

Day 6

Info - Openshift Service

- represents a groups of load-balanced pods from the same deployment
- service will forward the call to any one of the Pod within a single deployment

Info - What is Ingress?

- routing/forwarding rules
- Ingress helps in forwarding the calls to multiple different services pointing to different deployments
- Ingress is not a service
- We can declaratively create ingress rules, which are retrieved by Ingress Controller, which then configures the load balancer with the forwarding rules we listing in the ingress
- For Ingress to work, we need the below
 - Ingress (rules)
 - Ingress Controller
 - Load Balancer

Info - What is Ingress Controller?

- Ingress Controller is Controller like Deployment Controller, ReplicaSet Controller
- Ingress Controller keeps an eye on every new Ingress created in any project namespace
- Ingress Controller monitors any change done to existing Ingress resources under any project namespace
- Ingress Controller also will monitor when Ingress is deleted in any project namespace
- Ingress Controller picks the rules we mentioned in the Ingress resource and configures the load balancer accordingly
- There are two popular ingress controllers
 - Nginx Ingress Controller
 - HAProxy Ingress Controller
- In our lab setup, we are using HAProxy Load Balancer, hence we need to use HAProxy Ingress Controller

Info - Deployment vs DeploymentConfigs

- In older version of Kubernetes, we had to use ReplicationController to deploy applications into Kubernetes/OpenShift
- The Red Hat openshift team, at that time added DeploymentConfig to allow deploying application in the declarative style as the ReplicationController

doesn't support deploying application in the declarative style

- Meanwhile, the Google Kubernetes team & community added Deployment and ReplicaSet resource as an alternate for ReplicationController
- Hence, in Openshift the Red Hat team deprecated the use of DeploymentConfig as Deployment and DeploymentConfig pretty does the same
- Kubernetes, deprecated the use of ReplicationController
- Hence, whenever we deploy new application we need choose Deployment over the DeploymentConfig as DeploymentConfig internally uses ReplicationController

Info - ReplicationController vs ReplicaSet

- ReplicationController support both Rolling update and Scale up/down, which violates Single Responsibility Principle
- New applications should consider using Deployment over the ReplicationController
- When we create deployment, it automatically creates K8s Deployment resource and K8s ReplicaSet resource
- K8s Deployment resource is managed by Deployment Controller
- K8s ReplicaSet resource is managed by ReplicaSet Controller
- Deployment Controller supports rolling update
- ReplicaSet controller support scale up/down
- Hence, we should not use ReplicationController any more though it is there for backward compatibility purpose

Info - NodePort vs Route

- NodePort is an external service
- It is a K8s features, which is also supported in openshift
- Kubernetes/OpenShift reserve ports in range 30000-32767 for the purpose of NodePorts
- For each, NodePort service we create one of the ports from the above range will be allotted for the service
- the chosen nodeport is opened in all the nodes for the nodeport service
- if we create 100 nodeport services, we end up opening 100 firewall ports on all the nodes, which is a security concern
- also nodeport service is not end-user friendly or developer friendly as they are accessed via node hostname/ip address, ideally the end-user should not have worry about how many nodes are part of openshift
- routes is based on Kubernetes ingress, which provides an easy to access public url which is user-friendly as opposed to nodeport service
- hence, in openshift for internal service, we can create clusterip service
- for external access, we just need to expose the clusterip service as a route
- we don't have use node-port service in openshift

Info - Ingress vs Route

Lab - Creating your own Custom Resource in Openshift

```
cd ~/openshift-tekton-june2024
git pull
cd Day6/crd

oc get trainings
oc get training
oc get train

oc apply -f training-crd.yml

oc get trainings
oc get training
oc get train

oc apply -f devops-training.yml
oc apply -f openshift-training.yml

oc get trainings
oc get training
oc get train
```

Lab - Deploying wodpress & mariadb multi-pod application

Before deploy the wordpress and mariadb, you need to modify the mariadb-pv.yml, mariadb-pvc.yml, mariadb-deploy.yml, wordpress-pv.yml, wordpress-pvc.yml and wordpress-deploy.yml with your linux server IP and replace 'jegan' with your name.

```
cd ~/openshift-tektutor-june2024.git
git pull
cd Day6/wordpress
./deploy.sh
```

Expected output

```
jegan@tektutor.org ~ /openshift-tekton-june2024/Day5 ] $ main ] ls
crc1.png crc2.png crc3.png README.md
jegan@tektutor.org ~ /openshift-tekton-june2024/Day5 ] $ main ] cd ../Day6
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6 ] $ main ] ls
BuildConfig crd DevfileAndDeployDemo ingress README.md wordpress
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6 ] $ main ] cd wordpress
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6/wordpress ] $ main ] ls
delete-all.sh mariadb-pvc.yml wordpress-deploy.yml wordpress-route.yml
deploy.sh mariadb-pv.yml wordpress-pvc.yml wordpress-svc.yml
mariadb-deploy.yml mariadb-svc.yml wordpress-pv.yml
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6/wordpress ] $ main ] ./deploy.sh
\nDeploying mariadb ...
persistentvolume/mariadb-pv-jegan created
persistentvolumeclaim/mariadb-pvc-jegan created
deployment.apps/mariadb created
service/mariadb created
\nDeploying wordpress ...
persistentvolume/wordpress-pv-jegan created
persistentvolumeclaim/wordpress-pvc-jegan created
deployment.apps/wordpress created
service/wordpress created
route.route.openshift.io/wordpress created
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6/wordpress ] $ main ] sudo systemctl stop firewalld
jegan@tektutor.org ~ /openshift-tekton-june2024/Day6/wordpress ] $ main ]
```

