

Developer Experience

OpenShift Console

-- What's New in 4.8

Serena Chechile Nichols
@serenamarie125

Add page redesign

The screenshot displays the OpenShift Developer console interface. The top navigation bar shows the 'okd' logo, a hamburger menu, and user information (kube:admin). A blue banner at the top indicates the user is logged in as a temporary administrative user and provides a link to update the cluster OAuth configuration. The left sidebar contains navigation links for Developer, Topology, Monitoring, Search, Builds, Pipelines, Environments, Helm, Project, Pods, Services, and PipelineRuns. The main content area is titled 'Add' and prompts the user to select a way to create an Application, component or service from one of the options. A 'Details on' link is visible. The 'Add' page is overlaid on the main console interface, showing a 'Getting started resources' section with three columns of links and descriptions. The first column, 'Create applications using samples', includes links for 'Basic Quarkus' and 'Basic Spring Boot', and a 'View all samples' link. The second column, 'Build with guided documentation', includes links for 'Get started with Quarkus using s2i' and 'Get started with Spring', and a 'View all quick starts' link. The third column, 'Explore new developer features', includes links for 'Discover certified Helm Charts' and 'Start building your application quickly in topology', and a 'What's new in OpenShift 4.7' link. Below the 'Add' page, the main console interface shows a grid of cards for 'Operator backed', 'From Dockerfile', 'Upload JAR file', 'Samples', 'Helm Chart', and 'Event Source'.

Getting started resources ?

- Create applications using samples**
Choose a code sample to get started creating an application with.
[Basic Quarkus →](#)
[Basic Spring Boot →](#)
[View all samples](#)
- Build with guided documentation**
Follow guided documentation to build applications and familiarize yourself with key features.
[Get started with Quarkus using s2i →](#)
[Get started with Spring →](#)
[View all quick starts](#)
- Explore new developer features**
Explore new features and resources within the developer perspective.
[Discover certified Helm Charts →](#)
[Start building your application quickly in topology →](#)
[What's new in OpenShift 4.7 ↗](#)

Operator backed
Browse the catalog to discover and deploy operator managed services

From Dockerfile
Import your Dockerfile from your Git repository to be built and deployed

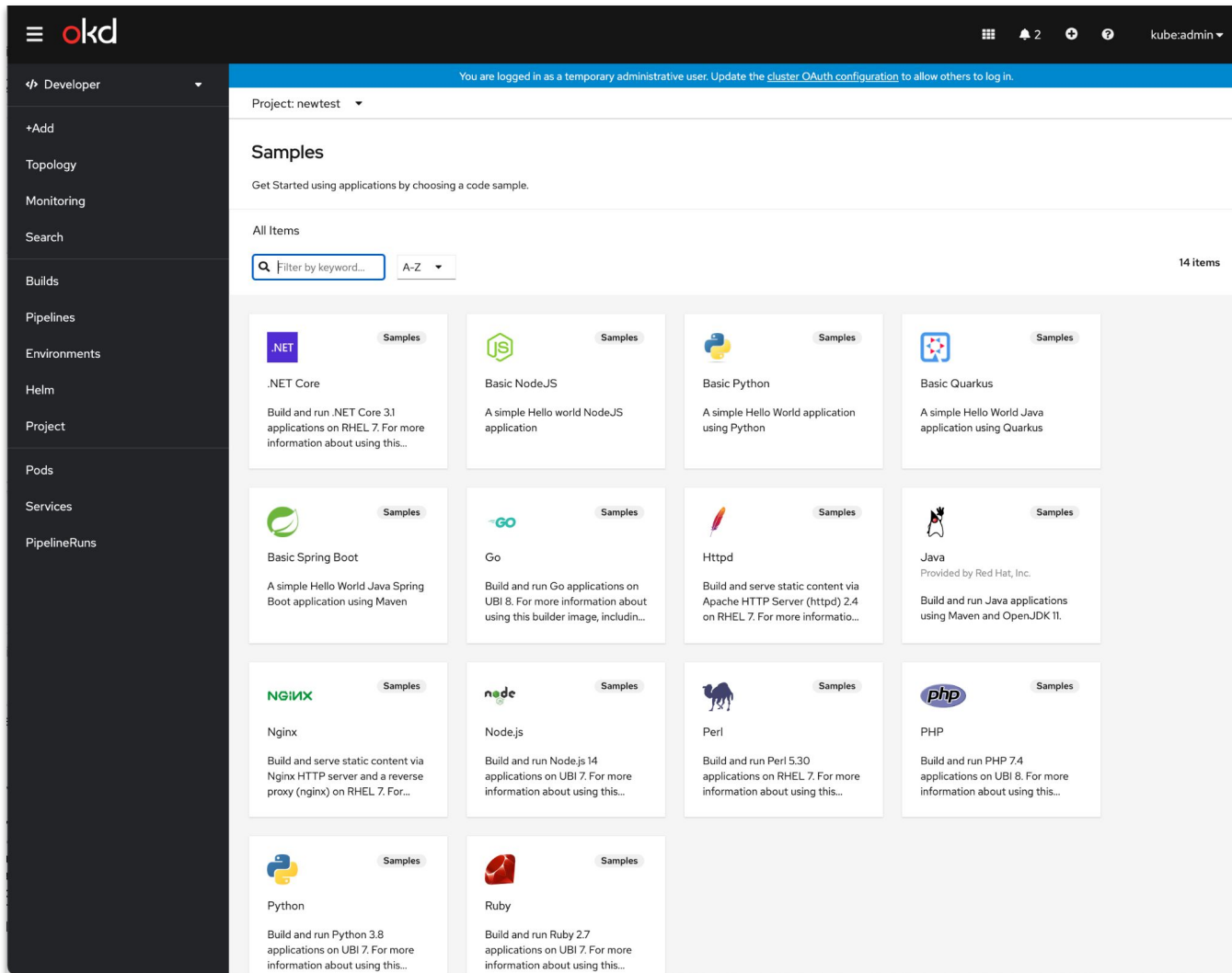
Upload JAR file
Upload a JAR file from your local desktop to OpenShift

