OpenShift Container Platform 4.7

Release notes

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

Last Updated: 2021-05-06

OpenShift Container Platform 4.7 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 4.7 RELEASE NOTES 6
1.1. ABOUT THIS RELEASE 6
1.2. MAKING OPEN SOURCE MORE INCLUSIVE 6
1.3. NEW FEATURES AND ENHANCEMENTS 6
1.3.1. Red Hat Enterprise Linux CoreOS (RHCOS) 6
1.3.1.1. Enhanced disk provisioning for LUKS, RAID, and FBA DASD 6
1.3.1.2. Update the bootloader by using bootupd 7
1.3.1.3. RHCOS now supports RHEL 8.3 7
1.3.1.4. RHCOS now supports kdump service (Technology Preview) 7
1.3.1.5. Ignition updates 7
1.3.1.6. Configuring the timeout value used when trying to acquire a DHCP lease 7
1.3.1.7. RHCOS supports multipath 8
1.3.1.8. Fetching configs on AWS from Instance Metadata Service Version 2 (IMDSv2) 8
1.3.1.9. Qemu guest agent is now included in RHCOS 8
1.3.2. Installation and upgrade 8
1.3.2.1. Installing a cluster into the AWS C2S Secret Region 8
1.3.2.2. Installing cluster on GCP with disk encryption using a personal encryption key 9
1.3.2.3. Installing a cluster on RHOSP that uses bare metal machines 9
1.3.2.4. Improved RHOSP requirements validation at installation 9
1.3.2.5. Custom subnets for new compute machines on RHOSP 9
1.3.2.6. Easier access to RHOSP user-provisioned infrastructure playbooks 9
1.3.2.7. Support for the QEMU Guest Agent on RHOSP 9
1.3.2.8. Increasing the persistent volume limit for clusters on RHOSP 9
1.3.2.9. Deprecation of the computeFlavor property in the install-config.yaml file 9
1.3.2.10. Using static DHCP reservations for the bootstrap host for clusters with installer-provisioned infrastructure 10
1.3.2.11. Enhancements to installer-provisioned installation 10
1.3.2.12. Installer-provisioned clusters can convert DHCP leases to static IP addresses 10
1.3.2.13. Updates are immediately blocked if a machine config pool is degraded 10
1.3.3. Web console 10
1.3.3.1. Web console localization 10
1.3.3.2. Quick start tutorials 10
1.3.3.3. Insights plug-in 11
1.3.3.4. Developer perspective 11
1.3.3.5. IBM Z and LinuxONE 12
Notable Enhancements 13
Supported features 13
Restrictions 13
1.3.3.6. IBM Power Systems 14
Notable Enhancements 14
Supported Features 14
Restrictions 15
1.3.4. Security and compliance 15
1.3.4.1. Managing user-owned OAuth access tokens 15
1.3.4.2. Cloud Credential Operator support for deletion of GCP root credentials after installation 15
1.3.4.3. CIS Kubernetes Benchmark profile for the Compliance Operator 15
1.3.4.4. Secure Boot support for installer-provisioned clusters 16
1.3.4.5. Advanced Cluster Management 2.2 integration 16
1.3.5. Networking 16
1.3.5.1. Expanded platform support for migrating from the OpenShift SDN cluster network provider to the OVN-Kubernetes cluster network provider 16

1.3.5.2. Network connection health checks for API servers, load balancers, and nodes 16
1.3.5.3. OVN-Kubernetes egress firewall support for DNS rules 16
1.3.5.4. Library for interacting with SR-IOV virtual functions in DPDK mode within containers 16
1.3.5.5. Egress router CNI (Technology Preview) 17
1.3.5.6. OVN-Kubernetes IPsec support for encrypted traffic between pods 17
1.3.5.7. SR-IOV network node policy enhancement for deployment with Red Hat OpenStack Platform (RHOSP) 17
1.3.5.8. RHOSP Kuryr support for services without pod selectors 17
1.3.5.9. Adjusting HTTP header names 17
1.3.5.10. Kubernetes NMState Operator (Technology Preview) 17
1.3.6. Storage 18
1.3.6.1. Persistent storage using CSI volume snapshots is generally available 18
1.3.6.2. Persistent storage using the GCP PD CSI Driver Operator (Technology Preview) 18
1.3.6.3. Persistent storage using the OpenStack Cinder CSI Driver Operator 18
1.3.6.4. vSphere Problem Detector Operator 18
1.3.6.5. Local Storage Operator now collects custom resources 18
1.3.7. Registry 18
1.3.7.1. Open Container Initiative images support 18
1.3.7.2. New image stream metrics 18
1.3.8. Operator lifecycle 18
1.3.8.1. Safe Operator upgrades 19
1.3.8.2. Adding pull secrets to catalog sources 19
1.3.8.3. Mirroring the content of an Operator catalog into a container image registry 19
1.3.8.4. Creating new install plan for better experience 19
1.3.8.5. Mirroring images to a disconnected registry by first mirroring the images to local files 19
1.3.9. Operator development 20
1.3.9.1. Operator SDK now fully supported 20
1.3.10. Builds 21
Print the buildah version to the build log 21
Cluster Samples Operator assistance for mirroring 21
1.3.11. Machine API 21
1.3.11.1. Machine sets running on AWS support Dedicated Instances 21
1.3.11.2. Machine sets running on GCP support customer-managed encryption keys 21
1.3.11.3. Machine API components honor cluster-wide proxy settings 21
1.3.11.4. Some machine configuration updates no longer cause automatic reboot 21
1.3.11.5. BareMetalHost API supports soft shutdown 22
1.3.12. Nodes 22
1.3.12.1. Descheduler is generally available 22
1.3.12.2. Scheduler profiles (Technology Preview) 22
1.3.12.3. Autoscaling for memory utilization GA 23
1.3.12.4. Autoscaling to zero machines for clusters on RHOSP 23
1.3.12.5. Non-preempting option for priority classes (Technology Preview) 23
1.3.12.6. Specifying CPUs for node host processes with CRI-O 23
1.3.13. Red Hat OpenShift Logging 23
Cluster Logging becomes Red Hat OpenShift Logging 23
1.3.14. Monitoring 23
1.3.14.1. Alerting rule changes 23
1.3.14.2. Version updates to monitoring stack components and dependencies 25
1.3.14.3. AlertmanagerConfig CRD in Prometheus Operator not supported 25
1.3.14.4. New API Performance monitoring dashboard 25
1.3.14.5. Namespace (Pods) and Pod Kubernetes networking dashboards are enabled in Grafana 26
1.3.14.6. HWMon data collection for hardware telemetry for bare metal clusters 26
1.3.14.7. logLevel configuration field for Thanos Querier 26

1.3.14.8. Removed memory limit on config-reloader container for monitoring user-defined projects 26
1.3.14.9. Removed deprecated Technology Preview configuration for monitoring your own services 26
1.3.15. Scale 27
1.3.15.1. Cluster maximums 27
1.3.15.2. Test to determine CPU latency 27
1.3.15.3. New globallyDisableIrqLoadBalancing feature in Performance Addon Operator allows global device interrupt processing to be disabled for guaranteed pod CPUs 27
1.3.15.4. New VRF CNI plug-in allows secondary networks to be assigned to VRFs 27
1.3.15.5. The xt_u32 end-to-end test is enabled for CNF 27
1.3.16. Developer experience 27
1.3.16.1. Red Hat OpenShift GitOps (Technology Preview) 28
1.3.17. Insights Operator 28
1.3.17.1. Insights Operator data collection enhancements 28
1.3.18. Authentication and authorization 28
1.3.18.1. Running OpenShift Container Platform using AWS Security Token Service (STS) for credentials (Technology Preview) 28
1.4. NOTABLE TECHNICAL CHANGES 28
Operator Lifecycle Manager updated to use Kubernetes 1.20 29
1.5. DEPRECATED AND REMOVED FEATURES 29
1.5.1. Deprecated features 30
1.5.1.1. Scheduler policy 30
1.5.1.2. Catalog mirroring using filter-by-os flag 30
1.5.1.3. ImageChangesInProgress condition for Cluster Samples Operator 30
1.5.1.4. MigrationInProgress condition for Cluster Samples Operator 30
1.5.1.5. Use of v1 for apiVersion for OpenShift Container Platform resources 30
1.5.2. Removed features 31
1.5.2.1. Installer-provisioned clusters no longer require provisioningHostIP or bootstrapProvisioningIP 31
1.5.2.2. Images removed from samples imagestreams 31
1.5.2.3. oc items removed 31
1.6. BUG FIXES 31
1.7. TECHNOLOGY PREVIEW FEATURES 58
1.8. KNOWN ISSUES 59
1.9. ASYNCHRONOUS ERRATA UPDATES 64
1.9.1. RHEA-2020:5633 - OpenShift Container Platform 4.7.0 image release, bug fix, and security update advisory 64
1.9.2. RHBA-2021:0678 - OpenShift Container Platform 4.7.1 bug fix update 64
1.9.2.1. Upgrading 65
1.9.3. RHBA-2021:0749 - OpenShift Container Platform 4.7.2 bug fix update 65
1.9.3.1. Upgrading 65
1.9.4. RHBA-2021:0821 - OpenShift Container Platform 4.7.3 bug fix update 65
1.9.4.1. Upgrading 65
1.9.5. RHSA-2021:0957 - OpenShift Container Platform 4.7.4 bug fix and security update 65
1.9.5.1. Upgrading 66
1.9.6. RHSA-2021:1005 - OpenShift Container Platform 4.7.5 bug fix and security update 66
1.9.6.1. Features 66
1.9.6.1.1. Installing a cluster on VMC on AWS 66
1.9.6.1.2. Adding memory and uptime metadata to the Insights Operator archive 66
1.9.6.1.3. SAP license management enhancement 66
1.9.6.2. Upgrading 67
1.9.7. RHBA-2021:1075 - OpenShift Container Platform 4.7.6 bug fix update 67
1.9.7.1. Bug fixes 67
1.9.7.2. Features 68
1.9.7.2.1. BareMetal Operator Enhancement 68

1.9.7.2.2. Cluster API provider BareMetal (CAPBM) enhancement 68
1.9.7.3. Upgrading 68
1.9.8. RHBA-2021:1149 - OpenShift Container Platform 4.7.7 bug fix and security update 68
1.9.8.1. Bug fixes 68
1.9.8.2. Upgrading 69
1.9.9. RHSA-2021:1225 - OpenShift Container Platform 4.7.8 bug fix and security update 69
1.9.9.1. Upgrading 69
1.9.10. RHBA-2021:1365 - OpenShift Container Platform 4.7.9 bug fix and security update 69
1.9.10.1. Bug fixes 69
1.9.10.2. Upgrading 69

CHAPTER 2. OPENSHIFT CONTAINER PLATFORM VERSIONING POLICY 70

CHAPTER 1. OPENSHIFT CONTAINER PLATFORM 4.7 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 (RHEL) 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.

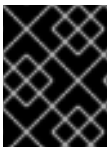
1.1. ABOUT THIS RELEASE

OpenShift Container Platform (RHSA-2020:5633) is now available. This release uses Kubernetes 1.20 with CRI-O runtime. New features, changes, and known issues that pertain to OpenShift Container Platform 4.7 are included in this topic.

OpenShift Container Platform 4.7 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 4.7 is supported on Red Hat Enterprise Linux (RHEL) 7.7 or later, as well as Red Hat Enterprise Linux CoreOS (RHCOS) 4.7.

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



IMPORTANT

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

With the release of OpenShift Container Platform 4.7, version 4.4 is now end of life. For more information, see the Red Hat OpenShift Container Platform Life Cycle Policy .

1.2. MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see Red Hat CTO Chris Wright's message .

1.3. NEW FEATURES AND ENHANCEMENTS

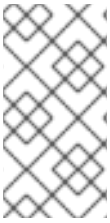
This release adds improvements related to the following components and concepts.

1.3.1. Red Hat Enterprise Linux CoreOS (RHCOS)

1.3.1.1. Enhanced disk provisioning for LUKS, RAID, and FBA DASD

OpenShift Container Platform 4.7 includes several improvements to disk provisioning for bare metal deployments. The following features are currently supported for new 4.7 clusters only:

- Native Ignition support for LUKS disk encryption provides additional configurability for encrypted root filesystems, as well as support for encryption of additional data filesystems.
- RHCOS now supports boot disk mirroring, except on s390x, providing redundancy in the case of disk failure. For more information, see Mirroring disks during installation.
- RHCOS on s390x can be installed onto fixed-block architecture (FBA)-type direct access storage device (DASD) disks.
- RHCOS now supports the primary disk being multipathed.



NOTE

On new clusters, LUKS configuration must use the native Ignition mechanism, as provisioning fails if the legacy **/etc/clevis.json** file is included in the machine config. On clusters that are upgrading from OpenShift Container Platform 4.6 or earlier, LUKS can only be configured by using **/etc/clevis.json**.

1.3.1.2. Update the bootloader by using bootupd

With **bootupd**, RHCOS users now have access to a cross-distribution, system-agnostic OS update tool that manages firmware and boot updates in UEFI and legacy BIOS boot modes that run on modern architectures.

1.3.1.3. RHCOS now supports RHEL 8.3

RHCOS is now using Red Hat Enterprise Linux (RHEL) 8.3 packages. OpenShift Container Platform 4.6 and below will stay with RHEL 8.2 packages. This enables you to have the latest fixes, features, and enhancements, such as NetworkManager features, as well as the latest hardware support and driver updates.

1.3.1.4. RHCOS now supports kdump service (Technology Preview)

The **kdump** service is introduced in Technology Preview in RHCOS to provide a crash-dumping mechanism for debugging kernel issues. You can use this service to save system memory content for later analysis. The **kdump** service is not managed at the cluster-level and must be enabled and configured manually on a per-node basis. For more information, see Enabling kdump.

1.3.1.5. Ignition updates

The following Ignition updates are now available:

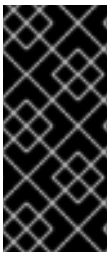
- RHCOS now supports Ignition config spec 3.2.0. This update provides support for disk partition resizing, LUKS encrypted storage, and **gs://** URLs.
- When executing in non-default AWS partitions, such as GovCloud or AWS China, Ignition now fetches **s3://** resources from the same partition.
- Ignition now supports AWS EC2 Instance Metadata Service Version 2 (IMDSv2).

1.3.1.6. Configuring the timeout value used when trying to acquire a DHCP lease

Previously, RHCOS DHCP kernel parameters were not working as expected because acquiring a DHCP lease would take longer than the default 45 seconds. With this fix, you now have the ability to configure the timeout value that is used when trying to acquire a DHCP lease. See BZ#1879094 for more information.

1.3.1.7. RHCOS supports multipath

RHCOS now supports multipath on the primary disk, allowing stronger resilience to hardware failure so that you can set up RHCOS on top of multipath to achieve higher host availability. See BZ#1886229 for more information.



IMPORTANT

Only enable multipathing with kernel arguments within a machine config as documented. Do not enable multipathing during installation.

For more information, see "Enabling multipath with kernel arguments" in Post-installation machine configuration tasks.

1.3.1.8. Fetching configs on AWS from Instance Metadata Service Version 2 (IMDSv2)

Ignition now supports fetching configs on AWS from Instance Metadata Service Version 2 (IMDSv2). With this enhancement, AWS EC2 instances can be created with IMDSv1 disabled so that IMDSv2 is needed to read the Ignition config from instance userdata. As a result, Ignition successfully reads its config from instance userdata, regardless of whether IMDSv1 is enabled or not. See BZ#1899220 for more information.

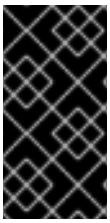
1.3.1.9. Qemu guest agent is now included in RHCOS

The Qemu guest agent is now included by default in RHCOS. With this enhancement, Red Hat Virtualization (RHV) administrators can see rich information about RHCOS nodes through the reporting of useful information about RHCOS back to the RHV management interface. See BZ#1900759 for more information.

1.3.2. Installation and upgrade

1.3.2.1. Installing a cluster into the AWS C2S Secret Region

You can now install a cluster on Amazon Web Services (AWS) into the Commercial Cloud Services (C2S) Secret Region. Because the C2S region does not have an RHCOS AMI published by Red Hat, you must upload a custom AMI that belongs to that region. You are also required to include the CA certificates for C2S in the **additionalTrustBundle** field of the **install-config.yaml** file during cluster installation. Clusters deployed to the C2S Secret Region do not have access to the Internet; therefore, you must configure a private image registry.



IMPORTANT

It is currently not possible to use the AWS Secure Token Service (STS), which is a Technology Preview feature, in a cluster installed into the AWS C2S Secret Region due to current OpenShift Container Platform limitations. This includes using temporary credentials provided from the C2S Access Portal (CAP).

The installation program does not support destroying a cluster deployed to the C2S region; you must manually remove the resources of the cluster.

For more information, see AWS government and secret regions.

1.3.2.2. Installing cluster on GCP with disk encryption using a personal encryption key

You can now install a cluster on Google Cloud Platform (GCP) and use a personal encryption key to encrypt both virtual machines and persistent volumes. This is done by setting the **controlPlane.platform.gcp.osDisk.encryptionKey**, **compute.platform.gcp.osDisk.encryptionKey**, or **gcp.defaultMachinePlatform.osDisk.encryptionKey** field in the **install-config.yaml** file.

1.3.2.3. Installing a cluster on RHOSP that uses bare metal machines

You can now install a cluster on your own Red Hat OpenStack Platform (RHOSP) infrastructure that uses bare metal machines. The cluster can have both control plane and compute machines running on bare metal, or just compute machines. For more information, see Deploying a cluster with bare metal machines.

This feature is not supported on clusters that use Kuryr.

1.3.2.4. Improved RHOSP requirements validation at installation

The OpenShift Container Platform installer now performs additional validations before attempting to install a cluster on RHOSP. These new validations include:

- Resource quotas
- Floating IP addresses duplication
- Custom cluster OS image availability

1.3.2.5. Custom subnets for new compute machines on RHOSP

You can now create compute machines in clusters that run on RHOSP that use a network and subnet of your choice.

1.3.2.6. Easier access to RHOSP user-provisioned infrastructure playbooks

Ansible playbooks for installing a cluster on your own RHOSP infrastructure are now packaged for retrieval by using a script in the installation documentation.

1.3.2.7. Support for the QEMU Guest Agent on RHOSP

You can now enable QEMU Guest Agent support during installation.

1.3.2.8. Increasing the persistent volume limit for clusters on RHOSP

You can now configure nodes to have more than 26 persistent Cinder volumes in clusters on RHOSP during installation.

1.3.2.9. Deprecation of the computeFlavor property in the install-config.yaml file

The **computeFlavor** property that is used in the **install-config.yaml** file is deprecated. As an alternative, you can now configure machine pool flavors in the **platform.openstack.defaultMachinePlatform** property.

1.3.2.10. Using static DHCP reservations for the bootstrap host for clusters with installer-provisioned infrastructure

In previous versions of OpenShift Container Platform, you could not assign a static IP address to the bootstrap host of a bare metal installation that used installer-provisioned infrastructure. Now, you can specify the MAC address that is used by the bootstrap virtual machine, which means you can use static DHCP reservations for the bootstrap host. See BZ#1867165 for more information.

1.3.2.11. Enhancements to installer-provisioned installation

The installer for installer-provisioned installation on bare metal nodes now automatically creates a storage pool for storing relevant data files required during the installation, such as ignition files.

The installer for installer-provisioned installation on bare metal nodes provides a survey which asks the user a minimal set of questions, and generates an **install-config.yaml** file with reasonable defaults. You can use the generated **install-config.yaml** file to create the cluster, or edit the file manually before creating the cluster.