The screenshot shows the Red Hat OpenShift console interface. On the left, there's a sidebar with navigation links: Developer, +Add, Topology (which is selected), Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area displays the 'Topology' view for the 'jegan' project. It shows two application icons: 'mariadb' and 'wordpress'. The 'wordpress' icon has a small 'D' icon next to it. To the right of the icons, there's a detailed panel for the 'wordpress' application. It includes sections for 'Health checks' (with a note about missing health checks), 'Resources' (listing a single pod), 'Pods' (showing one pod named 'wordpress-98c9cb676-znbr5' in a 'Running' state), 'Services' (listing a service 'wordpress' on port 8080), and 'Routes' (listing a route 'wordpress' pointing to 'http://wordpress-jegan.apps.ocp4.tektutor.org.labs').

wordpress-98c9cb676-znbr5 - Pod - Details - Red Hat OpenShift - Google Chrome

Project: jegan

wordpress-98c9cb676-znbr5 Running

Pod details

Name	wordpress-98c9cb676-znbr5	Status	Running
Namespace	jegan	Restart policy	Always restart
Labels	app=wordpress, pod-template-hash=98c9cb676	Edit	Active deadline seconds Not configured
Node selector	No selector	Pod IP	10.128.2.28
Tolerations	2 tolerations	Host IP	192.168.122.217
Annotations	4 annotations	Node	worker-2.ocp4.tektutor.org.labs
Created at	17 Jun 2024, 11:23	Image pull secret	default-dockercfg-4rhj8
PodDisruptionBudget			

wordpress-98c9cb676-znbr5 - Pod - Events - Red Hat OpenShift - Google Chrome

Project: jegan

wordpress-98c9cb676-znbr5 Running

Events

Showing 6 events

- Streaming events...
- Successfully pulled image "bitnami/wordpress:latest" in 27.476s (27.476s including waiting)
- Created container wordpress
- Started container wordpress
- Add eth0 [10.128.2.28/23] from ovn-kubernetes
- Generated from kubelet on worker-2.ocp4.tektutor.org.labs

<https://console-openshift-console.apps.ocp4.tektutor.org.labs/k8s/ns/jegan/pods/wordpress-98c9cb676-znbr5/events>

wordpress-98c9cb676-znbr5 - Pod - Logs - Red Hat OpenShift - Google Chrome

```

19 | wordpress 05:55:44.63 INFO >>> Configuring WordPress with settings provided via environment variables
20 | wordpress 05:55:45.71 INFO >>> Installing WordPress
21 | wordpress 05:55:47.95 INFO >>> Persisting WordPress installation
22 | wordpress 05:56:26.89 INFO >>> *** WordPress setup finished! ***
23 |
24 | wordpress 05:56:26.90 INFO >>> ** Starting Apache **
25 | [Mon Jun 17 05:56:27.017374 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.59 (Unix) OpenSSL/3.0.11 configured -- resuming normal operations
26 | [Mon Jun 17 05:56:27.017459 2024] [core:notice] [pid 1] AH00094: Command line: '/opt/bitnami/apache/bin/httpd -f /opt/bitnami/apache/conf/httpd.conf -D FOREGROUND'
27 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET / HTTP/1.1" 200 14789
28 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/blocks/navigation/style.min.css?ver=6.5.4 HTTP/1.1" 200 2290
29 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/blocks/image/style.min.css?ver=6.5.4 HTTP/1.1" 200 1597
30 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/blocks/navigation/view.min.js?ver=6.5.4 HTTP/1.1" 200 1135
31 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/images/tourist-and-building.webp HTTP/1.1" 200 66482
32 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/images/building-exterior.webp HTTP/1.1" 200 199724
33 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/js/dist/interactivity.min.js?ver=6.5.4 HTTP/1.1" 200 13147
34 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/images/window.webp HTTP/1.1" 200 126244
35 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/fonts/inter/Inter-VariableFont_sltnt_wght.woff2 HTTP/1.1" 200 326628
36 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/fonts/cardo/cardo_normal.400.woff2 HTTP/1.1" 200 105184
37 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-content/themes/twentytwentyfour/assets/fonts/cardo/cardo_normal_400.woff2 HTTP/1.1" 200 146060
38 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/jss/wp-emoji-release.min.js?ver=6.5.4 HTTP/1.1" 200 5062
39 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /favicon.ico HTTP/1.1" 302 -
40 | 10.128.0.2 - - [17/Jun/2024:05:56:44 +0000] "GET /wp-includes/images/w-logo-blue-white-bg.png HTTP/1.1" 200 4119

```

mariadb-5b9895469b-5j4zr - Pod - Details - Red Hat OpenShift - Google Chrome

Pod details	Value
Name	mariadb-5b9895469b-5j4zr
Status	Running
Namespace	jegan
Restart policy	Always restart
Labels	app=mariadb, pod-template-hash=5b9895469b
Active deadline seconds	Not configured
Node selector	No selector
Tolerations	2 tolerations
Annotations	4 annotations
Created at	17 Jun 2024, 11:23
PodDisruptionBudget	

The screenshot shows the Red Hat OpenShift web interface. On the left, a sidebar lists various project components like Developer, Topology, Observes, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area is titled 'Pods > Pod details' for a pod named 'mariadb-5b9895469b-5j4zr'. The 'Logs' tab is selected, showing log entries from line 66 to 87. The logs detail the MySQL startup process, including the configuration of transaction pools and the loading of buffer pools.

