name: private-jfrog-image-registry
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig > main > oc apply -f hello-deploy.yml
deployment.apps/hello created
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig > main > oc get deploy,po
NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/hello 0/3 3 0 3s
NAME READY STATUS RESTARTS AGE
pod/hello-1-build 0/1 Completed 0 47m
pod/hello-2-build 0/1 Completed 0 17m
pod/hello-688bb4b5f8-54s97 0/1 ContainerCreating 0 3s
pod/hello-688bb4b5f8-97472 0/1 ContainerCreating 0 3s
pod/hello-688bb4b5f8-cbqms 0/1 ContainerCreating 0 3s
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig > main > oc get po -w
NAME READY STATUS RESTARTS AGE
hello-1-build 0/1 Completed 0 47m
hello-2-build 0/1 Completed 0 17m
hello-688bb4b5f8-54s97 0/1 ContainerCreating 0 6s
hello-688bb4b5f8-97472 0/1 ContainerCreating 0 6s
hello-688bb4b5f8-cbqms 0/1 ContainerCreating 0 6s
hello-688bb4b5f8-cbqms 1/1 Running 0 26s
hello-688bb4b5f8-97472 1/1 Running 0 29s
hello-688bb4b5f8-54s97 1/1 Running 0 33s
^C
~/openshift-june-2024/Day5/buildconfig > main > oc get po
NAME READY STATUS RESTARTS AGE
hello-1-build 0/1 Completed 0 48m
hello-2-build 0/1 Completed 0 18m
hello-688bb4b5f8-54s97 1/1 Running 0 48s
hello-688bb4b5f8-97472 1/1 Running 0 48s
hello-688bb4b5f8-cbqms 1/1 Running 0 48s
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig > main >

```

Let's create an internal service for hello microservice

```

oc expose deploy/hello --type=ClusterIP --port=8080 -o yaml --dry-run=client
oc expose deploy/hello --type=ClusterIP --port=8080 -o yaml --dry-run=client > hello-svc.yml
oc apply -f hello-svc.yml
oc get svc
oc describe svc/hello

```

Expected output

The screenshot shows a terminal window with two tabs. The left tab is titled 'jegan@tektutor.org' and the right tab is also titled 'jegan@tektutor.org'. The terminal content is as follows:

```

Activities Terminal Jun 28 12:04 •
jegan@tektutor.org
jegan@tektutor.org
hello-688bb4b5f8-54s97 1/1 Running 0 33s
~ et po
NAME READY STATUS RESTARTS AGE
hello-1-build 0/1 Completed 0 48m
hello-2-build 0/1 Completed 0 18m
hello-688bb4b5f8-54s97 1/1 Running 0 48s
hello-688bb4b5f8-97472 1/1 Running 0 48s
hello-688bb4b5f8-cbqms 1/1 Running 0 48s
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤ oc expose deploy/hello --type=ClusterIP --port=8080 -o yaml --dry-run=client
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: hello
    name: hello
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    app: hello
    type: ClusterIP
status:
  loadBalancer: {}
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤ oc expose deploy/hello --type=ClusterIP --port=8080 -o yaml --dry-run=client > hello-svc.yaml
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤ oc apply -f hello-svc.yaml
service/Hello created
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤ oc get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
Hello ClusterIP 172.30.251.87 <none> 8080/TCP 2s
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤ oc describe svc/Hello
Name: Hello
Namespace: jegan
Labels: app=hello
Annotations: <none>
Selector: app=hello
Type: ClusterIP
IP Family Policy: SingleStack
IP Families: IP
IP: 172.30.251.87
IPs: 172.30.251.87
Port: <unset> 8080/TCP
TargetPort: 8080/TCP
Endpoints: 10.128.2.30:8080,10.130.0.107:8080,10.131.0.11:8080
Session Affinity: None
Events: <none>
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig ↵ main ➤

```

Let's create a route to access the hello microservice from outside the openshift cluster

```

oc expose svc/Hello -o yaml --dry-run=client
oc expose svc/Hello -o yaml --dry-run=client > hello-route.yaml

cat hello-route.yaml

oc apply -f hello-route.yaml
oc get route
oc describe route/Hello

```

Expected output

```

Activities Terminal Jun 28 12:08 • jegan@tektutor.org
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ cat hello-route.yml
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  creationTimestamp: null
  labels:
    app: hello
    name: hello
spec:
  port:
    targetPort: 8080
  to:
    kind: ""
    name: hello
    weight: null
status: {}
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ oc apply -f hello-route.yml
route.route.openshift.io/hello created
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ oc get route
NAME      HOST/PORT          PATH  SERVICES   PORT  TERMINATION   WILDCARD
hello     hello-jegan.apps.ocp4.tektutor.org.labs  hello       8080        None
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ oc describe route/hello
Name:           hello
Namespace:      jegan
Created:        10 seconds ago
Labels:         app=hello
Annotations:    kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"route.openshift.io/v1","kind":"Route","metadata":{"annotations":{},"creationTimestamp":null,"labels":{"app":"hello","name":"hello","namespace":"jegan"},"spec":{"port":{"targetPort":8080}, "to":{"kind":"","name":"hello","weight":null}},"status":{}}
               openshift.io/host.generated=true
Requested Host: hello-jegan.apps.ocp4.tektutor.org.labs
                  exposed on router default (host router-default.apps.ocp4.tektutor.org.labs) 10 seconds ago
Path:             <none>
TLS Termination: <none>
Insecure Policy: <none>
Endpoint Port:  8080

Service:        hello
Weight:         100 (100%)
Endpoints:     10.128.2.30:8080, 10.130.0.107:8080, 10.131.0.11:8080
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔

Activities Terminal Jun 28 12:08 • jegan@tektutor.org
jegan@tektutor.org > route.route.openshift.io/hello created
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ oc get route
NAME      HOST/PORT          PATH  SERVICES   PORT  TERMINATION   WILDCARD
hello     hello-jegan.apps.ocp4.tektutor.org.labs  hello       8080        None
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ oc describe route/hello
Name:           hello
Namespace:      jegan
Created:        10 seconds ago
Labels:         app=hello
Annotations:    kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"route.openshift.io/v1","kind":"Route","metadata":{"annotations":{},"creationTimestamp":null,"labels":{"app":"hello","name":"hello","namespace":"jegan"},"spec":{"port":{"targetPort":8080}, "to":{"kind":"","name":"hello","weight":null}},"status":{}}
               openshift.io/host.generated=true
Requested Host: hello-jegan.apps.ocp4.tektutor.org.labs
                  exposed on router default (host router-default.apps.ocp4.tektutor.org.labs) 10 seconds ago
Path:             <none>
TLS Termination: <none>
Insecure Policy: <none>
Endpoint Port:  8080

Service:        hello
Weight:         100 (100%)
Endpoints:     10.128.2.30:8080, 10.130.0.107:8080, 10.131.0.11:8080
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔ curl http://hello-jegan.apps.ocp4.tektutor.org.labs
Hello Microservice!
jegan@tektutor.org > ~/openshift-june-2024/Day5/buildconfig ✘ main ➔

```

The screenshot shows the Red Hat OpenShift web interface. The top navigation bar indicates it's a 'Topology - Red Hat OpenShift - Google Chrome' session from 'Jun 28 12:09'. The left sidebar has a 'Developer' section with options like '+Add', 'Topology', 'Observe', 'Search', 'Builds', 'Helm', 'Project', 'ConfigMaps', and 'Secrets'. The main content area is titled 'Topology' and shows a single application named 'hello' under the 'jegan' project. The application icon is a blue circle with a white square inside, which contains a red 'H' and two white diagonal lines. Below the icon is a button labeled 'hello'.

Hello Microservice !

Lab - Rolling update - updating hello microservice to its v2.0

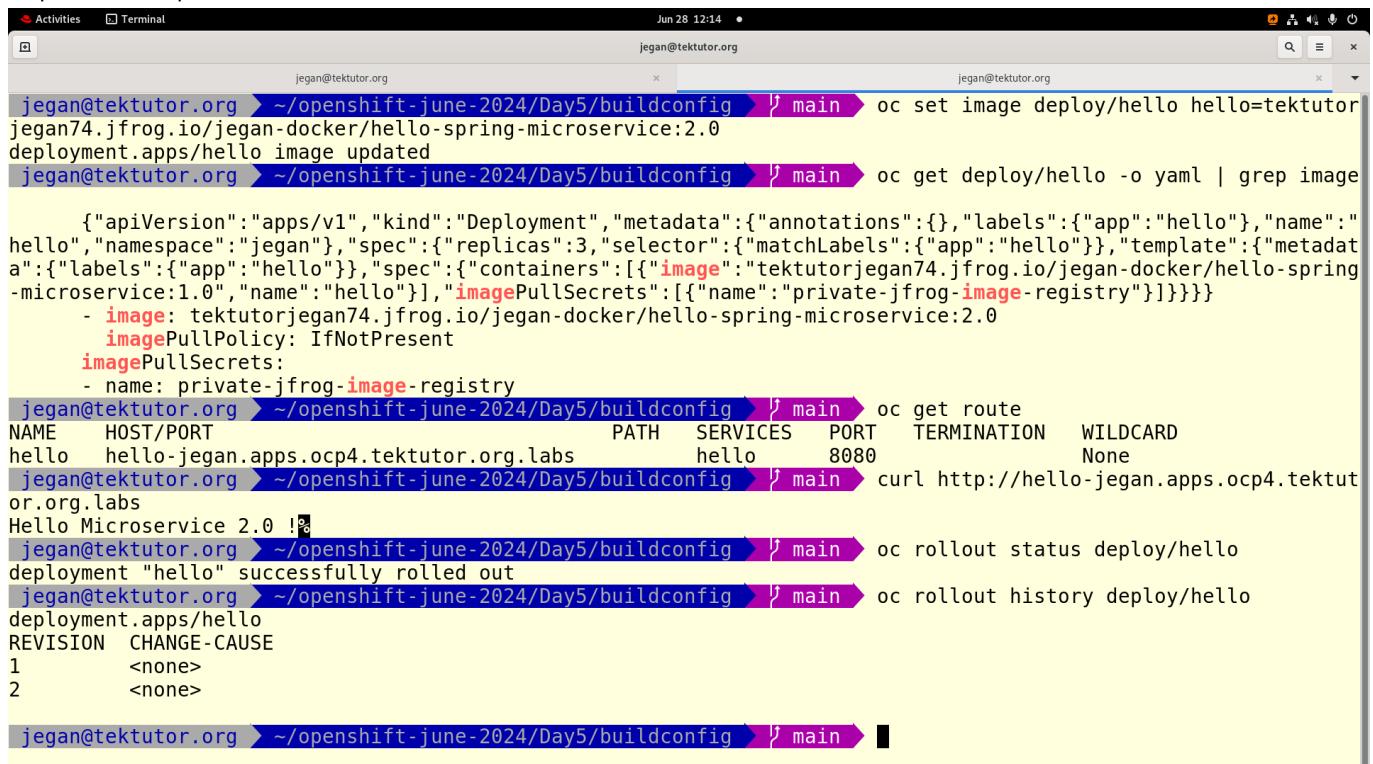
When the build config file is updated, we need to apply the changes in the openshift cluster

