OpenShift Container Platform 4.2

Release notes

Highlights of what is new and what has changed with the OpenShift Container Platform 4.2 release

Last Updated: 2019-10-31

OpenShift Container Platform 4.2 Release notes

Highlights of what is new and what has changed with the OpenShift Container Platform 4.2 release

Legal Notice

Abstract

The release notes for OpenShift Container Platform 4.2 summarize all new features and enhancements, notable technical changes, major corrections from the previous version, and any known bugs upon general availability.

Table of Contents

CHAPTER 1. OPENSHIFT CONTAINER PLATFORM 4.2 RELEASE NOTES

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

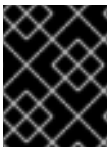
1.1. ABOUT THIS RELEASE

Red Hat OpenShift Container Platform (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 Container Platform 4.2 are included in this topic.

OpenShift Container Platform 4.2 clusters are available at https://cloud.redhat.com/openshift. The Red Hat OpenShift Cluster Manager application for OpenShift Container Platform allows you to deploy OpenShift clusters to either on-premise or cloud environments.

OpenShift Container Platform 4.2 is supported on Red Hat Enterprise Linux 7.6 and later, as well as Red Hat Enterprise Linux CoreOS 4.1.

You must use Red Hat Enterprise Linux CoreOS (RHCOS) for the control plane, or master, machines and can use either RHCOS or Red Hat Enterprise Linux 7.6 for compute, or worker, machines.



IMPORTANT

Because only Red Hat Enterprise Linux version 7.6 is supported for compute machines, you must not upgrade the Red Hat Enterprise Linux compute machines to version 8.

1.2. NEW FEATURES AND ENHANCEMENTS

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

1.2.1. Installation and upgrade

1.2.1.1. OpenShift Container Platform upgrades phased rollout

In OpenShift Container Platform 4.1, Red Hat introduced the concept of upgrade channels for recommending the appropriate upgrade versions to your cluster. Upgrade channels separate upgrade strategies and also are used to control the cadence of updates. Channels are tied to a minor version of OpenShift Container Platform. For instance, OpenShift Container Platform 4.2 channels will never include an upgrade to a 4.3 release. This ensures administrators make an explicit decision to upgrade to the next minor version of OpenShift Container Platform. Channels only control updates and have no impact on the version of the cluster you install; the **openshift-install** binary for a given patch level of OpenShift Container Platform always installs that patch level.

See OpenShift 4.2 Upgrades phased roll out for more information on the types of updates and upgrade channels.

Because upgrades are published to the channels as they are gradually rolled out to customers based on data from the Red Hat Service Reliability Engineering (SRE) team, you might not immediately see notification in the web console that updates from version 4.1.z to 4.2 are available at initial release.

1.2.1.2. CLI-based installation

OpenShift Container Platform 4.2 introduces a new CLI-based installer called **openshift-install**. It is designed to simplify provisioning OpenShift on immutable installer provisioned infrastructure using an interactive guided workflow within about 30 minutes. This approach provides an OpenShift deployment with full stack automation of host OS and platform updates and infrastructure management without the complexity of having to provision your own infrastructure. Only minimal user input is needed with all non-essential installation configuration options, which are now configured post-installation via component operator Custom Resource Definitions (CRDs).

Updates are performed largely the same way. OpenShift update content must be mirrored first to the local container registry first then the administrator tells **oc adm upgrade** where to pull the update content from.

1.2.1.3. Installations in restricted networks

OpenShift Container Platform 4.2 introduces support for installation and updating OpenShift Container Platform clusters in restricted network environments. It is designed to work with any **docker** 2.2 spec-compliant container registry for hosting OpenShift Container Platform content.

Administrators first must replicate content from Quay.io to their local container registry. After that is done, **openshift-install** can be configured to generate Ignition configs that pull content locally rather than from Quay.io. This is designed to work with user-provisioned infrastructure (UPI) deployments only.

You can still use Operators from the OLM catalog in restricted networks. However, you must manually pull the Operator sources in order to populate the the offline catalog. This manual process is only a temporary workaround for OpenShift Container Platform 4.2. An automated solution will be provided in a future release.

See Installing in restricted networks and Using Operator Lifecycle Manager on restricted networks for details.

1.2.1.4. Cluster-wide egress proxy

OpenShift Container Platform 4.2 introduces support for installing and updating an OpenShift Container Platform cluster through a corporate proxy server on user-provisioned infrastructure. Proxy information (httpProxy, httpsProxy, and noProxy) can be defined in the **install-config.yaml** file, which is used during the installation process and can also be managed post-installation via the **cluster** Proxy object.