The screenshot shows a WordPress blog page titled 'User's blog'. The page has a header with the title 'A commitment to innovation and sustainability' and a subtext stating 'Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.' There is a 'About us' button. Below the header is a large image of a modern building's facade with a unique, ribbed roof structure.

Ingress

Let's create a nginx deployment

```
oc project
oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
oc get deploy,rs,po
```

Let's create clusterip internal service for nginx deployment

```
oc expose deploy/nginx --port=8080
oc get svc
oc describe svc/nginx
```

Let's create a hello microservice deployment

```
oc project
oc create deployment hello --image=tektutor/spring-ms:1.0 --replicas=3
oc get deploy,rs,po
```

Let's create clusterip internal service for hello deployment

```
oc expose deploy/hello --port=8080
oc get svc
oc describe svc/hello
```

Now let's create the ingress

```
cd ~/openshift-tekton-june2024
git pull
cd Day6/ingress
cat ingress.yml
oc apply -f ingress.yml
oc get ingress
```

Access the different services using ingress url, to access nginx service

```
curl http://tektutor.apps.ocp4.tektutor.org.labs/nginx
```

Access the different services using ingress url, to access hello service

```
curl http://tektutor.apps.ocp4.tektutor.org.labs/hello
```

Expected output

```
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main vim ingress.yml
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc get deploy
No resources found in jegan namespace.
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
deployment.apps/nginx created
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc create deployment hello --image=tektutor/spring-ms:1.0 --replicas=3
deployment.apps/hello created
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc get deploy
NAME READY UP-TO-DATE AVAILABLE AGE
hello 0/3 3 0 8s
nginx 2/3 3 2 23s
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc get po
NAME READY STATUS RESTARTS AGE
hello-56d67b65ff-25s7s 0/1 ContainerCreating 0 11s
hello-56d67b65ff-4cxgv 0/1 ContainerCreating 0 11s
hello-56d67b65ff-mg7ln 0/1 ContainerCreating 0 11s
nginx-66c775969-8jhl6 1/1 Running 0 26s
nginx-66c775969-9b8hk 1/1 Running 0 26s
nginx-66c775969-lnzs5 0/1 ContainerCreating 0 26s
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc get po -w
NAME READY STATUS RESTARTS AGE
hello-56d67b65ff-25s7s 0/1 ContainerCreating 0 20s
hello-56d67b65ff-4cxgv 0/1 ContainerCreating 0 20s
hello-56d67b65ff-mg7ln 0/1 ContainerCreating 0 20s
nginx-66c775969-8jhl6 1/1 Running 0 35s
nginx-66c775969-9b8hk 1/1 Running 0 35s
nginx-66c775969-lnzs5 0/1 ContainerCreating 0 35s
nginx-66c775969-lnzs5 1/1 Running 0 38s
hello-56d67b65ff-mg7ln 1/1 Running 0 39s
hello-56d67b65ff-4cxgv 1/1 Running 0 41s
hello-56d67b65ff-25s7s 1/1 Running 0 48s
^C
x jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main

jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc expose deploy/nginx --port=8080
Error from server (AlreadyExists): services "nginx" already exists
x jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc expose deploy/hello --port=8080
service/hello exposed
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
hello ClusterIP 172.30.152.30 <none> 8080/TCP 3s
nginx ClusterIP 172.30.99.210 <none> 8080/TCP 90m
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main oc describe svc/nginx
Name: nginx
Namespace: jegan
Labels: app=nginx
Annotations: <none>
Selector: app=nginx
Type: ClusterIP
IP Family Policy: SingleStack
IP Families: IPv4
IP: 172.30.99.210
IPs: 172.30.99.210
Port: <unset> 8080/TCP
TargetPort: 8080/TCP
Endpoints: 10.128.2.32:8080,10.129.0.108:8080,10.131.0.9:8080
Session Affinity: None
Events: <none>
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ 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: IPv4
IP: 172.30.152.30
IPs: 172.30.152.30
Port: <unset> 8080/TCP
TargetPort: 8080/TCP
Endpoints: 10.128.0.80:8080,10.128.2.33:8080,10.131.0.11:8080
Session Affinity: None
Events: <none>
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress ↵ main
```

```
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress jegan@tektutor.org
jegan@tektutor.org > ./main ls
ingress.yml
jegan@tektutor.org > ./main cat ingress.yml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tektutor
  annotations:
    haproxy.router.openshift.io/rewrite-target: /
spec:
  rules:
  - host: tektutor.apps.ocp4.tektutor.org.labs
    http:
      paths:
        - backend:
            service:
              name: nginx
              port:
                number: 8080
            path: /nginx
            pathType: Prefix
        - backend:
            service:
              name: hello
              port:
                number: 8080
            path: /hello
            pathType: Prefix
jegan@tektutor.org > ./main oc apply -f ingress.yml
ingress.networking.k8s.io/tekstutor created
jegan@tektutor.org > ./main oc get ingress
NAME      CLASS      HOSTS      ADDRESS      PORTS      AGE
tekstutor <none>    tektutor.apps.ocp4.tektutor.org.labs   router-default.apps.ocp4.tektutor.org.labs   80          23s
jegan@tektutor.org > ./main
```

```
jegan@tektutor.org ~ -/openshift-tekton-june2024/Day6/ingress jegan@tektutor.org
number: 8080
path: /hello
pathType: Prefix
jegan@tektutor.org > ./main oc apply -f ingress.yml
ingress.networking.k8s.io/tekstutor created
jegan@tektutor.org > ./main oc get ingress
NAME      CLASS      HOSTS      ADDRESS      PORTS      AGE
tekstutor <none>    tektutor.apps.ocp4.tektutor.org.labs   router-default.apps.ocp4.tektutor.org.labs   80          23s
jegan@tektutor.org > ./main curl http://tektutor.apps.ocp4.tektutor.org.labs/hello
Greetings from Spring Boot!
jegan@tektutor.org > ./main curl http://tektutor.apps.ocp4.tektutor.org.nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
jegan@tektutor.org > ./main
```