Samples
Create an Application from a code sample

Helm Chart
Browse the catalog to discover and install Helm Charts

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

New samples available!



- New samples based on devfiles!
- Want to learn more about devfiles?

<https://devfile.io/>

Devfile catalog

- Our developer catalog now has a separate sub catalog for devfiles

The screenshot displays the OpenShift Developer Catalog interface. The top navigation bar includes the 'okd' logo, a user menu, and a notification bell. A blue banner at the top indicates the user is logged in as a temporary administrative user and provides a link to update the cluster OAuth configuration. The left sidebar contains a navigation menu with options like Developer, Topology, Monitoring, Search, Builds, Pipelines, Environments, Helm, Project, Pods, Services, and PipelineRuns. The main content area is titled 'Add' and includes a sub-header 'Select a way to create an Application, component or service from one of the options.' Below this, there are three columns of 'Getting started resources' with links to create applications using samples, build with guided documentation, and explore new developer features. The bottom section is divided into several cards: 'Developer Catalog' (All services, Database, Operator Backed, Helm Chart, Event Source), 'Git Repository' (From Git, From Devfile, From Dockerfile), 'Container images' (Deploy an existing image), 'Serverless' (Channel), 'From Local Machine' (Import YAML, Upload JAR file), 'Pipelines' (Create a Tekton Pipeline), and 'Samples' (Create an Application from a code sample).

Certified Helm Charts

- Certified Helm Charts from partners are now available
- A badge indicates the Chart is Certified

[Link to Helm Certification program announcement](#)

The screenshot shows the OpenShift Helm Charts interface. On the left is a sidebar with navigation options: Developer, +Add, Topology, Monitoring, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area displays the 'Helm Charts' section for the 'new-namespace' project. It lists 'All Items' with a search bar and a sort dropdown. A chart titled 'Vault v0.13.0' is highlighted with a pink box. This chart is provided by 'OpenShift Helm Charts' and is marked as 'Official HashiCorp Vault Chart'. A blue certification badge is visible next to the chart name. To the right of the chart list, a detailed view for 'Vault v0.13.0' is shown, also featuring the certification badge. This view includes an 'Install Helm Chart' button, a description of the chart, a README section, and prerequisites for installation.

Chart version
0.13.0

App version
1.7.3

Home page
<https://www.vaultproject.io>

Maintainers

Provider
OpenShift Helm Charts

Created at
1 janv. 1, 00:09

Description
Official HashiCorp Vault Chart

README
Vault Helm Chart
⚠ Please note: We take Vault's security and our users' trust very seriously. If you believe you have found a security issue in Vault Helm, *please responsibly disclose* by contacting us at security@hashicorp.com.

This repository contains the official HashiCorp Helm chart for installing and configuring Vault on Kubernetes. This chart supports multiple use cases of Vault on Kubernetes depending on the values provided.

For full documentation on this Helm chart along with all the ways you can use Vault with Kubernetes, please see the [Vault and Kubernetes documentation](#).

Prerequisites
To use the charts here, [Helm](#) must be configured for your Kubernetes cluster. Setting up Kubernetes and Helm is outside the scope of this README. Please refer to the Kubernetes and Helm documentation.

The versions required are:

- **Helm 3.0+** - This is the earliest version of Helm tested. It is possible it works with earlier versions but this chart is untested for those versions.
- **Kubernetes 1.14+** - This is the earliest version of Kubernetes tested. It is possible that this chart works with earlier versions but it is untested.

Import Multi-doc YAML

Project: All Projects ▾

Import YAML

Drag and drop YAML or JSON files into the editor, or manually enter files and use `---` to separate each definition.

```
4   name: example-user-menu0
5 spec:
6   href: 'https://www.example.com'
7   location: UserMenu
8   text: User Menu Link0
9
10  ---
11
12  apiVersion: console.openshift.io/v1
13  kind: ConsoleLink
14  metadata:
15    name: example-user-menu1
16  spec:
17    href: 'https://www.example.com'
18    location: UserMenu
19    text: User Menu Link1
20
21  ---
22
23  apiVersion: console.openshift.io/v1
24  kind: ConsoleLink
25  metadata:
26    name: example-user-menu2
27  spec:
28    href: 'https://www.example.com'
29    location: UserMenu
30    text: User Menu Link2
31
```