IMPORTANT

The cluster-wide proxy is only supported if you used a user-provisioned infrastructure installation for a supported provider.

Also, there is now support for providing your own CA bundles allowing the corporate proxy to MITM HTTPS.

1.2.1.5. New platform boundary

OpenShift Container Platform is now aware of the entire infrastructure and brings the operating system under management. This makes installation and updates seamless across the platform and the underlying operating system. Everything is managed as a single entity.

1.2.1.6. IPI and UPI

In OpenShift Container Platform 4.2, there are two primary installation experiences: Full stack automation (IPI) and pre-existing infrastructure (UPI).

With full stack automation, the installer controls all areas of the installation including infrastructure provisioning with an opinionated best practices deployment of OpenShift Container Platform. With pre-existing infrastructure deployments, administrators are responsible for creating and managing their own infrastructure allowing greater customization and operational flexibility.

1.2.1.7. Full stack automated deployments

In OpenShift Container Platform 4.2, there is expanded support for full stack automated deployments to include AWS, Microsoft Azure, Google Cloud Platform (GCP), and Red Hat OpenStack Platform, as well as adding GCP to the existing list of user provisioned infrastructure supported providers that already includes AWS, Bare Metal, and VMware vSphere.

1.2.1.8. Red Hat Cluster Application Migration Tool and Red Hat Control Plane Migration Assistant

See Where can I find Red Hat Cluster Application Migration Tool (CAM) and Red Hat Control Plane Migration Assistant (CPMA) now that OpenShift 4.2 has GA'ed for information on CAM and CPMA.

1.2.2. Operators

1.2.2.1. New location for Operator product documentation

The OpenShift Container Platform product documentation related to Operators that was previously found in the **Applications** guide is now located in the new **Operators** guide. This includes existing and updated information on Operators, the Operator Lifecycle Manager, and the Operator SDK, as well new content specific to OpenShift Container Platform 4.2.

1.2.2.2. Scoped Operator installations

Previously, only users carrying **cluster-admin** roles were allowed to install Operators. In OpenShift Container Platform 4.2, the **cluster-admin** can select namespaces in which namespace administrators can install Operators self-sufficiently. The **cluster-admin** defines the ServiceAccount in this namespace; all installed Operators in this namespace get equal or lower permissions of this ServiceAccount.

See Creating policy for Operator installations and upgrades for details.

1.2.2.3. Ingress Operator

The Ingress Operator supports all ingress features on 4.2 with installer-provisioned infrastructure on Azure and GCP.

1.2.2.4. Machine Config Operator

The Machine Config Operator (MCO) provides cluster-level configuration, enables rolling upgrades, and prevents drift between new and existing nodes.

1.2.2.5. Node Feature Discovery Operator

The Node Feature Discovery (NFD) Operator detects hardware features available on each node and advertises those features using node labels.

CPU features managed by NFD include:

- cpuid
- hardware_multithreading
- power
- pstate

Kernel features managed by NFD include:

- config
- selinux_enabled
- version
- os_version

Other features managed by NFD include:

- NVMe
- NUMA
- SR-IOV
- GPUs

The NFD Operator manages the installation and lifecycle of the NFD DaemonSet. Access the NFD Operator in OperatorHub.

1.2.2.6. Node Tuning Operator enhancements

The Node Turning Operator was first introduced in OpenShift Container Platform 4.1 and manages cluster node-level tuning; The default CR is meant for delivering standard node-level tuning. The enhancements in OpenShift Container Platform 4.2 allow for customizing the tunings for things such as high performance.

For custom tuning, create your own tuned custom resources (CRs). Newly created CRs will be combined with the default CR and custom tuning applied to nodes based on node or pod labels and profile priorities.

1.2.3. Storage

1.2.3.1. Persistent volumes using the Local Storage Operator

Persistent volumes using the Local Storage Operator is now available in OpenShift Container Platform 4.2.

1.2.3.2. OpenShift Container Storage Interface (CSI)

Container Storage Interface (CSI) is now available in OpenShift Container Platform 4.2. CSI is introduced in Kubernetes to enable Red Hat OpenShift Container Storage (OCS) and partners with their CSI plug-ins. With the adoption of the CSI, the Kubernetes volume layer becomes truly extensible.

For now, only the API is available. CSI drivers contained in operators will be available in future releases.

