



# **OpenShift Container Platform 3.4 Release Notes**

---

Red Hat OpenShift Documentation  
Team





## Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

---

## Table of Contents

<b>CHAPTER 1. OVERVIEW .....</b>	<b>3</b>
1.1. VERSIONING POLICY .....	3
<b>CHAPTER 2. OPENSIFT CONTAINER PLATFORM 3.4 RELEASE NOTES .....</b>	<b>4</b>
2.1. OVERVIEW .....	4
2.2. ABOUT THIS RELEASE .....	4
2.3. NEW FEATURES AND ENHANCEMENTS .....	4
2.4. NOTABLE TECHNICAL CHANGES .....	15
2.5. BUG FIXES .....	16
2.6. TECHNOLOGY PREVIEW FEATURES .....	25
2.7. KNOWN ISSUES .....	26
2.8. ASYNCHRONOUS ERRATA UPDATES .....	26
<b>CHAPTER 3. XPAAS RELEASE NOTES .....</b>	<b>31</b>
<b>CHAPTER 4. COMPARING WITH OPENSIFT ENTERPRISE 2 .....</b>	<b>32</b>
4.1. OVERVIEW .....	32
4.2. ARCHITECTURE CHANGES .....	32
4.3. APPLICATIONS .....	32
4.4. CARTRIDGES VS IMAGES .....	33
4.5. BROKER VS MASTER .....	34
4.6. DOMAIN VS PROJECT .....	34
<b>CHAPTER 5. REVISION HISTORY: RELEASE NOTES .....</b>	<b>35</b>
5.1. THU FEB 09 2017 .....	35
5.2. TUE JAN 31 2017 .....	35
5.3. TUE JAN 24 2017 .....	35
5.4. WED JAN 18 2017 .....	35



# CHAPTER 1. OVERVIEW

The following release notes for OpenShift Container Platform 3.4 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). OpenShift Container Platform will continue to support older clients against newer servers. For example, a 3.1 client will work against 3.0, 3.1, and 3.2 servers.

Changes of APIs for non-security related reasons will involve, at minimum, two minor releases (3.1 to 3.2 to 3.3, for example) to allow older clients to update. Using new capabilities may require newer clients. A 3.2 server may have additional capabilities that a 3.1 client cannot use and a 3.2 client may have additional capabilities that are not supported by a 3.1 server.

**Table 1.1. Compatibility Matrix**

	X.Y (Client)	X.Y+N <sup>[a]</sup> (Client)
X.Y (Server)	1	3
X.Y+N <sup>[a]</sup> (Server)	2	1
[a] Where N is a number greater than 1.		

Fully compatible.

Client may not be able to access server features.

Client may provide options and features that may not be compatible with the accessed server.

## CHAPTER 2. OPENSIFT CONTAINER PLATFORM 3.4 RELEASE NOTES

### 2.1. OVERVIEW

Red Hat OpenShift Container Platform is a Platform as a Service (PaaS) that 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 Google 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 brings the OpenShift PaaS platform to customer data centers, enabling organizations to implement a private PaaS that meets security, privacy, compliance, and governance requirements.

### 2.2. ABOUT THIS RELEASE

Red Hat OpenShift Container Platform version 3.4 ([RHBA-2017:0066](#)) is now available. This release is based on [OpenShift Origin 1.4](#). New features, changes, bug fixes, and known issues that pertain to OpenShift Container Platform 3.4 are included in this topic.

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 a Cluster](#) topics in the [Installation and Configuration](#) documentation.

### 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. Kubernetes Deployments Support (Technology Preview)

OpenShift Container Platform 3.4 now supports both *Kubernetes deployments* objects (currently in [Technology Preview](#)) and the existing deployment configurations objects.

Like deployment configurations, Kubernetes deployments describe the desired state of a particular component of an application as a pod template. Kubernetes deployments create replica sets (an iteration of replication controllers), which orchestrate pod lifecycles.

See [Kubernetes Deployments Support](#) for more details on usage and support.

##### 2.3.1.2. Pod Disruption Budgets (Technology Preview)

*Pod disruption budgets* (currently in [Technology Preview](#)) allow the specification of safety constraints on pods during operations. Users with **cluster-admin** privileges can use them to limit the number of pods that are down simultaneously for a given project.



