OpenShift Container Platform 3.9

Release Notes

Last Updated: 2019-02-20

Legal Notice

Abstract

Table of Contents

CHAPTER 1. OVERVIEW

The following release notes for OpenShift Container Platform 3.9 summarize all new features, major corrections from the previous version, and any known bugs upon general availability.

1.1. 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).

The OpenShift Container Platform version must match between master and node hosts, excluding temporary mismatches during cluster upgrades. For example, in a 3.9 cluster, all masters must be 3.9 and all nodes must be 3.9. However, OpenShift Container Platform will continue to support older **oc** clients against newer servers. For example, a 3.4 **oc** will work against 3.3, 3.4, and 3.5 servers.

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

Table 1.1. Compatibility Matrix

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

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

1 Fully compatible.

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

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

CHAPTER 2. OPENSHIFT CONTAINER PLATFORM 3.9 RELEASE NOTES

2.1. OVERVIEW

Red Hat OpenShift Container Platform provides developers and IT organizations with a cloud application platform for deploying new 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, Ruby, and PHP.

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

2.2. ABOUT THIS RELEASE

Red Hat OpenShift Container Platform version 3.9 (RHBA-2018:0489) is now available. This release is based on OpenShift Origin 3.9. New features, changes, bug fixes, and known issues that pertain to OpenShift Container Platform 3.9 are included in this topic.

To better synchronize versions of OpenShift Container Platform with Kubernetes, Red Hat did not publicly release OpenShift Container Platform 3.8 and, instead, is releasing OpenShift Container Platform 3.9 directly after version 3.7. See Installation for information on how this impacts installation and upgrade processes.

OpenShift Container Platform 3.9 is supported on RHEL 7.3, 7.4, and 7.5 with the latest packages from Extras, including Docker 1.13. It is also supported on Atomic Host 7.4.5 and newer. The **docker-latest** package is now deprecated.

TLSV1.2 is the only supported security version in OpenShift Container Platform version 3.4 and later. You must update if you are using TLSV1.0 or TLSV1.1.

For initial installations, see the Installing a Cluster topics in the Installation and Configuration documentation.

To upgrade to this release from a previous version, see the Upgrading Clusters topic.

2.3. NEW FEATURES AND ENHANCEMENTS

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

2.3.1. Container Orchestration

2.3.1.1. Soft Image Pruning

Now, when pruning images, you do not have to remove the actual image, just update etcd storage.

It is safer to run `--keep-tag-revisions` and `--keep-younger-than`. After this is run, administrators can choose to run hard prune (which is safe to run as long as the registry is put in read-only mode).

2.3.1.2. Red Hat CloudForms Management Engine 4.6 Container Management

The installation playbooks in OpenShift Container Platform 3.9 have been updated to support Red Hat CloudForms Management Engine (CFME) 4.6, which is now currently available. See the new Deploying Red Hat CloudForms on OpenShift Container Platform topics for further information.

In addition, this release includes the following new features and updates:

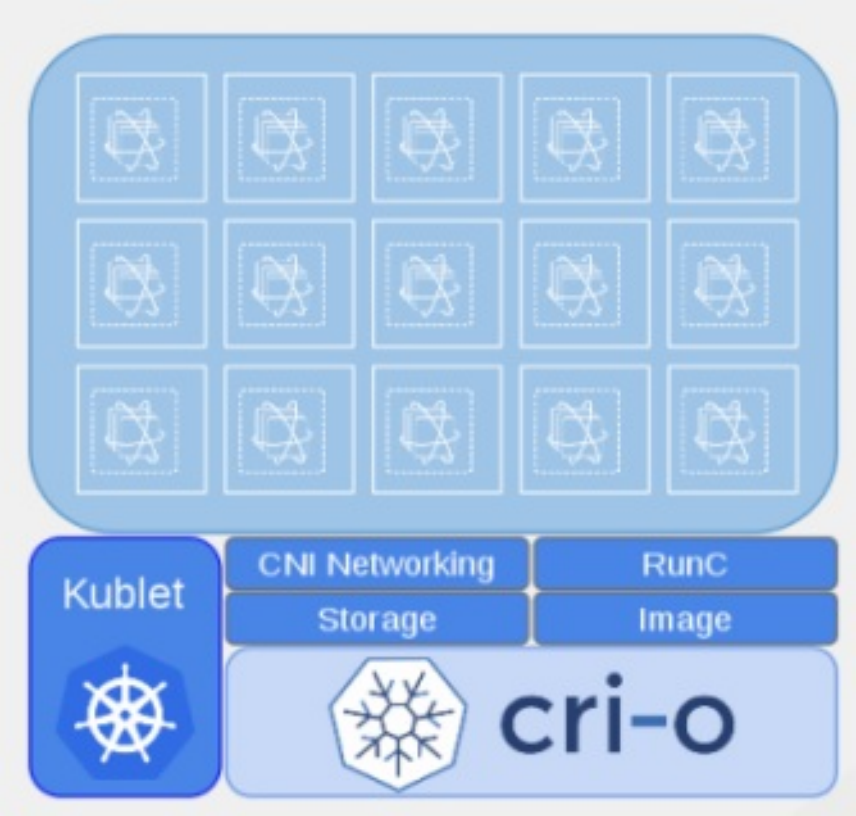
- OpenShift Container Platform template provisioning
- Offline OpenScapScans
- Alert management: You can choose Prometheus (currently in Technology Preview) and use it in CloudForms.
- Reporting enhancements
- Provider updates
- Chargeback enhancements
- UX enhancememts

2.3.1.3. CRI-O v1.9

CRI-O is a lightweight, native Kubernetes container runtime interface. By design, it provides only the runtime capabilities needed by the kubelet. CRI-O is designed to be part of Kubernetes and evolve in lock-step with the platform.

CRI-O brings:

- A minimal and secure architecture.
- Excellent scale and performance.
- The ability to run any Open Container Initiative (OCI) or docker image.
- Familiar operational tooling and commands.



To install and run CRI-O alongside **`docker`**, set the following in the **`[OSEv3:vars]`** section Ansible inventory file during cluster installation:

```
openshift_use_crio=true
openshift_crio_use_rpm=true (1)
```

(1) CRI-O can only be installed as an RPM. The previously-available system container for CRI-O has been dropped from Technology Preview as of OpenShift Container Platform 3.9.

NOTE

The **`atomic-openshift-node`** service must be RPM- or system container-based when using CRI-O; it cannot be **`docker`** container-based. The installer protects against using CRI-O with **`docker`** container nodes and will halt installation if detected.

When CRI-O use is enabled, it is installed alongside **`docker`**, which currently is required to perform build and push operations to the registry. Over time, temporary **`docker`** builds can accumulate on nodes. You can optionally set the following parameter to enable garbage collection. You must configure the node selector so that the garbage collector runs only on nodes where **`docker`** is configured for **`overlay2`** storage.

```
openshift_crio_enable_docker_gc=true
openshift_crio_docker_gc_node_selector={'runtime': 'cri-o'}
```

For example, the above would ensure it is only run on nodes with the **`runtime: cri-o`** label. This can be helpful if you are running CRI-O only on some nodes, and others are only running **`docker`**.

See the upstream documentation for more information on CRI-O.

2.3.2. Storage

2.3.2.1. PV Resize

You can expand persistent volume claims online from OpenShift Container Platform for CNS glusterFS, Cinder, and GCE PD.

1. Create a storage class with **`allowVolumeExpansion=true`**.
2. The PVC uses the storage class and submits a claim.
3. The PVC specifies a new increased size.
4. The underlying PV is resized.

2.3.2.2. End-to-end Online Expansion and Resize for Containerized GlusterFS PV

You can expand persistent volume claims online from OpenShift Container Platform for CNS glusterFS volumes.

This can be done online from OpenShift Container Platform. Previously, this was only available from the Heketi CLI. You edit the PVC with the new size, triggering a PV resize. This is fully qualified for glusterFs backed PVs. Gluster-block PV resize was added with RHEL 7.5.

1. Add **`allowVolumeExpansion=true`** to the storage class.
2. Run:

```
$ oc edit pvc claim-name
```

3. Edit the **`spec.resources.requests.storage`** field with the new value.

2.3.2.3. Container Native Storage GlusterFS PV Consumption Metrics Available from OpenShift Container Platform

Container Native Storage GlusterFS is extended to provide volume metrics (including consumption) through Prometheus or Query.

Metrics are available from the PVC endpoint. This adds visibility to what is being allocated and what is being consumed. Previously, you could only see allocated size of the PVs. Now, you know how much is really consumed so, if needed, you can expand it before it runs out of space. This also allows administrators to do billing based on consumption, if needed.

Examples of added metrics include:

- **`kubelet_volume_stats_capacity_bytes`**
- **`kubelet_volume_stats_inodes`**
- **`kubelet_volume_stats_inodes_free`**
- **`kubelet_volume_stats_inodes_used`**
- **`kubelet_volume_stats_used_bytes`**

2.3.2.4. Automated CNS Deployment with OpenShift Container Platform Advanced Installation

In the OpenShift Container Platform advanced installer, the CNS block provisioner deployment is fixed and the CNS Un-install Playbook is added. This resolves the issue of CNS block deployment with OpenShift Container Platform and also provides a way to uninstall a failed installation of CNS.

CNS storage device details are added to the installer's inventory file. The advanced installer manages configuration and deployment of CNS, file and block provisioners, registry, and ready-to-use PVs.

2.3.2.5. Tenant-driven Storage Snapshotting (Technology Preview)

Tenant-driven storage snapshotting is currently in Technology Preview and not for production workloads.

Tenants now have the ability to leverage the underlying storage technology backing the persistent volume (PV) assigned to them to make a snapshot of their application data. Tenants can also now restore a given snapshot from the past to their current application.

An external provisioner is used to access the EBS, GCE pDisk, and HostPath, and Cinder snapshotting API. This Technology Preview feature has tested EBS and HostPath. The tenant must stop the pods and start them manually.

1. The administrator runs an external provisioner for the cluster. These are images from the Red Hat Container Catalog.
2. The tenant made a PVC and owns a PV from one of the supported storage solutions.The administrator must create a new **`StorageClass`** in the cluster with:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: snapshot-promoter
provisioner: volumesnapshot.external-storage.k8s.io/snapshot-
promoter
```

3. The tenant can create a snapshot of a PVC named **`gce-pvc`** and the resulting snapshot will be called **`snapshot-demo`**.

```
$ oc create -f snapshot.yaml

apiVersion: volumesnapshot.external-storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: snapshot-demo
  namespace: myns
spec:
  persistentVolumeClaimName: gce-pvc
```

4. Now, they can restore their pod to that snapshot.

```
$ oc create -f restore.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: snapshot-pv-provisioning-demo
```

```
  annotations:
    snapshot.alpha.kubernetes.io/snapshot: snapshot-demo
spec:
  storageClassName: snapshot-promoter
```

2.3.3. Scale

2.3.3.1. Cluster Limits

Updated guidance around Cluster Limits for OpenShift Container Platform 3.9 is now available.

2.3.3.2. Device Plug-ins (Technology Preview)

This is a feature currently in Technology Preview and not for production workloads.

Device plug-ins allow you to use a particular device type (GPU, InfiniBand, or other similar computing resources that require vendor-specific initialization and setup) in your OpenShift Container Platform pod without needing to write custom code. The device plug-in provides a consistent and portable solution to consume hardware devices across clusters. The device plug-in provides support for these devices through an extension mechanism, which makes these devices available to containers, provides health checks of these devices, and securely shares them.

A device plug-in is a gRPC service running on the nodes (external to **`atomic-openshift-node.service`**) that is responsible for managing specific hardware resources.

See the Developer Guide for further conceptual information about Device Plug-ins.

2.3.3.3. CPU Manager (Technology Preview)

CPU Manager is a feature currently in Technology Preview and not for production workloads.

CPU Manager manages groups of CPUs and constrains workloads to specific CPUs.

CPU Manager is useful for workloads that have some of these attributes:

- Require as much CPU time as possible.
- Are sensitive to processor cache misses.
- Are low-latency network applications.
- Coordinate with other processes and benefit from sharing a single processor cache.

See Using CPU Manager for more information.

2.3.3.4. Device Manager (Technology Preview)

Device Manager is a feature currently in Technology Preview and not for production workloads.

Some users want to set resource limits for hardware devices within their pod definition and have the scheduler find the node in the cluster with those resources. While at the same time, Kubernetes needed a way for hardware vendors to advertise their resources to the kubelet without forcing them to change core code within Kubernetes

The kubelet now houses a device manager that is extensible through plug-ins. You load the driver

support at the node level. Then, you or the vendor writes a plug-in that listens for requests to stop/start/attach/assign the requested hardware resources seen by the drivers. This plug-in is deployed to all the nodes via a daemonSet.

See Using Device Manager for more information.

2.3.3.5. Huge Pages (Technology Preview)

Huge pages is a feature currently in Technology Preview and not for production workloads.

Memory is managed in blocks known as pages. On most systems, a page is 4Ki. 1Mi of memory is equal to 256 pages; 1Gi of memory is 256,000 pages, and so on. CPUs have a built-in memory management unit that manages a list of these pages in hardware. The Translation Lookaside Buffer (TLB) is a small hardware cache of virtual-to-physical page mappings. If the virtual address passed in a hardware instruction can be found in the TLB, the mapping can be determined quickly. If not, a TLB miss occurs, and the system falls back to slower, software-based address translation, resulting in performance issues. Since the size of the TLB is fixed, the only way to reduce the chance of a TLB miss is to increase the page size.

A huge page is a memory page that is larger than 4Ki. On x86_64 architectures, there are two common huge page sizes: 2Mi and 1Gi. Sizes vary on other architectures. In order to use huge pages, code must be written so that applications are aware of them. Transparent Huge Pages (THP) attempt to automate the management of huge pages without application knowledge, but they have limitations. In particular, they are limited to 2Mi page sizes. THP can lead to performance degradation on nodes with high memory utilization or fragmentation due to defragmenting efforts of THP, which can lock memory pages. For this reason, some applications may be designed to (or recommend) usage of pre-allocated huge pages instead of THP.

In OpenShift Container Platform, applications in a pod can allocate and consume pre-allocated huge pages.

See Managing Huge Pages for more information.

2.3.4. Networking

2.3.4.1. Semi-automatic Namespace-wide Egress IP

All outgoing external connections from a project share a single, fixed source IP address and send all traffic via that IP, so that external firewalls can recognize the application associated with a packet.

It is *semi-automatic* in that in the first half of implementing the automatic namespace-wide egress IP feature, it implements the "traffic" side. Namespaces with automatic egress IPs will send all traffic via that IP. However, it does not implement the "management" side. Nothing automatically assigns egress IPs to nodes yet. The administrator must do that manually.

See Managing Networking for more information.

2.3.4.2. Support Our Own HAProxy RPM for Consumption by the Router

Route configuration changes and process upgrades performed under heaving load have typically required a stop and start sequence of certain services, causing temporary outages.

In OpenShift Container Platform 3.9, HAProxy 1.8 sees no difference between updates and upgrades; a new process is used with a new configuration, and the listening socket's file descriptor is transferred from the old to the new process so the connection is never closed. The change is seamless, and enables our

ability to do things, like HTTP/2, in the future.

2.3.5. Master

2.3.5.1. StatefulSets, DaemonSets, and Deployments Now Supported

In OpenShift Container Platform, statefulsets, daemonsets, and deployments are now stable, supported, and out of Technology Preview.

2.3.5.2. Central Audit Capability

Provides auditing of items that administrators would like to see, including:

- The event timestamp.
- The activity that generated the entry.
- The API endpoint that was called.
- The HTTP output.
- The item changed due to an activity, with details of the change.
- The user name of the user that initiated an activity.
- The name of the namespace the event occurred in, where possible.
- The status of the event, either success or failure.

Provides auditing of items that administrators would like to trace, including:

- User login and logout from (including session timeout) the web interface, including unauthorized access attempts.
- Account creation, modification, or removal.
- Account role or policy assignment or de-assignment.
- Scaling of pods.
- Creation of new project or application.
- Creation of routes and services.
- Triggers of builds and/or pipelines.
- Addition or removal or claim of persistent volumes.

Set up auditing in the ***master-config file***, and restart the **master-config** service:

```
auditConfig:
  auditFilePath: "/var/log/audit-ocp.log"
  enabled: true
  maximumFileRetentionDays: 10
  maximumFileSizeMegabytes: 10
  maximumRetainedFiles: 10
```

```
  logFormat: json
  policyConfiguration: null
  policyFile: /etc/origin/master/audit-policy.yaml
  webHookKubeConfig: ""
  webHookMode:
```

Example log output:

```
{"kind":"Event","apiVersion":"audit.k8s.io/v1beta1","metadata":
{"creationTimestamp":"2017-09-
29T09:46:39Z"},"level":"Metadata","timestamp":"2017-09-
29T09:46:39Z","auditID":"72e66a64-c3e5-4201-9a62-
6512a220365e","stage":"ResponseComplete","requestURI":"/api/v1/securitycon
textconstraints","verb":"create","user":
{"username":"system:admin","groups":["system:cluster-
admins","system:authenticated"]},"sourceIPs":["10.8.241.75"],"objectRef":
{"resource":"securitycontextconstraints","name":"scc-
lg","apiVersion":"/v1"},"responseStatus":{"metadata":{},"code":201}}
```

2.3.5.3. Add Support for Deployments to oc status

The **`oc status`** command provides an overview of the current project. This provides similar output for upstream deployments as can be seen for downstream DeploymentConfigs, with a nested deployment set:

```
$ oc status
In project My Project (myproject) on server https://127.0.0.1:8443

svc/ruby-deploy - 172.30.174.234:8080
  deployment/ruby-deploy deploys istag/ruby-deploy:latest <-
    bc/ruby-deploy source builds https://github.com/sclorg/ruby-ex.git on
istag/ruby-22-centos7:latest
      build #1 failed 5 hours ago - bbb6701: Merge pull request #18 from
durandom/master (Joe User <joeuser@users.noreply.github.com>)
    deployment #2 running for 4 hours - 0/1 pods (warning: 53 restarts)
    deployment #1 deployed 5 hours ago
```

Compare this to the output from OpenShift Container Platform 3.7:

```
$ oc status
In project dc-test on server https://127.0.0.1:8443

svc/ruby-deploy - 172.30.231.16:8080
  pod/ruby-deploy-5c7cc559cc-pvq9l runs test
```

2.3.5.4. Dynamic Admission Controller Follow-up (Technology Preview)

Dynamic Admission Controller Follow-up is a feature currently in Technology Preview and not for production workloads.

An admission controller is a piece of code that intercepts requests to the Kubernetes API server prior to persistence of the object, but after the request is authenticated and authorized. Example use cases include mutation of pod resources and security response.

See Custom Admission Controllers for more information.

2.3.5.5. Feature Gates

Platform administrators now have the ability to turn off specific features to the entire platform. This assists in the control of access to alpha, beta, or Technology Preview features in production clusters.

Feature gates use a key=value pair in the master and kubelet configuration files that describe the feature you want to block.

Control Plane: master-config.yaml

```
kubernetesMasterConfig:
  apiServerArguments:
    feature-gates:
    - CPUManager=true
```

kubelet: node-config.yaml

```
kubeletArguments:
  feature-gates:
  - DevicePlugin=true
```

2.3.6. Installation

2.3.6.1. Improved Playbook Performance

OpenShift Container Platform 3.9 introduces significant refactoring and restructuring of the playbooks to improve performance. This includes:

- Restructured playbooks to push all fact-gathering and common dependencies up into the initialization plays so they are only called once rather than each time a role needs access to their computed values.
- Refactored playbooks to limit the hosts they touch to only those that are truly relevant to the playbook.

2.3.6.2. Quick Installation (Deprecated)

Quick Installation is now deprecated in OpenShift Container Platform 3.9 and will be completely removed in a future release.

Quick installation will only be capable of installing 3.9. It will not be able to upgrade from 3.7 or 3.8 to 3.9.

2.3.6.3. Automated 3.7 to 3.9 Control Plane Upgrade

The installer automatically handles stepping the control plane from 3.7 to 3.8 to 3.9 and node upgrade from 3.7 to 3.9.

Control plane components (API, controllers, and nodes on control plane hosts) are upgraded seamlessly from 3.7 to 3.8 to 3.9. Data migration happens pre- and post- OpenShift Container Platform 3.8 and 3.9 control plane upgrades. Other control plane components (router, registry, service catalog, and brokers) are upgraded from OpenShift Container Platform 3.7 to 3.9. Nodes (node, docker, ovs) are upgraded

directly from OpenShift Container Platform 3.7 to 3.9 with only one drain of nodes. OpenShift Container Platform 3.7 nodes operate indefinitely against 3.8 masters should the upgrade process need to pause in this state. Logging and metrics are updated from OpenShift Container Platform 3.7 to 3.9.

It is recommended that you upgrade the control plane and nodes independently. You can still perform the upgrade through an all-in-one playbook, but rollback is more difficult. Playbooks do not allow for a clean installation of OpenShift Container Platform 3.8.

See Upgrading Clusters for more information.

2.3.7. Metrics and Logging

2.3.7.1. Journald for System Logs and JSON File for Container Logs

Docker log driver is set to **`json-file`** as the default for all nodes. Docker **`log-driver`** can be set to **`journal`**, but there is no log rate throttling with journal driver. So, there is always a risk for denial-of-service attacks from rogue containers.

Fluentd will automatically determine which log driver (**`journald`** or **`json-file`**) the container runtime is using. Fluentd will now always read logs from journald and also ***/var/log/containers*** (if **`log-driver`** is set to **`json-file`**). Fluentd will no longer read from ***/var/log/messages***.

See Aggregating Container Logs for more information.

2.3.7.2. Prometheus (Technology Preview)

Prometheus remains in Technology Preview and is not for production workloads. Prometheus, AlertManager, and AlertBuffer versions are now updated and node-exporter is now included:

- prometheus 2.1.0
- Alertmanager 0.14.0
- AlertBuffer 0.2
- node_exporter 0.15.2

You can deploy Prometheus on an OpenShift Container Platform cluster, collect Kubernetes and infrastructure metrics, and get alerts. You can see and query metrics and alerts on the Prometheus web dashboard. Alternatively, you can bring your own Grafana and hook it up to Prometheus.

See Prometheus on OpenShift for more information.

2.3.8. Developer Experience

2.3.8.1. Jenkins Memory Usage Improvements

Previously, Jenkins worker pods would often consume too much or too little memory. Now, a startup script intelligently looks at pod limits and environment variables are appropriately set to ensure limits are respected for spawned JVMs.

2.3.8.2. CLI Plug-ins (Technology Preview)

CLI plug-ins are currently in Technology Preview and not for production workloads.

Usually called *plug-ins* or *binary extensions*, this feature allows you to extend the default set of **oc** commands available and, therefore, allows you to perform new tasks.

See Extending the CLI for information on how to install and write extensions for the CLI.

2.3.8.3. Ability to Specify Default Tolerations via the buildconfig Defaulter

Previously, there was not a way to set a default toleration on build pods so they could be placed on build-specific nodes. The build defaulter is now updated to allow the specification of a toleration value, which is applied to the build pod upon creation.

See Configuring Global Build Defaults and Overrides for more information.

2.3.8.4. Default Hard Eviction Thresholds

OpenShift Container Platform uses the following default configuration for **`eviction-hard`**.

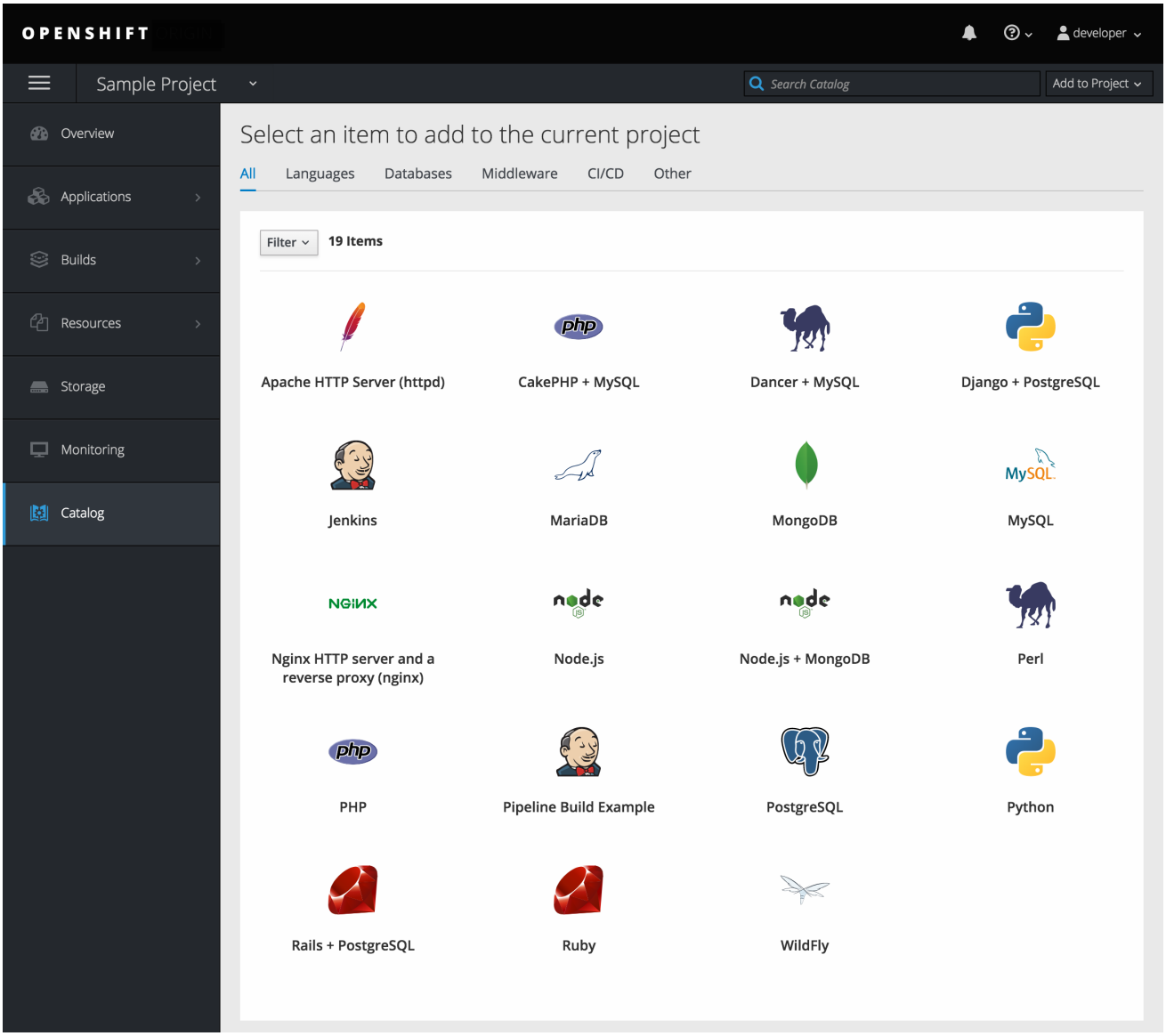
```
...
kubeletArguments:
  eviction-hard:
  - memory.available<100Mi
  - nodefs.available<10%
  - nodefs.inodesFree<5%
  - imagefs.available<15%
...
```

See Handling Out of Resource Errors for more information.

2.3.9. Web Console

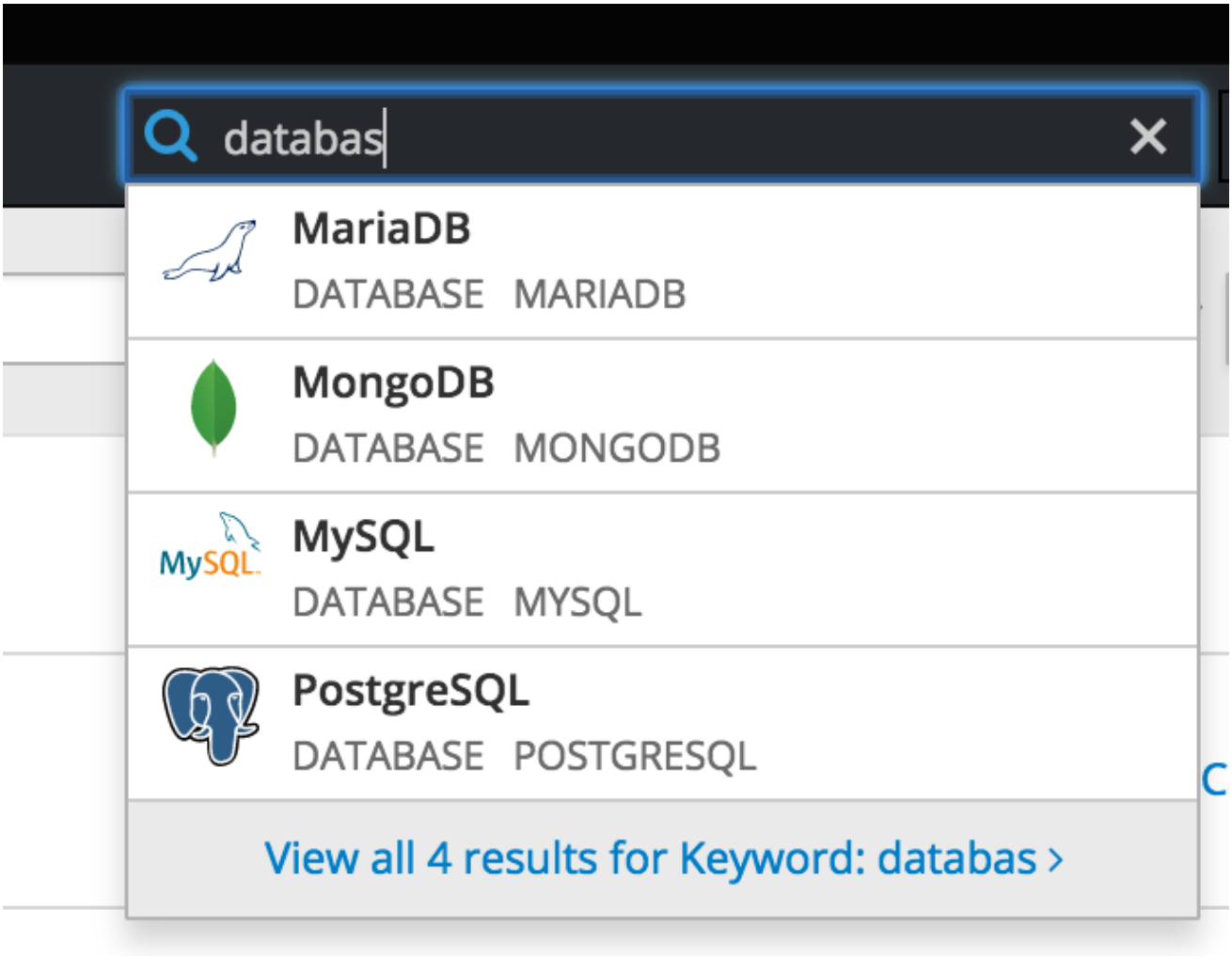
2.3.9.1. Catalog from within Project View

Quickly get to the catalog from within a project by clicking **Catalog** in the left navigation.



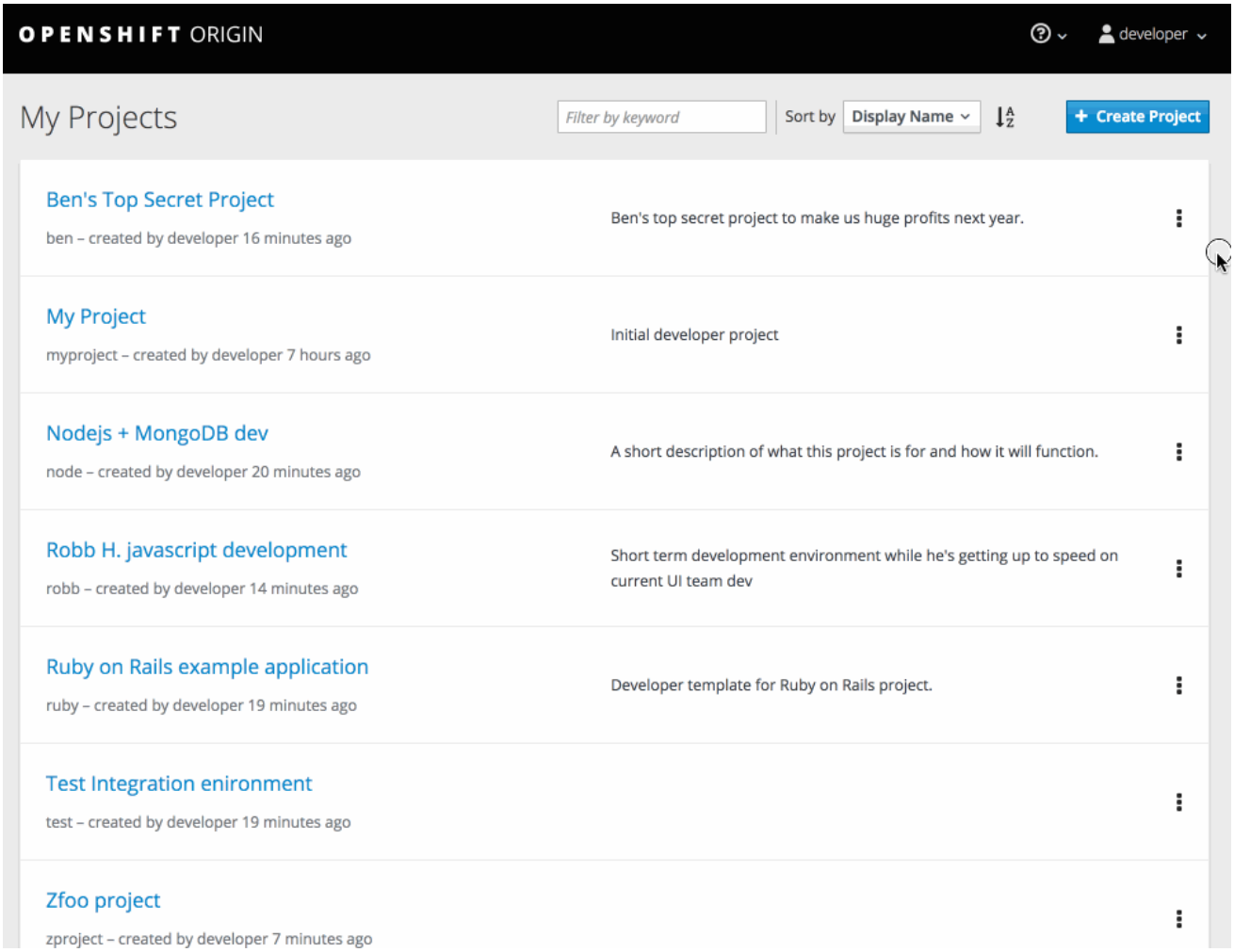
2.3.9.2. Quickly Search the Catalog from within Project View

To quickly find services from within project view, type in your search criteria.



2.3.9.3. Select Preferred Home Page

You can now jump straight to certain pages after login. Access the menu from the account dropdown, choose your option, then log out, then log back in.



2.3.9.4. Configurable Inactivity Timeout

You can now configure the web console to log users out after a set timeout. The default is `0` (never). Set the Ansible variable to the number of minutes:

```
openshift_web_console_inactivity_timeout_minutes=n
```

2.3.9.5. Web Console as a Separate Pod

The web console is now separated out of the API server. The web console is packaged as a container image and deployed as a pod. Configure via the ConfigMap. Changes are auto-detected.

Masters are now schedulable and required to be schedulable for the web consoles deployments to work.

2.4. NOTABLE TECHNICAL CHANGES

OpenShift Container Platform 3.9 introduces the following notable technical changes.

Manual Upgrade Process Now Unsupported
As of OpenShift Container Platform 3.9, manual upgrades are not supported. In a future release, this process will be removed.

Masters Marked as Schedulable Nodes by Default
In previous versions of OpenShift Container Platform, master hosts were marked as unschedulable nodes by default by the installer, meaning that new pods could not be placed on the hosts. Starting with

OpenShift Container Platform 3.9, however, masters are marked schedulable automatically during installation and upgrade. This change is mainly so that the web console, which used to run as part of the master itself, can instead be run as a pod deployed to the master.

Default Node Selector Set By Default and Automatic Node Labeling

Starting in OpenShift Container Platform 3.9, masters are now marked as schedulable nodes by default. As a result, the default node selector (defined in the master configuration file's **`projectConfig.defaultNodeSelector`** field to determine which node that projects will use by default when placing pods, and previously left blank by default) is now set by default during cluster installations and upgrades. It is set to **`node-role.kubernetes.io/compute=true`** unless overridden using the **`osm_default_node_selector`** Ansible variable.

In addition, whether **`osm_default_node_selector`** is set or not, the following automatic labeling occurs for hosts defined in your inventory file during installations and upgrades:

- non-master, non-dedicated infrastructure nodes hosts (by default, this means nodes with a **`region=infra`** label) are labeled with **`node-role.kubernetes.io/compute=true`**, which assigns the **`compute`** node role.
- master nodes are labeled with **`node-role.kubernetes.io/master=true`**, which assigns the **`master`** node role.

This ensures that the default node selector has available nodes to choose from when determining pod placement. See Configuring Node Host Labels for more details.

Ansible Must Be Installed via the rhel-7-server-ansible-2.4-rpms Channel

Starting in OpenShift Container Platform 3.9, Ansible must be installed via the **`rhel-7-server-ansible-2.4-rpms`** channel, which is included in RHEL subscriptions.

Several oc secrets Subcommands Now Deprecated

OpenShift Container Platform 3.9 deprecates the following **`oc secrets`** subcommands in favor of **`oc create secret`**:

- **`new`**
- **`new-basicauth`**
- **`new-dockercfg`**
- **`new-sshauth`**

Updated Default Values for template_service_broker_prefix and template_service_broker_image_name in the Installer

Default values for **`template_service_broker_prefix`** and **`template_service_broker_image_name`** in installer have been updated to be consistent with other settings.

Previous values are:

- **`template_service_broker_prefix="registry.example.com/openshift3/"`**
- **`template_service_broker_image_name="ose-template-service-broker"`**

New values are:

- **`template_service_broker_prefix="registry.example.com/openshift3/ose-"`**

- `template_service_broker_image_name="template-service-broker"`

Removed Several Instances of 'become: no' on Certain Tasks and Playbooks Inside of openshift-anisble

In an effort to provide greater flexibility for users, several instances of **`become: no`** on certain tasks and playbooks inside of **`openshift-anisble`** are now removed. These statements were primarily applied on **`local_action`** and **`delegate_to: localhost`** commands for creating temporary files on the host running Ansible.

If a user is running Ansible from a host that does not allow password-less **`sudo`**, some of these commands may fail if you run the **`ansible-playbook`** with the **`-b`** (**`become`**) command line switch, or if it has **`ansible_become=True`** applied to the local host in the inventory or **`group_vars`**.

Elevated permissions are not required on the local host when running **`openshift-ansible`** plays.

If target hosts (where OpenShift Container Platform is being deployed) require the use of **`become`**, it is recommended that you add **`ansible_become=True`** for those hosts or groups in inventory or **`group_vars`**/**`host_vars`**.

If the user is running as root on the local host or connection to the root user on the remote hosts instead of using become, then you should not notice a change.

Unqualified Image Specifications

Unqualified image specifications now default to **`docker.io`** and require API server configuration to resolve to different registries.

batch/v2alpha1 ScheduledJob Objects Are No Longer Supported

The **`batch/v2alpha1 ScheduledJob`** objects are no longer supported. Use CronJobs instead.

The autoscaling/v2alpha1 API Group Is Removed

The **`autoscaling/v2alpha1`** API group has been removed

Start Node Requires Swap to be Disabled

For new installations of OpenShift Container Platform 3.9 , disabling swap is a strong recommendation. For OpenShift Container Platform 3.8, the OpenShift Container Platform start node requires swap to be disabled. This is already done as part of the Ansible node installation.

oadm Command Is Deprecated

The **`oadm`** command is now deprecated. Use **`oc adm`** instead.

StatefulSets, DaemonSets, and Deployments Now Fully Supported

The core workloads API, which is composed of the **`DaemonSet`**, **`Deployment`**, **`ReplicaSet`**, and **`StatefulSet kinds`**, has been promoted to GA stability in the **`apps/v1`** group version. As such, the **`apps/v1beta2`** group version is deprecated, and all new code should use the kinds in the apps/v1 group version. For OpenShift Container Platform this means the statefulsets, daemonsets, and deployments are now stable and supported.

Administrator Solutions Guide Removed

In OpenShift Container Platform 3.9, the Administrator Solutions guide is removed from the OpenShift Container Platform documentation. See the Day Two Operations Guide instead.

2.5. BUG FIXES

This release fixes bugs for the following components:

Builds

- Previously, builds selected the secret to be used for pushing the output image at the time they were started. When a build started before the default service account secrets for a project were created, the build may not have found a suitable secret for pushing the image, resulting in the build failing when it went to push the image. With this fix, the build is held until the default service account secrets exist, ensuring that if the default secret is suitable for pushing the image, it can and will be used. As a result, initial builds in a newly created project are no longer at risk of failing if the build is created before the default secrets are populated. (**BZ#1333030**)

Command Line Interface

- The **`systemd`** units for masters changed without the diagnostics being updated. This caused the diagnostics to silently check for master **`systemd`** units that did not exist, and problems were not reported. With this fix, diagnostics check for correct master unit names and problems with master **`systemd`** units and logs may be found. (**BZ#1378883**)

Containers

- If a container shares namespace with another container, then they would share the namespace path. If you run the **`exec`** command in the first container, it only reads the namespace paths stored in the file and joins those namespaces. So, if the second container has already been stopped, the **`exec`** command in the first container will fail. As a result, this fix saves namespace paths no matter if containers share namespaces. (**BZ#1510573**)

Images

- Docker has a known "zombie process" phenomenon that impacted the OpenShift Jenkins image, causing operating system-level resources to be exhausted as these “zombie processes” accumulated. With this fix, the OpenShift Jenkins image now leverages one of the Docker image **`init`** implementations to launch Jenkins, monitor, and handle any “zombie child processes”. As a result, “zombie processes” no longer accumulate. (**BZ#1528548**)

- Due to a fault in the scheduler implementation, the **`ScheduledImageImportMinimumIntervalSeconds`** setting was not correctly observed, causing OpenShift Container Platform to attempt to import scheduled images at the wrong intervals. This is now resolved. (**BZ#1543446**)

- Previously, OpenShift would erroneously re-import all tags on an image stream, regardless if marked as scheduled or not, if any tag on the image stream was marked as scheduled. This behavior is now resolved. (**BZ#1515060**)

Image Registry

- The signature importer tried to import signatures from the internal registry without credentials, causing the registry to check if the anonymous user could get signatures using SAR requests. With this bug fix, the signature importer skips the internal registry because the internal registry and the signature importer work with the same storage, resulting in no SAR requests. (**BZ#1543122**)

- There was no check of the number of components in the path, causing the data to be placed in the storage but not be written to the database. With this bug fix, an early check of the path was added. (**BZ#1528613**)

Installer

- The Kubernetes service IP address was not added to **`no_proxy`** list for the docker-registry

during installation. As a result, internal registry requests would be forced to use the proxy, preventing logins and pushes to the internal registry. The installer was changed to add the Kubernetes service IP to the **`no_proxy`** list. (BZ#1504464)

- The installer was pulling the incorrect efs-provisioner image, which caused the installation of the provisioner pod to fail to deploy. The installer was changed to pull the correct image. (BZ#1523534)

- When installing OpenShift Container Platform with a custom registry, the installer was using the default registry. The registry console default image is now defined as a fully qualified image **`registry.access.redhat.com/openshift3/registry-console`** which means that when a custom registry is specified via **`oreg_url`** and image streams are modified to use that custom registry the registry console will also utilize the custom registry. (BZ#1523638)

- Running the ***redeploy-etcd-ca.yml*** playbook did not update the **`ca.crt`** used by etcd system container. The code was changed so that the playbook properly updates the etcd ca.crt in ***/etc/etcd/ca.crt*** as expected. (BZ#1466216)

- Following a successful deployment of CNS/CRS with glusterblock, OpenShift Container Platform logging and metrics can be deployed using glusterblock as their backend storage for fault-tolerant, distributed persistent storage. (BZ#1480835)

- When upgrading from 3.6 to 3.7, the user wanted the Hawkular OpenShift Agent pods deactivated. But, after upgrade, the HOSA pods are still being deployed. A new playbook, **uninstall_hosa.yaml**, has been created to remove HOSA from a OpenShift Container Platform cluster when **`openshift_metrics_install_hawkular_agent=false`** in the Ansible inventory file. (BZ#1497408)

- Because registry credentials for the broker were stored in a ConfigMap, sensitive credentials could be exposed in plain text. A secret is now created to store the credentials Registry credentials are no longer visible in plaintext. (BZ#1509082)

- Because of incorrect naming, the uninstall playbook did not remove the **tuned-profiles-atomic-openshift-node** package. The playbook is now corrected and the package is removed upon uninstallation of OpenShift Container Platform. (BZ#1509129)

- When running the installer with the **`openshift_hosted_registry_storage_volume_size`** parameter configured with Jnja code, the installation failed during persistent volume creation. The code is now fixed to properly interpret the Jinja code. (BZ#1518386)

- During disconnected installations, the service catalog was attempting to pull down images from the configured registry. This caused the installation to fail as the registry is not available during a disconnected installation. The **`imagePullPolicy`** in the installer was changed to **`ifNotPresent`**. If the image is present, the service catalog will not attempt to pull it again, and the disconnected installation of the service catalog will proceed. (BZ#1524805)

- When provisioning hosts with an SSH proxy configured, the masters would never appear marked as up. With this bug fix, the task is changed to use an Ansible module that respects SSH proxy configuration. As a result, Ansible is able to connect to the hosts and they are marked as up. (BZ#1541946)

- In an HTTPS environment, the service catalog installation was failing because the playbook attempted to contact the API server using cURL without the **`--noproxy`** option specified. The command in the playbook was changed to include **`--noproxy`** and the installer performs as expected. (BZ#1544645)

- Previously, the storage type for Elasticsearch data centers was not preserved when upgrading/rerunning. This caused the existing storage type to be overwritten. This bug fix preserves the storage type as the default (using an inventory variable if specified). (**BZ#1496758**)

- Previously, the docker daemon was incorrectly restarted when redeploying node certificates. This caused unnecessary downtime in nodes since **`atomic-openshift-node`** was the only component loading the kubeconfig. This bug fix adds a flag to check if a new Certificate Authority (CA) is being deployed. If not, then restarting Docker is skipped. (**BZ#1537726**)

- Previously, the **`docker_image_availability`** check did not take into account variables that override specific container images used for containerized components. This caused the check to incorrectly report failures when looking for the default images when the overridden images were actually available. As a result of this bug fix, the check should accurately report whether the necessary images are available. (**BZ#1538806**)

- When determining if a persistent volume claim (PVC) should be created for Elasticsearch, we used a legacy variable, which did not correctly evaluate if a PVC was necessary when creating a Network File System (NFS)-backed persistent volume (PV). This bug fix correctly evaluates if a PVC is necessary for the deployment configuration. (**BZ#1538995**)

- Previously, when configuring the registry for Azure Blob storage, the realm of **`core.windows.net`** was specified by default. This bug fix allows you to change **`openshift_hosted_registry_storage_azure_blob_realm`** to the value that you want to use. (**BZ#1491100**)

- A new playbook has been introduced that uninstalls an existing GlusterFS deployment. This playbook removes all existing resources, including pods and services. This playbook also, optionally, removes all data and configuration from the hosts that were running GlusterFS pods. (**BZ#1497038**)

Logging

- Previously, the OpenShift Container Platform logging system did not support CRI-O. This bug fix added a parser for CRI-O formatted logs. As a result, both system and container logs can be collected. (**BZ#1517605**)

- When redeploying logging, we previously attempted to maintain any changes that were made to the ConfigMaps post-installation. It was difficult to let users specify the contents of a ConfigMap file while still needing the ability to provide the configurations required for the different Elasticsearch, Fluentd, and Kibana (EFK) stack components. This bug fix created a patch based on changes made post-deployment and applies that patch to the files provided by the installer. (**BZ#1519619**)

Web Console

- The Kibana page previously displayed **OPENSHIFT ORIGIN** in the upper left-hand corner of the OpenShift Container Platform web console. This bug fix replaces the Origin header image with the OpenShift Container Platform header image. As a result, the Kibana page now displays the desired header. (**BZ#1546311**)

- Both the OpenShift Container Platform **`DeploymentConfig`** and Kubernetes extensions/v1beta1 Deployment resources were labeled with deployment on the web console overview, so you could not differentiate the resources. **`DeploymentConfig`** resources on the **Overview** page are now labelled with **`DeploymentConfig`**. (**BZ#1488380**)

- The web console's pod status filter did not correctly display pod init status when an error prevented the pod from initializing, including and init status of error. If a pod has an `Init:Error` status, the pod status correctly displays **Init Error** instead of **Pod Initializing**. (BZ#1512473)

- Previously, switching tabs in the web console page for a pipeline build configuration caused some content on the page to no longer be visible while the page reloaded. Switching tabs no longer reloads the entire page, and content is correctly displayed. (BZ#1527346)

- By default, an old version of the builder image was shown when you added a builder to a project and selected by default during builder configuration. This gave the wrong impression that your only choice was an old version of a language or framework. The version number is no longer shown in the wizard title, and the newest available version is selected by default. (BZ#1542669)

- If you used some browsers, you could not consistently use the right click menu to copy and paste text from internal editors that used the ACE editor library, including the YAML, Jenkinsfile, and Dockerfile editors. This update uses a newer version of the ACE editor library, so the right click menu options work throughout the console. (BZ#1463617)

- Previously, browsers would use the default behavior for the Referrer-Policy because Referrer-Policy header was not sent by the console. Now the console correctly sends the Referrer-Policy header, which is set to `strict-origin-when-cross-origin`, and browsers that listen to the Referrer-Policy header follow the `strict-origin-when-cross-origin policy` for the web console. (BZ#1504571)

- Previously, users with read access to the project saw webhook secret values because they were stored as strings in the build. These users could use these values to trigger builds even though they had only read access to the project. Now webhook secrets are defined as secret objects in the build instead of strings. Users with read only access to the project cannot see the secret values or use them to trigger builds by using the webhook. (BZ#1504819)

- Previously, adding the same persistent volume claim more than once to a deployment in the web console caused pods for that deployment to fail. The web console incorrectly created a new volume when it added the second PVC to the deployment instead of reusing the existing volume from the pod template spec. Now, the web console reuses the existing volume if the same PVC is listed more than once. This behavior lets you add the same PVC with different mount paths and subpaths as needed. (BZ#1527689)

- Previously, it was not clear enough that you can not select an **Image Name** from the Deploy Image window if you are also creating a new project. The help text that explains that you can only set an **Image Name** for existing projects is easier to find. (BZ#1535917)

- Previously, the secrets page in the web console did not display labels. You can now view the labels for a secret like other resources. (BZ#1545828)

- Sometimes the web console displayed a process template page even if you did not have permissions to process templates. If you tried to process the template, an error displayed. Now you can no longer view process templates if you cannot process them. (BZ#1510786)

- Previously, the **Clear Changes** button did not correctly clear edits to the **Environment From** variables in the web console environment variable editor. The button now correctly resets edits to **Environment From** variables. (BZ#1515527)

- By default, dialogs in the web console can be dismissed by clicking in the negative space surrounding the dialog. IAs a result, the warning dialog could be inadvertently dismissed. With this bug fix, the warning dialog's configuration was changed so that it can only be dismissed by

clicking one of the buttons in the dialog. The warning dialog can no longer be inadvertently dismissed by the user, as clicking one of the dialog's buttons is now required in order to close the dialog. (BZ#1525819)

Master

- Due to a fault in the scheduler implementation, the **`ScheduledImageImportMinimumIntervalSeconds`** setting was not correctly observed, causing OpenShift Container Platform to attempt to import scheduled images at the wrong intervals. With this bug fix, the issue is now resolved. (BZ#1515058)

Networking

- The OpenShift Container Platform node was not waiting long enough for the VNID while the master assigns the VNID and it could take a while to propagate to the node. As a result, pod creation fails. Increase the timeout from 1 to 5 seconds for fetching VNID on the node. This bug fix allows pod creation to succeed. (BZ#1509799)

- It is now possible to specify a subnet length as part of the **`EGRESS_SOURCE`** variable passed to an egress router (for example, **`192.168.1.100/24`** rather than **`192.168.1.100`**). In some network configurations (such as if the gateway address was a virtual IP that might be backed by one of several physical IPs at different times), ARP traffic between the egress router and its gateway might not function correctly if the egress router is not able to send traffic to other hosts on its local subnet. By specifying **`EGRESS_SOURCE`** with a subnet length, the egress router setup script will configure the egress pod in a way that will work with these network setups. (BZ#1527602)

- In some circumstances, iptables rules could become reordered in a way that would cause the **per-project static IP address** feature to stop working for some IP addresses. (For most users, egress IP addresses that ended with an even number would continue to work, but egress IP addresses ending with an odd number would fail.) Therefore, external traffic from pods in a project that was supposed to use a per-project static IP address would end up using the normal node IP address instead. The iptables rules are changed so that they now have the expected effect even when they get reordered. With this bug fix, the per-project static egress IP feature now works reliably. (BZ#1527642)

- Previously, the egress IP initialization code was only run when doing a full SDN setup, and not when OpenShift services were restarted and found any existing running SDN. This resulted in failure to create new per-project static egress IPs (**`HostSubnet.EgressIPs`**). This issue is now fixed and per-project static egress IPs works correctly after a node restart. (BZ#1533153)

- Previously, OpenShift was setting colliding host-subnet values, which resulted in pod IP network to became unavailable across the nodes. This was because the stale OVS rules were not cleared during node startup. This is now fixed and the stale OVS rules are cleared on node startup. (BZ#1539187)

- With previous version, if an static IP addressed was removed from a project and then added back to the same project, it did not worked correctly. This is now fixed, removing and re-adding static egress IPs works. (BZ#1547899)

- Previously, when OpenShift was deployed on OpenStack, there were few required **`iptables`** rules that were not created automatically, which resulted in errors in pop-to-pod communication between pods on different nodes. The Ansible OpenShift installer now sets the required **`iptables`** rules automatically. (BZ#1493955)

- There was a race condition in the startup code that relied on the node setup, setting a field that

the userspace proxy needed. When the network plugin was not used (or if it was fast) the userspace proxy setup ran sooner and resulted in reading a nil value for the IP address of the node. Later when the proxy (or the **`unidler`** which uses it) was enabled, it would crash because of the nil IP address value. This issue is now fixed. A retry loop is added that waits for the IP address value to be set and the userspace proxy and **`unidler`** work as expected. (BZ#1519991)

- In some circumstances, nodes were receiving a duplicate out-of-order HostSubnet **`deleted`** event from the master. During processing of this duplicate event, the node ended up deleting OVS flows corresponding to an active node, disrupting communications between these two nodes. In the latest version. the HostSubnet event-processing now checks for and ignores duplicate events. Thus, the OVS flows are not deleted, and pods communicate normally. (BZ#1544903)

- Previously, the **`openshift ex dockergc`** command to cleanup docker images, failed occasionally. This issue is now fixed. (BZ#1511852)

- Previously, nested secrets did not get mounted in pod. This issue is now fixed. (BZ#1516569)

- HAproxy versions earlier than version 1.9 dropped new connections during a reload. This issue is now fixed. By using HAproxy's seamless reload feature, HAproxy now passes open sockets when reloading, fixing reload issues. fixed. (BZ#1464657)

- There was a spurious error in system logs. The error **`Stat fs failed. Error: no such file or directory`** appeared in logs frequently. This was because of calling the **`syscall.Statfs`** function in code when the path does not exist. This issue is now fixed. (BZ#1511576)

- Previously, a reject routes error message showed up when using router shards. This issue is now fixed and the rejected routes error messages are now suppressed in HAproxy if router shards are used. (BZ#1491717)

- Previously, if creating a route with the host set to **`localhost`**, and if the **`ROUTER_USE_PROXY_PROTOCOL`** environment variable was not set to **`true`**, any route reloads would fail. This is because the hostname being set to the default resulted in mismatches in route configurations. The **`-H`** option is now available when using **`curl`**, meaning the health check does not pass the hostname when set to 'localhost', and routes reload successfully. (BZ#1542612)

- Previously, updating TLS certificates was not possible for cluster administrators. Because it is an expected task of the cluster administrator, the role has been changed to update TLS certificates. (BZ#1524707)

Service Broker

- Previously, the APBs for MariaDB, PostgreSQL, and MySQL were tagged as "databases" instead of "database". This is corrected with the tag "database" matching other services which is now properly shown in search results. (BZ#1510804)

- Async bind and unbind is an experimental feature for the OpenShift Ansible broker (OAB) and is not supported or enabled by default. Red Hat's officially released APBs (PostgreSQL, MariaDB, MySQL, and Mediawiki) also do not support async bind and unbind. (BZ#1548997)

- Previously, the etcd server was not accessible when using the **`etcdctl`** command. This was caused by the tcp being set to "0.0.0.0" instead of the expected **`--advertise-client-urls`** value of the **`asb-etcd`** deployment configuration. The command had been updated and the etcd server is now accessible. (BZ#1514417)

- Previously, the **`apb push -o`** command failed when using it outside the cluster. This was because the Docker registry service of the desired service was set to hit only the route used by internal operations. The appropriate Ansible playbook has been updated to point to the appropriate route instead. (**BZ#1519193**)

- Previously, when typing **`asbd --help`** or **`asbd -h`**, the **`--help`** argument returned a code that was being misinterpreted as an error, resulting in errors printing out twice. The fix corrects errors to only print once and also to interpret the help command return code as valid. As a result, the help command now only prints once. (**BZ#1525817**)

- Previously, setting the **`white-list`** variable in an RHCC registry would maintain searching for any options, even after those options are removed from the configuration. This was caused by an error in the **`white-list`** code. The error has been fixed by this bug. (**BZ#1526887**)

- Previously, if the registry configuration did not have **`auth_type`** set to **`config`** error messages would appear. This bug ensures that registry configurations work correctly without the **`auth_type`** setting. (**BZ#1526949**)

- Previously, the broker would return a 400 status code when the user did not have the permissions to execute a task instead of the 403 status code. This bug fixes the error, and the correct status code is now returned. (**BZ#1510486**)

- Previously, any MariaDB configuration options were displayed with MySQL options. This is because MariaDB uses MySQL variables upstream. This bug fix ensures that, in terms of OpenShift, the variables are called out as MariaDB. (**BZ#1510294**)

Storage

- Previously, OpenShift checked mounted NFS volume with root squash. OpenShift permissions while running as root were squashed to the 'nobody' user, who did not have permissions to access mounted NFS volume. This caused any OpenShift checks to fail, and it did not unmount NFS volumes. Now, OpenShift does not access mounted NFS volumes, and checks for mounts by parsing /proc filesystem. NFS volumes with root squash option are unmounted. (**BZ#1518237**)

- Previously, when a node that had an OpenStack Cinder type of persistent volume attached was shut down or crashed, the attached volume did not detach.Consequence: Because the persistent volume was unavailable, the pods did not migrate from the failed node, and the volumes were inaccessible from other nodes and pods. Now a node fails, all of its attached volumes are detached after a time-out. (**BZ#1523142**)

- Previously, downward API, secrets, ConfigMap, and projected volumes fully managed their content and did not allow any other volumes to be mounted on top of them. This meant that users could not mount any volume on top of the aforementioned volumes. With this bug fix, the volumes now touch only the files they create. As a result, users can mount any volume on top of the aforementioned volumes. (**BZ#1430322**)

Upgrade

- The upgrade playbooks did not previously regenerate the registry certificate when upgrading from releases prior to 3.6, which lacked the name 'docker-registry.default.svc'. As such, the configuration variables were not updated to push to the registry via DNS. The 3.9 upgrade playbooks now regenerate the certificate when needed, ensuring that all environments upgraded to 3.9 now push to the registry via DNS. (**BZ#1519060**)

- The etcd host validation now accepts one or more etcd hosts, allowing greater flexibility in the number of etcd hosts configured. The recommended number of etcd hosts is still 3. (**BZ#1506177**)

2.6. TECHNOLOGY PREVIEW FEATURES

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

Technology Preview Features Support Scope

In the table below, features marked **TP** indicate *Technology Preview* and features marked **GA** indicate *General Availability*.

Table 2.1. Technology Preview Tracker

Feature	OCP 3.6	OCP 3.7	OCP 3.9
Prometheus Cluster Monitoring	-	TP	TP
Local Storage Persistent Volumes	-	TP	TP
CRI-O for runtime pods	-	TP	GA* [a]
Tenant Driven Snapshotting	-	TP	TP
oc CLI Plug-ins	-	TP	TP
Service Catalog	TP	GA	-
Template Service Broker	TP	GA	-
OpenShift Ansible Broker	TP	GA	-
Network Policy	TP	GA	-
Service Catalog Initial Experience	TP	GA	-
New Add Project Flow	TP	GA	-
Search Catalog	TP	GA	-
CFME Installer	TP	GA	-

Feature	OCP 3.6	OCP 3.7	OCP 3.9
Cron Jobs	TP	TP	GA
Kubernetes Deployments	TP	TP	GA
StatefulSets	TP	TP	GA
Explicit Quota	TP	TP	GA
Mount Options	TP	TP	GA
System Containers for docker, CRI-O	TP	TP	Dropped
System Container for installer and Kubelet	TP	TP	GA
Hawkular Agent	TP	Dropped	
Pod PreSets	TP	Dropped	
experimental-qos-reserved	-	TP	TP
Pod sysctls	TP	TP	TP
Central Audit	-	TP	GA
Static IPs for External Project Traffic	-	TP	GA
Template Completion Detection	-	TP	GA
`replicaSet`	TP	TP	GA
Mux	-	TP	TP
Clustered MongoDB Template	TP	Community	-
Clustered MySQL Template	TP	Community	-
Image Streams with Kubernetes Resources	TP	TP	GA

Feature	OCP 3.6	OCP 3.7	OCP 3.9
Device Manager	-	-	TP
Persistent Volume Resize	-	-	TP
Huge Pages	-	-	TP
CPU Manager	-	-	TP
Device Plug-ins	-	-	TP
syslog Output Plug-in for fluentd	-	-	GA

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

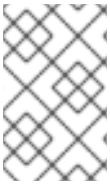
2.7. KNOWN ISSUES

- There is a known issue in the initial GA release of OpenShift Container Platform 3.9 that causes the installation and upgrade playbooks to consume more memory than previous releases. The node scale-up and installation Ansible playbooks may have consumed more memory on the control host (the system where you run the playbooks from) than expected due to the use of `include_tasks` in several places. This issue has been addressed with the release of RHBA-2018:0600; the majority of these instances have now been converted to `import_tasks` calls, which do not consume as much memory. After this change, memory consumption on the control host should be below 100MiB per host; for large environments (100+ hosts), a control host with at least 16GiB of memory is recommended. (**BZ#1558672**)

2.8. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Container Platform 3.9 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 3.9 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 3.9. Versioned asynchronous releases, for example with the form OpenShift Container Platform 3.9.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 upgrading your cluster properly.

2.8.1. RHBA-2018:1566 - OpenShift Container Platform 3.9.27 Bug Fix and Enhancement Update

Issued: 2018-05-16

OpenShift Container Platform release 3.9.27 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:1566 advisory. The container images included in the update are provided by the RHBA-2018:1567 advisory.

Space precluded documenting all of the bug fixes and images for this release in the advisory. See the following sections for notes on upgrading and details on the bug fixes and images included in this release.

2.8.1.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.1.2. Bug Fixes

- Build pods use multiple containers. Binary builds need to specify which container to stream content into, and for custom builds the name of the container is different from non-custom builds. When streaming binary content into a custom build, the expected container, git-clone, does not exist and the build fails. The logic for streaming binary content into a custom build pod will be changed to reference the correct container name, custom-build. With this bug fix, binary content will successfully stream into the custom build container. (**BZ#1560659**)

- Resource constraints can lead to the readiness probe in the example Jenkins templates readiness probes citing failure prematurely. Jenkins deployments would fail unnecessarily. With this bug fix, the readiness probe was relaxed in the templates. As a result, there is a decrease in unnecessary Jenkins deployment failures due to the aggressive readiness probe. (**BZ#1559675**)

- The master ***admin.kubeconfig*** file was added to the **`oc command`** to allow the operation to have the proper authorization and access to the necessary resources. (**BZ#1561247**)

- The installer improperly tried to set the SELinux context on a path that may not exist. This task was meant to work around a problem in CRI-O that no longer exists and, as such, that task has been removed. (**BZ#1564949**)

- Service catalog pods had a high log verbosity set by default. Therefore, service catalog pods on the master node produced a large amount of log data. The default log verbosity is now reset to a lower level. (**BZ#1564179**)

- The Elasticsearch server TLS certificate does not have an external host name in the subject alt. name list. Clients accessing Elasticsearch externally cannot turn on the MITM server certificate validation. When configuring Elasticsearch to allow external access, add the external host name in the subject alt. name list. TLS clients can turn on server certificate validation. (**BZ#1554878**)

- The Fluentd plug-in logs the entire error response on failure, which fills up the on-disk logs. The entire response is now only logged when in debug mode and on-disk logs no logger consume the disk. (BZ#1554885)

- The default write operation for Fluentd to Elasticsearch is `index`. Writes can trigger unnecessary `delete` operations for Elasticsearch, causing extra load that affects performance. Use the `create` operation. Writes to elasticsearch will only create records or skip updates if the records are duplicates reducing the load on the server. (BZ#1565909)

- The curator pod was crash-looping because it was unable to find its entry point script due to a bad merge from origin into downstream dist-git. The pod was not functional and cycled crash-looping. With this bug fix, the code was synced with upstream. (BZ#1572419)

- The Fluentd secure-forward plug-in supports the host name placeholder `${hostname}` in the configuration file. Although the value is case-sensitive, the upper case string `${HOSTNAME}` was set and it failed to pick up the correct hostname of the Fluentd container. The bug is now fixed. (BZ#1553576)

- After manually typing a URL with non-existing image, page load messaging would remain on the page, signaling that the page load is ongoing, even though it is done and the **The image stream details could not be loaded** alert is shown. Set the `loaded` scope variable when the image is or is not loaded and use it in the view to hide the **loading** messaging. After the attempt to load the image data, the **loading** messaging is now hidden, even if the image cannot be loaded. (BZ#1550797)

- Previously, the web console would not let you add new keys when editing a ConfigMap that was empty. Clicking **Add Item** in the editor would have no effect. With this bug fix, you can now correctly add items when editing a ConfigMap that has none. (BZ#1558863)

- Restricting DaemonSet nodes with the project's default node selector resulted in the deletion and creation of DaemonSet pods in a loop on those nodes that were restricted by adding project default node selector. With this bug fix, the upstream DaemonSet logic is now updated to be aware of the project's default node selector. (BZ#1571093)

- The Hawkular Alerts components has been removed from Hawkular Metrics. This change has no functional impact on Hawkular Metrics. (BZ#1543647)

- Previously there was incorrect management of OVS flows. If two nodes rebooted and swapped IP addresses when they came back up, then other nodes might not be able to send traffic to pods on one or both of those nodes. The code that manages OVS flows is now more careful to make the correct changes in cases of node IP reassignment. Pod-to-pod traffic should continue to work correctly even after nodes swap IP addresses. (BZ#1570394)

- The update Egress policy needed blocking outgoing traffic, patching OVS flows, and then re-enabling traffic. However, the OVS flow generation for DNS names was slow. This resulted in a few seconds of Egress traffic downtime, which may not be acceptable. With this bug fix, update Egress policy handling is updated to pre-populate all new OVS flows before blocking the outgoing traffic. This reduces the downtime during Egress policy updates. (BZ#1571430)

- When using per-namespace static egress IPs, all external traffic is routed through the egress IP. *External* means all traffic, which is not directed to another pod, and so this includes traffic from the pod to the pod's node. When pods are told to use the node's IP address for DNS, and the pod is using a static egress IP, then DNS traffic will be routed to the egress node first, and then back to the original node, which might be configured to not accept DNS requests from other hosts, causing the pod to be unable to resolve DNS. Pod-to-node DNS requests now bypass the egress IP and go directly to the node and DNS works. (BZ#1570398)

- This bug fix addresses an issue on the node where setting disabling **`cpu-cfs-quota`** did not prevent CPU CFS limits from being set on pods when **`cgroups-per-qos`** was enabled. (**BZ#1558155**)

- This bug fix addresses an issue where clusters running with OpenStack cloud integration have nodes removed when the corresponding instance is stopped. Node resources whose instances are stopped are no longer removed from the cluster. (**BZ#1558422**)

- Nodes entered an impaired state when a volume is forcefully detached and not rebooted. Any new volume attached to the node is stuck in an attaching state. Any node that has a volume stuck in an attaching state for more than 21 minutes will be tainted and must be removed from cluster, then added back to remove the taint and fix the impaired state of the node. With this bug fix, impaired are removed from scheduling, giving the OpenShift Container Platform administrator the ability to fix the node and bring it back. (**BZ#1455680**)

- Previous releases of OpenShift Container Platform would improperly reconfigure **`docker`** to mark the internal registry as insecure when it should not have. This has been fixed in OpenShift Container Platform 3.9 and should no longer happen. (**BZ#1502028**)

2.8.1.3. Enhancements

- Use CRI-O as an RPM to use CRI-O as the container runtime. To install CRI-O as an RPM, set the following two options:

  ```
  openshift_use_crio=True
  openshift_crio_use_rpm=True
  ```

  (**BZ#1553186**)

- The yedit module now generates unique backup files. Previously, if changes were made to the same resource multiple times, only the latest diff would be saved. (**BZ#1555426**)

- Administrators can now see messages for which we are unable to determine the proper namespace to associate with them. Otherwise, messages appear to be missing and are not viewable for review. A Kibana Index pattern will be created for administrators if it does not exist. (**BZ#1519522**)

- In the absence of inventory values, reuse the values used for the current deployment to preserve tuned values. In the case of Elasticsearch, when a user had done tuning of the cluster but did not propagate those values into variables, upgrading logging would use role default values, which may put the cluster in a bad state and lead to loss of log data. Values are now honored in order for EFK: inventory → existing environment → role defaults. (**BZ#1561196**)

- The number of Kibana index-patterns for cluster administrators is now limited. Previously, the list was unmanageable and unneeded on large clusters with many namespaces. Cluster administrators now only see a limited subset of index-patterns. (**BZ#1563230**)

2.8.2. RHBA-2018:1796 - OpenShift Container Platform 3.9.30 Bug Fix and Enhancement Update

Issued: 2018-06-06

OpenShift Container Platform release 3.9.30 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:1796 advisory. The container images included in the update are provided by the RHBA-2018:1797 advisory.

Space precluded documenting all of the bug fixes and images for this release in the advisory. See the following sections for notes on upgrading and details on the bug fixes and images included in this release.

2.8.2.1. Bug Fixes

- Jenkins no_proxy processing could not handle suffixes like **`".svc"`**. As a result, communication between a Jenkins Kubernetes agent pod and the Jenkins master would attempt to go through a configured **`http_proxy`** and fail. With this bug fix, the OpenShift Container Platform jenkins agent images are updated to automatically include the jenkins master and jnlp hosts in the **`no_proxy`** list. The Jenkins limitation for **`no_proxy`** processing is now circumvented. (BZ#1578989)

- When creating the Elasticsearch server certificate, the external Elasticsearch host names were unconditionally added to the **`subjectAltName`**. Installation would fail because only host name components beginning with a letter are allowed in the **`subjectAltName`**, so host names like **`es.0xdeadbeef.com`** were disallowed and would cause an error. A warning is now issued if the Elasticsearch host name contains a component which does not begin with a letter, and it is not added to the **`subjectAltName`**. Logging installation now completes successfully. (BZ#1567767)

- The plug-in only caught the **`KubeException`**, but not more general exceptions. Therefore, consumers were stuck cycling until the API server could be contacted. Metadata fetch is now more relaxed and gracefully catches the exception, returning no metadata, and subsequently the record is orphaned. (BZ#1560170)

- **`logging-elasticsearch-ops`** was missing in the delete` configmaps` list in the openshift-ansible **`delete_logging`** role. The **`logging-elasticsearch-ops`** configmap still exists after running the uninstall Ansible playbook for logging. **`logging-elasticsearch-ops`** is added to the delete configmaps list. All of the logging configmaps including **`logging-elasticsearch-ops`** are now uninstalled by running the uninstall Ansible playbook for logging. (BZ#1549220)

- The **Create Project** button was incorrectly displayed to users when they had no projects and self-provisioning had been disabled on the projects list page of the web console. The action would always fail, so the button should have been hidden. The bug is now fixed, and **Create Project** is now correctly hidden in the console when self-provisioning is disabled. (BZ#1577359)

- This bug fix addresses an issue pulling images from a private docker hub registry. (BZ#1578088)

- This bug fix addresses where **`cfs_quota`** might still be set on a pod even when **`cpu-cfs-quota`** is set to **`false`** on the node. (BZ#1581860)

2.8.2.2. Enhancements

- Users are now allowed to disable JSON payload parsing. Parsing each log message into JSON and attaching it to the final payload is an expensive operation. Fluentd can now be configured to disable parsing of message payloads. This is the initial configuration change to deprecating the feature from the fluent-plugin-kubernetes_metadata_filter. (BZ#1569825)

2.8.2.3. Images

This release updates the Red Hat Container Registry (***registry.access.redhat.com***) with the following images:

```
openshift3/apb-base:v3.9.30-2
openshift3/container-engine:v3.9.30-2
openshift3/cri-o:v3.9.30-2
openshift3/image-inspector:v3.9.30-2
openshift3/jenkins-2-rhel7:v3.9.30-2
openshift3/jenkins-slave-base-rhel7:v3.9.30-2
openshift3/jenkins-slave-maven-rhel7:v3.9.30-2
openshift3/jenkins-slave-nodejs-rhel7:v3.9.30-2
openshift3/local-storage-provisioner:v3.9.30-2
openshift3/logging-auth-proxy:v3.9.30-2
openshift3/logging-curator:v3.9.30-2
openshift3/logging-elasticsearch:v3.9.30-2
openshift3/logging-eventrouter:v3.9.30-2
openshift3/logging-fluentd:v3.9.30-2
openshift3/logging-kibana:v3.9.30-3
openshift3/mariadb-apb:v3.9.30-2
openshift3/mediawiki-123:v3.9.30-2
openshift3/mediawiki-apb:v3.9.30-2
openshift3/metrics-cassandra:v3.9.30-2
openshift3/metrics-hawkular-metrics:v3.9.30-2
openshift3/metrics-hawkular-openshift-agent:v3.9.30-2
openshift3/metrics-heapster:v3.9.30-2
openshift3/mysql-apb:v3.9.30-2
openshift3/node:v3.9.30-2
openshift3/oauth-proxy:v3.9.30-2
openshift3/openvswitch:v3.9.30-2
openshift3/ose-ansible-service-broker:v3.9.30-2
openshift3/ose-ansible:v3.9.30-3
openshift3/ose-cluster-capacity:v3.9.30-2
openshift3/ose-deployer:v3.9.30-2
openshift3/ose-docker-builder:v3.9.30-2
openshift3/ose-docker-registry:v3.9.30-2
openshift3/ose-egress-http-proxy:v3.9.30-2
openshift3/ose-egress-router:v3.9.30-2
openshift3/ose-f5-router:v3.9.30-2
openshift3/ose-haproxy-router:v3.9.30-2
openshift3/ose-keepalived-ipfailover:v3.9.30-2
openshift3/ose-pod:v3.9.30-2
openshift3/ose-recycler:v3.9.30-2
openshift3/ose-service-catalog:v3.9.30-2
openshift3/ose-sti-builder:v3.9.30-2
openshift3/ose-template-service-broker:v3.9.30-2
openshift3/ose-web-console:v3.9.30-2
openshift3/ose:v3.9.30-2
openshift3/postgresql-apb:v3.9.30-2
openshift3/prometheus-alert-buffer:v3.9.30-2
openshift3/prometheus-alertmanager:v3.9.30-2
openshift3/prometheus-node-exporter:v3.9.30-2
openshift3/prometheus:v3.9.30-2
openshift3/registry-console:v3.9.30-2
openshift3/snapshot-controller:v3.9.30-2
openshift3/snapshot-provisioner:v3.9.30-2
```

2.8.2.4. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.3. RHSA-2018:2013 - OpenShift Container Platform 3.9.31 Security, Bug Fix, and Enhancement Update

Issued: 2018-06-27

OpenShift Container Platform release 3.9.31 is now available. The list of packages and security fixes included in the update are documented in the RHSA-2018:2013 advisory. The container images included in the update are provided by the RHBA-2018:2014 advisory.

Space precluded documenting all of the bug fixes and enhancements for this release in the advisory. See the following sections for notes on upgrading and details on the bug fixes and enhancements included in this release.

2.8.3.1. Bug Fixes

- The webhook payload can contain an empty commit array, which results in an array indexing error when processed by the API server. As a result, the API server crashes. Check for an empty array before attempting to index into it. With this bug fix, empty commit payloads are handled without crashing the API server. (**BZ#1586076**)

- A secret with a wrong password causes pull failures for all images. Any public image from the same registry pull will fail. This bug fix adds retry logic for the **401 error** when the password is wrong. Now, if the image is public, the image is pulled and the wrong secret is ignored. (**BZ#1506175**)

- The **`openshift-jenkins-sync`** plug-in assumed the Jenkins service and pipeline strategy build were in the same project when constructing the build URL for the OpenShift Container Platform web console. When Jenkins is in one project and the pipeline strategy build is in another project, the view log link in the OpenShift Container Platform web console points to the wrong URL because if cannot find the Jenkins service/route. The **`openshift-jenkins-sync`** plug-in now looks for the Jenkins service/route in the namespace it is running in. Also, if the user has explicitly configured the root URL in Jenkins, there is greater precedence. The URL for a given pipeline strategy build in the OpenShift Container Platform web console now renders correctly. (**BZ#1542460**)

- Image validation used to validate an old image object and the image signature import controller would generate such an image. As a result, invalid images were pushed to etcd. With this big fix, validation is changed to validate a new image object and logic to fix some invalid images is now introduced. The controller no longer generates invalid images and it is no longer possible to upload an invalid image object. (**BZ#1560311**)

- The transfer of plug-ins from the RPM installation location to the Jenkins home directory were not occurring properly with the OpenShift Container Platform v2 Jenkins RHEL image when Jenkins was previously deployed on an OpenShift Container Platform pod with a persistent volume. An upgrade of the OpenShift Container Platform v2 Jenkins RHEL image would not result in the deployment having the most recent plug-ins associated with the newer image. TheOpenShift Container Platform v2 Jenkins RHEL image **`run`** script is now updated to properly transfer the plug-ins. An upgrade of the OpenShift Container Platform v2 Jenkins RHEL image now results in the deployment having the most recent plug-ins associated with the newer image. (**BZ#1550193**)

- If the Jenkins root URL could not be retrieved from the route from the Jenkins template, then the unusable URL could be used in constructing the various annotations for pipeline builds. The

associated annotation links would not render when referenced from the OpenShift Container Platform web console. To help account for those edge cases, the sync plug-in now looks for explicitly configured root URLs in Jenkins. The links associated with the pipeline build annotations now render if the root URL is properly configured. (**BZ#1558997**)

- Allowed registries for import configuration settings were considered only for image imports. You could easily get around the image import validation by editing image streams manually and use any desired image. With this bug fix, image streams are now also validated. You cannot use an external image that does not match an entry in whitelisted registry entries. (**BZ#1505315**)

- In certain cases, an existing etcd installation might not have updated configuration variables, causing services to fail. This bug fix ensures the ***etcd.conf*** file is verified during upgrades and that all variables are set as expected. (**BZ#1529575**)

- To enable support for storage devices on Microsoft Azure, the seboolean **`virt_use_samba`** is required. (**BZ#1537872**)

- The node configuration file had hardcoded labels in the CRI-O section. Therefore, double labels could occur if labels were set elsewhere in the installer. Remove the unecessary hardcoded labels, eliminatingthe possibility of double labels. (**BZ#1553012**)

- The **secure-forward** template generated in the configMap does not include the **`<store>`** tag, as mentioned in the documentation. The configuration fails when more stores are defined. Add the enclosing **`<store>`** tag for the template. Removing the comments provides a syntactically valid configuration. (**BZ#1498398**)

- To label nodes for Fluentd, a scrip was run out of ***/tmp***. When the **`noexec`** option was set for ***/tmp***, the playbook failed. Instead of running a script where paused, label with a pause using the **`shell`** Ansible task. With this bug fix, you are able to pause and run to completion. (**BZ#1588009**)

- There were changes to the kube-proxy iptables rules in upstream Kubernetes. Network performance and overall system performance was severely impacted on extremely large clusters like OpenShift Online. With this bug fix, there are multiple optimizations of the kube-proxy iptables rule and performance problems are resolved. (**BZ#1514174**)

- A version of the OVS RPM was used that did not have the right SELinux policy. Therefore, OVS failed due to SELinux. Get the correct version of the OVS RPM with the correct rules. With this bug fix, OVS now works. (**BZ#1548677**)

- When using the static per-project egress IPs feature, egress IPs may stop working in some circumstances if an egress IP is moved from one project to another, or from one node to another. Additionally, if the same egress IP is assigned to two different projects, or two different nodes, then it might not work correctly, even after the duplicate assignment is removed. This bug fix resolves the issue and static per-project egress IPs should work more reliably. (**BZ#1553294**)

- OpenShift Container Platform's default network plug-in has not been updated to implement the new NetworkPolicy features introduced upstream in Kubernetes (policies for controlling egress, and policies based on IP addresses rather than pods or namespaces). Therefore, in OpenShift Container Platform 3.9, creating a NetworkPolicy with an **`ipBlock`** section would cause nodes to crash, and creating a NetworkPolicy that contained only egress rules would erroneously cause ingress traffic to be blocked. The code is now aware of the unsupported NetworkPolicy features, though it does not yet implement them. If a NetworkPolicy contains **`ipBlock`** rules, those rules are ignored. This may cause the policy to be treated as **`deny all`** if the **`ipBlock`** rule was the only rule in the policy. If a NetworkPolicy contains only egress rules, it is ignored completely and does not affect ingress. (**BZ#1585243**)

- There was an regression issue in which the docker client in use by the kubelet qualifies image paths without a domain with docker.io client-side, resulting in all unqualified image paths attempting the pull from docker.io and ignoring the domain search list in the docker daemon. With this bug fix, the regression issue is resolved. (**BZ#1588768**)

- Unbinding a template service instance throws an error if the template service instance was deleted. It becomes impossible to unbind a service instance if the template service instance was manually deleted, including if the project containing the TSI was deleted. The template service broker will return **success**/**gone** in cases where the unbind refers to a non-existent template service instance. The unbind can now proceed even if the TSI no longer exists. (**BZ#1540819**)

- When deleting a namespace, the objects within the namespace are deleted by the namespace controller, not the user. Service bindings, when deleted, get unbound via an unbind request associated with the user doing the deletion. This leads to an unbind request coming from the namespace controller, which did not have all permissions required to perform an unbind. Change what permissions are required for unbind to align them with the permissions the namespace controller has. The unbind triggered by the namespace controller deleting the binding will succeed. (**BZ#1554141**)

- This bug fix adds a small compatibility check to eliminate a pain point with API endpoints changing from 3.7 to 3.9. (**BZ#1554145**)

- You may now define a set of hooks to run arbitrary tasks during the node upgrade process. To implement these hooks, set **`openshift_node_upgrade_pre_hook`**, **`openshift_node_upgrade_hook`**, or **`openshift_node_upgrade_post_hook`** to the path of the task file you want to execute. **`The openshift_node_upgrade_pre_hook`** hook is executed after draining the node and before it is upgraded. The **`openshift_node_upgrade_hook`** is executed after the node has been drained and packages updated but before it is marked schedulable again. The **`openshift_node_upgrade_post_hook`** hook is executed after the node is marked schedulable immediately before moving on to other nodes. (**BZ#1572786**)

- Improper input validation of the OpenShift Container Platform routing configuration can cause an entire shard to be brought down. A malicious user can use this vulnerability to cause a Denial of Service attack for other users of the router shard. (**BZ#1553035**)

- OpenShift and Atomic Enterprise Ansible deploys a misconfigured etcd file that causes the SSL client certificate authentication to be disabled. Quotations around the values of **`ETCD_CLIENT_CERT_AUTH`** and **`ETCD_PEER_CLIENT_CERT_AUTH`** in ***etcd.conf*** result in etcd being configured to allow remote users to connect without any authentication if they can access the etcd server bound to the network on the master nodes. An attacker could use this flaw to read and modify all the data about the OpenShift Container Platform cluster in the etcd datastore, potentially adding another compute node, or bringing down the entire cluster. (**BZ#1557822**)

- A privilege escalation flaw was found in the source-to-image component of OpenShift Container Platform, which allows the assemble script to run as the root user in a non-privileged container. An attacker can use this flaw to open network connections, and possibly other actions, on the host which are normally only available to a root user. (BZ#1579096) (**BZ#1579096**)

2.8.3.2. Enhancements

- A new flag is now added to the **`oc adm drain`** command to allow you to select nodes by label. There was a need to be able to drain multiple nodes, without having to perform the **`drain`** operation on each individual node. The **`oc adm drain`** command now supports a **`--selector`** flag, which results in all nodes matching a given label being drained. (**BZ#1466390**)

2.8.3.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.4. RHBA-2018:2213 - OpenShift Container Platform 3.9.33 Bug Fix Update

Issued: 2018-07-18

OpenShift Container Platform release 3.9.33 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2213 advisory. The list of container images included in the update are documented in the RHBA-2018:2212 advisory.

2.8.4.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.5. RHBA-2018:2335 - OpenShift Container Platform 3.9.40 Bug Fix and Enhancement Update

Issued: 2018-08-09

OpenShift Container Platform release 3.9.40 is now available. The list of packages and bug fixes included in the update are documented in the RHBA-2018:2335 advisory. The container images included in the update are provided by the RHBA-2018:2336 advisory.

Space precluded documenting all of the bug fixes and enhancements for this release in the advisory. See the following sections for notes on upgrading and details on the bug fixes and enhancements included in this release.

2.8.5.1. Bug Fixes

- The link generation code assumed all project logs are written to indices that have a common naming pattern. Therefore, users were linked to non-existent indices. With this bug fix, project logs that will be archived to different indices are annotated with the required information to properly build the link. Users are now routed using a link that will query the data store correctly and return data. (**BZ#1523047**)

- The service catalog pods were previously labeled with node selectors that were limited to the first master. Now, the pods are properly labeled with a label that applies to all masters, which enables service catalog pods to deploy to all masters properly. (**BZ#1554623**)

- When adding IP route and host name packages to the **`openshift-ansible`** image, Ansible uses common user-space utilities for determining default facts. The **`ansible_default_ipv4.address`** fact is populated using utilities from the **iproute** package, and this fact is used for populating the OpenShift Container Platform IP in ***roles/openshift_facts/library/openshift_facts.py***. (**BZ#1558689**)

- Due to corrupt file chunk buffers, Fluentd was blocked processing messages until the buffer was removed. This bug fix introduces a handler to remove corrupt messages from the processing work flow. Corrupt messages are now sent to a dead letter queue while continuing to process other messages and the pipleline is no longer blocked. (**BZ#1562004**)

- Curator checked the readiness of Elasticsearch at startup. If Elasticsearch was not ready after

one minute, Curator gave up. This was repeated 5 times with exponential backoff via pod restart policy with default **`backofflimit=5`**. Curator could not be deployed without Elasticsearch. Now, Curator checks for Elasticsearch readiness indefinitely before each run. Curator and Elasticsearch can be deployed independently. (**BZ#1564350**)

- Headless service had **`service.Spec.ClusterIP=None`** set and this was not ignored as part of unidling. This generates an incorrect endpoint ID in the HAProxy configuration and the configuration will fail to load. This leads to the router not servicing any routes. Headless services are now ignored during unidle handling and, therefore, there is now no problem with HAProxy configuration loading. The router will service routes as expected. (**BZ#1571479**)

- The incoming data had a field **`record["event"]`** that is a String value and not the hash value expected by the **`transform_eventrouter code`**. This causes the code to throw an error and fluentd to emit an error like:

```
error_class=NoMethodError error="undefined method `key?' for
\"request\":String"
```

  The **`transform_eventrouter`** code is now changed to only process the **`record["event"]`** field if it is a hash. Records can flow through to Elasticsearch again. (**BZ#1588828**)

- Recently, **`cloudResourceSyncManager`** was implemented, which continuously fetched node addresses from cloud providers. The kubelet then received node addresses from the **`cloudResourceSyncManager`**. At the time of node registration or kubelet start, the kubelet fetches node addresses in a blocking loop from **`cloudResourceSyncManager`**. The issue was that **`cloudResourceSyncManager`** was not started before the kubelet had started fetching node addresses from it for the first time. Due to this, the kubelet got stuck in the blocking loop and never returned. It caused node failures at the network level, and no node could be registered. Also, as the kubelet was blocked early, the **`cloudResourceSyncManager`** never got a chance to start. The **`cloudResourceSyncManager`** is now started early in the kubelet startup process so that the kubelet does not get blocked on it and **`cloudResourceSyncManager`** is always started. (**BZ#1601813**)

- Ansible 2.6.0 will not evaluate undefined variables with **`|bool`** as **`false`**. You must define a **`| default(false)`** for **`logging_elasticsearch_rollout_override`**. Specify a default on the **`when`** condition. The playbook executes successfully. (**BZ#1602015**)

2.8.5.2. Enhancements

- This feature allows the number of indices and replicas to be configured using environment variables. The logs collected for infra services consumes a large portion of the available disk space. Spreading the data across available nodes by modifying the replica and shard settings allow Elasticsearch to better support these large amounts of data. This feature results in improved performance in Elasticsearch when there are large amounts of data from infra services. (**BZ#1553257**)

2.8.5.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.6. RHBA-2018:2549 - OpenShift Container Platform 3.9.41 Bug Fix Update

Issued: 2018-08-28

OpenShift Container Platform release 3.9.41 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2549 advisory. The list of container images included in the update are documented in the RHBA-2018:2548 advisory.

2.8.6.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.7. RHBA-2018:2658 - OpenShift Container Platform 3.9.43 Bug Fix Update

Issued: 2018-09-24

OpenShift Container Platform release 3.9.43 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2658 advisory. The list of container images included in the update are documented in the RHBA-2018:2659 advisory.

2.8.7.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.8. RHSA-2018:2908- OpenShift Container Platform 3.9.51 Security and Bug Fix Update

Issued: 2018-11-19

OpenShift Container Platform release 3.9.51 is now available. The list of packages and security fixes included in the update are documented in the RHSA-2018:2908 advisory. The container images included in the update are provided by the RHBA-2018:2907 advisory.

2.8.8.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.9. RHBA-2018:3748 - OpenShift Container Platform 3.9.57 Bug Fix and Enhancement Update

Issued: 2018-12-13

OpenShift Container Platform release 3.9.57 is now available. The list of packages and bug fixes included in the update are documented in the RHBA-2018:3748 advisory. The container images included in the update are provided by the RHBA-2018:3747 advisory.

Space precluded documenting all of the bug fixes and enhancements for this release in the advisory. See the following sections for notes on upgrading and details on the bug fixes and enhancements included in this release.

2.8.9.1. Bug Fixes

- There were several problems related to updates. Spec changes for instances were blocked, even if there was not an ongoing operation; deleting a service instance that was updated to an invalid service plan would cause a crash; instances were not updated properly if a previous

update had failed. (BZ#1507595)

- The wizard was setting the **valid** state prior to retrieving data. The primary button was set to **Bind** due to prematurely setting the wizard state to **valid**. Do not set the wizard state to valid until the data is retrieved and you determined that more steps are necessary. At start up, the wizard is now not in the **valid** state. (BZ#1520828)

- The playbook for the **openshift-metrics** component was not verifying the return code of some commands. Some incorrect parameters could be used to set up metrics. The playbook succeeded, but metrics were incorrectly set up. With this bug fix, the playbook verifies exit code for **oc apply** commands. Installation fails when incorrect commands are used. (BZ#1534538)

- During upgrade, all masters were restarted simultaneously. In a multi-master environment, leader election could have failed A short pause is now added between master restarts and leader election no longer fails. (BZ#1540054)

- Unicode characters in pull specs for images were treated incorrectly. Installation stopped with a cryptic message. Unicode handling in image paths is now fixed and a correct error message is displayed. (BZ#1544648)

- The default node selector for Elasticsearch was not set. Installation with default settings and enabled logging failed. Correct handling of a default node selector is now implemented. The logging installation completes without a node selector set. (BZ#1547210)

- There were bugs in the per-project static IP code. If you removed the static IP from a project and then re-added it, it would not always work correctly. With this bug fix, removing and re-adding static egress IPs now works. (BZ#1547899)

- Patch files for logging ConfigMaps were incorrectly generated. There were failures when applying the file because reference lines were not there. This bug fix changes how white-listed lines were handled. They are prevented from ending up in patch files that were generated, but the patch is still allowed to be applied after. Logging ConfigMaps are now correctly patched based on changes from current deployments. (BZ#1552744)

- When fluentd is configured as the combination of collectors and MUX, event logs from the event were supposed to be processed by MUX, not by the collector for the both **MUX_CLIENT_MODE** maximal and minimal. This occurred because if an event log is formatted in the collector (the event record is put under the Kubernetes key), the log is forwarded to MUX and passed to the k8s-meta plug-in there and the existing Kubernetes record is overwritten. It removed the event information from the log. + **Fix 1**: To avoid the replacement, if the log is from event router, the tag is rewritten to **${tag}.raw** in ***input-post-forward-mux.conf***, which treats the log in the **MUX_CLIENT_MODE=minimal** way. + **Fix 2**: There was another bug in Ansible in that the environment variable **TRANSFORM_EVENTS** was not set in MUX even if **openshift_logging_install_eventrouter** was set to **true**. + With these two fixes, the event logs are correctly logged when MUX is configured with **MUX_CLIENT_MODE=maximal** as well as **minimal**. (BZ#1554293)

- The ability to skip disabling swap by use of **openshift_disable_swap=False** is removed from OpenShift Container Platform 3.9. This feature was undocumented and should not be used. (BZ#1557200)

- The kube-proxy and kubelet parts of the OpenShift Container Platform node process were being given different default values for the config options describing how to interact with iptables. OpenShift Container Platform would periodically add a bogus iptables rule that would cause some per-project static egress IPs to not be used for some length of time, until the bogus rule was removed again. While the bogus rule was present, traffic from those projects would use the

node IP address of the node hosting the egress IP, rather than the egress IP itself. The inconsistent configuration was resolved, causing the bogus iptables rule to no longer be added. Projects now consistently use their static egress IPs. (BZ#1560584)

- A node selector for TSB was not documented. Therefore, users were not aware that **`template_service_broker_selector`** and **`openshift_hosted_infra_selector`** could be changed. With this bug fix, **`hosts.example`** now lists **`openshift_hosted_infra_selector`** and **`template_service_broker_selector`**. TSB and the infra node selector is now more visible. (BZ#1569165)

- Kibana index pattern seeding was modified to reduce the number of entries in the drop-list that were visible to cluster admins. Cluster admins navigating to Kibana for logging from the web console were no longer sent to the correct URL. Take into consideration a user's **`cluster-admin`** role when creating the web console link to pod logs in Kibana. With this bug fix, the link is now created correctly. (BZ#1598305)

- The unstaller attempted to deploy **`node_exporter`** during a Prometheus uninstall. Uninstall failed due to **`node_exporter`** not being able to deploy to the Prometheus namespace. This bug fix adds a check to deploy **`node_exporter`** if Prometheus is installed and present. Now, Prometheus uninstall completes successfully without attempting to install **`node_exporter`** into the non-existent namespace. (BZ#1598642)

- In certain environments, the image stream content changes rapidly enough to prevent replacement with new content due to the object being modified. The upgrade would fail. The image stream update now retries three times. With this bug fix, the upgrade should be successful. (BZ#1623642)

- When wildcard routes are enabled and namespace ownership checks are disabled, non-wildcard routes get removed and immediately re-added on the resync interval boundaries and this causes a brief route outage and results in intermittent errors on a route. Do not remove and re-add the routes on resync interval in the specific case when wildcard routes enabled and namespace ownership checks are disabled. With this bug fix, non-wildcard routes continue to serve without any intermittent errors. (BZ#1624078)

- Javascript runtime errors are caused by accessing an undefined key on the secrets object. Use the **`lodash _.get()`** method, which will not error out if accessing an undefined key on the secrets object. The secrets drop-down is now a visible event if there is a secret without any data. (BZ#1633285)

- **`openshift_facts`** uses default images to determine the version in a containerized installation. When custom images and the internal registry is used, the default images cannot be pulled. Therefore, **`openshift.common.version`** is left empty and the upgrade fails when OpenShift Container Platform is not installed. If there is a **`custom_image`** name, pass it to **`openshift_facts`** to use. With this bug fix, **`openshift.common.version`** is set in containerized installations with the registry mirror and installation succeeds. (BZ#1634004)

- Systemd has a default service timeout of 90 seconds and, in some cases, **`ovsdb-server`** does not start up in that time. Therefore, openvswitch, which has a dependency on **`ovsdb-server`**, cannot start and installation fails. Increase systemd timeout for **`ovsdb-server`** to five minutes. As a result, **`ovsdb-server`** does not timeout and openvswitch starts successfully. (BZ#1636234)

- The OSB Client Library used by the Service Catalog controller pod was not closing and freeing TCP connections used to communicate with brokers. Over a period of time, many TCP connections would remain open and, eventually, the communication between the Service

Catalog controller and brokers would fail. Additionally, the pod would become unresponsive. Use a new version of the OSB Client Library, which contains a fix to close connections when finishing a HTTP request and free idle connections. (BZ#1638726)

- When using docker with log-driver journald, the setting in */etc/sysconfig/docker* has changed to use **`--log-driver journald`** instead of **`--log-driver=journald`**. Fluentd cannot detect that journald is being used, so assumes **`json-file`**, and cannot read any kubernetes metadata because it does notlook for the journald **`CONTAINER_NAME`** field. This results in lots of Fluentd errors. Change the way Fluentd detects the docker log driver so that it looks for **`--log-driver journald`** in addition to **`--log-driver=journald`**. Fluentd can now detect the docker log driver, and can correctly process Kubernetes container logs. (BZ#1638900)

- The master API sent an error without details, but the registry was not ready for this and the registry panicked. Check if the **`details`** field is available. The registry sends a proper error to the client and logs the full error from the master API. (BZ#1639839)

- Running **`oc logs $fluentd_pod`** suggests that you run **`oc exec <pod_name> /opt/app-root/src/utils/logs`**, which includes the non-existing utility logs path. Therefore, **`oc exec <pod_name> /opt/app-root/src/utils/logs`** fails with **no such file or directory**. This bug fix updates the suggested command line to **`oc exec <pod_name> -- logs`** since the utility **`logs`** is now in the ***PATH*** and there is no need to specify the full path. As a result, **`oc logs $fluentd_pod`** suggests the correct command line to show the fluentd logs. (BZ#1643340)

- There was inconsistent user usage when creating and removing temporary directories. Removing the directories can fail if the user does not have proper access. With this bug fix update, tasks use same **`become:false settings`** and the removal does not fail when using different user. (BZ#1643732)

- Unnecessarily short timeout resulted in a failure to reuse artifacts from a previous build when incremental builds were selected with s2i. This could occur when the size of the artifacts being reused was particularly large or the host system was running particularly slowly. Invalid artifacts could be used in a subsequent build, or artifacts would be recreated instead of reused resulting in performance degradation. Timeout has been increased to a sufficiently large value as to avoid this problem. Artifact reuse should no longer timeout. (BZ#1644343)

- Prometheus was automatically discovering two separate scrape endpoints for the Kubernetes router. One of the endpoints was not able to be scraped due to a missing auth config. This appeared as a failing scrape in the Prometheus status. This bug fix removed the duplicate prometheus scrape endpoint for the Kubernetes router. Now, Prometheus only scrapes a single, working endpoint for the Kubernetes router. (BZ#1644544)

- Egress IP-related iptables rules were not recreated if they were deleted. If a user restarted firewalld or **`iptables.service`** on a node that hosted egress IPs, then those egress IPs would stop working. Traffic that should have used the egress IP would use the node's normal IP instead. Egress IP iptables rules are now recreated if they are removed and egress IPs work reliably. (BZ#1653382)

- Previously, the metrics playbooks attempted to verify the availability of the **`python-passlib`** library in a manner that was unpredictable. This library is also a dependency of the **`openshift-ansible`** packaging, so there is no need for this additional check and it is now removed. (BZ#1654887)

2.8.9.2. Enhancements

- Master nodes are now schedulable. Web console pods are now restricted to only run on masters. Therefore, master nodes are no longer marked as non-schedulable. (**BZ#1535673**)
- The APB tool now works with Minishift. (**BZ#1536687**)

2.8.9.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.10. RHBA-2019:0028 - OpenShift Container Platform 3.9.60 Bug Fix Update

Issued: 2019-01-10

OpenShift Container Platform release 3.9.60 is now available. The packages and bug fixes included in the update are documented in the RHBA-2019:0028 advisory. The list of container images included in the update are documented in the RHBA-2019:0027 advisory.

2.8.10.1. Known Issues

- There is a regression error when running on OpenStack. The function **`normalize_openstack_facts`** pulled in a commit that broke how the inventory file is parsed when hosted on OpenStack. When using OpenStack without a floating IP for your cluster, you will see an error. To address this issue, you must manually patch the ***/usr/share/ansible/openshift-ansible/playbooks/init/cluster_facts.yml*** file with:

```
--- /usr/share/ansible/openshift-
ansible/roles/openshift_facts/library/openshift_facts.py       2019-
01-14 20:09:27.723830675 +0000
+++ /usr/share/ansible/openshift-
ansible/roles/openshift_facts/library/openshift_facts.py.fixed 2019-
01-14 17:08:33.433853271 +0000
@@ -337,12 +337,12 @@
     for f_var, h_var, ip_var in [('hostname', 'hostname', 'local-
ipv4'),
                                  ('public_hostname', 'public-
hostname', 'public-ipv4')]:
         try:
-            if socket.gethostbyname(metadata['ec2_compat'][h_var])
== metadata['ec2_compat'][ip_var].split(',')[0]:
+            if socket.gethostbyname(metadata['ec2_compat'][h_var])
== metadata['ec2_compat'][ip_var]:
                 facts['network'][f_var] = metadata['ec2_compat']
[h_var]
             else:
-                facts['network'][f_var] = metadata['ec2_compat']
[ip_var].split(',')[0]
+                facts['network'][f_var] = metadata['ec2_compat']
[ip_var]
         except socket.gaierror:
-            facts['network'][f_var] = metadata['ec2_compat']
[ip_var].split(',')[0]
+            facts['network'][f_var] = metadata['ec2_compat']
```

