OpenShift Container Platform 4.6

Release notes

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

Last Updated: 2020-11-30

OpenShift Container Platform 4.6 Release notes

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

Legal Notice

Abstract

The release notes for OpenShift Container Platform 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 4.6 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-2020:4196) is now available. This release uses Kubernetes 1.19 with CRI-O runtime. New features, changes, and known issues that pertain to OpenShift Container Platform 4.6 are included in this topic.

Red Hat did not publicly release OpenShift Container Platform 4.6.0 as the GA version and, instead, is releasing OpenShift Container Platform 4.6.1 as the GA version.

OpenShift Container Platform 4.6 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 4.6 is supported on Red Hat Enterprise Linux 7.7 or later, as well as Red Hat Enterprise Linux CoreOS (RHCOS) 4.6.

You must use RHCOS for the control plane, which are also known as master machines, and can use either RHCOS or Red Hat Enterprise Linux 7.7 or later for compute machines, which are also known as worker machines.

IMPORTANT

Because only Red Hat Enterprise Linux version 7.7 or later is supported for compute machines, you must not upgrade the Red Hat Enterprise Linux compute machines to version 8.

OpenShift Container Platform 4.6 is an Extended Update Support (EUS) release. More information on Red Hat OpenShift EUS is available in OpenShift Life Cycle and OpenShift EUS Overview.

With the release of OpenShift Container Platform 4.6, version 4.3 is now end of life. For more information, see the Red Hat OpenShift Container Platform Life Cycle Policy .

1.2. NEW FEATURES AND ENHANCEMENTS

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

1.2.1. Red Hat Enterprise Linux CoreOS (RHCOS)

1.2.1.1. RHCOS PXE and ISO now live environment

The PXE media and ISO available for RHCOS are now a fully live environment. Unlike the previous dedicated PXE media and ISO used for RHCOS installation for OpenShift Container Platform clusters on user-provisioned infrastructure, the RHCOS live environment can be configured with Ignition and contains all the same packages as the main RHCOS image, such as **coreos-installer**, **nmcli**, and **podman**. This allows arbitrary scripting of pre- or post-installation workflows. For example, you could run **coreos-installer** and then make an HTTP request to signal success to a provisioning server. PXE boots use the normal **ignition.config.url**. The ISO can be configured with Ignition by using the following command:

```
$ coreos-installer iso ignition embed
```

1.2.1.2. coreos-installer has been rewritten

The **coreos-installer** is now rewritten to support more features including:

- Modifying the kernel arguments of the installed system.
- Fetching Ignition configs.
- Preserving previously existing partitions.
- Configuring Ignition for the new live ISO using the **coreos-installer iso ignition** command.

1.2.1.3. Ignition Spec updated to v3

RHCOS now uses Ignition spec v3 as the only supported spec version of Ignition. This allows for more complex disk configuration support in the future.

The change should be mostly transparent for those using installer-provisioned infrastructure. For user-provisioned infrastructure installations, you must adapt any custom Ignition configurations to use Ignition spec 3. The **openshift-install** program now generates Ignition spec 3.

If you are creating Machine Configs for day 1 or day 2 operations that use Ignition snippets, they should be created using Ignition spec v3. However, the Machine Config Operator (MCO) still supports Ignition spec v2.

1.2.1.4. Additional steps to add nodes to existing clusters

Because of the changes to the Ignition specification, if you want to add more nodes to your OpenShift Container Platform cluster by using the latest version of RHCOS boot media, you might need to manually modify your Ignition configuration as part of the process.

1.2.1.5. Extensions now supported for RHCOS and MCO

RHCOS and the MCO now support the following extensions to the default RHCOS installation.

- **kernel-devel**
- **usbguard**

1.2.1.6. 4Kn Disks now supported

RHCOS now supports installing to disks that use 4K sector sizes.

1.2.1.7. /var partitions now supported

RHCOS now supports /**var** being a separate partition, as well as any other subdirectory of /**var**.

1.2.1.8. Static IP configuration for vSphere using OVA

You can now override default Dynamic Host Configuration Protocol (DHCP) networking in vSphere. This requires setting the static IP configuration and then setting a **guestinfo** property before booting a VM from an OVA in vSphere.

1. Set your static IP:

```
$ export IPCFG="ip=<ip>::<gateway>:<netmask>:<hostname>:<iface>:none
nameserver=srv1 [nameserver=srv2 [nameserver=srv3 [...]]]"
```

Example command

```
$ export IPCFG="ip=192.168.100.101::192.168.100.254:255.255.255.0:::none
nameserver=8.8.8.8"
```

2. Set the **guestinfo.afterburn.initrd.network-kargs** property before booting a VM from an OVA in vSphere:

```
$ govc vm.change -vm "<vm_name>" -e "guestinfo.afterburn.initrd.network-kargs=${IPCFG}"
```

This lowers the barrier for automatic Red Hat Enterprise Linux CoreOS (RHCOS) deployment in environments without DHCP. This enhancement allows for higher-level automation to provision an RHCOS OVA in environments with static networking.

For more information, see BZ1785122.

1.2.2. Installation and upgrade

1.2.2.1. Installing a cluster into an AWS GovCloud region

You can now install a cluster on Amazon Web Services (AWS) into a GovCloud region. AWS GovCloud is designed for US government agencies, contractors, educational institutions, and other US customers that must run sensitive workloads.

Because GovCloud regions do not have RHCOS AMIs published by Red Hat, you must upload a custom AMI that belongs to that region.

For more information, see Installing a cluster on AWS into a government region .

1.2.2.2. Defining custom AWS API endpoints

You can now define a **serviceEndpoints** field in the **install-config.yaml** file, which lets you specify a list of custom endpoints to override the default service endpoints on AWS.

1.2.2.3. Installing a cluster into a Microsoft Azure Government region

You can now install a cluster on Azure into a Microsoft Azure Government (MAG) region. Microsoft Azure Government (MAG) is designed for US government agencies and their partners that must run sensitive workloads.

For more information, see Installing a cluster on Azure into a government region .

1.2.2.4. User-defined outbound routing for clusters running on Azure

You can now choose your own outbound routing for a cluster running on Azure to connect to the internet. This allows you to skip the creation of public IP addresses and public load balancers.

For more information, see User-defined outbound routing.

1.2.2.5. Installing a cluster to vSphere version 7.0

You can now deploy a cluster to VMware vSphere version 7.0. See VMware vSphere infrastructure requirements for more information.

1.2.2.6. Installing a cluster on bare metal using installer-provisioned infrastructure

OpenShift Container Platform 4.6 introduces support for installing a cluster on bare metal using installer-provisioned infrastructure.

For more information, see Installing a cluster on bare metal

1.2.2.7. Handling credential requests for cloud API access on AWS, Azure, and GCP

There is now a new **credentialsMode** field in the **install-config.yaml** file that defines how `CredentialsRequest`s are handled for OpenShift Container Platform components requiring cloud API access on AWS, Azure, and GCP. There are three new modes that can be configured:

- Mint
- Passthrough
- Manual

IMPORTANT

Azure and GCP do not support the Manual mode configuration by using the **install-config.yaml** file due to a known issue found in BZ#1884691.

If the **credentialsMode** field is set to any of the three modes, the installation program does not check the credential for proper permissions prior to installing OpenShift Container Platform. This is useful for when the supplied user credentials cannot be properly validated due to limitations in the cloud policy simulator.

For more information on these modes, see Cloud Credential Operator.

1.2.2.8. Specifying disk type and size for control plane and compute nodes

You can now configure the disk type and size on control plane and compute nodes for clusters running on Azure and GCP. This can be specified in the **install-config.yaml** file with the following fields:

- **osDisk.diskSizeGB**

- **osDisk.diskType**

For example:

```
...
compute:
...
  platform:
  - osDisk:
      diskSizeGB: 120
      diskType: pd-standard
  replicas: 3
controlPlane:
...
  platform:
  - osDisk:
      diskSizeGB: 120
      diskType: pd-ssd
...
```

1.2.2.9. Minimum disk size for control plane nodes has increased for Azure installations

The minimum disk size requirement for control plane nodes for Azure installations has increased from 512 GB to 1024 GB.

1.2.2.10. Latest version of Operators required before cluster upgrade

Starting in OpenShift Container Platform 4.6, the Red Hat-provided default catalogs used by Operator Lifecycle Manager (OLM) and OperatorHub are now shipped as index images specific to the minor version of OpenShift Container Platform. Cluster administrators must ensure all Operators previously installed through OLM are updated to their latest versions in their latest channels before upgrading to OpenShift Container Platform 4.6.

See Default Operator catalogs now shipped per cluster version for more details and important Operator upgrade prerequisites.

1.2.2.11. Deployment without a provisioning network

OpenShift Container Platform now supports deployment without a provisioning network and for RedFish Virtual Media.

See Setting up the environment for an OpenShift installation for more information.

1.2.2.12. Deployment now supports root device hints

Deployment now supports root device hints.

1.2.2.13. Installer improvements

Deployment now performs introspection on nodes to ensure that nodes meet installation requirements instead of generating errors if they do not.

1.2.2.14. RHOSP availability zones selection at installation

You can now select Red Hat OpenStack Platform (RHOSP) Compute (Nova) availability zones while installing a cluster on RHOSP.

For more information, see the OpenShift Container Platform on RHOSP installation documentation.

1.2.2.15. Floating IP addresses are no longer required for installation on RHOSP

You no longer need floating IP addresses to complete a OpenShift Container Platform installation on RHOSP.

For more information, see the OpenShift Container Platform on RHOSP installation documentation.

1.2.2.16. Select disk for RHCOS installation

Previously, when you used infrastructure that the installation program creates to deploy a bare metal cluster, you could not specify which disk to the deploy RHCOS on. Now, you can select the disk to install RHCOS on, and **rootDeviceHints** provide guidance about selecting the target disk. (BZ#1805237)

1.2.2.17. AWS clusters now default to use M5 instances

M5 instances are now preferred for IPI and UPI installations on AWS. Thus, new clusters that are deployed on AWS now use M5 instances by default. If an M5 instance is not available, the installer uses an M4 instance. (BZ#1710981)

1.2.2.18. IBM Z and LinuxONE

With this release, IBM Z and LinuxONE is now compatible with OpenShift Container Platform 4.6. See Installing a cluster on IBM Z and LinuxONE for installation instructions.

Restrictions
Note the following restrictions for OpenShift Container Platform on IBM Z and LinuxONE:

- OpenShift Container Platform for IBM Z does not include the following Technology Preview features:
  - Log forwarding
  - Precision Time Protocol (PTP) hardware
  - CSI volume snapshots
  - OpenShift Pipelines
- The following OpenShift Container Platform features are unsupported:
  - OpenShift Container Platform Virtualization
  - Red Hat OpenShift Service Mesh
  - CodeReady Containers (CRC)
  - OpenShift Container Platform Metering
  - Multus CNI plug-in
  - FIPS cryptography

    - Encrypting data stored in etcd
    - Automatic repair of damaged machines with machine health checking
    - Tang mode disk encryption during OpenShift Container Platform deployment
    - OpenShift Container Platform Serverless
    - Helm command-line interface (CLI) tool
    - Controlling overcommit and managing container density on nodes
    - etcd cluster Operator
    - CSI volume cloning
    - NVMe
    - Persistent storage using Fibre Channel
- Worker nodes must run Red Hat Enterprise Linux CoreOS (RHCOS).
- Persistent shared storage must be provisioned using NFS.
- Persistent non-shared storage must be provisioned using local storage, like iSCSI, FC, or LSO with DASD/FCP.
- These features are available only for OpenShift Container Platform on IBM Z for 4.6:
    - HyperPAV enabled on IBM System Z /LinuxONE for the virtual machines for FICON attached ECKD storage

Supported features
With this release, the following features are supported on IBM Z and LinuxONE:

- Persistent storage using iSCSI
- Persistent storage using local volumes (Local Storage Operator)
- OpenShift Do (odo)

1.2.2.19. IBM Power Systems

With this release, IBM Power Systems are now compatible with OpenShift Container Platform 4.6. See Installing a cluster on IBM Power Systems or Installing a cluster on IBM Power Systems in a restricted network.

Restrictions
Note the following restrictions for OpenShift Container Platform on IBM Power Systems:

- OpenShift Container Platform for IBM Power Systems does not include the following Technology Preview features:
    - OpenShift Virtualization
    - OpenShift Serverless (knative, FaaS integrations)
- The following OpenShift Container Platform features are unsupported:

    - Red Hat OpenShift Service Mesh (istio, jaeger, kiali)
    - CodeReady Workspaces
    - CodeReady Containers (CRC)
    - OpenShift Pipelines based on Tekton
    - OpenShift Container Platform Metering
    - Multus Plugins (SR-IOV, IPVAN, Bridge with VLAN, Static IPAM)
    - SR-IOV CNI plug-in
    - Red Hat Single Sign-On
    - OpenShift Metering (Presto, Hive)
- Worker nodes must run Red Hat Enterprise Linux CoreOS (RHCOS).
- Persistent storage must be of the **Filesystem** mode using local volumes, Network File System (NFS), OpenStack Cinder, or Container Storage Interface (CSI).
- Networking must use either DHCP or static addressing with Red Hat Openshift SDN.
- AdoptOpenJDK with OpenJ9
- Installer-provisioned infrastructure
- Device Manager for NVIDIA GPUs
- Special Resources Operator
- OpenShift Ansible Service Broker Operator (deprecated)
- dotNET on RHEL

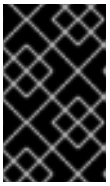
Supported Features

- Currently, four Operators are supported:
    - Cluster-Logging-Operator
    - Cluster-NFD-Operator
    - Elastic Search-Operator
    - Local Storage Operator
- User-provisioned infrastructure deployment scenario on bare-metal
- OpenShift Cluster Monitoring
- Node Tuning Operator
- OpenShift Jenkins
- OpenShift Logging

- OpenShift Do (odo)
- Machine Configuration Operator, which is used in installations with installer-provisioned infrastructure
- Node Feature Discovery Operator
- OpenShift Container Platform core (CVO Operators)
- Installation program for clusters that use user-provisioned infrastructure
- OVS
- RHEL8 Based container support
- RHEL CoreOS
- Ansible Engine
- Red Hat Software Collections
- HostPath
- iSCSI
- 4k Disk support

1.2.2.20. Enhancements to Red Hat Virtualization (RHV) full-stack installer

- You can use Container Storage Interface (CSI) Driver Operators to dynamically provision storage from RHV storage domains to an OpenShift Container Platform cluster.
- You can use auto-scaling additional of RHV virtual machine worker nodes to improve control of workloads and resources.
- You can perform disconnected or restricted installations by using a local mirror. This capability is beneficial for financial, public sector, and secure environments.
- You can install OpenShift Container Platform on RHV with user-provisioned infrastructure , such as an external load balancer. This process uses a series of Ansible playbooks to enable a more flexible installation.
- OpenShift Container Platform version 4.6 requires RHV version 4.4.2 or later.



IMPORTANT

If you are running OpenShift Container Platform version 4.5 on RHV version 4.3, upgrade RHV to version 4.4.2 or later before updating OpenShift Container Platform to version 4.6.

1.2.2.21. Improvements to the remediation of failed nodes for bare metal deployments using installer-provisioned infrastructure

Reboot-based remediation of failed control plane nodes is now possible. The labels and annotations of those nodes are preserved when using the reboot-based remediation method.

1.2.3. Security and compliance

1.2.3.1. Compliance Operator

The Compliance Operator is now available. This feature allows the use of OpenSCAP tools to check that a deployment complies with security standards and provides remediation options. See Understanding the Compliance Operator for more information.

1.2.3.2. Configure OAuth token inactivity timeout

You can now configure OAuth tokens to expire after a certain amount of time that they have been inactive. By default, there is no token inactivity timeout set. You can configure the timeout for the internal OAuth server and for OAuth clients.

See Configuring token inactivity timeout for the internal OAuth server and Configuring token inactivity timeout for an OAuth client for more information.

1.2.3.3. Secure OAuth token storage format

OAuth access token and OAuth authorize token object names are now stored as non-sensitive object names.

Previously, secret information was used as the OAuth access token and OAuth authorize token object names. When etcd is encrypted, only the value is encrypted, so this sensitive information was not encrypted.

IMPORTANT

If you are upgrading your cluster to OpenShift Container Platform 4.6, old tokens from OpenShift Container Platform 4.5 will still have the secret information exposed in the object name. By default, the expiration for tokens is 24 hours, but this setting can be changed by administrators. Sensitive data can still be exposed until all old tokens have either expired or have been deleted by an administrator.