Resources successfully created

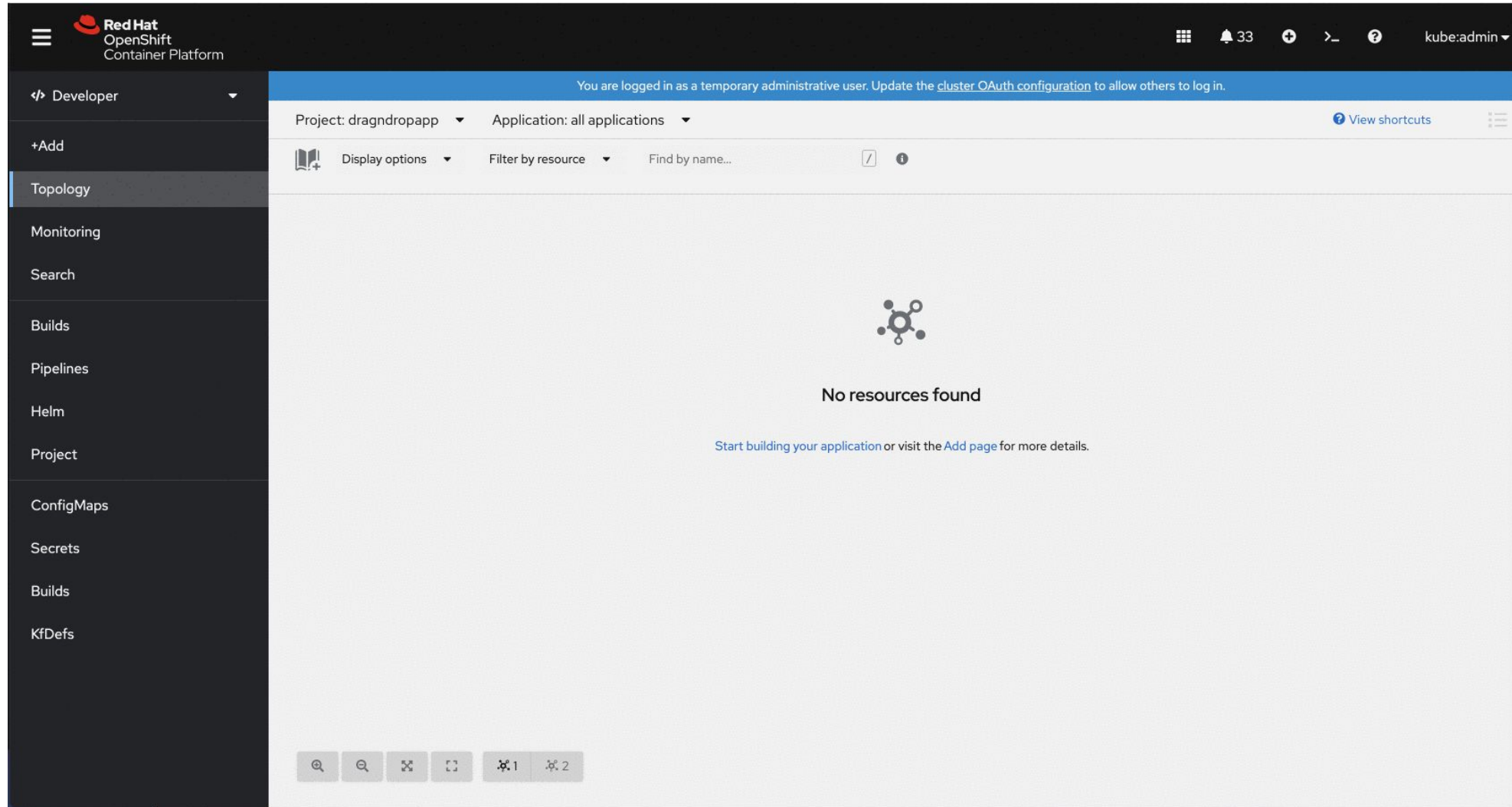
Name	Namespace	Creation status
 example-user-menu0	-	 Created
 example-user-menu1	-	 Created
 example-user-menu2	-	 Created

[Import more YAML](#)