1.2.3.3. Raw block volume support

The following raw block volumes are now fully supported with OpenShift Container Platform 4.2:

- Local volumes
- Cloud providers (AWS, GCP, Azure, and vSphere)

Raw block volumes using iSCSI are now in Technology Preview.

1.2.4. Scale

1.2.4.1. Cluster limits

Updated guidance around Cluster Limits for OpenShift Container Platform 4.2 is now available.

Use the OpenShift Container Platform Limit Calculator to estimate cluster limits for your environment.

1.2.5. Developer experience

1.2.5.1. 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 Container Platform 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.

1.2.5.2. CodeReady Containers

CodeReady Containers provides a local desktop instance of a minimal OpenShift Container Platform 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.

1.2.6. Nodes

1.2.6.1. CRI-O support

CRI-O is a kubernetes-specific container engine that tracks and versions identical to Kubernetes, simplifying support permutations. Adoption is trivial because all existing Docker and OCI containers are supported and run well with CRI-O. CRI-O is a light weight, kubernetes-native, OCI-compatible container runtime that is life-cycled and managed by OpenShift Container Platform. You do not need to worry about which container runtime is in use. With OpenShift Container Platform, it is always the right one and it provide a complete implementation of the Kubernetes Container Runtime Interface (CRI).

Also, you do not need to separately manage the container engine. CRI-O has some tuneables that provide control and security for CRI-O; they are easily configured in a CRD, and the settings are propagated across the cluster.

1.2.6.2. Whitelisting of sysctls configuration

System administrators can whitelist sysctl on a per-node basis. All safe sysctls are enabled by default; all unsafe sysctls are disabled by default. See Using sysctls in containers for more information.

1.2.6.3. Master nodes are now schedulable

In OpenShift Container Platform 4.2, master nodes are schedulable. See Working with nodes for more information.

1.2.7. Networking

1.2.7.1. Installer-provisioned OpenShift on OpenStack

OpenShift Container Platform 4.2 introduces full-stack automation support for deploying OpenShift Container Platform clusters on Red Hat OpenStack.

The installer uses the OpenStack APIs in conjunction with the Kubernetes OpenStack Cloud Provider to create all the required OpenStack resources, such as virtual machines (VMs), networks, security groups, and object storage needed to deploy OpenShift Container Platform and properly configure the cluster to run on the Red Hat OpenStack Platform.

OpenShift Container Platform 4.2 can be deployed on Red Hat OpenStack version 13.

1.2.7.2. OVN (Technology Preview)

Open Virtual Networking (OVN) for Open vSwitch, currently in Technology Preview, has many advantages, including acceleration of customer-driven feature requirements, some of which are pre-enabled, including:

- Low barrier to integration; an implementation of virtual networking via OVS.
- SDN portfolio consolidation.
- Virtually eliminate iptables scale issues.
- Heterogeneous clusters with Windows nodes.
- The capability to span on-premise and cloud nodes
- Full Network Policy support.
- Egress IP per pod.
- Distributed L4 Ingress/Egress firewall.
- Distributed services load balancer.
- Traffic isolation and multi-tenancy.
- Data Plane Development Kit (DPDK) support.

- Encrypted tunnels.
- IPv6 and DHCPv6.
- QoS, control and data plane separation.

1.2.7.3. Enable internal Ingress Controllers for private clusters

When creating an Ingress Controller on cloud platforms, the Ingress Controller is published by a public cloud load balancer by default.

Users can now publish Ingress Controllers with internal cloud load balancers. For example:

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  namespace: openshift-ingress-operator
  name: internal
spec:
  endpointPublishingStrategy:
    type: LoadBalancerService
    loadBalancer:
      scope: Internal
```

See the Kubernetes Services documentation for implementation details.

Once set, **.spec.endpointPublishingStrategy.loadBalancer.scope** cannot be changed. To change the scope, delete and recreate the Ingress Controller.

The **default** Ingress Controller can be made internal by deleting and recreating it.

See Ingress Operator in OpenShift Container Platform for more information.

1.2.7.4. Kubernetes CNI plug-in additions and enhancements

Several Kubernetes CNI plug-ins are added or enhanced in OpenShift Container Platform 4.2 to grow capability.

These SR-IOV solutions remain in Technology Preview in OpenShift Container Platform 4.2:

- RDMA and RoCE Support
- DPDK Mode for SR-IOV VFs
- Admission Controller
- Operator

New CNI plug-ins:

- IPVLAN
- Bridge with VLAN
- Static IPAM

1.2.7.5. Enablement of GPUs in an OpenShift cluster

GPUs are now supported on Red Hat Enterprise Linux (RHEL) 7 nodes. They are not supported on RHEL CoreOS nodes because of lack of support for CUDA driver and driver containers.

1.2.8. Web console

1.2.8.1. Console customization options

You can customize the OpenShift Container Platform 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.

See Customizing the web console for more information.

1.2.8.2. 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.

1.2.8.3. Machine Autoscaler

You can now scale your cluster with the Machine Autoscaler. The Machine Autoscaler adjusts the number of Machines in the MachineSets being deployed in your cluster. Increase Machines when the cluster runs out of resources to support more deployments. Any changes, such as the minimum or maximum number of instances, are immediately applied to the MachineSet that MachineAutoscalers target.

1.2.8.4. 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 Container Platform 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.

1.2.8.5. Prometheus queries

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

1.2.8.6. Identity providers

On the cluster OAuth configuration page, more identity providers (IDPs) are provided for the user to log in to the cluster. The IDPs include GitHub, GitLab, Google, LDAP, Keystone, and so on.

1.2.8.7. 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 Container Platform 4.2 introduces the following notable technical changes.

corsAllowedOrigins
corsAllowedOrigins can now be configured. See Allowing JavaScript-based access to the API server from additional hosts for more information.

New CNI plug-ins
There are two new CNI plug-ins for Multus: bridge and ipvlan.

Cluster Network Operator supports SimpleMacvlan
Cluster Network Operator (CNO) now supports configuring SimpleMacvlan.

Builds maintain their layers
In OpenShift Container Platform 4.2, builds keep their layers by default.

Builds on Windows
Builds are not scheduled on Windows nodes.

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

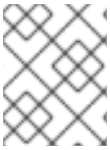
New and upgraded ingress controllers will no longer support these TLS versions.

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 Container Platform 4.2.

To work with OperatorHub in OpenShift Container Platform 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 Container Platform 4.2 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 Container Platform.

Global catalog namespace change
In OpenShift Container Platform 4.1, the default global catalog namespace, where CatalogSources are installed by default, is **openshift-operator-lifecycle-manager**. Starting with OpenShift Container Platform 4.2, this has changed to the **openshift-marketplace** namespace.

If you have installed an Operator from OperatorHub on an OpenShift Container Platform 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 Container Platform 4.2, 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 Container Platform 4.2, the Service Catalog, the Template Service Broker, the Ansible Service Broker, and their Operators are deprecated. They will be removed in a future OpenShift Container Platform 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 Container Platform 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.

oc adm migrate commands
The **oc adm migrate** command and all of its subcommands except for **oc adm migrate template-instances** are now deprecated.

Persistent volume snapshots
Persistent volume snapshots are deprecated in OpenShift Container Platform 4.2.

EFS
In the OpenShift Container Platform 4.1 Release Notes, EFS was incorrectly marked as generally available. This is being included as a Technology Preview feature in OpenShift Container Platform 4.2.

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)

Cloud Credential Operator

- Previously, there was a memory limit on the Pod. As a result, the Credential Operator could

crash on clusters with large numbers of projects/namespaces. With this bug fix, the memory limit is removed, the Operator no longer crashes, and memory is handled by the cluster itself. (BZ#1711402)