Lab - S2I using Docker strategy by using declarative build config manifest file

First we need create the imagestream to store the image in openshift internal container registry

```
cd ~/openshift-tekton-june2024
git pull
cd Day6/buildconfig
cat imagestream.yml
oc apply -f imagestream.yml
oc get imagestreams
```

```
oc get imagestream  
oc get is
```

Now, let's create the builconfig and start the build

```
cd ~/openshift-tekton-june2024  
git pull  
cd Day6/buildconfig  
oc apply -f buildconfig.yml  
oc get buildconfigs  
oc get buildconfig  
oc get bc
```

To check the build log, you may try this

```
oc logs -f bc/hello
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab is titled 'jegan@tektutor.org' and contains the command 'ls'. The right tab is also titled 'jegan@tektutor.org' and contains the command 'cat buildconfig.yml'. The terminal output shows the YAML configuration for the 'hello' BuildConfig, which includes triggers for ConfigChange, a Git source from a GitHub repository, a Docker strategy, and an output to an ImageStreamTag named 'tektutor-spring-hello:latest'.

```
jegan@tektutor.org ~ ~/openshift-tekton-june2024/Day6/BuildConfig ⌘ main ➔ ls  
buildconfig-pushto-artifactory.yml Dockerfile hello-route.yml Jenkinsfile src  
buildconfig.yml hello-deploy.yml hello-svc.yml pom.xml  
jegan@tektutor.org ~ ~/openshift-tekton-june2024/Day6/BuildConfig ⌘ main ➔ cat buildconfig.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-tekton-june2024.git"  
      ref: main  
      contextDir: "Day6/BuildConfig"  
  strategy:  
    type: Docker  
  output:  
    to:  
      kind: ImageStreamTag  
      name: 'tektutor-spring-hello:latest'  
jegan@tektutor.org ~ ~/openshift-tekton-june2024/Day6/BuildConfig ⌘ main ➔
```

```
jegan@tektutor.org
jegan@tektutor.org
DATED
tekton-spring-hello    image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get is
NAME          IMAGE REPOSITORY
DATED
tekton-spring-hello    image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc apply -f buildconfig.yml
buildconfig.build.openshift.io/hello created
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get buildconfigs
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get buildconfig
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get bc
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get builds
NAME      TYPE   FROM      STATUS    STARTED   DURATION
hello-1   Docker  Git@main Pending
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get builds
NAME      TYPE   FROM      STATUS    STARTED   DURATION
hello-1   Docker  Git@main Pending
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc logs -f bc/hello
Cloning "https://github.com/tektutor/openshift-tekton-june2024.git" ...
Commit: d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b (Update buildconfig-pushto-artifactory.yml)
Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
Date: Mon Jun 17 12:31:09 2024 +0530
```

```
jegan@tektutor.org
jegan@tektutor.org
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get buildconfigs
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get buildconfig
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get bc
NAME      TYPE   FROM      LATEST
hello    Docker  Git@main  1
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get builds
NAME      TYPE   FROM      STATUS    STARTED   DURATION
hello-1   Docker  Git@main Pending
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get builds
NAME      TYPE   FROM      STATUS    STARTED   DURATION
hello-1   Docker  Git@main Pending
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc logs -f bc/hello
Cloning "https://github.com/tektutor/openshift-tekton-june2024.git" ...
Commit: d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b (Update buildconfig-pushto-artifactory.yml)
Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
Date: Mon Jun 17 12:31:09 2024 +0530
time="2024-06-17T07:06:40Z" 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"
I0617 07:06:40.289898       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...
```

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

NAME	TYPE	FROM	LATEST
hello	Docker	Git@main	1

```
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > ⏎ main ● oc get builds
```

NAME	TYPE	FROM	STATUS	STARTED	DURATION
hello-1	Docker	Git@main	Pending		

```
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > ⏎ main ● oc get builds
```

NAME	TYPE	FROM	STATUS	STARTED	DURATION
hello-1	Docker	Git@main	Pending		