1.3.2.12. Installer-provisioned clusters can convert DHCP leases to static IP addresses

Cluster nodes deployed with installer-provisioned installation on bare metal clusters can deploy with static IP addresses. To deploy a cluster so that nodes use static IP addresses, configure a DHCP server to provide infinite leases to cluster nodes. After the installer finishes provisioning each node, a dispatcher script will execute on each provisioned node and convert the DHCP infinite lease to a static IP address using the same static IP address provided by the DHCP server.

1.3.2.13. Updates are immediately blocked if a machine config pool is degraded

If a machine config pool (MCP) is in a **degraded** state, the Machine Config Operator (MCO) now reports its **Upgradeable** status as **False**. As a result, you are now prevented from performing an update within a minor version, for example, from 4.7 to 4.8, until all machine config pools are healthy. Previously, with a degraded machine config pool, the Machine Config Operator did not report its **Upgradeable** status as **false**. The update was allowed and would eventually fail when updating the Machine Config Operator because of the degraded machine config pool. There is no change in this behavior for updates within z-stream releases, for example, from 4.7.1 to 4.7.2. As such, you should check the machine config pool status before performing a z-stream update.

1.3.3. Web console

1.3.3.1. Web console localization

The web console is now localized and provides language support for global users. English, Japanese, and Simplified Chinese are currently supported. The displayed language follows your browser preferences, but you can also select a language to override the browser default. From the **User** drop-down menu, select **Language preferences** to update your language setting. Localized date and time is now also supported.

1.3.3.2. Quick start tutorials

A quick start is a guided tutorial with user tasks. In the web console, you can access quick starts under the **Help** menu. They are especially useful for getting oriented with an application, Operator, or other product offering.

See Creating quick start tutorials in the web console for more information.

1.3.3.3. Insights plug-in

The Insights plug-in is now integrated into the OpenShift Container Platform web console. Insights provides cluster health data, such as the number of total issues and total risks of the issues. Risks are labeled as **Critical**, **Important**, **Moderate**, or **Low**. You can quickly navigate to Red Hat OpenShift Cluster Manager for further details about the issues and how to fix them.

1.3.3.4. Developer perspective

- The console now provides an extensibility mechanism that allows Red Hat Operators to build and package their own user interface extending the console. It also enables customers and Operators to add their own quick starts. Hints, filters, and access from both **Administrator** and **Developer** perspectives are now added to make quick starts and the relevant content more accessible.
- You can now quickly search for deployed workloads and application groupings in the topology **List** and **Graph** views to add them to your application.
- Persistent storage of user preferences is now provided so that when users move from one machine or browser to another they have a consistent experience.
- If you have the OpenShift GitOps Operator installed on your cluster, you can use the **Argo CD** link in the **Environments** view to navigate to the Argo CD user interface.
- Usability enhancements such as, the in-context menus mapping to the **Developer Catalog** features and **Form** or **YAML** options to update Pipelines, Helm, and Event Sources configurations have been added.
- Ability to see filtered entries is now added in the **Developer Catalog** for specified services such as Operator Backed, Helm, Builder Image, Template, and Event Source.
- After you have the Quay Container Security Operator installed on your cluster:
  - You can view a list of the following vulnerabilities for a selected project:
    - The total count of vulnerabilities and vulnerable images,
    - Severity-based counts of all vulnerable images,
    - Count of fixable vulnerabilities,
    - Number of affected pods for each vulnerable image
  - You can see the severity details of a vulnerability and also launch the Quay user interface, in the context of the manifest of the vulnerable image stored in that repository, to get more details about the vulnerability.
- After you have the OpenShift Virtualization Operator installed in your cluster, you can create virtual machines by selecting the **Virtual Machines** option on the **+Add** view and then using the templates in the **Developer Catalog**.

- The web terminal usability is now enhanced:
  - All users can access the web terminal on the console regardless of their privilege level.
  - When the web terminal is inactive for a long period, it stops and provides the user an option to restart it.
- The pipelines workflow is now enhanced:
  - The pipeline creation process now makes better use of pipelines over the default build config system. Build configs are no longer created by default along with the Pipelines using the **Import from git** workflow and the pipeline starts as soon as you create the application.
  - You can now configure pipelines in the **Pipeline builder** page using either the **Pipeline builder** option or the **YAML view** option. You can also use the Operator-installed, reusable snippets and samples to create detailed Pipelines.
  - The **PipelineRun** page now contains a **TaskRuns** tab that lists the associated task runs. You can click on the required task run to see the details of the task run and debug your pipelines.
  - You can now see the following metrics for your pipelines in the **Pipeline Details** page, per pipeline: pipeline run duration, task run duration, number of pipeline runs per day and the pipeline success ratio per day.
  - An **Events** tab is now available on the **Pipeline Run details** and the **Task Run details** pages, which shows the events for a particular PipelineRun or TaskRun.
- The serverless usability is now enhanced:
  - You can access the **Serving** and **Eventing** pages from the **Administrator** perspective and create serverless components using the console.
  - You can create Camel connectors using the event source creation workflow.
- The Helm charts usability is now enhanced.
  - As a cluster administrator, you can:
    - Add or remove Chart Repositories.
    - Remove the ability to use Helm charts.
    - Use the quick start to learn how to manage Helm Chart Repositories.
  - As a developer, you can:
    - See the name of the chart repository on the chart card in the catalog to distinguish charts with the same name, but from different chart repositories.
    - Get more insight into the charts at the catalog level on the cards.
    - Filter the catalog by chart repositories if multiple repositories are configured.

1.3.3.5. IBM Z and LinuxONE

With this release, IBM Z and LinuxONE are now compatible with OpenShift Container Platform 4.7. See Installing a cluster with z/VM on IBM Z and LinuxONE or Installing a cluster with z/VM on IBM Z and LinuxONE in a restricted network for installation instructions.

Notable Enhancements
The following new features are supported on IBM Z and LinuxONE with OpenShift Container Platform 4.7:

- KVM on RHEL 8.3 or later is supported as a hypervisor for user-provisioned installation of OpenShift Container Platform 4.7 on IBM Z and LinuxONE. See Installing a cluster with RHEL KVM on IBM Z and LinuxONE for installation instructions.
- Multipathing
- OpenShift Pipelines TP
- OpenShift Service Mesh
- OVN-Kubernetes with an initial installation of OpenShift Container Platform 4.7
- Persistent storage using Fibre Channel
- Persistent storage using Raw Block
- z/VM Emulated FBA devices on SCSI disks

Supported features
The following features are also supported on IBM Z and LinuxONE:

- CodeReady Workspaces
- Developer CLI - odo
- Persistent storage using iSCSI
- Persistent storage using local volumes (Local Storage Operator)

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:
  - Precision Time Protocol (PTP) hardware
  - CSI volume snapshots
- The following OpenShift Container Platform features are unsupported:
  - Log forwarding
  - OpenShift Virtualization
  - CodeReady Containers (CRC)
  - OpenShift 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 Serverless
    - Helm command-line interface (CLI) tool
    - Controlling overcommit and managing container density on nodes
    - etcd cluster Operator
    - CSI volume cloning
    - NVMe
    - 4K FCP block device
- Worker nodes must run Red Hat Enterprise Linux CoreOS (RHCOS).
- Persistent shared storage must be provisioned by using either NFS or other supported storage protocols
- Persistent non-shared storage must be provisioned using local storage, like iSCSI, FC, or using LSO with DASD, FCP, or EDEV/FBA.
- These features are available only for OpenShift Container Platform on IBM Z for 4.7:
    - HyperPAV enabled on IBM System Z /LinuxONE for the virtual machines for FICON attached ECKD storage

1.3.3.6. IBM Power Systems

With this release, IBM Power Systems are now compatible with OpenShift Container Platform 4.7. See Installing a cluster on IBM Power Systems or Installing a cluster on IBM Power Systems in a restricted network for installation instructions.

Notable Enhancements
The following new features are supported on IBM Power Systems with OpenShift Container Platform 4.7:

- Multipathing
- OpenShift Pipelines TP
- OpenShift Service Mesh
- OVN-Kubernetes with an initial installation of OpenShift Container Platform 4.7
- Persistent storage using Fibre Channel
- Persistent storage using Raw Block
- 4K Disk Support

Supported Features
The following features are also supported on IBM Power Systems:

- Currently, four Operators are supported:
  - Cluster-Logging-Operator
  - Cluster-NFD-Operator
  - Elastic Search-Operator
  - Local Storage Operator
- Developer CLI - odo
- CodeReady Workspaces
- Persistent storage using iSCSI
- HostPath

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

- The following OpenShift Container Platform features are unsupported:
  - OpenShift Metering
  - OpenShift Serverless
  - OpenShift Virtualization
  - CodeReady Containers (CRC)
- Worker nodes must run Red Hat Enterprise Linux CoreOS (RHCOS).
- Persistent storage must be of the Filesystem type that uses local volumes, Network File System (NFS), or Container Storage Interface (CSI)

1.3.4. Security and compliance

1.3.4.1. Managing user-owned OAuth access tokens

Users can now manage their own OAuth access tokens. This allows users to review their tokens and delete any tokens that have timed out or are no longer needed.

For more information, see Managing user-owned OAuth access tokens.

1.3.4.2. Cloud Credential Operator support for deletion of GCP root credentials after installation

You can now remove or rotate the GCP admin-level credential that the Cloud Credential Operator uses in Mint mode. This option requires the presence of the admin-level credential during installation, but the credential is not stored in the cluster permanently and does not need to be long-lived.

1.3.4.3. CIS Kubernetes Benchmark profile for the Compliance Operator

You can now use the Compliance Operator to perform Center for Internet Security (CIS) Kubernetes Benchmark checks. CIS profiles for OpenShift Container Platform are based on the CIS Kubernetes checks.

Until the CIS OpenShift Container Platform Benchmark is published, you can refer to the Red Hat OpenShift Container Platform Hardening Guide.

1.3.4.4. Secure Boot support for installer-provisioned clusters

You can now deploy a cluster with Secure Boot when using installer-provisioned infrastructure on bare metal nodes. Deploying a cluster with Secure Boot requires UEFI boot mode and Red Fish Virtual Media. You cannot use self-generated keys with Secure Boot.

1.3.4.5. Advanced Cluster Management 2.2 integration

Red Hat Advanced Cluster Management 2.2 now integrates with the Compliance Operator.

1.3.5. Networking

1.3.5.1. Expanded platform support for migrating from the OpenShift SDN cluster network provider to the OVN-Kubernetes cluster network provider

A migration to the OVN-Kubernetes cluster network provider is now supported on installer-provisioned clusters on the following platforms:

- Bare metal hardware
- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure
- Red Hat OpenStack Platform (RHOSP)
- VMware vSphere

1.3.5.2. Network connection health checks for API servers, load balancers, and nodes

To assist you with diagnosing cluster network connectivity issues, the Cluster Network Operator (CNO) now runs a connectivity check controller to perform connection health checks in your cluster. The results of the connection tests are available in **PodNetworkConnectivityCheck** objects in the **openshift-network-diagnostics** namespace. For more information, see Verifying connectivity to an endpoint .

1.3.5.3. OVN-Kubernetes egress firewall support for DNS rules

When configuring an egress firewall rule, you can now use a DNS domain name instead of an IP address. With the addition of DNS support in the OVN-Kubernetes cluster network provider egress firewall implementation, parity is achieved with the OpenShift SDN cluster network provider egress firewall implementation.

1.3.5.4. Library for interacting with SR-IOV virtual functions in DPDK mode within containers

For containers interacting with SR-IOV virtual functions (VFs) in Data Plane Development Kit (DPDK) mode, the **app-netutil** library now provides the following functions: **GetCPUInfo()**, **GetHugepages()**, and **GetInterfaces()**. For more information, see DPDK library for use with container applications.

1.3.5.5. Egress router CNI (Technology Preview)

The egress router CNI plug-in is introduced in Technology Preview. You can use the plug-in to deploy an egress router in redirect mode. This egress router provides parity for OVN-Kubernetes compared to OpenShift SDN, but for redirect mode only. The plug-in does not perform in HTTP proxy or DNS proxy modes, and this is a difference with the implementation for OpenShift SDN. For more information, see Deploying an egress router pod in redirect mode .

1.3.5.6. OVN-Kubernetes IPsec support for encrypted traffic between pods

When you install a cluster, you can configure the OVN-Kubernetes cluster network provider with IPsec enabled. With IPsec enabled, all cluster network traffic between pods is sent over an encrypted IPsec tunnel. You cannot enable or disable IPsec after cluster installation.

The IPsec tunnel is not used for network traffic between pods that are configured to use the host network. However, traffic sent from a pod on the host network and received by a pod that uses the cluster network does use the IPsec tunnel. For more information, see IPsec encryption configuration.

1.3.5.7. SR-IOV network node policy enhancement for deployment with Red Hat OpenStack Platform (RHOSP)

The SR-IOV Network Operator is enhanced to support an additional field, **spec.nicSelector.netFilter**, in the custom resource for an SR-IOV network node policy. You can use the new field to specify an RHOSP network by the network ID. For more information, see Configuring an SR-IOV network device.

1.3.5.8. RHOSP Kuryr support for services without pod selectors

Clusters that run on RHOSP and use Kuryr now support services that do not have pod selectors specified.

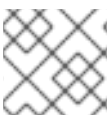
1.3.5.9. Adjusting HTTP header names

If legacy applications are sensitive to the capitalization of HTTP header names, use the Ingress Controller **spec.httpHeaders.headerNameCaseAdjustments** API field for a solution to accommodate legacy applications until they can be fixed.

OpenShift Container Platform will update to HAProxy 2.2, which down-cases HTTP header names by default, for example, changing **Host: xyz.com** to **host: xyz.com**. Make sure to add the necessary configuration by using **spec.httpHeaders.headerNameCaseAdjustments** before upgrading OpenShift Container Platform when HAProxy 2.2 is available.

1.3.5.10. Kubernetes NMState Operator (Technology Preview)

OpenShift Container Platform 4.7 provides post-installation state-driven network configuration on the secondary network interfaces of cluster nodes using the Kubernetes NMState Operator as a Technology Preview feature. For more information, see Using Kubernetes NMState (Technology Preview) .



NOTE

Configuration must occur before scheduling pods.

1.3.6. Storage

1.3.6.1. Persistent storage using CSI volume snapshots is generally available

You can use the Container Storage Interface (CSI) to create, restore, and delete a volume snapshot when using CSI drivers that provide support for volume snapshots. This feature was previously introduced as a Technology Preview feature in OpenShift Container Platform 4.4 and is now generally available and enabled by default in OpenShift Container Platform 4.7.

For more information, see Using CSI volume snapshots .

1.3.6.2. Persistent storage using the GCP PD CSI Driver Operator (Technology Preview)

The Google Cloud Platform (GCP) persistent disk (PD) CSI driver is automatically deployed and managed on GCP environments, allowing you to dynamically provision these volumes without having to install the driver manually. The GCP PD CSI Driver Operator that manages this driver is in Technology Preview.

For more information, see GCP PD CSI Driver Operator .

1.3.6.3. Persistent storage using the OpenStack Cinder CSI Driver Operator

You can now use CSI to provision a persistent volume using the CSI driver for OpenStack Cinder.

For more information, see OpenStack Cinder CSI Driver Operator .

1.3.6.4. vSphere Problem Detector Operator

The vSphere Problem Detector Operator periodically checks functionality of OpenShift Container Platform clusters installed in a vSphere environment. The vSphere Problem Detector Operator is installed by default by the Cluster Storage Operator, allowing you to quickly identify and troubleshoot common storage issues, such as configuration and permissions, on vSphere clusters.

1.3.6.5. Local Storage Operator now collects custom resources

The Local Storage Operator now includes a must-gather image, allowing you to collect custom resources specific to this Operator for diagnostic purposes. See **BZ#1756096** for more information.

1.3.7. Registry

1.3.7.1. Open Container Initiative images support

The OpenShift Container Platform internal registry and image streams now support Open Container Initiative (OCI) images. You can use OCI images in the same way you would use Docker **schema2** images.

1.3.7.2. New image stream metrics

The need to understand if clients are leveraging image stream imports using docker registry v1 protocol resulted in this enhancement, which exports Operator metrics to telemetry. Metrics related to protocol v1 usage are now visible in telemetry. See **BZ#1885856** for more information.

1.3.8. Operator lifecycle

1.3.8.1. Safe Operator upgrades

To make upgrades more robust, it is recommend that Operators actively communicate with the service that is about to be updated. If a service is processing a critical operation, such as live migrating virtual machines (VMs) in OpenShift Virtualization or restoring a database, it might be unsafe to upgrade the related Operator at that time.

In OpenShift Container Platform 4.7, Operators can take advantage of the new **OperatorCondition** resource to communicate a non-upgradeable state to Operator Lifecycle Manager (OLM), such as when a related service is performing a critical operation. The non-upgradeable state delays any pending Operator upgrade, whether automatically or manually approved, until the Operator finishes the operation and reports upgrade readiness.

See Operator conditions for more about how OLM uses this communication channel.

See Managing Operator conditions for details on overriding states in OLM as a cluster administrator.

See Enabling Operator conditions for details on updating your project as an Operator developer to use the communication channel.

1.3.8.2. Adding pull secrets to catalog sources

If certain images relevant to Operators managed by Operator Lifecycle Manager (OLM) are hosted in an authenticated container image registry, also known as a private registry, OLM and OperatorHub are unable to pull the images by default. To enable access, you can create a pull secret that contains the authentication credentials for the registry.

By referencing one or more secrets in a catalog source, some of these required images can be pulled for use in OperatorHub, while other images require updates to the global cluster pull secret or namespace-scoped secrets.

See Accessing images for Operators from private registries for more details.

1.3.8.3. Mirroring the content of an Operator catalog into a container image registry

Cluster administrators can use the **oc adm catalog mirror** command to mirror the content of an Operator catalog into a container image registry. This enhancement updates the **oc adm catalog mirror** command to also now mirror the index image being used for the operation into the registry, which was previously a separate step requiring the **oc image mirror** command. See BZ#1832968 for more information.

1.3.8.4. Creating new install plan for better experience

Deleting an **InstallPlan** object that is waiting for user approval causes the Operator to be stuck in an unrecoverable state as the Operator installation cannot be completed. This enhancement updates Operator Lifecycle Manager (OLM) to create a new install plan if the previously pending one is deleted. As a result, users can now approve the new install plan and proceed with the Operator installation. (BZ#1841175)

1.3.8.5. Mirroring images to a disconnected registry by first mirroring the images to local files

This enhancement updates the **oc adm catalog mirror** command to support mirroring images to a disconnected registry by first mirroring the images to local files. For example:

```
$ oc adm catalog mirror <source_registry>/<repository>/<index_image>:<tag> file:///local/index
```

Then you can move the local **v2/local/index** directory to a location within the disconnected network and mirror the local files to the disconnected registry:

```
$ oc adm catalog mirror file:///v2/local/index <disconnected_registry>/<repository>
```

See BZ#1841885 for more information.

1.3.9. Operator development

1.3.9.1. Operator SDK now fully supported

As of OpenShift Container Platform 4.7, the Operator SDK is now a fully supported Red Hat offering. With the downstream release of Operator SDK v1.3.0, officially supported and branded Operator SDK tooling is now available for download directly from Red Hat.

The Operator SDK CLI assists Operator developers and independent software vendor (ISV) partners in writing Operators that provide a great user experience and are compatible with OpenShift distributions and Operator Lifecycle Manager (OLM).

The Operator SDK enables Operator authors with cluster administrator access to a Kubernetes-based cluster, such as OpenShift Container Platform, to develop their own Operators based on Go, Ansible, or Helm. For Go-based Operators, Kubebuilder is embedded into the SDK as the scaffolding solution; this means existing Kubebuilder projects can be used as is with the SDK and continue to work.