- The **cloud-credential** ClusterOperator did not define related resources, causing Operator logs to not be present in tarballs generated by the **oc adm must-gather** command. This bug fix updates the Operator to add related resources, and as a result logs are now included. (BZ#1717631)

Containers

- **rshared** propogation 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 4.2 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 Container Platform 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 Container Platform 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 Container Platform 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 Container Platform 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)

Image Registry

- Previously the CRD did not have OpenAPI schema, which meant **oc explain** did not work for the **config.imageregistry** resource. This bug fix enabled the generation of the OpenAPI schema, so **oc explain** can now provide information about the **config.imageregistry.operator.openshift.io** resource. (BZ#1705752)

- The Image Registry Operator did not register the **openshift-image-registry** namespace as a related object. In some situations, no data from the image registry or Image Registry Operator would be collected from **must-gather**. This bug fix ensures the **openshift-image-registry** namespace is always included in the Image Registry Operator's related objects. As a result, basic information from the **openshift-image-registry** namespace including pods, deployments, and services is always collected by **must-gather**. (BZ#1748436)

- Previously the CVO, the Image Registry Operator, and the **service-ca** injection controller simultaneously watched and updated the CA bundle used by the image registry. This caused the CA bundle used to establish trust between the internal registry and the rest of OpenShift to be constantly removed and recreated. With this bug fix, the CVO no longer manages the CA bundle. The Image Registry Operator ensures that the ConfigMap to hold the CA bundle is created, but does not manage its content. Now the ConfigMap holding the CA bundle for the internal registry is only updated as needed by the **service-ca** injection controller. (BZ#1734564)

- TLS keys were not added to registry routes. This is because TLS keys were stored in **Secret.StringData** and the Operator was unable to see the real data in the secret. Now, Secret.Data is used instead and the Operator can see the values. (BZ#1719965)

- The drain process would take up to 600 seconds to evict the image-registry pod. This was because the image registry was running from sh and signals were not propagated to the image registry, and unable to receive SIGTERM. Now, the registry process uses exec and the registry is the pid 1 process and able to receive SIGTERM. Drain now evicts successfully. (BZ#1729979)

- **must-gather** did not collect PVCs and events in the **openshift-image-registry** namespace. Now, PVCs are collected as part of the **must-gather** process. (BZ#1737558)

Installer

- Default minimal installations of Red Hat Enterprise Linux would install and enable firewalld.service. The firewall service would block ports that prevented **oc rsh/exec/logs** from working as expected. Now, **firewalld.service** is disabled if it is installed to conform to testing standards. (BZ#1740439)

- Workers were not created when the default path for images was changed from the default settings. Now, the installer creates and uses it's own storage pool. (BZ#1707479)

kube-apiserver

- It was possible to shorten certificate rotation by creating a ConfigMap. This behavior is not supported. Now, if you create this ConfigMap, **Upgradable** is set to **False** on the kubeapiserver-operator, which means that you can no longer update your cluster. (BZ#1722550)

Logging

- The dynamic seeding function of Elasticsearch was inefficient, and clusters with a large number of projects made too many calls. This issue was compounded by a lack of caching. Because of the inefficient seeding and lack of caching, calls to Elasticsearch were timing out before a response was returned. This bug fix added API call caching and ACL seeding. These improvements reduce the opportunity for page timeouts. (BZ#1705589)

- The Logging Operator relied on the type of information stream to set the log type, so logs sent to **stdout** were tagged as **INFO** and logs sent to **stderr** were tagged as **ERROR**. This method does not always correctly convey the information type. Now, the log level is no longer set based on the information stream. Instead, it is set to **unknown** if the correct log level cannot be determined. (BZ#1726639)

- During Cluster Logging instance deletion, resources under Cluster Logging were deleted independently. Therefore, there could be a short time period when the Fluentd or Rsyslog DaemonSet was still running but its log collector service account was removed. This made the logs processed in this time miss Kubernetes information, including the namespace name. With this bug fix, the service accounts now wait to be deleted until all descendant resources are deleted. There is now no chance for the collector DaemonSets to run without the log collector service account. (BZ#1715370)

