Red Hat Ceph Storage 6.0

Release Notes

Release notes for Red Hat Ceph Storage 6.0

Last Updated: 2023-12-08

Red Hat Ceph Storage 6.0 Release Notes

Release notes for Red Hat Ceph Storage 6.0

Legal Notice

Abstract

The release notes describes the major features, enhancements, known issues, and bug fixes implemented for the Red Hat Ceph Storage 6.0 product release. 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 our CTO Chris Wright's message.

Table of Contents

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 our CTO Chris Wright’s message .

PROVIDING FEEDBACK ON RED HAT CEPH STORAGE DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so, create a Bugzilla ticket:

1. Go to the *Bugzilla* website.
2. In the Component drop-down, select **Documentation**.
3. In the Sub-Component drop-down, select the appropriate sub-component.
4. Select the appropriate version of the document.
5. Fill in the **Summary** and **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
6. Optional: Add an attachment, if any.
7. Click **Submit Bug**.

CHAPTER 1. INTRODUCTION

Red Hat Ceph Storage is a massively scalable, open, software-defined storage platform that combines the most stable version of the Ceph storage system with a Ceph management platform, deployment utilities, and support services.

The Red Hat Ceph Storage documentation is available at https://access.redhat.com/documentation/en-us/red_hat_ceph_storage/6.

CHAPTER 2. ACKNOWLEDGMENTS

Red Hat Ceph Storage version 6.0 contains many contributions from the Red Hat Ceph Storage team. In addition, the Ceph project is seeing amazing growth in the quality and quantity of contributions from individuals and organizations in the Ceph community. We would like to thank all members of the Red Hat Ceph Storage team, all of the individual contributors in the Ceph community, and additionally, but not limited to, the contributions from organizations such as:

- Intel®
- Fujitsu ®
- UnitedStack
- Yahoo ™
- Ubuntu Kylin
- Mellanox ®
- CERN ™
- Deutsche Telekom
- Mirantis ®
- SanDisk ™
- SUSE ®

CHAPTER 3. NEW FEATURES

This section lists all major updates, enhancements, and new features introduced in this release of Red Hat Ceph Storage.

Compression on-wire with msgr2 protocol is now available

With this release, in addition to encryption on wire, compression on wire is also supported to secure network operations within the storage cluster.

See the *Encryption and key management* section in the *Red Hat Ceph Storage Data Security and Hardening Guide* for more details.

Python notifications are more efficient

Previously, there were some unused notifications that no modules needed at the moment. This caused inefficiency.

With this release, the **NotifyType** parameter is introduced. It is annotated, which events modules consume at the moment, for example **NotifyType.mon_map**, **NotifyType.osd_map**, and the like. As a consequence, only events that modules ask for are queued. The events that no modules consume are issued. Because of these changes, python notifications are now more efficient.

The changes to **pg_num** are limited

Previously, if drastic changes were made to **pg_num** that outpaced **pgp_num**, the user could hit the per-osd placement group limits and cause errors.

With this release, the changes to **pg_num** are limited to avoid the issue with per-osd placement group limits.

New **pg_progress** item is created to avoid dumping all placement group statistics for progress updates

Previously, the **pg_dump** item included unnecessary fields that wasted CPU if it was copied to **python-land**. This tended to lead to long **ClusterState::lock** hold times, leading to long **ms_dispatch** delays and generally slowing the processes.

With this release, a new **pg_progress** item is created to dump only the fields that **mgr tasks** or **progress** needs.

The **mgr_ip** is no longer re-fetched

Previously, the **mgr_ip** had to be re-fetched during the lifetime of an active Ceph manager module.

With this release, the **mgr_ip** does not change for the lifetime of an active Ceph manager module, thereby, there is no need to call back into Ceph Manager for re-fetching.

WORM compliance is now supported

Red Hat now supports WORM compliance.

See the *Enabling object lock for S3* for more details.

Set rate limits on users and buckets

With this release, you can set rate limits on users and buckets based on the operations in a Red Hat Ceph Storage cluster. See the *Rate limits for ingesting data* for more details.

librbd plugin named persistent write log cache to reduce latency

With this release, the new **librbd** plugin named Persistent Write Log Cache (PWL) provides a persistent, fault-tolerant write-back cache targeted with SSD devices. It greatly reduces latency and also improves performance at low **io_depths**. This cache uses a log-ordered write-back design which maintains checkpoints internally, so that writes that get flushed back to the cluster are always crash consistent. Even if the client cache is lost entirely, the disk image is still consistent; but the data will appear to be stale.

Ceph File System (CephFS) now supports high availability asynchronous replication for snapshots

Previously, only one **cephfs-mirror** daemon would be deployed per storage cluster, thereby a CephFS supported only asynchronous replication of snapshots directories.

With this release, multiple **cephfs-mirror** daemons can be deployed on two or more nodes to achieve concurrency in snapshot synchronization, thereby providing high availability.

See the *Ceph File System mirroring* section in the *Red Hat Ceph Storage File System Guide* for more details.

BlueStore is upgraded to V3

With this release, BlueStore object store is upgraded to V3. Following are the two features:

- The allocation metadata is removed from RocksDB and now performs a full destage of the allocator object with the OSD allocation.
- With cache age binning, older onodes might be assigned a lower priority than the hot workload data. See the *Ceph BlueStore* for more details.

Use **cephadm** to manage operating system tuning profiles

With this release, you can use **cephadm** to create and manage operating system tuning profiles for better performance of the Red Hat Ceph Storage cluster. See the *Managing operating system tuning profiles with `cephadm`* for more details.

The new **cephfs-shell** option is introduced to mount a filesystem by name

Previously, cephfs-shell could only mount the default filesystem.

With this release, a CLI option is added in cephfs-shell that allows the mounting of a different filesystem by name, that is, something analogous to the **mds_namespace=** or **fs= options** for **kclient** and **ceph-fuse**.

Day-2 tasks can now be performed through the Ceph Dashboard

With this release, on the Ceph dashboard, user can perform day-2 tasks that require daily or weekly frequency of actions. This enhancement improves the Dashboard’s assessment capabilities, customer experience, and strengthens its usability and maturity. In addition to this, new on-screen elements are also included to help and guide the user in retrieving additional information to complete a task.

3.1. THE CEPHADM UTILITY

OS tuning profiles added to manage kernel parameters using **cephadm**

With this release, to achieve feature parity with **ceph-ansible**, users can apply **tuned** profile specifications that cause **cephadm** to set OS tuning parameters on the hosts matching the specifications

See the *Managing operating system tuning profiles with `cephadm`* for more details.

Users can now easily set the Prometheus TSDB retention size and time in the Prometheus specification

Previously, users could not modify the default 15d retention period and disk consumption from Prometheus.

With this release, users can customize these settings through **cephadm** so that they are persistently applied, thereby making it easier for users to specify how much and for how long they would like their Prometheus instances to return data.

The format for achieving this is as follows:

Example

```
service_type: prometheus
placement:
  count: 1
spec:
  retention_time: "1y"
  retention_size: "1GB"
```

New Ansible playbook is added to define an insecure registry

Previously, when deploying a Red Hat Ceph Storage cluster with a large number of hosts in a disconnected installation environment, it was tedious to populate the **/etc/containers/registries.conf** file on each host.