The following features highlight some of the capabilities of the Operator SDK:

Native support for Operator Bundle Format

The Operator SDK includes native support for the Operator Bundle Format introduced in OpenShift Container Platform 4.6. All metadata required to package an Operator for OLM is generated automatically. Operator developers can use this functionality to package and test their Operator for OLM and OpenShift distributions directly from their CI pipelines.

Operator Lifecycle Manager integration

The Operator SDK provides developers with a streamlined experience for quickly testing their Operator with OLM from their workstation. You can use the **run bundle** subcommand to run Operator on a cluster and test whether the Operator behaves correctly when managed by OLM.

Webhook integration

The Operator SDK supports webhook integration with OLM, which simplifies installing Operators that have admission or custom resource definition (CRD) conversion webhooks. This feature relieves the cluster administrator of having to manually register the webhooks, add TLS certificates, and set up certificate rotation.

Validation scorecard

Operator authors should validate that their Operator is packaged correctly and free of syntax errors. To validate an Operator, the scorecard tool provided by the Operator SDK begins by creating all resources required by any related custom resources (CRs) and the Operator. The scorecard then creates a proxy container in the deployment of the Operator, which is used to record calls to the API server and run some of the tests. The tests performed also examine some of the parameters in the CRs.

Upgrade readiness reporting

Operator developers can use the Operator SDK to take advantage of code scaffolding support for Operator conditions, including reporting upgrade readiness to OLM.

Trigger Operator upgrades

You can quickly test upgrading your Operator by using OLM integration in the Operator SDK, without requiring you to manually manage index images and catalog sources. The **run bundle-upgrade** subcommand automates triggering an installed Operator to upgrade to a later version by specifying a bundle image for the later version.

NOTE

Operator SDK v1.3.0 supports Kubernetes 1.19.

See Developing Operators for full documentation on the Operator SDK.

1.3.10. Builds

Print the buildah version to the build log
In the current version, when the OpenShift Container Platform performs a build and the log level is five or higher, the cluster writes the buildah version information to the build log. This information helps Red Hat Engineering reproduce bug reports. Previously, this version information was not available in the build logs.

Cluster Samples Operator assistance for mirroring
OpenShift Container Platform now creates a config map named **imagestreamtag-to-image** in the **openshift-cluster-samples-operator** namespace that contains an entry, the populating image, for each image stream tag. You can use this config map as a reference for which images need to be mirrored for your image streams to import.

For more information, see Cluster Samples Operator assistance for mirroring .

1.3.11. Machine API

1.3.11.1. Machine sets running on AWS support Dedicated Instances

Machine sets running on AWS now support Dedicated Instances. Configure Dedicated Instances by specifying a dedicated tenancy under the **providerSpec** field in the machine set YAML file.

For more information, see Machine sets that deploy machines as Dedicated Instances .

1.3.11.2. Machine sets running on GCP support customer-managed encryption keys

You can now enable encryption with a customer-managed key for machine sets running on GCP. Users can configure an encryption key under the **providerSpec** field in the machine set YAML file. The key is used to encrypt the data encryption key, not to encrypt the customer's data.

For more information, see Enabling customer-managed encryption keys for a machine set .

1.3.11.3. Machine API components honor cluster-wide proxy settings

The Machine API now honors cluster-wide proxy settings. When a cluster-wide proxy is configured, all Machine API components will route traffic through the configured proxy.

1.3.11.4. Some machine configuration updates no longer cause automatic reboot

The Machine Config Operator (MCO) no longer automatically reboots all corresponding nodes for the following machine configuration changes:

- changes to the SSH key in the **spec.config.ignition.passwd.users.sshAuthorizedKeys** parameter of a machine config
- changes to the global pull secret or pull secret in the **openshift-config** namespace
- changes to the **/etc/containers/registries.conf** file, such as adding or editing an **ImageContentSourcePolicy** object

For more information, see Understanding the Machine Config Operator.

1.3.11.5. BareMetalHost API supports soft shutdown

In OpenShift Container Platform 4.6, when the online flag in the BareMetalHost API is set to **false**, the Bare Metal Operator shuts down nodes "hard." That is, it turns the power off without giving the operating system or workloads time to react. In OpenShift Container Platform 4.7 and subsequent releases, the API sends the node's operating system a signal telling it to shut down, and then waits for the node to power off in "soft" mode. If the operating system does not shut down the node within three minutes, the Bare Metal Operator executes a "hard" shutdown.

OpenShift Container Platform 4.8 will execute a "hard" shutdown for remediation purposes, such as if there is a known problem with the node. The behavior of executing a "hard" shutdown for remediation purposes will be back ported to OpenShift Container Platform 4.7.

1.3.12. Nodes

1.3.12.1. Descheduler is generally available

The descheduler is now generally available. The descheduler provides the ability to evict a running pod so that the pod can be rescheduled onto a more suitable node. You can enable one or more of the following descheduler profiles:

- **AffinityAndTaints**: evicts pods that violate inter-pod anti-affinity, node affinity, and node taints.
- **TopologyAndDuplicates**: evicts pods in an effort to evenly spread similar pods, or pods of the same topology domain, among nodes.
- **LifecycleAndUtilization**: evicts long-running pods and balances resource usage between nodes.

NOTE

With the GA, you can enable descheduler profiles and configure the descheduler interval. Any other settings that were available during Technology Preview are no longer available.

For more information, see Evicting pods using the descheduler.

1.3.12.2. Scheduler profiles (Technology Preview)

You can now specify a scheduler profile to control how pods are scheduled onto nodes. This is a replacement for configuring a scheduler policy. The following scheduler profiles are available:

- **LowNodeUtilization**: This profile attempts to spread pods evenly across nodes to get low resource usage per node.
- **HighNodeUtilization**: This profile attempts to place as many pods as possible onto as few nodes as possible, to minimize node count with high usage per node.
- **NoScoring**: This is a low-latency profile that strives for the quickest scheduling cycle by disabling all score plug-ins. This might sacrifice better scheduling decisions for faster ones.

For more information, see Scheduling pods using a scheduler profile .

1.3.12.3. Autoscaling for memory utilization GA

Autoscaling for memory utilization is now generally available. You can create horizontal pod autoscaler custom resources to automatically scale the pods associated with a deployment config or replication controller to maintain the average memory utilization you specify, either a direct value or a percentage of requested memory. For more information, see Creating a horizontal pod autoscaler object for memory utilization.

1.3.12.4. Autoscaling to zero machines for clusters on RHOSP

Clusters that run on RHOSP can now autoscale to zero machines.

1.3.12.5. Non-preempting option for priority classes (Technology Preview)

You can now configure a priority class to be non-preempting by setting the **preemptionPolicy** field to **Never**. Pods with this priority class setting are placed in the scheduling queue ahead of lower priority pods, but do not preempt other pods.

For more information, see Non-preempting priority classes.

1.3.12.6. Specifying CPUs for node host processes with CRI-O

CRI-O now supports specifying CPUs for node host processes (such as kubelet, CRI-O, and so forth). Using the **infra_ctr_cpuset** parameter in the **crio.conf** file allows you to reserve CPUs for the node host processes allowing OpenShift Container Platform pods that require guaranteed CPUs to operate without any other processes running on those CPUs. Pods that request guaranteed CPUs do not have to compete for CPU time with the node host process. See **BZ#1775444** for more information.

1.3.13. Red Hat OpenShift Logging

Cluster Logging becomes Red Hat OpenShift Logging
With this release, *Cluster Logging* becomes *Red Hat OpenShift Logging* , version 5.0. For more information, see Red Hat OpenShift Logging 5.0 release notes .

1.3.14. Monitoring

1.3.14.1. Alerting rule changes

OpenShift Container Platform 4.7 includes the following alerting rule changes:

> **Example 1.1. Alerting rule changes**

- The **AlertmanagerClusterCrashlooping** alert is added. The critical alert provides notification if at least half of the Alertmanager instances in a cluster are crashlooping.
- The **AlertmanagerClusterDown** alert is added. The critical alert provides notification if at least half of the Alertmanager instances in a cluster are down.
- The **AlertmanagerClusterFailedToSendAlerts** alert is added. The critical alert provides notification if all Alertmanager instances in a cluster failed to send notifications.
- The **AlertmanagerFailedToSendAlerts** alert is added. The warning alert provides notification if an Alertmanager instance failed to send notifications.
- The **etcdBackendQuotaLowSpace** alert is added. The critical alert provides notification if the database size of an etcd cluster exceeds the defined quota on an etcd instance.
- The **etcdExcessiveDatabaseGrowth** alert is added. The warning alert provides notification if there is an observed surge in etcd writes that caused a 50% increase in database size on an etcd instance over a four-hour period.
- The **etcdHighFsyncDurations** alert is added. The critical alert provides notification if the 99th percentile **fsync** durations of an etcd cluster are too high.
- The **KubeletClientCertificateRenewalErrors** alert is added. The warning alert provides notification if Kubelet failed to renew its client certificate.
- The **KubeletServerCertificateRenewalErrors** alert is added. The warning alert provides notification if Kubelet failed to renew its server certificate.
- The **NTODegraded** alert is added. The warning alert provides notification if the Node Tuning Operator is degraded.
- The **NTOPodsNotReady** alert is added. The warning alert provides notification if a specific pod on a node is not ready.
- The **PrometheusOperatorNotReady** alert is added. The warning alert provides notification if a Prometheus Operator instance is not ready.
- The **PrometheusOperatorRejectedResources** alert is added. The warning alert provides notification if specific resources are rejected by the Prometheus Operator.
- The **PrometheusOperatorSyncFailed** alert is added. The warning alert provides notification if the controller of a Prometheus Operator failed to reconcile specific objects.
- The **PrometheusTargetLimitHit** alert is added. The warning alert provides notification if Prometheus has dropped targets because some scrape configurations have exceeded the limit of the targets.
- The **ThanosSidecarPrometheusDown** alert is added. The critical alert provides notification that the Thanos sidecar cannot connect to Prometheus.
- The **ThanosSidecarUnhealthy** alert is added. The critical alert provides notification that the Thanos sidecar is unhealthy for a specified amount of time.
- The **NodeClockNotSynchronising** alert is updated to prevent false positives in environments that use the chrony time service, **chronyd**.

- The **NodeNetworkReceiveErrs** alert is updated to ensure that the alert does not fire when only a small number of errors are reported. The rule now uses the ratio of errors to total packets instead of the absolute number of errors.
- The **NodeNetworkTransmitErrs** alert is updated to ensure that the alert does not fire when only a small number of errors are reported. The rule now uses the ratio of errors to total packets instead of the absolute number of errors.
- The **etcdHighNumberOfFailedHTTPRequests** alerts with severities of **warning** and **critical** are removed. These alerts fired if a high percentage of HTTP requests failed on an etcd instance.

NOTE

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

1.3.14.2. Version updates to monitoring stack components and dependencies

OpenShift Container Platform 4.7 includes version updates to the following monitoring stack components and dependencies:

- The Prometheus Operator is now on version 0.44.1.
- Thanos is now on version 0.17.2.

1.3.14.3. AlertmanagerConfig CRD in Prometheus Operator not supported

Modifying Alertmanager configurations by using the **AlertmanagerConfig** custom resource definition (CRD) in the Prometheus Operator is not supported.

For more information, see Support considerations for monitoring.

1.3.14.4. New API Performance monitoring dashboard

The **API Performance** dashboard is now available from the web console. This dashboard can be used to help troubleshoot performance issues with the Kubernetes API server or the OpenShift API server. You can access the **API Performance** dashboard from the web console in the **Administrator** perspective by navigating to **Monitoring → Dashboards** and selecting the **API Performance** dashboard.

This dashboard provides API server metrics, such as:

- Request duration
- Request rate
- Request termination
- Requests in flight
- Requests aborted
- etcd request duration

- etcd object count
- Long-running requests
- Response status code
- Response size
- Priority and fairness

1.3.14.5. Namespace (Pods) and Pod Kubernetes networking dashboards are enabled in Grafana

The **Namespace (Pods)** and **Pod** Kubernetes networking dashboards are now enabled in Grafana. You can access the **Namespace (Pods)** and **Pod** dashboards from the web console in the **Administrator** perspective by navigating to **Monitoring → Dashboards → Grafana UI**.

These dashboards provide networking metrics, such as:

- Current rate of bytes received per namespace or per pod
- Current rate of bytes transmitted per namespace or per pod
- Bandwidth received
- Bandwidth transmitted
- Rate of received packets
- Rate of transmitted packets
- Rate of received packets dropped
- Rate of transmitted packets dropped

1.3.14.6. HWMon data collection for hardware telemetry for bare metal clusters

HWMon data collection is enabled for hardware health telemetry such as CPU temperature and fan speeds for bare metal clusters.

1.3.14.7. logLevel configuration field for Thanos Querier

You can now configure the Thanos Querier **logLevel** field for purposes such as debugging.

1.3.14.8. Removed memory limit on config-reloader container for monitoring user-defined projects

The memory limit was removed on the **config-reloader** container in the **openshift-user-workload-monitoring** namespace for Prometheus and Thanos Ruler pods. This update prevents OOM kill of the **config-reloader** container, which previously occurred when the container used more memory than the defined limit.

1.3.14.9. Removed deprecated Technology Preview configuration for monitoring your own services

The previous Technology Preview configuration that enabled users to monitor their own services is now removed and not supported in OpenShift Container Platform 4.7. The **techPreviewUserWorkload** field is removed from the **cluster-monitoring-config ConfigMap** object and is no longer supported.

See Understanding the monitoring stack for more information on monitoring user defined projects.

1.3.15. Scale

1.3.15.1. Cluster maximums

Updated guidance around cluster maximums for OpenShift Container Platform 4.7 is now available.

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

1.3.15.2. Test to determine CPU latency

The latency test, a part of the CNF-test container, provides a way to measure if the isolated CPU latency is below the requested upper bound.

For information about running a latency test, see Running the latency tests.

1.3.15.3. New globallyDisableIrqLoadBalancing feature in Performance Addon Operator allows global device interrupt processing to be disabled for guaranteed pod CPUs

The Performance Addon Operator manages host CPUs by dividing them into reserved CPUs for cluster and operating system housekeeping duties, and isolated CPUs for workloads. A new performance profile field **globallyDisableIrqLoadBalancing** is available to manage whether or not device interrupts are processed by the isolated CPU set.

New pod annotations **irq-load-balancing.crio.io** and **cpu-quota.crio.io** are used in conjunction with **globallyDisableIrqLoadBalancing** to define whether or not device interrupts are processed for a pod. When configured, CRI-O disables device interrupts only as long as the pod is running.

For more information, see Managing device interrupt processing for guaranteed pod isolated CPUs .

1.3.15.4. New VRF CNI plug-in allows secondary networks to be assigned to VRFs

A new VRF CNI plugin that allows you to assign additional networks to a VRF is now available. When you create a secondary network using a **rawConfig** configuration for the CNO custom resource and configure a VRF for it, the interface created for the pod is associated with the VRF. You can also use the VRF CNI plug-in to assign an SR-IOV network to a VRF.

For more information, see Assigning a secondary network to a VRF and Assigning an SR-IOV network to a VRF.

1.3.15.5. The xt_u32 end-to-end test is enabled for CNF

xt_u32 is an iptables kernel module that allows packet filtering based on arbitrary content. It can look beyond headers or special protocols that are not covered by other iptables modules.

For more information, see Performing end-to-end tests for platform verification .

1.3.16. Developer experience

1.3.16.1. Red Hat OpenShift GitOps (Technology Preview)

The Red Hat OpenShift GitOps 1.0 Technology Preview release introduces a declarative way to implement continuous deployment for cloud native applications. You can use Red Hat OpenShift GitOps to adopt GitOps principles for managing cluster configurations, and for automating secure, predictable, traceable, and repeatable application delivery across hybrid, multi-cluster, Kubernetes environments. It uses Argo CD as the core, and adds other tooling to enable teams to implement GitOps workflows across clusters. For more information, see, Understanding OpenShift GitOps .

1.3.17. Insights Operator

1.3.17.1. Insights Operator data collection enhancements

In OpenShift Container Platform 4.7, the Insights Operator collects the following additional information:

- The top 100 **InstallPlan** entries to identify invalid Operator Lifecycle Manager (OLM) installations
- The service accounts from the Kubernetes default namespace and the **openshift*** built-in namespaces
- The **ContainerRuntimeConfig** and **MachineConfigPools** configuration files to verify container storage limits
- The configuration files for all available **operator.openshift.io** control pane resources to identify Operators in unmanaged states
- The **NetNamespaces** names, including their **netID** and egress IP addresses
- A list of all installed Operator Lifecycle Manager Operators, including version information
- The Persistent Volume definition, if used in the **openshift-image-registry** configuration
- Appearances of certain log entries of pods in the **openshift-apiserver-operator** namespace
- Appearances of certain log entries of **sdn** pods in the **openshift-sdn** namespace

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

1.3.18. Authentication and authorization

1.3.18.1. Running OpenShift Container Platform using AWS Security Token Service (STS) for credentials (Technology Preview)

You can now configure the Cloud Credential Operator (CCO) to use the Amazon Web Services Security Token Service (AWS STS). When the CCO is configured to use STS, it assigns components IAM roles that provide short-term, limited-privilege security credentials.

For more information, see Support for Amazon Web Services Secure Token Service (AWS STS) .

1.4. NOTABLE TECHNICAL CHANGES

OpenShift Container Platform 4.7 introduces the following notable technical changes.

Operator Lifecycle Manager updated to use Kubernetes 1.20
Operator Lifecycle Manager (OLM) strives to keep up to date with Kubernetes releases when they become available. The OLM-provided **ClusterServiceVersion** (CSV) resource is composed of a number of core Kubernetes resources. When OLM increments Kubernetes dependencies, the embedded resources are updated as well.

As of OpenShift Container Platform 4.7, OLM and its associated components have been updated to use Kubernetes 1.20. Typically, Kubernetes is backwards compatible with a few of its previous versions. Operator authors are encouraged to keep their projects up to date to maintain compatibility and take advantage of updated resources.

See OpenShift Container Platform Versioning Policy for more specific details on backwards compatibility guarantees.

See Kubernetes documentation for details about version skew policies in the upstream Kubernetes project.

1.5. 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 4.7, 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 4.5	OCP 4.6	OCP 4.7
OperatorSource objects	DEP	REM	REM
Package Manifest Format (Operator Framework)	DEP	DEP	DEP
oc adm catalog build	DEP	DEP	DEP
--filter-by-os flag for **oc adm catalog mirror**	GA	GA	DEP
v1beta1 CRDs	DEP	DEP	DEP
Docker Registry v1 API	GA	DEP	DEP
Metering Operator	GA	DEP	DEP

Feature	OCP 4.5	OCP 4.6	OCP 4.7
Scheduler policy	GA	GA	DEP
ImageChangesInProgress condition for Cluster Samples Operator	GA	GA	DEP
MigrationInProgress condition for Cluster Samples Operator	GA	GA	DEP
Use of **v1** in **apiVersion** for OpenShift Container Platform resources	GA	GA	DEP
Bring your own RHEL 7 compute machines	GA	DEP	DEP

1.5.1. Deprecated features

1.5.1.1. Scheduler policy

Using a scheduler policy to control pod placement is deprecated and is planned for removal in a future release. For more information on the Technology Preview alternative, see Scheduling pods using a scheduler profile.

1.5.1.2. Catalog mirroring using filter-by-os flag

When using the **oc adm catalog mirror** command to mirror catalogs, the **--filter-by-os** flag was previously allowed to filter architectures of mirrored content. This would break references to those images in the catalog that point to the manifest list and not the manifest. The **--filter-by-os** flag now only filters the index image that is pulled and unpacked. To clarify this, the new **--index-filter-by-os** flag is now added and should be used instead.