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 Container Platform 4.2 web console now shows config map and secret keys in their proper case. (BZ#1752572)

Networking

- Egress IP addressess did not work correctly in namespaces with restrictive NetworkPolicies. As a result, Pods that accepted traffic from specific sources would not be able to send egress traffic through egress IPs, because the response from the external server would be mistakenly rejected by their NetworkPolicies. With this bug fix, replies from egress traffic are now correctly recognized as replies rather than as new connections. Egress IPs and NetworkPolicy work together. (BZ#1700431)

- If a pod using an external IP address to contact an external host received an external IP address not responding condition, the egress IP monitoring code mistakenly determined that the host was not responding. As a result, highly-available egress IPs could get switched to a different node to another. The monitoring code was fixed to distinguish between a egress node not responding and a final destination not responding conditions. Highly available egress IPs are not be switched between nodes unnecessarily. (BZ#1717639)

- By default, the etcd namespace was created without specifying a net id. As a result, API server components could not connect to etcd. The code was fixed to specify **netid=1** for the etcd namespace. (BZ#1719653)

Node

- The algorithm used when merging additional IP addresses added to a node was incorrect. As a result, when adding an additional IP address to a node, the list of addresses was out of order, resulting in the node being unable to communicate to the API server. The merge algorithm for addresses was changed to not reorder the addresses. Adding secondary IP addresses to a node no longer changes the ordering and the node is able to continue communication with the API server. (BZ#1696628)

- Because of a problem with the kubeConfig controller, changes to the kubelet Config are reverted if the cluster is upgraded to a version that uses a different OS release. The code was fixed to specify the correct controller in the source. Customizations to the kubeletConfig will be retained. (BZ#1718726)

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)

OLM

- Marketplace Cluster Operator was reporting **degraded** when restarting or upgrading the Marketplace Operator. Therefore, the OpenShift Container Platform upgrade tests were failing. With this bug fix, the upgrade tests are passing again because OpenShift Container Platform is no longer reporting **degraded** when the Operator is stopped. (BZ#1706867)

- A bug during upgrade was causing Marketplace Operator to degrade and exit. The ClusterOperator status did not give an accurate description of the health of the Marketplace Operator.
  The following issues were contributing to this:

  - Multiple Marketplace Operators were running and updating the ClusterOperatorStatus at the same time.

  - The sync would fail if an error occurred while reconciling an operand (OperatorSources or CatalogSourceConfigs). This allowed network issues or invalid operands to drive to a degraded state. The Marketplace Operator also degraded when failed syncs surpassed a threshold of total syncs.

  - It was difficult to identify why a ClusterOperatorStatus condition was in a given state via telemetry. Marketplace did not set a reason, although telemetry includes the state and reason.
    With this bug fix:

  - Leader election provided by the Operator SDK prevents multiple Marketplace Operators from updating the ClusterOperatorStatus at the same time.

  - Marketplace only reports a degraded state when it is unable to get or update its operands, rather than whenever an error is encountered while reconciling an operand.

  - Marketplace now includes a reason when setting a condition so that telemetry provides better insights into the state of the Marketplace Operator. (BZ#1721537)

- The CatalogSourceConfig (**csc**) and OperatorSource (**opsrc**) Custom Resource Definitions (CRDs) owned by Marketplace did not include a description. Therefore, **oc explain csc** and **oc explain opsrc** would return empty descriptions. With this bug fix, OpenAPI CRD definitions are now added so that **oc explain csc** and **oc explain opsrc** now work. (BZ#1723835)

- The Marketplace Operator was overwriting the Pod deployment spec of registry deployments associated with OperatorSources. Therefore, users were unable to add NodeSelectors. With this bug fix, required fields are only replaced in the deployment spec on OperatorSource updates, allowing users to add NodeSelectors to the Operator registry Pod associated with OperatorSources. By default, no NodeSelector is present. Users can now add NodeSelectors to the Pod spec in the registry Pod deployments. For example, with the **community-operators** OperatorSource, you would edit the **community-operators** deployment in the **openshift-marketplace** namespace. (BZ#1696726)

- The Marketplace Operator was scaling the registry deployment down and then scaling it back up to force updates. This could cause the Pod to crash-loop if there was an issue in the registry Pod. This bug fix uses annotation to force the update, rather than scaling the registry deployment up and down so that the deployment will not become unavailable. Note that this alone will not fix the bug. A fix is required to the end-to-end test that does not fail it on crash-looping Pods in the 'openshift-marketplace' namespace. (BZ#1700100)