```
cd ~/openshift-june-2024
git pull
cd Day5/buildconfig
cat buildconfig-pushto-artifactory.yml
oc apply -f buildconfig-pushto-artifactory.yml
oc get buildconfigs
oc start-build bc/hello
```

Once the new image v2.0 is pushed successfully to JFrog Artifactory, you can proceed as shown below

```
oc set image deploy/hello hello=tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:2.0
oc get deploy/hello -o yaml | grep image
oc get route
curl http://hello-jegan.apps.ocp4.tektutor.org.labs
```

Expected output



The screenshot shows a terminal window with two tabs open. The left tab is titled 'Activities' and the right tab is titled 'Terminal'. The terminal window has a title bar 'jegan@tektutor.org' and a status bar 'Jun 28 12:14'. The terminal content is as follows:

```
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > oc set image deploy/hello hello=tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:2.0
jegan@tektutor.org deployment.apps/hello image updated
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > oc get deploy/hello -o yaml | grep image
{"apiVersion": "apps/v1", "kind": "Deployment", "metadata": {"annotations": {}, "labels": {"app": "hello"}, "name": "hello"}, "namespace": "jegan", "spec": {"replicas": 3, "selector": {"matchLabels": {"app": "hello"}}, "template": {"metadata": {"labels": {"app": "hello"}}, "spec": {"containers": [{"image": "tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:1.0", "name": "hello"}, {"imagePullSecrets": [{"name": "private-jfrog-image-registry"}]}]}}, "image": "tektutorjegan74.jfrog.io/jegan-docker/hello-spring-microservice:2.0", "imagePullPolicy": "IfNotPresent", "imagePullSecrets": [{"name": "private-jfrog-image-registry"}]}
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > oc get route
NAME      HOST/PORT          PATH      SERVICES    PORT      TERMINATION   WILDCARD
hello     hello-jegan.apps.ocp4.tektutor.org.labs   hello        8080           None
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > curl http://hello-jegan.apps.ocp4.tektutor.org.labs
Hello Microservice 2.0 !%
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > oc rollout status deploy/hello
deployment "hello" successfully rolled out
jegan@tektutor.org ~/openshift-june-2024/Day5/buildconfig > \ main > oc rollout history deploy/hello
deployment.apps/hello
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
```

The screenshot shows the Red Hat OpenShift Topology interface. The left sidebar has a 'Topology' section selected. The main area displays a single application named 'hello' within the 'jegan' project. The application icon is a blue circle with a white square containing a red 'H'. Below the icon, there's a button labeled 'hello' with a blue dot and three vertical dots.

At the bottom of the screen, there is another browser window titled 'Hello Microservice 2.0 !' which displays the message 'Hello Microservice 2.0 !'.

Info - What is Continuous Integration?

- Jenkins - is a CI Build Server
- We can create a Jenkins Job - to monitor code commits done in GitHub/BitBucket or any version control
- Whenever Jenkins detects code commit in the version control, it will start the build
- As part of the CI Build, it will first clone the latest source code from GitHub/BitBucket code repository
- Then it will start the application build (maven build, dotnet build)
- As part of the build, you also have to have some automated test cases

which runs part of the build

- If any test cases fails, the build will also fail
- If code is compiling and all test cases are executed successfully then the build will succeed.

Info - What is Continuous Deployment?

- the dev team certified CI builds, will automatically deploy the application binaries into QA environment for further automated testing
- if all the automated test cases added by QA team succeeds then the build is good to go live in production
- it might require some manual approvals

Info - What is Continuous Delivery?

- the QA certified build will automatically be deployed into pre-prod environment for the customer to check and approve to decide to make them live in production

Info - What is a Jenkins CI/CD Job?

- could build application source and run automated test cases
- could build custom docker images
- could deploy application binaries to JFrog, Weblogic or JBoss
- could deploy application into Openshift

Info - What is Jenkins Pipeline?

- Pipelines involves many Jenkins Job that run one after the other in sequence or in parallel
- Pipelines consists of many Stages
- Each Stage will have one Jenkins Job
- When the First Stage Job succeeds it will trigger next downstream jenkins job in the pipeline
- If the second stage Job succeeds it will trigger the next downstream jenkins job in the pipeline
- this goes on until all the jobs complete successfully
- if any one of the stage fails, it won't trigger the next downstream jenkins job and the build will fail

Lab - Deploying Jenkins in Openshift via Helm

The screenshot shows the Red Hat OpenShift Topology interface. The top navigation bar indicates it's a Google Chrome session from June 28 at 15:34. The main content area is titled "Topology - Red Hat OpenShift - Google Chrome". It displays a graph icon and the message "No resources found". Below this, there's a link to "Start building your application or visit the Add page for more details." The left sidebar has a "Developer" section with options like "+Add", "Topology", "Observe", "Search", "Builds", "Helm", "Project", "ConfigMaps", and "Secrets". The "Topology" option is currently selected.

Click on Add

The screenshot shows the Red Hat OpenShift "+Add" interface. The top navigation bar indicates it's a Google Chrome session from June 28 at 15:35. The main content area is titled "+Add - Red Hat OpenShift - Google Chrome". It displays a "Getting started resources" section with links to "Create applications using samples", "Build with guided documentation", and "Explore new developer features". Below this, there are five main sections: "Developer Catalog", "Git Repository", "Container Images", "Sharing", and "From Local Machine". Each section contains sub-links. For example, the "Developer Catalog" section has links to "All services" and "Database". The "Sharing" section has a link to "Samples". The "From Local Machine" section has links to "Import YAML" and "Upload JAR file". The left sidebar has a "Developer" section with options like "+Add", "Topology", "Observe", "Search", "Builds", "Helm", "Project", "ConfigMaps", and "Secrets". The "+Add" option is currently selected.

Click on Developer Catalog --> All Service

The screenshot shows the Red Hat OpenShift Developer Catalog interface. The left sidebar is titled 'Developer' and includes sections for Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area is titled 'Developer Catalog' and displays a grid of items. A search bar at the top of the catalog page shows 'Project: jegan'. The catalog lists items under 'All items' and 'CI/CD'. Some items shown include '.NET' builder images, .NET applications, A10 Helm Charts, akeyless-api-gateway Helm Charts, Alaz Helm Charts, Apache HTTP Server Templates, Apache HTTP Server (httpd) Builder Images, and Axual Governance Helm Charts.

Search jenkins and select Jenkins Helm Charts

This screenshot shows the same Red Hat OpenShift Developer Catalog interface as above, but with a search filter applied. The search bar now contains 'cloud'. In the 'CI/CD' section of the catalog, only the 'Jenkins' item is visible, indicating it has been selected or is highlighted.

Click on Create when you see the below screen

The screenshot shows the Red Hat OpenShift Developer Catalog interface. On the left, a sidebar menu includes options like Developer, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area is titled "Developer Catalog" and shows a search bar with "jenkins" and a dropdown menu with "All items". Below the search bar, there are three card-like entries under the heading "Templates": "Jenkins" (Persistent Storage), "Jenkins" (Community), and "Jenkins Monitored (Ephemeral)". To the right, a detailed view of the "Jenkins" chart is displayed, including its latest version (0.0.3), product version (1.16.0), source (Community), provider (Red Hat), home page (N/A), repository (OpenShift Helm Charts), maintainers (N/A), creation date (Jan 1, 05:53), and support (N/A). A "Create" button is visible at the top of this panel.

Accept the default values and click on Create button

The screenshot shows the "Create Helm Release" form for the Jenkins chart. The left sidebar is identical to the previous screenshot. The main form has fields for "Release name" (set to "jenkins") and "Chart version" (set to "0.0.3 / App Version 1.16.0 (Provided by OpenShift Helm Charts)"). Below these, a note states: "Note: Some fields may not be represented in this form view. Please select "YAML view" for full control." The "Required Resources" section contains fields for "fullnameOverride" (set to "jenkins"), "service", "replicaCount" (set to "1"), and "nameOverride". At the bottom of the form are "Create" and "Cancel" buttons.

The screenshot shows the Red Hat OpenShift Topology interface in Google Chrome. The URL is <https://console-openshift-console.apps.ocp4.tektutor.org.labs/topology/ns/jegan?view=graph>. The page displays a network graph of Jenkins pods. A specific pod, labeled "jenkins" with a blue DC icon, is highlighted with a dashed red border. This pod is part of a deployment configuration (DC) named "jenkins". Below the graph, there are search and filter tools.

