Red Hat Developer Hub 1.8

Using dynamic plugins in Red Hat Developer Hub

Using Red Hat Developer Hub plugins to access your development infrastructure and software development tools

Last Updated: 2026-01-22

Red Hat Developer Hub 1.8 Using dynamic plugins in Red Hat Developer Hub

Using Red Hat Developer Hub plugins to access your development infrastructure and software development tools

Legal Notice

Abstract

The following sections provide information about how you can use Red Hat Developer Hub (RHDH) dynamic plugins.

Table of Contents

CHAPTER 1. USING ANSIBLE PLUG-INS FOR RED HAT DEVELOPER HUB

Ansible plug-ins for Red Hat Developer Hub deliver an Ansible-specific portal experience with curated learning paths, push-button content creation, integrated development tools, and other opinionated resources.

IMPORTANT

The Ansible plug-ins are a Technology Preview feature only.

Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

Additonal resources

- Technology Preview Features Scope
- Red Hat Developer Support Policy
- *Using Ansible plug-ins for Red Hat Developer Hub*

CHAPTER 2. USING THE ARGO CD PLUGIN

You can use the Argo CD plugin to visualize the Continuous Delivery (CD) workflows in OpenShift GitOps. This plugin provides a visual overview of the application's status, deployment details, commit message, author of the commit, container image promoted to environment and deployment history.

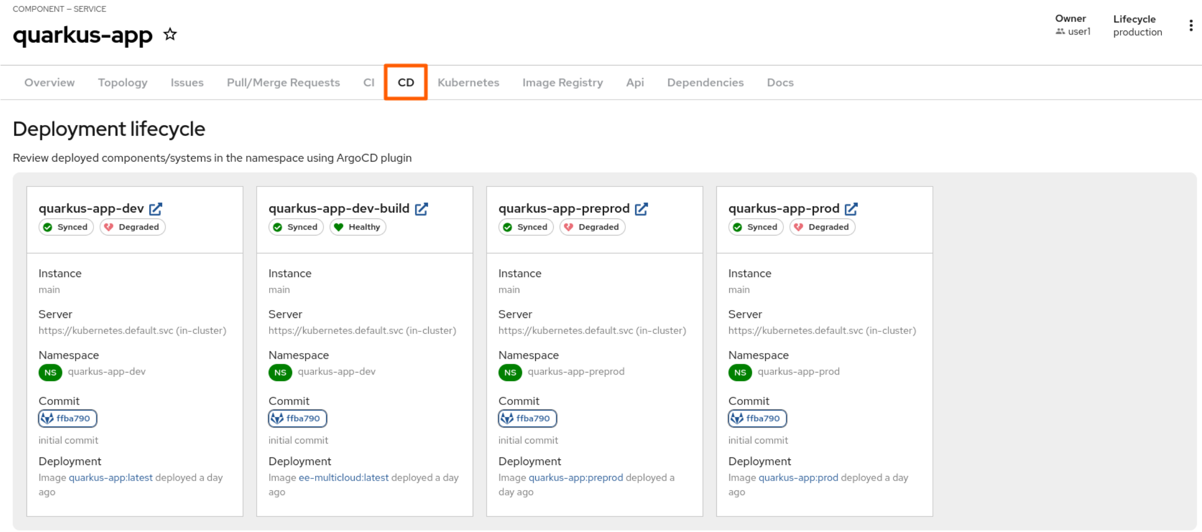
Prerequisites

- You have enabled the Argo CD plugin in Red Hat Developer Hub RHDH.