- The must-gather tool requires a field, **RelatedObjects**, in the ClusterOperator Custom Resource to be populated with ObjectReferences of the resources associated with the Operator. Because this field was missing for Marketplace, the must-gather tool was not able to gather enough information about the Marketplace Operator. This update now populates the **RelatedObjects** field with the Operator's namespace and the OperatorSource/CatalogSourceConfig/CatalogSource resources. This enables the must-gather tool to gather enough information about the Marketplace Operator. (BZ#1717439)

OpenShift Controller Manager

- Setting the OpenShift Controller Manager Operator to **Unmanaged** or **Removed** is unsupported, so it would cause the conditions on the corresponding ClusterOperator object to go to an **Unknown** status. With this bug fix, the OpenShift Controller Manager Operator now ignores the unsupported **Unmanaged' and `Removed** settings for management state. A message now explains this in the ClusterOperator status conditions. (BZ#1719188)

Red Hat Enterprise Linux CoreOS (RHCOS)

- Previously, SSH connections were hanging when the ClientAliveInterval within the sshd configuration was not set to 180 as required by Microsoft Azure. This bug fix now defaults the sshd config to 180 so that SSH no longer hangs within Azure. (BZ#1701050)

Service Broker

- Previously, the Automation Broker always created a network policy to give a transient namespace access to the target namespace. As a consequence, the target namespace locked down to the newly created policy and namespaces could communicate with each other. This fix causes the Automation Broker to check if there are network policies in place for the target namespace, and if there are none, to not create a new network policy. This fix allows the Automation Broker to perform Ansible Playbook Bundle actions actions without affecting the existing services running in the target namespace. (BZ#1643303)

- Previously, the OpenShift Ansible Service Broker Operator did not pass metrics to Prometheus unless the correct permissions were manually applied. With this update, the Operator now automatically installs with the required permissions. (BZ#1692281)

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 Container Platform 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 being made against the OpenShift Container Platform API server. This fix leverages the informer/listener API and reduces activity against the OpenShift Container Platform 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)

1.5. TECHNOLOGY PREVIEW FEATURES

Some features in this release are currently in Technology Preview. These experimental features are not intended for production use. Note the following scope of support on the Red Hat Customer Portal for these features:

Technology Preview Features Support Scope

In the table below, features marked **TP** indicate *Technology Preview* and features marked **GA** indicate *General Availability*. Features marked as **-** indicate that the feature is removed from the release or deprecated.

Table 1.1. Technology Preview Tracker

Feature	OCP 3.11	OCP 4.1	OCP 4.2
Prometheus Cluster Monitoring	GA	GA	GA
Local Storage Persistent Volumes	TP	TP	TP
CRI-O for runtime pods	GA* [a]	GA	GA
Tenant Driven Snapshotting	TP	TP	TP
oc CLI Plug-ins	TP	TP	TP
Service Catalog	GA	GA	–
Template Service Broker	GA	GA	–
OpenShift Ansible Service Broker	GA	GA	–
Network Policy	GA	GA	GA
Multus	-	GA	GA
New Add Project Flow	GA	GA	GA
Search Catalog	GA	GA	GA

Feature	OCP 3.11	OCP 4.1	OCP 4.2
Cron Jobs	GA	GA	GA
Kubernetes Deployments	GA	GA	GA
StatefulSets	GA	GA	GA
Explicit Quota	GA	GA	GA
Mount Options	GA	GA	GA
System Containers for Docker, CRI-O	-	-	-
Hawkular Agent	-	-	-
Pod PreSets	-	-	-
experimental-qos-reserved	TP	TP	TP
Pod sysctls	GA	GA	GA. See Known issues for current limitations.
Central Audit	GA	-	-
Static IPs for External Project Traffic	GA	GA	GA
Template Completion Detection	GA	GA	GA
replicaSet	GA	GA	GA
Fluentd Mux	TP	TP	TP
Clustered MongoDB Template	-	-	-
Clustered MySQL Template	-	-	-
ImageStreams with Kubernetes Resources	GA	GA	GA

Feature	OCP 3.11	OCP 4.1	OCP 4.2
Device Manager	GA	GA	GA
Persistent Volume Resize	GA	GA	GA
Huge Pages	GA	GA	GA
CPU Pinning	GA	GA	GA
Admission Webhooks	TP	GA	GA
External provisioner for AWS EFS	TP	TP	TP
Pod Unidler	TP	TP	TP
Node Problem Detector	TP	TP	TP
Ephemeral Storage Limit/Requests	TP	TP	TP
Descheduler	TP	TP	TP
CephFS	TP	TP	TP
Podman	TP	TP	TP
Kuryr CNI Plug-in	TP	TP	TP
Sharing Control of the PID Namespace	TP	TP	TP
Manila Provisioner	TP	TP	TP
Cluster Administrator console	GA	GA	GA
Cluster Autoscaling	GA	GA	GA
Container Storage Interface (CSI)	TP	TP	GA
Operator Lifecycle Manager	TP	GA	GA

Feature	OCP 3.11	OCP 4.1	OCP 4.2
Red Hat OpenShift Service Mesh	TP	GA	GA
"Fully Automatic" Egress IPs	GA	GA	GA
Pod Priority and Preemption	GA	GA	GA
Multi-stage builds in Dockerfiles	TP	GA	GA
OVN SDN		TP	TP
HPA custom metrics adapter based on Prometheus		TP	TP
Machine health checks		TP	TP
Raw Block with iSCSI			TP