With this release, a new Ansible playbook is added to define an insecure registry in the **/etc/containers/registries.conf** file. Therefore, the deployment of such a Ceph cluster in a disconnected installation environment is now easier as the user can populate **/etc/containers/registries.conf** with this new playbook.

3.2. CEPH DASHBOARD

Improved Ceph Dashboard features for **rbd-mirroring** is now available

Previously, there was no Ceph Block Device Snapshot mirroring support from the user interface.

With this release, the Ceph Block Device Mirroring tab on the Ceph Dashboard is enhanced with the following features that were previously present only in the command-line interface (CLI):

- Support for enabling or disabling mirroring in images.
- Support for promoting and demoting actions.
- Support for resyncing images.
- The improvement of visibility for editing site names and creating bootstrap keys.
- A blank page consisting a button to automatically create an **rbd-mirror** if none exists.

A new logging functionality is added to the Ceph dashboard

With this release, a centralized logging functionality for a single cluster, named Daemon Logs, is implemented on the dashboard under the Cluster → Logs section. This makes it easier for users to monitor logs in an efficient manner.

A new TTL cache is added between the Ceph Manager and its modules

Big Ceph clusters generate a lot of data, which might overload the cluster and render modules unsuitable.

With this release, a new TTL cache is added between the Ceph Manager and its modules to help alleviate loads and prevent the cluster from overloading.

A new information message is provided on Ceph Dashboard to troubleshoot issues with Grafana

When Grafana is deployed with self-signed TLS certificates instead of certificates signed by a Certificate Authority, most browsers, such as Chrome or Firefox, do not allow the embedded Grafana iframe to be displayed within the Ceph Dashboard.

This is a security limitation imposed by browsers themselves. Some browsers, like Firefox, still display a security warning: **Your connection is not secure**, but still allow users to accept the exception and load the embedded Grafana iframe. However, other browsers, for example Chrome, silently fail and do not display any kind of error message, therefore users were not aware of the failure.

With this release, a new notification is displayed on the Ceph Dashboard:

> If no embedded Grafana Dashboard appeared below, please follow this link to check if Grafana is reachable and there are no HTTPS certificate issues. You may need to reload this page after accepting any Browser certificate exceptions.

The number of repaired objects in pools is exposed under Prometheus metrics

Previously, data regarding auto-repaired objects was gathered through log parsing which was inefficient.

With this release, the number of repaired objects per pool is now exposed as Prometheus metrics on the Ceph Dashboard.

Ceph Dashboard now clearly indicates errors on certain CephFS operations

Previously, when a user tried to perform an operation on a filesystem directory, but did not have permission, the Ceph Dashboard reported a generic 500 internal server-side error. However, these errors are actually imputable to users, since permissions are the same for preventing certain actions for given users.

With this release, when the user tries to perform an unauthorized operation, they receive a clear explanation on the permission error.

Users can now see new metrics for different storage class in Prometheus

With this release three new metrics, **ceph_cluster_by_class_total_bytes**, **ceph_cluster_by_class_total_used_bytes**, and **ceph_cluster_by_class_total_used_raw_bytes**, are added for different storage class in Prometheus filtered by device class which would help to follow up the performances and the capacity of the infrastructure.

The WAL and DB devices now get filters pre selected automatically

Previously, the user had to manually apply filters to the selected WAL or DB devices, which was a repetitive task.

With this release, when the user selects devices on the primary devices table, the appropriate filters are pre selected for WAL and DB devices.

A new shortcut button for silencing alerts is added

With this release, users can create a silence for every alert in the notification bar on the Ceph Dashboard using the newly created silent shortcut.

Users can now add server side encryption to the Ceph Object Gateway bucket from the Dashboard

Previously, there was no option on the Ceph Dashboard to add server side encryption (SSE) to the Ceph Object Gateway buckets.

With this release, it is now possible to add SSE while creating the Ceph Object Gateway bucket through the Ceph Dashboard.

Cross origin resource sharing is now allowed

Previously, IBM developers were facing issues with their storage insights product when they tried to ping the REST API using their front-end because of the tight cross origin resource sharing (CORS) policies setup in the REST API.

With this release, the **cross_origin_url** option is added, which can be set to a particular URL. The REST API now allows communicating with only that URL.

Example

```
[ceph: root@host01 /]# ceph config set mgr mgr/dashboard/cross_origin_url http://localhost:4200
```

3.3. CEPH FILE SYSTEM

Users can now set and manage quotas on subvolume group

Previously, the user could only apply quotas to individual subvolumes.

With this release, the user can now set, apply and manage quotas for a given subvolume group, especially when working on a multi-tenant environment.

The Ceph File System client can now track average read, write, and metadata latencies

Previously, the Ceph File System client would track only the cumulative read, write, and metadata latencies. However, average read, write, and metadata latencies are more useful to the user.

With this feature, the client can start tracking average latencies and forward it to the metadata server to display in the **perf stats** command output and the **cephfs-top** utility.

The **cephfs-top** utility is improved with the support of multiple file systems

Previously, the **cephfs-top** utility with multiple file systems was not reliable. Moreover, there was no option to display the metrics for only the selected file system.

With this feature, the **cephfs-top** utility now supports multiple file systems and it is now possible to select an option to see the metrics related to a particular file system.

Users can now use the **fs volume info** command to display basic details about a volume

Previously, there was no command in Ceph File System to list only the basic details about a volume.

With this release, the user can list the basic details about a volume by running the **fs volume info** command.

See *Viewing information about a Ceph file system volume* in *Red Hat Ceph Storage File System Guide*.

Users can list the in-progress or pending clones for a subvolume snapshot

Previously, there was no way of knowing the set of clone operations in-progress or pending for a subvolume snapshot, unless the user knew the clone subvolume’s name and used **clone status** command to infer the details.

With this release, for a given subvolume snapshot name, the in-progress or pending clones can be listed.

Users can now use the **--human-readable** flag with the **fs volume info** command

Previously, all the sizes were displayed only in bytes on running the **fs volume info** command.

With this release, users can now see the sizes along with the units on running **fs volume info** command.

3.4. CEPH OBJECT GATEWAY

New S3 bucket lifecycle notifications are now generated

With this release, S3 bucket notifications are generated for current and non-current versions, delete-marker expiration generated by lifecycle processing. This capability is potentially useful for application workflow, among other potential uses.

The objects are transitioned to the S3 cloud endpoint as per the set lifecycle rules

In Red Hat Ceph Storage, we use a special storage class of tier type **cloud-s3** to configure the remote cloud S3 object store service to which the data needs to be transitioned. These are defined in terms of zonegroup placement targets and unlike regular storage classes, do not need a data pool.

With this release, users can transition Ceph Object Gateway objects from a Ceph Object Gateway server to a remote S3 cloud-point through storage classes. However, the transition is unidirectional, as such data cannot be transitioned back from the remote server.

The Ceph Object Gateway S3 policy errors are now more useful

Previously, Ceph Object Gateway S3 policy error messages were opaque and not very useful. The initial issue with not being able to access data in the buckets after upgrading versions seemed to be the result of an accepted but invalid principal being ignored silently on ingest but rejected on use later due to a code change.

With this release, the policy now prints detailed and useful error messages. There is also a new **rgw-policy-check** command that lets policy documents be tested in the command line, and a new option **rgw policy reject invalid principals** that is **false** by default and that rejects, with an error message, invalid principals on ingest only rather than ignoring them without error.