Procedure

1. Select the **Catalog** tab and choose the component that you want to use.

2. Select the **CD** tab to view insights into deployments managed by Argo CD.



Deployment lifecycle

Review deployed components/systems in the namespace using ArgoCD plugin

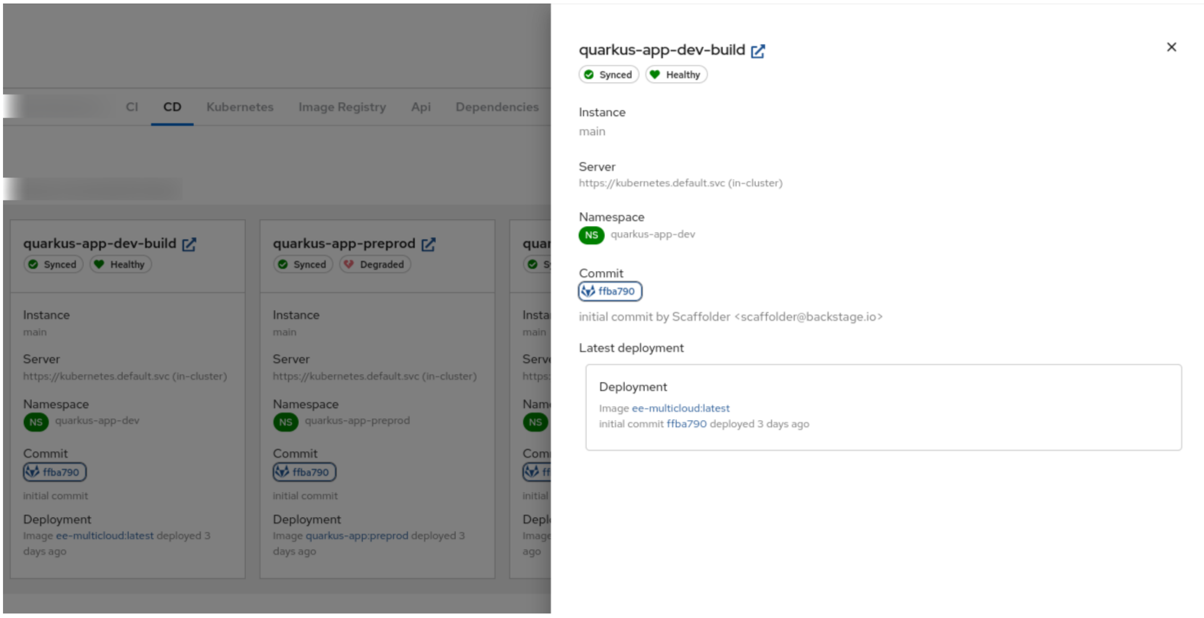
quarkus-app-dev ✓ Synced ♥ Degraded
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-dev
Commit: ffba790 initial commit
Deployment: Image quarkus-app:latest deployed a day ago

quarkus-app-dev-build ✓ Synced ♥ Healthy
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-dev
Commit: ffba790 initial commit
Deployment: Image ee-multicloud:latest deployed a day ago

quarkus-app-preprod ✓ Synced ♥ Degraded
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-preprod
Commit: ffba790 initial commit
Deployment: Image quarkus-app:preprod deployed a day ago

quarkus-app-prod ✓ Synced ♥ Degraded
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-prod
Commit: ffba790 initial commit
Deployment: Image quarkus-app:prod deployed a day ago

3. Select an appropriate card to view the deployment details (for example, commit message, author name, and deployment history).



quarkus-app-dev-build ✓ Synced ♥ Healthy
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-dev
Commit: ffba790 initial commit
Deployment: Image ee-multicloud:latest deployed 3 days ago

quarkus-app-preprod ✓ Synced ♥ Degraded
Instance: main
Server: https://kubernetes.default.svc (in-cluster)
Namespace: NS quarkus-app-preprod
Commit: ffba790 initial commit
Deployment: Image quarkus-app:preprod deployed 3 days ago

quarkus-app-dev-build ✓ Synced ♥ Healthy

Instance: main

Server: https://kubernetes.default.svc (in-cluster)