OperatorHub			GA

[a] Features marked with * indicate delivery in a z-stream patch.

1.6. KNOWN ISSUES

- If you have Service Mesh installed, upgrade Service Mesh before upgrading OpenShift Container Platform. See Issue OSSM-39 for more information. For a workaround, see Openshift Container Platform 4.x Upgrade Fails When OpenShift Service Mesh is Installed . (**BZ#1747472**)
- Cluster-scoped resources are not yet handled by the application migration tool, including resources such as Cluster Role Bindings and Security Context Constraints. If applications you are migrating depend on these kind of cluster-scoped resources on the source cluster, manually ensure they are recreated on the destination cluster. Coverage will be expanded in a future release to handle these resources.
- 4.2.0 Dynamic Host Configuration Protocol (DHCP) does not currently work with any of the Multus CNI plug-ins. (**BZ#1754686**)
- Cluster Loader fails when called without configuration. (**BZ#1761925**)
- The Cluster Network Operator does not a remove **NetworkAttachmentDefinition** that the Operator created previously, when the additional network is removed from the **additionalNetworks** collection. (**BZ#1755586**)

- The Prometheus Operator deploys **StatefulSet** and creates a memory limit that is too small on the **rules-configmap-reloader** container. (BZ#1735691)
- DHCP mode fails when configuring it in Multus CNI IPAM. (BZ#1754682)
- Schema change in **ClusterResourceQuota** from version 4.1 to 4.2 results in breakage. (BZ#1755125)
- Disaster recovery is broken for various deployments, including bare metal and vSphere. (BZ#1718436)
- Removing **simpleMacvlanConfig** from the Cluster Network Operator does not delete the old **network-attachment-definition**. You must delete the resources manually. (BZ#1755586)
- The SRIVO Operator pod crashes when NAD is manually created. (BZ#1755188)
- No proxy is set for **kube-controller-manager**. (BZ#1753467)
- **git clone** operations that go through an HTTPS proxy will fail. Non-TLS (HTTP) proxies can be used successfully. (BZ#1750650)
- Builds that use image references that correlate to an image mirror, which is the case in a disconnected environment, will fail to pull or push those image references if the mirror requires authentication. (BZ#1745192)
- Image stream import does not use mirrors. This is often used in disconnected environments. (BZ#1741391)
- OpenShift Container Platform 4.2 will not work on Red Hat OpenStack Platform 15 until this Red Hat OpenStack Platform 15 bug is resolved. (BZ#1751942)
- If builds use a build secret, it is strongly recommended that layers are squashed using **imageOptimizationPolicy: SkipLayers**. Otherwise, secrets might leak in **source** and **docker** strategy builds.
- AllowVolumeExpansion, and other StorageClass attributes, are not updated when upgrading OpenShift Container Platform. It is recommended to delete the default StorageClass and allow the ClusterStorageOperator to recreate this with the correct attributes after the upgrade is completed. (BZ#1751641)
- Non-serverless workloads show resources for serverless workloads in the **Topology** resources panel. (BZ#1760810)
- Pod status is depicted inconsistently in the **Topology** view, **Resources** side panel, and the **Deployment Config Details** page. (BZ#1760827)
- When an application is created using the **Add** page options, the deployed image ignores the selected target port and always uses the first entry. (BZ#1760836)
- Certain features like the name of the application and the build status are not rendered in the **Topology** view on the Edge browser. (BZ#1760858)
- Determination of active pods when a rollout fails can be incorrect in the **Topology** view. (BZ#1760828)

CHAPTER 2. OPENSHIFT CONTAINER PLATFORM VERSIONING POLICY

OpenShift Container Platform provides strict backwards compatibility guarantees for all supported APIs, excluding alpha APIs (which may be changed without notice) and beta APIs (which may occasionally be changed in a non-backwards compatible manner).

Red Hat did not publicly release OpenShift Container Platform 4.0 and, instead, released OpenShift Container Platform 4.1 directly after version 3.11.

The OpenShift Container Platform version must match between master and node hosts, excluding temporary mismatches during cluster upgrades. For example, in a 4.2 cluster, all masters must be 4.2 and all nodes must be 4.2. If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.2. You must download and install the new version of **oc**.

Changes of APIs for non-security related reasons will involve, at minimum, two minor releases (3.4 to 3.5 to 3.6, for example) to allow older **oc** to update. Using new capabilities may require newer **oc**. A 3.2 server may have additional capabilities that a 3.1 **oc** cannot use and a 3.2 **oc** may have additional capabilities that are not supported by a 3.1 server.

Table 2.1. Compatibility Matrix

	X.Y (**oc** Client)	**X.Y+N** [a] (**oc** Client)
X.Y (Server)	1	3
X.Y+N [a] (Server)	2	1

[a] Where **N** is a number greater than 1.

1 Fully compatible.

2 **oc** client may not be able to access server features.

3 **oc** client may provide options and features that may not be compatible with the accessed server.