1.2.3.4. File Integrity Operator is now available

The File Integrity Operator, an OpenShift Container Platform Operator that continually runs file integrity checks on the cluster nodes, is now available. It deploys a daemon set that initializes and runs privileged advanced intrusion detection environment (AIDE) containers on each node, providing a status object with a log of files that are modified during the initial run of the daemon set Pods.

1.2.3.5. Cluster scripts updated for cluster restore failures

The **cluster-backup.sh** and **cluster-restore.sh** scripts were updated to provide better feedback, so that users can better understand why a restore has failed.

1.2.4. Machine API

1.2.4.1. Support for multiple block device mappings

The Machine API now supports multiple block device mappings for machines running on AWS. If more than one block device is given, you can now store logs, data in empty directory pods, and docker images in block devices that are separate from the root device on a machine.

1.2.4.2. Defaults and validation for the Machine API providerSpec

Defaults and validation are now enabled on a particular cloud provider API before input from the **providerSpec** is persisted to etcd. Validation is run against machines and MachineSets when they are created. Feedback is returned when the configuration is known to prevent machines from being created by the cloud provider. For example, a MachineSet is rejected if location information is required but is not provided.

1.2.4.3. MachineSets running on Azure support Spot VMs

MachineSets running on Azure now support Spot VMs. You can create a MachineSet that deploys machines as Spot VMs to save on costs compared to standard VM prices. For more information, see MachineSets that deploy machines as Spot VMs .

Configure Spot VMs by adding **spotVMOptions** under the **providerSpec** field in the MachineSet YAML file:

```
providerSpec:
  value:
    spotVMOptions: {}
```

1.2.4.4. MachineSets running on GCP support preemptible VM instances

MachineSets running on GCP now support preemptible VM instances. You can create a MachineSet that deploys machines as preemptible VM instances to save on costs compared to normal instance prices. For more information, see MachineSets that deploy machines as preemptible VM instances .

Configure preemptible VM instances by adding **preemptible** under the **providerSpec** field in the MachineSet YAML file:

```
providerSpec:
  value:
    preemptible: true
```

1.2.5. Web console

1.2.5.1. Improved upgrade experience in the web console

- Administrators are now better informed about the differences between the upgrade channels by helpful text and links in the web console.
- A link to the list of bug fixes and enhancements is now included for each minor or patch release.
- There is now a visual representation of the different upgrade paths.
- Alerts now inform administrators when new patch releases, new minor releases, and news channels become available.

1.2.5.2. Improved Operator installation workflow with OperatorHub

When administrators install Operators with OperatorHub, they now get immediate feedback to ensure that the Operator is installing properly.

1.2.5.3. Improved operand details view

You can now see the schema grouping of **specDescriptor** fields and the status of your Operands on the operand's details view, so that you can easily see the status and configure the **spec** of the operand instance.

1.2.5.4. View related objects for cluster Operators

Previously, when viewing a cluster Operator, it was not clear what resources the Operator was associated with. When troubleshooting a cluster Operator, it could be challenging to locate the logs for all the resources that the Operator managed, which might be needed for troubleshooting. Now, with OpenShift Container Platform 4.6, you can expose a list of related objects of a cluster Operator and easily review one of the related objects' details or YAML code for troubleshooting.

1.2.5.5. Warning messages when editing managed resources

Some resources are managed, such an Operator managed by a deployment, route, service, or ConfigMap. Users are discouraged from editing these resources. Instead, users should edit the custom resources for the Operator and its operand, and expect the Operator to update its related resources. With this update:

- A **Managed by** label now appears below the resource name with a clickable resource link for the managing resource.
- When the resource is modified or deleted, a message appears warning the user that their changes might be reverted.

1.2.5.6. The k8sResourcePrefix specDescriptor supports CRD instance

Operator authors, maintainers, and providers can now specify the **k8sResourcePrefix** specDescriptor with **Group/Version/Kind** for assigning a CRD resource type besides Kubernetes core API.

For more information, see OLM Descriptor Reference.

1.2.5.7. Column management on resources page

A **Manage columns** icon  is now added to some resources pages, for example the Pods page. When you click on the icon, default column names are listed with check boxes on the left side of the modal and additional column names are listed on the right. Deselecting a check box will remove that column from the table view. Selecting a check box will add that column to the table view. A maximum combination of nine columns from both sides of the modal are available for display at one time. Clicking **Save** will save the changes that you make. Clicking **Restore Default Columns** will restore the default settings of the columns.

1.2.5.8. Developer Perspective

- Based on user access roles or privileges, the user is now directed to either the **Administrator** or **Developer** perspective.
- An interactive getting started tour of the functionalities in the **Developer** perspective is now provided when a user logs in.
- The **List** view and **Topology** view now provide the same information so that the user can choose the best view based on the application size and number of components.

- Support for all workload types is now provided in the **Topology** and the **List** view to get a better idea of the compute resources being used.
- You can now select the Helm chart version and the application version while installing the chart from the **Developer catalog**. You can also switch between the form and YAML editor while preserving the values that have been entered.
- The Knative eventing workflow was enhanced:
  - Support for Knative Eventing Channels to build a reliable event delivery mechanism has been added.
  - You can now create a subscription for channels and triggers with the associated filters for brokers, and select a Knative service as the subscriber.
  - When creating Event Sources, you can now specify the sink as any Knative resource such as, Knative service, channel, or broker, from that namespace; or a URI.
  - You can now visualize the relationship for the event source subscribed by a Knative service through a channel, subscription, broker, or trigger. Details of the event source relationship can be seen in the side panel as well.
  - Ability to filter for a specific event type has been provided.
- Usability enhancements like adding runtime labels to see the appropriate runtime icons and tooltips have been added.
- You can now add, edit, and delete a basic horizontal pod autoscaling (HPA) from a workload and create an HPA and specify the workload to assign it to.
- If OpenShift Service Mesh is enabled on the cluster and the given namespace is enabled, you can now click the Kiali link on the **Topology** view to navigate to the configured Kiali dashboard into the right namespace.
- The **Monitoring** view now provides the ability to filter resource specific metrics in the **Monitoring** dashboard. You can also view the firing alerts, silence them, and see the alert rules configured for your project.

1.2.6. Scale

1.2.6.1. Cluster maximums

Updated guidance around cluster maximums for OpenShift Container Platform 4.6 is now available.

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

1.2.6.2. Real-time profile added to the Node Tuning Operator

Partial Tuned real-time profile support became available in OpenShift Container Platform 4.4. Now, the real-time profiles are fully compatible with what the real-time profiles do in Tuned on Red Hat Enterprise Linux (RHEL).

1.2.6.3. The Performance Addon Operator is now fully supported

The Performance Addon Operator helps the administrator with tuning worker nodes for low latency and real-time workloads. It takes a high-level tuning intent in the form of a **PerformanceProfile** custom

resource and translates it into all the actions necessary to configure the Linux kernel, operating system, huge pages, and kubelet for low latency purposes.

In addition to the features provided in the previous pre-releases, this version includes the following:

- CPU load balancing can be enabled per Pod.
- Multiple huge page sizes can be specified at the same time.
- Improvements to supportability, such as integration gathering and better status reporting.
- A method for in-field emergency configuration overrides was devised and documented.

1.2.7. Developer experience

1.2.7.1. oc set probe command has been extended

The **oc set probe** command was extended to support setting startup probes.

1.2.7.2. oc adm upgrade command now mentions upgradeable condition

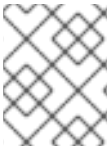
The **oc adm upgrade** command now mentions any **Upgradeable=False** conditions, so that administrators are aware that a particular update might be rejected due to an **Upgradeable=False** condition.

1.2.8. Networking

1.2.8.1. OVN-Kubernetes cluster networking provider GA

The OVN-Kubernetes cluster networking provider is now GA. The networking provider is implemented as a Kubernetes Container Network Interface (CNI) plug-in. For more information, including details on feature parity with OpenShift SDN, refer to About the OVN-Kubernetes Container Network Interface (CNI) network provider.

For this release, OpenShift SDN remains the default cluster networking provider.



NOTE

OVN-Kubernetes is not supported on Red Hat Enterprise Linux (RHEL) 7.8 at OpenShift Container Platform 4.6 GA.

1.2.8.2. Expand node service port range

The node service port range is expandable beyond the default range of **30000-32767**. You can use this expanded range in your **Service** objects. For more information, refer to Configuring the node port service range.

1.2.8.3. SR-IOV Network Operator InfiniBand device support

The Single Root I/O Virtualization (SR-IOV) Network Operator now supports InfiniBand (IB) network devices. For more information on configuring an IB network device for your cluster, refer to Configuring an SR-IOV InfiniBand network attachment.

1.2.8.4. DHCP range increased for provisioning network

To better support large deployments, the default DHCP range for the provisioning network is increased to include the remainder of the subnet in this release. Users who would like to use less of the subnet for DHCP can still configure it to their needs. (BZ#1841135)

1.2.8.5. Pod network connectivity checks

Operators can now configure **PodNetworkConnectivityCheck** resources to check each network connection from the Pods that are managed by the Operator. This allows you to more easily identify and troubleshoot issues with important network connections in your cluster.

This resource keeps track of the latest reachable condition, the last 10 successes, the last 10 failures, and details about detected outages. The results are also logged and events are created when outages are detected and resolved.

By default, the following network connections are checked:

- Between the Kubernetes API server and:
  - the OpenShift API server service
  - each OpenShift API server endpoint
  - each etcd endpoint
  - the internal API load balancer
  - the external API load balancer
- Between the OpenShift API server and:
  - the Kubernetes API server service
  - each Kubernetes API server endpoint
  - each etcd endpoint
  - the internal API load balancer
  - the external API load balancer

1.2.8.6. Secondary device metrics can be associated with network attachments

Secondary devices, or interfaces, are used for different purposes. It is important to have a way to classify them so that you can aggregate the metrics for secondary devices with the same classification.

The kubelet is already publishing a set of network-observable related metrics. The labels in these metrics contain, among others:

- Pod name
- Pod namespace
- Interface name, such as eth0

This works well until new interfaces are added to the Pod, for example via Multus, as it will not be clear

what the interface names refer to. The interface label refers to the interface name, but it is not clear what that interface is meant for. In case of many different interfaces, it would be impossible to understand what network the metrics we are monitoring refer to. This is addressed by introducing the new **pod_network_name_info** metric, which can be used to build queries containing both the values exposed by the kubelet and the name of the network attachment definition the metrics relates to, which identifies the type of network.

See Associating secondary interfaces metrics to network attachments for more information.

1.2.8.7. CNF tests can be run in discovery mode

There is an optional mode where the Cloud-native Network Functions (CNF) tests try to look for configurations on the cluster instead of applying the new ones. The CNF tests image is a containerized version of the CNF conformance test suite. It is intended to be run against a CNF-enabled OpenShift Container Platform cluster where all the components required for running CNF workloads are installed.

The tests must perform an environment configuration every time they are executed. This involves items such as creating SR-IOV Node Policies, Performance Profiles, or PtpProfiles. Allowing the tests to configure an already configured cluster might affect the functionality of the cluster. Also, changes to configuration items such as SR-IOV Node Policy might result in the environment being temporarily unavailable until the configuration change is processed.

Discovery mode validates the functionality of a cluster without altering its configuration. Existing environment configurations are used for the tests. The tests attempt to find the configuration items needed and use those items to execute the tests. If resources needed to run a specific test are not found, the test is skipped, providing an appropriate message to the user. After the tests are finished, no cleanup of the pre-configured configuration items is done, and the test environment can be used immediately for another test run.

1.2.8.8. HAProxy version upgrade

Ingress in OpenShift Container Platform 4.6 now uses HAProxy version 2.0.16.

1.2.8.9. Control over X-Forwarded headers

Control over X-Forwarded headers is now possible by setting the **forwardedHeaderPolicy** parameter.

Application and configuration of X-forwarded headers on a per-route basis is now supported with the introduction of the **haproxy.router.openshift.io/set-forwarded-headers** route annotations.

See Using X-Forwarded headers for more information.

1.2.8.10. Modify route path

Modifying route paths for incoming requests is now supported with the **haproxy.router.openshift.io/rewrite-target** variable.

See Route configuration for more information.

1.2.8.11. Ingress termination policies

Termination policies can now be defined by using the **route.openshift.io/termination** annotation for Ingress objects.

See Creating a route through an Ingress object for more information.

1.2.8.12. Ingress Controller Network Load Balancer for AWS

Configuration of an Ingress Controller Network Load Balancer (NLB) for new and existing AWS clusters is now supported.

See Configuring ingress cluster traffic on AWS using a Network Load Balancer for more information.

1.2.8.13. Ingress Operator endpoint configuration for AWS Route53

AWS Route53 endpoint configuration is now supported on the Ingress Operator.

See Ingress Operator endpoint configuration for AWS Route53 for more information.

1.2.9. Storage

1.2.9.1. CSI drivers now managed by the Cluster Storage Operator

The Container Storage Interface (CSI) Driver Operators and drivers for AWS Elastic Block Store (EBS), Red Hat Virtualization (oVirt), and OpenStack Manila shared file system service are now managed by the Cluster Storage Operator in OpenShift Container Platform.

For AWS EBS and oVirt, this feature installs the CSI Driver Operator and driver in the **openshift-cluster-csi-drivers** namespace by default. For Manila, the CSI Driver Operator is installed in **openshift-cluster-csi-drivers** and the driver is installed in the **openshift-manila-csi-driver** namespace.

IMPORTANT

If you installed a CSI Driver Operator and driver on an OpenShift Container Platform 4.5 cluster:

- The AWS EBS CSI Driver Operator and driver must be uninstalled before you update to a newer version of OpenShift Container Platform.
- The OpenStack Manila CSI Driver Operator is no longer available in Operator Lifecycle Manager (OLM). It has been automatically converted by the Cluster Version Operator.

1.2.9.2. Automatic device discovery and provisioning with the Local Storage Operator (Technology Preview)

The Local Storage Operator now has the ability to:

- Automatically discover a list of available disks in a cluster. You can select a list of nodes, or all nodes, for auto-discovery to be continuously applied to.
- Automatically provision local persistent volumes from attached devices. Appropriate devices are filtered and persistent volumes are provisioned based on the filtered devices.

For more information, see Automating discovery and provisioning for local storage devices .

1.2.10. Registry

1.2.10.1. Image pruner tolerates invalid images

The image pruner now tolerates invalid image references by default on new installations of OpenShift Container Platform, which allows pruning to continue even if it finds invalid images.

1.2.10.2. Change the image pruner's log level

Cluster administrators can now configure **logLevel** in the Pruning Custom Resource to debug logs.

1.2.10.3. Image registry supports Azure Government

The image registry can now be set up and configured for Azure Government.

See Configuring registry storage for Azure Government for more information.

1.2.10.4. Change the Image Registry Operator's log level

Cluster administrators can now configure **logLevel** in the Image Registry Operator to debug logs.

The supported values for **logLevel** are:

- **Normal**
- **Debug**
- **Trace**
- **TraceAll**

Example Image Registry Operator YAML file

```
spec:
logLevel: Normal
operatorLogLevel: Normal
```

1.2.10.5. Change the Image Registry Operator's spec.storage.managementState

The Image Registry Operator now sets the **spec.storage.managementState** parameter to **Managed** on new installations or upgrades of clusters using installer-provisioned infrastructure on AWS or Azure.

- **Managed**: Determines that the Image Registry Operator manages underlying storage. If the Image Registry Operator's **managementState** is set to **Removed**, then the storage is deleted.
    - If the **managementState** is set to **Managed**, the Image Registry Operator attempts to apply some default configuration on the underlying storage unit. For example, if set to **Managed**, the Operator tries to enable encryption on the S3 bucket before making it available to the registry. If you do not want the default settings to be applied on the storage you are providing, make sure the **managementState** is set to **Unmanaged**.
- **Unmanaged**: Determines that the Image Registry Operator ignores the storage settings. If the Image Registry Operator's **managementState** is set to **Removed**, then the storage is not deleted. If you provided an underlying storage unit configuration, such as a bucket or container name, and the **spec.storage.managementState** is not yet set to any value, then the Image Registry Operator configures it to **Unmanaged**.

1.2.11. Operator lifecycle

1.2.11.1. Operator version dependency

Operator developers can now ensure their Operators include dependencies on specific versions of other Operators by using the **olm.package** type in the **dependencies.yaml** file.

See Operator Lifecycle Manager dependency resolution for more information.