Namespace: NS quarkus-app-dev

Commit: ffba790 initial commit by Scaffolder <scaffolder@backstage.io>

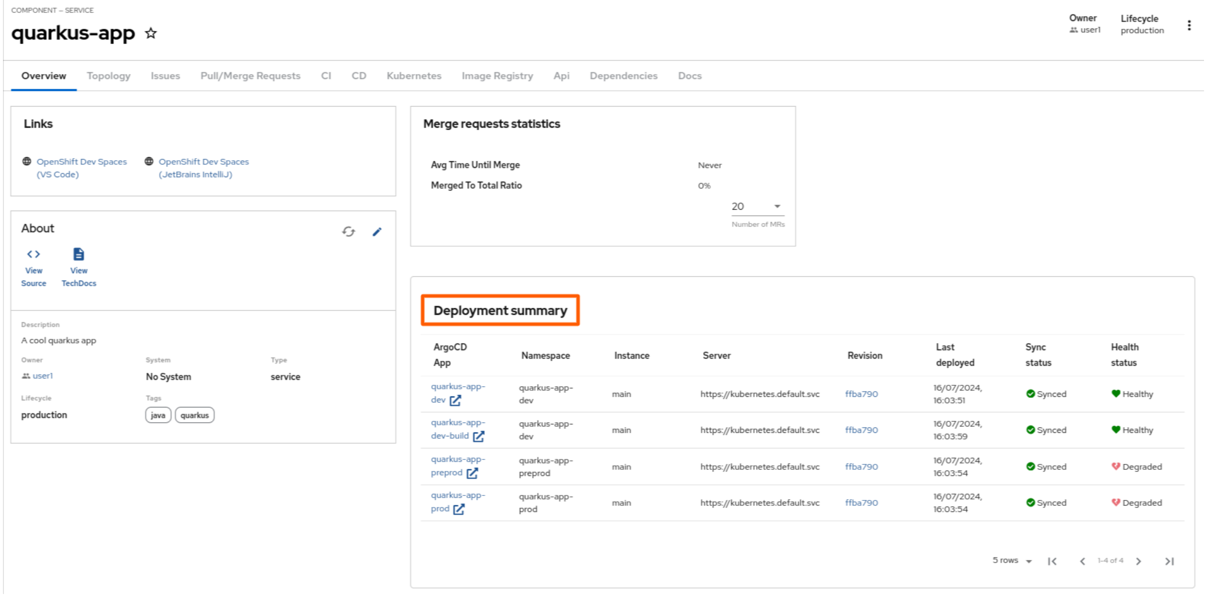
Latest deployment

Deployment
Image ee-multicloud:latest
initial commit ffba790 deployed 3 days ago

a. Click the link icon (↗) to open the deployment details in Argo CD.

4. Select the **Overview** tab and navigate to the Deployment summary section to review the summary of your application's deployment across namespaces. Additionally, select an

appropriate Argo CD app to open the deployment details in Argo CD, or select a commit ID from the Revision column to review the changes in GitLab or GitHub.



Description
A cool quarkus app

Owner: user1
System: No System
Type: service
Lifecycle: production
Tags: java, quarkus

Merge requests statistics

Avg Time Until Merge: Never
Merged To Total Ratio: 0%
Number of MRs: 20

Deployment summary

ArgoCD App	Namespace	Instance	Server	Revision	Last deployed	Sync status	Health status
quarkus-app-dev	quarkus-app-dev	main	https://kubernetes.default.svc	ffba790	16/07/2024, 16:03:51	Synced	Healthy
quarkus-app-dev-build	quarkus-app-dev	main	https://kubernetes.default.svc	ffba790	16/07/2024, 16:03:59	Synced	Healthy
quarkus-app-preprod	quarkus-app-preprod	main	https://kubernetes.default.svc	ffba790	16/07/2024, 16:03:54	Synced	Degraded
quarkus-app-prod	quarkus-app-prod	main	https://kubernetes.default.svc	ffba790	16/07/2024, 16:03:54	Synced	Degraded

Additional resources

- Installing and viewing plugins in Red Hat Developer Hub

CHAPTER 3. USING THE JFROG ARTIFACTORY PLUGIN

The JFrog Artifactory plugin displays information about your container images within the Jfrog Artifactory registry.

IMPORTANT

The JFrog Artifactory plugin is a Technology Preview feature only.

Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information on Red Hat Technology Preview features, see Technology Preview Features Scope.