Level 20 Ceph Object Gateway log messages are reduced when updating bucket indices

With this release the Ceph Object Gateway level 20 log messages are reduced when updating bucket indices to remove messages that do not add value and to reduce size of logs.

3.5. MULTI-SITE CEPH OBJECT GATEWAY

Lifecycle policy now runs on all zones in multi-site configurations

With this release, lifecycle policy runs on all zones in multi-site Red Hat Ceph Storage configurations, which makes lifecycle processing more resilient in these configurations. The changes are made to also permit new features, such as conditional processing in archive zones.

Multi-site configuration supports dynamic bucket index resharding

Previously, only manual resharding of the buckets for multi-site configurations was supported.

With this release, dynamic bucket resharding is supported in multi-site configurations. Once the storage clusters are upgraded, enable the **resharding** feature and reshard the buckets either manually with **radosgw-admin bucket reshard** command or automatically with dynamic resharding, independently of other zones in the storage cluster.

Sites can now customize STS **max-session-duration** parameter with the role interface

Previously, the **max-session-duration** parameter controlling duration of STS sessions could not be configured because it was not exposed on the interface.

With this release, it is possible to customize STS **max-session-duration** parameter through the role interface.

3.6. RADOS

Kafka SASL/SCRAM security mechanism is added to bucket notifications

With this release, Kafka SASL/SCRAM security mechanism is added to bucket notifications.

To know how to use the feature, refer to the "kafka" section in *Creating a topic*. Note that end-to-end configuration for the feature,in case of testing, is out of scope of Ceph.

Low-level log messages are introduced to warn user about hitting throttle limits

Previously, there was a lack of low-level logging indication that throttle limits were hit, causing these occurrences to incorrectly have the appearance of a networking issue.

With this release, the introduction of low-level log messages makes it much clearer that the throttle limits are hit.

The user is warned on Filestore deprecation through **ceph status** and **ceph health detail** commands

BlueStore is the default and widely used objectstore.

With this release, if there are any OSDs that are on Filestore, the storage cluster goes into the **HEALTH_WARN** status due to the **OSD_FILESTORE** health check. The end user has to migrate the OSDs which are on Filestore to BlueStore to clear this warning.

User can now take advantage of a tunable KernelDevice buffer in BlueStore

With this release, users can now configure custom alignment for read buffers using **bdev_read_buffer_alignment** command in BlueStore. This removes the limitations imposed by the default 4 KiB alignment space, when buffers are intended to be backed up by huge pages.

Additionally, BlueStore, through KernelDevice, gets a configurable pool with **bdev_read_preallocated_huge_buffer_num** parameter of MAP_HUGETLB-based read buffers for workloads with cache-unfriendly access patterns, which undergo recycling and are not cacheable.

Taken together, these features allow to shorten a scatter-gather list that is passed by the storage component to NICs, thereby improving the handling of huge page-based read buffers in BlueStore.

OSDs report the slow operation details in an aggregated format to the Ceph cluster log

Previously, slow requests would overwhelm a cluster log with too many details, filling up the monitor database.

With this release, slow requests by operation type and by pool information gets logged to the cluster log in an aggregated format.

Users can now blocklist a CIDR range

With this release, you can blocklist a CIDR range, in addition to individual client instances and IPs. In certain circumstances, you would want to blocklist all clients in an entire data center or rack instead of specifying individual clients to blocklist.

For example, failing over a workload to a different set of machines and wanting to prevent the old workload instance from continuing to partially operate.

This is now possible using a "blocklist range" analogous to the existing "blocklist" command.

3.7. RADOS BLOCK DEVICES (RBD)

librbd SSD-based persistent write-back cache to reduce latency now fully supported.

With this release, the **pwl_cache** librbd plugin provides a log-structured write-back cache targeted at SSD devices. Just as with the already provided log-structured write-back cache targeted at PMEM devices, the updates to the image are batched and flushed in-order, retaining the actual image in a crash-consistent state. The benefits and use cases remain the same, but users no longer need to procure more expensive PMEM devices to take advantage of them.

The librbd compare-and-write operation is improved and new**rbd_aio_compare_and_writev** API method is introduced

- **The semantics of compare-and-write C++ API now match those of C API.**
  Previously, the compare-and-write C++ API, that is **Image::compare_and_write** and **Image::aio_compare_and_write** methods, would compare up to the size of the compare buffer. This would cause breakage after straddling a stripe unit boundary.

  With this release, the compare-and-write C++ API matches the semantics of C API and both compare and write steps operate only on **len** bytes even if the respective buffers are larger.

- **The compare-and-write operation is no longer limited to 512-byte sectors.**
  With this release, the compare-and-write can operate on stripe units, if the access is aligned properly. The stripe units are 4 MB by default.

- **New rbd_aio_compare_and_writev API method is now available.**
  With this release, the **rbd_aio_compare_and_writev** API method is included to support scatter/gather on both compare and write buffers, which complements existing **rbd_aio_readv** and **rbd_aio_writev** methods.

Layered client-side encryption is now supported

With this release, cloned images can be encrypted, each with its own encryption format and passphrase, potentially different from that of the parent image. The efficient copy-on-write semantics used for unformatted regular cloned images are retained.

CHAPTER 4. BUG FIXES

This section describes bugs with significant impact on users that were fixed in this release of Red Hat Ceph Storage. In addition, the section includes descriptions of fixed known issues found in previous versions.

4.1. THE CEPHADM UTILITY

The PID limit is removed and workloads in the container no longer crash

Previously, in Red Hat Enterprise Linux 9 deployments, pid limits were enforced which limited the number of processes able to run inside the container. Due to this, certain operations, such as Ceph Object Gateway sync, would crash.

With this fix, the **pid limit** is set to **unlimited** on all Ceph containers, preventing the workloads in the container from crashing.

(BZ#2165644)

Cephadm no longer randomly temporarily removes config and keyring files

Previously, due to incorrect timing on when to calculate the client conf and keyrings, **cephadm** would calculate that there should be no config and keyrings placed on any host and would subsequently remove all of them.

With this fix, the timing of the calculation is changed to guarantee up-to-date information for the calculation. **Cephadm** no longer randomly, temporarily removes config and keyring files it is managing.

(BZ#2125002)

The Ceph Object Gateway daemons now bind to loopback addresses correctly

Previously, **cephadm** excluded loopback interfaces when looking for a valid IP address on a host to bind the Ceph Object Gateway daemon, thereby the daemons would not bind to the loopback addresses.

With this fix, the Ceph Object Gateway daemons can be bound to loopback addresses by performing an explicit check. If a loopback interface is detected, the **127.0.0.1** address is used for IPv4 and **::1** is used for IPv6 as the loopback address.

(BZ#2018245)

Cephadm now splits the device information into multiple entries after exceeding the Ceph monitor store size limit

Previously, **cephadm** was unable to refresh the hosts and complete most operations when the device information exceeded the monitor store default maximum size limit of 64K. This caused an entry size error. As a result, users had to raise the default limit if they had hosts with a large number of disks.

With this fix, **cephadm** now splits the device information into multiple entries if it takes more space than the size limit. Users no longer have to raise the monitor store entry size limit if they have hosts with a large number of disks.