The second screenshot is identical to the first, showing the same Jenkins pod highlighted with a dashed red border and part of a deployment configuration named "jenkins". The interface includes a sidebar with developer tools like Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets.

The screenshot shows the Red Hat OpenShift web interface. On the left, a sidebar menu includes options like Developer, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area displays a project named 'jegan'. A central component is a 'Topology' diagram showing a single pod labeled 'jenkins' with a Jenkins icon. To the right, a detailed view of the 'jenkins' DeploymentConfig (DC) is shown. The DC is marked as deprecated in OpenShift 4.14. It lists one pod, with the name 'jenkins' and namespace 'jegan'. The 'Latest version' is 1, and there is a note about a 'config change'. Buttons for 'Edit' and 'Update strategy' are present.

Click on "Login with Openshift"

This screenshot shows a Jenkins login screen. At the top, it says 'Log in to Jenkins using your OpenShift credentials - Google Chrome'. Below that is a Jenkins logo and the word 'Jenkins'. To its right is the Red Hat OpenShift logo. A message reads 'Log in to Jenkins using your OpenShift credentials'. A prominent blue button at the bottom says 'Log in with OpenShift'.

What makes the Serveless architecture work in Openshift or Kubernetes

- You need to install OpenShift Serverless Operator
- The Serverless Operator installs knative serverless framework

Knative and Red Hat Servless

- Red Hat Serverless is based on Knative opensource project
- Knative provides a serverless application layer on top of OpenShift/Kubernetes
- Knative consists of 3 building blocks
 - Build
 - Eventing
 - Serving

What does Serverless mean ?

- serverless doesn not mean the absence of servers
- is an architecture model for running applications in an environment that is abstracted away from developers
- developers can focus more on developing their application that where their code runs
- an ideal serverless workload executes a single task
- a function that retrieves data from a database can be an execellent serverless workload
- serverless model is the idea of the cold start
- when using serverless, there is a period between the request and creating the pod environment. This period is called cold start.
- Examples
 - openshift serverless workloads follow this workflow
 - a request comes in
 - a pod is spun up to service the request
 - the serves the request
 - the pod is destroy when there is no user traffic to handle
 - your service will be scaled down all the way upto 0 pod when there is 0 zero
 - Another example of a serverless workload can be an image processing function
 - an event could be a photo upload. The uploaded photo triggers an event to run an application to process the image
 - For example, the application may overtext text, create a banner, or make thumbnail
 - Once the image is stored permanently, the application has served its purpose and is no longer needed

Serverless Features

- Stateless Function
 - a function to query a database and return the data
 - a function to query weather report and return the data
- Event Driven
 - a serverless model relies on a trigger to execute the code
 - could be a request to an API or an event on a queue
- Auto Scales to zero

- Being able to scale to zero means your code only runs when it needs to respond to an event
- once the request is served, resources are released

Lab - Installing Red Hat OpenShift Serverless Operator as OpenShift Administrator

The screenshot shows the Red Hat OpenShift OperatorHub interface. The left sidebar navigation includes 'Administrator', 'Home', 'Operators' (selected), 'OperatorHub' (selected), 'Installed Operators', 'Workloads', 'Networking', 'Storage', 'Builds', 'Observe', 'Compute', 'User Management', and 'Administration'. The main content area displays the 'OperatorHub' page with the message 'Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace'. A search bar at the top right contains the query 'knative'. Below the search bar, there are two tabs: 'All Items' (selected) and 'All Items'. A search result for 'knative' is shown, with three items listed:

- Kogito Knative Eventing Source** (Community): Provided by Red Hat. Description: Knative Eventing Source for Kogito Services. This controller will enable easy configuration of...
- Red Hat OpenShift Serverless** (Red Hat): Provided by Red Hat. Description: Deploy and manage event-driven serverless applications and functions using Knative.
- YAKS Operator** (Community): Provided by Citrus Framework. Description: YAKS is a platform to enable Cloud Native BDD testing on Kubernetes.

The URL in the browser address bar is <https://console.openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/ns/jegan?keyword=knative&details-item=serverless-operator-redhat-operators-openshift-marketplace>.

The screenshot shows the detailed view for the Red Hat OpenShift Serverless operator. The left sidebar is identical to the previous screenshot. The main content area shows the operator details:

- Red Hat OpenShift Serverless** (1.33.0 provided by Red Hat)
- Channel**: stable
- Version**: 1.33.0
- Prerequisites**: Knative Serving (and Knative Eventing respectively) can only be installed into the `knative-serving` (`knative-eventing`) namespaces. These namespaces will be automatically created when installing the operator.
- Supported Features**:
 - Easy to get started: Provides a simplified developer experience to deploy and run cloud native applications on Kubernetes, providing powerful abstractions.
 - Immutable Revisions: Deploy new features performing canary, A/B or blue-green testing with gradual traffic rollout following best practices.
 - Use any programming language or runtime of choice: From Java, Python, Go and JavaScript to Quarkus, SpringBoot or Node.js.
 - Automatic scaling: Removes the requirement to configure numbers of replicas or idling behavior. Applications automatically scale to zero when not in use, or scale up to meet demand, with built-in reliability and fault tolerance.
 - Event Driven Applications: You can build loosely coupled, distributed applications that can be connected to a variety of either built in or third party event sources, powered by operators.
 - Ready for the hybrid cloud: Provides true, portable serverless functionality, that can run
- Valid Subscriptions**: OpenShift Container

The URL in the browser address bar is <https://console.openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/ns/jegan?keyword=knative&details-item=serverless-operator-redhat-operators-openshift-marketplace&channel=stable&version=1.33.0>.

Operator Installation - Red Hat OpenShift - Google Chrome

Jun 28 16:57 •

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/subscribe?pkg=serverless-operator&catalog=redhat-operators&catalogNamespace=openshift-marketplace&targetNamespace=jegan&channel=stable

Activities Google Chrome Log in to JFrog RE: Webex Cisco Webex 24MAN085 Operator In Log in to Jenkins Paused Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks kube:admin

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

Administrator

Home Operators OperatorHub Installed Operators Workloads Networking Storage Builds Observe Compute User Management Administration

Update channel * stable Version * 1.33.0

Installation mode * All namespaces on the cluster (default) Operator will be available in all Namespaces. A specific namespace on the cluster This mode is not supported by this Operator

Installed Namespace * Operator recommended Namespace: openshift-serverless Select a Namespace **Namespace creation** Namespace openshift-serverless does not exist and will be created.

Update approval * Automatic Manual

Install Cancel

Activities Google Chrome Jun 28 16:57 •

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/install/openshift-marketplace/serverless-operator/v1.33.0/to/openshift-serverless

Installing Operator - Red Hat OpenShift - Google Chrome

Activities Google Chrome Log in to JFrog RE: Webex Cisco Webex 24MAN085 Operator In Log in to Jenkins Paused Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks kube:admin

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

Administrator

Home Operators OperatorHub Installed Operators Workloads Networking Storage Builds Observe Compute User Management Administration

Red Hat OpenShift Serverless serverless-operator:v1.33.0 provided by Red Hat

Installing Operator

The Operator is being installed. This may take a few minutes.
[View installed Operators in Namespace openshift-serverless](#)

Activities Google Chrome Jun 28 16:58 •

Installing Operator - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/install/openshift-marketplace/serverless-operator/v1.33.0/to/openshift-serverless

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks

Red Hat OpenShift

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

Administrator

- Home
- Operators
- Workloads
- Networking
- Storage
- Builds
- Observe
- Compute
- User Management
- Administration

Red Hat OpenShift Serverless
serverless-operator.v1.33.0 provided by Red Hat

Installed operator: ready for use

[View Operator](#) [View installed Operators in Namespace openshift-serverless](#)

...org.labs/k8s/ns/openshift-serverless/operators.coreos.com-v1alpha1-ClusterServiceVers...

Activities Google Chrome Jun 28 16:59 •

serverless-operator.v1.33.0 - Cluster Service Version - Details - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/openshift-serverless/operators.coreos.com-v1alpha1-ClusterServiceVersion/serverless-operator.v1.33.0

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks

Red Hat OpenShift