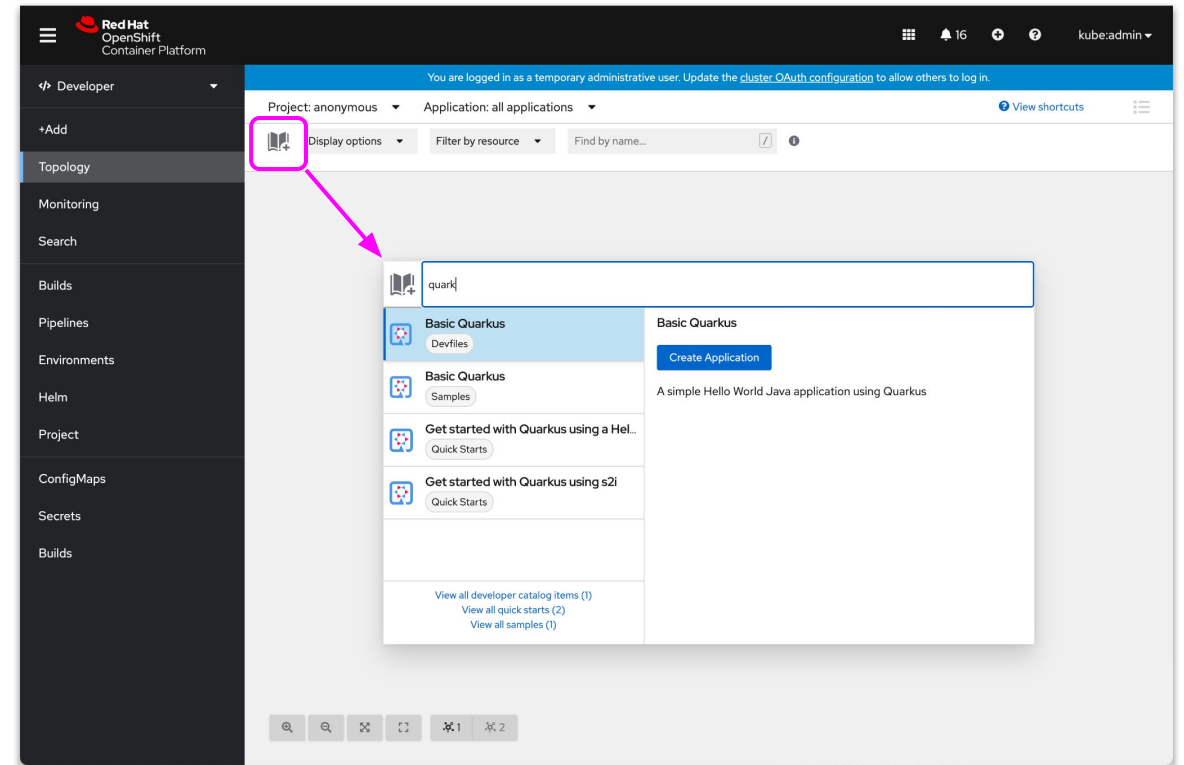
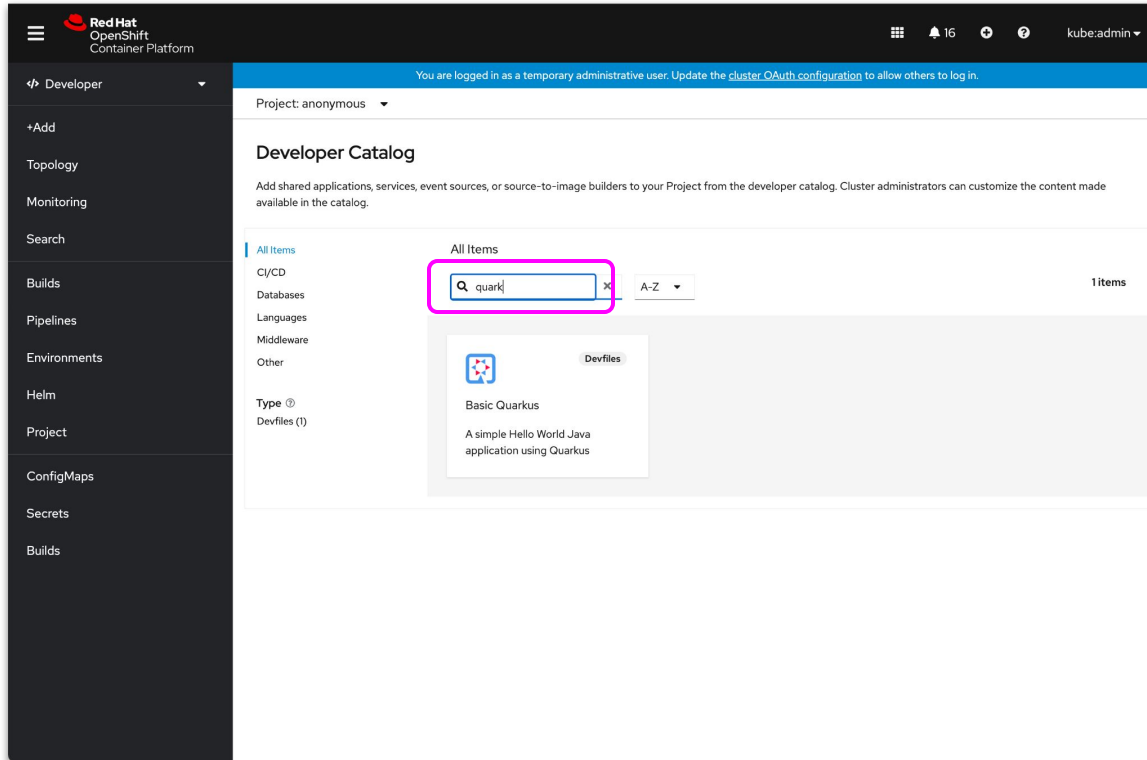
Create