```
[ip_var]

     return facts
```

(**openshift/openshift-ansible#10956**)

2.8.10.2. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.11. RHBA-2019:0098 - OpenShift Container Platform 3.9.65 Bug Fix Update

Issued: 2019-01-31

OpenShift Container Platform release 3.9.65 is now available. The packages and bug fixes included in the update are documented in the RHBA-2019:0098 advisory. The list of container images included in the update are documented in the RHBA-2019:0100 advisory.

2.8.11.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.12. RHBA-2019:0331 - OpenShift Container Platform 3.9.68 Bug Fix Update

Issued: 2019-02-20

OpenShift Container Platform release 3.9.68 is now available. The packages and bug fixes included in the update are documented in the RHBA-2019:0331 advisory. The list of container images included in the update are documented in the RHBA-2019:0330 advisory.

2.8.12.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.7 or 3.9 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

CHAPTER 3. XPAAS RELEASE NOTES

The release notes for xPaaS docs have migrated to their own book on the Red Hat customer portal.

CHAPTER 4. COMPARING WITH OPENSHIFT ENTERPRISE 2

4.1. OVERVIEW

OpenShift Container Platform 3 is based on the OpenShift version 3 (v3) architecture, which is very different product than OpenShift version 2 (v2). Many of the same terms from OpenShift v2 are used in v3, and the same functions are performed, but the terminology can be different, and behind the scenes things may be happening very differently. Still, OpenShift remains an application platform.

This topic discusses these differences in detail, in an effort to help OpenShift users in the transition from OpenShift v2 to OpenShift v3.

4.2. ARCHITECTURE CHANGES

Gears Versus Containers

Gears were a core component of OpenShift v2. Technologies such as kernel namespaces, cGroups, and SELinux helped deliver a highly-scalable, secure, containerized application platform to OpenShift users. Gears themselves were a form of container technology.

OpenShift v3 takes the gears idea to the next level. It uses Docker as the next evolution of the v2 container technology. This container architecture is at the core of OpenShift v3.

Kubernetes

As applications in OpenShift v2 typically used multiple gears, applications on OpenShift v3 will expectedly use multiple containers. In OpenShift v2, gear orchestration, scheduling, and placement was handled by the OpenShift broker host. OpenShift v3 integrates Kubernetes into the master host to drive container orchestration.

4.3. APPLICATIONS

Applications are still the focal point of OpenShift. In OpenShift v2, an application was a single unit, consisting of one web framework of no more than one cartridge type. For example, an application could have one PHP and one MySQL, but it could not have one Ruby, one PHP, and two MySQLs. It also could not be a database cartridge, such as MySQL, by itself.

This limited scoping for applications meant that OpenShift performed seamless linking for all components within an application using environment variables. For example, every web framework knew how to connect to MySQL using the **`OPENSHIFT_MYSQL_DB_HOST`** and **`OPENSHIFT_MYSQL_DB_PORT`** variables. However, this linking was limited to within an application, and only worked within cartridges designed to work together. There was nothing to help link across application components, such as sharing a MySQL instance across two applications.

While most other PaaSes limit themselves to web frameworks and rely on external services for other types of components, OpenShift v3 makes even more application topologies possible and manageable.

OpenShift v3 uses the term "application" as a concept that links services together. You can have as many components as you desire, contained and flexibly linked within a project, and, optionally, labeled to provide grouping or structure. This updated model allows for a standalone MySQL instance, or one shared between JBoss components.

Flexible linking means you can link any two arbitrary components together. As long as one component can export environment variables and the second component can consume values from those

environment variables, and with potential variable name transformation, you can link together any two components without having to change the images they are based on. So, the best containerized implementation of your desired database and web framework can be consumed directly rather than you having to fork them both and rework them to be compatible.

This means you can build anything on OpenShift. And that is OpenShift's primary aim: to be a container-based platform that lets you build entire applications in a repeatable lifecycle.

4.4. CARTRIDGES VERSUS IMAGES

In OpenShift v3, an image has replaced OpenShift v2's concept of a cartridge.

Cartridges in OpenShift v2 were the focal point for building applications. Each cartridge provided the required libraries, source code, build mechanisms, connection logic, and routing logic along with a preconfigured environment to run the components of your applications.

However, cartridges came with disadvantages. With cartridges, there was no clear distinction between the developer content and the cartridge content, and you did not have ownership of the home directory on each gear of your application. Also, cartridges were not the best distribution mechanism for large binaries. While you could use external dependencies from within cartridges, doing so would lose the benefits of encapsulation.

From a packaging perspective, an image performs more tasks than a cartridge, and provides better encapsulation and flexibility. However, cartridges also included logic for building, deploying, and routing, which do not exist in images. In OpenShift v3, these additional needs are met by Source-to-Image (S2I) and configuring the template.

Dependencies

In OpenShift v2, cartridge dependencies were defined with **`Configure-Order`** or **`Requires`** in a cartridge manifest. OpenShift v3 uses a declarative model where pods bring themselves in line with a predefined state. Explicit dependencies that are applied are done at runtime rather than just install time ordering.

For example, you might require another service to be available before you start. Such a dependency check is always applicable and not just when you create the two components. Thus, pushing dependency checks into runtime enables the system to stay healthy over time.

Collection

Whereas cartridges in OpenShift v2 were colocated within gears, images in OpenShift v3 are mapped 1:1 with containers, which use pods as their colocation mechanism.

Source Code

In OpenShift v2, applications were required to have at least one web framework with one Git repository. In OpenShift v3, you can choose which images are built from source and that source can be located outside of OpenShift itself. Because the source is disconnected from the images, the choice of image and source are distinct operations with source being optional.

Build

In OpenShift v2, builds occurred in application gears. This meant downtime for non-scaled applications due to resource constraints. In v3, builds happen in separate containers. Also, OpenShift v2 build results used rsync to synchronize gears. In v3, build results are first committed as an immutable image and

published to an internal registry. That image is then available to launch on any of the nodes in the cluster, or available to rollback to at a future date.

Routing

In OpenShift v2, you had to choose up front as to whether your application was scalable, and whether the routing layer for your application was enabled for high availability (HA). In OpenShift v3, routes are first-class objects that are HA-capable simply by scaling up your application component to two or more replicas. There is never a need to recreate your application or change its DNS entry.

The routes themselves are disconnected from images. Previously, cartridges defined a default set of routes and you could add additional aliases to your applications. With OpenShift v3, you can use templates to set up any number of routes for an image. These routes let you modify the scheme, host, and paths exposed as desired, with no distinction between system routes and user aliases.

4.5. BROKER VERSUS MASTER

A master in OpenShift v3 is similar to a broker host in OpenShift v2. However, the MongoDB and ActiveMQ layers used by the broker in OpenShift v2 are no longer necessary, because **etcd** is typically installed with each master host.

4.6. DOMAIN VERSUS PROJECT

A project is essentially a v2 domain.