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

Administrator

Project: openshift-serverless

Installed Operators > Operator details

Red Hat OpenShift Serverless
1.33.0 provided by Red Hat

[Actions](#)

Details	YAML	Subscription	Events	All instances	Knative Serving	Knative Eventing	Knative Kafka
Provided APIs							
Knative Serving A platform for streamlined application deployment, traffic-based auto-scaling from zero to N, and traffic-split rollouts Create instance			Knative Eventing An event-driven application platform that leverages CloudEvents with a simple HTTP interface Create instance		Knative Kafka An extension to Knative Eventing, merging HTTP accessibility with Apache Kafka's proven efficiency and reliability Create instance		
Description The Red Hat OpenShift Serverless operator provides a collection of APIs that enables containers, microservices and functions to run "serverless". Serverless applications can scale up and down (to zero) on demand and be triggered by a number of event sources. OpenShift Serverless integrates with a number of platform services, such as Monitoring and it is based on the open source project Knative.							
Prerequisites Knative Serving (and Knative Eventing respectively) can only be installed into the <code>knative-serving</code> (<code>knative-eventing</code>) namespace. These namespaces will be automatically created when installing the operator.							

Source Repository
<https://github.com/openshift-knative/serverless-operator>

Maintainers
Serverless Team
support@redhat.com

console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/openshift-serverless/.../new

Jun 28 16:59 • Create KnativeServing - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/openshift-serverless/clusterserviceversions/serverless-operator.v1.33.0/operator.knative.dev~v1beta1-KnativeServing/~new

Administrator Project: openshift-serverless

Create KnativeServing

Create by completing the form. Default values may be provided by the Operator authors.

Configure via Form view YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name * knative-serving

Labels app=frontend

security The security configuration for Knative Serving

controller-custom-certs Enabling the controller to trust registries with self-signed certificates

high-availability Allows specification of HA control plane

Knative Serving provided by Red Hat
A platform for streamlined application deployment, traffic-based auto-scaling from zero to N, and traffic-split rollouts

Jun 28 17:00 • Create KnativeServing - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/openshift-serverless/clusterserviceversions/serverless-operator.v1.33.0/operator.knative.dev~v1beta1-KnativeServing/~new

Administrator Project: openshift-serverless

Create KnativeServing

Create by completing the form. Default values may be provided by the Operator authors.

Configure via Form view YAML view

security The security configuration for Knative Serving

controller-custom-certs Enabling the controller to trust registries with self-signed certificates

high-availability Allows specification of HA control plane

deployments A mapping of deployment name to override

ingress The ingress configuration for Knative Serving

podDisruptionBudgets A mapping of podDisruptionBudget name to override

workloads A mapping of deployment or statefulset name to override

services A mapping of service name to override

Create **Cancel**

Jun 28 17:00 •

Create KnativeEventing - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/openshift-serverless/clusterserviceversions/serverless-operator.v1.33.0/operator.knative.dev~v1beta1-KnativeEventing/~new

Administrator Project: openshift-serverless

Create KnativeEventing

Create by completing the form. Default values may be provided by the Operator authors.

Configure via Form view YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name * knative-eventing

Labels app=frontend

defaultBrokerClass

The default broker type to use for the brokers Knative creates. If no value is provided, MTChannelBasedBroker will be used.

deployments A mapping of deployment name to override

high-availability Allows specification of HA control plane

defaultBrokerClass

The default broker type to use for the brokers Knative creates. If no value is provided, MTChannelBasedBroker will be used.

deployments A mapping of deployment name to override

high-availability Allows specification of HA control plane

podDisruptionBudgets A mapping of podDisruptionBudget name to override

services A mapping of service name to override

sinkBindingSelectionMode

Specifies the selection mode for the sinkbinding webhook. If the value is 'inclusion', only namespaces/objects labelled as 'bindings.knative.dev/include:true' will be considered. If 'exclusion' is selected, only 'bindings.knative.dev/exclude:true' label is checked and these will NOT be considered. The default for Openshift Serverless is 'inclusion'.

workloads A mapping of deployment or statefulset name to override

Create **Cancel**

Activities Google Chrome Jun 28 17:06 Pods - Red Hat OpenShift - Google Chrome

Editing opens... Inbox (7,601) Your GitHub Digital Learn... Login - JFrog Login to your... RE: Webex lin... RE: Webex lin... Cisco Web... 24MAN0852... Pods - Red Ha...

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/knative-serving/core-v1~Pod

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks

Red Hat OpenShift

Administrator

Project: knative-serving

Pods

Create Pod

Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
activator-69d945764d-dddkw	Running	2/2	0	RS activator-69d945764d	38.8 MiB	-	28 Jun 2024, 17:05
activator-69d945764d-rr9x6	Running	2/2	0	RS activator-69d945764d	0.0 MiB	-	28 Jun 2024, 17:04
autoscaler-59f6769648-b95t8	Running	2/2	0	RS autoscaler-59f6769648	0.0 MiB	-	28 Jun 2024, 17:04
autoscaler-59f6769648-z7g4k	Running	2/2	0	RS autoscaler-59f6769648	0.0 MiB	-	28 Jun 2024, 17:04
autoscaler-hpa-cf9b7cd64-tt4z4	Running	2/2	0	RS autoscaler-hpa-cf9b7cd64	28.2 MiB	-	28 Jun 2024, 17:04
autoscaler-hpa-cf9b7cd64-x6rcs	Running	2/2	0	RS autoscaler-hpa-cf9b7cd64	31.4 MiB	-	28 Jun 2024, 17:04
controller-858df8d47f-9tvn7	Running	2/2	0	RS controller-858df8d47f	0.0 MiB	-	28 Jun 2024, 17:04
controller-858df8d47f-nmmgb6	Running	2/2	0	RS controller-858df8d47f	0.0 MiB	-	28 Jun 2024, 17:04
controller-57854dc59c-fmm4w	ContainerCreating	0/2	0	RS controller-57854dc59c	0.0 MiB	-	28 Jun 2024, 17:04
storage-version-migration-serving-serving-112-133.0-ztqgh	Completed	0/1	0	I storage-version-migration-serving-serving-112-133.0-ztqgh	-	-	28 Jun 2024, 17:04
webhook-7fdc7fd58-8tcxz	Running	2/2	0	RS webhook-7fdc7fd58	37.8 MiB	-	28 Jun 2024, 17:05

Activities Google Chrome Jun 28 17:06 Pods - Red Hat OpenShift - Google Chrome

Editing opens... Inbox (7,601) Your GitHub Digital Learn... Login - JFrog Login to your... RE: Webex lin... RE: Webex lin... Cisco Web... 24MAN0852... Pods - Red Ha...

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/knative-eventing/core-v1~Pod

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks

Red Hat OpenShift

Administrator

Project: knative-eventing

Pods

Create Pod

Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
eventing-controller-6ffff654788-4vg7l	ContainerCreating	0/2	0	RS eventing-controller-6ffff654788	0.0 MiB	-	28 Jun 2024, 17:05
eventing-controller-6ffff654788-bhnzs	ContainerCreating	0/2	0	RS eventing-controller-6ffff654788	0.0 MiB	-	28 Jun 2024, 17:05
eventing-webhook-6c98cdc545-gn28p	ContainerCreating	0/2	0	RS eventing-webhook-6c98cdc545	0.0 MiB	-	28 Jun 2024, 17:05
eventing-webhook-6c98cdc545-vf5sd	ContainerCreating	0/2	0	RS eventing-webhook-6c98cdc545	0.0 MiB	-	28 Jun 2024, 17:05
imc-controller-6c54cb675d-lgjbp	ContainerCreating	0/2	0	RS imc-controller-6c54cb675d	0.0 MiB	-	28 Jun 2024, 17:05
imc-controller-6c54cb675d-wtst8	ContainerCreating	0/2	0	RS imc-controller-6c54cb675d	0.0 MiB	-	28 Jun 2024, 17:05
imc-dispatcher-944ccfd9f-bln8s	ContainerCreating	0/2	0	RS imc-dispatcher-944ccfd9f	0.0 MiB	-	28 Jun 2024, 17:05
imc-dispatcher-944ccfd9f-qbw7	ContainerCreating	0/2	0	RS imc-dispatcher-944ccfd9f	0.0 MiB	-	28 Jun 2024, 17:05
mt-broker-controller-57d9bd8bc4-7ng5n	ContainerCreating	0/2	0	RS mt-broker-controller-57d9bd8bc4	0.0 MiB	-	28 Jun 2024, 17:05
mt-broker-controller-	ContainerCreating	0/2	0	RS mt-broker-controller-	0.0 MiB	-	28 Jun 2024, 17:05

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

Command Line Tools

[Copy login command](#)

oc - OpenShift Command Line Interface (CLI)

With the OpenShift command line interface, you can create applications and manage OpenShift projects from a terminal.

The oc binary offers the same capabilities as the kubectl binary, but it is further extended to natively support OpenShift Container Platform features.

- [Download oc for Linux for x86_64](#)
- [Download oc for Mac for x86_64](#)
- [Download oc for Windows for x86_64](#)
- [Download oc for Linux for ARM 64](#)
- [Download oc for Mac for ARM 64](#)
- [Download oc for Linux for IBM Power little endian](#)
- [Download oc for Linux for IBM Z](#)
- [LICENSE](#)

helm - Helm 3 CLI

Helm 3 is a package manager for Kubernetes applications which enables defining, installing, and upgrading applications packaged as Helm Charts.

[Download Helm](#)

kn - OpenShift Serverless Command Line Interface (CLI)

The OpenShift Serverless client kn is a CLI tool that allows you to fully manage OpenShift Serverless Serving, Eventing, and Function resources without writing a single line of YAML.

- [Download kn for Linux for x86_64](#)
- [Download kn for Linux for ARM 64](#)
- [Download kn for Linux for IBM Power little endian](#)

Lab - Deploying your first knative service

