### **Getting Started with OpenShift Pipelines**

* Install the OpenShift Pipelines operator
* Deploy a partial application
* Create reusable Tekton Tasks
* Create a Tekton Pipeline
* Create PipelineResources
* Trigger the created pipeline to finish your application deployment.

OpenShift Pipelines is a cloud-native, continuous integration and delivery (CI/CD) solution for building pipelines using [Tekton](https://tekton.dev). Tekton is a flexible, Kubernetes-native, open-source CI/CD framework that enables automating deployments across multiple platforms (e.g. Kubernetes, serverless, VMs, and so forth) by abstracting away the underlying details.

OpenShift Pipelines features:

* Standard CI/CD pipeline definition based on Tekton
* Build container images with tools such as [Source-to-Image (S2I)](https://docs.openshift.com/container-platform/latest/builds/understanding-image-builds.html#build-strategy-s2i_understanding-image-builds) and [Buildah](https://buildah.io/)
* Deploy applications to multiple platforms such as Kubernetes, serverless, and VMs
* Easy to extend and integrate with existing tools
* Scale pipelines on-demand
* Portable across any Kubernetes platform
* Designed for microservices and decentralized teams
* Integrated with the OpenShift Developer Console
* **Step 1 - Install the Pipelines Operator**

**Step 1 - Install the Pipelines Operator$ echo -e "Welcome to your interactive environment. OpenShift is configured and ready to use."**

**Welcome to your interactive environment. OpenShift is configured and ready to use.**

**$ oc login -u admin -p admin**

**Login successful.**

**You have access to 51 projects, the list has been suppressed. You can list all projects with 'oc projects'**

**Using project "default".**

**$ oc describe packagemanifest openshift-pipelines-operator -n openshift-marketplace**

**Name: openshift-pipelines-operator**

**Namespace: openshift-marketplace**

**Labels: catalog=community-operators**

**catalog-namespace=openshift-marketplace**

**olm-visibility=hidden**

**openshift-marketplace=true**

**opsrc-datastore=true**

**opsrc-owner-name=community-operators**

**opsrc-owner-namespace=openshift-marketplace**

**opsrc-provider=community**

**provider=Red Hat**

**provider-url=**

**Annotations: <none>**

**API Version: packages.operators.coreos.com/v1**

**Kind: PackageManifest**

**Metadata:**

**Creation Timestamp: 2019-12-12T06:58:03Z**

**Self Link: /apis/packages.operators.coreos.com/v1/namespaces/openshift-marketplace/packagemanifests/openshift-pipelines-operator**

**Spec:**

**Status:**

**Catalog Source: community-operators**

**Catalog Source Display Name: Community Operators**

**Catalog Source Namespace: openshift-marketplace**

**Catalog Source Publisher: Red Hat**

**Channels:**

**Current CSV: openshift-pipelines-operator.v0.10.8**

**Current CSV Desc:**

**Annotations:**

**Alm - Examples: [**

**{**

**"apiVersion": "operator.tekton.dev/v1alpha1",**

**"kind": "Config",**

**"metadata": {**

**"name": "name.must.be-cluster"**

**},**

**"spec": {**

**"targetNamespace": "openshift-pipelines"**

**}**

**}**

**]**

**Capabilities: Basic Install**

**Categories: Developer Tools, Integration & Delivery**

**Certified: false**

**Container Image: quay.io/openshift-pipeline/openshift-pipelines-operator:v0.10.8**

**Created At: 2019-03-15T19:44:21Z**

**Description: OpenShift Pipelines is a cloud-native CI/CD solution for building pipelines using Tekton concepts which run natively on OpenShift and Kubernetes.**

**operators.operatorframework.io/internal-objects: ["config.operator.tekton.dev"]**

**Repository: https://github.com/openshift/tektoncd-pipeline-operator**

**Support: Red Hat, Inc.**

**Apiservicedefinitions:**

**Customresourcedefinitions:**

**Owned:**

**Description: OpenShift Pipelines is a cloud-native CI/CD solution for building pipelines using Tekton concepts which run natively on OpenShift and Kubernetes.**