Cancel

Easy import for App artifacts



Improved search in Catalog & Topology



Parity w/ 3.x – Form based edit for Deployment/DCs

The screenshot shows the OpenShift console interface for editing a DeploymentConfig. The left sidebar contains navigation links: Developer, +Add, Topology, Monitoring, Search, Builds, Pipelines, Environments, Helm, Project, ConfigMaps, Secrets, and Builds. The main content area is titled 'Edit DeploymentConfig' and includes a status bar at the top indicating the user is logged in as a temporary administrative user. Below the status bar, there are filters for 'Project: jeff' and 'Application: all applications'. The 'Configure via' section has two radio buttons: 'Form view' (selected) and 'YAML view'. A blue informational banner states: 'Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.' The 'Deployment strategy' section features a 'Strategy type' dropdown menu set to 'Recreate', with a description: 'The recreate strategy has basic rollout behavior and supports lifecycle hooks for injecting code into the deployment process.' Below this is a 'Timeout' input field set to '1200', with a description: 'The number of seconds to wait for a pod to scale up before giving up'. A link 'Show additional parameters and lifecycle hooks' is visible. The 'Images' section shows a 'Container' dropdown set to 'jupyterhub', a checkbox for 'Deploy image from an image stream tag' (unchecked), and an 'Image Name' input field containing 'jupyterhub-img:latest'. At the bottom, there are 'Save', 'Reload', and 'Cancel' buttons.

Red Hat OpenShift Container Platform

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

Project: jeff Application: all applications

Edit DeploymentConfig

Configure via: ☒ Form view ☐ YAML view

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

Deployment strategy

Strategy type

Recreate

The recreate strategy has basic rollout behavior and supports lifecycle hooks for injecting code into the deployment process.

Timeout

1200

The number of seconds to wait for a pod to scale up before giving up

[Show additional parameters and lifecycle hooks](#)