```
kn service create hello \
--image ghcr.io/knative/helloworld-go:latest \
--port 8080
--env TARGET=World
```

Accessing the knative application from command line

```
curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
```

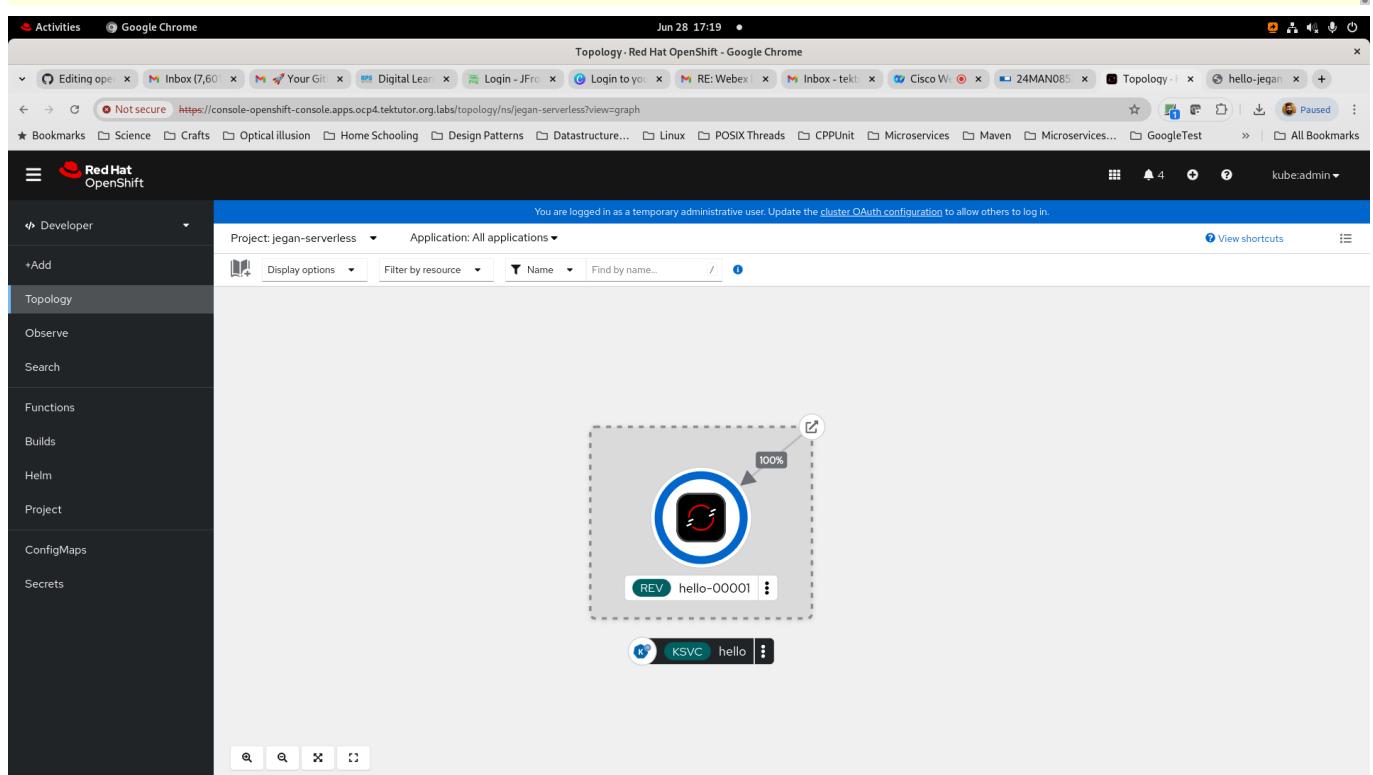
Expected output

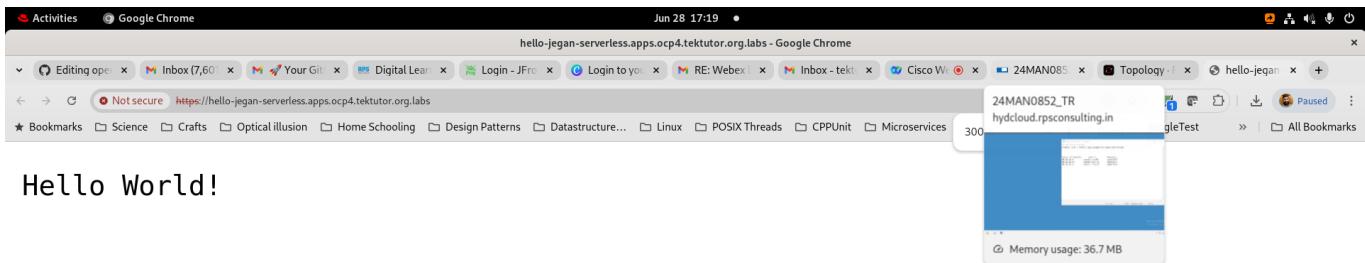
```

Activities Terminal Jun 28 17:18 •
jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main oc new-project jegan-serverless
Now using project "jegan-serverless" on server "https://api.ocp4.tektutor.org.labs:6443".
You can add applications to this project with the 'new-app' command. For example, try:
  oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
  kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.43 -- /agnhost serve-hostname
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn service create hello \
> --image ghcr.io/knative/helloworld-go:latest \
> --port 8080 \
> --env TARGET=World
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'jegan-serverless':
  0.064s The Route is still working to reflect the latest desired specification.
  0.100s ...
  0.112s Configuration "hello" is waiting for a Revision to become ready.
  17.006s ...
  17.050s Ingress has not yet been reconciled.
  17.119s Waiting for load balancer to be ready
  17.323s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main

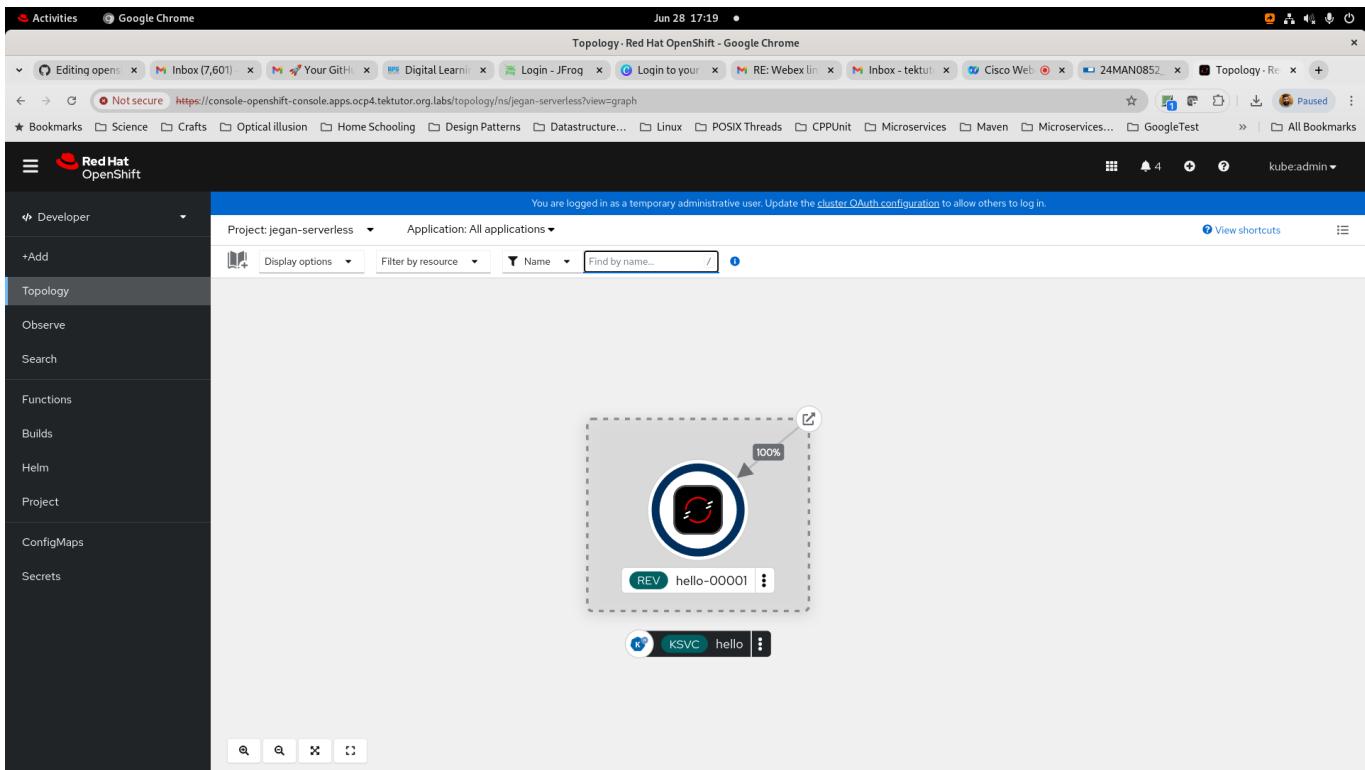
```





```
Activities Terminal Jun 28 17:21 •
jegan@tektutor.org
You can add applications to this project with the 'new-app' command. For example, try:
oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.43 -- /agnhost serve-hostname
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main kn service create hello \
> --image ghcr.io/knative/helloworld-go:latest \
> --port 8080 \
> --env TARGET=World
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'jegan-serverless':
0.064s The Route is still working to reflect the latest desired specification.
0.100s ...
0.112s Configuration "hello" is waiting for a Revision to become ready.
17.006s ...
17.050s Ingress has not yet been reconciled.
17.119s Waiting for load balancer to be ready
17.323s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main kn service list
NAME      URL          LATEST     AGE      CONDITIONS   READY   REASON
hello    https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs  hello-00001  3m58s  3 OK / 3  True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main
```