**Display Name: OpenShift Pipelines Config**

**Kind: Config**

**Name: config.operator.tekton.dev**

**Version: v1alpha1**

**Description: OpenShift Pipelines is a cloud-native continuous integration and delivery**

**(CI/CD) solution for building pipelines using [Tekton](https://tekton.dev).**

**Tekton is a flexible Kubernetes-native open-source CI/CD framework which**

**enables automating deployments across multiple platforms (Kubernetes,**

**serverless, VMs, etc) by abstracting away the underlying details.**

**## Features**

**\* Standard CI/CD pipelines definition**

**\* Build images with Kubernetes tools such as S2I, Buildah, Buildpacks, Kaniko, etc**

**\* Deploy applications to multiple platforms such as Kubernetes, serverless and VMs**

**\* Easy to extend and integrate with existing tools**

**\* Scale pipelines on-demand**

**\* Portable across any Kubernetes platform**

**\* Designed for microservices and decentralised team**

**\* Integrated with OpenShift Developer Console**

**## Installation**

**\_OpenShift Pipelines Operator\_ gets installed into a single namespace which would then install \_OpenShift Pipelines\_ into the same namespace. \_OpenShift Pipelines\_ is however cluster-wide and can run pipelines created in any namespace.**

**## Getting Started**

**In order to get familiar with \_OpenShift Pipelines\_ concepts and create your first pipeline, follow the [OpenShift Pipelines Tutorial](https://github.com/openshift/pipelines-tutorial).**

**### CLI**

**Tekton Pipelines cli project provides a CLI for interacting with OpenShift Pipelines.**

**[Download Tekton CLI](https://github.com/tektoncd/cli/releases/latest)**

**Display Name: OpenShift Pipelines Operator**

**Install Modes:**

**Supported: false**

**Type: OwnNamespace**

**Supported: false**

**Type: SingleNamespace**

**Supported: false**

**Type: MultiNamespace**

**Supported: true**

**Type: AllNamespaces**

**Provider:**

**Name: Red Hat**

**Version: 0.10.8**

**Name: canary**

**Current CSV: openshift-pipelines-operator.v0.10.7**

**Current CSV Desc:**

**Annotations:**

**Alm - Examples: [**

**{**

**"apiVersion": "operator.tekton.dev/v1alpha1",**

**"kind": "Config",**

**"metadata": {**

**"name": "name.must.be-cluster"**

**},**

**"spec": {**

**"targetNamespace": "openshift-pipelines"**

**}**

**}**

**]**

**Capabilities: Basic Install**

**Categories: Developer Tools, Integration & Delivery**

**Certified: false**

**Container Image: quay.io/openshift-pipeline/openshift-pipelines-operator:v0.10.7**

**Created At: 2019-03-15T19:44:21Z**

**Description: OpenShift Pipelines is a cloud-native CI/CD solution for building pipelines using Tekton concepts which run natively on OpenShift and Kubernetes.**

**operators.operatorframework.io/internal-objects: ["config.operator.tekton.dev"]**

**Repository: https://github.com/openshift/tektoncd-pipeline-operator**

**Support: Red Hat, Inc.**

**Apiservicedefinitions:**

**Customresourcedefinitions:**

**Owned:**

**Description: OpenShift Pipelines is a cloud-native CI/CD solution for building pipelines using Tekton concepts which run natively on OpenShift and Kubernetes.**

**Display Name: OpenShift Pipelines Config**

**Kind: Config**

**Name: config.operator.tekton.dev**

**Version: v1alpha1**

**Description: OpenShift Pipelines is a cloud-native continuous integration and delivery**

**(CI/CD) solution for building pipelines using [Tekton](https://tekton.dev).**

**Tekton is a flexible Kubernetes-native open-source CI/CD framework which**

**enables automating deployments across multiple platforms (Kubernetes,**

**serverless, VMs, etc) by abstracting away the underlying details.**

**## Features**

**\* Standard CI/CD pipelines definition**

**\* Build images with Kubernetes tools such as S2I, Buildah, Buildpacks, Kaniko, etc**

**\* Deploy applications to multiple platforms such as Kubernetes, serverless and VMs**

**\* Easy to extend and integrate with existing tools**

**\* Scale pipelines on-demand**

**\* Portable across any Kubernetes platform**

**\* Designed for microservices and decentralised team**

**\* Integrated with OpenShift Developer Console**

**## Installation**

**\_OpenShift Pipelines Operator\_ gets installed into a single namespace which would then install \_OpenShift Pipelines\_ into the same namespace. \_OpenShift Pipelines\_ is however cluster-wide and can run pipelines created in any namespace.**