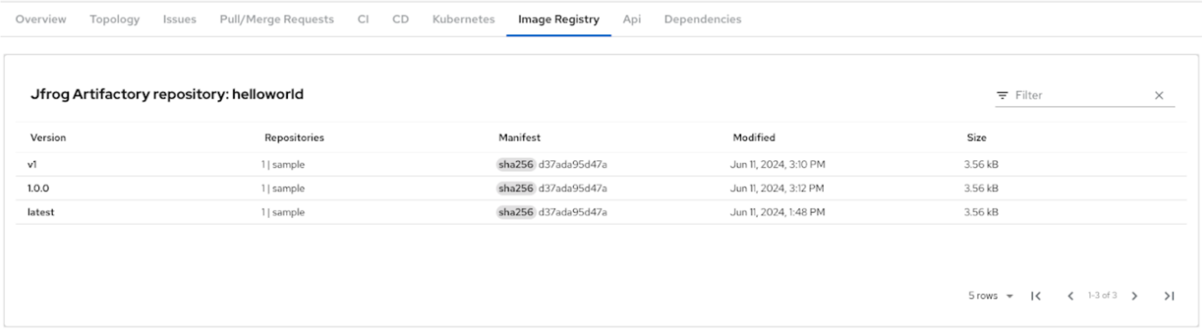
Additional detail on how Red Hat provides support for bundled community dynamic plugins is available on the Red Hat Developer Support Policy page.

Prerequisites

- Your Developer Hub application is installed and running.
- You have enabled the JFrog Artifactory plugin.

Procedure

1. Open your Developer Hub application and select a component from the **Catalog** page.
2. Go to the **Image Registry** tab.
   The **Image Registry** tab contains a list of container images within your Jfrog Artifactory repository and related information, such as **Version**, **Repositories**, **Manifest**, **Modified**, and **Size**.



Jfrog Artifactory repository: helloworld

Version	Repositories	Manifest	Modified	Size
v1	1 | sample	sha256 d37ada95d47a	Jun 11, 2024, 3:10 PM	3.56 kB
1.0.0	1 | sample	sha256 d37ada95d47a	Jun 11, 2024, 3:12 PM	3.56 kB
latest	1 | sample	sha256 d37ada95d47a	Jun 11, 2024, 1:48 PM	3.56 kB

CHAPTER 4. USING KEYCLOAK

The Keycloak backend plugin, which integrates Keycloak into Developer Hub, has the following capabilities:

- Synchronization of Keycloak users in a realm.
- Synchronization of Keycloak groups and their users in a realm.

4.1. IMPORTING USERS AND GROUPS IN DEVELOPER HUB USING THE KEYCLOAK PLUGIN

After configuring the plugin successfully, the plugin imports the users and groups each time when started.

NOTE

If you set up a schedule, users and groups will also be imported.

Procedure

1. in Red Hat Developer Hub, go to the **Catalog** page.
2. Select **User** from the entity type filter to display the list of imported users.
3. Browse the list of users displayed on the page.
4. Select a user to view detailed information imported from Keycloak.
5. To view groups, select **Group** from the entity type filter.
6. Browse the list of groups shown on the page.
7. From the list of groups, select a group to view the information imported from Keycloak.

CHAPTER 5. USING THE NEXUS REPOSITORY MANAGER PLUGIN

The Nexus Repository Manager plugin displays the information about your build artifacts in your Developer Hub application. The build artifacts are available in the Nexus Repository Manager.

IMPORTANT

The Nexus Repository Manager plugin is a Technology Preview feature only.

Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend using them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information on Red Hat Technology Preview features, see Technology Preview Features Scope.

Additional detail on how Red Hat provides support for bundled community dynamic plugins is available on the Red Hat Developer Support Policy page.

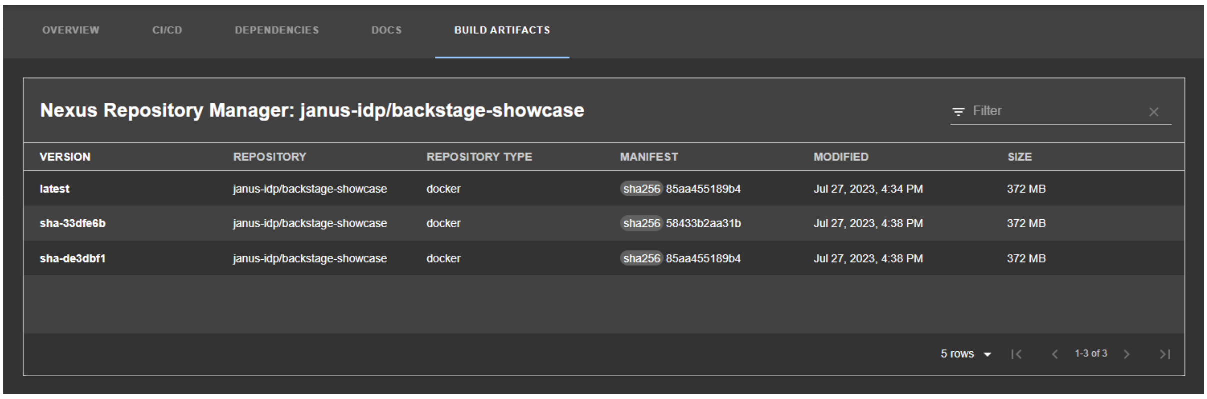
The Nexus Repository Manager is a front-end plugin that enables you to view the information about build artifacts.