The **--filter-by-os** flag is also now deprecated.

1.5.1.3. ImageChangesInProgress condition for Cluster Samples Operator

Image stream image imports are no longer tracked in real time by conditions on the Cluster Samples Operator configuration resource. In-progress image streams no longer directly affect updates to the **ClusterOperator** instance **openshift-samples**. Prolonged errors with image streams are now reported by Prometheus alerts.

1.5.1.4. MigrationInProgress condition for Cluster Samples Operator

Upgrade tracking is now achieved by the other conditions and both the individual image stream config maps and the **imagestream-to-image** config map.

1.5.1.5. Use of v1 for apiVersion for OpenShift Container Platform resources

Currently, **oc** fixes **apiVersion** in YAML or JSON resource files OpenShift Container Platform resources from **v1** to the correct value for the object. For example, **v1** is corrected to **apps.openshift.io/v1** for **DeploymentConfig** objects. This behavior is deprecated and is planned for

removal in a future release, and every resource that includes ***.openshift.io** must match the **apiVersion** value found in the API index.

This release adds a warning that displays the correct value of **apiVersion** when it is missing from an object.

```
Using non-groupfied API resources is deprecated and will be removed in a future release, update apiVersion to "apps.openshift.io/v1" for your resource
```

When you encounter this message, update your resource file to use the correct value.

1.5.2. Removed features

1.5.2.1. Installer-provisioned clusters no longer require provisioningHostIP or bootstrapProvisioningIP

When using installer-provisioned installation on bare metal nodes, OpenShift Container Platform 4.6 required providing two IP addresses from the **baremetal** network to the **provisioningHostIP** and **bootstrapProvisioningIP** configuration settings when deploying without a **provisioning** network. These IP addresses and configuration settings are no longer required in OpenShift Container Platform 4.7 when using installer provisioned infrastructure on bare metal nodes and deploying without a **provisioning** network.

1.5.2.2. Images removed from samples imagestreams

The following images are no longer included in the samples imagestreams provided with OpenShift Container Platform:

```
registry.redhat.io/ubi8/go-toolset:1.13.4
registry.redhat.io/rhdm-7/rhdm-decisioncentral-rhel8:7.8.1
registry.redhat.io/rhdm-7/rhdm-decisioncentral-rhel8:7.8.0
registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.8.1
registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.8.0
registry.redhat.io/rhpam-7/rhpam-businesscentral-monitoring-rhel8:7.8.1
registry.redhat.io/rhpam-7/rhpam-businesscentral-monitoring-rhel8:7.8.0
registry.redhat.io/rhpam-7/rhpam-businesscentral-rhel8:7.8.0
registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.8.1
registry.redhat.io/rhpam-7/rhpam-kieserver-rhel8:7.8.0
registry.redhat.io/rhpam-7/rhpam-smartrouter-rhel8:7.8.1
registry.redhat.io/rhpam-7/rhpam-smartrouter-rhel8:7.8.0
```

1.5.2.3. oc items removed

With this release, the following items that are used with **oc** are removed:

- The **--config** option.
- The **OC_EDITOR** environment variable.
- The **convert** subcommand.

1.6. BUG FIXES

api-server-auth

- Previously, the **openshift-service-ca** namespace was labeled with **openshift.io/run-level: 1**, which caused the pods in this namespace to run with extra privileges. This label has been removed, and now the pods in this namespace run with the appropriate privileges. (BZ#1806915)

- Previously, the **openshift-service-ca-operator** namespace was labeled with **openshift.io/run-level: 1**, which caused the pods in this namespace to run with extra privileges. This label has been removed for new installations, and now the pods in this namespace run with the appropriate privileges. For upgraded clusters, you can remove this label manually and restart the affected pods. (BZ#1806917)

- Previously, the configuration to scrape the OAuth API server pods in the **openshift-oauth-apiserver** namespace was missing, and metrics for the OAuth API server pods could not be queried in Prometheus. The missing configuration has been added, and OAuth API server metrics are now available in Prometheus. (BZ#1887428)

- Previously, a missed condition in the Cluster Authentication Operator code caused its log to be flooded with messages about updates to a deployment that did not occur. The logic for deciding whether to update the Operator status was updated and the Cluster Authentication Operator log no longer receives messages for a deployment update that did not occur. (BZ#1891758)

- Previously, the Cluster Authentication Operator only watched configuration resources named **cluster**, which caused the Operator to ignore changes in ingress configuration, which was named **default**. This led to incorrectly assuming that there were no schedulable worker nodes when ingress was configured with a custom node selector. The Cluster Authentication Operator now watches all resources regardless of their name, and the Operator now properly observes ingress configuration changes and reconciles worker node availability. (BZ#1893386)

Bare Metal Hardware Provisioning

- Previously on some systems, the installer would communicate with Ironic before it was ready and fail. This is now prevented. (BZ#1902653)

- Previously, when using virtual media on a Dell system, if the virtual media was already attached before the deployment commenced it would fail. Ironic now retries if this occurs. (BZ#1910739)

- Previously, master nodes were losing their IPv6 link-local address on the provisioning interface preventing provisioning from working with IPv6. A workaround has been added to **toggle addr_gen_mode** to prevent this from occurring. (BZ#1909682)

- Previously the **cluster-baremetal-operator** used the incorrect logging library. This issue resulted in command line arguments not being consistent with other Operators and not all Kubernetes library logs were getting logged. Switching the logging library has fixed this issue. (BZ#1906143)

- When using IPv6 on an interface, after a certain amount of time Network Manager removes the link-local IPv6 address. This issue led to PXE boot failures occurring for nodes after the IPv6 link-local address is removed. A workaround has been added to toggle the interface IPv6 addr_gen_mode which will cause the link-local address to be added back. (BZ#1901040)

- Previously Supermicro nodes boot to PXE upon reboot after successful deployment to disk. This issue is now fixed by always setting **BootSourceOverrideEnabled** when setting **BootSourceOverrideTarget**. Supermicro nodes now boot to disk persistently after deployment. (BZ#1918558)

- Service agent images shipped with **baremetal** IPI can now run on systems with UEFI secure boot enabled. Since network boot is not compatible with secure boot, using virtual media is required in this case. (BZ#1893648)

- Node auto-discovery is no longer enabled in **baremetal** IPI. It was not handled correctly and caused duplicate bare metal hosts registration. (BZ#1898517)

- Previously, the syslinux-nonlinux package was not included with bare metal provisioning images. As a result, virtual media installations on machines that used BIOS boot mode failed. The package is now included in the image. (BZ#1862608)

- Previously, certain Dell firmware versions reported the Redfish PowerState inaccurately. Updating Dell iDRAC firmware to version 4.22.00.53 resolves the issue. (BZ#1873305)

- Previously, Redfish was not present in the list of interfaces that can get and set BIOS configuration values. As a result, Redfish could not be used in BIOS configuration. Redfish is now included in the list, and it can be used in BIOS configuration. (BZ#1877105)

- Previously, the Redfish interface that is used to set BIOS configurations was not implemented properly. As a result, Dell iDRACs could not set BIOS configuration values. The implementation error was corrected. Now, the Redfish interface can set BIOS configurations.(BZ#1877924)

- Previously, differences in how Supermicro handles boot device settings through IPMI caused Supermicro nodes that use IPMI and UEFI to fail after an image was written to disk. Supermicro nodes are now passed an appropriate IPMI code to boot from disk. As a result, Supermicro nodes boot from disk correctly after deployment. (BZ#1885308)

- Bare metal installations on installer-provisioned infrastructure no longer silently skip writing an image when invalid root device hints are provided. (BZ#1886327)

- Previously, incomplete boot mode information for Supermicro nodes caused deployment by using Redfish to fail. That boot mode information is now included. As a result, Supermicro nodes can be deployed using Redfish. (BZ#1888072)

- The Ironic API service that is embedded in bare-metal installer-provisioned infrastructure now uses four workers instead of eight workers. As a result, RAM usage is reduced. (BZ#1894146)

Builds

- Previously, Dockerfile builds could not change permissions of the **/etc/pki/ca-trust** directory or create files inside it. This issue was caused by fixing BZ#1826183 in version 4.6, which added support for HTTPS proxies with CAs for builds and always mounted **/etc/pki/ca-trust**, which prevented builds that included their own CAs or modified the system trust store from working correctly at runtime. The current release fixes this issue by reverting Bug 1826183. Now, builder images that include their own CAs work again. (BZ#1891759)

- Previously, after upgrading from OpenShift Container Platform version 4.5 to version 4.6, running **git clone** from a private repository failed because builds did not add proxy information to the Git configuration that was used to pull the source code. As a result, the source code could not be pulled if the cluster used a global proxy and the source was pulled from a private Git repository. Now, Git is configured correctly when the cluster uses a global proxy and the **git clone** command can pull source code from a private Git repository if the cluster uses a global proxy. (BZ#1896446)

- Previously, the node pull secret feature did not work. Node pull secrets were not used if **forcePull: true** was set in the Source and Docker strategy builds. As a result, builds failed to pull images that required the cluster-wide pull secret. Now, node pull secrets are always merged

with user-provided pull secrets. As a result, builds can pull images when **forcePull: true** is set, and the source registry requires the cluster-wide pull secret. (BZ#1883803)

- Previously, OpenShift Container Platform builds failed on **git clone** when SCP-style SSH locations were specified because of Golang URL parsing, which does not accommodate Git SCP-styled SSH locations. As a result, OpenShift Container Platform builds and Source-to-Image (S2I) failed when those types of source URLs were supplied. Now, builds and S2I bypass Golang URL parsing and strip the **ssh://** prefix to accommodate Git SCP-styled SSH locations (BZ#1884270)

- Previously, build errors caused by invalid build pull secrets, whose auth keys were not base64-encoded, did not propagate through the build stack. As a result, determining the root cause of these errors was difficult. The current release fixes this issue, so these types of build errors propagate through the build stack. Now, determining the root cause of invalid build pull secret keys is easier for users. (BZ#1918879)

Cloud Compute

- Previously, the Machine API did not provide feedback to users when their credentials secret was invalid, thus making it difficult to diagnose when there were issues with the cloud provider credentials. Users are now warned if there is an issue with their credentials when creating or updating machine sets, for example if the credential secret does not exist or is in the wrong format. (BZ#1805639)

- Previously, the bare metal actuator deleted the underlying host by also deleting the **Machine** object, which is not the intended operation of the machine controller. This update sets the **InsufficientResourcesMachineError** error reason on machines when the search for a host is unsuccessful, and thus ensures that machines without a host are scaled down first. Machines are moved into the **Failed** phase if the host is deprovisioned. Now, a machine health check deletes failed machines and the **Machine** object is no longer automatically deleted. (BZ#1868104)

- Previously, when a machine entered a **Failed** state, the state of the cloud provider no longer reconciled. Thus, the machine status reported the cloud VM state as **Running** after it was possible to remove the VM. The machine status now more accurately reflects the observed state of the cloud VM as **Unknown** if the machine is in a **Failed** state. (BZ#1875598)

- Previously, several Machine API custom resource definitions contained broken links in the template schema description to corresponding reference documents. The links were updated to the correct upstream locations and are no longer broken. (BZ#1876469)

- Previously, the command **oc explain Provisioning** did not return the custom resource definition (CRD) description because an older version of the CRD definition was in use. The CRD version was updated, thus **oc explain** for the **Provisioning** CRD now returns the expected information. (BZ#1880787)

- Previously, when a user created or updated machines with a disk size less than the recommended minimum size, the machines failed to boot without warning when the disk size was too low. The disk size must be greater than the initial image size. The user is now notified with a warning that the disk size is low and that this might cause their machine or machine set to not start. (BZ#1882723)

- Previously, the state of a machine did not persist across reconciliation, thus the **Machine** object **instance-state** annotation and **providerStatus.instanceState** occasionally showed different values. Now, the machine state is replicated on the reconciled machine, and the **instance-state** annotation is consistent with the **providerStatus.instanceState** value. (BZ#1886848)

- Previously, machine sets running on Microsoft Azure in a disconnected environment failed to

boot and scale if the **publicIP** option was set to true in the **MachineSet** resource object. Now, to prevent machines from failing, users cannot create machine sets in disconnected environments with this invalid **publicIP** configuration. (BZ#1889620)

- Previously when creating a machine, only certain errors caused the **mapi_instance_create_failed** failure metric to update. Now, any error that occurs for machine creation appropriately increments the **mapi_instance_create_failed** metric. (BZ#1890456)

- Previously, the cluster autoscaler used a template node for node scaling decisions in certain circumstances. Occasionally, the **nodeAffinity** predicate failed to scale up as intended, and pending pods could not be scheduled. With this update, the template node includes as many labels as possible to ensure that the cluster autoscaler can scale up and pass node affinity checks. (BZ#1891551)

- Previously, the machine set default delete priority, which is **random**, did not prioritize nodes in the **Ready** state over nodes that were still building. As a result, especially when scaling a large number of machines, all nodes in the **Ready** state could potentially be deleted when scaling up a machine set and then immediately scaling down. This could also result in the cluster becoming unavailable. Now, a lower priority is assigned to machines that are not yet **Ready**. Thus, a large scale up of machines followed immediately by a scale down deletes machines that are still building before deleting machines that are running workloads. (BZ#1903733)

Cluster Version Operator

- Previously, a message in the installation and upgrade processes showed that the current process was 100% complete before it completed. This incorrect message was due to a rounding error. Now, the percentage is no longer rounded up, and the message shows both the number of finished subprocesses and an accurate percent compelte value. (BZ#1768255)

- Previously, the Cluster Version Operator (CVO) compared the pullspecs with the exact **available-update** and **current-target** values when it merged Cincinnati metadata like channel membership and errata URI. As a result, if you installed from or updated to mirrored release images that used valid alternative pullspecs, you did not receive Cincinnati metadata. Now, the CVO compares releases by digest and correctly associates Cincinnati metadata such as channel membership, regardless of which registry hosts the image. (BZ#1879976)

- Previously, a race condition with the metrics-serving goroutine sometimes caused the CVO become stuck on shutdown. As a result, CVO behavior like managed-object reconciliation and monitoring was not possible, and updates and installs might freeze. Now, the CVO times out after a few minutes, abandons any stuck metrics goroutines, and shuts down as intended. (BZ#1891143)

- Previously, some CVO log error messages did not render the variable for the type of changes that they were detecting correctly. Now, the variable is rendered correctly, and the error messages display as intended. (BZ#1921277)

CNF Platform Validation

- Previously, performing the end-to-end tests for platform validation results in an error for the SCTP validation step when a machine config does not include a config specification. This bug fix skips the SCTP test when the config specification is not found. (BZ#1889275)

- Previously, when the Performance Addon Operator ran the **hugepages** test on a host with two or more NUMA nodes and the performance profile requested huge pages distributed across the nodes, the test failed. This bug fix corrects how the **hugepages** test determines the number of huge pages for a NUMA node. (BZ#1889633)

config-operator

- Previously, the deprecated **status.platformStatus** field was not being populated during upgrade, in clusters upgraded since OpenShift Container Platform 4.1. As a consequence, the upgrade could fail. This fix modified the Cluster Config Operator to populate this field. As a result, the upgrade does not fail because of this field not being populated. (BZ#1890038)

Console Kubevirt Plugin

- Previously, the storage class was not propagating to the VM disk list from persistent volume claims for the **DataVolume** source. The storage class is now visible in the VM disk list of the web console. (BZ#1853352)

- Previously, imported SR-IOV networks could be set to different network interface types. With this fix, imported SR-IOV networks are now set only to the SR-IOV network interface type. (BZ#1862918)

- Previously, if a VM name was reused in the cluster, VM events displayed in the events screen were not correctly filtered and contained events mixed together from both VMs. Now, events are filtered properly and the events screen displays only the events belonging to the current VM. (BZ#1878701)

- Previously, the **V2VVMWare** and **OvirtProvider** objects created by the **VM Import** wizard were not cleaned up properly. Now, the **V2VVMWare** and **OvirtProvider** objects are removed as expected. (BZ#1881347)

- Previously, utilization data was not displayed for a Virtual Machine Interface (VMI) that did not have an associated VM. Now, if utilization data is available for a VMI, it is displayed. (BZ#1884654)

- Previously, when a PVC was cloned, its VM state was reported as **pending**, but additional information was not displayed. Now, when a PVC is cloned, the VM state is reported as **importing** along with a progress bar and additional info which contains a link to the pod or PVC. (BZ#1885138)

- Previously, the VM import status displayed an incorrect VM import provider. Now, the VM import status displays the correct VM import provider. (BZ#1886977)

- Previously, the default pod network interface type was to set to the wrong value. Now, the default pod network interface type is set to masquerade. (BZ#1887797)

Console Storage Plugin

- Previously, when the Local Storage Operator (LSO) was installed, the disks on a node were not displayed and there was no way to initiate a discovery of the disks on that node. Now, when the LSO is installed, the **Disk** tab is enabled and a **Discover Disks** option is available if a discovery is not already running. (BZ#1889724)

- With this update, the **Disk Mode** option has been renamed **Volume Mode**. (BZ#1920367)

Web console (Developer perspective)

- Previously, the user was denied access to pull images from other projects, due to insufficient user permissions. This bug fix removes all the user interface checks for role bindings and shows the **oc** command alert to help users use the command line. With this bug fix, the user is no longer blocked from creating images from different namespaces and is now able to deploy images from their other projects. (BZ#1894020)

- The console used a prior version of the **KafkaSource** object that used the **resources** and **service account** fields in their specification. The latest **v1beta1** version of the **KafkaSource** object removed these fields, due to which the user was unable to create the **KafkaSource** object with the v1beta1 version. This issue has been fixed now and the user is able to create the **KafkaSource** object with the v1beta1 version. (BZ#1892653)

- Previously, when you created an application using source code from Git repositories with the **.git** suffix, and then clicked the edit source code link, a **page not found** error was displayed. This fix removes the **.git** suffix from the repository URL and transforms the SSH URL to an HTTPS URL. The generated link now leads to the correct repository page. (BZ#1896296)

- Previously, the underlying **SinkBinding** resources were shown in the **Topology** view, along with the actual source created in the case of **Container Source** and **KameletBinding** resources, confusing users. This issue was fixed. Now, only the actual resource created for the event source is displayed in the **Topology** view, and the underlying **SinkBinding** resources, if created, are displayed in the sidebar. (BZ#1906685)

- Previously, when you installed the Serverless Operator, without creating the eventing custom resource, a channel card was displayed. When you clicked the card, a confusing alert message was displayed. This issue has now been fixed. The channel card, with a proper alert message, is now displayed only if the channel custom resource definition is present. (BZ#1909092)

- Previously, when you closed the web terminal connection, all the terminal output from that session disappeared. This issue has been fixed. The terminal output is now retained even after the session is closed. (BZ#1909067)

- Technology preview badges were displayed on the Eventing user interface although it had its GA release with OpenShift Container Platform 4.6. The Technology preview badges are now removed and the changes were back-ported to the OpenShift Container Platform 4.6.9 version. (BZ#1894810)

- Previously, volume mounts for deployments were not preserved if the deployment was edited using the console edit flows. The modified deployment YAML overwrote or removed the volume mounts in the pod template specification. This issue has been fixed. The volume mounts are now preserved even when the deployment is edited using the console edit flows. (BZ#1867965)

- In case of multiple triggers, one subscribing to Knative service and another to In Memory Channel as subscriber, the Knative resources were not displayed on the **Topology** view. This issue has been fixed now, so that the Knative data model returns proper data, and the Knative resources are displayed on the **Topology** view. (BZ#1906683)