1.2.11.2. Additional objects supported in Operator bundles

The Operator Bundle Format now supports the following additional Kubernetes objects:

- PodDisruptionBudget
- PriorityClass
- VerticalPodAutoScaler

See Operator Framework packaging formats for more information.

1.2.11.3. Selective bundle image mirroring with opm

Operator administrators can now to select which bundle images to mirror by using the **opm index prune** command.

See Pruning an index image for more information.

1.2.11.4. Conversion webhook support for global Operators

Operator developers can now use conversion webhooks for Operators that target all namespaces, also known as global Operators.

See Defining webhooks for more information.

1.2.11.5. Operator API now supported

The Operator API introduced in OpenShift Container Platform 4.5 as a Technology Preview feature is now supported and enabled by default. Installing Operators using Operator Lifecycle Manager (OLM) has required cluster administrators to be aware of multiple APIs, including CatalogSources, Subscriptions, ClusterServiceVersions, and InstallPlans. This single Operator API resource is a first step towards a more simplified experience discovering and managing the lifecycle of Operators in a OpenShift Container Platform cluster.

Relevant resources are now automatically labeled accordingly for the new Operator API for any Operators where the CSV is installed using a Subscription. Cluster administrators can use the CLI with this single API to interact with installed Operators. For example:

```
$ oc get operators
```

```
$ oc describe operator <operator_name>
```

1.2.11.5.1. Removing Technology Preview Operator API before cluster upgrade

If you enabled the Technology Preview feature version of the Operator API in OpenShift Container Platform 4.5, you must disable it before upgrading to OpenShift Container Platform 4.6. Failure to do so blocks your cluster upgrade, because the feature required a Cluster Version Operator (CVO) override .

Prerequisites

- OpenShift Container Platform 4.5 cluster with Technology Preview Operator API enabled

Procedure

1. Because Operator API labels are applied to relevant resources automatically in OpenShift Container Platform 4.6, you must remove any **operators.coreos.com/<name>** labels you previously applied manually.

   a. You can check which resources are currently labeled for your Operator by running the following command and reviewing the **status.components.refs** section:

   ```
   $ oc describe operator <operator_name>
   ```

   For example:

   ```
   $ oc describe operator etcd-test
   ```

   #### Example output

   ```
   ...
   Status:
     Components:
       Label Selector:
         Match Expressions:
           Key:      operators.coreos.com/etcd-test
           Operator:  Exists
       Refs:
         API Version:  apiextensions.k8s.io/v1
         Conditions:
           Last Transition Time:  2020-07-02T05:50:40Z
           Message:               no conflicts found
           Reason:                NoConflicts
           Status:                True
           Type:                  NamesAccepted
           Last Transition Time:  2020-07-02T05:50:41Z
           Message:               the initial names have been accepted
           Reason:                InitialNamesAccepted
           Status:                True
           Type:                  Established
         Kind:                    CustomResourceDefinition (1)
         Name:                    etcdclusters.etcd.database.coreos.com (2)
   ...
   ```

   (1) (2) Resource type.

   Resource name.

   b. Remove the labels from all relevant resources. For example:

```
$ oc label sub etcd operators.coreos.com/etcd-test- -n test-project
$ oc label ip install-6c5mr operators.coreos.com/etcd-test- -n test-project
$ oc label csv etcdoperator.v0.9.4 operators.coreos.com/etcd-test- -n test-project
$ oc label crd etcdclusters.etcd.database.coreos.com operators.coreos.com/etcd-test-
$ oc label crd etcdbackups.etcd.database.coreos.com operators.coreos.com/etcd-test-
$ oc label crd etcdrestores.etcd.database.coreos.com operators.coreos.com/etcd-test-
```

2. Delete the Operator custom resource definition (CRD):

```
$ oc delete crd operators.operators.coreos.com
```

3. Remove the **OperatorLifecycleManagerV2=true** feature gate from the OLM Operator.

   a. Edit the Deployment for the OLM Operator:

```
$ oc -n openshift-operator-lifecycle-manager \
    edit deployment olm-operator
```

   b. Remove the following flags from the **args** section in the Deployment:

```
...
    spec:
      containers:
      - args:
...
        - --feature-gates 1
        - OperatorLifecycleManagerV2=true 2
```

   1 2 Remove these flags.

   c. Save your changes.

4. Re-enable CVO management of OLM:

```
$ oc patch clusterversion version \
   --type=merge -p \
   '{
     "spec":{
       "overrides":[
         {
           "kind":"Deployment",
           "name":"olm-operator",
           "namespace":"openshift-operator-lifecycle-manager",
           "unmanaged":false,
           "group":"apps/v1"
         }
       ]
     }
   }'
```

5. Verify that the Operator resource is no longer available:

```
$ oc get operators
```

Example output

```
error: the server doesn't have a resource type "operators"
```

Your upgrade to OpenShift Container Platform 4.6 should now no longer be blocked by this feature.

1.2.11.6. Node Maintenance Operator now validates maintenance requests

The Node Maintenance Operator now validates maintenance requests for master nodes, preventing master (etcd) quorum violation. As a result, master nodes can only be set into maintenance if the **etcd-quorum-guard** pod disruption budget (PDB) allows it. (BZ#1826914)

1.2.11.7. Set log levels separately for Image Registry Operator and operand

Users can now set log levels separately for the Image Registry Operator and operand. (BZ#1808118)

1.2.12. Builds

1.2.12.1. Builds support Git clones behind an HTTPS proxy

Builds now support Git clones behind an HTTPS proxy.

1.2.13. Images

1.2.13.1. Support for Cloud Credential Operator modes

In addition to the existing default mode of operation, the Cloud Credential Operator (CCO) can now be explicitly configured to operate in the following modes: **Mint**, **Passthrough**, and **Manual**. This feature provides transparency and flexibility in how the CCO uses cloud credentials to process **CredentialRequests** in the cluster for installation and other tasks.

1.2.13.2. Cluster Samples Operator on Power and Z

Imagestreams and templates for Power and Z architectures are now available and installed by the Cluster Samples Operator by default.

1.2.13.3. Cluster Samples Operator Alerts

If samples do not import, the Cluster Samples Operator now notifies you with an alert instead of moving to a degraded status.

See Using Cluster Samples Operator imagestreams with alternate or mirrored registries for more information.

1.2.14. Metering

1.2.14.1. Configuring a retention period of metering Reports

You can now set a retention period on a metering Report. The metering Report custom resource has a new **expiration** field. If the **expiration** duration value is set on a Report, and no other Reports or ReportQueries depend on the expiring Report, the Metering Operator removes the Report from your cluster at the end of its retention period. For more information, see metering Reports expiration.

1.2.15. Nodes

1.2.15.1. Configure the node audit log policy

You can now control the amount of information that is logged to the node audit logs by choosing the audit log policy profile to use.

See Configuring the node audit log policy for more information.

1.2.15.2. Configure pod topology spread constraints

You can now configure pod topology spread constraints for more fine-grained control the placement of your pods across nodes, zones, regions, or other user-defined topology domains. This can help you improve high availability and resource utilization.

See Controlling pod placement by using pod topology spread constraints for more information.

1.2.15.3. New descheduler strategy is available (Technology Preview)

The descheduler now allows you to configure the **PodLifeTime** strategy. This strategy evicts pods after they reach a certain, configurable age.

See Descheduler strategies for more information.

1.2.15.4. Descheduler filtering by namespace and priority (Technology Preview)

You can now configure whether descheduler strategies should consider pods for eviction based on their namespace and priority.

See Filtering pods by namespace and Filtering pods by priority for more information.

1.2.15.5. New parameter for the RemoveDuplicates descheduler strategy (Technology Preview)

The **RemoveDuplicates** strategy now provides an optional parameter, **ExcludeOwnerKinds**, that allows you to specify a list of **Kind** types. If a pod has any of these types listed as an **OwnerRef**, that pod is not considered for eviction.

See Descheduler strategies for more information.

1.2.15.6. Generate ImageContentSourcePolicy scoped to a registry

The **oc adm catalog mirror** command generates an **ImageContentSourcePolicy** (ICSP) that maps the original container image repository to a new location where it will be mirrored, typically inside a disconnected environment. When a new or modified ICSP is applied to a cluster, it is converted to a config file for CRI-O and placed onto each node. The process of placing the config file on a node includes rebooting that node.

This enhancement adds the **--icsp-scope** flag to **oc adm catalog mirror**. Scopes can be registry or repository. By default, the **oc adm catalog mirror** command generates an ICSP where each entry is specific to a repository. For example, it would map **registry.redhat.io/cloud/test-db** to **mirror.internal.customer.com/cloud/test-db**. Widening the mirror to registry scope in the ICSP file minimizes the number of times the cluster must reboot its nodes. Using the same example, **registry.redhat.io** would map to **mirror.internal.customer.com**.

Having a widely scoped ICSP reduces the number of times the ICSP might need to change in the future and, thus, reduces the number of times a cluster must reboot all of its nodes.

1.2.16. Cluster logging

Log Forwarding API is generally available
The Log Forwarding API is now generally available. The Log Forwarding API allows you to send container, infrastructure, and audit logs to specific endpoints within and outside your cluster by configuring a custom resource with the endpoints to forward the logs. The Log Forwarding API now supports forwarding to Kafka brokers and supports syslog RFC 3164 and RFC 5424 including TLS. You can also forward application logs from a specific projects to an endpoint.

With the GA, the Log Forwarding API has a number of changes, including changes to parameter names in the Log Forwarding custom resource (CR). If you used the Log Forwarding Technology Preview, you need to manually make the needed changes to your existing Log Forwarding CR.

Adding labels to log messages
The Log Forwarding API allows you to add free-text labels to log messages that are affixed to outbound log messages. For example, you could label logs by data center or label the logs by type. Labels added to objects are also forwarded with the log message.

New cluster logging dashboards
Two new dashboards have been added to the OpenShift Container Platform web console that display charts with important, low-level metrics for detailed investigation and troubleshooting of your cluster logging and Elasticsearch instances.

The **OpenShift Logging** dashboard contains charts that show details about your Elasticsearch instance at a cluster-level, including cluster resources, garbage collection, shards in the cluster, and Fluentd statistics.

The **Logging/Elasticsearch Nodes** dashboard contains charts that show details about your Elasticsearch instance, many at node-level, including details on indexing, shards, resources, and so forth.

New parameters for tuning Fluentd
New Fluentd parameters allow you to performance-tune your Fluentd log collector. With these parameters, you can change:

- the size of Fluentd chunks and chunk buffer
- the Fluentd chunk flushing behavior
- the Fluentd chunk forwarding retry behavior

These parameters can help you determine the trade-offs between latency and throughput in your cluster logging instance.

1.2.17. Monitoring

1.2.17.1. Monitoring for user-defined projects

In OpenShift Container Platform 4.6, you can enable monitoring for user-defined projects in addition to the default platform monitoring. You can now monitor your own projects in OpenShift Container Platform without the need for an additional monitoring solution. Using this new feature centralizes monitoring for core platform components and user-defined projects.

With this new feature, you can perform the following tasks:

- Enable and configure monitoring for user-defined projects
- Create recording and alerting rules that use metrics from your own pods and services
- Access metrics and information about alerts through a single, multi-tenant interface
- Cross-correlate the metrics for user-defined projects with platform metrics

For more information, see Understanding the monitoring stack.

1.2.17.2. Alerting rule changes

OpenShift Container Platform 4.6 includes the following alerting rule changes:

- The **PrometheusOperatorListErrors** alert is added. The alert provides notification of errors when running list operations on controllers.
- The **PrometheusOperatorWatchErrors** alert is added. The alert provides notification of errors when running watch operations on controllers.
- The **KubeQuotaExceeded** alert is replaced by **KubeQuotaFullyUsed**. Previously, the **KubeQuotaExceeded** alert fired if a resource quota exceeded a 90% threshold. The **KubeQuotaFullyUsed** alert fires if a resource quota is fully used.
- etcd alerts now support the addition of custom labels for metrics.
- The **KubeAPILatencyHigh** and **KubeAPIErrorsHigh** alerts are replaced by the **KubeAPIErrorBudgetBurn** alert. **KubeAPIErrorBudgetBurn** combines API error and latency alerts and fires only when the conditions are severe enough.
- The readiness and liveness probe metrics exposed by the kubelet are now scraped. This provides historical liveness and readiness data for containers, which can be helpful when troubleshooting container issues.
- The alerting rules for the Thanos Ruler are updated so that alerts are paged if recording rules and alerting rules are not correctly evaluated. This update ensures that critical alerts are not lost when rule and alert evaluation in the Thanos Ruler is not completed.
- The **KubeStatefulSetUpdateNotRolledOut** alert is updated so that it does not fire when a stateful set is being deployed.
- The **KubeDaemonSetRolloutStuck** alert is updated to account for daemon set roll out progress.
- The severity of cause-based alerts are adjusted from **critical** to **warning**.

NOTE

Red Hat does not guarantee backward compatibility for metrics, recording rules, or alerting rules.

1.2.17.3. Prometheus rule validation

OpenShift Container Platform 4.6 introduces validation of Prometheus rules through a webhook that calls the validating admission plug-in. With this enhancement, PrometheusRule custom resources in all projects are checked against the Prometheus Operator rule validation API.

1.2.17.4. Metrics and alerting rules added for the Thanos Querier

The Thanos Querier aggregates and optionally deduplicates core OpenShift Container Platform metrics and metrics for user-defined projects under a single, multi-tenant interface. In OpenShift Container Platform 4.6, a service monitor and alerting rules are now deployed for the Thanos Querier, which enables monitoring of the Thanos Querier by the monitoring stack.

1.2.17.5. Virtual Machine **Pending Changes** alert updates

The Virtual Machine **Pending Changes** alert is now more informative. (BZ#1862801)

1.2.18. Insights Operator

1.2.18.1. Insights Operator data collection enhancements

In OpenShift Container Platform 4.6, the Insights Operator collects the following additional information:

- Pod disruption budgets
- The volume snapshot custom resource definition
- The latest pod logs from pods that are not healthy
- Data about running Red Hat images, including the number of containers using an image and the age of the corresponding pods
- JSON dumps for pods that have crash looping containers
- The MachineSet resource configuration
- The anonymized HostSubnet resource configuration
- The MachineConfigPools configuration
- The InstallPlans for **default** and **openshift-*** projects and their count
- ServiceAccounts statistics from Openshift namespaces

Additionally, with this release the Insights Operator collects information about all cluster nodes, while previous versions only collected information about unhealthy nodes.

With this additional information, Red Hat can provide improved remediation steps in Red Hat OpenShift Cluster Manager.

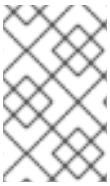
1.3. NOTABLE TECHNICAL CHANGES

OpenShift Container Platform 4.6 introduces the following notable technical changes.

Default Operator catalogs now shipped per cluster version
Starting in OpenShift Container Platform 4.6, the Red Hat-provided default catalogs used by Operator Lifecycle Manager (OLM) and OperatorHub are now shipped as index images specific to the minor version of OpenShift Container Platform. This allows Operator providers to ship intentional ranges of Operator versions per cluster version.

These index images, based on the Bundle Format, replace the App Registry catalog images, based on

the deprecated Package Manifest Format, that are distributed for previous versions of OpenShift Container Platform 4. OpenShift Container Platform 4.1 through 4.5 will continue to share a single App Registry catalog.



NOTE

While App Registry catalog images are not distributed by Red Hat for OpenShift Container Platform 4.6 and later, custom catalog images based on the Package Manifest Format are still supported.

See Operator Framework packaging formats for more information on the Bundle Format and index images.

Important Operator upgrade requirements
Cluster administrators must ensure all Operators previously installed through Operator Lifecycle Manager (OLM) are updated to their latest versions in their latest channels before upgrading to OpenShift Container Platform 4.6. Updating the Operators ensures that they have a valid upgrade path when the default OperatorHub catalogs switch from using the App Registry catalogs in OpenShift Container Platform 4.5 to the new index image-based catalogs in OpenShift Container Platform 4.6 during the cluster upgrade.

See Upgrading installed Operators for more information on ensuring installed Operators are on the latest channels and upgraded either using automatic or manual approval strategies.