Prerequisites

- Your Developer Hub application is installed and running.
- You have installed the Nexus Repository Manager plugin.

Procedure

1. Open your Developer Hub application and select a component from the **Catalog** page.
2. Go to the **BUILD ARTIFACTS** tab.
   The **BUILD ARTIFACTS** tab contains a list of build artifacts and related information, such as **VERSION**, **REPOSITORY**, **REPOSITORY TYPE**, **MANIFEST**, **MODIFIED**, and **SIZE**.



Nexus Repository Manager: janus-idp/backstage-showcase

VERSION	REPOSITORY	REPOSITORY TYPE	MANIFEST	MODIFIED	SIZE
latest	janus-idp/backstage-showcase	docker	sha256 85aa455189b4	Jul 27, 2023, 4:34 PM	372 MB
sha-33dfe6b	janus-idp/backstage-showcase	docker	sha256 58433b2aa31b	Jul 27, 2023, 4:38 PM	372 MB
sha-de3dbf1	janus-idp/backstage-showcase	docker	sha256 85aa455189b4	Jul 27, 2023, 4:38 PM	372 MB

CHAPTER 6. USING THE TEKTON PLUGIN

You can use the Tekton plugin to visualize the results of CI/CD pipeline runs on your Kubernetes or OpenShift clusters. The plugin allows users to visually see high level status of all associated tasks in the pipeline for their applications.

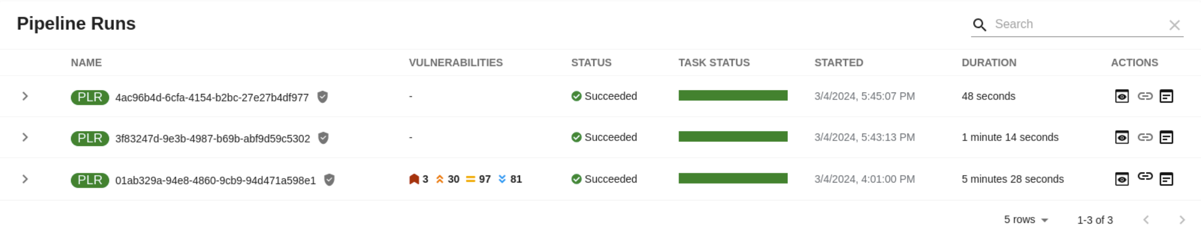
You can use the Tekton front-end plugin to view **PipelineRun** resources.

Prerequisites

- You have installed the Red Hat Developer Hub (RHDH).
- You have installed the Tekton plugin. For the installation process, see Installing and configuring the Tekton plugin.