(BZ#2053276)

Crash daemon now correctly records crash events and reports them to the storage cluster

Previously, crash daemon would not authenticate properly when sending crash reports to the storage cluster, which caused it to not correctly record crash events to send to the cluster.

With this fix, crash daemon now properly uses its authentication information when sending crash reports. It now correctly records crash events and reports them to the cluster.

(BZ#2062989)

Log rotation of the **cephadm.log** should no longer cause issues

Previously, the **logrotate** command would cause issues if the **/var/log/ceph** directory was created by something other than **cephadm**, for example **ceph-common** or **ceph-ansible**. As a consequence, the **cephadm.log** could not be rotated.

With this fix, **su root root** was added to the logrotate configuration to rotate as a **root** user. The **logrotate** command no longer causes an issue with the ownership of **var/log/ceph** directory, therefore the **cephadm.log** is rotated as expected.

(BZ#2099670)

Cephadm logging configurations are updated.

Previously, **cephadm** scripts were logging all output to **stderr**. As a result, cephadm bootstrap logs signifying successful deployment were also being sent to **stderr** instead of **stdout**.

With this fix, **cephadm** script now has different logging configurations for certain commands and the one used for bootstrap now only logs errors to **stderr**.

(BZ#2103707)

The network check no longer causes the hosts to be excluded from the monitor network

Previously, the network check would fail because **cephadm** would look for the exact match between the host network and some of the configured public networks. This caused the hosts with valid network configuration, which are the hosts with an interface that belonged to **public_network**, to be excluded from the monitor network.

With this fix, instead of looking for an exact match, it checks if the host network overlaps with any configured public networks, therefore valid hosts are no longer excluded from the monitor network.

(BZ#2104947)

cephadm no longer removes osd_memory_target config settings at host level

Previously, if **osd_memory_target_autotune** was turned off globally, **cephadm** would remove the values that the user set for **osd_memory_target** at the host level. Additionally, for hosts with FQDN name, even though the crush map uses a short name, **cephadm** would still set the configuration option using the FQDN. Due to this, users could not manually set **osd_memory_target** at the host level and **osd_memory_target** auto tuning would not work with FQDN hosts.

With this fix, the **osd_memory_target** config settings is not removed from **cephadm** at the host level if **osd_memory_target_autotune** is set to **false**. It also always uses a short name for hosts when setting host level **osd_memory_target**. If at the host level **osd_memory_target_autotune** is set to **false**, users can manually set the **osd_memory_target** and have the options not be removed by **cephadm**. Additionally, autotuning should now work with hosts added to **cephadm** with FQDN names.

(BZ#2107849)

cephadm rewrites Ceph OSD configuration files

Previously, while redeploying OSDs, **cephadm** would not write configuration used for Ceph OSDs, thereby the OSDs would not get the updated monitor configuration in its configuration file when the Ceph Monitor daemons were either added or removed.

With this fix, **cephadm** rewrites the configuration files automatically when redeploying OSDs and the OSD configuration files get updated to show the new location of the monitors when the monitors are added or deleted without user intervention.

(BZ#2111525)

Users can now drain hosts that are listed in explicit placements

Previously, draining hosts that were listed as part of an explicit placement would cause the hosts not to be properly drained and tracebacks would be logged until the drain was stopped or hosts were removed from any explicit placement .

With this fix, the handling of explicit placements is implemented internally and **cephadm** is able to determine if it needs to remove daemons from the hosts. Consequently, users can now drain hosts that are listed as part of an explicit placement without having to first remove the host from the placement.

However, users still need to remove the host from any explicit placement before removing the host fully or specifications that explicitly list the host cannot be applied.

(BZ#2112768)

cephadm returns non-zero code when **--apply-spec** option fails during bootstrap

Previously, **cephadm** bootstrap always returned code **0** if the operation was complete. If there were any failures in the deployment using the **--apply-spec** option, it would not reflect any failures in the return code.

With this fix, **cephadm** returns a non-zero value when applying specification fails during bootstrap.

(BZ#2116689)

Complex OSD deployment or replacement with shared DB devices now does not need to be done all at once

Previously, devices already used as **db** devices for previous OSDs were filtered out as unavailable devices, when **cephadm** created OSDs. As a result, complex OSD deployment where all the OSDs that were meant to use a device as their DB, but were not deployed at once, would not work as the DB device would be filtered out when creating subsequent OSDs even though they should not be according to the OSD specification.

With this fix, complex OSD deployment with shared DB devices now does not need to be done all at once. If users update an OSD specification to include additional data devices to be paired up with the already listed db devices in the specifications, **cephadm** should be able to create these new OSDs.

(BZ#2119715)

Proper errors are raised if invalid **tuned-profile** specification is detected by **cephadm**

Previously, **cephadm** would not validate YAML specification for **tuned-profile** and consequently would not return any error or warning while applying invalid and missing data in invalid **tuned-profile** specification.

With this fix, several checks are added to validate the **tuned-profile** specifications. Proper errors are now raised when invalid **tuned-profile** specification is detected by **cephadm**:

- Invalid tunable is mentioned under “settings” in YAML specification.
- “settings” section in YAML specification is empty.
- Invalid placement is detected.

(BZ#2123609)

4.2. CEPH DASHBOARD

The host device *Life Expectancy* column now shows the correct value on the Ceph Dashboard

Previously, the host device *Life Expectancy* column would show an empty value because the column had no default value.

With this fix, the default value is assigned to the host device *Life Expectancy* column, and the column now shows the correct value.

(BZ#2021762)

Users are now able to change the Ceph Object Gateway subuser permissions

Previously, the users could not change the Ceph Object Gateway subuser permissions as the request was not implemented properly.

With this fix, the request to edit Ceph Object Gateway subuser permissions is implemented properly, therefore the user can now change the subuser permissions.

(BZ#2042888)

The overall performance graphs of the pools show correct values on the Ceph Dashboard

Previously, there was an issue in the query related to the Ceph Dashboard pools’ overall performance graphs and showed multiple entries of the same pool in the pool overview.

With this fix, the related query is fixed and overall performance graphs of the pools show correct values.

(BZ#2062085)

The Ceph Dashboard is now aligned with the command line interface’s (CLI) way of NFS exports creation

Previously, the **squash** field for the NFS exports creation was shown as a mandatory field in the edit form. Additionally, if exports were created from the backend and specified a different kind of squash name, the form would return an empty field.

With this fix, the **required** condition is removed from the **squash** field and the issue with the squash field coming up as empty in the edit form is also resolved.

(BZ#2062456)

Pool count shows correct values on the Ceph Dashboard

Previously, there was an issue in the query related to pool count. In the Overall performance tab, the Ceph Dashboard showed pool count as fractional value.

With this fix, the related query is fixed and pool count shows correct values on the Ceph Dashboard.

(BZ#2062590)

Validation is required when creating a new service name

Previously, there was no validation when creating a new service on the Ceph Dashboard, as a result, users were allowed to create a new service with an existing name. This would overwrite existing services and cause the user to lose a current running service on the hosts.

With this fix, validation is required before creating a new service on the dashboard and using an existing service name is not possible.

(BZ#2064850)

External snapshot creation in Grafana now disabled by default

Previously, creating external Grafana snapshots would generate broken links. This would make the infrastructure vulnerable to DDoS attacks ,as someone could gain insights into the environment by looking at the metric patterns.

With this fix, external Grafana snapshots are disabled and removed from the dashboard share options.