```

Activities Terminal Jun 28 17:22 •
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org

You can add applications to this project with the 'new-app' command. For example, try:
oc new-app rails-postgresql-example

to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.43 -- /agnhost serve-hostname

jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main kn service create hello \
> --image ghcr.io/knative/helloworld-go:latest \
> --port 8080 \
> --env TARGET=World
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'jegan-serverless':

0.064s The Route is still working to reflect the latest desired specification.
0.100s ...
0.112s Configuration "hello" is waiting for a Revision to become ready.
17.006s ...
17.050s Ingress has not yet been reconciled.
17.119s Waiting for load balancer to be ready
17.323s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main kn service list
NAME          URL           LATEST      AGE        CONDITIONS   READY   REASON
hello         https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs  hello-00001  3m58s  3 OK / 3  True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless > main

```

Update the service

```

kn service update hello --env TARGET=Knative!
kn revisions list

```

Accessing the knative application from command line

```

curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs

```

Expected output

```

Activities Terminal Jun 28 17:23 •
jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
0.064s The Route is still working to reflect the latest desired specification.
0.100s ...
0.112s Configuration "hello" is waiting for a Revision to become ready.
17.006s ...
17.050s Ingress has not yet been reconciled.
17.119s Waiting for load balancer to be ready
17.323s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn service list
NAME URL LATEST AGE CONDITIONS READY REASON
hello https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs hello-00001 3m58s 3 OK / 3 True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn service update hello --env TARGET=Knative!
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Updating Service 'hello' in namespace 'jegan-serverless':

0.036s The Configuration is still working to reflect the latest desired specification.
1.672s Traffic is not yet migrated to the latest revision.
1.715s Ingress has not yet been reconciled.
1.758s Waiting for load balancer to be ready
1.971s Ready to serve.

Service 'hello' updated to latest revision 'hello-00002' is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn revisions list
NAME SERVICE TRAFFIC TAGS GENERATION AGE CONDITIONS READY REASON
hello-00002 hello 100% 2 6s 4 OK / 4 True
hello-00001 hello 1 6m10s 3 OK / 4 True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main

```

Splitting the traffic between two revisions

```

kn service update hello --traffic hello-00001=50 --traffic @latest=50
kn revisions list

```

Expected output

```

Activities Terminal Jun 28 17:26 •
jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn revision list
NAME SERVICE TRAFFIC TAGS GENERATION AGE CONDITIONS READY REASON
hello-00002 hello 100% 2 2m7s 3 OK / 4 True
hello-00001 hello 1 8m11s 3 OK / 4 True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn service update hello --traffic hello-00001=50 --traffic @latest=50
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Updating Service 'hello' in namespace 'jegan-serverless':

0.036s The Route is still working to reflect the latest desired specification.
0.083s Ingress has not yet been reconciled.
0.132s Waiting for load balancer to be ready
0.288s Ready to serve.

Service 'hello' with latest revision 'hello-00002' (unchanged) is available at URL:
https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main kn revision list
NAME SERVICE TRAFFIC TAGS GENERATION AGE CONDITIONS READY REASON
hello-00002 hello 50% 2 2m46s 3 OK / 4 True
hello-00001 hello 50% 1 8m50s 3 OK / 4 True
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello Knative!!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main curl -k https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs
Hello World!
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ⌘ main

```

Delete the knative service

```
kn service list
kn service delete hello
kn service list
```

Expected output

The screenshot shows a terminal window with four tabs, all labeled 'jegan@tektutor.org'. The current tab is 'Terminal'. The command 'kn service list' is run, showing two services: 'hello-00002' and 'hello-00001'. Then, 'kn service delete hello' is run, which outputs a message confirming the deletion: 'Service "hello" successfully deleted in namespace "jegan-serverless".' Finally, 'kn service list' is run again, showing that no services are found.

```
jegan@tektutor.org ~ -/openshift-june-2024/Day5/serverless [main] $ kn service list
NAME        URL          LATEST   AGE      CONDITIONS   READY   REASON
hello      https://hello-jegan-serverless.apps.ocp4.tektutor.org.labs  hello-00002  9m47s  3 OK / 3  True
jegan@tektutor.org ~ -/openshift-june-2024/Day5/serverless [main] $ kn service delete hello
Service "hello" successfully deleted in namespace "jegan-serverless".
jegan@tektutor.org ~ -/openshift-june-2024/Day5/serverless [main] $ kn service list
No services found.
```

Lab - Knative eventing

Let's deploy a sink service

```
oc project jegan-serverless
kn service create eventinghello --concurrency-target=1 --
image=quay.io/rhdevelopers/eventinghello:0.0.2
```

Expected output

```

Activities Terminal Jun 28 17:31 •
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org

jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main oc project
Using project "jegan-serverless" on server "https://api.ocp4.tektutor.org.labs:6443".
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main kn service create eventinghello --concurrency-target=1 --image=quay.io/rhdevelopers/eventinghello:0.0.2
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'eventinghello' in namespace 'jegan-serverless':

0.066s The Route is still working to reflect the latest desired specification.
0.076s ...
0.104s Configuration "eventinghello" is waiting for a Revision to become ready.
24.858s ...
24.899s Ingress has not yet been reconciled.
24.947s Waiting for load balancer to be ready
25.144s Ready to serve.

Service 'eventinghello' created to latest revision 'eventinghello-00001' is available at URL:
https://eventinghello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main

```

Let's create an event source application

```
kn source ping create eventinghello-ping-source --schedule="*/2 * * * *" --data '{"message": "Thanks for your message"}' --sink ksvc:eventinghello
```

Expected output