Procedure

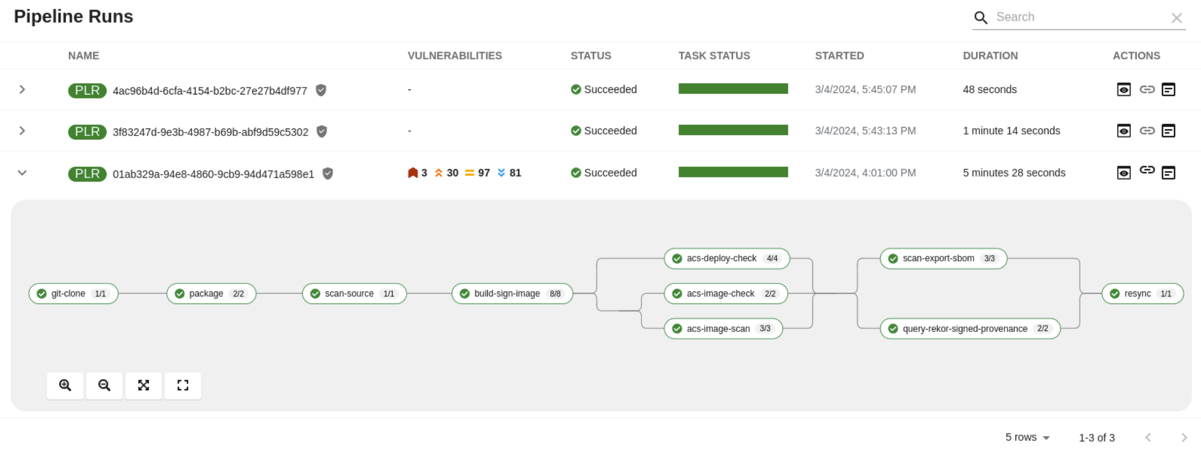
1. Open your RHDH application and select a component from the **Catalog** page.
2. Go to the **CI** tab.
   The **CI** tab displays the list of PipelineRun resources associated with a Kubernetes cluster. The list contains pipeline run details, such as **NAME**, **VULNERABILITIES**, **STATUS**, **TASK STATUS**, **STARTED**, and **DURATION**.



Pipeline Runs

	NAME	VULNERABILITIES	STATUS	TASK STATUS	STARTED	DURATION	ACTIONS
›	PLR 4ac96b4d-6cfa-4154-b2bc-27e27b4df977	-	✓ Succeeded	🟩	3/4/2024, 5:45:07 PM	48 seconds	
›	PLR 3f83247d-9e3b-4987-b69b-abf9d59c5302	-	✓ Succeeded	🟩	3/4/2024, 5:43:13 PM	1 minute 14 seconds	
›	PLR 01ab329a-94e8-4860-9cb9-94d471a598e1	3 30 97 81	✓ Succeeded	🟩	3/4/2024, 4:01:00 PM	5 minutes 28 seconds	

3. Click the expand row button besides PipelineRun name in the list to view the PipelineRun visualization. The pipeline run resource includes tasks to complete. When you hover the mouse pointer on a task card, you can view the steps to complete that particular task.



Pipeline Runs

	NAME	VULNERABILITIES	STATUS	TASK STATUS	STARTED	DURATION	ACTIONS
›	PLR 4ac96b4d-6cfa-4154-b2bc-27e27b4df977	-	✓ Succeeded	🟩	3/4/2024, 5:45:07 PM	48 seconds	
›	PLR 3f83247d-9e3b-4987-b69b-abf9d59c5302	-	✓ Succeeded	🟩	3/4/2024, 5:43:13 PM	1 minute 14 seconds	
⌄	PLR 01ab329a-94e8-4860-9cb9-94d471a598e1	3 30 97 81	✓ Succeeded	🟩	3/4/2024, 4:01:00 PM	5 minutes 28 seconds	

CHAPTER 7. USING THE TOPOLOGY PLUGIN

Topology is a front-end plugin that enables you to view the workloads as nodes that power any service on the Kubernetes cluster.

7.1. ENABLING USERS TO USE THE TOPOLOGY PLUGIN

The Topology plugin is defining additional permissions. When Authorization in Red Hat Developer Hub is enabled, to enable users to use the Topology plugin, grant them:

- The **kubernetes.clusters.read** and **kubernetes.resources.read**, **read** permissions to view the Topology panel.
- The **kubernetes.proxy use** permission to view the pod logs.
- The **catalog-entity read** permission to view the Red Hat Developer Hub software catalog items.

Prerequisites

- You are managing Authorization in Red Hat Developer Hub by using external files .

Procedure

- Add the following permission policies to your **rbac-policy.csv** file to create a **topology-viewer** role that has access to the Topology plugin features, and add the role to the users requiring this authorization:

```
g, user:default/<YOUR_USERNAME>, role:default/topology-viewer
p, role:default/topology-viewer, kubernetes.clusters.read, read, allow ①
p, role:default/topology-viewer, kubernetes.resources.read, read, allow ②
p, role:default/topology-viewer, kubernetes.proxy, use, allow ③
p, role:default/topology-viewer, catalog-entity, read, allow ④
```

① ② Grants the user the ability to see the Topology panel.