Additional resources

- See the following Red Hat Knowledgebase Article for a list of minimum versions of deployed Red Hat Integration components (including Red Hat Fuse, Red Hat AMQ, and Red Hat 3scale) that are required for OpenShift Container Platform 4.6: https://access.redhat.com/articles/5423161

CNI network provider now uses OVS installed on cluster nodes
Both the OpenShift SDN and OVN-Kubernetes Container Network Interface (CNI) cluster networking providers now use the Open vSwitch (OVS) version installed on the cluster nodes. Previously, OVS ran in a container on each node, managed by a DaemonSet. Using the host OVS eliminates any possible downtime from upgrading the containerized version of OVS.

Warnings when using deprecated APIs
Warnings are now visible in **client-go** and **oc** on every invocation against a deprecated API. Calling a deprecated API returns a warning message containing the target Kubernetes removal release and replacement API, if applicable.

For example:

```
warnings.go:67] batch/v1beta1 CronJob is deprecated in v1.22+, unavailable in v1.25+
```

This is new functionality included with Kubernetes 1.19.

COPY and ADD build instructions improved
The performance of **COPY** and **ADD** instructions in OpenShift Container Platform builds are improved. The initial implementation of **COPY** and **ADD** instructions in **buildah** had noticeable performance regressions compared to **docker**. With this enhancement, builds now run more quickly, especially with large source repositories. (BZ#1833328)

Operator SDK v0.19.4

OpenShift Container Platform supports Operator SDK v0.19.4, which introduces the following notable technical changes:

- Operator SDK now aligns with the OpenShift Container Platform-wide switch to using UBI-8 and Python 3. Downstream base images now use UBI-8 and include Python 3.
- The command **run --local** is deprecated in favor of **run local**.
- The commands **run --olm** and **--kubeconfig** are deprecated in favor of **run packagemanifests**.
- The default CRD version changed from **apiextensions.k8s.io/v1beta1** to **apiextensions.k8s.io/v1** for commands that create or generate CRDs.
- The **--kubeconfig** flag is added to the **<run|cleanup> packagemanifests** command.

Ansible-based Operator enhancements include:

- The Ansible Operator is now available as a supported release.
- The Ansible Operator now includes a **healthz** endpoint and **liveness** probe.

Helm-based Operator enhancements include:

- Helm Operators can watch and reconcile when cluster-scoped release resources are changed.
- Helm Operators can now reconcile logic by using three-way strategic merge patches for native Kubernetes objects so that array patch strategies are correctly honored and applied.
- Helm Operators have the default API version changed to **helm.operator-sdk/v1alpha1**.

UBI 8 used for all images in OpenShift Container Platform
All images in OpenShift Container Platform now use universal base image (UBI) version 8 by default.

Jenkins Node.js agent upgrade
The default Jenkins Node.js agent has been upgraded to Node.js version 12.

Audit logs not gathered by default for oc adm must-gather command
The **oc adm must-gather** command no longer collects audit logs by default. You must include an additional parameter to gather audit logs using the **oc** command. For example:

```
$ oc adm must-gather -- /usr/bin/gather_audit_logs
```

Binary sha256sum.txt.sig file has been renamed for OpenShift Container Platform releases
The **sha256sum.txt.sig** file included in OpenShift Container Platform releases has been renamed to **sha256sum.txt.gpg**. This binary file contains a hash of each of the installer and client binaries, which are used to verify their integrity.

The renamed binary file allows for GPG to correctly verify **sha256sum.txt**, which was not possible previously due to naming conflicts.

1.4. DEPRECATED AND REMOVED FEATURES

Some features available in previous releases have been deprecated or removed.

Deprecated functionality is still included in OpenShift Container Platform and continues to be supported; however, it will be removed in a future release of this product and is not recommended for

new deployments. For the most recent list of major functionality deprecated and removed within OpenShift Container Platform 4.6, refer to the table below. Additional details for more fine-grained functionality that has been deprecated and removed are listed after the table.

In the table, features are marked with the following statuses:

- **GA**: *General Availability*
- **DEP**: *Deprecated*
- **REM**: *Removed*

Table 1.1. Deprecated and removed features tracker

Feature	OCP 4.4	OCP 4.5	OCP 4.6
Service Catalog	DEP	REM	REM
Template Service Broker	DEP	REM	REM
OperatorSources	DEP	DEP	REM
CatalogSourceConfigs	DEP	REM	REM
Package Manifest Format (Operator Framework)	DEP	DEP	DEP
oc adm catalog build	DEP	DEP	DEP
v1beta1 CRDs	GA	DEP	DEP
Metering Operator	GA	GA	DEP

1.4.1. Deprecated features

1.4.1.1. Bring your own RHEL 7 compute machines

The strategy to bring your own (BYO) Red Hat Enterprise Linux (RHEL) 7 compute machines is now deprecated. Support for using RHEL 7 compute machines is planned for removal in OpenShift Container Platform 4.9.

1.4.1.2. TLS verification falling back to the Common Name field

The behavior of falling back to the Common Name field on X.509 certificates as a host name when no Subject Alternative Names are present is deprecated. In a future release, this behavior will be removed, and certificates must properly set the Subject Alternative Names field.

1.4.1.3. Metering Operator

The Metering Operator is deprecated and will be removed in a future release.

1.4.2. Removed features

1.4.2.1. OperatorSources removed

The OperatorSources resource, part of the Marketplace API for the Operator Framework, has been deprecated for several OpenShift Container Platform releases and is now removed. In OpenShift Container Platform 4.6, the default catalogs for OperatorHub in the **openshift-marketplace** namespace now only use CatalogSources with the **polling** feature enabled. The default catalogs poll for new updates in their referenced index images every 15 minutes.

1.4.2.2. MongoDB templates removed

All MongoDB-based samples have been replaced, deprecated, or removed.

1.5. BUG FIXES

apiserver-auth

- Previously, certain conditions caused the Ingress Operator to not push the CA certificate to its **router-certs** secret, so the Cluster Authentication Operator could not construct a trust chain to the certificate in its health checks, causing it to go Degraded and prevent an upgrade. The CA is now always included from the **default-ingress-cert** ConfigMap during the default router CA check, so the Cluster Authentication Operator no longer blocks upgrades. (BZ#1866818)

- Previously, the Cluster Authentication Operator failed to parse HTML pages that were returned from OIDC servers that ignore **Accept: application/json** when a login flow that they do not support is requested, because the Operator was expecting a JSON response. As a result, the Operator failed to honor the IDP configuration. The Cluster Authentication Operator no longer fails with an error when an HTML page is returned from an OIDC server because it does not support the requested flow. (BZ#1877803)

- Previously, ConfigMaps and secrets were not properly validated by the Cluster Authentication Operator, which could cause a new deployment of the OAuth server to roll out with invalid or missing files, causing the pods to crash. ConfigMaps and secrets are now properly validated by the Cluster Authentication Operator, so a new deployment should not roll out when the ConfigMaps or secrets referenced contain invalid data. (BZ#1777137)

Bare Metal Hardware Provisioning

- Previously, the ironic-image container configuration was missing the setting to enable the idrac-redfish-virtual-media boot driver. Because of this, users were unable to select the idrac-virtual-media boot URL for Metal3. The missing ironic-image container configuration is now included, so users are able to select the idrac-virtual-media URL for Metal3. (BZ#1858019)

- Previously, certain Dell firmware versions dropped support for configuring persistent boot using Redfish. Updating Dell iDRAC firmware to version 4.20.20.20 resolves the issue. (BZ#1828885)

- In this release, an issue that resulted in inspection timeouts if many nodes were inspected at the same time has been fixed. (BZ#1830256)

- Previously, the **ironic-image** container configuration was missing the setting to enable the **idrac-redfish-virtual-media** boot driver. Because of this, users were unable to select the **idrac-virtual-media** boot URL for Metal3. The missing **ironic-image** container configuration is now included, so users are able to select the **idrac-virtual-media** URL for Metal3. (BZ#1853302)

- Previously, the HTTPd container in the **metal3** pod in the **openshift-machine-api** namespace that is used for serving bare metal ironic images allowed directory listings. With this release, directory listings are no longer allowed in this container. (BZ#1859334)"

Build

- Errors in **buildah** libraries could ignore certain HTTP errors. Builds could fail to push images due to temporary issues with the target registry. This bug fix updates **buildah** to respect these errors when pushing image blobs. As a result, **buildah** now fails to push an image if the upstream registry is temporarily unavailable. (BZ#1816578)

- Previously, the container image signature policy used in OpenShift Container Platform builds did not contain any configuration for local images. When only allowing images from specific registries, postCommit scripts in builds failed because it was not allowed to use local images. The container image signature policy has been updated to always allow images that reference local storage layers directly. Now builds can successfully complete if they contain a postCommit hook. (BZ#1838372)

- Previously, if a Dockerfile used in Docker strategy builds used the ARG instruction to define build arguments before the first FROM instruction occurred in the Dockerfile, that instruction was dropped when the Dockerfile was preprocessed to incorporate any overrides that were specified in the Build or BuildConfig. References to those arguments were subsequently not resolved properly while building an image using the preprocessed Dockerfile. The preprocessing logic has been modified to preserve ARG instructions which are encountered before the first FROM instruction when generating the updated Dockerfile contents, so this problem no longer occurs. (BZ#1842982)

- Previously, Buildah erased image architecture and OS fields on images. This caused common container tools to fail because the resulting images could not identify their architecture and OS. This bug fix prevents Buildah from overwriting the image and architecture unless there are explicit overrides. This ensures that images always have architecture and OS fields, and the image mismatch warning does not appear. (BZ#1858779)

- Previously, Dockerfile builds failed because they did not expand build arguments correctly in some situations. This update fixes the Dockerfile build argument processing, and thus Dockerfile builds now succeed. (BZ#1839683)

- Previously, Buildah made an extraneous call to read an image from its blob cache, which caused Source-to-Image (S2I) builds to fail. This issue was fixed in Buildah v1.14.11, which was vendored into OpenShift Container Platform builds in 4.6. (BZ#1844469)

- Previously, buildah could not reference images in **COPY –from** Dockerfile instructions. As a result, multistage Dockerfile builds that contained **COPY –from=<image>** failed. Buildah has been updated to a version that supports **COPY –from** instructions. Builds that contain these instructions can now succeed. (BZ#1844596)

Cloud Compute

- Previously, if the **replicas** field on a MachineSet was set to a nil value, the autoscaler could not determine the current number of replicas within the MachineSet and therefore could not perform scaling operations. With this release, the autoscaler uses the last number of observed replicas in the MachineSet as reported by the **replicas** field in the status if a nil value is set. (BZ#1852061)

- Previously, the autoscaler did not balance workloads across different failure domains if a memory discrepancy of more than 128 MB existed between nodes of the same type. With this release, the maximum memory discrepancy is increased to 256 MB. (BZ#1824215)

- Previously, the MachineSet replicas field did not have a default value. As a result, if the field were not present, the MachineSet controller failed silently. The replicas field now has a default value. A default of one replica is used if the replicas field is not set. (BZ#1844596)

- Previously, the MachineHealthCheck controller did not check if a machine had been deleted previously before it attempted to delete it. As a result, the controller could send multiple deletion requests, resulting in spurious logging and event reports. The MachineHealthCheck controller now checks if a machine has been deleted before attempting to delete it. As a result, duplicate logs and events are reduced. (BZ#1844986)

- Previously, the machine API Operator updated the cluster operator machine API when it was in a stable state. As a result, the resource cycled rapidly between states. The resource's status now changes only after changes are rolled out. The status remains stable. (BZ#1855839)

- Previously, setting the ClusterAutoscaler resource values of **balanceSimilarNodeGroups**, **ignoreDaemonSetsUtilization**, or **skipNodesWithLocalStorage** to **false** did not register when the cluster autoscaler was deployed. These values are now read properly when the cluster autoscaler is deployed. (BZ#1854907)

- Rarely, duplicate machine API controller instances could be deployed. As a result, clusters could leak machines that would become inaccessible. Leader election mechanisms are now added to all machine API components to ensure that duplicate instances are not created. Machine API controllers only run the prescribed number of instances. (BZ#1861896)

- On Red Hat Virtualization (RHV) clusters, manual machine scaling could fail. Scaling machines from the Console or CLI now works. (BZ#1817853)

- Previously, must-gather did not collect BareMetalHost records. As a result, debugging information could be incomplete. BareMetalHost records are now collected by must-gather. (BZ#1841886)

- Previously, on clusters that run on Azure, compute machines converted into a "Failed" stage at installation. As a result, VMs were not recognized after being created. Attempts to contact the machines flooded logs with errors, and VMs could fail after starting correctly. As a fix, machines in the **Creating** state are identified as being created already. Logs contain fewer errors, and machines are less likely to fail. (BZ#1836141)

- Previously, MachineHealthCheck could accept negative values for **spec.maxUnhealthy**. As a result, at negative values, numerous events were produced for each reconciliation attempt. Negative values for **spec.maxUnhealthy**. Are now treated as **0**, which reduces spurious log messages. (BZ#1862556)

Cloud Credential Operator

- Previously, when upgrading from OpenShift Container Platform version 4.5 to version 4.6, some fields were updated to the 4.6 default values. This affected the ability to downgrade from 4.6 to 4.5 because the 4.5 field values were not preserved. Rather than leaving the fields unspecified in 4.5, this bug fix explicitly preserves the 4.5 values so they can be specified as default values again on a downgrade attempt. Now downgrading from 4.6 to 4.5 can succeed. (BZ#1868376)

- Previously, the Cloud Credential Operator leader election used the default values from **controller-runtime** and as a result, wrote to **etcd** every two seconds. This release implements custom leader election, which now writes only every 90 seconds and releases the lock immediately on normal shutdown. (BZ#1858403)

Cluster Version Operator

- The Cluster Version Operator served metrics using HTTP instead of HTTPS, and was subject to man-in-the-middle attacks as a result of unencrypted data. Now, The Cluster Version Operator serves metrics using HTTPS and data is encrypted. (BZ#1809195)

- When cluster administrators had configured ClusterVersion overrides, the upgrade process would get stuck. Now, the upgrades are blocked when the override is set. Upgrades will not begin until an administrator removes the overrides. (BZ#1822844)

- The Cluster Version Operator used to load trusted CAs from the ConfigMap referenced by the Proxy configuration's **trustedCA** property. The referenced ConfigMag is user-maintained, and a user setting corrupted certificates would interrupt the Operator's access to the Proxy. Now, the Operator loads the trustedCAs from the **openshift-config-managed/trusted-ca-bundle**, which the Network Operator populates when the Proxy configuration's referenced **trustedCA** ConfigMap is valid. (BZ#1797123)

- HTTPS signatures retrieved serialized stores, causing potential time outs before the Cluster Version Operator could complete the tasks. Now, external HTTPS signature retrieval is parallel and all stores will be attempted. (BZ#1840343)

- Previously, during z-stream cluster upgrades using the option **--to-image**, for example **oc adm upgrade --to-image**, the Cluster Version Operator was using the cluster version being upgraded to, rather than the current cluster version for validation purposes. This caused z-stream upgrades to fail. Now z-stream cluster upgrades using the option **--to-image** are allowed even when Cluster Version Operator has **Upgradeable=false**. (BZ#1822513)

- Previously, the Cluster Version Operator (CVO) was not syncing the shareProcessNamespace parameter in the Pod spec, which caused the Registry Operator to not update the shareProcessNamespace setting. The CVO now syncs shareProcessNamespace, DNSPolicy, and TerminationGracePeriodSeconds, fixing the Registry Operator update issues. (BZ#1866554)

Console Kubevirt Plugin

- Previously, NICs that had the same NIC profile could not be imported successfully, or the wrong network was chosen. The UI now forces users to select the same network, which is not the pod network, for such NICs. (BZ#1852473)

- Virtual machines were not displayed when logged in as a non-administrative user, due to the VM list waiting for the **virtualmachineimports** data to render. Now, the VM list is rendered correctly. (BZ#1843780)

- The Create VM Wizard Edit Disk Modal did not respect cloned PVC namespaces. it was not possible to edit edit **datavolume** disks from a different namespace. Now, the disk modal properly registers the correct namespace of **datavolume** disks. (BZ#1859518)

- Virtual machines and templates could not have the same name when referring to a URL source because the **datavolume** name was hard coded. Now, an auto generated unique string is added to the **datavolume** name and new virtual machines and templates can have the same name. (BZ#1860936)

- Network utilization data would show **Not Available** due to an empty array of data. Now, a check has been implemented and empty arrays are interpreted as no data. (BZ#1850438)

- With this release, the Import VM function is removed from the Developer perspective of the web console. (BZ#1876377)

Console Metal3 Plugin

- The user interface did not detect older Node maintenance CRDs because the EI was searching for the latest version. As a result, the Node Maintenance action would miss the older NodeMaintenance CR if present. Now, the UI observes both NodeMaintenance CRs.