(BZ#2079847)

The services can now be safely put to the **unmanaged** mode on the Ceph Dashboard

Previously, when a user tried to create or modify the services, such as ingress or SNMP, in the **unmanaged** mode, the form would return a 500 error message and would fail to create the services. This happened because the form would not show some fields that needed to be filled even if the service was going directly to **unmanaged**.

With this fix, the form now shows the necessary fields and the validation is improved as well, therefore, all services can now be safely put to the **unmanaged** mode.

(BZ#2080316)

The Ceph Dashboard now securely connects with the hostname instead of IP address

Previously, an attempt to access the Object Gateway section of the Ceph Dashboard would throw a 500 - internal server error. This error was as a result of the Ceph Dashboard trying to establish the HTTPS connection to the Ceph Object Gateway daemons by IP address instead of hostname, which requires that the hostname of the server matches the hostname in the TLS certificate.

With this fix, the Ceph Dashboard can now correctly establish a HTTPS connection and successfully connect to the Object gateways using the hostname.

(BZ#2080485)

ceph-dashboard now prompts users when creating an ingress service

Previously, ingress service creation would fail with a 500 internal server error, when the form was submitted without specifying frontend and monitor port values. As a result, an ingress service could not be created from the Ceph Dashboard.

With this fix, **ceph-dashboard** prompts users to fill in all the mandatory fields, when creating an ingress service on the Ceph Dashboard.

(BZ#2080916)

The service instance column of the host table on the Ceph Dashboard now shows all the services deployed on the particular host

Previously, the service instance column of *Cluster* → *Hosts* table only showed **ceph** services and not **cephadm** services as the frontend was lacking subscription to some of the services.

With this fix, the service instance column now shows all the services deployed on the particular host in the host table on the Ceph Dashboard.

(BZ#2101771)

The Ceph Dashboard now raises appropriate error message, if user tries to create a snapshot with an existing name

Previously, Ceph Dashboard would not validate Ceph File System snapshot creation with an existing name and would throw a 500 - internal server error.

With this fix, the correct error message is added, which throws an appropriate error message when user creates a snapshot with an existing name.

(BZ#2111650)

Ceph node "network packet" drop alerts are shown appropriately on the dashboard

Previously, there was an issue in the query related to Ceph node "network packet" drop alerts. As a consequence, those alerts would be seen frequently on the Ceph Dashboard.

With this fix, related query no longer causes the issues and Ceph node Network Packet drop alerts are shown appropriately.

(BZ#2125433)

4.3. CEPH FILE SYSTEM

MDS daemon now resets the heartbeat in each thread after each queued work

Previously, a thread would hold the **mds_lock** for a longtime if it had a lot of work to do. This caused other threads to be starved of resources and be stuck for a longtime, as a result MDS daemon would fail to report the heartbeat to monitor in time and be kicked out of the cluster.

With this fix, the MDS daemon resets the heartbeat in each thread after each queued work.

(BZ#2060989)

Ceph Metadata Server no longer crashes during concurrent lookup and unlink operations

Previously, an incorrect assumption of an assertion placed in the code, would hit the concurrent lookup and unlink operations from a Ceph client, causing Ceph Metadata Server crash.

With this fix, the assertion is moved to the relevant place where the assumption, during concurrent lookup and unlink operation, is valid, resulting in the continuation of Ceph Metadata Server serving the Ceph client operations without crashing.

(BZ#2074162)

A replica MDS is no longer stuck, if a client sends a getattr client request just after it was created

Previously, if a client sent a getattr client request just after the replica MDS was created, the client would make a path of **#INODE-NUMBER** because the **CInode** was not linked yet. The replica MDS would keep retrying until the auth MDS flushed the **mdlog** and the **C_MDS_openc_finish** and **link_primary_inode** were called 5 seconds later at most.

With this fix, the replica MDS trying to find the **CInode** from auth MDS would manually trigger **mdslog flush**, if it could not find it.

(BZ#2091491)

Ceph File System subvolume groups created by the user are now displayed when listing subvolume groups

Previously, the Ceph File System (CephFS) subvolume groups listing included CephFS internal groups instead of CephFS subvolume groups created by users.

With this fix, the internal groups are filtered from CephFS subvolume group list. As a result, CephFS subvolume groups created by the user are displayed now.

(BZ#2093258)

Saved snap-schedules are reloaded from Ceph storage

Previously, restarting Ceph Managers caused retention policy specifications to get lost because it was not saved to the Ceph storage. As a consequence, retention would stop working.

With this fix, all changes to snap-schedules are now persisted to the Ceph storage, therefore when Ceph Managers are restarted, the saved snap-schedule is reloaded from the Ceph storage and restarted with specified retention policy specifications.

(BZ#2125773)

The API for deleting RADOS objects is updated

Previously, deleting a RADOS object would result in the program crash and would create tracebacks in the logs.

With this fix, the API is updated to correctly remove the RADOS object after an upgrade and no stack traces are dumped in logs.

(BZ#2126269)

MDS now stores all damaged dentries

Previously, metadata servers (MDS) would only store dentry damage for a **dirfrag** if dentry damage would not already exist in that **dirfrag**. As a result, only the first damaged dentry would be stored in the damage table and subsequent damage in the **dirfrag** would be forgotten.

With this fix, MDS can now properly store all the damaged dentries.

(BZ#2129968)

The ceph-mds daemon no longer crashes during the upgrade

Previously, the Ceph Metadata Server daemons (**ceph-mds**) would crash during an upgrade due to an incorrect assumption in the Metadata Servers when recovering inodes. It caused **ceph-mds** to hit an assert during an upgrade.

With this fix, the **ceph-mds** makes correct assumptions during inode recovery and the **ceph-mds** no longer crashes during an upgrade.

(BZ#2130081)

The standby-replay Metadata Server daemon is no longer unexpectedly removed

Previously, the Ceph Monitor would remove a standby-replay Metadata Server (MDS) daemon from the MDS map under certain conditions. This would cause the standby-replay MDS daemon to get removed from the Metadata Server cluster, which generated cluster warnings.

With this fix, the logic used in Ceph Monitors during the consideration of removal of an MDS daemon from the MDS map now includes information about the standby-replay MDS daemons holding a rank. As a consequence, the standby-replay MDS daemons are no longer unexpectedly removed from the MDS cluster.

(BZ#2130118)

The **subvolume snapshot info** command no longer has the **size** field in the output

Previously, the output of the **subvolume snapshot** command would return an incorrect snapshot **size**. This was due to the fact that the **snapshot info** command relies on **rstats** to track the snapshot size. The **rstats** tracks the size of the snapshot from its corresponding subvolume instead of the snapshot itself.

With this fix, the **size** field is removed from the output of the **snapshot info** command until the **rstats** is fixed.

(BZ#2130422)

The disk full scenario does not corrupt the configuration file anymore

Previously, the configuration files were being written directly to the disk without using the temporary files, which involved truncating the existing configuration file and writing the configuration data. This led to the empty configuration files when the disk was full as the truncate was successful, however writing new configuration data failed with **no space** error. Additionally, it led to the failure of all the operations on corresponding subvolumes.

With this fix, the configuration data is written to a temporary configuration file and renamed to the original configuration file and prevents truncating the original configuration file.