```
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > ⏎ main ● oc logs -f bc/Hello
```

```
Cloning "https://github.com/tektutor/openshift-tekton-june2024.git" ...
Commit: d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b (Update buildconfig-pushto-artifactory.yml)
Author: Jeganathan Swaminathan <mail2jegan@gmail.com>
Date: Mon Jun 17 12:31:09 2024 +0530

time="2024-06-17T07:06:40Z" 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"
I0617 07:06:40.289898      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 sha256:6c215442f70bd949a6f2e8092549943905e2d4f9c87a4f532d7740ae8647d33a
Copying blob sha256:234b70d0479d7f16d7ee8d04e4ffdacc57d7d14313faf59d332f18b2e9418743
Copying blob sha256:5d6f1e8117dbb1c6a57603cb4f321a861a08105a81bcc6b01b0ec2b78c8523a5
Copying blob sha256:004f1eed87df3f75f5e2a1a649fa7edd7f713d1300532fd0909bb39cd48437d7
Copying blob sha256:48c2faf66abec3dce9f54d6722ff592fce6dd4fb58a0d0b72282936c6598a3b3
Copying blob sha256:d7eb6c022a4e6128219b32a8e07c8c22c89624ff440ebac1506121794bc15ccc
```

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

```
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.6.3/junit-bom-5.6.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/junit-bom/5.6.3/junit-bom-5.6.3.pom (4.9 kB at 137 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/hamcrest/hamcrest/2.2/hamcrest-2.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/hamcrest/hamcrest/2.2/hamcrest-2.2.pom (1.1 kB at 39 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-5.7.0/junit-jupiter-5.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-5.7.0/junit-jupiter-5.7.0.pom (3.2 kB at 107 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-api/5.7.0/junit-jupiter-api-5.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-api/5.7.0/junit-jupiter-api-5.7.0.pom (3.2 kB at 103 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apiguardian/apiguardian-api/1.1.0/apiguardian-api-1.1.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apiguardian/apiguardian-api/1.1.0/apiguardian-api-1.1.0.pom (1.2 kB at 38 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/opentest4j/opentest4j/1.2.0/opentest4j-1.2.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/opentest4j/opentest4j/1.2.0/opentest4j-1.2.0.pom (1.7 kB at 58 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/platform/junit-platform-commons/1.7.0/junit-platform-commons-1.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/platform/junit-platform-commons/1.7.0/junit-platform-commons-1.7.0.pom (2.8 kB at 86 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-params/5.7.0/junit-jupiter-params-5.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-params/5.7.0/junit-jupiter-params-5.7.0.pom (3.0 kB at 104 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-engine/5.7.0/junit-jupiter-engine-5.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/jupiter/junit-jupiter-engine/5.7.0/junit-jupiter-engine-5.7.0.pom (3.2 kB at 107 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/junit/platform/junit-platform-engine/1.7.0/junit-platform-engine-1.7.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/junit/platform/junit-platform-engine/1.7.0/junit-platform-engine-1.7.0.pom (3.2 kB at 110 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/mockito/mockito-core/3.6.28/mockito-core-3.6.28.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/mockito/mockito-core/3.6.28/mockito-core-3.6.28.pom (2.9 kB at 95 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy/1.10.19/byte-buddy-1.10.19.pom
Downloaded from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy/1.10.19/byte-buddy-1.10.19.pom (11 kB at 356 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy-parent/1.10.19/byte-buddy-parent-1.10.19.pom
Downloaded from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy-parent/1.10.19/byte-buddy-parent-1.10.19.pom (42 kB at 1.3 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy-agent/1.10.19/byte-buddy-agent-1.10.19.pom
Downloaded from central: https://repo.maven.apache.org/maven2/net/byteduddy/byte-buddy-agent/1.10.19/byte-buddy-agent-1.10.19.pom (9.6 kB at 319 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/objenesis/objenesis/3.1/objenesis-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/objenesis/objenesis/3.1/objenesis-3.1.pom (3.5 kB at 118 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/objenesis/objenesis/3.1/objenesis-parent-3.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/objenesis/objenesis-parent/3.1/objenesis-parent-3.1.pom (18 kB at 569 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/mockito/mockito-junit-jupiter/3.6.28/mockito-junit-jupiter-3.6.28.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/mockito/mockito-junit-jupiter/3.6.28/mockito-junit-jupiter-3.6.28.pom (2.7 kB at 86 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/skyscreamer/jsonassert/1.5.0/jsonassert-1.5.0.pom (5.2 kB at 167 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/vaadin/external/google/android-json/0.0.20131108.vaadin1/android-json-0.0.20131108.vaadin1.pom
```

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

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

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

Red Hat OpenShift kube:admin Paused

Developer Project: jegan Details Metrics YAML Environment Logs Events

⚠ Some lines have been abridged because they are exceptionally long.
To view unabridged log content, you can either open the raw file in another window or download it.

Log stream ended. Search Show full log Wrap lines Raw Download Expand