**## Getting Started**

**In order to get familiar with \_OpenShift Pipelines\_ concepts and create your first pipeline, follow the [OpenShift Pipelines Tutorial](https://github.com/openshift/pipelines-tutorial).**

**### CLI**

**Tekton Pipelines cli project provides a CLI for interacting with OpenShift Pipelines.**

**[Download Tekton CLI](https://github.com/tektoncd/cli/releases/latest)**

**Display Name: OpenShift Pipelines Operator**

**Install Modes:**

**Supported: false**

**Type: OwnNamespace**

**Supported: false**

**Type: SingleNamespace**

**Supported: false**

**Type: MultiNamespace**

**Supported: true**

**Type: AllNamespaces**

**Provider:**

**Name: Red Hat**

**Version: 0.10.7**

**Name: dev-preview**

**Default Channel: dev-preview**

**Package Name: openshift-pipelines-operator**

**Provider:**

**Name: Red Hat**

**Events: <none>**

**$ oc apply -f ./operator/subscription.yaml**

**subscription.operators.coreos.com/openshift-pipelines-operator created**

**$ until oc api-resources --api-group=tekton.dev | grep tekton.dev &> /dev/null**

**> do**

**> echo "Operator installation in progress..."**

**> sleep 5**

**> done**

**Operator installation in progress...**

**Operator installation in progress...**

**Operator installation in progress...**

**$**

**$ echo "Operator ready"**

**Operator ready**

**$ until oc api-resources --api-group=tekton.dev | grep tekton.dev &> /dev/null**

**> do**

**> echo "Operator installation in progress..."**

**> sleep 5**

**> done**

**$**

**$ echo "Operator ready"**

**Operator ready**

**$ oc api-resources --api-group=tekton.dev**

**NAME SHORTNAMES APIGROUP NAMESPACED KIND**

**clustertasks tekton.dev false ClusterTask**

**conditions tekton.dev true Condition**

**pipelineresources tekton.dev true PipelineResource**

**pipelineruns pr,prs tekton.dev true PipelineRun**

**pipelines tekton.dev true Pipeline**

**taskruns tr,trs tekton.dev true TaskRun**

**tasks tekton.dev true Task**

**$ oc auth can-i create pipeline.tekton.dev**

**yes**

**$ oc auth can-i create Pipeline**

**yes**

**$ oc auth can-i create PipelineResource**

**yes**

**$**

#### Step 2 - Pipeline Service Account

**$ oc get serviceaccount pipeline**

**NAME SECRETS AGE**

**pipeline 2 3m37s**

**$**

**$**

#### $Step 3 - Deploying a Sample Application

#### Step 3 - Deploying a Sample Application

#### Step 3 - Deploying a Sample Application

**oc get dc/mongodb**

**oc get dc/mongodb$ oc new-project lab-tekton**

**Now using project "lab-tekton" on server "https://openshift:6443".**

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

**oc new-app django-psql-example**

**to build a new example application in Python. Or use kubectl to deploy a simple Kubernetes application:**

**kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-node**

**$ oc create -f sampleapp/sampleapp.yaml**

**imagestream.image.openshift.io/nodejs-ex created**

**deploymentconfig.apps.openshift.io/nodejs-ex created**

**service/nodejs-ex created**

**route.route.openshift.io/nodejs-ex created**

**$ oc new-app centos/mongodb-36-centos7 -e MONGODB\_USER=admin MONGODB\_DATABASE=mongodb MONGODB\_PASSWORD=secret MONGODB\_ADMIN\_PASSWORD=super-secret**

**--> Found container image 7b2fb5f (7 weeks old) from Docker Hub for "centos/mongodb-36-centos7"**

**MongoDB 3.6**

**-----------**

**MongoDB (from humongous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. This container image contains programs to run mongod server.**