(BZ#2130450)

Do not abort MDS in case of unknown messages

Previously, metadata servers (MDS) would abort if it received a message that it did not understand. As a result, any malicious client would crash the server by just sending a message of a new type to the server. Beside malicious clients, this also meant that whenever there was a protocol issue, such as a new client erroneously sending new messages to the server, the whole system would crash instead of just the new client.

With this fix, MDS no longer aborts if it receives an unknown request from a client, instead it closes the session, blocklists, and evicts the client. This protects the MDS and the whole system from any intentional attacks like the denial of service from any malicious clients.

(BZ#2130984)

Directory listing from a NFS client now works as expected for NFS-Ganesha exports

Previously, Ceph File System (CephFS) Metadata Server (MDS) would not increment the change attribute, (**change_attr**) of a directory inode during CephFS operations which only changed the directory inode's **ctime**. Therefore, an NFS kernel client would not invalidate its **readdir** cache when it is supposed to. This is because the NFS Ganesha server backed by CephFS would sometimes report incorrect change attribute value of the directory inode. As a result, the NFS client would list stale directory contents for NFS Ganesha exports backed by CephFS.

With this fix, CephFS MDS now increments the change attribute of the directory inode during operations and the directory listing from the NFS client now works as expected for NFS Ganesha server exports backed by CephFS.

(BZ#2135573)

The CephFS now has the correct directory access

Previously, directory access was denied even to the UID of **0** due to incorrect Discretionary Access Control (DAC) management.

With this fix, directory access is allowed to UID **0** even if the actual permissions for the directory user, group, and others are not permissible for UID **0**. This results in the correct Ceph File System (CephFS) behavior for directory access to UID 0 by effectively granting superuser privileges.

(BZ#2147460)

4.4. THE CEPH VOLUME UTILITY

The **ceph-volume inventory** command no longer fails

Previously, when a physical volume was not a member of any volume group, **ceph-volume** would not ignore the volume, and instead would try to process it, which caused the **ceph-volume inventory** command to fail.

With this fix, the physical volumes that are not a member of a volume group are filtered and the **ceph-volume inventory** command no longer fails.

(BZ#2140000)

4.5. CEPH OBJECT GATEWAY

Users can now use MD5 for non-cryptographic purposes in a FIPS environment

Previously, in a FIPS enabled environment, the usage of MD5 digest was not allowed by default, unless explicitly excluded for non-cryptographic purposes. Due to this, a segfault occurred during the S3 complete multipart upload operation.

With this fix, the usage of MD5 for non-cryptographic purposes in a FIPS environment for S3 complete multipart **PUT** operations is explicitly allowed and the S3 multipart operations can be completed.

(BZ#2088571)

The Ceph Object Gateway no longer crashes on accesses

Previously, the Ceph Object Gateway would crash on some access due to the changes from in-place to allocated buckets as a malformed bucket URL caused a void pointer dereference to a bucket value that was not always initialized.

With this fix, the Ceph Object Gateway properly checks that the pointer is non-null before doing permission checks and throws an error if it is not initialized.

(BZ#2118423)

The code that parses dates **z-amz-date** format is changed

Previously, the standard format for **x-amz-date** was changed which caused issues, since the new software uses the new date format. The new software built with the latest **go** libraries would not talk to the Ceph Object Gateway.

With this fix, the code in the Ceph Object Gateway that parses dates in **x-amz-date** format is changed to also accept the new date format.

(BZ#2121564)

Ceph Object Gateway's Swift implicit tenant behavior is restored

Previously, a change to Swift tenant parsing caused the failure of Ceph Object Gateway's Swift implicit tenant processing.

With this fix, Swift tenant parsing logic is corrected and the Swift implicit tenant behavior is restored.

(BZ#2123177)

The Ceph Object Gateway no longer crashes after running continuously for an extended period

Previously, an index into a table would become negative after running continuously for an extended period, resulting in the Ceph Object Gateway crash.

With this fix, the index is not allowed to become negative and the Ceph Object Gateway no longer crashes.

(BZ#2155894)

Variable access no longer causes undefined program behavior

Previously, a coverity scan would identify two cases, where variables could be used after a move, potentially causing an undefined program behavior to occur.

With this fix, variable access is fixed and the potential fault can no longer occur.

(BZ#2155916)

4.6. MULTI-SITE CEPH OBJECT GATEWAY

Roles and role policy now transparently replicate when multi-site is configured

Previously, the logic to replicate S3 roles and role policy was not implemented in Ceph Object Gateway, thereby, roles and role policy created in any zone in a multi-site replicated setup were not transparently replicated to other zones.

With this fix, role and role policy replication are implemented and they transparently replicate when multi-site is configured.

(BZ#2136771)

4.7. RADOS

Slow progress and high CPU utilization during backfill is resolved

Previously, the worker thread with the smallest index in an OSD shard would return to the main worker loop, instead of waiting until an item could be scheduled from the mClock queue or until notified. This resulted in the busy loop and high CPU utilization.

With this fix, the worker thread with the smallest thread index reacquires the appropriate lock and waits until notified, or until time period lapses as indicated by the mClock scheduler. The worker thread now waits until an item can be scheduled from the mClock queue or until notified and then returns to the main worker loop thereby eliminating the busy loop and solving the high CPU utilization issue.

(BZ#2114612)

Renaming large objects no longer fails when using temporary credentials returned by STS

Previously, due to incorrect permission evaluation of **iam** policies while renaming large objects, renaming large objects would fail when temporary credentials returned by STS were used.

With this fix, **iam** policies are correctly evaluated when temporary credentials returned by STS are used to rename large objects.

(BZ#2166572)

The small writes are deferred

Previously, Ceph would defer writes while allocating units. When the allocation unit was large, like 64 K, no small write was eligible for deferring.

With this update, the small writes are deferred as they operate on disk blocks even when large allocation units are deferring.

(BZ#2107406)

The Ceph Monitor no longer crashes after reducing the number of monitors

Previously, when the user reduced the number of monitors in the quorum using the **ceph orch apply mon *NUMBER*** command, **cephadm** would remove the monitor before shutting it down. This would trigger an assertion because Ceph would assume that the monitor is shutting down before the monitor removal.

With this fix, a sanity check is added to handle the case when the current rank of the monitor is larger or equal to the quorum rank. The monitor no longer exists in the monitor map, therefore its peers do not ping this monitor, because the address no longer exists. As a result, the assertion is not triggered if the monitor is removed before shutdown.

(BZ#1945266)

The Ceph Manager checks that deals with the initial service map is now relaxed

Previously, when upgrading a cluster, the Ceph Manager would receive several **service_map** versions from the previously active Ceph manager. This caused the manager daemon to crash, due to an incorrect check in the code, when the newly activated manager received a map with a higher version sent by the previously active manager.

With this fix, the check in the Ceph Manager that deals with the initial service map is relaxed to correctly check service maps and no assertion occurs during the Ceph Manager failover.

(BZ#1984881)

The **ceph --help** command now shows that yaml formatters are only valid for **ceph orch** commands

Previously, it was implied by lack of specification in the **ceph --help** command that the yaml formatter option was valid for any ceph command, including the **ceph config dump** command.

With this fix, the output of the **ceph --help** command shows that yaml formatters are only valid for the **ceph orch** commands.