(BZ#1837156)

- The user interface did not correctly evaluate graceful shutdowns, causing incorrect warnings to appear when shutting down the console. Now, the interface waits for the node pods to load and the correct warnings are displayed upon shutdown. (BZ#1872893)

Containers

- Previously, the logic that handled COPY or ADD instructions for copying content from the build context did not efficiently filter when a **.dockerignore** file was present. COPY and ADD would be noticeably slowed by the cumulative overhead of evaluating whether each item in the source location needed to be copied to the destination. This bug fix rewrote the logic and the presence of a **.dockerignore** file will no longer noticeably slow the speed at which COPY and ADD instructions are handled during a build. (BZ#1828119, BZ#1813258)

- Previously, Image builds and pushes would fail with the 'error reading blob from source' error because the builder logic would compute the contents of new layers twice. The logic, which cached layer contents, depended on the products of those calculations remaining consistent. If a new layer's contents changed between the two computations, the cache was unable to supply the layers contents when they were needed. The contents of new layers are no longer computed twice and image builds and pushes will no longer fail. (BZ#1720730)

Web console (Developer perspective)

- Previously, when you tried to delete a Knative application through the **Topology** view, a false positive error about a non-existing **Knative route** was reported. This issue is now fixed and the error is no longer displayed. (BZ#1866214)

- Previously, the Developer Console did not allow images from insecure registries to be imported. This bug fix adds a checkbox that allows users to use the insecure registries in the **Deploy image** form. (BZ#1826740)

- When a user selected the **From Catalog** option to create an application, the **Developer Catalog** displayed a blank page instead of a list of templates to create an application. This was caused when the 1.18.0 Jaeger Operator was installed. This issue has now been fixed and the templates are displayed as expected. (BZ#1845279)

- When deleting a parallel task in a Pipeline through the **Pipeline Builder** in the Developer Console, the interface was rearranging the tasks connected to the parallel task incorrectly, creating orphan tasks. With this fix, the tasks connected to the deleted parallel task are reconnected with the original Pipeline. (BZ#1856155)

- The web console was crashing with a JavaScript exception when the user canceled the creation of a Pipeline through the web console with a side panel opened at the same time. This was fixed by improving the internal state handling. (BZ#1856267)

- A user with the required permissions was unable to retrieve and deploy an image from another project. The required RoleBindings have now been created to fix this issue. (BZ#1843222)

- When you tried to deploy an application from a Git repository with the **Import from Git** function, the Developer Console reported a false positive error **Git repository is not reachable** for private repositories reachable by the cluster. This was fixed by adding information on making the private repository available to the cluster in the error message. (BZ#1877739)

- When a Go application was created through the Developer Console, a route to the application was not created. This was caused by a bug in **build-tools** and incorrectly configured ports. The issue has been fixed by picking either the user-provided port or the default port 8080 as the

target port. (BZ#1874817)

- When you created an application with the **Import from Git** function, a subsequent change of the application's Git repository from the web console was not possible. This was caused by changing the application name in subsequent editing of the Git repository URL. This was fixed by making the application name read-only when editing the application Git repository URL. (BZ#1873095)

- Previously, a user without administrative or project listing privileges could not see the metrics of any projects. This bug fix removes the checks for user privileges when accessing the cluster metrics. (BZ#1842875)

- Users with the @ character in their user names, like **user@example.com**, could not start a Pipeline from the Developer Console. This was caused by a limitation in Kubernetes labels. The issue was fixed by moving the "Started by" metadata from a Kubernetes label to a Kubernetes annotation. (BZ#1868653)

- Previously, when a user selected a metric the QueryEditor would show the query. However, if the user deleted or modified the query and selected the same metric again, the QueryEditor would not update. With this fix, if a query is cleared by the user and they tried to select the same query again, the query input text area will show the query. (BZ#1843387)

- The che-workspace-operator removed support for the Workspace resource and replaced it with the DevWorkspace CRD. As a result, the command line terminal was not enabled with the latest che-workspace-operator. With this fix, the OpenShift command line terminal was moved to use the DevWorkspace resource. The command line terminal will now be enabled in the OpenShift Console when the che-workspace-operator is installed. (BZ#1844938)

- Previously, the route decorator for Knative services redirected users to a revision specific route if traffic was distributed across multiple revisions. The route decorator has been updated to always point to the Knative base service route. (BZ#1860768)"

DNS

- Previously, intermittent invalid memory address or nil pointer dereference errors occurred and were followed by timeouts for Kube API access when running CoreDNS 1.6.6. This is now fixed by correctly handling errors with Endpoint Tombstones. Now CoreDNS behaves as intended without intermittent panics. (BZ#1868315)

- Previously, the DNS Operator repeatedly attempted to update DNS and Service objects in response to default values that were set by the API. With this update, the DNS Operator now considers values that are left unspecified by the DNS Operator to be equal to values in DNS and Service objects. Thus, the DNS Operator no longer updates a DNS or Service object in response to API default values. (BZ#1842741)

etcd

- Previously, the bootstrap endpoint in **ETCDCTL_ENDPOINTS** was not removed after the bootstrap node was removed, so **etcdctl** commands showed unexpected errors. The bootstrap endpoint is no longer added to **ETCDCTL_ENDPOINTS**, so **etcdctl** commands do not show errors related to the bootstrap endpoint. (BZ#1835238)

Image

- Previously, image imports using digests on manifests lists failed. This update fixes the conversion from manifest list to manifest by using the digest of the selected manifest inside the manifest list. Thus, imports by digest of manifest lists now work as expected. (BZ#1751258)

Image Registry

- Previously, some internal packages used an internal error structure, causing a null pointer issue. Now, the internal error interface is returned and nil errors are converted correctly. (BZ#1815562)

- The Operator did not generate **httpSecrets** when empty, causing the value to not set correctly. Now, the Operator generates the **httpSecret** and uses it for all replicas when the configuration file is created. (BZ#1824834)

- Previously, an invalid alert would appear when the image pruner was disabled. This alert has now been removed. (BZ#1845642)

- Registry Operator type assertions were made twice for a variable and the second time the result was not checked. This caused false assertions and created panic conditions. Now, checked type assertions are used and the Operator does not panic. (BZ#1879426)

- Previously, the Operator bootstrap **storageManaged** setting was set to true, causing conflicts if the user manually updated the configuration file. Now, an additional configuration field, **spec.storage.storageManagementState** has been created. Users can indicate **Managed** or **Unmanaged** and the Operator will respect the setting. (BZ#1833109)

- Removing the Image Registry on OpenStack when content was written to storage resulted in storage for the Image Registry not being removed and a 409 HTTP return code error being logged. This bug fix removes the storage content before removing the storage. Now when the Image Registry Operator is removed, its storage is also removed. (BZ#1835770)

- During installation on OpenStack, if there was a failure to access Swift storage during the Image Registry Operator bootstrap process, it caused an incomplete bootstrap. This resulted in a failure to create the Image Registry configuration resource, which blocked fixes or changes its configuration. This bug fix prevents failure during bootstrap in case of a problem when accessing Swift storage. If there is an error, it is logged allowing the bootstrap to finish and configuration resources to be created. Now the Image Registry Operator is now more flexible, and if it cannot access Swift storage, it bootstraps the internal Image Registry using a PVC. (BZ#1846263)

- The Image Registry Operator avoids calling Azure endpoints too many times, because Azure enforces quotas and previously the Operator was constantly querying for storage account keys. This bug fix caches the keys locally for a set period of time to avoid going remotely to get the keys every time they are needed. (BZ#1853734)

- Previously, the operator interpreted a running job as successful when reporting on the operator status, even though the job could potentially result in a failed state. Now the running jobs are ignored when reporting the operator status. (BZ#1857684)

- The image registry purging process failed when running over s3 storage because it was listing directories twice. Now, directorues are listed once and the image purging process completes successfully. (BZ#1861304)

- Previously, if the Image Registry operator was removed, the pruner job failed because it could not reach a non-existent registry. Now, the pruner job removes etcd objects only and does not attempt to ping the registry if it is removed. (BZ#1867792)

- If a user manually added a bucket name, the operator did not create the bucket. Now, the operator creates a bucket successfully based on the user-provided name. (BZ31875013)

- Previously, the Image Registry Operator could not get events from the cluster when it encountered "Too large resource version" errors. With this release, the **client-go** library is

updated to fix the reflector so that the Operator can recover from "Too large resource version" errors. (BZ#1880354)

- Previously, the value provided for **spec.requests.read/write.maxWaitInQueue** in the Image Registry Operator configuration file was not validated. If the provided value was a string that could not be parsed as a duration, the change was not applied, and a message informing about the incorrect value was repeatedly logged. This release adds validation so that a user cannot provide invalid values for this field. (BZ#1833245)

- Previously, dependency tracking between image objects was slow when pruning images. Image pruning would sometimes take a long time to complete. The underlying image pruning mechanism has been redesigned. Now, image pruning is faster and has improved parallelism. (BZ#1860163)

Installer

- Previously, when a machine tried to get the **resourcePoolPath**, it found multiple resource pools and was unable to resolve the correct one. With this release, a property added to the machine sets with the **resourcePoolPath** information helps resolve the correct one. (BZ#1852545)

- Previously, a hardcoded value was set when calculating the end of the DHCP allocation pool when provisioning the nodes subnet. This caused an error when deploying on an OpenShift Container Platform cluster on OpenStack with a machine CIDR smaller than 18. This bug fix removes hardcoding the number of nodes and, instead, dynamically calculates the end of the DHCP allocation pool. Now it is possible to deploy a cluster on OpenStack with a machine CIDR of any length, provided it is large enough for all required nodes. (BZ#1871048)

- Previously, some available networks were not shown during cluster installation because of an **ovirt-engine-sdk-go** API error that affected oVirt network parsing. The issue is now resolved. (BZ#1838559)

- Previously, in the vSphere console wizard, only networks of type **Network** and **DistributedVirtualPortgroup** were displayed even though **OpaqueNetwork** is also a valid option. **OpaqueNetwork** is now an option in the wizard, so that type of network can be selected. (BZ#1844103)

- Previously, the Manila Operator did not support custom self-signed certificates, so the Manila Operator failed to deploy Manila CSI driver on some environments that used self-signed certificates. Now, the Operator fetches the user-provided CA certificate from the system config map, mounts it to the driver's containers, and update the driver's configuration. As a result, the Manila Operator can deploy and manage the Manila CSI driver on environments with self-signed certificates. (BZ#1839226)

- Previously, configuring the **platform.aws.userTags** parameter to add **name** or **kubernetes.io/cluster/** tags to resources that the installation program creates caused the machine-api to fail to identify existing control plane machines and create another set of control plane hosts, which created problems with etcd cluster membership. Now you can no longer set error-prone tags in the **platform.aws.userTags** parameter, and your cluster is less likely to have extra control plane hosts and broken etcd clusters. (BZ#1862209)

- Previously, healthcheck probes were not defined on load balancers deployed in Azure clusters on user-provisioned infrastructure. Because the load balancers were not defined, they did not detect when an API endpoints were no longer available and still directed traffic to them, which caused client failures. Now, the healthcheck probes are used on the load balancers, and they correctly detect when API endpoints are not available and stop routing traffic to offline nodes. (BZ#1836016)

- Previously, the installation program did not accept * as a valid value for the **proxy.noProxy** field, so you could not create a cluster with no proxy set to * during installation. Now, the installation program accepts * as a valid value for that parameter, so you can set no proxy to * during installation. (BZ#1877486)

- Previously when you installed a cluster in GCP, **US** was always used as the location, and it was not possible to install a cluster in some regions outside of US. Now, the installation program sets the right location for the region that you specify, and installations succeed in other locations. (BZ#1871030)

- Previously when you installed a cluster on vSphere with installer-provisioned infrastructure, it was possible to assign the same IP address to ingress and the API, which caused the bootstrap machine and one of the master machines to have the same IP address. Now, the installation program validates that the IP addresses are different, and the master and bootstrap machines have unique IP addresses. (BZ#1853859)

- Previously, when an interface obtained a new dhcp4 or dhcp6 lease, the **local-dns-prepender** did not update the **resolv.conf** file to include all the required resolvers for the cluster. Now, the **dhcp4-change** and **dhcp6-change** action types always make the **local-dns-prepender** start updates.(BZ#1866534)

- Previously, you could not deploy clusters to the following GCP regions: **asia-northeast3**, **asia-southeast-2**, **us-west3**, and **us-west4**. You can now use these regions. (BZ#1847549)

- Previously, the OpenStack installation program used inconsistent output format for the **InstanceID()** function. It obtained the instance ID from either metadata or by sending requests to the server. In the latter case, the result always had '/' prefix, which is the correct format. If the instance ID came from the metadata, the system failed to verify its node existence and failed because of the error. Because, the metadata format now also contains the '/' prefix there, the ID format is always correct, and the system can always successfully verify node existence. (BZ#1850149)

- Previously, provisioning services for the bare metal installer-provisioned infrastructure platform failed when FIPS was enabled for a cluster. With this update, provisioning services run as expected when FIPS is enabled and installation successfully completes. (BZ#1804232)

- Previously, you could configure the DHCP range so that the provisioning network consumed the entire subnet, including the cluster provisioning VIP. Thus, installation would fail because a bootstrap VM IP and cluster provisioning IP could not be assigned. This fix validates VIPs to ensure that they do not overlap with the DHCP range. (BZ#1843587)

- Previously, installation failed when a certificate was followed by two end-of-line character sequences. This update fixes the certificate authority (CA) certificate trust bundle parser, which now ignores invisible characters. Thus, the CA certificate trust bundle can now feature an arbitrary number of end-of-line character sequences before, between, and after certificates. (BZ#1813354)

- Previously, when the interactive installer prompt requested the **ExternalNetwork** for the cluster, all possible network choices were listed, which included invalid options. With this update, the interactive installer now filters the possible options and lists only external networks. (BZ#1881532)

- Previously, the bare metal **kube-apiserver** health check probe used a hardcoded IPv4 address to communicate with the load balancer. This update fixes the health check to use **localhost**, which covers both IPv4 and IPv6 cases. Additionally, the **readyz** endpoint for the API server is checked rather than **healthz**. (BZ#1847082)

- Due to a Terraform step that did not list all of its dependent steps, clusters on Red Hat OpenStack Platform (RHOSP) experienced a race condition that sometimes caused the Terraform job to fail with a "Resource not found" error. The step is now listed as **depends_on** to avoid the race condition. (BZ#1734460)

- Previously, the cluster API did not validate RHOSP flavors before updating machines. As a result, machines that had invalid flavors failed to boot. Flavors are now validated before updating machines, and machines that have invalid flavors return errors immediately. (BZ#1820421)

- Previously, clusters on RHOSP that had periods in their names failed at the bootstrap phase in installation. Periods are no longer allowed in the names of clusters on RHOSP. If a cluster name contains a period, an error message is generated early in the installation process. (BZ#1857158)

- Previously, health check probes were not defined for load balancers in user-provisioned infrastructure deployed clusters on AWS. The load balancers did not detect when API endpoints became unavailable, which caused client failures. In this update, health check probes have been added and the load balancers do not route traffic to offline nodes. (BZ#1836018)

- Previously, the **DiskPostCloneOperation** function in the Terraform vSphere provider checked the **thin_provisioned** and **eagerly_scrub** properties of virtual machines cloned from Red Hat CoreOS OVA images. The check failed because the provisioning type changed during cloning and did not match the source provisioning type. The **DiskPostCloneOperation** function now ignores those properties and the Red Hat CoreOS OVA cloning succeeds. (BZ#1862290)

- When running **./openshift-install destroy cluster**, the installer attempted to remove installer tags before the resources using those tags were removed. The installer subsequently failed to remove the tags. With this update, the installer removes the tags after the corresponding resources are deleted. (BZ#1846125)

kube-apiserver

- Previously, the Cluster Version Operator (CVO) marked a cluster as not upgradeable when the **LatencySensitive** Feature Gate was in use. With this update, the CVO no longer considers **LatencySensitive** Feature Gate as a block for cluster upgrades. To resolve the issue, force the upgrade to the latest 4.5.z or stable version, which include the solution. (BZ#1861431)

kube-controller-manager

- Previously, the DaemonSet controller did not clear expectations when recreating a daemon set. As a result, the daemon set might get stuck for five minutes until the expectations expired. Now the DaemonSet controller correctly clears the expectations. (BZ#1843319)