- Previously, in a disconnected environment, the Helm charts were not displayed in the **Developer Catalog** due to an invalid configuration while fetching code. This issue has been fixed by ensuring that proxy environment variables are considered and the Helm charts are now displayed on the **Developer Catalog**. (BZ#1918748)

- While running a Pipeline, the log tab of the **TaskRun** resource displayed the string as **undefined** after the command in the output. This was caused due to some edge cases where some internal string operations printed **undefined** to the log output. This issue has been fixed now, and the pipeline log output does not drop empty lines from the log stream and does not print the string **undefined** any longer. (BZ#1915898)

- Previously, the **Port** list in the **Add** flow only provided options for exposed ports and did not allow you to specify a custom port. The list has now been replaced by a typeahead select menu, and now it is possible to specify a custom port while creating the application. (BZ#1881881)

- Previously, when conditional tasks failed, the completed pipeline runs showed a permanent pending task for each failed conditional task. This issue has been fixed by disabling the failed conditional tasks and by adding skipped icons to them. This gives a better picture of the state of the pipeline run. (BZ#1880389)

- Previously, the pod scale up or down buttons were available for a single pod resource, and the page crashed when the user pressed the scale button. This issue has been fixed by not showing the scale up or down buttons for a single pod resource. (BZ#1909678)

- Previously, the chart URL for downloading the chart to instantiate a helm release was unreachable. This happened because the **index.yaml** file from the remote repository, referenced in the Helm chart repository, was fetched and used as is. Some of these index files contained relative chart URLs. This issue has now been fixed by translating relative chart URLs to absolute URLs, which makes the chart URL reachable. (BZ#1912907)

- With Serverless 0.10, the latest supported versions were updated for **trigger**, **subscription**, **channel**, and **IMC**. Static models corresponding to each showed an API version of **beta**. The API version for eventing resources is now updated to **v1** and the UI now shows the latest supported version. (BZ#1890104)

- Previously, when the user switched between workloads on the **Monitoring Dashboard** tab, for example, from a specific deployment to **All workloads**, the dashboard displayed a white canvas and no chart. This issue has been fixed; the dashboard now displays charts when the user switches between workloads. (BZ#1911129)

- Previously, monitoring alerts with severity levels such as **critical** and **warning** were treated as **info** level alerts. As a result, the **Monitoring Alert** icon was not displayed on the workload in the **Topology** view for these alerts. This issue is now fixed; alerts like **critical** are treated as **warning** level alerts and a **Monitoring Alert** icon is displayed. (BZ#1925200)

- Previously, in the **YAML view** of the Helm installation form only the YAML code was shown. Now a **Schema** viewer is added in the YAML editor to show the schema and its description. (BZ#1886861)

- Previously, all Pods were failing with the **ErrImagePull** and **ImagePullBackOff** errors, even after an Image Pull Secret was added to access the external private image container registry. This is because the image download failed as it had no permissions for the external image registry and the cluster tried to load the container image directly from the external URL without the provided secret. As a result, the deployment was stuck until the service account or deployment was updated manually. Now, the issue is fixed and new deployments can start pods from the internal private container registry and import a container image from an external private container registry without any additional changes to the service account or deployment. (BZ#1924955)

- While creating a sample application, the **Developer** perspective creates multiple resources that depend on each other and must be completed in a specific order. Previously, the admission plug-in sometimes could not check one of these resources, preventing the **Developer** perspective from generating the sample application. This issue has been fixed. The code creates the resources in the required order, so creating a sample application is more stable. (BZ#1933665)

- Previously, the API server sometimes failed to create a resource and returned a 409 conflict response status code due to a conflict while updating a resource quota resource. This issue has been fixed. Now, if it receives a 409 status code, the OpenShift web console retries the request up to three times. If it continues to receive the 409 status code, the console displays an error message. (BZ#1928228)

DNS

- Previously, a cluster might experience intermittent DNS resolution errors because the **/etc/hosts** file on some nodes included invalid entries. With this release, DNS resolution no longer fails because of an **/etc/hosts** file with invalid entries. (BZ#1882485)

etcd

- Previously, the etcd readiness probe used **lsof** and **grep** commands, which could leave defunct processes. The etcd readiness probe now uses a TCP port probe, which is less expensive and does not create defunct processes. (BZ#1844727)

- Previously, when an IP address was changed on a control plane node, which causes the certificates on disk to be invalid, the etcd error messages were not clear why etcd was failing to connect with peers. An IP address change on a control plane node is now detected, an event is reported, and **EtcdCertSignerController** is marked as **Degraded**. (BZ#1882176)

- Previously, new static pod revisions could occur when the etcd cluster had less than three members, which caused temporary quorum loss. Static pod revisions are now avoided when all control plane nodes are not available, and these temporary quorum losses are avoided. (BZ#1892288)

- Previously, etcd backups included a recovery YAML file that was specific to the control plane node where the backup was taken from, so backups taken from one control plane node could not be restored on another control plane node. The recovery YAML file is now more generic so that the etcd backup can be restored on any control plane node. (BZ#1895509)

- Previously, the etcd backup script used the last modified timestamp to determine the latest revision, which caused the incorrect static pod resources to be stored in the etcd backup. The etcd backup script now uses the manifest file to determine the latest revision, and the correct static pod resources are now stored in the etcd backup. (BZ#1898954)

- Previously, the bootstrap rendering logic failed to detect a usable machine network CIDR when using IPv6 dual stack mode unless the IPv4 CIDR was the first element in the install-config machine network CIDR array. The parsing logic was fixed to loop through all machine network CIDRs, so the IPv4 address is now correctly loaded among the machine network CIDRs in dual stack mode. (BZ#1907872)

- Previously, if the **openshift-etcd** namespace was deleted, the **etcd-endpoints** config map was not recreated, and the cluster would not recover. The **etcd-endpoints** config map is now recreated if it is missing, allowing the cluster to recover. (BZ#1916853)

Image Registry

- The last Kubernetes update enforced a timeout on APIs. This timeout results in every long standing request being dropped after 34 seconds. When importing large repositories, specifically ones with several tags, the timeout is reached, not allowing the import to succeed as in previous versions. There is a flag to set a different timeout on **oc** client but there was not an example provided, making it difficult for the client to understand how to bypass the API timeout. Providing an example of the flag usage on **oc** help made things clear for the client, now it is easier to find this option. (BZ#1878022)

- Previously, using two distinct versions of the same logging package resulted in Operator logs being partially lost. This fix makes logging package versions equal, which means the upgraded logging package used by the Operator matches the one used by client-go. Now, logs are not lost. (BZ#1883502)

- Previously the pruner was trying to detect the registry name using image streams, but when there were no image streams the pruner failed to detect the registry name. With this fix, the Image Registry Operator provides the pruner with the registry name. Now, the pruner does not depend on the existence of image streams to detect the registry name. (BZ#1887010)

- Previously the Operator pod did not have memory requests, which in case of memory pressure on the node, the Operator could be killed because it was out of memory before other **BestEffort** containers. This fix added memory requests. Now, the Operator is not killed when it is out of memory if there are other **BestEffort** containers on the node. (BZ#1888118)

- Previously the pruner was trying to detect the registry name using image streams, but when there were no image streams the pruner failed to detect the registry name. With this fix, the Image Registry Operator provides the pruner with the registry name if the registry is configured or disables registry pruning if the registry is not installed. (BZ#1888494)

- Previously, there was a lack of analysis on Operand deployment status when defining the Operator status. This meant that in some scenarios the Image Registry Operator was presenting itself with two contradicting pieces of information. It was informing the user that it was not Available and at the same time not Degraded. These two conditions were still being presented even after the deployment stopped trying to get the image registry up and running. In this scenario the Degraded flag should be set by the Operator. By taking image registry deployment into account, the Operator now sets itself to Degraded if the Operand deployment reaches its progress deadline when trying to get the application running. Now, when the Deployment fails, after the progress deadline has been reached, the Operator sets itself to Degraded. The Operator still reports itself as Progressing while the Operator deployment is progressing. (BZ#1889921)

- Previously the Image Registry Operator did not use its entrypoint because an explicit command was provided. So a cluster-wide **trusted-ca** was not used by the Operator and the Operator could not connect to storage providers that do not work without custom **trusted-ca**. This fix removed the explicit command from the pod spec. Now, the image entrypoint is used by the container that applies **trusted-ca**. (BZ#1892799)

- Previously the default log level for the pruner was **2**. So when an error happened, the pruner was dumping stack trace. This fix changed the default log level to **1**. Now, only the error message is printed without stack traces. (BZ#1894677)

- Previously the **configs.imageregistry.operator.openshift.io** status field did not update during the Operator sync, which meant the status field was not presenting the most up to date applied swift configuration. With this fix, the sync process updates the **configs.imageregistry.operator.openshift.io** status to the spec values. The spec and status fields are in sync with the status field, presenting the applied configuration. BZ#1907202)

- Previously a lack of retries on a HTTP/2 protocol caused a related retryable error, which in turn caused mirroring to be cancelled with an error message. This fix added a retry when the error message corresponds to the HTTP/2 protocol related error. Now, for these errors, the mirror operation is cancelled after attempting multiple times. (BZ#1907421)

- Previously the absence of explicit user and group IDs on the **node-ca** daemon set confused the interpretation of what user and group were in use in the **node-ca** pods. This fix explicitly provides the **node-ca** daemon set with **runAsUser** and **runAsGroup** configuration. Now, there is a clear definition of user and group when inspecting the **node-ca** DaemonSet YAML file. (BZ#1914407)

ImageStreams

- Previously, the image pruner did not account for images that were used by **StatefulSet**, **Job**,

and **Cronjob** objects when it gathered lists of images that were in use. As a result, the wrong images could be pruned. The image pruner now accounts for images in use by these objects when it creates image lists. Images that are in use by these objects are no longer pruned. (BZ#1880068)

- Previously, newly created image streams were not decorated with **publicDockerImageRepository** values. Watchers did not receive **publicDockerImageRepository** values for new objects. Image streams are now decorated with the correct values. As a result, watchers get image streams with **publicDockerImageRepository** values. (BZ#1912590)

Insights Operator

- Previously, due to incorrect error handling, the Operator would end its process ambiguously when a file that it observed changed. Error handling for the Operator is improved. Now, the Operator continues to run and no longer sends an ending process signal when an observed file changes. (BZ#1884221)

- Previously, the Operator did not use the namespace of a resource while archiving reports. As a result, resources that had identical names in different namespaces were overwritten. The Operator now uses report paths in combination with namespaces while archiving data. As a result, all reports are collected for each namespace. (BZ#1886462)

Installer

- Previously when virtual-media was used, fast-track mode would not work as expected as nodes were rebooted between operations. This issue is now fixed. (BZ#1893546)

- Previously when using dual stack deployments, worker node host names did not match the name inspected before deployment causing nodes to need manual approval. This is now fixed. (BZ#1895909)

- Bare metal provisioning now does not fail if there is a small, up to one hour, clock skew between the control plane and a host being deployed. (BZ#1906448)

- When upper case letters were included in the vCenter host name, the OpenShift Container Platform installation program for VMware vSphere waited a long time for the cluster to complete before finally failing. The installation program now validates that the vCenter host name does not contain upper case letters early in the installation process, avoiding long wait times. (BZ#1874248)

- Previously, the internal Terraform backend for the OpenShift Container Platform installation program did not support large inputs from Terraform core to the Terraform provider, like Amazon Web Services (AWS). When the **bootstrap.ign** file was passed to the AWS provider as a string, the input limit could be exceeded, causing the installation program to fail when creating a bootstrap Ignition S3 bucket. This bug fix modifies the Terraform backend to pass the **bootstrap.ign** as a path on disk, allowing the AWS provider to read the large file by circumventing the input size limit. Now, the installation program succeeds when performing a Calico installation that creates the bootstrap Ignition file larger than the input limits. (BZ#1877116)

- Previously, pre-flight installer validation for Red Hat OpenStack Platform (RHOSP) was performed on the flavor metadata. This could prevent installations to flavors detected as **baremetal**, which might have the required capacity to complete the installation. This is usually caused by RHOSP administrators not setting the appropriate metadata on their bare metal flavors. Validations are now skipped on flavors detected as **baremetal**, to prevent incorrect failures from being reported. (BZ#1878900)

- Previously, the installation program did not allow the **Manual** credentials mode for clusters being installed to GCP and Azure. Because of this, users could not install their clusters to GCP or Azure using manual credentials. The installation program can now validate manual credentials provided for GCP and Azure. (BZ#1884691)

- Previously, the installation program could not verify that a resource group existed before destroying a cluster installed to Azure. This caused the installation program to continuously loop with errors. The installation program now verifies the resource group exists before destroying a cluster, allowing the cluster to be destroyed successfully. (BZ#1888378)

- Previously, the installation program did not check to ensure AWS accounts had **UnTagResources** permissions when creating a cluster with shared resources. Because of this, when destroying a cluster, the installation program did not have permission to delete tags added to the pre-existing network. This bug fix adds a permission check for **UnTagResources** when creating cluster with shared network resources to make sure the account has proper permissions before finishing the installation process. (BZ#1888464)

- For the **openshift-install destroy cluster** command to work properly, the cluster objects the installation program initially created must be removed. In some instances, the hosted zone object is already removed, causing the installation program to hang. The installation program now skips the removal of the object if the object has already been removed, allowing the cluster to successfully be destroyed. (BZ#1890228)

- Previously, the control plane ports on Red Hat OpenStack Platform (RHOSP) were not assigned the additional user-defined security groups. This caused the additional user-defined security group rules to not be applied properly to control plane nodes. The additional user-defined security groups are now assigned to the control plane ports, allowing the security group rules to correctly apply to the control plane nodes. (BZ#1899853)

- Previously, rules on the default AWS security group that sourced another security group prevented the installation program from deleting that other security group when destroying the cluster. This caused the cluster destroy process to never complete and left AWS resources remaining. The rules from the default security group are now deleted, unblocking the deletion of other security groups. This allows all AWS resources to be deleted from the cluster. (BZ#1903277)

- A missing guard in Red Hat OpenStack Platform (RHOSP) validations could fetch the list of subnets with an empty subnet ID, and cause some non-RHOSP clouds to return unexpected values. The unexpected error code would fail validation and prevent OpenShift Container Platform from installing on these non-RHOSP clouds. This bug fix adds the missing guard against the empty subnet ID, allowing for proper validations. (BZ#1906517)

- Previously, the reference load balancer for a user-provisioned infrastructure installation on VMware vSphere was configured for a simple TCP check, and the health checks did not consider the health of the api server. This configuration sometimes led to failed API requests whenever the API server restarted. Now, the health checks now verify API server health against the **/readyz** endpoint, and the reference API load balancer now handles requests during API server restarts gracefully. (BZ#1836017)

- Previously, when you pressed CTRL+C while using the installation program, the program was not always interrupted and did not always exit as expected. Now, when you press CTRL+C while using the installation program, the program always interrupts and exits. BZ#1855351)

- Previously, if you attempted to delete a cluster in Azure while using invalid credentials, such as when your service principal expired, and did not display the debug logs, it appeared that the cluster was deleted when it was not. In addition to not deleting the cluster, the locally stored cluster metadata was deleted, which made it impossible to remove the cluster by running the

openshift-install destroy cluster command again. Now, if you attempt to delete a cluster while using invalid Azure credentials, the installation program exits with an error, and you can update your credentials and try again. (BZ#1866925)

- Previously, the **install-config.yaml** file for the installer-provisioned infrastructure bare metal installation method incorrectly used the **provisioningHostIP** name instead of the **clusterProvisioningIP** name, which caused a disconnect between documentation and the actual field name used in the YAML file. Now, the **provisioningHostIP** field is deprecated in favor of **clusterProvisioningIP**, which removes the disconnect. (BZ#1868748)

- Previously, the installation program did not check for expired certificates in the Ignition configuration files. The expired certificates caused installation to fail without explanation. Now, the installation program checks for expired certificates and prints warning if certificates are expired. (BZ#1870728)

kube-apiserver

- Previously, the **preserveUnknownFields** field was set to **true** in **v1beta1** CRDs, and there was no error when **oc explain** did not explain CRD fields. A validation condition was added, and the status of **v1beta** CRDs without the **preserveUnknownFields** field set to **false** will show an error of **spec.preserveUnknownFields: Invalid value: true: must be false**. (BZ#1848358)

- Previously, the **LocalStorageCapacityIsolation** feature gate was disabled by default in OpenShift Container Platform on IBM Cloud clusters. When disabled, setting an ephemeral storage request or limit causes the pod to be unschedulable. This fix changed the code so that if the **LocalStorageCapacityIsolation** feature gate is disabled, ephemeral storage requests or limits are ignored and pods can be scheduled as expected. (BZ#1886294)

Red Hat OpenShift Logging

With this release, *Cluster Logging* becomes *Red Hat OpenShift Logging*, version 5.0. For more information, see Red Hat OpenShift Logging 5.0 release notes .