(BZ#2040709)

Corrupted dups entries of a PG Log can be removed by off-line and on-line trimming

Previously, trimming of PG log dups entries could be prevented during the low-level PG split operation, which is used by the PG autoscaler with far higher frequency than by a human operator. Stalling the trimming of dups resulted in significant memory growth of PG log, leading to OSD crashes as it ran out of memory. Restarting an OSD would not solve the problem as the PG log is stored on disk and reloaded to RAM on startup.

With this fix, both off-line, using the **ceph-objectstore-tool** command, and on-line, within OSD, trimming can remove corrupted dups entries of a PG log that jammed the on-line trimming machinery and were responsible for the memory growth. A debug improvement is implemented that prints the number of dups entries to the OSD’s log to help future investigations.

(BZ#2093106)

The **starts** message is added to notify that scrub or deep-scrub process has begun

Previously, users were not able to determine when the scrubbing process started for a placement group (PG) because the **starts** message was missing from the cluster log. This made it difficult to calculate the time taken to scrub or deep-scrub a PG.

With this fix, the **scrub starts** or the **deep-scrub starts** message appears to notify the user that scrub or deep-scrub process has begun for the PG.

(BZ#2094955)

The **autoscale-status** command no longer displays the **NEW PG_NUM** value if PG-autoscaling is disabled

Previously, the **autoscale-status** command would show the **NEW PG_NUM** value even though PG-autoscaling was not enabled. This would mislead the end user by suggesting that PG autoscaler applied **NEW PG_NUM** value to the pool, which was not the case.

With this fix, if the **noautoscale** flag is set, the **NEW PG_NUM** value is not shown in the **autoscale-status** command output.

(BZ#2099193)

Users can remove cloned objects after upgrading a cluster

Previously, after upgrading a cluster from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5 , removing snapshots of objects created in earlier versions would leave clones, which could not be removed. This was because the SnapMapper key's were wrongly converted.

With this fix, SnapMapper's legacy conversation was updated to match the new key format and cloned objects in earlier versions of Ceph can now be easily removed after an upgrade.

(BZ#2107404)

The **ceph daemon heap stats** command now returns required usage details for daemon

Previously, the **ceph daemon osd.x heap stats** command would return empty output instead of the current heap usage for a Ceph daemon. Consequently, users were compelled to use the **ceph tell heap stats** command to get the desired heap usage.

With this fix, the **ceph daemon heap stats** command returns heap usage information for a Ceph daemon similar to what we get using **ceph tell** command.

(BZ#2119101)

The Prometheus metrics now reflect the correct Ceph version for all Ceph Monitors whenever requested

Previously, the Prometheus metrics reported mismatched Ceph versions for Ceph Monitors when the monitor was upgraded. As a result, the active Ceph Manager daemon needed to be restarted to resolve this inconsistency. Additionally, the Ceph manager used to update the monitor metadata through the **handle_mon_map** parameter, which gets triggered when the monitors are removed or added from the cluster or the active **mgr** is restarted or **mgr** failsover.

With this fix, instead of relying on fetching **mon** metadata using **ceph mon metadata** command, MON now explicitly sends metadata update requests with **mon** metadata to **mgr**.

(BZ#2121265)

The correct set of replicas are used for **remapped** placement groups

Previously, for **remapped** placement groups, the wrong set of replicas would be queried for the scrub information causing a failure of the scrub process, after identifying mismatches that would not exist.

With this fix, the correct set of replicas are now queried.

(BZ#2130666)

The targeted **rank_removed** no longer gets stuck in **live_pinging** and **dead_pinging** states

Previously, in some cases, the **paxos_size** of the Monitor Map would get updated before the rank of the monitor was changed. For example, **paxos_size** would get reduced from 5 to 4, but the highest rank of the Monitors was still 4, thus the old code would skip deleting the rank from **dead_pinging** state. This would cause the targeted rank to remain in **dead_pinging** forever, which would then cause strange **peer_tracker** scores in **election strategy: 3**.

With this fix, a case is added when **rank_removed == paxos_size()** that erases the targeted **rank_removed** from both the **live_pinging** and **dead_pinging** states and the rank does not get stuck forever in either of these sets.

(BZ#2142143)

Ceph Monitors are not stuck during failover of a site

Previously, the **removed_ranks** variable would not discard its content for every update of the Monitor map. Thus it would replace monitors in a 2-site stretch cluster and fail over of one of the site would cause connection scores, including ranks associated with the scores, to be inconsistent.

Inconsistent connection scores would cause deadlock during the monitor election period, which would result in Ceph to become unresponsive. Once this happened, there was no way for the monitor rank associated with the connection score to correct itself.

With this fix, the **removed_ranks** variable gets cleared with every update of the monitor map. Monitors are no longer stuck in the election period and Ceph no longer becomes unresponsive when replacing monitors and failing over a site. Moreover, there is a way to manually force the connection scores to correct themselves with the **ceph daemon mon.*NAME* connection scores reset** command.

(BZ#2142983)

Users are now able to set the replica **size** to 1

Previously, users were unable to set the pool **size** to **1**. The **check_pg_num()** function would incorrectly calculate the projected placement group number of the pool, which resulted in an underflow. Because of the false result, it appeared that the **pg_num** was larger than the maximum limit.

With this fix, the recent **check_pg_num()** function edits are reverted and the calculation is now working properly without resulting in an underflow and the users are now able to set the replica size to **1**.

(BZ#2153654)

Ceph cluster issues a health warning if the **require-osd-release** flag is not set to the appropriate release after a cluster upgrade.

Previously, the logic in the code that detects the **require-osd-release** flag mismatch after an upgrade was inadvertently removed during a code refactoring effort. Since the warning was not raised in the **ceph -s** output post an upgrade, any change made to the cluster without setting the flag to the appropriate release resulted in issues, such as, placement groups (PG) stuck in certain states, excessive Ceph process memory consumption, slow requests, among many other issues.

With this fix, the Ceph cluster issues a health warning if the **require-osd-release** flag is not set to the appropriate release after a cluster upgrade.

(BZ#1988773)

4.8. NFS GANESHA

The NFS-Ganesha package is based on 4.0.8 version

With this release, the **nfs-ganesha** package is now based on the upstream version 4.0.8, which provides a number of bug fixes and enhancements from the previous version.

(BZ#2121236)

CHAPTER 5. TECHNOLOGY PREVIEWS

This section provides an overview of Technology Preview features introduced or updated in this release of Red Hat Ceph Storage.

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

QoS in the Ceph OSD is based on the mClock algorithm

Previously, the scheduler defaulted to the Weighted Priority Queue (WPQ). Quality of service (QoS) based on the mClock algorithm was in an experimental phase and was not yet recommended for production.

With this release, the mClock based operation queue enables QoS controls to be applied to Ceph OSD specific operations, such as client input and output (I/O) and recovery or backfill, as well as other background operations, such as **pg scrub**, **snap trim**, and **pg deletion**. The allocation of resources to each of the services is based on the input and output operations per second (IOPS) capacity of each Ceph OSD and is achieved using built-in mClock profiles.

Also, this release includes the following enhancements:

Hands-off automated baseline performance measurements for the OSDs determine Ceph OSD IOPS capacity with safeguards to fallback to default capacity when an unrealistic measurement is detected. Setting sleep throttles for background tasks is eliminated. Higher default values for recoveries and max backfills options with the ability to override them using an override flag. Configuration sets using mClock profiles hide complexity of tuning mClock and Ceph parameters.