- UID range allocation is not updated when a project is removed, and thus the UID range can be exhausted on a cluster that has high namespace creation and removal turnover. To resolve the issue, the **kube-controller-manager** pod is periodically restarted, which triggers the repair procedure and clears the UID range. (BZ#1808588)

- Previously, the endpoints controller sent large quantities of API requests at every informer resync interval, which caused degradation in large clusters with many endpoints. Various inconsistencies in storage and the comparison of endpoints caused the endpoints controller to incorrectly assume that there were many additional updates that needed to be made to a cluster than were actually necessary. This fix updates storage and comparison functions for endpoints so that they are more consistent. Thus, the endpoints controller now only sends updates when necessary. (BZ#1854434)

- Previously, NotFound errors were mishandled by controller logic. This caused controllers such as

Deployment, DaemonSet, and ReplicaSet controllers to not be aware of missing pods. This fix updates controllers to correctly react to a pod NotFound event, which indicates that the pod was previously removed. (BZ#1843187)

kube-scheduler

- Previously, when evicting a pod in dry run mode, certain descheduler strategies logged the eviction twice. Now only a single log entry is created for the eviction. (BZ#1841187)

Logging

- Previously, the Elasticsearch index metrics were collected in Prometheus by default. As a consequence, Prometheus would run out of storage quickly, due to the size of the index-level metrics, and the user would need to intervene, for example, reducing the Prometheus retention time, to keep Prometheus functioning. The default behavior was changed to not collect the Elasticsearch index metrics. (BZ#1858249)

- Because the Elasticsearch Operator was not updating the secret hash for the Kibana deployment, Kibana pods would not be restarted if the secret gets updated. The code was changed to correctly update the hash for the deployment, which triggers a the pods to redeploy, as expected. (BZ#1834576)

- Because of the introduction of Fluentd init containers, Fluentd could not be deployed in a cluster without deploying the Elasticsearch Operator (EO). The code was changed to alllow Fluentd without the EO being present. As a result, in a cluster without Elasticsearch, Fluentd works as expexted. (BZ#1849188)

- Because the ability to open Kibana in an iframe was not expressly denied, it opened Kibana to the possibility of attack, such as clickjacking. The code was changed to explicitly set **x-frame-options:deny**, which blocks the use of iframes. (BZ#1832783)

Machine Config Operator

- In bare metal environments, an **infra-dns** container runs on each host to support node name resolutions and other internal DNS records. A NetworkManager script also updates the **/etc/resolv.conf** on the host to point to the **infra-dns** container. Additionally, when pods are created, they receive their DNS configuration file (the **/etc/resolv.conf** file) from their hosts. If an HAProxy pod was created before NetworkManager scripts update the **/etc/resolv.conf** file on the host, the pod can repeatedly fail because the **api-int** internal DNS record is not resolvable. This bug fix updates the Machine Config Operator (MCO) to now verify that the **/etc/resolv.conf** file of the HAProxy pod is identical to the host **/etc/resolv.conf** file. As a result, the HAProxy pod no longer experiences these errors. (BZ#1849432)

- Keepalived is used to provide high availability (HA) for both the API and default router. The Keepalived instance in each node monitors local health by curling the health endpoint of the local entity, for example the local **kube-apiserver**. Previously, the **curl** command failed only when the TCP connection failed, and not on HTTP non-200 errors. This caused Keepalived to sometimes not failover to another healthy node, even when the local entity was unhealthy, which lead to errors in API requests.
  This bug fix updates the Machine Config Operator (MCO) to modify the **curl** command to now also fail when the server replies with a non-200 retcode. As a result, the API and Ingress router now correctly failover to a healthy node in case of failure in a local entity. (BZ#1844387)

- In bare metal environments, some DNS records were hard-coded for IPv4. This caused some records to not be served correctly in IPv6 environments, which might necessitate creating those records in an external DNS server. This bug fix updates the Machine Config Operator (MCO) so

that DNS records are now populated correctly based on the Internet Protocol version in use. As a result, internal records are now served correctly in both IPv4 and IPv6. (BZ#1820785)

- Previously, kernel arguments specified in MachineConfigs needed to be split out into individual argument strings in the array. These arguments were not validated before being concatenated into an **rpm-ostree** command. Multiple kernel arguments concatenated using a space, as allowed in a single line in the kernel command line, would create an invalid **rpm-ostree** command. This bug fix updates the MachineConfigController to parse each **kernelArgument** item in a similar manner as the kernel. As a result, users can supply multiple arguments concatenated using a space without errors. (BZ#1812649)

- Previously, the control plane would always be schedulable for user workloads in bare metal environments. This bug fix updates the Machine Config Operator (MCO) so that control plane nodes are now correctly configured as **NoSchedule** in a typical deployment with workers. (BZ#1828250)

- Previously with the Machine Config Operator (MCO), unnecessary API VIP moves could cause client connection errors. This bug fix updates API VIP health checks to limit the number of times it moves. As a result, there are now fewer errors caused by API VIP moves. (BZ#1823950)

Web console (Administrator perspective)

- The Operand's tab for Operand list view was missing. With this bug fix, the issue is resolved. (BZ#1842965)

- When IPv6 was disabled, the downloads pod socket could not bind and the downloads pod crashed. If IPv6 is not enabled, IPv4 is now used for the socket. The downloads pod now works regardless of enabling IPv4 and IPv6. (BZ#1846922)

- The Operator status display values did not account for manual approval strategy. Therefore, the **upgrade available** status was displayed, which did not convey that further action was required in order to upgrade. A new status message was added for Operators that are waiting for manual approval to upgrade. You can now clearly tell when an Operator upgrade requires further action. (BZ#1826481)

- The catch logic for a failed operand form submission was attempting to access a deep property in the resulting error object that is not always defined. For specific failed request types, this would cause a runtime error. With this bug fix, the issue is resolved. (BZ#1846863)

- Victory does not handle all-zero data sets well. Y-axis tick marks for all-zero data sets were repeated zeros. The area chart logic was updated to force a Y-domain of **[0,1]** when all data sets are all-zero. Now, Y-axis tick marks for all-zero data sets are **0** and **1**. (BZ#1856352)

- The currently installed version was not shown in the Operator details pane on OperatorHub. Therefore, a user could not determine if the currently installed version was latest. When the currently installed Operator version is not the latest, the installed version is now shown in the Operator details pane on OperatorHub. (BZ#1856353)

- The **create role binding** links were inconsistently namespaced or used cluster links, so the **create role binding** page was incorrect. This bug fix updates the links to use the namespace, where available, and the cluster in other cases. The correct page is now used. (BZ#1871996)

- Previously, the web console did not support Grafana **valueMaps** for singlestat panels, so singlestat panels were unable to display Grafana valueMaps. Support is now added and singlestat panels can display Grafana **valueMaps**. (BZ#1866928)

- **oc debug** was updated with BZ#1812813 to create a new debug project with an empty node

selector in order to work around a problem where **oc debug** only works against worker nodes. The web console avoided this problem by asking users to choose a namespace when visiting the **Node > Terminal** page before the terminal is opened, resulting in inconsistency in the user experience compared to **oc**. The web console now creates a new debug project with an empty node selector upon visiting the **Node > Terminal** page. The behavior of the web console **Node > Terminal** page now aligns with the behavior of **oc debug**. (BZ#1881953)

- When all receivers were deleted, the web console would experience a runtime error and the user was presented with a blank screen. Null checks are now added in the code and users ate presented with a **No Receivers** empty state screen instead. (BZ#1849556)

- In some cases, the value passed to the resource requirements widget in the legacy operand creation form might not be an immutablejs map instance. A runtime error was thrown when trying to reference the immutablejs **Map.getIn** function on the resource requirement widget current value. Use optional chaining when referencing the immutablejs **Map.getIn** function. No runtime error is thrown and the widget is rendered without a value. (BZ#1883679)

- A blank page appeared when using search on the **imagemanifestvuln** resource. The component was using **props.match.params.ns**, which was sometimes undefined. As a result, there was a runtime error. This issue is now resolved. (BZ#1859256)

- Previously, the action menu to the right of operand lists could close immediately after opening. This could be seen on the **Installed Operators** details page when clicking one of the tabs for the Operator-provided APIs. The menu now functions correctly and will no longer close without user interaction. (BZ#1840706)

- There was a missing keyword field in the API and users were unable to search by keyword in OperatorHub. With this bug fix, the issue is resolved. (BZ#1840786)

- Previously, the Operator Hub in the web console sometimes showed the incorrect icon for an Operator. The issue has been resolved in this release. (BZ#1844125)

- In this release, a broken link to the cluster monitoring documentation from the web console OperatorHub install page is corrected. (BZ#1856803)

- Previously, clicking Create EtcdRestore on the EtcdRestores page caused the web console to stop responding. With this release, the Create EtcdRestore form view workflow loads correctly. (BZ#1845815)

- Previously, the Operand form array and object fields did not have logic to retrieve and show field descriptions on the form. As a result, descriptions were not rendered for array or object type fields. This bug fix adds logic to now display array and object field descriptions on the Operand creation form. (BZ#1854198)

- Previously, the Deployment Configuration Overview page sometimes crashed with the error **e is undefined** when a new Pod was starting up. The issue has been resolved in this release. (BZ#1853705)

- Previously, the cluster upgrade interface in the web console was visible on OpenShift Dedicated, even though OpenShift Dedicated users cannot perform cluster upgrades. The cluster upgrade interface is now hidden for OpenShift Dedicated. (BZ#1874257)

- Previously, empty objects in an operand template were pruned when submitting the form or switching between the form and YAML view. With this release, template data is used as a mask when pruning empty structures from the form data, and only values that are not defined in the template are pruned. Empty values that are defined in a template remain. (BZ#1847921)

- The tooltip details were sometimes truncated when hovering over the Cluster Logging donut chart. This bug fix enables the entire tooltip description to display for this chart. (BZ#1842408)

- For the Registry field on the Create Instance page, some of the text in the schema property descriptions matched the format used to create fuzzy hyperlinks in Linkify. This resulted in unintended hyperlinks. This bug fix disables the fuzzy link feature in Linkify. Now, only URL strings using a proper protocol scheme are rendered as hyperlinks. (BZ#1841025)

- The AWS Secret field always displayed the Loading icon even when the ListDropdown component was not in a loading state. This bug fixes the component logic so that the Loading icon only displays when the drop-down list is in a loading state. A placeholder displays if the drop-down list is not in a loading state. (BZ#1845817)

- Previously, Resource Log pages in the web console became slow or unresponsive when the logs contained long lines. This bug fixes the performance issues by limiting the number of characters per log line that displays on the Resource Log page and providing an alternate method to view the full log content. (BZ#1874558)

- Previously, the node file system calculations for Used and Total were incorrect due to an incorrect query. Now, the query is updated and calculates the data correctly. (BZ31874028)

- The Overview page in the web console incorrectly included the container level usage metric to display network utilization data. This bug fix replaces the incorrect metric with the node level usage metric to accurately display network utilization data. (BZ#1855556)

- Previously, the Created At timestamp format for operators did not require a specific format, resulting in the timestamp displaying inconsistently on the web console. Now, if the Created At timestamp value is entered as a valid date for an operator, the timestamp displays in a consistent manner with other timestamps in the console. If the value is not entered using a valid date format, the Created At timestamp displays as an invalid string. (BZ#1855378)

- The web console displayed an old logo for OperatorHub. This bug fix replaces the old logo with the current logo. (BZ#1810046)

- Previously, the operator resource field names displayed inconsistently in the web console as either camelCase or Start Case. Now, the operand form generation logic labels the form fields using Start Case by default. The default can be overridden by CSV or CRD objects. (BZ#1854196)

- Previously, the Resource Log view did not show single line logs and did not show the last line of a pod log if it was missing a newline control character (/n). Now, the log view has been updated to show the entire pod log content for single line logs and final lines that are not terminated with a newline character. (BZ#1876853)

- Previously, the OpenShift Container Platform web console YAML editor allowed **metadata.namespace** entries for all resources. An error was returned when **namespace: <namespace>** was added for resources that did not take a namespace value. **metadata.namespace** entries are now removed when a configuration is saved, if a resource does not take a namespace value. (BZ#1846894)

- Previously, the delete icon (**-**) for the name value editor did not provide a tooltip, so it was not clear to users what this icon did. The the delete icon (**-**) now provides a tooltip to make its action easier to understand. (BZ#1853706)

- Previously, resource names with special characters, such as **(** and **)**, could prevent resource details from being displayed in the OpenShift Container Platform console. Resource details are now displayed properly when resource names have special characters. (BZ#1845624)

Monitoring

- Previously, the configuration reload for Prometheus was sometimes triggered while the configuration was not fully generated, which triggered the **PrometheusNotIngestingSamples** and **PrometheusNotConnectedToAlertmanagers** alerts because there were no scrape or alerting targets. The configuration reload process now ensures that the configuration on disk is valid before reloading Prometheus, and the alerts are not fired. (BZ#1845561)

- Previously the **AlertmanagerConfigInconsistent** alert could fire during an upgrade because some of the Alertmanager pods were temporarily not running due to a rolling update of the stateful set. Even though the alert resolved itself, this could cause confusion for cluster administrators. The **AlertmanagerConfigInconsistent** no longer considers the number of running Alertmanager pods, so it no longer fires during upgrades when some of the Alertmanager pods are in a not-running, transient state. (BZ#1846397)

- Previously, some alerts did not have the correct severity set or were incorrect, which caused upgrade issues. The severity level for many alerts were changed from critical to warning, the **KubeStatefulSetUpdateNotRolledOut** and **KubeDaemonSetRolloutStuck** alerts were adjusted, and the **KubeAPILatencyHigh** and **KubeAPIErrorsHigh** alerts were removed. These alerts are now correct and should not cause upgrade issues. (BZ#1824988)

- Previously, **KubeTooManyPods** alert used the **kubelet_running_pod_count**, which includes completed pods, so was incorrect for the **KubeTooManyPods** alert. Now, **container_memory_rss** is leveraged to find the actual number of pods running on a node for the **KubeTooManyPods** alert. (BZ#1846805)

- Previously, the **node_exporter** daemon set defaulted to a **maxUnavailable** of **1**, so the rollout was entirely serialized and slow on large clusters. Since the **node_exporter** daemon set does not affect workload availability, the **maxUnavailable** now scales with the cluster size, which allows for a faster rollout. (BZ#1867603)

Networking

- With this release, Kuryr-Kubernetes now attempts to detect the bridge interface for Pod subports on the Container Network Interface (CNI) level, instead of using a value set in **kuryr.conf**. This approach supports cases when VMs call the interface without using a value set in **kuryr.conf**. (BZ#1829517)

- Previously, OpenShift SDN exposed metrics unencrypted over HTTP. Now OpenShift SDN exposes metrics over TLS. (BZ#1809205)

- Previously, when idling a service OpenShift SDN did not always delete a service and its endpoints in the correct order. As a result, sometimes the node port for a service was not deleted. When the service scaled up again, it was therefore unreachable. Now OpenShift SDN ensures that the node port is always deleted correctly. (BZ#1857743)

- Previously, an egress router pod failed to initialize because of a missing dependency on legacy iptables binaries. Now an egress router pod successfully initializes. (BZ#1822945)

- Previously, when deleting network policies on a cluster that is using the OVN-Kubernetes cluster networking provider, a race condition prevented network policies from being reliably deleted when deleting the associated namespace. Now network policy objects are always correctly deleted. (BZ#1859682)

- Previously, when using the OpenShift SDN pod network provider in multitenant isolation mode, pods were unable to reach services configured with **externalIPs** set. Now pods can reach services with a service external IP address configured. (BZ#1762580)

- Previously, OVN-Kubernetes exposed metrics unencrypted over HTTP. Now OVN-Kubernetes exposes metrics over TLS. (BZ#1822720)

- Previously, on OpenShift Container Platform on Red Hat OpenStack Platform, when using the OVN-Octavia driver, it was not possible to attach listeners to different protocols on the same port. Now it is possible to expose several protocols on the same port. (BZ#1846396)

Node

- Previously, a Kubelet failed to start if a soft eviction threshold and grace period were not specified. With this release, the presence of these values is verified during Kubelet configuration. (BZ#1805019)

- Because the user could enter invalid characters, such as negative values, non-numeric characters, and so forth, in the CPU and memory requests for a kubeconfig object, the kubelet would not start. The code has been changed to verify that the kubelet config memory request values are valid. As a result, invalid values are rejected. (BZ#1745919)