**Tags: database, mongodb, rh-mongodb36**

**\* An image stream tag will be created as "mongodb-36-centos7:latest" that will track this image**

**\* This image will be deployed in deployment config "mongodb-36-centos7"**

**\* Port 27017/tcp will be load balanced by service "mongodb-36-centos7"**

**\* Other containers can access this service through the hostname "mongodb-36-centos7"**

**\* This image declares volumes and will default to use non-persistent, host-local storage.**

**You can add persistent volumes later by running 'oc set volume dc/mongodb-36-centos7 --add ...'**

**--> Creating resources ...**

**imagestream.image.openshift.io "mongodb-36-centos7" created**

**deploymentconfig.apps.openshift.io "mongodb-36-centos7" created**

**service "mongodb-36-centos7" 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 svc/mongodb-36-centos7'**

**Run 'oc status' to view your app.**

**$ oc get services**

**NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE**

**mongodb-36-centos7 ClusterIP 172.30.157.149 <none> 27017/TCP 5s**

**nodejs-ex ClusterIP 172.30.161.18 <none> 8080/TCP,8443/TCP,8778/TCP 12s**

**$ oc set env dc/nodejs-ex MONGO\_URL="mongodb://admin:secret@mongodb-36-centos7:27017/mongodb"**

**deploymentconfig.apps.openshift.io/nodejs-ex updated**

**$ oc get pods**

**NAME READY STATUS RESTARTS AGE**

**mongodb-36-centos7-1-7s745 1/1 Running 0 42s**

**mongodb-36-centos7-1-deploy 0/1 Completed 0 55s**

**$ oc get dc/mongodb**

**Error from server (NotFound): deploymentconfigs.apps.openshift.io "mongodb" not found**

**$ 5~**

#### Step 4 - Create Tasks

$ oc apply -f ./operator/subscription.yaml

subscription.operators.coreos.com/openshift-pipelines-operator created

$ until oc api-resources --api-group=tekton.dev | grep tekton.dev &> /dev/null

> do

> echo "Operator installation in progress..."

> sleep 5

> done

Operator installation in progress...

Operator installation in progress...

Operator installation in progress...

Operator installation in progress...

$

$ echo "Operator ready"

Operator ready

$ oc api-resources --api-group=tekton.dev

NAME SHORTNAMES APIGROUP NAMESPACED KIND

clustertasks tekton.dev false ClusterTask

conditions tekton.dev true Condition

pipelineresources tekton.dev true PipelineResource

pipelineruns pr,prs tekton.dev true PipelineRun

pipelines tekton.dev true Pipeline

taskruns tr,trs tekton.dev true TaskRun

tasks tekton.dev true Task

$ oc auth can-i create pipeline.tekton.dev

yes

$ oc auth can-i create Pipeline

yes

$ oc auth can-i create Task

yes

$ oc auth can-i create PipelineResource

yes

$ oc auth can-i create PipelineRun

yes

$ oc get serviceaccount pipeline

NAME SECRETS AGE

pipeline 2 32s

$ oc new-project lab-tekton

Now using project "lab-tekton" on server "https://openshift:6443".

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

oc new-app django-psql-example

to build a new example application in Python. Or use kubectl to deploy a simple Kubernetes application:

kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-node

$ oc create -f sampleapp/sampleapp.yaml

imagestream.image.openshift.io/nodejs-ex created

deploymentconfig.apps.openshift.io/nodejs-ex created

service/nodejs-ex created

route.route.openshift.io/nodejs-ex created

$ oc new-app centos/mongodb-36-centos7 -e MONGODB\_USER=admin MONGODB\_DATABASE=mongodb MONGODB\_PASSWORD=secret MONGODB\_ADMIN\_PASSWORD=super-secret

--> Found container image 7b2fb5f (7 weeks old) from Docker Hub for "centos/mongodb-36-centos7"

MongoDB 3.6

-----------

MongoDB (from humongous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. This container imagecontains programs to run mongod server.

Tags: database, mongodb, rh-mongodb36

\* An image stream tag will be created as "mongodb-36-centos7:latest" that will track this image

\* This image will be deployed in deployment config "mongodb-36-centos7"

\* Port 27017/tcp will be load balanced by service "mongodb-36-centos7"

\* Other containers can access this service through the hostname "mongodb-36-centos7"

\* This image declares volumes and will default to use non-persistent, host-local storage.

You can add persistent volumes later by running 'oc set volume dc/mongodb-36-centos7 --add ...'

--> Creating resources ...

imagestream.image.openshift.io "mongodb-36-centos7" created

deploymentconfig.apps.openshift.io "mongodb-36-centos7" created

service "mongodb-36-centos7" 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 svc/mongodb-36-centos7'

Run 'oc status' to view your app.