Images

Container: ☒ jupyterhub

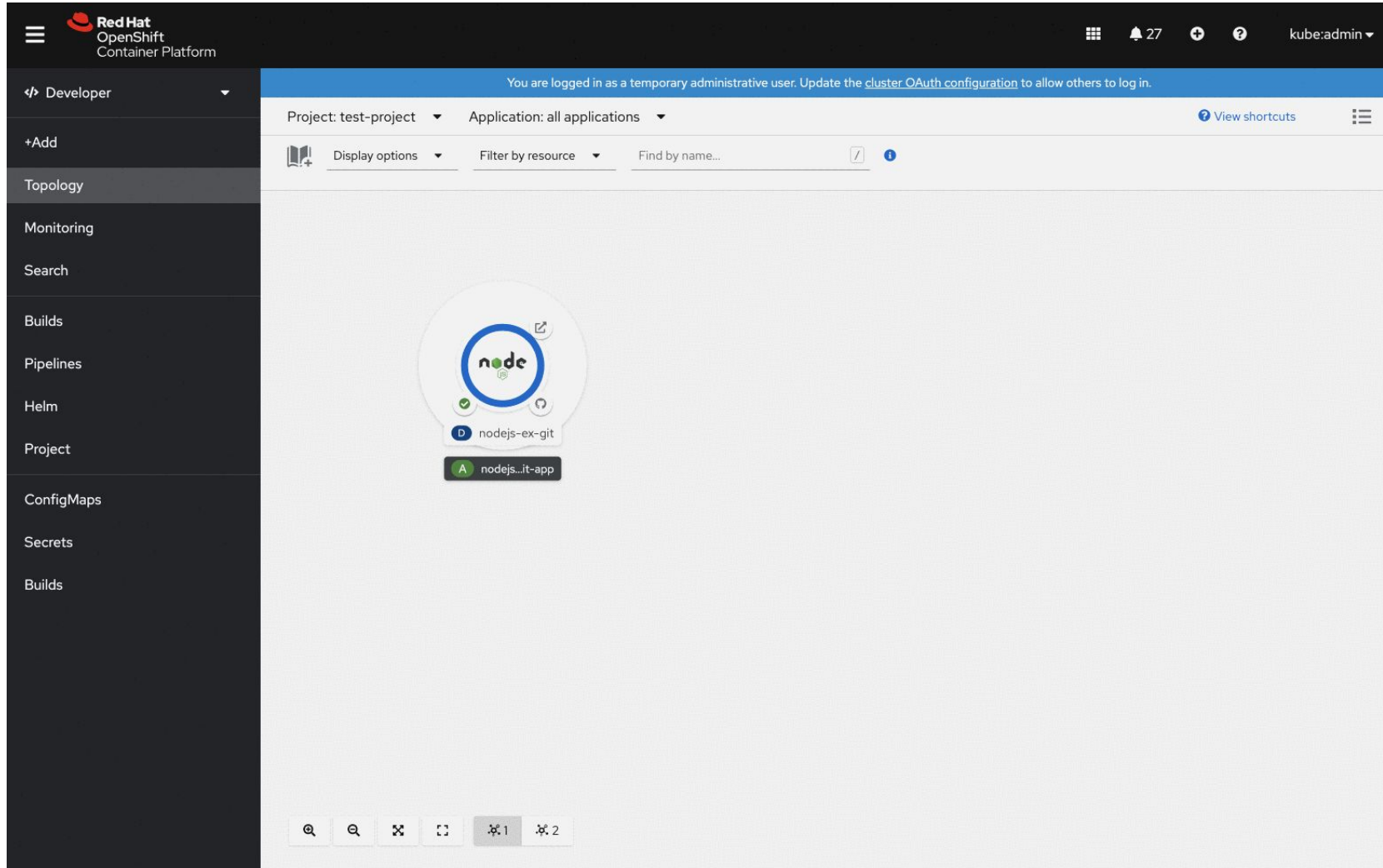
☐ Deploy image from an image stream tag

Image Name *

jupyterhub-img:latest

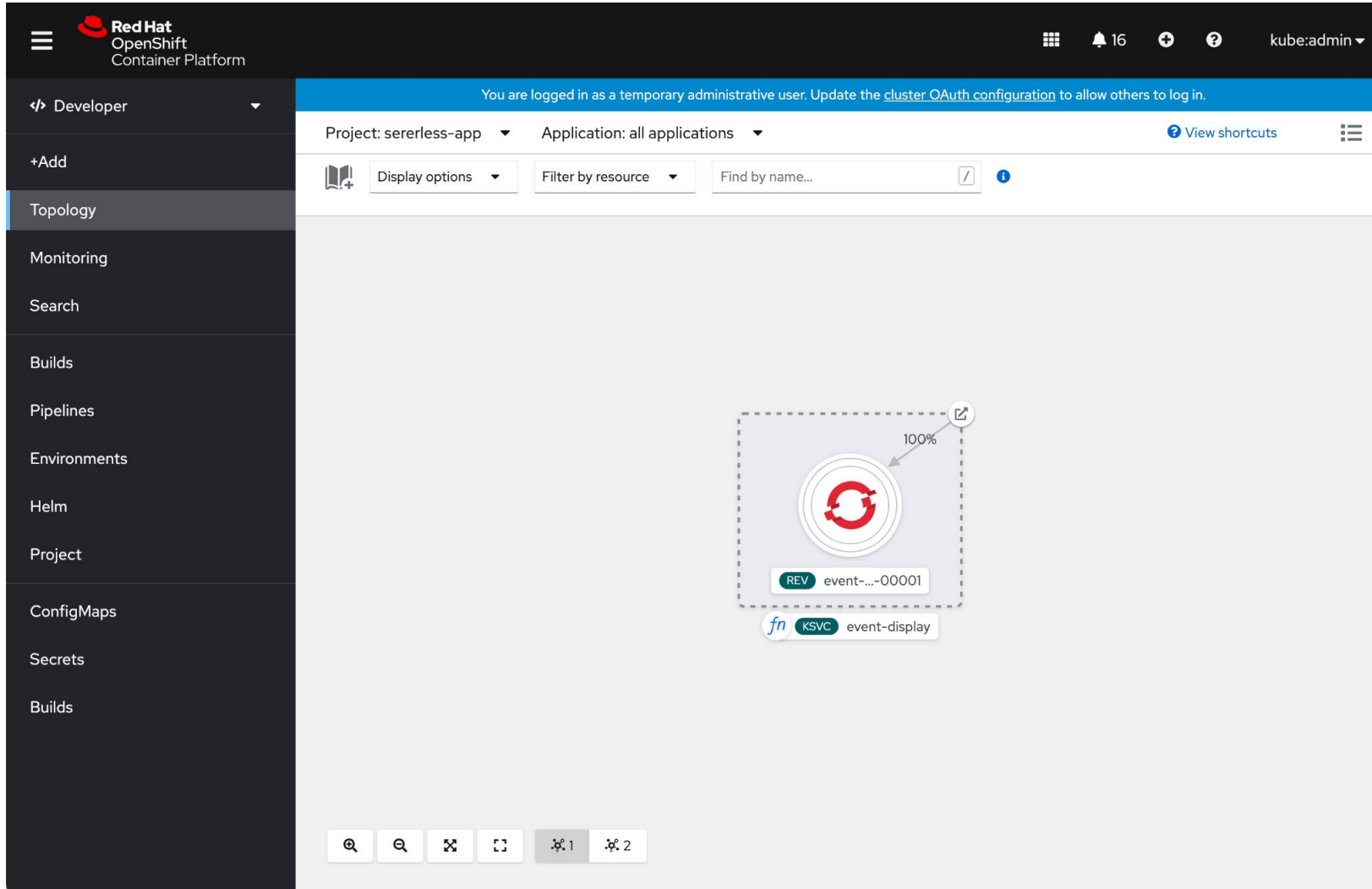
Save Reload Cancel

Expanded UI for Serverless



- **Make Serverless is Tech Preview**

Expanded UI for Serverless



- **Cloud Functions** in Topology
- Event Sources can specify cloud functions as their SINK

Expanded UI for Serverless

› [Show advanced Routing options](#)

Scaling

Set the autoscaler parameters around pods and concurrency limits in this section.

Min Pods

The lower limit for the number of Pods that can be set by autoscaler. If not specified defaults to 0.

Max Pods

The upper limit for the number of Pods that can be set by autoscaler.

Concurrency target

Defines how many concurrent requests are wanted per instance of the Application at a given time (soft limit) and is the recommended configuration for autoscaling. If not specified, will be defaulted to the value set in the Cluster config.

Concurrency limit

Limits the amount of concurrent requests allowed into one instance of the Application at a given time (hard limit), and is configured in the revision template. If not specified, will be defaulted to the value set in the Cluster config.

Concurrency utilization

Percentage of concurrent requests utilization before scaling up.

Autoscale window

Duration to look back for making auto-scaling decisions. The service is scaled to zero if no request was received in during that time.