- Previously, if a system were using a device mapper for the root device, a number of key host level IO metrics returned by cadvisor were incorrectly set to zero. cadvisor was fixed to report these metrics if the device mapper is used for root. (BZ#1831908)

oauth-apiserver

- Previously, when the Authentication Operator receives an HTML payload from an OpenID Connect Authentication (OIDC) server that ignores the **Accept: application/json** header, the Operator logged an error about the payload. Now the Operator includes the URL of the page that it requested to aid in troubleshooting the OIDC server response. (BZ#1861789)

oc

- Previously, the **oc project** command required the privileges of a **self-provisioner** role to switch projects, which meant that some users could not switch projects if they did not have that role. The **self-provisioner** role requirement was removed, so anyone with access to a project can now switch projects using **oc project**. (BZ#1849983)

- Previously, sorting events by **lastTimestamp** could cause an error when sorting an event that has an empty **lastTimestamp**. Sorting is now resistant to empty elements, and works properly when sorting by **lastTimestamp**. (BZ#1880283)

- Previously, the **oc create job** command was missing logic for the **--save-config** flag, so the **--save-config** option did not work as expected. Logic was added for the **--save-logic** flag, and it now works properly. (BZ#1844998)

OLM

- Operator Lifecycle Manager (OLM) exposes the **subscription.spec.config.nodeSelector** field in the subscription CRD, but previously did not apply **nodeSelectors** labels to the deployments defined in the ClusterServiceVersion (CSV). This made users unable to set **nodeSelectors** on their CSV deployments. This bug fix updates OLM to now propagate **nodeSelector** labels defined in the **subscription.spec.config.nodeSelector** field to deployments in the CSV. As a result, the field now works as expected. (BZ#1860035)

- Previously, Operator Lifecycle Manager (OLM) did not reuse existing valid CA certificates when installing a ClusterServiceVersion (CSV) that entered the **installing** phase multiple times. OLM applied a new webhook hash to the deployment, causing a new replica set to be created. The

running Operator then redeployed, possibly many times during an installation. This bug fix updates OLM to now check if the CA already exists and reuses it if valid. As a result, if OLM detects existing valid CAs, OLM now reuses the CAs. (BZ#1868712)

- Operator Lifecycle Manager (OLM) appends **OwnerReferences** metadata to service resources installed for Operators that provide API services. Previously, whenever an Operator of this class was redeployed by OLM, for example during a certificate rotation, a duplicate **OwnerReference** was appended to the related service, causing the number of **OwnerReferences** to grow unbounded. With this bug fix, when adding **OwnerReferences**, OLM now updates an existing **OwnerReference** if found. As a result, the number of **OwnerReferences** appended to a service by OLM is bounded. (BZ#1842399)

- Operator Lifecycle Manager (OLM) previously did not pull bundle images before attempting to unpack them, causing the **opm alpha bundle validate** command to fail with "image not found" or similar errors. This bug fix updates OLM to now pull bundle images before attempting to unpack in the bundle validator. As a result, the **opm alpha bundle validate** command successfully pulls and unpacks images before performing validation. (BZ#1857502)

- Previously, the web console would choose Operator icons to display in OperatorHub by returning the icon from the first channel declared in the package. This sometimes caused the displayed icon to be different than the latest icon published to the package. This has been fixed by choosing the icon from the default channel, which ensures the latest icon is displayed. (BZ#1843652)

- Previously, whiteout files appeared in unpacked content when using **podman** or **docker** tooling options. With this release, whiteout files are no longer present after unpacking with **podman** and **docker** tooling options. (BZ#1841178)

openshift-controller-manager

- Previously, intermittent availability issues with the API server could lead to intermittent issues with the OpenShift Controller Manager Operator retrieving deployments. Failure to retrieve a deployment sometimes caused the Operator to panic. With this release, checks have been added to handle and report this error condition, and to retry the operation. The Operator now properly handles intermittent issues retrieving deployments from the API server. (BZ#1852964)

RHCOS

- Previously, machines with a large number of NICs on networks without DHCP took a long time to boot. This was caused by the initramfs using legacy network scripts that attempted to bring up DHCP on every interface on the machine, one interface at a time. Now, initramfs uses NetworkManager instead of the legacy scripts. NetworkManager does not attempt DHCP on any interface that does not have a physical connection. The NetworkManager also attempts DHCP on interfaces in parallel, instead of one-at-a-time. These changes reduce the waiting time for DHCP timeouts. (BZ#1836248)

- Previously, it was not possible to modify kernel arguments during installation. Now, kernel arguments can be modified on an installed system using the **coreos-installer** command. For example, you can configure the installed system to use a different serial console argument using:

```
$ coreos-installer install ... \
 --delete-karg console=ttyS0,115200n8 \
 --append-karg console=ttyS1,115200n8
```

(BZ#1712511)

- When MCO was being used to deploy a worker node, it failed to load the file because the user-configured iSCSI initiator name in the Ignition configuration was automatically replaced by a dynamically generated name. Now, the iSCSI initiator name is generated dynamically only when a name has not been specified in the Ignition configuration. (BZ#1868174)

- Manual changes to Azure VMs during provisioning altered the incarnation number, which then caused the afterburn read state reporting to fail due to the incarnation numbers not matching. Afterburn now fetches a fresh incarnation number just before posting the ready state. (BZ#1853068)

- Some interfaces, such as bond interfaces, were not visible in the console. The NetworkManager dispatch scripts replaced the previously used Udev rules. This fix enables network interfaces that have either permanent hardware addresses or that are backed by devices with permanent hardware addresses to display in the console. (BZ#1866048)

Routing

- Previously, the HAProxy router 503 page was not compliant with the standards used by some web application firewalls. The 503 page has been updated to resolve this issue. (BZ#1852728)

- When the Ingress Operator reconciles an IngressController object configured for the NodePortService endpoint publishing strategy type, the Operator gets the ingresscontroller's NodePort service from the API to determine whether the operator needs to create or update the service. If the service does not exist, the operator creates it, with an empty value for the **spec.sessionAffinity** field. If the service does exist, the operator compares it with what the Ingress Operator expects in order to determine whether an update is needed for that service. In this comparison, if the API has set the default value, **None**, for the service **spec.sessionAffinity** field, the Operator detects the update and tries to set the **spec.sessionAffinity** field back to the empty value.
  As a result, the Ingress Operator repeatedly attempts to update the NodePort Services in response to the blank. The Ingress Operator was changed to consider unspecified values and default values to be equal when comparing NodePort service. The operator no longer updates an IngressController NodePort service in response to API defaulting. (BZ#1842742)

- Previously, if you updated a cluster with improper routes, HAProxy would initialize into a defunct state. However, the update would not trigger any alerts and the cluster would incorrectly report the Ingress Controller as available. HAProxy initial sync logic was fixed so that upgrades with broken routes fail. As a result, upgrading a cluster with improper routes is not successful and the HAProxyReloadFail or HAProxyDown alerts are reported. (BZ#1861455)

- Because of the risk of connection re-use/coalescing when using HTTP/2 ALPN, a warning message was added to the output of the Ingress Controller in the CLI to use to enable HTTP/2 ALPN only on a route that uses custom (non-wildcard) certificate. As a result, routes that do not have their own custom certificate will not be HTTP/2 ALPN-enabled on either the frontend or the backend. (BZ#1827364)

- Previously, when HAProxy was reloaded, HAProxy Prometheus counter metrics would decrease in value, which explicitly violates the definition of a counter metric. The router code was fixed to note the time of the last metrics scrape. This prevents scraping beyond the preserved counter values during a reload. As a result the counter metrics do not show a sudden increase followed by a decrease when the router is reloading. (BZ#1752814)

Samples

- Previously, an alert rule for the Samples Operator misspelled the registry.redhat.io host name. The rule now uses the correct host name in the alert message. (BZ#1863014)

- Previously, when upgrading OpenShift Container Platform the Samples Operator might block the upgrade if the API server is intermittently unavailable. Now the Operator handles intermittent connectivity gracefully and upgrades are no longer blocked. (BZ#1854857)

Storage

- Error message content for Local Storage Operator logging was too generic to help with debugging. Additional details are now provided when a LocalVolume object is created and the specified device is either not found or invalid. (BZ#1840127)

- Storage Operator stopped reconciling when **Upgradable=False** on v1alpha1 CRDs. As a result, the Cluster Storage Operator could not perform z-stream upgrades when these CRDs were detected on the cluster. This fix changed the order of the reconciliation loop. Now, z-stream upgrades complete successfully and v1alpha1 CRDs are detected without errors. (BZ#1873299)

- Manila volumes could not be unmounted after the NFS driver pod restarted because the restart process assigned a new IP address to the pod. Now the pod uses the host network and host IP address to mount and unmount volumes, even after restarting the driver pod. (BZ#1867152)

- This bug fix reduces log noise when changing fsGroup of a small volume. (BZ#1877001)

- A condition in the vSphere cloud provider could cause failures in persistent volume provisioning in rare cicumstances due to a heavy load. This bug fixes the condition and allows vSphere volumes to provision reliably. (BZ#1806034)

- Upgrades from OCP 4.3.z to 4.4.0 fail when clusters have installed v1alpha1 VolumeSnapshot CRDs manually or by an independent CSI driver. OpenShift Container Platform 4.4 introduced v1beta1 VolumeSnapshot CRDs, which are not compatible with v1alpha1 VolumeSnapshot CRDs. Now, the Cluster Storage Operator checks for the presence of v1alpha1 VolumeSnapshot CRDs. If detected, a message displays stating that the v1alpha1 VolumeSnapshot CRDs must be removed for the upgrade to proceed. (BZ#1835869)

- When VolumeSnapshotContent deletion policies were modified, the VolumeSnapshot instances could not be deleted because the finalizers associated with those instances were not updated. This bug fix removes the finalizers when the VolumeSnapshotContent deletion policies are modified, and allows the VolumeSnapshot instances to be deleted after the associated resource objects are removed. (BZ#1842964)

- Previously, the default OpenShift RBAC rules did not allow regular users to access or create VolumeSnapshot and VolumeSnapshotClass instances. Now, the default OpenShift RBAC rules allow basic-users to read/write VolumeSnapshots and read VolumeSnapshotClasses. Additionally, storage-admins can read/write VolumeSnapshotContents by default. (BZ#1842408)

- Previously, there was no validation to prevent the Local Storage Operator from creating one device to multiple PVs. This release adds validation for this scenario so that trying to create a PV on a block device where one is already provisioned by the Local Storage Operator will fail. (BZ#1744385)

Insights Operator

- Previously, the Insights Operator could gather data for an unlimited number of certificate signing requests (CSRs) in a single report. This resulted in excessive data collection for clusters with many CSRs. The Insights Operator now gathers data for a maximum of 5000 CSRs in a single report. (BZ#1881044)

1.6. 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 are marked with the following statuses:

- **TP**: *Technology Preview*
- **GA**: *General Availability*
- **-**: *Not Available*

Table 1.2. Technology Preview tracker

Feature	OCP 4.4	OCP 4.5	OCP 4.6
Precision Time Protocol (PTP)	TP	TP	TP
oc CLI Plug-ins	TP	TP	TP
experimental-qos-reserved	TP	TP	TP
Pod Unidler	TP	GA	GA
Ephemeral Storage Limit/Requests	TP	TP	TP
Descheduler	TP	TP	TP
Podman	TP	TP	TP
Sharing Control of the PID Namespace	TP	GA	GA
OVN-Kubernetes cluster networking provider	TP	TP	GA
HPA custom metrics adapter based on Prometheus	TP	TP	TP
HPA for memory utilization	TP	TP	TP
Three-node bare metal deployments	TP	GA	GA
Service Binding	TP	TP	TP
Log forwarding	TP	TP	GA
Monitoring for user-defined projects	TP	TP	GA

Feature	OCP 4.4	OCP 4.5	OCP 4.6
Compute Node Topology Manager	TP	GA	GA
Raw Block with Cinder	TP	TP	TP
External provisioner for AWS EFS	TP	TP	TP
CSI volume snapshots	TP	TP	TP
CSI volume cloning	TP	TP	GA
CSI AWS EBS Driver Operator	-	TP	TP
OpenStack Manila CSI Driver Operator	-	GA	GA
Red Hat Virtualization (oVirt) CSI Driver Operator	-	-	GA
CSI inline ephemeral volumes	-	TP	TP
Automatic device discovery and provisioning with Local Storage Operator	-	-	TP
OpenShift Pipelines	TP	TP	TP
Vertical Pod Autoscaler	-	TP	TP
Operator API	-	TP	GA
Adding kernel modules to nodes	TP	TP	TP
Docker Registry v1 API			DEP

1.7. KNOWN ISSUES

- Currently, upgrading from OpenShift Container Platform 4.5 to 4.6 with the OVN-Kubernetes cluster networking provider will not work. This will be resolved in a future 4.6.z release. (BZ#1880591)
- Currently, when scaling to greater than 75 nodes in a cluster, the OVN-Kubernetes cluster networking provider database might become corrupt, leaving the cluster in an unusable state. (BZ#1887585)
- Currently, scaling up Red Hat Enterprise Linux (RHEL) worker nodes on a cluster with the OVN-Kubernetes cluster networking provider will not work. This will be resolved in a future RHEL 7.8.z and RHEL 7.9.z release. (BZ#1884323, BZ#1871935)

- Currently, when scaling up a worker node that is running on Red Hat Enterprise Linux (RHEL) 7.8, the OVN-Kubernetes cluster networking provider fails to initialize on the new node. (BZ#1884323)

- Downgrading from OpenShift Container Platform 4.6 to 4.5 will be fixed in a future 4.5.z release. (BZ#1882394, BZ#1886148, BZ#1886127)

- Currently, upgrading from OpenShift Container Platform 4.5 to 4.6 with Red Hat Enterprise Linux (RHEL) worker nodes does not work. This will be resolved in a future 4.6.z release. First, upgrade RHEL, then upgrade the cluster, and then run the normal RHEL upgrade playbook again. (BZ#1887607)

- OpenShift Container Platform 4.5 to 4.6 upgrade fails when an external network is configured on a bond device; the **ovs-configuration** service fails and nodes becomes unreachable. This will be resolved in a future 4.6.z release. (BZ#1887545)

- Currently, HugePages are not detected properly when requested in several Non-Uniform Memory Access (NUMA) nodes. This is caused by the cnf-tests suite reporting errors when clusters contain several NUMA nodes because the tests compare the number of HugePages on one NUMA with the total number of HugePages on the entire node. (BZ#1889633)

- The Data Plane Development Kit (DPDK) test used to check packet forwarding and receiving always fails. (BZ#1889631)

- The Stream Control Transmission Protocol (SCTP) validation phase fails when there is at least one machine configuration with no raw configuration. For example, this would include machine configurations containing only kernel arguments. (BZ#1889275)

- The Precision Time Protocol (PTP) validation phase fails because the cnf-tests suite does not properly detect the number of nodes running PTP. (BZ#1889741)

- The Network Interface Card (NIC) validation phase fails because it does not wait for the devices on the node to become available. The waiting time for a pod to start running on the node is too short, so a pod could still have the status **Pending** and be incorrectly tested. (BZ#1890088)

- The **ose-egress-dns-proxy** image has a known defect that prevents the container from starting. This image is also broken in earlier releases, so this is not considered a regression in 4.6. (BZ#1888024)

- In OpenShift Container Platform 4.1, anonymous users could access discovery endpoints. Later releases revoked this access to reduce the possible attack surface for security exploits because some discovery endpoints are forwarded to aggregated API servers. However, unauthenticated access is preserved in upgraded clusters so that existing use cases are not broken.
  If you are a cluster administrator for a cluster that has been upgraded from OpenShift Container Platform 4.1 to 4.6, you can either revoke or continue to allow unauthenticated access. It is recommended to revoke unauthenticated access unless there is a specific need for it. If you do continue to allow unauthenticated access, be aware of the increased risks.

WARNING

If you have applications that rely on unauthenticated access, they might receive HTTP 403 errors if you revoke unauthenticated access.