③ Grants the user the ability to view the pod logs.

④ Grants the user the ability to see the catalog item.

7.2. USING THE TOPOLOGY PLUGIN

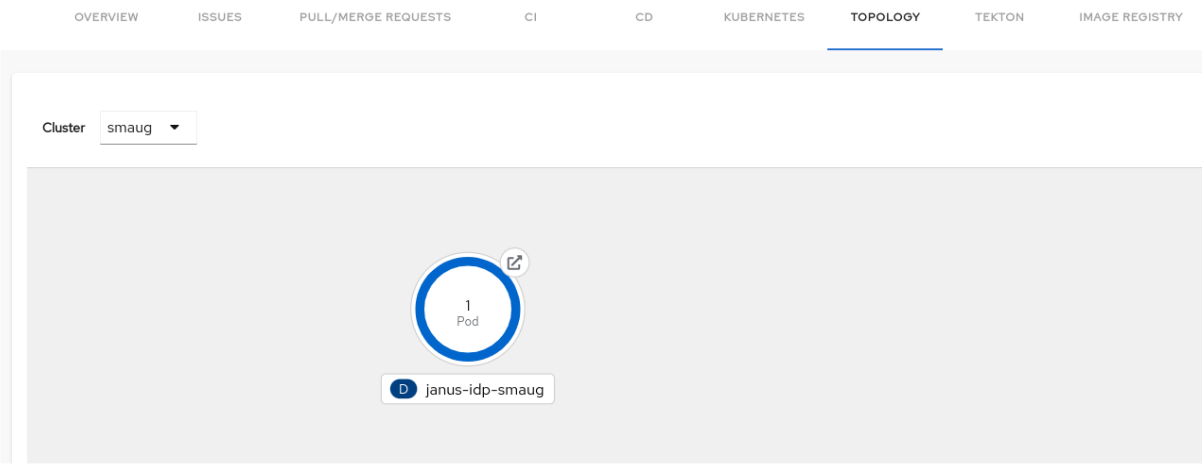
Prerequisites

- Your Red Hat Developer Hub instance is installed and running.
- You have installed the Topology plugin.
- You have enabled the users to use the Topology plugin .

Procedure

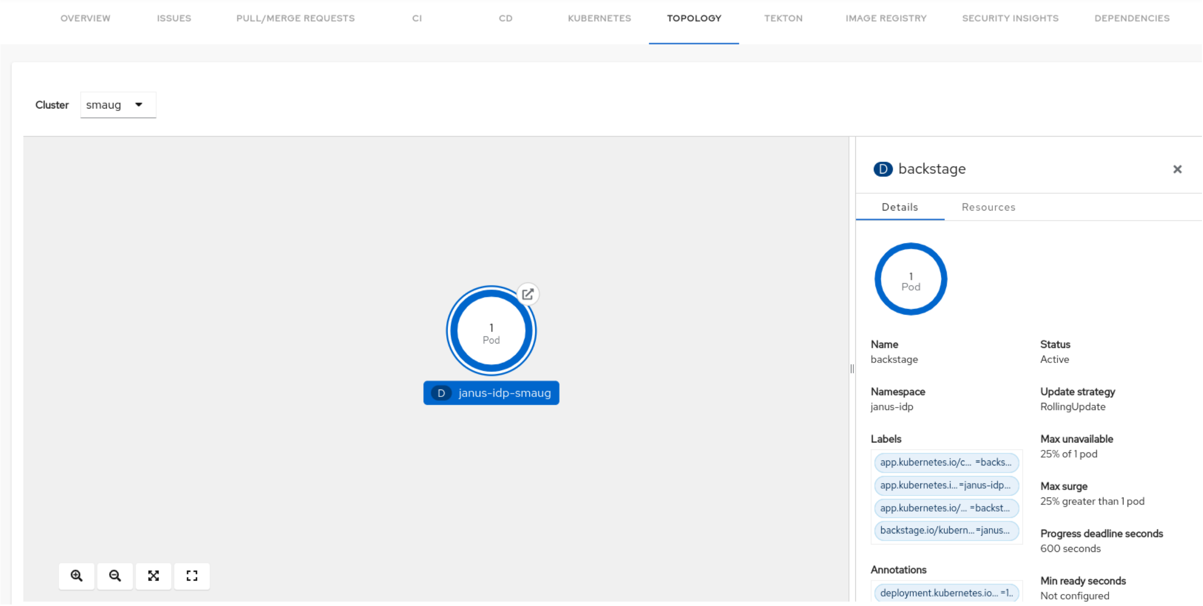
1. Open your RHDH application and select a component from the **Catalog** page.