$ oc get services

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

mongodb-36-centos7 ClusterIP 172.30.124.214 <none> 27017/TCP 5s

nodejs-ex ClusterIP 172.30.223.182 <none> 8080/TCP,8443/TCP,8778/TCP 11s

$ oc set env dc/nodejs-ex MONGO\_URL="mongodb://admin:secret@mongodb-36-centos7:27017/mongodb"

deploymentconfig.apps.openshift.io/nodejs-ex updated

$ oc get pods

NAME READY STATUS RESTARTS AGE

mongodb-36-centos7-1-deploy 1/1 Running 0 25s

mongodb-36-centos7-1-p9sr6 0/1 ContainerCreating 0 12s

$ oc create -f tektontasks/s2i-nodejs-task.yaml

task.tekton.dev/s2i-nodejs created

$ oc create -f tektontasks/openshift-client-task.yaml

task.tekton.dev/openshift-client created

$ tkn task ls

NAME AGE

openshift-client 3 seconds ago

s2i-nodejs 10 seconds ago

$ oc create -f pipeline/deploy-pipeline.yaml

pipeline.tekton.dev/deploy-pipeline created

$ tkn pipeline ls

NAME AGE LAST RUN STARTED DURATION STATUS

deploy-pipeline 2 seconds ago --- --- --- ---

$ oc create -f resources/git-pipeline-resource.yaml

pipelineresource.tekton.dev/nodejs-ex-git created

$ oc create -f resources/image-pipeline-resource.yaml

pipelineresource.tekton.dev/nodejs-ex-image created

$ tkn resource ls

NAME TYPE DETAILS

nodejs-ex-git git url: https://github.com/sclorg/nodejs-ex

nodejs-ex-image image url: image-registry.openshift-image-registry.svc:5000/lab-tekton/nodejs-ex:latest

$ tkn resource describe nodejs-ex-git

Name: nodejs-ex-git

Namespace: lab-tekton

PipelineResource Type: git

Params

NAME VALUE

url https://github.com/sclorg/nodejs-ex

Secret Params

No secret params

$ tkn resource describe nodejs-ex-image

Name: nodejs-ex-image

Namespace: lab-tekton

PipelineResource Type: image

Params

NAME VALUE

url image-registry.openshift-image-registry.svc:5000/lab-tekton/nodejs-ex:latest

Secret Params

No secret params

$ tkn pipeline start deploy-pipeline \

> -r app-git=nodejs-ex-git \

> -r app-image=nodejs-ex-image \

> -s pipeline

Pipelinerun started: deploy-pipeline-run-rkwrk

Showing logs...

tkn pr ls

[build : create-dir-image-jcrfz] {"level":"warn","ts":1586169599.745336,"logger":"fallback-logger","caller":"logging/config.go:69","msg":"Fetch GitHub commit ID from kodata failed: \"KO\_DATA\_PATH\" does not exist or is empty"}

[build : create-dir-image-jcrfz] {"level":"info","ts":1586169599.75175,"logger":"fallback-logger","caller":"bash/main.go:64","msg":"Successfully executed command \"sh -c mkdir -p /workspace/output/image\"; output "}

[build : git-source-nodejs-ex-git-98gp9] {"level":"warn","ts":1586169600.276721,"logger":"fallback-logger","caller":"logging/config.go:69","msg":"Fetch GitHub commit ID from kodata failed: \"KO\_DATA\_PATH\" does not exist or is empty"}

[build : git-source-nodejs-ex-git-98gp9] {"level":"info","ts":1586169601.9952238,"logger":"fallback-logger","caller":"git/git.go:103","msg":"Successfully cloned https://github.com/sclorg/nodejs-ex @ master in path /workspace/source"}

[build : generate] Application dockerfile generated in /gen-source/Dockerfile.gen

[build : build] STEP 1: FROM centos/nodejs-12-centos7

[build : build] Getting image source signatures

[build : build] Copying blob sha256:ab5ef0e5819490abe86106fd9f4381123e37a03e80e650be39f7938d30ecb530

[build : build] Copying blob sha256:9a62185b95350e0cf3dba655a9461824080fd70a8718f4655f04fa8dde3a2b44

[build : build] Copying blob sha256:0ca27d7286e3ce558e7898856e42b8db25c919084e6261e95ec8291376857e79