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

Create

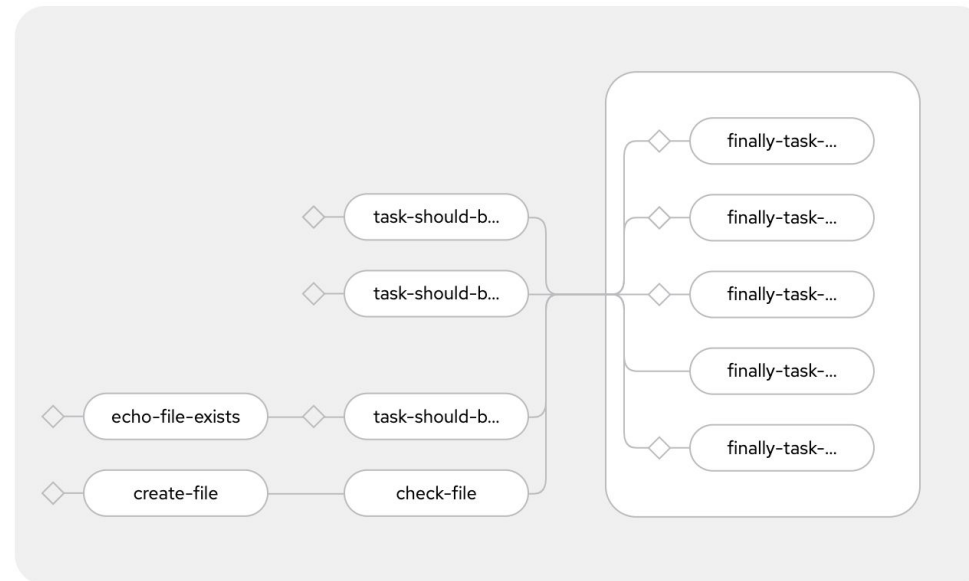
Cancel

Enhanced Scaling options for Knative Services

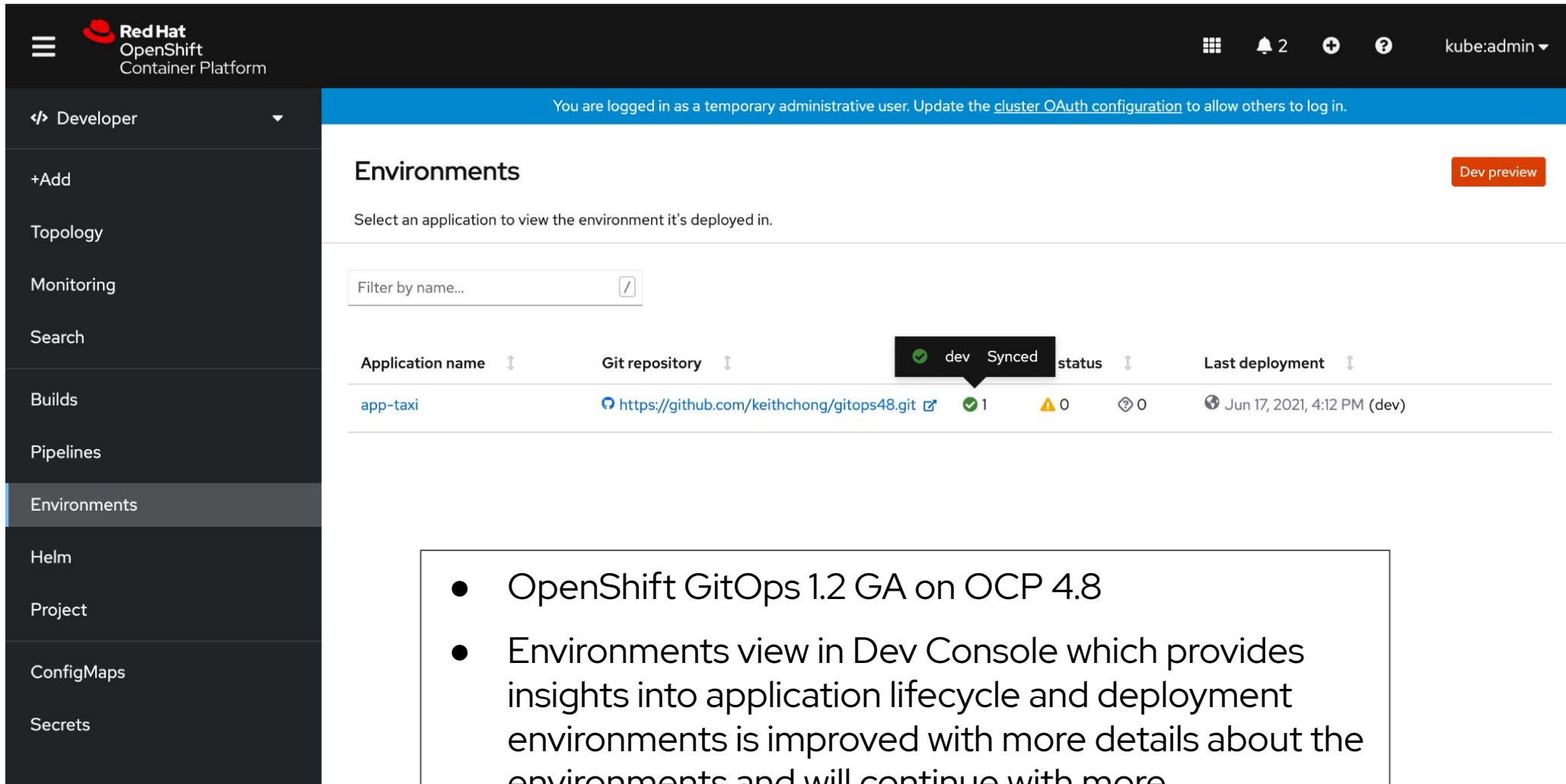
- Concurrency utilization
- Autoscale window