Machine Config Operator

- Previously, when deploying on Red Hat OpenStack Platform (RHOSP) and using an HTTP proxy with a host name, sometimes the installation process can fail to pull container images and report the error message **unable to pull image**. This bug fix corrects how the proxy is set in environment variables and nodes can pull container images from remote registries. (BZ#1873556)

- Previously, during an upgrade, the Machine Config Controller (MCC) for the previous release could react to a configuration change from the newer Machine Config Operator (MCO). The MMC then introduced another change that resulted in an unnecessary reboot during the upgrade process. This bug fix prevents the MCC from reacting to a configuration change from a newer MCO and avoids an unnecessary reboot. (BZ#1879099)

- Previously, the forward plugin for CoreDNS distributed queries randomly to all the configured DNS servers. Name resolution failed intermittently because CoreDNS would query a DNS server that was not functional. This bug fix sets the forward plugin to use the sequential policy so that queries are sent to the first DNS server that is responsive. (BZ#1882209)

- Previously, the Machine Config Operator was reading enabled systemd target units only from the **multi-user.target.wants** directroy. As a consequence, any unit that does not target the **multi-user.target.wats** directory was changed to target that directory. This fix modified the MCO to use the systemd-preset file to create a preset file in the MCO. As a result, all systemd services are enabled and disabled as expected. (BZ#1885365)

- Previously, when migrating a cluster to the OVN-Kubernetes default Container Network Interface (CNI), bond options on a pre-configured Linux bond interface. As a consequence, bonds are configured using round-robin instead of the mode specified and the bonds might not function. The ovs-configuration.service (configure-ovs.sh) was modified to copy all of the previous bond options on the Linux bond to **ovs-if-phys0** Network Manager connection. As a result, all bonds should work as originally configured. (BZ#1899350)

- In OpenShift Container Platform 4.6, a change was made to use the Budget Fair Queueing (BFQ) Linux I/O scheduler. As a consequence, there was an increased fsync I/O latency in etcd. This fix modified the I/O scheduler to use the mq-deadline scheduler, except for NVMe devices, which are configured to not use an I/O scheduler. For Red Hat Enterprise Linux CoreOS (RHCOS) updates, the BFQ scheduler is still used. As a result, latency times have been reduced to acceptable levels. (BZ#1899600)

Web console (Administrator perspective)

- Previously, an issue with a dependency resulted in the persistent unmounting and remounting of the **YAML Editor** in the OpenShift Container Platform web console. As a consequence, the YAML editor jumped to the top of the YAML file every few seconds This fix removed a default parameter value for the dependency. As a result, the **YAML Editor** behaves as expected. (BZ#1903164)

- Previously, a link in the Operator description in the OpenShift Container Platform web console was rendered in a sandboxed iframe, which disables javascript within that iframe. As a consequence: when user clicked the link, the sandbox limitations are inherited by the new tab, so JavaScript did not run the linked page. The links were fixed by adding an **allow-popups-to-escape-sandbox** parameter to Operator description iframe sandbox attribute, which opens new tabs outside of the sandbox. As a result, the link from Operator descriptions now open and run normally. (BZ#1905416)

- Previously, the scale pods function in the OpenShift Container Platform web console was not using the **scale** subresource, any custom role without the **patch** verb in the deployment config and deployment could not scale the pods in the web console. The fix changed the code so that the scale pods function is now using the **scale** subresource. As a result, users can scale pods in the web console without adding the **patch** verb. (BZ#1911307)

- Previously, creating a custom resource in the OpenShift Container Platform web console where a **fieldDependency** description was applied to a schema property that used a control field with an identical name the **getJSONSchemaPropertySortWeight** helper function would recurse infinitely. As a consequence, the **DynamicForm** component would throw an exception and the web browser could crash. This fix modified the **getJSONSchemaPropertySortWeight** helper function to keep track of the current path and use the entire path to determine dependency relationship instead of just the field names. As a result, the **DynamicForm** component no longer throws an exception under the above condition. (BZ#1913969)

- Previously, the **SamplesTBRInaccessibleOnBoot** alert description contained a misspelling of the word "bootstrapped". The alert description is now correct. (BZ#1914723)

- Previously, the CPU and Memory **specDescriptor** fields added an empty string in the YAML editor. Now, these fields no longer add an empty string in the YAML editor. (BZ#1797766)

- Previously, The **Subscription** and **CSV** objects were both displayed on the **Installed Operators** page during Operator installation. Now, this duplication has been fixed so that the **Subscription** Operator is not displayed on the **Installed Operators** page if a matching **CSV** object already exists. (BZ#1854567)

- Previously, empty resource utilization charts were displayed on the **Build details** page when a

build was started over an hour prior, but the default was set to display only the last hour. Now, the utilization charts on the **Build details** page shows data for the time that the build ran. (BZ#1856351)

- Previously, OpenAPI definitions were only updated on the initial page load. The OpenAPI definitions are now updated on a 5-minute interval and whenever the models are fetched from the API. OpenAPI definitions stay up to date without a page refersh. (BZ#1856354)

- In this release, the broken link to the cluster monitoring documentation has been fixed. (BZ#1856803)

- Previously, the **utm_source** parameter was missing from Red Hat Marketplace URLs. In this release, the **utm_source** parameter was added to Red Hat Marketplace URLs for attribution. (BZ#1874901)

- Previously, the project selection drop down could not be closed by using the **Escape** key. The handler for the **Escape** key is now updated and the user can exit and close the project selection drop down. (BZ#1874968)

- Previously, the font colors used for Scheduling Status was not in compliance with accessibility. The font and font colors were updated to be accessible. The scheduling disabled node is displayed in a yellow warning icon (exclamation icon). (BZ#1875516)

- Previously, the patch paths on some API calls were incorrect. This caused spec descriptor fields to not update resource properties. In this release, the logic for building a patch path from a descriptor was updated. (BZ#1876701)

- Previously, the **Unschedulable** status field only appeared when it was set to **True**. In this release, a new UX design was implemented to display status information more clearly. (BZ#1878301)

- Previously, subscriptions with an automatic approval strategy behave as if they have a manual approval strategy if another subscription in the same namespace has a manual approval strategy. In this release, an update was made to notify the user that a subscription with a manual approval strategy causes all subscriptions in the namespace to behave as manual. (BZ#1882653)

- Previously, a manual install plan can affect more than one Operator. However, the UI did not clearly indicate that is the case when it is true and presents the UI requesting approval. As a result, a user could be approving the install plan for multiple Operators, but the UI did not clearly communicate that. In this release, the UI lists all Operators affected by the manual approval plan and it clearly indicates which Operators will be installed. (BZ#1882660)

- Previously, creating a duplicate namepsace from the create namespace modal would result in a rejection error. In this release, we added an error handler for when you create projects and creating duplicate projects will not result in a rejection error. (BZ#1883563)

- Previously, the Prometheus swagger definition contained a **$ref** property that could not be resolved, so a runtime error occurred on the Prometheus operand creation form. Now, the **definitions** property is added to the schema that was returned by the **definitionFor** helper function, so the **$ref** resolves and no runtime error occurs.(BZ#1884613)

- Previously, users had to wait for the needed resources to load in the background before the install status page appears. Now, the install status page was updated so that it immediately appears once the user starts the Operator install. (BZ#1884664)

- Previously, iOS did not support connecting via secured Websocket with self-signed certificate,

so a white screen displayed on the console. Now, the connection falls back to https if the Websocket with self-assigned certificate is not successful, so the console loads properly. (BZ#1885343)

- Previously, system roles are not present when users create a new role binding in the web console. Now, system roles appears in the Role name dropdown, so users can now select a system role when creating a new role binding. (BZ#1886154)

- Previously, the terminal assumed all pods are Linux pods and did not account for Windows pods, so the terminal would not work with Windows pods as it defaulted to the sh command. Now, the terminal detects the pod type and changes the command as necessary. (BZ#1886524)

- Previously, new provisioners names did not contain the **kubernetes.io**/ prefix, so users could select the RWX and RWO access mode when creating PVC by aws-ebs-csi-driver(gp2-csi) in the web-console. Now, additional provisioners have been added to the AccessMode mapping, so RWX and RWO access modes are not available when creating PVC by aws-ebs-csi-driver(gp2-csi) in the web-console. (BZ#1887380)

- Previously, the logic for maintaining active Namespace didn't account for deleting the currently active namespace, so a namespace that was recently deleted in the UI could remain set as the currently active Namespace. Now, the active namespace logic has been updated so that, in a current browser session, it defaults to "All namespaces" when a user deletes the currently active namespace. So now when the user deletes the currently active Namespace, the active namespace in that same browser session is automatically updated to "All Namespaces". (BZ#1887465)

- Previously, the console vendor's 'runc' module in v0.1.1 contained a potential security issue, so frog xray flags the 'runc' dependency as a potential vulnerability. Now, the 'runc' module is pinned to the v1.0.0-rc8 version, which contains the fix, so the 'runc' dependency is no longer flagged as a potential vulnerability. (BZ#1887864)

- Previously, the CSV and PackageManifests listed every provided API version instead of just the latest version, so the CSV and PackageManifest pages could show duplicate APIs. Now, an update to the logic for retrieving APIs so that only the latest version of each provided API is displayed for each. (BZ#1888150)

- Previously, the Install Operand Form description was missing the 'SynchMarkdownView' component, so it is not formatted with markdown. Now, the Install Operand Form is formatted with markdown, so the Install Operand Form description is properly formatted.(BZ#1888036)

- Previously, the **fieldDependency specDescriptor** was not designed or tested with non-sibling dependencies. Consequently, non-sibling dependencies were not guaranteed to behave as expected. This update revises the logic to ensure that non-sibling dependencies behave as expected. (BZ#1890180)

- Previously, an exception was thrown if a local **ensureKind** function did not properly handle null **data** argument. This update adds null coalescence when using **data** argument to ensure that no exceptions are thrown, which allows graceful handling of null **data** arguments. (BZ#1892198)

- Previously, TLS secrets were not editable in the console. This update adds a **type** field so that TLS secrets can be updated in the console. (BZ#1893351)

- This update fixes an issue where the web console displayed incorrect filesystem capacity and usage data. (BZ#1893601)

- Previously, the web console was incorrectly granting permissions to the wrong service account, the Prometheus Operator, for scraping metrics for Operator Lifecycle Manager (OLM)

Operators. The console now correctly grants permissions to the prometheus-k8s service account, allowing metrics to be scraped. (BZ#1893724)

- Previously, the console pod's **TopologyKey** was set to **kubernetes.io/hostname**, which created availability problems during updates and zone outages. This update sets the **TopologyKey** to **topology.kubernetes.io/zone**, which improves availability during updates and zone outages. (BZ#1894216)

- Previously, an OperatorGroup with a missing **status** block in any namespace could cause a runtime error in the web console when installing a new Operator from OperatorHub. The problem has been resolved. (BZ#1895372)

- Previously, the console filtered out Custom Resource Definitions (CRDs) from the Provided APIs list if the model for the CRD did not exist. Consequently, the Details tab did not present Provided API cards upon initial install, which gave the impression that the Operator offered no APIs. This update removes the filter from the API cards so that they appear even if the model has yet to exist. As a result, the Provided API cards and their corresponding tabs always match, and the UI will no longer present an empty state if the models are not yet available. (BZ#1897354)

- In some cases, the lodash **startCase** function was being applied to the operand form descriptor field. Consequently, the field label would be formatted as Start Case, which would override the **displayName** property of the descriptor. This update applies **startCase** only when a descriptor **displayName** is not provided, which properly shows **displayName** on the operand form. (BZ#1898532)

- Previously, the **react-jsonschema-form** did not properly handle array type schemas that were explicitly set to null. If the form data passed to the DynamicForm component contained an array type property set to null, a runtime exception would occur. This update adds a null check in the array fields, ensuring that exceptions are no longer thrown in this scenario. (BZ#1901531)

Monitoring

- Previously, the **prometheus-adapter** did not implement an OpenAPI spec. As a result, the API server logged a message every 60 seconds that the OpenAPI did not exist while the Prometheus Adapter was deployed into the cluster. Additionally, the **KubeAPIErrorsHigh** alert might have fired due to the errors in the logs. This fix introduces the OpenAPI spec into **prometheus-adapter**, which complies with other core API resources within OpenShift Container Platform. (BZ#1819053)

- Previously, certain scenarios that elevated security context constraints (SCCs) caused Prometheus stateful set deployments to fail. Now, the **nonroot** SCC is used for stateful set deployments for monitoring. This fix requires the following configuration of Kubernetes security context settings for all monitoring stateful set deployments, which are Alertmanager, Prometheus, and Thanos Ruler:

```
securityContext:
  fsGroup: 65534 (1)
  runAsNonRoot: true
  runAsUser: 65534 (2)
```

(1) The filesystem group ID is set to the **nobody** user, ID **65534**. Kubelet recursively sets the group ID at pod startup. See the Kubernetes documentation for more information on configuring volume permission and ownership change policy for pods.

(2) All stateful set monitoring deployments run as the **nobody** user, ID **65534**.

(BZ#1868976)

- Previously, CPU steal time, which is the time that a virtual CPU waits for a real CPU while the hypervisor is servicing another virtual processor, impacted the metrics that reported CPU consumption. As a result, CPU usage could be reported as higher than the CPU count on a node. Now, the metrics that report CPU consumption do not take into account CPU steal time, and thus reported CPU usage accurately reflects the actual CPU usage. (BZ#1878766)

- Previously, authenticated requests without elevated permissions could access the **/api/v1/query** and **/api/v1/query_range** endpoints of Prometheus in user-defined projects. Thus, users with access to the token for a regular service account could read metrics from any monitored target. Now, **kube-rbac-proxy** is configured to allow requests to only the **/metrics** endpoint. Authenticated requests without cluster-wide permissions for the **/metrics** endpoint receive an HTTP 404 status code in response to a query to the **api/v1/query** and **/api/v1/query_range** endpoints. (BZ#1913386)

Networking

- The code in **ovn-kube** that detects the default gateway was not taking into consideration multipath environments. As a result, Kubernetes nodes failed to start because they could not find the default gateway. The logic has been modified to consider the first available gateway if multipath is present. OVN-Kubernetes now works in environments with multipath and multiple default gateways. (BZ#1914250)

- When deploying a cluster in dual stack mode OVN-Kubernetes was using the wrong source of truth.
  The OVN-Kubernetes master node performs an initial synchronization to keep OVN and Kubernetes system databases in sync. This issue resulted in race conditions on OVN-Kubernetes startup leading to some of the Kubernetes services becoming unreachable. Bootstrap logic deleted these services as they were considered orphans.

  This bug fix ensures Kubernetes is used as the source of truth. OVN-Kubernetes now starts correctly and keeps both OVN and Kubernetes in sync on startup. (BZ#1915295)

- When creating an additional network by specifying the **additionalNetworks** stanza in the Cluster Network Operator (CNO) configuration object, the CNO manages the lifecycle for the NetworkAttachmentDefinition object that is created. However, that object was never deleted if the CNO configuration was updated to exclude the additional network from the **additionalNetworks** stanza. In this release, the CNO now deletes all objects related to the additional network. (BZ#1755586)

- For the OVN-Kubernetes cluster network provider, if an egress IP address was configured and one of the nodes hosting the egress IP address became unreachable, any egress IP addresses assigned to the unreachable node were never reassigned to other nodes. In this release, if a node hosting an egress IP address is found to be unreachable, the egress IP address is assigned to another node. (BZ#1877273)

- For the OVN-Kubernetes cluster network provider, the route priority of the **br-ex** bridge could be superseded by the default route for a secondary network interface added after installing the cluster. When the default route for the secondary device supersedes the br-ex bridge on a node, the cluster network no longer functions. In this release, the default route for **br-ex** bridge cannot be superseded. (BZ#1880259)

- For clusters using the OVN-Kubernetes cluster network provider, when adding a Red Hat Enterprise Linux (RHEL) 7 worker node to the cluster, the new worker node was unable to connect to the cluster network. In this release, you can now add RHEL worker nodes successfully. (BZ#1882667)

- For clusters using the OVN-Kubernetes cluster network provider, it was not possible to use a VLAN or bonded network device as the default gateway on a node. In this release, OVN-Kubernetes now works with these network devices. (BZ#1884628)

- For clusters using the Kuryr cluster network provider, unnecessary Neutron ports were created for pods using on the host network. In this release, Neutron ports are no longer created for host network pods. (BZ#1886871)

- For clusters using the OVN-Kubernetes cluster network provider, the **br-ex** bridge did not support the attachment of other interfaces, such as **veth<N>** pairs, and any interface added to the bridge did not function correctly. In this release, new interfaces can be attached to the **br-ex** interface and function correctly. (BZ#1887456)

- For clusters using the OVN-Kubernetes cluster network provider, if an ExternalIP address was configured, any node in the cluster not configured to use that IP address did not route traffic sent to the externalIP correctly. Now, every node in the cluster is configured with the necessary routes for an ExternalIP. (BZ#1890270)

- For clusters using the OpenShift SDN cluster network provider, the order in which a namespace and a network namespace were deleted mattered. If the NetNamespace object associated with a Namespace object were deleted first, it was not possible to recreate that network namespace again. In this release, a namespace and its associated network namespace may be deleted in any order. (BZ#1892376)

- For clusters using the OpenShift SDN cluster network provider, previously the network provider logged the following message: **unable to allocate netid 1**. Because this message is harmless for any NETID less than **10**, in this release OpenShift SDN no longer emits the message for any NETID less than **10**. (BZ#1897073)

- If the cluster is using the OVN-Kubernetes cluster network provider, all inbound ICMPv6 was erroneously sent to both the node and OVN. In this release, only ICMPv6 Neighbor Advertisements and Route Advertisements are sent to both the host and OVN. As a result, a ping sent to a node in the cluster no longer results in duplicate responses. (BZ#1897641)

- Previously, in a cluster with a large number of nodes, excessive multicast DNS (mDNS) traffic was generated. As a result network switches might overflow. This release limits mDNS queries to once per second.

- Previously, creating an additional network attachment that used IPv6, the Whereabouts CNI plug-in, and specified excluded subnet ranges would ignore the excluded subnet ranges. This bug fix corrects the plug-in so that subnet ranges can be excluded. (BZ#1900835)

- Previously, under certain circumstances, pods did not terminate due to an error condition with Multus. Multus includes the message **failed to destroy network for pod sandbox** in logs when the problem occurs. This bug fix makes Multus tolerate a deleted cache file and pods can terminate. (BZ#1900835)

- Previously, when using the OpenShift SDN network provider with network policies, it was possible for for pods to experience network connectivity problems even in namespaces that do not use network policies. This bug fix ensures that the underlying Open vSwitch (OVS) flows that implement the network policy are valid. (BZ#1914284)

- Previously, when using the OVN-Kubernetes network provider and using multiple pods to serve as external gateways, scaling down the pods prevented other pods in the namespace from routing traffic to the remaining external gateways. Instead, traffic was routed to the default gateway of the node. This bug fix enables the pods to continue routing traffic to the remaining external gateways. (BZ#1917605)

Node

- Previously clusters under load can timeout if pod or container creation requests take too long. The kubelet attempts to re-request that resource even though CRI-O is still working on creating that resource, causing the requests to fail with the *name is reserved* error. After CRI-O finishes the original request, it notices the request timed out, and cleans up the failed pod/container, starting the process over. As a consequence, pod and container creation can stall and multiple *name is reserved* errors are reported by the kubelet. This also causes an already overloaded node to be further overloaded. This fix modified CRI-O to save the progress of any pod or container creation that timeout due to system load. CRI-O also stalls new requests from the kubelet so there are fewer *name is reserved* errors. As a result, when clusters are under load, CRI-O slows the kubelet and reduces the load on the cluster. The overall load on the node is reduced and Kubelet and CRI-O should reconcile more quickly. (BZ#1785399)

- Previously, deep directories in volumes cause long SELinux relabeling times. As a consequence, container creation requests can timeout, and the kubelet attempts to re-request that resource, causing the *error reserving ctr name* or *Kubelet may be retrying requests that are timing out in CRI-O due to system load* error. This fix modified CRI-O to save the progress of any pod or container creation that timeout due to system load. As a result, the container request are fulfilled in a timely manner. (BZ#1806000)

- Previously, CRI-O used only IPv4 iptables for managing the host port mapping. As a consequence: The host port does not work for IPv6. This fix modified CRI-O to support IPv6 host ports. As a result, host ports function with IPv6 as expected. (BZ#1872128)

- Previously, HTTP/2 transports did not have the correct options attached to the connections that provide timeout logic, which caused VMWare network interfaces (and other scenarios) to blip for a few seconds causing connections to fail silently. As a consequence, connections lingered, which caused other related failures, such as nodes not being detected as down, API calls using stale connections and failing, and so forth. This fix added proper timeouts. As a result, HTTP/2 connections within the system are more reliable, and side-effects are mitigated. (BZ#1873114)

- Previously, the Topology Manager end-to-end test (**openshift-tests run-test**) required the Machine Config Daemon (MCD) to be running on each worker node, which is the case for nodes deployed on Red Hat Enterprise Linux CoreOS (RHCOS) nodes but not for nodes deployed on Red Hat Enterprise Linux (RHEL). As a consequence, the Topology Manager end-to-end test incorrectly failed with a false-negative when running against clusters deployed on RHEL. This fix modified the test to skip any nodes where it does not detect an MCD. As a result, the false-negative failures are no longer reported. (BZ#1887509)

- Previously, the Kubelet did not handle transitions properly when statuses were missing. As a consequence some terminated pods did not get restarted. This fix fix added a **failed** container status to allow the container to be restarted as needed. As a result, kubelet pod handling does not result in an invalid state transition. (BZ#1888041)

- Previously, machine metrics from **cAdvisor** were missing in Kubernetes 1.19 and later. This fix modified the code to properly collect the **CAdvisor** machine metrics. As a result, the machine meterics are present. (BZ#1913096)

- Previously, the Horizontal Pod Autoscaler (HPA) ignored pods with incomplete metrics, such as pods that have init containers. As a consequence, any pod with an init container would not be scaled. This fix makes the Prometheus Adapter send complete metrics for init containers. As a result, HPA can scale pods with init containers. (BZ#1867477)

- Previously, the Vertical Pod Autoscaler (VPA) did not have access to monitor deployment configs. As a consequence, the VPA was unable to scale deployment config workloads. This fix

added the appropriate permissions to the VPA to monitor deployment configs. As a result, the VPA can scale deployment config workloads. (BZ#1885213)