[build : build] Copying blob sha256:211f13f4e617a585a4c3a9cf73734c780febbec14dd3785ab75ffdb100295d57

[build : build] Copying blob sha256:d296a828de14f5d33a5db8f27c698f9d90fdddcb446090b591c197ed89bf67be

[build : build] Copying blob sha256:6916189c40907fdd6e44b7798cbd19e959b4087fe909ecabade99905586929b0

[build : build] Copying blob sha256:b66848a4a2ebfab0dccc592b85313f90a479a26d128de096ca44b36415acdcd2

[build : build] Copying blob sha256:ffc6d8ab562d169a6e6bacdc29e3a99c7366d3a50468be74f0dcbb132b6004fa

[build : build] Copying blob sha256:5a868fdae9eccb17fe8709a3a025e9a21149190f810277b8b0f87b0be2ed4aab

[build : build] Copying config sha256:0a69a0e307683f37316cb38134a4fe0e26f7ba82366c663ff763e2cd314fea51

[build : build] Writing manifest to image destination

[build : build] Storing signatures

[build : build] STEP 2: LABEL "io.openshift.s2i.build.image"="centos/nodejs-12-centos7" "io.openshift.s2i.build.source-location"="."

[build : build] a4c8784df0731e44d234d054ee51da1ee22b33299e633cbce8e073835a7a3ba3

[build : build] STEP 3: USER root

[build : build] adb70fb6faec30eb756f6958945834cf041064940f102fb3795abe35c02a71fc

[build : build] STEP 4: COPY upload/src /tmp/src

[build : build] 3aac03417688fd8ec872b47b6776aa215def7c5f87d65f647153c8accabe06d7

[build : build] STEP 5: RUN chown -R 1001:0 /tmp/src

[build : build] c7f223718742cc356607000b4556a1b630095992fdf80f31ce5d88025f9cada5

[build : build] STEP 6: USER 1001

[build : build] bd8a4dcf69593a774d492de8f2d9bce12559a392d565477ad801a751470a4bee

[build : build] STEP 7: RUN /usr/libexec/s2i/assemble

[build : build] ---> Installing application source ...

[build : build] ---> Installing all dependencies

[build : build] npm WARN deprecated to-iso-string@0.0.2: to-iso-string has been deprecated, use @segment/to-iso-string instead.

[build : build] npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)

[build : build] npm WARN deprecated jade@0.26.3: Jade has been renamed to pug, please install the latest version of pug instead of jade

[build : build] npm WARN deprecated mkdirp@0.3.0: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)

[build : build] npm WARN deprecated minimatch@0.3.0: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue

[build : build]

[build : build] > ejs@2.7.4 postinstall /opt/app-root/src/node\_modules/ejs

[build : build] > node ./postinstall.js

[build : build]

[build : build] Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)

[build : build]

[build : build] npm notice created a lockfile as package-lock.json. You should commit this file.

[build : build] added 121 packages from 342 contributors and audited 205 packages in 21.398s

[build : build] found 10 vulnerabilities (3 low, 2 moderate, 4 high, 1 critical)

[build : build] run `npm audit fix` to fix them, or `npm audit` for details

[build : build] ---> Building in production mode

[build : build] ---> Pruning the development dependencies

[build : build] audited 205 packages in 3.309s

[build : build] found 10 vulnerabilities (3 low, 2 moderate, 4 high, 1 critical)

[build : build] run `npm audit fix` to fix them, or `npm audit` for details

[build : build] /opt/app-root/src/.npm is not a mountpoint

[build : build] ---> Cleaning the npm cache /opt/app-root/src/.npm

[build : build] /tmp is not a mountpoint

[build : build] ---> Cleaning the /tmp/npm-\*

[build : build] 6380c85e998547c58150a9baf6982286068558afd42775099cb6149634561fb0

[build : build] STEP 8: CMD /usr/libexec/s2i/run

[build : build] STEP 9: COMMIT image-registry.openshift-image-registry.svc:5000/lab-tekton/nodejs-ex:latest

[build : build] 4c580aa11f746b699763d25ad67e1405e9ce7a054ff43587eedc09c12a629db1

[build : build] 4c580aa11f746b699763d25ad67e1405e9ce7a054ff43587eedc09c12a629db1

[build : push] Getting image source signatures

[build : push] Copying blob sha256:7f4829bde3419ae4b1374b4f4fab15f71fd69ac230b64509905d2d1c68760418