```

1000 lines
980 [2/2] STEP 3/6: EXPOSE 8080
981 --> 72a97dc008b7
982 [2/2] STEP 4/6: ENTRYPOINT ['java','-jar','app.jar']
983 --> a00cdac6bc49
984 [2/2] STEP 5/6: ENV 'OPENSHIFT_BUILD_NAME="hello-1" "OPENSHIFT_BUILD_NAMESPACE="jegan" "OPENSHIFT_BUILD_SOURCE="https://github.com/tektutor/openshift-tekn-june2024.git" "OPENSHIFT_BUILD_REFERENCE="main" "OPENSHIFT_BUILD_COMMIT="d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b"
985 --> 750388d706f2
986 [2/2] STEP 6/6: LABEL "io.openshift.build.commit.author="Jeganathan Swaminathan <mail2jegan@gmail.com>" "io.openshift.build.commit.date="Mon Jun 17 12:31:09 2024 +0530" "io.openshift.
987 [2/2] COMMIT temp.builder.openshift.io/jegan/hello-1:6b9ae299
988 --> 20a199c8ff22
989 Successfully tagged temp.builder.openshift.io/jegan/hello-1:6b9ae299
990 20a199c8ff22b056aafe92d87047ef401af15e8a0aab81c07e325d5f7593d67
991 Pushing image image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello:latest ...
992 Getting image source signatures
993 Copying blob sha256:6bb8b50357633e3d0a7b9e9bed5b614a06bab6cd3a569d0326d73ab8b0c0bee0b
994 Copying blob sha256:8393debac2ec316f87eab40a0f5451d57df2863f844f545d40447e46e025c9663
995 Copying blob sha256:1846180440d3041e084880b0bb735f024f3bc5e2d39f1ab4ad490f630863d903
996 Copying config sha256:1846180440d3041e084880b0bb735f024f3bc5e2d39f1ab4ad490f630863d903
997 Copying config sha256:20a199c8ff22b056aafe92d87047ef401af15e8a0aab81c07e325d5f7593d67
998 Writing manifest to image destination
999 Successfully pushed image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello@sha256:12fabab3b75b0db6670d473289092eed3bd277828988e1d001cfdfb79b190fa4
1000 Push successful

```

jegan@tektutor.org

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 765 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: 19.012 s

[INFO] Finished at: 2024-06-17T07:08:03Z

[INFO] -----

--> 282ac96c486a

[2/2] STEP 1/6: FROM registry.access.redhat.com/ubi8/openjdk-11

[2/2] STEP 2/6: COPY --from=stage1 target/*.jar app.jar

[2/2] STEP 3/6: EXPOSE 8080

[2/2] STEP 4/6: ENTRYPOINT ['java','-jar','app.jar']

[2/2] STEP 5/6: ENV 'OPENSHIFT_BUILD_NAME="hello-1" "OPENSHIFT_BUILD_NAMESPACE="jegan" "OPENSHIFT_BUILD_SOURCE="https://github.com/tektutor/openshift-tekn-june2024.git" "OPENSHIFT_BUILD_REFERENCE="main" "OPENSHIFT_BUILD_COMMIT="d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b"

[2/2] STEP 6/6: LABEL "io.openshift.build.commit.author="Jeganathan Swaminathan <mail2jegan@gmail.com>" "io.openshift.build.commit.date="Mon Jun 17 12:31:09 2024 +0530" "io.openshift.build.commit.id="d040a084a8c49f2fa26f817e0cd9f8cc1ed7ac3b" "io.openshift.build.commit.message="Update buildconfig-push-to-artifactory.yml" "io.openshift.build.commit.ref="main" "io.openshift.build.name="hello-1" "io.openshift.build.namespace="jegan" "io.openshift.build.source-context-dir="Day6/BuildConfig" "io.openshift.build.source-location="https://github.com/tektutor/openshift-tekn-june2024.git"

[2/2] COMMIT temp.builder.openshift.io/jegan/hello-1:6b9ae299

--> 20a199c8ff22

Successfully tagged temp.builder.openshift.io/jegan/hello-1:6b9ae299

20a199c8ff22b056aafe92d87047ef401af15e8a0aab81c07e325d5f7593d67

Pushing image image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello:latest ...

Getting image source signatures

Copying blob sha256:6bb8b50357633e3d0a7b9e9bed5b614a06bab6cd3a569d0326d73ab8b0c0bee0b

Copying blob sha256:8393debac2ec316f87eab40a0f5451d57df2863f844f545d40447e46e025c9663

Copying blob sha256:1846180440d3041e084880b0bb735f024f3bc5e2d39f1ab4ad490f630863d903

Copying config sha256:20a199c8ff22b056aafe92d87047ef401af15e8a0aab81c07e325d5f7593d67

Writing manifest to image destination

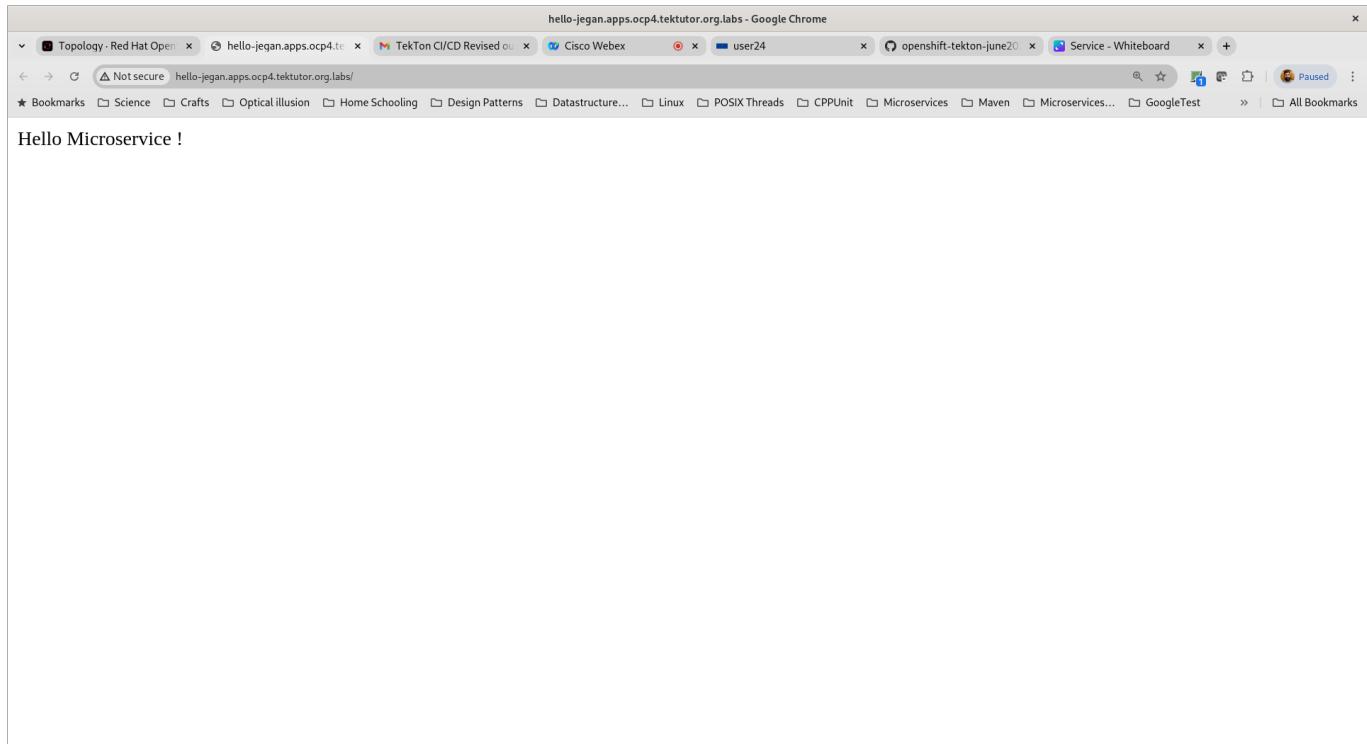
Successfully pushed image-registry.openshift-image-registry.svc:5000/jegan/tektutor-spring-hello@sha256:12fabab3b75b0db6670d473289092eed3bd277828988e1d001cfdfb79b190fa4

Push successful

jegan@tektutor.org > /-openshift-tekn-june2024/Day6/BuildConfig > ⚡ main •