```

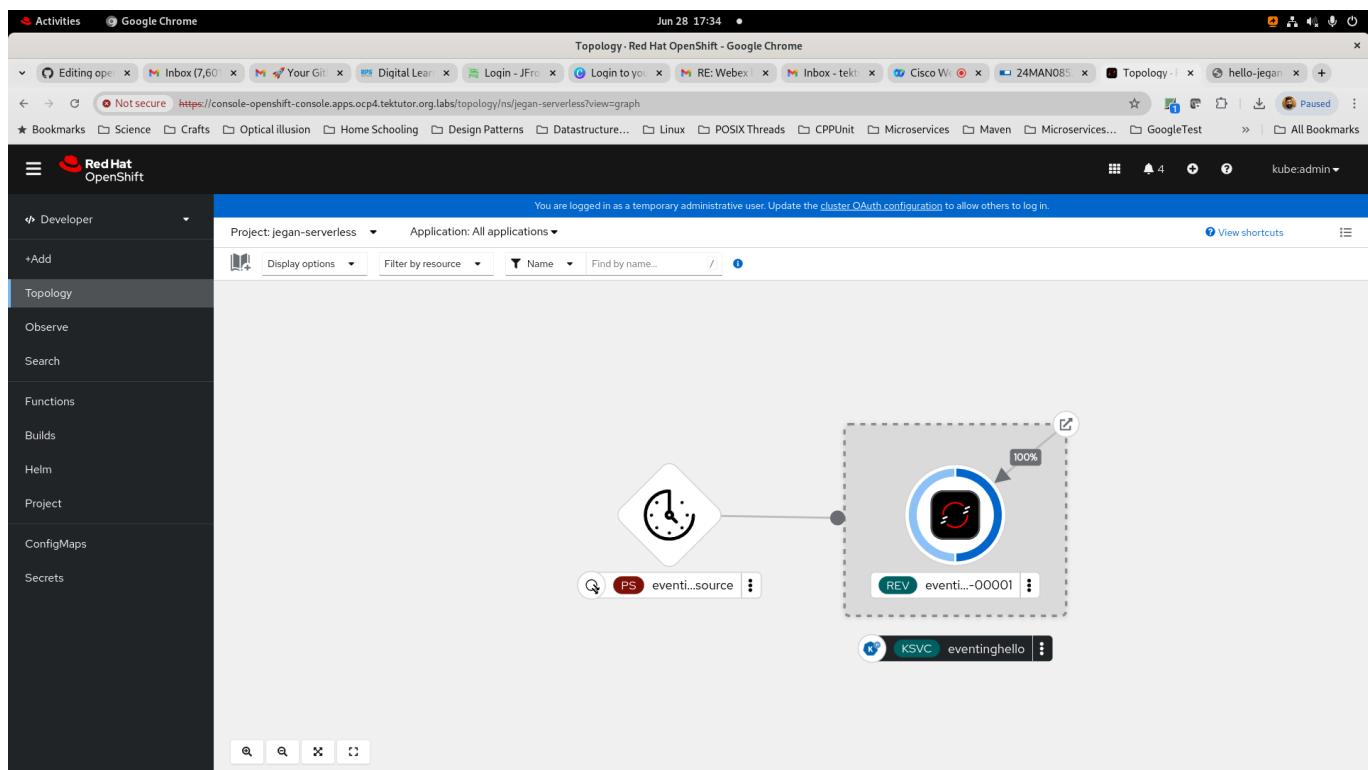
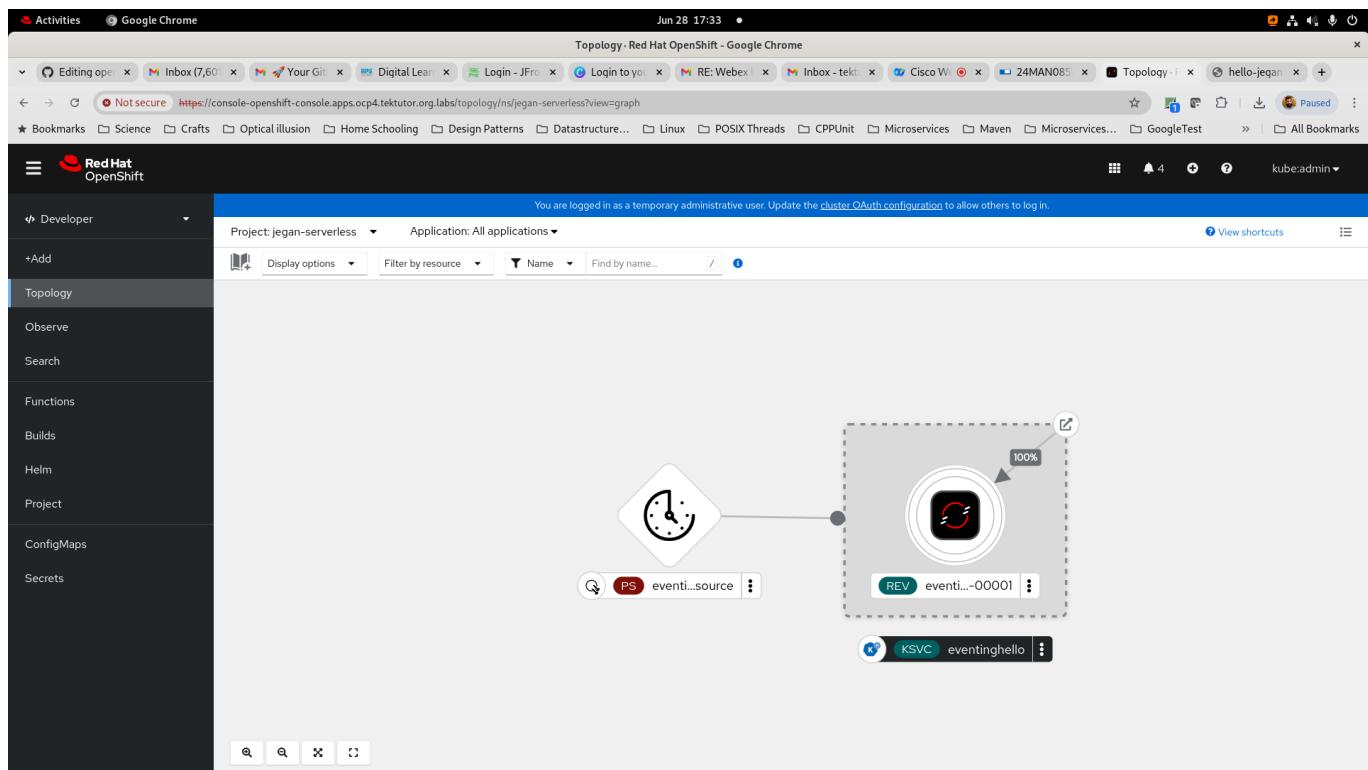
Activities Terminal Jun 28 17:33 •
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org

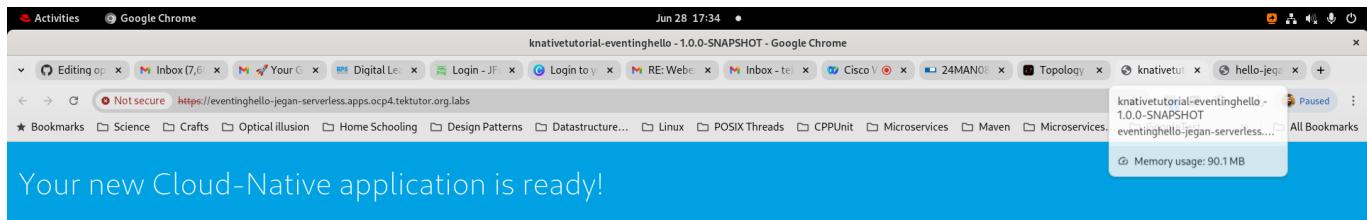
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main oc project
Using project "jegan-serverless" on server "https://api.ocp4.tektutor.org.labs:6443".
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main kn service create eventinghello --concurrency-target=1 --image=quay.io/rhdevelopers/eventinghello:0.0.2
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'eventinghello' in namespace 'jegan-serverless':

0.066s The Route is still working to reflect the latest desired specification.
0.076s ...
0.104s Configuration "eventinghello" is waiting for a Revision to become ready.
24.858s ...
24.899s Ingress has not yet been reconciled.
24.947s Waiting for load balancer to be ready
25.144s Ready to serve.