[build : push] Copying blob sha256:a5e478d4632e9df8ecc14ec0c31a8662bf1d185703b023e46fd3a68dd6dfe22b

[build : push] Copying blob sha256:2744ad4c43f00dca56f4f3e7d23fd643a9ca99d9f51f7c2293a95426d787bbdf

[build : push] Copying blob sha256:6ab5724e7e288c33c38e0369f39198205286c0caa36bc099d5cc008d10d9f3ea

[build : push] Copying blob sha256:a7909a534bf73758c923a53108dabafc995ae4c82ff501f126a7e78ba07087a9

[build : push] Copying blob sha256:77b174a6a187b610e4699546bd973a8d1e77663796e3724318a2a4b24cb07ea0

[build : push] Copying blob sha256:9beef8c1e6cd493ec149f7f48286aeec78132286b727a939a22e04be62c9461a

[build : push] Copying blob sha256:f27952b83afaaf9d0c26c25d18504561a54e76240364d0f6784f5c11dcb2fab6

[build : push] Copying blob sha256:3421ae71e861db7bfcf838e238a2427b7a623c42d9461ede73bc9ed8be5533ab

[build : push] Copying blob sha256:e1fa05df89a5fa993e2a61e1216bec70ca5ca01e55bdd4e43632f07396e8a239

[build : push] Copying blob sha256:7da920d26e1edac69b571983f8c83aa559cb0db9446fdc973e0c379f7211e229

[build : push] Copying blob sha256:5d993119f83398ed9b485f1e5e6cddf7c4a45f8d3f28340fce47f1a8d8521d61

[build : push] Copying config sha256:4c580aa11f746b699763d25ad67e1405e9ce7a054ff43587eedc09c12a629db1

[build : push] Writing manifest to image destination

[build : push] Copying config sha256:4c580aa11f746b699763d25ad67e1405e9ce7a054ff43587eedc09c12a629db1

[build : push] Writing manifest to image destination

[build : push] Storing signatures

[build : image-digest-exporter-m7qxt] {"level":"warn","ts":1586169890.153792,"logger":"fallback-logger","caller":"logging/config.go:69","msg":"Fetch GitHub commit ID from kodata failed: \"KO\_DATA\_PATH\" does not exist or is empty"}

[build : image-digest-exporter-m7qxt] {"level":"info","ts":1586169890.1541405,"logger":"fallback-logger","caller":"imagedigestexporter/main.go:58","msg":"ImageResource nodejs-ex-imagedoesn't have an index.json file: stat /builder/home/image-outputs/image/index.json: no suchfile or directory"}

[deploy : oc] deploymentconfig.apps.openshift.io/nodejs-ex rolled out

$

$

$ tkn pr ls

NAME STARTED DURATION STATUS

deploy-pipeline-run-rkwrk 7 minutes ago 7 minutes Succeeded

$

$

$ id

uid=0(root) gid=0(root) groups=0(root)

$ oc whoami

admin

$ oc projects

You have access to the following projects and can switch between them with 'oc project <projectname>':

default

kube-node-lease

kube-public

kube-system

\* lab-tekton

openshift

openshift-apiserver

openshift-apiserver-operator

openshift-authentication

openshift-authentication-operator

openshift-cloud-credential-operator

openshift-cluster-machine-approver

openshift-cluster-node-tuning-operator

openshift-cluster-samples-operator

openshift-cluster-storage-operator

openshift-cluster-version

openshift-config

openshift-config-managed

openshift-console

openshift-console-operator

openshift-controller-manager

openshift-controller-manager-operator

openshift-dns

openshift-dns-operator

openshift-etcd

openshift-image-registry

openshift-infra

openshift-ingress

openshift-ingress-operator

openshift-insights

openshift-kni-infra

openshift-kube-apiserver

openshift-kube-apiserver-operator

openshift-kube-controller-manager

openshift-kube-controller-manager-operator

openshift-kube-scheduler

openshift-kube-scheduler-operator

openshift-machine-api

openshift-machine-config-operator

openshift-marketplace

openshift-monitoring

openshift-multus

openshift-network-operator

openshift-node

openshift-openstack-infra

openshift-operator-lifecycle-manager

openshift-operators

openshift-pipelines

openshift-sdn

openshift-service-ca

openshift-service-ca-operator

openshift-service-catalog-apiserver-operator

openshift-service-catalog-controller-manager-operator

Using project "lab-tekton" on server "https://openshift:6443".