```
jegan@tektutor.org
deployment.apps/hello 1/3 3 1 7s
NAME READY STATUS RESTARTS AGE
pod/hello-1-build 0/1 Completed 0 6m11s
pod/hello-6b994499c8-jpzfv 1/1 Running 0 7s
pod/hello-6b994499c8-phjmc 0/1 ContainerCreating 0 7s
pod/hello-6b994499c8-qgh7f 0/1 ContainerCreating 0 7s
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get po -w
NAME READY STATUS RESTARTS AGE
hello-1-build 0/1 Completed 0 6m16s
hello-6b994499c8-jpzfv 1/1 Running 0 12s
hello-6b994499c8-phjmc 1/1 Running 0 12s
hello-6b994499c8-qgh7f 1/1 Running 0 12s
^C
x jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● ls
buildconfig-pushto-artifactory.yml Dockerfile hello-route.yml imagestream.yml pom.xml
buildconfig.yml hello-deploy.yml hello-svc.yml Jenkinsfile src
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● vim hello-vc
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● vim hello-route.yml
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc apply -f hello-svc.yml
service/hello created
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc apply -f hello-route.yml
route.route.openshift.io/hello created
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● oc get svc,route
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/hello ClusterIP 172.30.64.115 <none> 8080/TCP 9s
NAME HOST/PORT PATH SERVICES PORT TERMINATION WILDCARD
route.route.openshift.io/hello hello-jegan.apps.ocp4.tektutor.org.labs hello 8080 None
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● curl http://hello-jegan.apps.ocp4.tektutor.org.labs:8080
curl: (7) Failed to connect to hello-jegan.apps.ocp4.tektutor.org.labs port 8080: Connection refused
x jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ● curl http://hello-jegan.apps.ocp4.tektutor.org.labs
Hello Microservice !%
jegan@tektutor.org > ~/openshift-tekton-june2024/Day6/BuildConfig > main ●
```

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 'Topology' option is selected. In the center, a large circular icon represents the application, with a red 'H' symbol inside. Below the icon, there's a button labeled 'D hello' and three vertical dots. To the right, a detailed view of the 'hello' deployment is shown. It has a summary section with '3 Pods'. Under the 'Details' tab, it lists the deployment configuration: Name: hello, Namespace: jegan, Labels: app=hello, Update strategy: RollingUpdate, Max unavailable: 25% of 3 pods, Max surge: 25% greater than 3 pods, and Progress deadline seconds: 600 seconds. A note at the top right says 'Container hello does not have health checks to ensure your application is running correctly. Add health checks.'



Info - What is Openshift Config Map?

- it is map data structure
- we can store key-value pairs
- the data is organized based on the key
- we can use configmap to store several configuration data
- Examples
 - We could use configmap to store environment variables like
JAVA_HOME=/usr/lib/jvm/jdk8
M2_HOME=/usr/share/maven
 - Developers use xml, properties file to store configuration data, which could be stored within K8s as config map
 - this is useful to store non-sensitive data as value stored against the key are visible in plain text

Info - What is Openshift Secret?

- internally secret uses the map data structure just like configmap
- the key is stored as plain text, while the value is stored as base64 encoded values, hence somewhat secured
- this can be used to store sensitive data like password, login credentials, certs, etc.,
- the commercial alternate for this is HashiCorp Vault

Lab - Using ConfigMap and Secrets in wordpress & mariadb multi-pod application deployment