See *The mClock OSD scheduler* in Red Hat Ceph Storage Administration Guide for details.

Users can archive older data to an AWS bucket

With this release, users can enable data transition to a remote cloud service, such as Amazon Web Services (AWS), as part of the lifecycle configuration. See the *Transitioning data to Amazon S3 cloud service* for more details.

Expands the application of S3 select to Apache Parquet format

With this release, there are now two S3 select workflows, one for CSV and one for Parquet, that provide S3 select operations with CSV and Parquet objects. See the *S3 select operations* in the *Red Hat Ceph Storage Developer Guide* for more details.

Bucket granular multi-site sync policies is now supported

Red Hat now supports bucket granular multi-site sync policies. See the *Using multi-site sync policies* section in the *Red Hat Ceph Storage Object Gateway Guide* for more details.

Server-Side encryption is now supported

With this release, Red Hat provides the support to manage Server-Side encryption. This enables S3 users to protect data at rest with a unique key through Server-Side encryption with Amazon S3-managed encryption keys (SSE-S3).

Users can use the **PutBucketEncryption S3** feature to enforce object encryption

Previously, to enforce object encryption in order to protect data, users were required to add a header to each request which was not possible in all cases.

With this release, Ceph Object Gateway is updated to support **PutBucketEncryption S3** action. Users can use the **PutBucketEncryption S3** feature with the Ceph Object Gateway without adding headers to each request. This is handled by the Ceph Object Gateway.

CHAPTER 6. KNOWN ISSUES

This section documents known issues found in this release of Red Hat Ceph Storage.

6.1. THE CEPHADM UTILITY

The Cephadm fails the upgrade of the Red Hat Ceph Storage version if the OS is unsupported

Currently, Cephadm does not manage the host operating system (OS), therefore, during an upgrade, it does not verify if the Red Hat Ceph Storage version the user is upgrading to is supported on the OS of the Ceph cluster nodes.

As a workaround, you can manually check the OS on which the Red Hat Ceph Storage version is supported, and follow the recommended upgrade path for OS and Red Hat Ceph Storage versions. This ensures that the Cephadm upgrades the cluster without raising any warning or error, even when the host OS of the nodes is unsupported for that Red Hat Ceph Storage release.

(BZ#2161325)

6.2. CEPH OBJECT GATEWAY

Resharding a bucket removes the bucket’s metadata

Resharding a bucket removes the bucket’s metadata if the bucket was created with **bucket_index_max_shards** as **0**. You can recover the affected buckets by restoring the bucket index.

The recovery can be done in two ways:

- By executing **radosgw-admin object reindex --bucket *BUCKET_NAME* --object *OBJECT_NAME*** command.
- By executing the script **rgw-restore-bucket-index [--proceed] *BUCKET_NAME* [*DATA_POOL_NAME*]**. This script in turn invokes **radosgw-admin object reindex ...**.

Post performing the above steps, ensure to perform either a **radosgw-admin bucket list** or **radosgw-admin radoslist** command on the bucket for the bucket stats to correctly reflect the number of objects in the bucket.

NOTE

Prior to the execution of the script, perform **microdnf install jq** inside the **cephadm** shell. The tool does not work for versioned buckets.

```
[root@argo031 ~]# time rgw-restore-bucket-index  --proceed serp-bu-ver-1
default.rgw.buckets.data

NOTICE: This tool is currently considered EXPERIMENTAL.
`marker` is e871fb65-b87f-4c16-a7c3-064b66feb1c4.25076.5.
`bucket_id` is e871fb65-b87f-4c16-a7c3-064b66feb1c4.25076.5.

Error: this bucket appears to be versioned, and this tool cannot work with versioned
buckets.
```

The tool's scope is limited to a single site only and not on a multi-site. If you execute the **rgw-restore-bucket-index** tool at site-1, it does not recover objects on site-2 and vice versa. On a multi-site, the recovery tool and the object reindex command should be executed at both sites for a bucket.

(BZ#2178991)

6.3. CEPH DASHBOARD

The Red Hat Ceph StorageDashboard shows **NaN undefined** in some fields in the host table

Currently, when a new host is added, it takes some time to load its daemons, devices and other stats. During this time delay, data may not be available for some fields in the host table, as a result, during expansion, the host adds **NAN undefined** for those fields.

When the data is not available for some field in the host table, it shows N/A. Presently, there is no workaround for this issue.

(BZ#2046214)

"Throughput-optimized" option is recommended for clusters containing SSD and NVMe devices

Previously, whenever the cluster had either only SSD devices or both SSDs and NVMe devices, the "Throughput-optimized" option would be recommended, even though it should not be and it had no impact either on the user or the cluster.

As a workaround, users can use the "Advanced" mode for deploying OSDs according to their desired specifications and all the options in the "Simple" mode are still usable apart from this UI issue.

(BZ#2101680)

6.4. MULTI-SITE CEPH OBJECT GATEWAY

Multi-site replication may stop during upgrade

Multi-site replication may stop if clusters are on different versions during the process of an upgrade. We would need to suspend sync until both clusters are upgraded to the same version.

(BZ#2178909)

6.5. RADOS

The mclock_scheduler has performance issues with small object workloads and OSDs created on HDD devices

The **mclock_scheduler** has performance issues with small object workloads and with OSDs created on HDD devices. Due to this, with small object workloads, client throughput is impacted due to on-going recovery operations.

(BZ#2174467)

The Ceph OSD benchmark test might get skipped

Currently, the Ceph OSD benchmark test boot-up might sometimes not run even with the **osd_mclock_force_run_benchmark_on_init** parameter set to **true**. As a consequence, the **osd_mclock_max_capacity_iops_[hdd,ssd]** parameter value is not overridden with the default values.

As a workaround, perform the following steps:

1. Set **osd_mclock_force_run_benchmark_on_init** to **true**:

   Example

   ```
   [ceph: root@host01 /]# ceph config set osd osd_mclock_force_run_benchmark_on_init true
   ```

2. Remove the value on the respective OSD:

   Syntax

   ```
   ceph config rm OSD.OSD_ID osd_mclock_max_capacity_iops_[hdd,ssd]
   ```

   Example

   ```
   [ceph: root@host01 /]# ceph config rm osd.0 osd_mclock_max_capacity_iops_hdd
   ```

3. Restart the OSD

This results in the **osd_mclock_max_capacity_iops_[ssd,hdd]** parameter being either set with the default value or the new value if it is within the threshold setting.

(BZ#2126559)

CHAPTER 7. REMOVED FUNCTIONALITY

This section provides an overview of functionality that has been removed in this release of Red Hat Ceph Storage.

7.1. ISCSI

RBD iSCSI gateway support is now retired.

With this release onwards, Red Hat Ceph Storage will no longer ship with iSCSI gateway components. RBD iSCSI gateway has been dropped in favor of the future RBD NVMEoF gateway.

(BZ#2089287)

CHAPTER 8. SOURCES

The updated Red Hat Ceph Storage source code packages are available at the following location:

- For Red Hat Enterprise Linux 8: http://ftp.redhat.com/redhat/linux/enterprise/8Base/en/RHCEPH/SRPMS/
- For Red Hat Enterprise Linux 9: https://ftp.redhat.com/redhat/linux/enterprise/9Base/en/RHCEPH/SRPMS/