$ tkn pr ls

NAME STARTED DURATION STATUS

deploy-pipeline-run-rkwrk 14 minutes ago 7 minutes Succeeded

$

=======================FROM GUI OpenShift console MONGODB =======================

# DeploymentConfigDC[mongodb-36-centos7](https://console-openshift-console-2886795278-443-simba07.environments.katacoda.com/k8s/ns/lab-tekton/deploymentconfigs/mongodb-36-centos7)

##  Pods

* PodP[mongodb-36-centos7-1-p9sr6](https://console-openshift-console-2886795278-443-simba07.environments.katacoda.com/k8s/ns/lab-tekton/pods/mongodb-36-centos7-1-p9sr6)
* Running[View Logs](https://console-openshift-console-2886795278-443-simba07.environments.katacoda.com/k8s/ns/lab-tekton/pods/mongodb-36-centos7-1-p9sr6/logs)

## Builds

No Build Configs found for this resource.

## Services

* ServiceS[mongodb-36-centos7](https://console-openshift-console-2886795278-443-simba07.environments.katacoda.com/k8s/ns/lab-tekton/services/mongodb-36-centos7)
  + Service port: 27017-tcp
  + Pod Port: 27017

## Routes

No Routes found for this resource.

=================================FROM GUI NODE.JS o/p =======================

# Welcome to your Node.js application on OpenShift

## How to use this example application

For instructions on how to use this application with OpenShift, start by reading the [Developer Guide](http://docs.okd.io/latest/dev_guide/templates.html#using-the-quickstart-templates).

## Deploying code changes

The source code for this application is available to be forked from the [OpenShift GitHub repository](https://www.github.com/sclorg/nodejs-ex). You can configure a webhook in your repository to make OpenShift automatically start a build whenever you push your code:

1. From the Web Console homepage, navigate to your project
2. Click on Browse > Builds
3. Click the link with your BuildConfig name
4. Click the Configuration tab
5. Click the "Copy to clipboard" icon to the right of the "GitHub webhook URL" field
6. Navigate to your repository on GitHub and click on repository settings > webhooks > Add webhook
7. Paste your webhook URL provided by OpenShift in the "Payload URL" field
8. Change the "Content type" to 'application/json'
9. Leave the defaults for the remaining fields — that's it!

After you save your webhook, if you refresh your settings page you can see the status of the ping that Github sent to OpenShift to verify it can reach the server.

Note: adding a webhook requires your OpenShift server to be reachable from GitHub.

### Working in your local Git repository

If you forked the application from the OpenShift GitHub example, you'll need to manually clone the repository to your local system. Copy the application's source code Git URL and then run:

$ git clone <git\_url> <directory\_to\_create>

# Within your project directory

# Commit your changes and push to OpenShift

$ git commit -a -m 'Some commit message'

$ git push

After pushing changes, you'll need to manually trigger a build if you did not setup a webhook as described above.

## Managing your application

Documentation on how to manage your application from the Web Console or Command Line is available at the [Developer Guide](http://docs.okd.io/latest/dev_guide/overview.html).

### Web Console

You can use the Web Console to view the state of your application components and launch new builds.

### Command Line

With the [OpenShift command line interface](http://docs.okd.io/latest/cli_reference/overview.html) (CLI), you can create applications and manage projects from a terminal.

## Development Resources

* [OpenShift Documentation](http://docs.okd.io/latest/welcome/index.html)
* [Openshift Origin GitHub](https://github.com/openshift/origin)
* [Source To Image GitHub](https://github.com/openshift/source-to-image)
* [Getting Started with Node.js on OpenShift](http://docs.okd.io/latest/using_images/s2i_images/nodejs.html)
* [Stack Overflow questions for OpenShift](http://stackoverflow.com/questions/tagged/openshift)
* [Git documentation](http://git-scm.com/documentation)

## Request information

Page view count: 1

### DB Connection Info:

|  |  |
| --- | --- |
| Type: | MongoDB |
| URL: |  |