2. Go to the **TOPOLOGY** tab and you can view the workloads such as deployments or pods as nodes.



Cluster smaug

3. Select a node and a pop-up appears on the right side that contains two tabs: **Details** and **Resources**.
   The **Details** and **Resources** tabs contain the associated information and resources for the node.



Cluster smaug

Name
backstage

Status
Active

Namespace
janus-idp

Update strategy
RollingUpdate

Labels
app.kubernetes.io/c... =backs...
app.kubernetes.i... =janus-idp...
app.kubernetes.io/... =backst...
backstage.io/kubern... =janus...

Max unavailable
25% of 1 pod

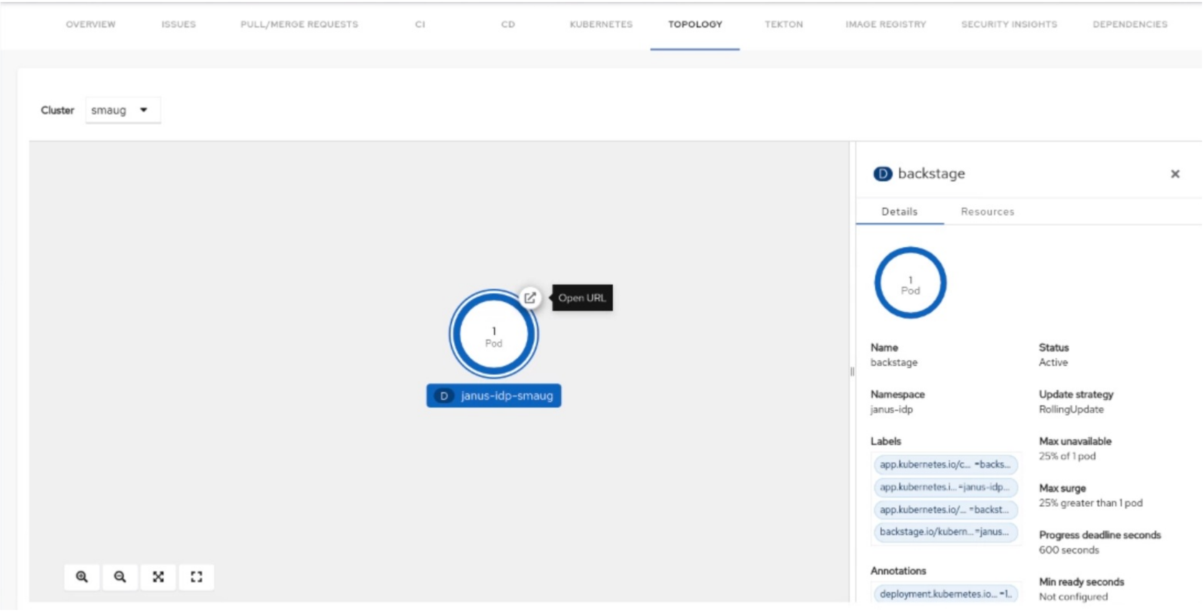
Max surge
25% greater than 1 pod

Progress deadline seconds
600 seconds

Annotations
deployment.kubernetes.io... =1..

Min ready seconds
Not configured

4. Click the **Open URL** button on the top of a node.



Cluster smaug

Name
backstage

Status
Active

Namespace
janus-idp

Update strategy
RollingUpdate

Labels
app.kubernetes.io/c... =backs...
app.kubernetes.i... =janus-idp...
app.kubernetes.io/... =backst...
backstage.io/kubern... =janus...

Max unavailable
25% of 1 pod

Max surge
25% greater than 1 pod

Progress deadline seconds
600 seconds

Annotations
deployment.kubernetes.io... =1...

Min ready seconds
Not configured

Click the **Open URL** button to access the associated **Ingresses** and run your application in a new tab.