```
cd ~/openshift-tektton-june2024
git pull
cd Day6/wordpress-with-configmap-and-secrets
./deploy.sh
```

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

Find your base domain of your openshift cluster

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

Expected output

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

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

First, let's generate a private key

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

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

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

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

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

Expected output

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

```
service/nginx exposed
```

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

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

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

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

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

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

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

```
jegan@tektutor.org ~ mkdir certs
jegan@tektutor.org ~ cd certs
jegan@tektutor.org ~/certs openssl version
OpenSSL 3.0.0 7 sep 2021 (Library: OpenSSL 3.0.7 1 Nov 2022)
jegan@tektutor.org ~/certs oc get ingresses.config/cluster -o jsonpath={.spec.domain}

apps.ocp4.tektutor.org.labs%
jegan@tektutor.org ~/certs #Generate private key
jegan@tektutor.org ~/certs openssl genrsa -out key.key

jegan@tektutor.org ~/certs ls
key.key
jegan@tektutor.org ~/certs #Generate public key using the private key generated in previous step
jegan@tektutor.org ~/certs openssl req -new -key key.key -out csr.csr -subj="/CN=hello-jegan.apps.ocp4.tektutor.org.labs"
jegan@tektutor.org ~/certs ls -l
total 8
-rw-r--r-- 1 jegan jegan 932 May 30 17:24 csr.csr
-rw----- 1 jegan jegan 1704 May 30 17:23 key.key
jegan@tektutor.org ~/certs #Sign the certificate
jegan@tektutor.org ~/certs openssl x509 -req -in csr.csr -signkey key.key -out crt.crt

jegan@tektutor.org ~/certs ls
crt.crt csr.csr key.key
jegan@tektutor.org ~/certs
```

```
jegan@tektutor.org ~/certs oc delete project jegan
project.project.openshift.io "jegan" deleted
jegan@tektutor.org ~/certs oc new-project jegan
Already on project "jegan" 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 ~/certs oc new-app --name=hello tektutor/spring-ms:1.0
--> Found container image 9175b94 (22 months old) from Docker Hub for "tektutor/spring-ms:1.0"

* An image stream tag will be created as "hello:1.0" that will track this image

--> Creating resources ...
imagestream.image.openshift.io "hello" created
deployment.apps "hello" created
service "hello" 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/hello'
Run 'oc status' to view your app.
jegan@tektutor.org ~/certs
```

```
jegan@tektutor.org ~/certs oc new-app --name=hello tektutor/spring-ms:1.0
--> Found container image 9175b94 (22 months old) from Docker Hub for "tektutor/spring-ms:1.0"

* An image stream tag will be created as "hello:1.0" that will track this image

--> Creating resources ...
imagestream.image.openshift.io "hello" created
deployment.apps "hello" created
service "hello" 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/hello'
Run 'oc status' to view your app.
jegan@tektutor.org ~/certs
```

```
jegan@tektutor.org ~ /certs oc new-app --name=hello tektutor/spring-ms:1.0
--> Found container image 9175b94 (22 months old) from Docker Hub for "tektutor/spring-ms:1.0"

* An image stream tag will be created as "hello:1.0" that will track this image

--> Creating resources ...
imagestream.image.openshift.io "hello" created
deployment.apps "hello" created
service "hello" 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/hello'
Run 'oc status' to view your app.
jegan@tektutor.org ~/certs oc create route edge --service hello --hostname hello-jegan.apps.ocp4.tektutor.org.labs --key key.key --cert crt.crt
route/hello created
jegan@tektutor.org ~/certs oc get route
NAME      HOST/PORT          PATH  SERVICES   PORT  TERMINATION  WILDCARD
hello     hello-jegan.apps.ocp4.tektutor.org.labs    hello       8080-tcp  edge        None
jegan@tektutor.org ~/certs curl -k https://hello-jegan.apps.ocp4.tektutor.org.labs
Greetings from Spring Boot!
jegan@tektutor.org ~/certs
```

Topology - Red Hat OpenShift - Google Chrome

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

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

Project: jegan Application: All applications View shortcuts

Developer

+Add

Topology

Observe

Search

Functions

Builds

Pipelines

Helm

Project

ConfigMaps

Secrets

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

A circular icon with a red 'E' symbol inside, with a blue border and a white center. A small edit icon is to its right.

D hello

Search icons: magnifying glass, refresh, clear, etc.