Node Tuning Operator

- When an invalid Tuned profile is created, the **openshift-tuned** supervisor process may ignore future profile updates and fail to apply the updated profile. This bug fix keeps state information about Tuned profile application success or failure. Now, **openshift-tuned** recovers from profile application failures on receiving new valid profiles. (BZ#1919970)

oauth-proxy

- Previously, there was legacy logging of a failed authentication check. Requests to services behind the oauth-proxy could cause a line written to the proxy log, which would cause log flood. This fix removed the uninformative log line from the proxy. Now, the proxy no longer experiences log spam. (BZ#1879878)

- Previously, invalid option handling caused a nil dereference when incorrect option combinations were specified with the **oauth-proxy** command. This resulted in a segmentation fault stack trace being output at the end of the usage message. The option handling is now improved and nil dereferences do not occur when incorrect option combinations are specified. The usage message is output without a stack track when incorrect options are now specified. (BZ#1884565)

oc

- Previously, changes in logging libraries caused goroutine stack traces to be printed even at a low log level of 2, which made debugging more difficult. The log level for goroutine stack traces was increased, and now they will only be printed at log level 6 and above. (BZ#1867518)

- Previously, users logging in with the OpenShift CLI (**oc**) to multiple clusters using the same user name had to log in to each cluster every time. The context name has been properly updated so that it is unique even when the user name is the same. Now, after logging in and switching context, it is not necessary to log in again. (BZ#1868384)

- Previously, when a release was mirrored to disk using **oc adm release mirror**, the manifest file names did not contain the architecture extension, for example **-x86_64**. This did not allow for mirroring multiple architectures to the same repository without having tag name collisions. File names now contain the appropriate architecture extension, which prevents tag name collisions. (BZ#1878972)

- Previously, an image verifier object was not set properly which could cause the OpenShift CLI (**oc**) to fail with a nil pointer exception when verifying images. The image verifier object is now set properly and the OpenShift CLI (**oc**) no longer fails with a nil pointer exception when verifying images. (BZ#1885170)

- Previously, the wrong user name was used when verifying image signatures using **oc adm verify-image-signature**, and image signature verification failed. The proper user name is now used when verifying image signatures and image signature verification now works as expected. (BZ#1890671)

- Previously, metadata providing version information was not produced during the build process and was not present on Windows binaries of the OpenShift CLI (**oc**). Proper Windows version information is now generated and available on Windows binaries. (BZ#1891555)

- Previously, a missing nil check for a route condition could cause the OpenShift CLI (**oc**) to crash when describing a route. A nil check was added and describing a route now works properly. (BZ#1893645)

- Previously, the OpenShift CLI (**oc**) had a low limit for client throttling, and the requests reaching for API discovery were limited by the client code. The client throttling limit was increased and client-side throttling should now appear less frequently. (BZ#1899575)

- Previously, support for init containers was lost during changes to the **oc debug** command, and it was not possible to debug init containers. Support for init containers was added to the **oc debug** command, and it is now possible to debug init containers. (BZ#1909289)

OLM

- The Marketplace Operator was written to report that the services it offered were degraded whenever the **marketplace-operator** pod exited gracefully, which would happen during routine cluster upgrades. This caused the pod to report as degraded during normal upgrades, which caused confusion. The Marketplace Operator no longer reports that it is degraded when it exits gracefully and is no longer flagged by the Telemeter client as degraded. (BZ#1838352)

- Previously during an Operator upgrade, Operator Lifecycle Manager (OLM) deleted existing cluster service versions (CSVs) before the upgrade was completed. This caused the new CSV to be stuck in a "Pending" state. This bug fix updates OLM to check the ownership of the service account to ensure the new service account is created for the new CSV. As a result, existing CSVs are no longer deleted until the new CSV reaches the "Succeeded" state correctly. (BZ#1857877)

- Previously, Operator Lifecycle Manager (OLM) would accept a **Subscription** object that specified a channel that did not exist. The subscription would appear to succeed, and no related error message was presented, which caused user confusion. This bug fix updates OLM to cause **Subscription** objects to fail in this scenario. Cluster administrators can review events in the **default** namespace for dependency resolution failure information, for example:

  ```
  $ oc get event -n default
  ```

  Example output

  ```
  LAST SEEN   TYPE      REASON           OBJECT                  MESSAGE
  6m22s       Warning   ResolutionFailed   namespace/my-namespace   constraints not satisfiable: my-operator is mandatory, my-operator has a dependency without any candidates to satisfy it
  ```

  (BZ#1873030)

- Previously, support for admission webhook configurations in Operator Lifecycle Manager (OLM) reused the CA certificate generation code used when deploying API servers. The mounting directory used by this code placed the certificate information at the following locations:

  - **/apiserver.local.config/certificates/apiserver.crt**
  - **/apiserver.local.config/certificates/apiserver.key**

  However, admission webhooks built using Kubebuilder or the Operator SDK expect the CA certificates to be mounted in the following locations:

  - **/tmp/k8s-webhook-server/serving-certs/tls.cert**
  - **/tmp/k8s-webhook-server/serving-certs/tls.key**

  This mismatch caused the webhooks to fail to run. This bug fix updates OLM to now mount the webhook CA certificates at the default locations expected by webhooks built with Kubebuilder or the Operator SDK. As a result, webhooks built with Kubebuilder or the Operator SDK can now be deployed by OLM. (BZ#1879248)

- When deploying an Operator with an API service, conversion webhook, or an admission webhook, Operator Lifecycle Manager (OLM) should retrieve the CA from an existing resource to calculate a CA hash annotation. This annotation influences a deployment hash that OLM relies on to confirm that the deployment is installed correctly. OLM currently does not retrieve the CA from conversion webhooks, resulting in an invalid deployment hash, which causes OLM to attempt to reinstall the cluster service version (CSV).
  If a CSV defines a conversion webhook but does not include an API service or an admission webhook, the CSV cycles through the "Pending", "ReadyToInstall", and "Installing" phases indefinitely. This bug fix updates OLM to use the existing conversion webhook to retrieve the value of the CA and correctly calculate the deployment hash. As a result, OLM can now install CSVs that define a conversion webhook without an API service or admission webhook. (BZ#1885398)

- In the **opm** command, the **semver-skippatch** mode previously allowed only bundles with later patch versions as valid replacements, ignoring any pre-release versions. Bundles with the same patch versions but later pre-release versions were not accepted as replacements. This bug fix updates the **opm** command to base the **semver-skippatch** check on the semantic versioning as a whole instead of just the patch version. As a result, later pre-release versions are now valid for the **semver-skippatch** mode. (BZ#1889721)

- Previously, the Marketplace Operator did not clean stale services during a cluster upgrade, and Operator Lifecycle Manager (OLM) accepted the stale service without validating the service. This caused the stale service to direct traffic to a catalog source pod that contained outdated content. This bug fix updates OLM to add spec hash information to the service and check to ensure the service has the correct spec by comparing the hash information. OLM then deletes and recreates the service if it is stale. As a result, the service spec now directs traffic to the correct catalog source pod. (BZ#1891995)

- After mirroring an Operator to a disconnected registry, the Operator install could fail due to a missing related bundle image. This issue was due to the bundle image not being present in the **index.db** database. This bug fix updates the **opm** command to ensure the bundle image is present in the **related_images** table of the database. (BZ#1895367)

- Previously, Operator authors could create cluster service versions (CSVs) that defined webhooks with container ports set outside of the **1** to **65535** range. This prevented the **ValidatingWebhookConfiguration** or **MutatingWebhookConfiguration** objects from being created because of a validation failure; CSVs could be created that never successfully installed. The custom resource definition (CRD) validation for CSVs now includes the proper minimum and maximum values for the **webhookDescription ContainerPort** field. This now defaults to **443** if the container port is not defined. CSVs with invalid container ports now fail validation before the CSV is created. (BZ#1891898)

- Stranded Operator image bundles that were not referenced by any channel entries remained after an **opm index prune** operation. This lead to unexpected index images being mirrored. Stranded image bundles are now removed when an index is pruned and the unexpected images are not included when the Operator catalog is later mirrored. (BZ#1904297)

- Previously, Operator updates could result in Operator pods being deployed before a new service

account was created. The pod could be deployed by using the existing service account and would fail to start with insufficient permissions. A check has been added to verify that a new service account exists before the cluster service version (CSV) is moved from a **Pending** to **Installing** state. If a new service account does not exist, the CSV remains in a **Pending** state which prevents the deployment from being updated. (BZ#1905299)

- Previously, when Operator Lifecycle Manager (OLM) copied a **ClusterServiceVersion** (CSV) object to multiple target namespaces, the **.status.lastUpdateTime** field in the copied CSV was set to the current time. If the current time was later than the last update time of the original CSV, a synchronization race condition was triggered where the copied CSV never converged to match the original. This was more likely to occur when many namespaces were present in a cluster. Now, the original **.status.lastUpdateTime** timestamp is preserved in the copied CSVs and the synchronization race condition is not triggered by a difference between the **.status.lastUpdateTime** values. (BZ#1905599)

- Previously, pod templates defined in the **StrategyDetailsDeployment** objects of a **ClusterServiceVersion** (CSV) object could include pod annotations that do not match those defined in the CSV. Operator Lifecycle Manager (OLM) would fail to install the Operator because the annotations in the CSV are expected to be present on the pods deployed as part of the CSV. The pod template annotations defined in the **StrategyDetailsDeployment** objects are now overwritten by those defined in the CSV. OLM no longer fails to deploy CSVs whose annotations conflict with those defined in the pod template. (BZ#1907381)

- When a default catalog source in the **openshift-marketplace** namespace is disabled through the OperatorHub API, you can create a custom catalog source with the same name as that default. Previously, custom catalog sources with the same name as a default catalog source were deleted by the Marketplace Operator when the marketplace was restarted. An annotation has been added to the default catalog sources that are created by the Marketplace Operator. Now, the Operator only deletes the catalog sources that contain the annotation when the marketplace is restarted. Custom catalog sources created with the same name as the default catalog sources are not deleted. (BZ#1908431)

- Previously, the **oc adm catalog mirror** command did not generate the proper mappings for Operator index images without namespaces. Additionally, the **--filter-by-os** option filtered the entire manifest list. This resulted in invalid references to the filtered images in the catalog. Index images without namespaces are now mapped correctly and an **--index-filter-by-os** option is added to filter only the index image that is pulled and unpacked. The **oc adm catalog mirror** command now generates valid mappings for index images without namespaces and the **--index-filter-by-os** option creates valid references to the filtered images. (BZ#1908565)

- Previously, Operators could specify a **skipRange** in the cluster service version (CSV) replacement chain that would cause Operator Lifecycle Manager (OLM) to attempt to update the Operator with itself. This infinite loop would cause an increase in CPU usage. The CSV replacement chain is now updated so that Operators do not become stuck in an infinite loop due to an invalid **skipRange**. (BZ#1916021)

- Previously, the **csv.status.LastUpdateTime** time comparison in the cluster service version (CSV) reconciliation loop always returned a **false** result. This caused the Operator Lifecycle Manager (OLM) Operator to continuously update the CSV object and trigger another reconciliation event. The time comparison is now improved and the CSV is no longer updated when there are no status changes. (BZ#1917537)

- Catalog update pods with polling intervals that were multiples of 15 greater than the default 15 minute resynchronization period would be continuously reconciled by the Catalog Operator. This would continue until the next poll time was reached, causing increased CPU load. The reconciliation requeuing logic is now improved so that the continuous reconciliation and the associated CPU load increases do not occur. (BZ#1920526)

- Previously, if no matching Operators were found during an attempt to create an Operator subscription, the constraints listed in the resolution failure event contained internal terminology. The subscription constraint strings did not describe the resolution failure reason from a user perspective. The constraint strings are now more meaningful. (BZ#1921954)

openshift-apiserver

- Previously, requests targeting the **deploymentconfigs/<name>/instantiate** subresource failed with **no kind "DeploymentConfig" is registered for version apps.openshift.io/**. The correct version for the **DeploymentConfig** is now set and these requests no longer fail. (BZ#1867380)

Operator SDK

- Previously, all **operator-sdk** subcommands attempted to read the **PROJECT** file, even if **PROJECT** was a directory. As a result, subcommands that did not require the **PROJECT** file failed. Now, subcommands that do not require the **PROJECT** file do not attempt to read it and succeed even if an invalid **PROJECT** file is present. (BZ#1873007)

- Previously, running the **operator-sdk cleanup** command did not clean up Operators that were deployed with the **operator-sdk run bundle** command. Instead, an error message was displayed and the Operator was not cleaned up. Now, the **operator-sdk cleanup** command has been updated, and Operators deployed with **run bundle** can be cleaned up by using the **cleanup** command. (BZ#1883422)

Performance Addon Operator

- Previously, incorrect wait in the must-gather logic resulted in early termination of log gathering. This issue resulted in, depending on timing, the log gathering operation being interrupted prematurely. This led to a partial log collection. This is now fixed by adding the correct wait in the must-gather logic. (BZ#1906355)

- Previously, must-gather collected an unbounded amount of kubelet logs on all nodes. This issue resulted in an excessive amount of data being transferred and collected, with no clear benefit for the user.
  This issue is fixed by collecting a bounded amount, the last eight hours, of kubelet logs only on worker nodes and not collecting kubelet logs on the control plane nodes. (BZ#1918691)

- Previously, when the machine config pool was degraded, the performance profile was not updated to display an accurate machine config pool state. Now, the performance profile node selector or machine config pool selector correctly watches the relevant machine config pools, and a degraded machine config pool reflects the correct status. (BZ#1903820)

RHCOS

- Previously, configuring additional Azure disks during RHCOS installation caused a failure because the **udev** rules for Azure disks were missing from the RHCOS initramfs. The necessary **udev** rules have been added so that configuring additional disks during installation now works properly. (BZ#1756173)

- Previously, the **rhcos-growpart.service** was being used in a way that was not a best practice. Now, the **rhcos-growpart.service** has been removed in favor of configuring disks via Ignition at installation time. To change disk configuration after initial RHCOS installation, you must reprovision your systems with the necessary disk configuration changes. (BZ#1851103)

- Previously, the Machine Config Operator would attempt to rollback rpm-ostree changes when running **rpm-ostree cleanup -p**, causing a "System transaction in progress" error to occur. This fix improves rpm-ostree code related to D-Bus handling so that the error no longer occurs.

(BZ#1865839)

- Previously, there was no support in ppc64le or s390x for NVME emulation in KVM in RHEL 8.2, which caused the **kola --basic-qemu-scenarios** using NVME emulation to fail. The tests for NVME emulation on ppc64le and s390x have been disabled so that the tests now succeed. (BZ#1866445)

- Previously, Ignition could not fetch a remote configuration over the network when the DHCP server took too long to respond to DHCP queries because NetworkManager would stop waiting for a DHCP answer and the network would not be configured in the initramfs. The new version of NetworkManager now understands the **rd.net.timeout.dhcp=xyz** and **rd.net.dhcp.retry=xyz** options when set as kernel parameters to increase the timeout and number of retries, allowing you to set those options to account for delayed DHCP answers. (BZ#1877740)

- Previously, an incorrect networking configuration was created because multiple **nameserver=** entries on the kernel command line could create multiple NetworkManager connection profiles. A newer version of NetworkManager in RHCOS now correctly handles multiple **nameserver=** entries so that networking configuration is properly generated when multiple **nameserver=** entries are provided. (BZ#1882781)

- Previously, a node process would seg fault due to a recursive call that was overflowing the stack. This logic error has been fixed so that there are no longer seg faults. (BZ#1884739)

- Previously, network-related service units were not strictly ordered, which sometimes meant that network configurations copied using **-copy-network** did not take effect on the first reboot into the installed system. The ordering of the relevant service units has been fixed so that they now always take effect on the first reboot. (BZ#1895979)

- Previously, when the **coreos-installer** command invoked fdasd to check for a valid DASD label on s390x, udev would reprobe the DASD device, causing the DASD formatting to fail because udev was still accessing the device. Now, after checking for a DASD label, **coreos-installer** waits for udev to finish processing the DASD to ensure that the DASD formatting is successful. (BZ#1900699)

- Previously, it could be confusing to query and modify connection settings in NetworkManager when using DHCP because a single NetworkManager connection was created by default that matched all interfaces. The user experience has been improved so that when using DHCP, NetworkManager now creates a separate connection for each interface by default. (BZ#1901517)

- Previously, failure to properly tear down network interfaces in the initrd before switching to the real root might cause static IP assignment to a VLAN interface to not be successfully activated in the real root. This fix changes how network interfaces are torn down in the initrd so that static IP assignments to VLAN interfaces are successfully activated in the real root. (BZ#1902584)

- Previously, if you had configured RHCOS to use dhclient for DHCP operations, you were left with systems that could not properly acquire a DHCP address because the dhclient binary was removed from RHCOS when the switch to using NetworkManager in the initramfs was made. The dhsclient binary is now included in RHCOS so that RHCOS systems can successfully perform DHCP operations using dhclient. (BZ#1908462)

- Previously, upgraded nodes would not receive uniquely generated initiator names because the service unit that regenerates the iSCSI initiator name only worked on first boot. With this fix, the service unit now runs on every boot so that upgraded nodes receive generated initiator names if one does not already exist. (BZ#1908830)

- Previously, you could not create ext4 filesystems with Ignition because **mkfs.ext4** failed when **/etc/mke2fs.conf** did not exist. With this fix, **/etc/mke2fs.conf** has been added to the initramfs so that Ignition successfully creates ext4 filesystems. (BZ#1916382)

Routing

- Previously, it was possible to set the **haproxy.router.openshift.io/timeout** annotation on a route with a value that exceeded 25 days. Values greater than 25 days caused the ingress controller to fail. This bug fix sets an upper limit of 25 days for the timeout. (BZ#1861383)

- Previously, an ingress controller would report a status of Available even if DNS was not provisioned or a required load balancer was not ready. This bug fix adds validation to the Ingress Operator to ensure that DNS is provisioned and the load balancer, if required, is ready before the ingress controller is reported as available. (BZ#1870373)

- Previously, it was possible to set the default certificate for an ingress controller to a secret that does not exist, such as by entering a typographical error. This bug fix adds validation to ensure the secret exists before changing the default certificate. (BZ#1887441)

- Previously, a route with a name that is longer than 63 characters could be created. However, after the route was created, it failed validation. This bug fix adds validation when the route is created. (BZ#1896977)

Storage

- Previously, the admission plug-in would add a failover domain and region labels, even when they were not configured properly, causing pods that used statically provisioned persistent volumes (PVs) to fail to start on OpenStack clusters with an empty region in the configuration. With this fix, the tables are now added to the PV only when they contain a valid region and failure domain so that pods using statically provisioned PVs behave the same as dynamically provisioned PVs on OpenStack clusters that have been configured with an empty region or failure domain. (BZ#1877681)

- Previously, the **LocalVolumeDiscoveryResult** object was displayed in the web console, implying that these could be manually defined. With this fix, the **LocalVolumeDiscoveryResult** type has been flagged as an internal object and is no longer displayed in the web console. To view local disks, navigate to **Compute → Nodes → Select Nodes → Disks** instead. (BZ#1886973)

- Previously when creating snapshots that require credentials, force deletion would not work for snapshots if the **VolumeSnapshotClass** CRD was already deleted. Now, instead of relying on the **VolumeSnapshotClass** CRD to exist, the credentials are fetched from the **VolumeSnapshotContent** CRD so that volume snapshots and volume snapshot contents that use credentials can be deleted provided the secret containing these credentials continues to exist. (BZ#1893739)

- Previously, the Kubernetes FibreChannel (FC) volume plug-in did not properly flush a multipath device before deleting it, and in rare cases, a filesystem on a multipath FC device was corrupted during pod destruction. Now, Kubernetes flushes data before deleting a FC multipath device to prevent filesystem corruption. (BZ#1903346)

Scale

- The **nosmt** additional kernel argument that configures hyperthreading was previously undocumented for use with OpenShift Container Platform. To disable hyperthreading, create a performance profile that is appropriate for your hardware and topology, and then set **nosmt** as an additional kernel argument.