OpenShift Pipelines

- OpenShift Pipelines 1.5 GA on OCP 4.8
- Feature parity with Tekton within the Pipeline Builder & other pipeline related flows
 - When expressions
 - Finally tasks



OpenShift GitOps



The screenshot shows the OpenShift Dev Console interface. The top navigation bar includes the Red Hat OpenShift Container Platform logo, a hamburger menu, and user information (kube:admin). A blue banner indicates the user is logged in as a temporary administrative user and provides a link to update the cluster OAuth configuration. The left sidebar contains a list of navigation items: Developer, +Add, Topology, Monitoring, Search, Builds, Pipelines, Environments (selected), Helm, Project, ConfigMaps, and Secrets. The main content area is titled 'Environments' and includes a 'Dev preview' button. Below the title, there is a filter input field and a table of environments. The table has columns for Application name, Git repository, status, and Last deployment. A tooltip is visible over the 'dev' status, showing 'Synced'. The table lists one environment, 'app-taxi', with its Git repository URL, status (1 success, 0 warnings, 0 errors), and last deployment time.

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

Environments

Select an application to view the environment it's deployed in.

Filter by name...

Application name	Git repository	status	Last deployment
app-taxi	https://github.com/keithchong/gitops48.git	dev Synced 1 0 0	Jun 17, 2021, 4:12 PM (dev)

- OpenShift GitOps 1.2 GA on OCP 4.8
- Environments view in Dev Console which provides insights into application lifecycle and deployment environments is improved with more details about the environments and will continue with more improvements in the following OCP versions

Customizing the Developer Experience

- Add page
- Project Access
- Quick Starts

New Quick Start features

Quick Starts now support copy & execute

Red Hat OpenShift Container Platform

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

Project: All Projects

Add

Select a Project to start adding to it or [create a Project](#).

Name Search by name...

Name	Display name	Status	Requester	Created
PR anonymous	No display name	Active	No requester	Jun 18, 2021, 7:23 AM
PR daves-proj	No display name	Active	No requester	Jun 18, 2021, 10:04 AM
PR deb	No display name	Active	kube:admin	Jun 18, 2021, 10:22 AM
PR default	No display name	Active	No requester	Jun 18, 2021, 6:08 AM
PR jai-test	No display name	Active	kube:admin	Jun 18, 2021, 8:15 AM

Command line terminal Tech preview

```
sh-4.4$
```

OpenShift Quick Starts now provide support for both a copy and execute feature.

The **execute** feature provides an ability for the user to execute the CLI in our Command line terminal. The Web Terminal Operator is required for execute.

Format to enable copy
``https://github.com/sclorg/ruby-ex.git`{{copy}}`

Format to enable copy & execute
``oc new-app ruby~https://github.com/sclorg/ruby-ex.git`{{execute}}`

More ways to customize the Developer Experience

Hide individual features from the Add page

Add

Select a way to create an Application, component or service from one of the options.

Show getting started resources ✕ ☒ Details on

Developer Catalog

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

Database
Browse the catalog to discover database services to add to your Application

Operator Backed
Browse the catalog to discover and deploy operator managed services

Helm Chart
Browse the catalog to discover and install Helm Charts

Git Repository

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

From Devfile
Import your Devfile from your Git repository to be built and deployed

From Dockerfile
Import your Dockerfile from your Git repository to be built and deployed

Pipelines
Create a Tekton Pipeline to automate delivery of your Application

Container images

Deploy an existing Image from an Image registry or Image stream tag

From Local Machine

Import YAML
Create resources from their YAML or JSON definitions

Upload JAR file
Upload a JAR file from your local desktop to OpenShift

Samples
Create an Application from a code sample

```
spec:
  customization:
    addPage:
      disabledActions:
        - import-from-dockerfile
```

More ways to customize the Developer Experience

Modify the available roles in the Project Access flow

The screenshot shows the OpenShift web console interface. The left sidebar contains navigation links: Developer, +Add, Topology, Monitoring, Search, Builds, Pipelines, Helm, Project (selected), ConfigMaps, Secrets, and Builds. The main content area is titled 'Project: test-project' and shows the 'Project access' tab. It displays a table of roles assigned to the project:

Name	Role
kube:admin	admin
pipeline	edit

Below the table, there is a '+ Add access' button and a dropdown menu showing available roles: admin, edit, view, and registry-admin. At the bottom, there are 'Save' and 'Reload' buttons.

```
spec:
  customization:
    projectAccess:
      availableClusterRoles:
        - admin
        - edit
        - view
        - registry-admin
```

Questions / Comments ?

<https://developers.redhat.com/products/openshift/whats-new>

@serenamarie125

twitter

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat