



# OpenShift Dedicated 4

## Release notes

Release notes for OpenShift Dedicated 4



# OpenShift Dedicated 4 Release notes

---

Release notes for OpenShift Dedicated 4

## Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Release notes for OpenShift Dedicated 4

---

## Table of Contents

<b>CHAPTER 1. OPENSIFT DEDICATED 4 RELEASE NOTES .....</b>	<b>3</b>
1.1. ABOUT THIS RELEASE	3
1.2. NEW FEATURES AND ENHANCEMENTS	3
1.2.1. OpenShift Do	3
1.2.2. CodeReady Containers	3
1.2.3. Web console	3
1.2.3.1. Console customization options	3
1.2.3.2. New API Explorer	3
1.2.3.3. Developer Perspective	4
1.2.3.4. Prometheus queries	4
1.2.3.5. General web console updates	4
1.3. NOTABLE TECHNICAL CHANGES	4
corsAllowedOrigins	4
Builds maintain their layers	4
Ingress controller support disabled	4
Reduce OperatorHub complexity by removing CatalogSourceConfig usage	4
Global catalog namespace change	5
Workaround for existing Subscriptions in the previous global catalog namespace	5
1.3.1. Deprecated features	5
Deprecation of the Service Catalog, the Template Service Broker, the Ansible Service Broker, and their Operators	5
Deprecation of cluster role APIs	5
Deprecation of OperatorSources	6
Deprecation of /oapi endpoint from oc	6
Deprecation of the -short flag of oc version	6
Recycle reclaim policy	6
1.4. BUG FIXES	6



# CHAPTER 1. OPENSIFT DEDICATED 4 RELEASE NOTES

Red Hat OpenShift Dedicated provides developers and IT organizations with a hybrid cloud application platform for deploying both new and existing applications on secure, scalable resources with minimal configuration and management overhead. OpenShift Dedicated supports a wide selection of programming languages and frameworks, such as Java, JavaScript, Python, Ruby, and PHP.

Built on Red Hat Enterprise Linux and Kubernetes, OpenShift Dedicated provides a more secure and scalable multi-tenant operating system for today's enterprise-class applications, while delivering integrated application runtimes and libraries. OpenShift Dedicated enables organizations to meet security, privacy, compliance, and governance requirements.

## 1.1. ABOUT THIS RELEASE

Red Hat OpenShift Dedicated ([RHBA-2019:2921](#)) is now available. This release uses Kubernetes 1.14 with CRI-O runtime. New features, changes, and known issues that pertain to OpenShift Dedicated 4 are included in this topic.

## 1.2. NEW FEATURES AND ENHANCEMENTS

This release adds improvements related to the following components and concepts.

### 1.2.1. OpenShift Do

OpenShift Do (odo) is a CLI tool for developers to create, build, and deploy applications on OpenShift. The odo tool is completely client-based and requires no server within the OpenShift Dedicated cluster for deployment. It detects changes to local code and deploys it to the cluster automatically, giving instant feedback to validate changes in real time. It supports multiple programming languages and frameworks.

### 1.2.2. CodeReady Containers

CodeReady Containers provides a local desktop instance of a minimal OpenShift Dedicated 4 or newer cluster. This cluster provides developers with a minimal environment for development and testing purposes. It includes the **crc** CLI to interact with the CodeReady Containers virtual machine running the OpenShift cluster.

### 1.2.3. Web console

#### 1.2.3.1. Console customization options

You can customize the OpenShift Dedicated web console to set a custom logo, links, notification banners, and command line downloads. This is especially helpful if you need to tailor the web console to meet specific corporate or government requirements.

#### 1.2.3.2. New API Explorer

You can now easily search and manage API resources in the **Explore API Resources** dashboard located at **Home → Explore**.

View the schema for each API and what parameters being supported, manage the instances of the API, and review the access of each API.

### 1.2.3.3. Developer Perspective

The Developer perspective adds a developer-focused perspective to the web console. It provides workflows specific to developer use cases, such as creation and deployment of applications to OpenShift Dedicated using multiple options. It provides a visual representation of the applications within a project, their build status, and the components and services associated with them, enabling easy interaction and monitoring. It incorporates Serverless capabilities (Technology Preview) and the ability to create workspaces to edit your application code using Eclipse Che.

### 1.2.3.4. Prometheus queries

You can now run Prometheus queries directly in the web console. Navigate to **Monitoring** → **Metrics**.

### 1.2.3.5. General web console updates

- The dashboard is redesigned with more metrics.
- **Catalog** is moved to the **Developer** perspective: **Developer** → **Add+** → **From Catalog**.
- Status of projects is now moved to the **Workloads** tab on the project details page.
- OperatorHub is now located under the **Operators** menu.
- There is now support for chargeback. You can break down the reserved and used resources requested by applications.
- There is now support for native templates without needing to enable the Service Catalog, which is now deprecated.

## 1.3. NOTABLE TECHNICAL CHANGES

OpenShift Dedicated 4 introduces the following notable technical changes.

### **corsAllowedOrigins**

#### **Builds maintain their layers**

In OpenShift Dedicated 4, builds keep their layers by default.

### **Ingress controller support disabled**

Ingress controller TLS 1.0 and 1.1 support is now disabled to match the Mozilla intermediate security profile.

New and upgraded ingress controllers will no longer support these TLS versions.

### **Reduce OperatorHub complexity by removing CatalogSourceConfig usage**

OperatorHub has been updated to reduce the number of API resources a cluster administrator must interact with and streamline the installation of new Operators on OpenShift Dedicated 4.

To work with OperatorHub in OpenShift Dedicated 4.1, cluster administrators primarily interacted with OperatorSource and CatalogSourceConfig API resources. OperatorSources are used to add external datastores where Operator bundles are stored.

CatalogSourceConfigs were used to enable an Operator present in the OperatorSource of your cluster. Behind the scenes, it configured an Operator Lifecycle Manager (OLM) CatalogSource so that the Operator could then be managed by OLM.

To reduce complexity, OperatorHub in OpenShift Dedicated 4 no longer uses CatalogSourceConfigs in

the workflow of installing Operators. Instead, CatalogSources are still created as a result of adding OperatorSources to the cluster, however Subscription resources are now created directly using the CatalogSource.



## NOTE

While OperatorHub no longer uses CatalogSourceConfig resources, they are still supported in OpenShift Dedicated.

### Global catalog namespace change

In OpenShift Dedicated 4.1, the default global catalog namespace, where CatalogSources are installed by default, is **openshift-operator-lifecycle-manager**. Starting with OpenShift Dedicated 4, this has changed to the **openshift-marketplace** namespace.

If you have installed an Operator from OperatorHub on an OpenShift Dedicated 4.1 cluster, the CatalogSource is in the same namespace as the Subscription. These Subscriptions are not affected by this change and should continue to behave normally after a cluster upgrade.

In OpenShift Dedicated 4, if you install an Operator from OperatorHub, the Subscription created refers to a CatalogSource located in the new global catalog namespace **openshift-marketplace**.

### Workaround for existing Subscriptions in the previous global catalog namespace

If you have existing CatalogSources in the old **openshift-operator-lifecycle-manager** namespace, any existing Subscription objects that are referring to the CatalogSource will fail to upgrade, and new Subscription objects that are referring to the CatalogSource will fail to install.

To workaround such upgrade failures:

### Procedure

1. Move the CatalogSource object from the previous global catalog namespace to the **openshift-marketplace** namespace.

## 1.3.1. Deprecated features

### Deprecation of the Service Catalog, the Template Service Broker, the Ansible Service Broker, and their Operators

In OpenShift Dedicated 4, the Service Catalog, the Template Service Broker, the Ansible Service Broker, and their Operators are deprecated. They will be removed in a future OpenShift Dedicated release.

The following related APIs will be removed in a future release:

- **\*.servicecatalog.k8s.io/v1beta1**
- **\*.automationbroker.io/v1alpha1**
- **\*.osb.openshift.io/v1**

### Deprecation of cluster role APIs

The following APIs are deprecated and will be removed in a future release:

- **ClusterRole.authorization.openshift.io** - Use **ClusterRole.rbac.authorization.k8s.io** instead.

- **ClusterRoleBinding.authorization.openshift.io** - Use **ClusterRoleBinding.rbac.authorization.k8s.io** instead.
- **Role.authorization.openshift.io** - Use **Role.rbac.authorization.k8s.io** instead.
- **RoleBinding.authorization.openshift.io** - Use **RoleBinding.rbac.authorization.k8s.io** instead.

### Deprecation of OperatorSources

In a future release, OperatorSources will be deprecated from OperatorHub and the **operatorsource.operators.coreos.com/v1** API will be removed.

### Deprecation of /oapi endpoint from oc

The usage of the **/oapi** endpoint from **oc** is being deprecated and will be removed in a future release. The **/oapi** endpoint was responsible for serving non-group OpenShift Dedicated APIs and was removed in 4.1.

### Deprecation of the -short flag of oc version

The **oc version --short** flag is now deprecated. The **--short** flag printed default output.

### Recycle reclaim policy

Recycle reclaim policy is now deprecated. Dynamic provisioning is recommended.

## 1.4. BUG FIXES

### Builds

- Blocked registries were not set in **registries.conf** used by Buildah. Therefore, Buildah could push an image to a registry blocked by the cluster image policy. With this bug fix, the **registries.conf** file generated for builds now includes blocked registries. Builds now respect the blocked registries setting for image pull and push. ([BZ#1730722](#))
- When shell variables were referenced in build configurations that used the source-to-image build strategy, logic that attempted to produce a Dockerfile, which could be used to perform the source-to-image build, would incorrectly attempt to evaluate those variables. As a result, some shell variables would be erroneously evaluated as empty values, leading to build errors, and other variables would trigger error messages from failed attempts to evaluate them. Shell variables referenced in build configurations are now properly escaped, so that they are evaluated at the expected time. These errors should no longer be observed. ([BZ#1712245](#))
- Due to a logic bug, attempts to push an image to a registry after it was built would fail if the build's BuildConfig specified an output of type DockerImage, but the name that was specified for that output image did not include a tag component. The attempt to push a built image would fail. The builder now adds the "latest" tag to a name if one is not specified. An image built using a BuildConfig specifying an output of type DockerImage, with a name that does not include a tag component, will now be pushed using the "latest" tag. ([BZ#1746499](#))

### Containers

- **rshared** propagation might cause the **/sys** filesystem to recursively mount on top of itself, causing container fails to start with "no space left on device" errors. This bug fix prevents that there are recursive **/sys** mounts on top of each other, and as a result containers run correctly with the **rshared: true** option set. ([BZ#1711200](#))
- When the Dockerfile builder handled **COPY** instructions that used the **--from** flag to specify content be copied from an image rather than either builder context or a previous stage, the image's name could be logged as though it had been specified in a **FROM** instruction. The name

would be listed multiple times if multiple **COPY** instructions specified it as the argument to a **--from** flag. This bug fix ensures the builder no longer attempts to trigger the pulling of images that are referred to in this way at the start of the build process. As a result, images that are referenced in **COPY** instructions using the **--from** flag are no longer pulled until their contents are required, and the build log no longer logs a **FROM** instruction that specifies the name of such an image. ([BZ#1684427](#))

- Logic which handled **COPY** and **ADD** instructions in cases where the build context directory included a **.dockerignore** file would not correctly handle some symbolic links and subdirectories. An affected build would fail while attempting to process a **COPY** or **ADD** instruction that triggered the bug. This bug fix extends the logic which handles this case, and as a result these errors should no longer occur. ([BZ#1707941](#))

## Images

- Long running jenkins agent or slave pods can experience the defect process phenomenon that has previously been observed with the jenkins master. Several defect processes show up in process listings until the pod is terminated. This bug fix employs **dumb-init** with the OpenShift Jenkins master image to clean up these defect processes, which occur during jenkins job processing. As a result, process listings within agent or slave pods, and on the hosts those pods reside, no longer include the defunct processes. ([BZ#1705123](#))
- Changes to OAuth support in 4 allow for different certificate configurations between the Jenkins service account certificate and the certificate used by the router for the OAuth server. As a result, you could not log into the Jenkins console. With this bug fix, the OpenShift Dedicated Jenkins login plug-in was updated to attempt TLS connections with the default certificates available to the JVM in addition to the certificates mounted into the pod. You can now log into the jenkins console. ([BZ#1709575](#))
- The OpenShift Dedicated Jenkins Sync plug-in confused ImageStreams and ConfigMaps with the same name when processing them for Jenkins Kubernetes plug-in PodTemplates, causing an event for one type to be able to delete the pod template created from another type. With this bug fix, the OpenShift Dedicated Jenkins Sync plug-in was modified to keep track of which API object type created the pod template of a given name. Now, Jenkins Kubernetes plug-in PodTemplates created by the OpenShift Dedicated Sync plug-in's mapping from ConfigMaps and ImageStreams are not inadvertently deleted when two types with the same name exist in the cluster. ([BZ#1711340](#))
- Quick, successive, deletes of the Samples Operator configuration object could lead to the last delete hanging and the Operator stuck in it's **ImageChangesInProgress** condition stuck in **True**, which resulted in the **clusteroperator** object for the Samples Operator being stuck in **Progressing==True**, causing indeterminate state for cluster samples. This bug fix introduced corrections to the coordination between the delete finalizer and Samples upsert. Quick, successive deletes of the Samples Operator configuration object now work as expected. ([BZ#1735711](#))
- Previously, the pruner was getting all images in a single request, which caused the request to take too long. This bug fix introduced the use of the pager to get all of the images. Now the pruner can get all of the images without timing out. ([BZ#1702757](#))
- Previously the importer could only import up to three signatures, but [registry.redhat.io](#) often has more than three signatures. This caused signatures to not be imported. This bug fix increased the limit of the importer so signatures can now be imported. ([BZ#1722568](#))

## Web Console

- Previously, console Operator logs for events would print some duplicate messages. A version update for a dependency repository has resolved this issue and messages are no longer being duplicated in console Operator logs. ([BZ#1687666](#))
- Users were not able to copy the whole webhook URL since the secret value was obfuscated. A link was added so that users are now able to copy the entire webhook URL with the secret value included. ([BZ#1665010](#))
- The Machine and Machine Set details pages in the web console did not contain an **Events** tab. An **Events** tab is now added and is now available from the **Machine** and **Machine Set** details pages. ([BZ#1693180](#))
- Previously, users could not view a node's status from its details page in the web console. A status field has been added and users can now view a node's status from its details page. ([BZ#1706868](#))
- Previously, you would occasionally see a blank popup in the web console if you attempted to create an Operator resource immediately after installing an Operator through the OperatorHub. With this bug fix, a clear message is now shown if you attempt to create a resource before it is available. ([BZ#1710079](#))
- Previously the Deployment Config Details page in the web console would say that the status was **Active** before the first revision had rolled out. With this bug fix, the status now says **Updating** before a rollout has occurred, and **Up to date** when the rollout is complete. ([BZ#1714897](#))
- Previously, the metrics charts for nodes in the web console could incorrectly total usage for more than one node in some circumstances. With this bug fix, the node page charts now correctly display the usage only for that node. ([BZ#1720119](#))
- Previously, the ca.crt value for OpenID identity providers was not set properly when created through the web console. The problem has been fixed, and the ca.crt is now correctly set. ([BZ#1727282](#))
- Previously, users would see an error in the web console when navigating to the ClusterResourceQuota instances from the CRD list. The problem has been fixed, and you can now successfully list ClusterResourceQuota instances from the CRD page. ([BZ#1742952](#))
- Previously, the web console did not show when a node was unscheduleable in the node list. This was inconsistent with the CLI. The console now shows when a node is unscheduleable from the node list and node details pages. ([BZ#1748405](#))
- Previously, the web console would show config map and secret keys with all caps styling in the resource details pages. This is a problem as key names are often file names and case sensitive. The OpenShift Dedicated 4 web console now shows config map and secret keys in their proper case. ([BZ#1752572](#))

## oc

- The wrong validation for node selector labels was causing empty values for keys on labels to not be accepted. This update fixes the node selector label validation mechanism so that an empty value for a key on label is a valid node selector. ([BZ#\\*1683819](#))
- The **oc get** command was not returning the proper information when it received an empty result list. This update improves the information that is returned when **oc get** receives an empty list. ([BZ#1708280](#))

## Templates

- Previously, the custom resource definition for the Samples Operator configuration object (**configs.samples.operator.openshift.io**) did not have openAPIV3Schema validation defined. Therefore, **oc explain** was unable to provide useful information about the object. With this fix, openAPIV3Schema validation was added, and now **oc explain** works on the object. ([BZ#1705753](#))
- Previously, the Samples Operator was using a direct OpenShift Dedicated go client to make GET calls in order to maintain controller/informer based watches for secrets, imagestreams, and templates. This resulted in unnecessary API calls were being made against the OpenShift Dedicated API server. This fix leverages the informer/listener API and reduces activity against the OpenShift Dedicated API server. ([BZ#1707834](#))
- Previously, the Samples Operator was not creating a cluster role that aggregated into the cluster-reader role. As a consequence, users with the cluster-reader role could not read the config object for the samples Operator. With this update, the manifest of the samples operator was updated to include a cluster role for read-only access to its config object, and this role aggregated into the cluster-reader role. Now, users with the cluster-reader role can read, list, and watch the config object for the samples Operator. ([BZ#1717124](#))