For more information, see About hyperthreading for low latency and real-time applications .

1.7. 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 4.5	OCP 4.6	OCP 4.7
Precision Time Protocol (PTP)	TP	TP	TP
oc CLI Plug-ins	TP	TP	TP
Descheduler	TP	TP	GA
OVN-Kubernetes Pod network provider	TP	GA	GA
HPA custom metrics adapter based on Prometheus	TP	TP	TP
HPA for memory utilization	TP	TP	GA
Service Binding	TP	TP	GA
Log forwarding	TP	GA	GA
Monitoring for user-defined projects	TP	GA	GA
Raw Block with Cinder	TP	TP	TP
External provisioner for AWS EFS	TP	TP	TP
CSI volume snapshots	TP	TP	GA
CSI volume cloning	TP	GA	GA
CSI volume expansion	TP	TP	TP

Feature	OCP 4.5	OCP 4.6	OCP 4.7
vSphere Problem Detector Operator	-	-	GA
CSI GCP PD Driver Operator	-	-	TP
CSI OpenStack Cinder Driver Operator	-	-	TP
CSI AWS EBS Driver Operator	TP	TP	TP
Red Hat Virtualization (oVirt) CSI Driver Operator	-	GA	GA
CSI inline ephemeral volumes	TP	TP	TP
Automatic device discovery and provisioning with Local Storage Operator	-	TP	TP
OpenShift Pipelines	TP	TP	GA
OpenShift GitOps	-	TP	GA
Vertical Pod Autoscaler	TP	TP	TP
Operator API	TP	GA	GA
Adding kernel modules to nodes	TP	TP	TP
Egress router CNI plug-in	-	-	TP
Scheduler profiles	-	-	TP
Non-preempting priority classes	-	-	TP
Kubernetes NMState Operator	-	-	TP
Assisted Installer	-	-	TP

1.8. KNOWN ISSUES

- 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 4.7, 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)

- If you are running cluster monitoring with an attached PVC for Prometheus, you might experience OOM kills during upgrade to OpenShift Container Platform 4.7. When persistent storage is in use for Prometheus, Prometheus memory usage doubles during cluster upgrade and for several hours after upgrade is complete. To avoid the OOM kill issue, allow worker nodes with double the size of memory that was available prior to the upgrade. (BZ#1925061)

- Starting and stopping pods rapidly can result in pods getting stuck in the **Terminating** state. As a workaround, you must remove the stuck pod by running the following command:

```
$ oc delete --force -n <project_name> <pod_name>
```

This issue will be fixed in a future release of OpenShift Container Platform. (BZ#1929463)

- 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 4.7. (BZ#1887007)

- It is currently not possible to use the AWS Secure Token Service (STS), which is a Technology

Preview feature, in a cluster installed into the AWS C2S Secret Region due to current OpenShift Container Platform limitations. This will be fixed in a future release of OpenShift Container Platform. (BZ#1927157)

- Installing a cluster into the AWS C2S Secret Region using your own infrastructure based on Red Hat's recommended CloudFormation templates does not work due to issues with creating the bootstrap nodes during the installation process. (BZ#1924080)

- Upgrading Performance Addon Operator from 4.6 to 4.7 fails with the error:

  > "Warning TooManyOperatorGroups 11m operator-lifecycle-manager csv created in namespace with multiple operatorgroups, can't pick one automatically"

  Before upgrading, follow the procedure as described in Upgrading Performance Addon Operator when previously installed to a specific namespace.

  (BZ#1913826)

- A reboot is sometimes required to enact SR-IOV changes on supported NICs. SR-IOV currently issues the reboot when it is ready. If this reboot coincides with changes in the Machine Config policy, the node can be left in an undetermined state. The Machine Config Operator assumes that the updated policy has been applied when it has not.

  NOTE

  This race condition can also be caused by adding a node to a Machine Config Pool that has MCP and SR-IOV changes.

  To avoid this issue, new nodes requiring MCO and SR-IOV changes should be completed sequentially. First, apply all MCO configuration and wait for the nodes to settle. Then, apply the SR-IOV configuration.

  If a new node is being added to a Machine Config Pool that includes SR-IOV, this issue can be avoided by removing the SR-IOV policy from the Machine Config Pool and then adding the new worker. Then, re-apply the SR-IOV policy.

  (BZ#1921321)

- The **stalld** service triggers a bug in the kernel, which results in the node freezing. In order to work around this issue, the Performance Addon Operator disables **stalld** by default. The fix impacts latency associated with DPDK based workloads, however the functionality will be restored once the kernel bug (BZ#1912118) is fixed.

- Fluentd pods with the **ruby-kafka-1.1.0** and **fluent-plugin-kafka-0.13.1** gems are not compatible with Apache Kafka version 0.10.1.0. For more information, see "Known issues" in the Red Hat OpenShift Logging 5.0 release notes.

- Precision Time Protocol (PTP) faults are observed on the Mellanox MT27800 Family [ConnectX-5] of adapter cards. In the **ptp4l** log, errors are observed which disturb clock synchronization.
  These errors result in larger than normal system clock updates due to the NIC hardware clock resetting. The root cause of this issue is unknown and no workaround currently exists.

  (BZ#1913279)

- Previously, a bug in the OpenStack SDK caused a failure when requesting server group **OSP16**.

Consequently, the UPI playbook **control-plane.yaml** fails during the task to create the control plane server. As a temporary workaround, you can request a hotfix to update the OpenStack SDK, which updates the OpenStack SDK on the bastion host to execute UPI Ansible tasks to at least **python-openstacksdk-0.36.4-1.20201113235938.el8ost**. With this hotfix, the playbook successfully runs. (BZ#1891816)

- When attempting an IPI installation on bare metal using the latest Dell firmware (04.40.00.00) nodes will not be deployed and an error is displayed in their status. This is due to Dell Firmware (4.40.00.00) using eHTML5 as the Virtual Console Plug-in.
To work around this issue, change the Virtual Console Plugin to HTML5 and run the deployment again. The nodes should now be successfully deployed. For more information, see Firmware requirements for installing with virtual media.

  (BZ#1915828)

- Installing a cluster on RHOSP that uses Kuryr times out with the following messages during bootstrapping:

  ```
  INFO Waiting up to 20m0s for the Kubernetes API at https://api.ostest.shiftstack.com:6443...
  INFO API v1.20.0+ba45583 up
  INFO Waiting up to 30m0s for bootstrapping to complete...
  ERROR Attempted to gather ClusterOperator status after wait failure: listing ClusterOperator objects: Get
  "https://api.ostest.shiftstack.com:6443/apis/config.openshift.io/v1/clusteroperators": dial tcp 10.46.44.166:6443: connect: connection refused
  INFO Use the following commands to gather logs from the cluster
  INFO openshift-install gather bootstrap --help
  FATAL failed to wait for bootstrapping to complete: timed out waiting for the condition
  ```

  The timeout is caused by changes in how Kuryr detects the RHOSP Networking service (neutron) subnet of the cluster's nodes.

  As a workaround, do not remove the control plane machine manifests as described by the "Creating the Kubernetes manifest and Ignition config files" section in the installation documentation. When you are instructed to run the following command:

  ```
  $ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml
  ```

  Run this command instead:

  ```
  $ rm -f openshift/99_openshift-cluster-api_worker-machineset-*.yaml
  ```

  (BZ#1927244)

- In OpenShift Container Platform 4.3 and 4.4, if the user has the console open in multiple tabs, some sidebar links in the **Developer** perspective do not directly link to the project, and there is an unexpected shift in the selected project. This will be resolved in a future release. (BZ#1839101)

- In OpenShift Container Platform 4.5, a user with scale permissions cannot scale a deployment or deployment config using the console if they do not have edit rights to the deployment or deployment config. This will be resolved in a future release. (BZ#1886888)

- In OpenShift Container Platform 4.5, when there is minimal or no data in the **Developer**

perspective, most of the monitoring charts or graphs (CPU consumption, memory usage, and bandwidth) show a range of -1 to 1. However, none of these values can ever go below zero. This will be resolved in a future release. (BZ#1904106)

- Currently, the prerequisites in the web console quick start cards appear as a paragraph instead of a list. This will be resolved in a future release. (BZ#1905147)

- Currently, in the **Search Page**, the **Pipelines** resources table is not immediately updated after the **Name** filter is applied or removed. However, if you refresh the page or close and expand the **Pipelines** section, the **Name** filter is applied. The same behavior is seen when you remove the **Name** filter. This will be resolved in a future release. (BZ#1901207).

- The Operator SDK CLI tool supports running on macOS, however the macOS binary is currently missing from the OpenShift mirror site. The macOS binary will be added in a future update. (BZ#1930357)

- Currently, on clusters with IPsec enabled, Red Hat Enterprise Linux (RHEL) 7.9 nodes cannot communicate with Red Hat Enterprise Linux CoreOS (RHCOS) nodes. (BZ#1925925)

- If you have clusters that expose the default Ingress Controller through an administrator-provisioned external load balancer that redirects all HTTP traffic to HTTPS, you must patch the new clear-text Ingress Canary route to use edge termination during the 4.6 to 4.7 upgrade process.

  Sample patch command

  ```
  $ oc patch route/canary -n openshift-ingress-canary -p '{"spec":{"tls":
  {"termination":"edge","insecureEdgeTerminationPolicy":"Redirect"}}}'
  ```

  (BZ#1932401)

- Updates to openvswitch ("net: openvswitch: reorder masks array based on usage") code causes the openvswitch et/openvswitch/flow_table::flow_lookup accessing per-cpu data condition on preemptible (and migratable) sections, leading to a real time kernel panic. As a result, the kernel-rt is unstable and will impact low latency applications.
  It is recommended not to upgrade to OpenShift Container Platform 4.7 until this is fixed.

  (BZ#1918456)

- The SR-IOV device plug-in does not allow VFIO devices on the nodes to be exposed as resources. This results in DPDK workloads being blocked on Intel devices.
  It is recommended that SR-IOV customers should not upgrade to OpenShift Container Platform 4.7 until this issue is fixed.

  (BZ#1930469)

- In OpenShift Container Platform 4.7, **ConfigInformers** objects added to the Operator infrastructure code unsuccessfully start. As a result, the **ConfigObserver** object fails to sync the cache. When this happens, the oVirt CSI Driver Operator shuts down after a couple of minutes, which leads to continual restarts. As a workaround, you can perform the following procedure:

  1. Switch the project to a cluster with the oVirt CSI Operator:

     ```
     $ oc project openshift-cluster-csi-drivers
     ```

  2. Check for **warning: restart** messages:

```
$ oc status
```

3. If there are no warnings, enter the following command:

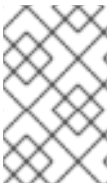
```
$ oc get pods
```

As a result, the oVirt CSI Driver Operator no longer continually restarts. (BZ#1929777)

1.9. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Container Platform 4.7 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 4.7 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 4.7. Versioned asynchronous releases, for example with the form OpenShift Container Platform 4.7.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.

1.9.1. RHEA-2020:5633 - OpenShift Container Platform 4.7.0 image release, bug fix, and security update advisory

Issued: 2021-02-24

OpenShift Container Platform release 4.7.0, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHSA-2020:5633 advisory. The RPM packages that are included in the update are provided by the RHSA-2020:5634 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 4.7.0 container image list

1.9.2. RHBA-2021:0678 - OpenShift Container Platform 4.7.1 bug fix update

Issued: 2021-03-08

OpenShift Container Platform release 4.7.1 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:0678 advisory. The RPM packages that are included in the update are provided by the RHBA-2021:0677 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 4.7.1 container image list

1.9.2.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.3. RHBA-2021:0749 - OpenShift Container Platform 4.7.2 bug fix update

Issued: 2021-03-15

OpenShift Container Platform release 4.7.2 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:0749 advisory. The RPM packages that are included in the update are provided by the RHBA-2021:0746 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 4.7.2 container image list

1.9.3.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.4. RHBA-2021:0821 - OpenShift Container Platform 4.7.3 bug fix update

Issued: 2021-03-22

OpenShift Container Platform release 4.7.3 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:0821 advisory. The RPM packages that are included in the update are provided by the RHBA-2021:0822 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 4.7.3 container image list

1.9.4.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.5. RHSA-2021:0957 - OpenShift Container Platform 4.7.4 bug fix and security update

Issued: 2021-03-29

OpenShift Container Platform release 4.7.4, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHSA-2021:0957 advisory. The RPM packages that are included in the update are provided by the RHSA-2021:0958 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 4.7.4 container image list

1.9.5.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.6. RHSA-2021:1005 - OpenShift Container Platform 4.7.5 bug fix and security update

Issued: 2021-04-05

OpenShift Container Platform release 4.7.5, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHSA-2021:1005 advisory. The RPM packages that are included in the update are provided by the RHSA-2021:1006 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 4.7.5 container image list

1.9.6.1. Features

1.9.6.1.1. Installing a cluster on VMC on AWS

You can now install an OpenShift Container Platform cluster on VMware vSphere infrastructure by deploying it to VMware Cloud (VMC) on AWS. See the documentation for deploying a cluster to VMC for more information.

1.9.6.1.2. Adding memory and uptime metadata to the Insights Operator archive

This update adds **uptime** and **memory alloc** metadata to the Insights Operator archive so that small memory leaks can be investigated properly. For more information, see **BZ#1935605**.

1.9.6.1.3. SAP license management enhancement

With this update, you can now use the following command to detect failure in the license management pod:

```
# oc logs deploy/license-management-l4rvh
```

Example output

```
Found 2 pods, using pod/license-management-l4rvh-74595f8c9b-flgz9
+ iptables -D PREROUTING -t nat -j VSYSTEM-AGENT-PREROUTING
+ true
+ iptables -F VSYSTEM-AGENT-PREROUTING -t nat
```

```
+ true
+ iptables -X VSYSTEM-AGENT-PREROUTING -t nat
+ true
+ iptables -N VSYSTEM-AGENT-PREROUTING -t nat
iptables v1.6.2: can't initialize iptables table `nat': Permission denied
```

If results return **Permission denied**, iptables or your kernal might need upgraded. For more information, see BZ#1939061.

1.9.6.2. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.7. RHBA-2021:1075 - OpenShift Container Platform 4.7.6 bug fix update

Issued: 2021-04-12

OpenShift Container Platform release 4.7.6 is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:1075 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 4.7.6 container image list

1.9.7.1. Bug fixes

- Previously, an error occurred when loading the **Topology** page. With this release, the issue is resolved and the **Topology** page loads successfully. (BZ#1940437)

- When upgrading from 4.6 to 4.7, the hostname set by the vsphere-hostname service was only applied on installation of the node. If the hostname was not statically set prior to upgrading, the hostname might have been lost. This update removes the condition which allowed the vsphere-hostname service to only run when a node is installed. As a result, vsphere-hostnames are no longer lost when upgrading. (BZ#1943143)

- Previously, BZ#1936587 set the global CoreDNS cache max TTL to 900 seconds. Consequently, NXDOMAIN records received from upstream resolvers were cached for 900 seconds. This update explicitly caches negative DNS response records for a maximum of 30 seconds. As a result, resolving NXDOMAINs records are no longer cached for 900 seconds. (BZ#1943826)

- Previously, the **growpart** script did not consider in-place LUKS rootfs file reprovisioning as **requiring growing**. Consequently, machines that enabled in-place LUKs encryption created rootfs files that were too small. With this update, the **growpart** script, now **ignition-ostree-growfs**, considers in-place LUKS rootfs file reprovisioning as **requiring growing**. As a result, machines that enable in-place LUKs encryption create rootfs files that take up all available disk space. (BZ#1941760)

- Previously, the **prjquota** kernel argument was dropped if rootfs reprovisioning, such as LUKS, was enabled. Consequently, disk space quota management features in OpenShift Container Platform would break. With this update, the **prjquota** kernel argument is now retained even if rootfs is reprovisioned. As a result, OpenShift Container Platform features that are dependent on that rootfs mount option now work. (BZ#1940966)

1.9.7.2. Features

1.9.7.2.1. BareMetal Operator Enhancement

This update adds new capabilities to the BareMetal Operator that allow for different reboot modes to be used. This provides a path for clients to quickly power down systems for remediation purposes, and to recover workloads as quickly as possible in the event of a node failure. For more information, see BZ#1936407.

1.9.7.2.2. Cluster API provider BareMetal (CAPBM) enhancement

This update adds new capabilities to the cluster API provider BareMetal (CAPBM) to request a hard power-off upon remediation. This enhancement leverages recent changes to the BareMetal Operator to support hard and soft reboot modes. As a result, the CAPBM requests hard reboot when remediation is required, bypassing the default soft power-off that the BareMetal Operator issues. For more information, see BZ#1936844.

1.9.7.3. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.8. RHBA-2021:1149 - OpenShift Container Platform 4.7.7 bug fix and security update

Issued: 2021-04-20

OpenShift Container Platform release 4.7.7, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:1149 advisory. The RPM packages that are included in the update are provided by the RHSA-2021:1150 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 4.7.7 container image list

1.9.8.1. Bug fixes

- Previously, and for unknown reasons, a kubelet could register the wrong IP address for a node. As a consequence, the node would be in a **NotReady** state until it was rebooted. Now, the systemd manager configuration is reloaded with the valid IP address as an environment variable, meaning that nodes no longer enter a **NotReady** state because a kubelet registered the wrong IP address. (BZ#1944394)

- Previously, after CVE-2021-3344 was fixed, builds did not automatically mount entitlement keys on the node. The fix minimized the amount of data copied from a pod's **/run/secrets** directory to the build container, causing the **/run/secrets/etc-pki-entitlements** file to be ommitted. As a result, the fix prevented entitled builds from working seamlessly when the entitlement certificates were stored on the OpenShift host or node.
  Now, the OpenShift build image and associated pod mount all entitlement-related files from **/run/secrets** into the build container. Entitled builds cannot pick up the certificates stored on the OpenShift host/node. Note that you can ignore warning messages like **level=warning msg="Path \"/run/secrets/etc-pki-entitlement\" from \"/etc/containers/mounts.conf\" doesn't exist, skipping** when running OpenShift Container Platform builds on Red Hat Enterprise Linux CoreOS (RHCOS) nodes.

(BZ#1945692)

1.9.8.2. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.9. RHSA-2021:1225 - OpenShift Container Platform 4.7.8 bug fix and security update

Issued: 2021-04-26

OpenShift Container Platform release 4.7.8, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHSA-2021:1225 advisory. The RPM packages that are included in the update are provided by the RHBA-2021:1226 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 4.7.8 container image list

1.9.9.1. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

1.9.10. RHBA-2021:1365 - OpenShift Container Platform 4.7.9 bug fix and security update

Issued: 2021-05-04

OpenShift Container Platform release 4.7.9, which includes security updates, is now available. The list of bug fixes that are included in the update is documented in the RHBA-2021:1365 advisory. The RPM packages that are included in the update are provided by the RHSA-2021:1366 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 4.7.9 container image list

1.9.10.1. Bug fixes

- Previously, the Cluster Samples Operator could make changes to the controller cache for objects it was watching, which caused errors when Kubernetes managed the controller cache. This update adds fixes to how the Cluster Samples Operator uses information in the controller cache. As a result, the Cluster Samples Operator does not cause errors by modifying controller caches. (BZ#1950808)

1.9.10.2. Upgrading

To upgrade an existing OpenShift Container Platform 4.7 cluster to this latest release, see Updating a cluster by using the CLI for instructions.

the operator could modify

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 4.7 cluster, all masters must be 4.7 and all nodes must be 4.7. If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.7. 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.