Use the following script to revoke unauthenticated access to discovery endpoints:

```
## Snippet to remove unauthenticated group from all the cluster role bindings
$ for clusterrolebinding in cluster-status-binding discovery system:basic-user
system:discovery system:openshift:discovery ;
do
### Find the index of unauthenticated group in list of subjects
index=$(oc get clusterrolebinding ${clusterrolebinding} -o json | jq 'select(.subjects!=null) |
.subjects | map(.name=="system:unauthenticated") | index(true)');
### Remove the element at index from subjects array
oc patch clusterrolebinding ${clusterrolebinding} --type=json --patch "[{'op': 'remove','path':
'/subjects/$index'}]";
done
```

This script removes unauthenticated subjects from the following cluster role bindings:

- **cluster-status-binding**
- **discovery**
- **system:basic-user**
- **system:discovery**
- **system:openshift:discovery**

(BZ#1821771)

- Running the **operator-sdk new** or **operator-sdk create api** commands without the **--helm-chart** flag builds a Helm-based Operator that uses the default boilerplate Nginx chart. While this example chart works correctly on upstream Kubernetes, it fails to deploy successfully on OpenShift Container Platform.
  To work around this issue, use the **--helm-chart** flag to provide a Helm chart that deploys successfully on OpenShift Container Platform. For example:

```
$ operator-sdk new <operator_name> --type=helm \
  --helm-chart=<repo>/<name>
```

(BZ#1874754)

- When installing OpenShift Container Platform on bare metal nodes with the Redfish Virtual Media feature, a failure occurs when the Baseboard Management Controller (BMC) attempts to load the virtual media image from the provisioning network. This happens if the BMC is not using the provisioning network, or its network does not have routing set up to the provisioning network. As a workaround, when using virtual media, the provisioning network must be turned off, or the BMCs must be routed to the provisioning network as a prerequisite. (BZ#1872787)

- The OpenShift Container Platform installation program does not support the Manual mode configuration by using the **install-config.yaml** file on GCP and Azure due to a known issue. Instead, you must manually insert a ConfigMap into the manifest directory during the manifest generation stage of the cluster installation process as documented in Manually creating IAM for Azure and Manually creating IAM for GCP. (BZ#1884691)

- On a power environment, when a pod is created using the FC persistent volume claim and the targetWWN as a parameter, the FC volume attach fails with “no fc disk found” error and the pod remains in **ContainerCreating state**. (BZ#1887026)

- When a node providing an egress IP is shut down, the pods hosted on that node are not moved to another node providing an egress IP. This causes the outgoing traffic of the pods to always fail when a node providing an egress IP is shut down. (BZ#1877273)

- Private, disconnected cluster installations are not supported for AWS GovCloud when installing in the **us-gov-east-1** region due to a known issue. (BZ#1881262).

- Firewall rules used by machines not prefixed with the infrastructure ID are preserved when destroying a cluster running on Google Cloud Platform (GCP) with installer-provisioned infrastructure. This causes the destroy process of the installation program to fail. As a workaround, you must manually delete the firewall rule of the machine in the GCP web console:

  ```
  $ gcloud compute firewall-rules delete <firewall_rule_name>
  ```

  Once the firewall rule of the machine with the missing infrastructure ID is removed, the cluster can be destroyed. (BZ#1801968)

- The **opm alpha bundle build** command fails on Windows 10. (BZ#1883773)

- In OpenShift Container Platform 4.6, the resource metrics API server provides support for custom metrics. The resource metrics API server does not implement the OpenAPI specification, and the following messages are sent to the **kube-apiserver** logs:

  ```
  controller.go:114] loading OpenAPI spec for "v1beta1.metrics.k8s.io" failed with: OpenAPI spec does not exist
  controller.go:127] OpenAPI AggregationController: action for item v1beta1.metrics.k8s.io: Rate Limited Requeue.
  ```

  In some cases, these errors might cause the **KubeAPIErrorsHigh** alert to fire, but the underlying issue is not known to degrade OpenShift Container Platform functionality. (BZ#1819053)

- Rules API back-ends are sometimes not detected if Store API stores are discovered before Rules API stores. When this occurs, a store reference is created without a Rules API client, and the Rules API endpoint from Thanos Querier does not return any rules. (BZ#1870287)

- If an AWS account is configured to use AWS Organizations service control policies (SCPs) that use a global condition to deny all actions or require a specific permission, the AWS policy simulator API that validates permissions produces a false negative. When the permissions cannot be validated, OpenShift Container Platform AWS installations fail, even if the provided credentials have the required permissions for installation.
  To work around this issue, you can bypass the AWS policy simulator permissions check by setting a value for the **credentialsMode** parameter in the **install-config.yaml** configuration file. The value of **credentialsMode** changes the behavior of the Cloud Credential Operator (CCO) to one of three supported modes.

  Example install-config.yaml configuration file

  ```
  apiVersion: v1
  baseDomain: cluster1.example.com
  credentialsMode: Mint 1
  compute:
  - architecture: amd64
    hyperthreading: Enabled
  ...
  ```

1 This line is added to set the **credentialsMode** parameter to **Mint**.

When bypassing this check, ensure that the credentials you provide have the permissions that are required for the specified mode.

(BZ#1829101)

- Clusters that run on RHOSP and use Kuryr create unnecessary Neutron ports for each **hostNetworking** pod. You can delete these ports safely. Automatic port deletion is planned for a future release of OpenShift Container Platform. (BZ#1888318)

- Deployments on RHOSP configured with Kuryr might experience kuryr-cni pods going into a CrashLoop, which reports a **NetlinkError: (17, 'File exists')** error. As a workaround, you must reboot the node. A fix is planned to resolve this issue in a future release of OpenShift Container Platform. (BZ#1869606)

- When deploying an egress router pod in DNS proxy mode, the pod fails to initialize. (BZ#1888024)

- RHCOS real time (RT) kernels are currently only supported on compute nodes, not control plane nodes. Compact clusters are not supported with RT kernels in OpenShift Container Platform 4.6. (BZ#1887007)

- To increase security, the **NET_RAW** and **SYS_CHROOT** capabilities are no longer available in the default list of CRI-O capabilities.

  - **NET_RAW**: If unprotected, this capability enables pods to craft packets that can change header fields, such as low ports, source IP address, and source MAC address. This functionality could allow malicious hacking attempts.

  - **SYS_CHROOT**: Normal workloads should not require **chroot**. Access to privileged operations should be granted only when required.

  The **NET_RAW** and **SYS_CHROOT** capabilities were removed as default capabilities in OpenShift Container Platform 4.5.16. To reduce impact to clusters created in releases before 4.5.16, the default capabilities list is now contained in separate machine configs: **99-worker-generated-crio-capabilities** and **99-master-generated-crio-capabilities**. OpenShift Container Platform creates the new machine configs when you upgrade from a previous release.

  After upgrading, it is recommended to disable the **NET_RAW** and **SYS_CHROOT** capabilities, and then test your workloads. When you are ready to remove these capabilities, delete the **99-worker-generated-crio-capabilities** and **99-master-generated-crio-capabilities** machine configs.

  Important: If you are upgrading from an earlier release, upgrade to 4.5.16 before you upgrade to 4.6. (BZ#1874671).

- The OpenShift Container Platform Machine API bare metal actuator currently deletes Machine objects when the underlying bare metal host is deleted. This behavior does not align with other cloud provider actuators, which move a Machine object to the failed phase rather than removing it altogether if the underlying cloud provider resource is deleted. (BZ#1868104)

- When upgrading a cluster from version 4.5 to 4.6 that was installed with installer-provisioned infrastructure on vSphere, the upgrade fails if the control plane node IP addresses change during the upgrade. As a workaround, you must reserve the control plane node IP addresses

before upgrading to version 4.6. Review your DHCP server's documentation for the configuration of reservations. (BZ#1883521)

- For **oc** commands that require TLS verification, if the certificates do not set a Subject Alternative Name, verification does not fall back to the Common Name field and the command fails with the following error:

  > x509: certificate relies on legacy Common Name field, use SANs or temporarily enable Common Name matching with GODEBUG=x509ignoreCN=0

  As a workaround, you can either use a certificate with a proper Subject Alternative Name set, or precede the **oc** command with **GODEBUG=x509ignoreCN=0** to temporarily override this behavior.

  A future 4.6 z-stream might return a warning instead of an error, to allow more time for users to update their certificates to be compliant.

  (BZ#1889204)

- When you install Agones using Helm package manager and try to examine the chart resources in your namespace using the **Developer** perspective, you see an error message instead of the resource details. (BZ#1866087)

- When you select a deployment in the **Topology** view, click **Actions → Edit <deployment_name>**, and then modify it; the modified **Deployment** YAML file overwrites or removes the volume mounts in the **Pod Template spec**. (BZ#1867965)

- No success or failure message is displayed in the **Developer** perspective when you use the **Add → From Catalog** option, filter by **Template**, select a template, and then instantiate a template. (BZ#1876535)

- PipelineRuns with skipped tasks incorrectly show the tasks as **Failed**. (BZ#1880389)

- The **Application Details** page, in the **Application Stages** view, provides inaccurate links for the projects in the application environment. (BZ#1889348)

- Under the condition of heavy Pod creation, creation fails with **error reserving pod name ...: name is reserved**. CRI-O's context for the CNI executable ends and it kills the process. Pod creation succeeds eventually, but it takes a lot of time. Therefore, the kubelet thinks that CRI-O did not create the Pod. The kubelet sends the request again and a name conflict occurs. This issue is currently under investigation. (BZ#1785399)

- If the cluster networking provider is OVN-Kubernetes, when using a Service external IP address that is not assigned to any node in the cluster, network traffic to the external IP address is not routable. As a workaround, ensure that a Service external IP address is always assigned to a node in the cluster. (BZ#1890270)

- Administrators can mirror the **redhat-operators** catalog to use Operator Lifecycle Manager (OLM) on OpenShift Container Platform 4.6 clusters in restricted network environments (also known as disconnected clusters). However, the following Operators return entries in the **mapping.txt** file with the private host name **registry-proxy.engineering.redhat.com** instead of the expected public host name **registry.redhat.io**:

  - amq-online.1.5.3
  - amq-online.1.6.0

This causes image pulls to fail against the inaccessible private registry, normally intended for internal Red Hat testing. To work around this issue, run the following command after generating your **mapping.txt** file:

```
$ sed -i -e 's/registry-proxy.engineering.redhat.com/registry.redhat.io/g' \
    -e 's/rh-osbs\/amq7-/amq7\//g' \
    -e 's/amq7\/tech-preview-/amq7-tech-preview\//g' \
    ./redhat-operator-index-manifests/imageContentSourcePolicy.yaml \
    ./redhat-operator-index-manifests/mapping.txt
```

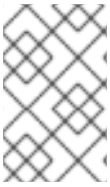
For OpenShift Container Platform on IBM Power Systems on PowerVM, the following requirements are preferred:

- 2 virtual CPUs for master nodes
- 4 virtual CPUs for worker nodes
- 0.5 processors for all nodes
- 32 GB virtual RAM for all nodes

1.8. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Container Platform 4.6 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 4.6 errata is available on the Red Hat Customer Portal . See the OpenShift Container Platform Life Cycle for more information about asynchronous errata.

Red Hat Customer Portal users can enable errata notifications in the account settings for Red Hat Subscription Management (RHSM). When errata notifications are enabled, users are notified via email whenever new errata relevant to their registered systems are released.



NOTE

Red Hat Customer Portal user accounts must have systems registered and consuming OpenShift Container Platform entitlements for OpenShift Container Platform errata notification emails to generate.

This section will continue to be updated over time to provide notes on enhancements and bug fixes for future asynchronous errata releases of OpenShift Container Platform 4.6. Versioned asynchronous releases, for example with the form OpenShift Container Platform 4.6.z, will be detailed in subsections. In addition, releases in which the errata text cannot fit in the space provided by the advisory will be detailed in subsections that follow.

IMPORTANT

For any OpenShift Container Platform release, always review the instructions on updating your cluster properly.

1.8.1. RHBA-2020:4196 - OpenShift Container Platform 4.6 image release and bug fix advisory

Issued: 2020-10-27

OpenShift Container Platform release 4.6 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2020:4196 advisory. The RPM packages that are included in the update are provided by the RHBA-2020:4197 advisory.

Space precluded documenting all of the container images for this release in the advisory. See the following article for notes on the container images in this release:

OpenShift Container Platform 4.6.1 container image list

1.8.2. RHSA-2020:4297 - Moderate: OpenShift Container Platform 4.6 package security updates

Issued: 2020-10-27

An update for **jenkins-2-plugins**, **openshift-clients**, **podman**, **runc**, and **skopeo** is now available for OpenShift Container Platform 4.6. Details of the update are documented in the RHSA-2020:4297 advisory.

1.8.3. RHSA-2020:4298 - Moderate: OpenShift Container Platform 4.6 image security updates

Issued: 2020-10-27

An update for several images is now available for OpenShift Container Platform 4.6. Details of the update are documented in the RHSA-2020:4298 advisory.

1.8.4. RHBA-2020:4339 - OpenShift Container Platform 4.6.3 bug fix update

Issued: 2020-11-09

OpenShift Container Platform release 4.6.3 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2020:4339 advisory. The RPM packages that are included in the update are provided by the RHBA-2020:4340 advisory.

Space precluded documenting all of the container images for this release in the advisory. See the following article for notes on the container images in this release:

OpenShift Container Platform 4.6.3 container image list

1.8.4.1. Bug fixes

- Due to a known issue, the GPU Operator and Node Feature Discovery (NFD) Operator were not available on a fresh installation of OpenShift Container Platform 4.6.1. You were required to install OpenShift Container Platform 4.5 and upgrade the cluster to version 4.6.1 to use the GPU and NFD Operators. This issue has been fixed, and the GPU and NFD Operators are now available on a fresh installation of OpenShift Container Platform 4.6.3 and later. (**BZ#1890673**)

1.8.4.2. Upgrading

To upgrade an existing OpenShift Container Platform 4.6 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.8.5. RHBA-2020:4987 - OpenShift Container Platform 4.6.4 bug fix update

Issued: 2020-11-16

OpenShift Container Platform release 4.6.4 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2020:4987 advisory. There are no RPM packages for this release.

Space precluded documenting all of the container images for this release in the advisory. See the following article for notes on the container images in this release:

OpenShift Container Platform 4.6.4 container image list

1.8.5.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.6 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.8.6. RHBA-2020:5115 - OpenShift Container Platform 4.6.6 bug fix update

Issued: 2020-11-30

OpenShift Container Platform release 4.6.6 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2020:5115 advisory. The RPM packages that are included in the update are provided by the RHBA-2020:5116 advisory.

Space precluded documenting all of the container images for this release in the advisory. See the following article for notes on the container images in this release:

OpenShift Container Platform 4.6.6 container image list

1.8.6.1. Bug fixes

- Prior to OpenShift Container Platform 4.6, when the Marketplace Operator used the **OperatorSource** custom resource definition (CRD), cluster administrators using a cluster on a restricted network (also known as a disconnected cluster) could disable the default **OperatorSource** objects in the **openshift-marketplace** namespace and create custom **CatalogSource** objects with the same name as the default sources. In OpenShift Container Platform 4.6, the Marketplace Operator uses **CatalogSource** objects directly now that the **OperatorSource** CRD is removed. As a result, the **openshift-marketplace** has default catalog sources that are managed by the OperatorHub API.
  After disabling the default catalog sources on a disconnected OpenShift Container Platform 4.6 cluster, when an administrator tried to create a catalog source with the same name as that of the default sources, the OperatorHub API previously removed the custom catalog source. If the catalog source was not disabled using the OperatorHub API and changes were made to the default catalog source (for example, changing the **spec.image** parameter to point to an internal registry for the disconnected environment), the spec was restored to the default spec.

  This bug fix allows cluster administrators to create, update, and delete custom catalog sources with the same name as the default sources, if they are disabled using the OperatorHub API. As a result, administrators can now disable the default catalog sources and create custom catalog sources using the default names without them being removed or overwritten. If the default catalog source is re-enabled, the default spec is restored. (**BZ#1895952**)

1.8.6.2. Upgrading

To upgrade an existing OpenShift Container Platform 4.6 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.8.7. RHSA-2020:5159 - Low: OpenShift Container Platform 4.6 package security updates

Issued: 2020-11-30

An update for **golang** is now available for OpenShift Container Platform 4.6. Details of the update are documented in the RHSA-2020:5159 advisory.

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 4.6 cluster, all masters must be 4.6 and all nodes must be 4.6. If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.6. 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 (4.1 to 4.2 to 4.3, for example) to allow older **oc** to update. Using new capabilities may require newer **oc**. A 4.3 server may have additional capabilities that a 4.2 **oc** cannot use and a 4.3 **oc** may have additional capabilities that are not supported by a 4.2 server.

Table 2.1. Compatibility Matrix

	X.Y (**oc** Client)	**X.Y+N** footnote:versionpolicyn[Where **N** is a number greater than 1.] (**oc** Client)
X.Y (Server)	1	3
X.Y+N footnote:versionpolicyn[] (Server)	2	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.