Service 'eventinghello' created to latest revision 'eventinghello-00001' is available at URL:
https://eventinghello-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main kn source ping create eventinghello-ping-source --schedule="*/2 * * * *" --data '{"message": "Thanks for your message"}' --sink ksvc:eventinghello
Warning: sources.knative.dev/v1beta2 PingSource is deprecated; see https://knative.dev/docs/eventing/sources/ping-source/ for instructions to migrate to sources.knative.dev/v1 PingSource
Ping source 'eventinghello-ping-source' created in namespace 'jegan-serverless'.
jegan@tektutor.org > ~/openshift-june-2024/Day5/serverless ✘ main

```





Congratulations, you have created a new Quarkus application.

Why do you see this?

This page is served by Quarkus. The source is in [src/main/resources/META-INF/resources/index.html](#).

What can I do from here?

If not already done, run the application in *dev mode* using: `mvn compile quarkus:dev`.

- Add REST resources, Servlets, functions and other services in `src/main/java`.
- Your static assets are located in `src/main/resources/META-INF/resources`.
- Configure your application in `src/main/resources/application.properties`.

How do I get rid of this page?

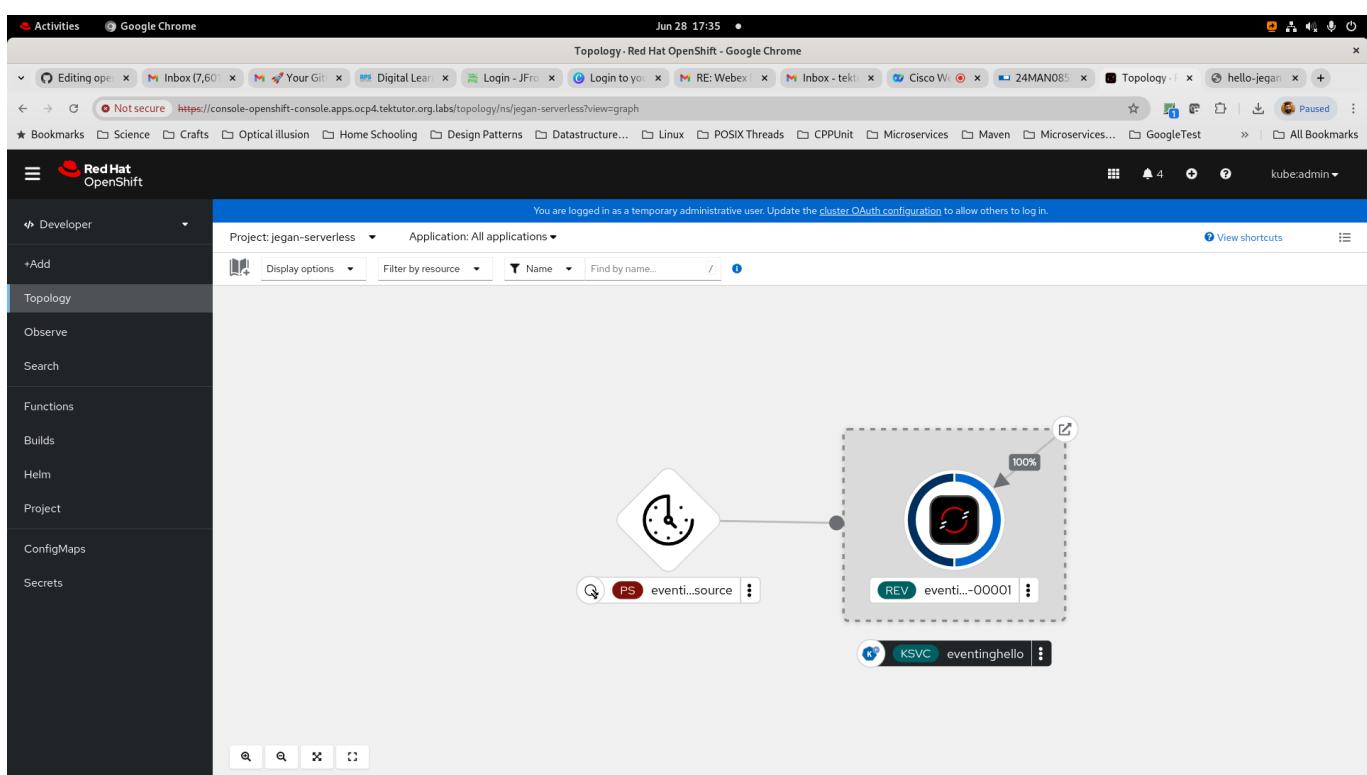
Just delete the `src/main/resources/META-INF/resources/index.html` file.

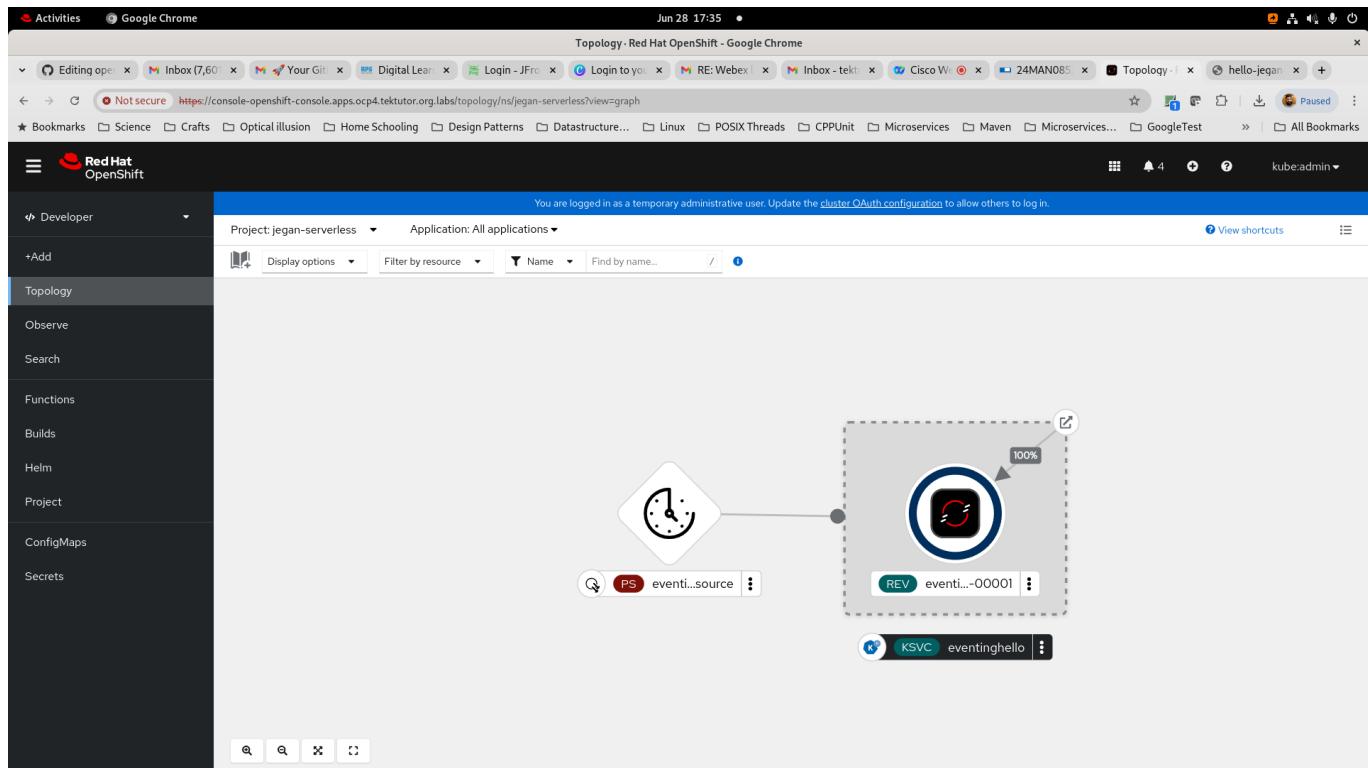
Application

GroupId: com.redhat.developers
ArtifactId: knativetutorial-eventinghello
Version: 1.0.0-SNAPSHOT
Quarkus Version: 1.4.2.Final

Next steps

[Setup your IDE](#)
[Getting started](#)
[Quarkus Web Site](#)





Lab - Developing a simple knative function in nodejs and deploying into Openshift cluster

This will generate a basic nodejs application in your current directory

```
kn func create -l node
```

If you wish to build your application

```
kn func build
```

If you wish to run the application locally and test it

```
kn func run
```

Deploy the nodejs application into openshift after building it

```
kn func deploy -n jegan-serverless
```

Test the knative function

```
curl -k https://functions-jegan.apps.ocp4.tektutor.org.labs
```

Expected output

```

Activities Terminal Jun 28 17:42 • jegan@tektutor.org
jegan@tektutor.org ~ /openshift-june-2024/Day5/serverless > main mkdir node-func
jegan@tektutor.org > /openshift-june-2024/Day5/serverless > main cd node-func
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main ls
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main kn func create -l node
Created node function in /home/jegan/openshift-june-2024/Day5/serverless/node-func
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main ls
func.yaml index.js package.json package-lock.json README.md test
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main kn func build
Building function image
Still building
Still building
Yes, still building
Function built: image-registry.openshift-image-registry.svc:5000/jegan-serverless/node-func:latest
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main kn func run
function up-to-date. Force rebuild with --build
Running on host port 8080
Environment:
  DEV_MODE=false
  NODE_ENV=production
  DEBUG_PORT=5858
Launching via npm...
npm info using npm@10.5.0
npm info using node@v20.12.2
> http-handler@0.1.0 start
> FUNC_LOG_LEVEL=info faas-js-runtime ./index.js

npm http fetch GET 200 https://registry.npmjs.org/npm 173ms
{"level":30,"time":1719576676668,"pid":17,"hostname":"51963da3ed5a","node_version":"v20.12.2","msg":"Server listening at http://[::]:8080"}
{"level":30,"time":1719576694347,"pid":17,"hostname":"51963da3ed5a","node_version":"v20.12.2","reqId":"req-1","req":{"method":"GET","url":"/","hostname":"localhost:8080","remoteAddress":"::ffff:172.17.0.1","remotePort":35774},"msg":"incoming request"}
{"level":30,"time":1719576694349,"pid":17,"hostname":"51963da3ed5a","node_version":"v20.12.2","reqId":"req-1","msg":"query"}
{"level":30,"time":1719576694349,"pid":17,"hostname":"51963da3ed5a","node_version":"v20.12.2","reqId":"req-1","msg":"body"}
{"level":30,"time":1719576694354,"pid":17,"hostname":"51963da3ed5a","node_version":"v20.12.2","reqId":"req-1","res":{"statusCode":200,"responseTime":5.9567319974303246,"msg":"request completed"}}

Activities Terminal Jun 28 17:46 • jegan@tektutor.org
jegan@tektutor.org ~ /openshift-june-2024/Day5/serverless/node-func > main kn func deploy -n jegan-serverless
function up-to-date. Force rebuild with --build
Please provide credentials for image registry (image-registry.openshift-image-registry.svc:5000).
? Username: kubeadmin
? Password: *****
Incorrect credentials, please try again.
? Username: E0628 17:44:46.091968 2834727 v2.go:150] use of closed network connection
X Sorry, your reply was invalid: Value is required
? Username: kubeadmin
? Password: *****
Credentials will not be saved.
If you would like to save your credentials in the future,
you can install docker credential helper https://github.com/docker/docker-credential-helpers.
Pushing function image to the registry "image-registry.openshift-image-registry.svc:5000" using the "kubeadmin" user credentials
↑ Deploying function to the cluster
🕒 Creating Triggers on the cluster
✓ Function deployed in namespace "jegan-serverless" and exposed at URL:
  https://node-func-jegan-serverless.apps.ocp4.tektutor.org.labs
jegan@tektutor.org > /openshift-june-2024/Day5/serverless/node-func > main

```

Activities Google Chrome Jun 28 17:47 • Topology - Red Hat OpenShift - Google Chrome

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

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks kube:admin

Red Hat OpenShift

Developer Project: jegan-serverless Application: All applications +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.

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

REV node-func-00001

fn KSVC node-func

Activities Google Chrome Jun 28 16:56 • OperatorHub - Red Hat OpenShift - Google Chrome

Not secure https://console-openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/ns/jegan?keyword=knative

Bookmarks Science Crafts Optical illusion Home Schooling Design Patterns Datastructure... Linux POSIX Threads CPPUnit Microservices Maven Microservices... GoogleTest All Bookmarks kube:admin

Red Hat OpenShift

Administrator Project: jegan Home Operators OperatorHub Installed Operators Workloads Networking Storage Builds Observe Compute User Management Administration

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

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace. You can install Operators on your clusters to provide optional add-ons and shared services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.

All Items

3 items

knative

Kogito Knative Eventing Source provided by Red Hat Knative Eventing Source for Kogito Services. This controller will enable easy configuration of...	Red Hat OpenShift Serverless provided by Red Hat Deploy and manage event-driven serverless applications and functions using Knative.	YAKS Operator provided by Citrus Framework YAKS is a platform to enable Cloud Native BDD testing on Kubernetes.
---	---	--

<https://console-openshift-console.apps.ocp4.tektutor.org.labs/operatorhub/ns/jegan?keyword=knative&details-item=serverless-operator-redhat-operators-openshift-marketplace>