**PodDisruptionBudget** is an API object that specifies the minimum number or percentage of replicas that must be up at a time. Setting these in projects can be helpful during node maintenance (such as scaling a cluster down or a cluster upgrade) and is only honored on voluntary evictions (not on node failures).

See [Managing Pods](#) for more details.

## 2.3.2. Cluster Infrastructure

### 2.3.2.1. Quota on Persistent Volume Claim Storage Requests

With dynamic storage provisioning enabled, cluster administrators needed to be able to set quota on the amount of storage a project can request. Cluster administrators can now do so by setting the **requests.storage** value for user's projects:

```
$ oc create quota my-quota \
  --hard=requests.storage=10Gi,persistentvolumeclaims=50

$ oc describe quota
Name:                my-quota
Namespace:           default
Resource             Used      Hard
-----
persistentvolumeclaims 1        50
requests.storage       10Gi     2Gi
```

See [Setting Quotas](#) for more details.

### 2.3.2.2. Disk-based Pod Eviction for Nodes

Cluster administrators could previously configure nodes' pod eviction policy based on available memory. With this release, eviction policy can now also be configured based on available disk.

Nodes supports two file system partitions when detecting disk pressure:

- ✎ The **nodefs** file system that the node uses for local disk volumes, daemon logs, etc. (for example, the file system that provides /)
- ✎ The **imagefs** file system that the container runtime uses for storing images and individual container writable layers

When configured, the node can report disk threshold violations, and the scheduler no longer tries to put pods on those nodes. The node ranks pods and then evicts pods to free up disk space.

See [Handling Out of Resource Errors](#) for more details.

## 2.3.3. Storage

### 2.3.3.1. Dynamic Storage Provisioning Using Storage Classes

Dynamic provisioning of persistent storage volumes for many storage providers was previously available in OpenShift Container Platform as a [Technology Preview](#) feature, but this release brings this feature into full support using the new *storage classes* implementation for the following:

- ✳ OpenStack Cinder
- ✳ AWS Elastic Block Store (EBS)
- ✳ GCE Persistent Disk (gcePD)
- ✳ GlusterFS
- ✳ Ceph RBD

See [Dynamic Provisioning and Creating Storage Classes](#) for more details.

## 2.3.4. Security

### 2.3.4.1. Service Accounts as OAuth Clients

Users can now more easily integrate with the OpenShift Container Platform-provided OAuth server from their own applications deployed within their project. You can now use service accounts as a scope-constrained OAuth client.

See [Service Accounts as OAuth Clients](#) for more details.

## 2.3.5. Networking

### 2.3.5.1. Subdomain Wildcard Router

Users can now use wildcard routes to determine the destination of all traffic for a domain and its subdomains. For example, `*.foo.com` can be routed to the same back-end service, which is configured to handle all the subdomains.

You can specify that a route allows wildcard support through an annotation, and the HAProxy router exposes the route to the service per the route's wildcard policy. The most-specific path wins; for example, `bar.foo.com` is matched before `foo.com`.

See [Creating Routes Specifying a Wildcard Subdomain Policy](#) and [Using Wildcard Routes \(for a Subdomain\)](#) for more details.

## 2.3.6. Installation

### 2.3.6.1. Upgrade Enhancements

This release includes a number of enhancements to improve the OpenShift Container Platform upgrade process from 3.3 to 3.4, including:

- ✳ A `--tags pre_upgrade` Ansible option for running a dry-run that performs all pre-upgrade checks without actually upgrading any hosts and reports any problems found.
- ✳ New playbooks broken up into smaller steps when possible, allowing you to upgrade the control plane and nodes in [separate phases](#).
- ✳ [Customizable node upgrades](#) by specific label or number of hosts.

- ✱ New **atomic-openshift-excluder** and **atomic-openshift-docker-excluder** packages that help ensure your systems stay locked down on the correct versions of OpenShift Container Platform and Docker when you are not trying to upgrade, according to the OpenShift Container Platform version. Usage is documented in relevant installation and upgrade steps.

## 2.3.7. Enterprise Container Registry

### 2.3.7.1. Image Layout View

A new image layout view has been added to the OpenShift Container Platform web console, providing additional information about specific images in the OpenShift Container Platform registry by clicking on their tags from the **Builds** → **Images** page.

Figure 2.1. Details Tab

Image Streams » my-jboss-app » :latest

my-jboss-app:latest

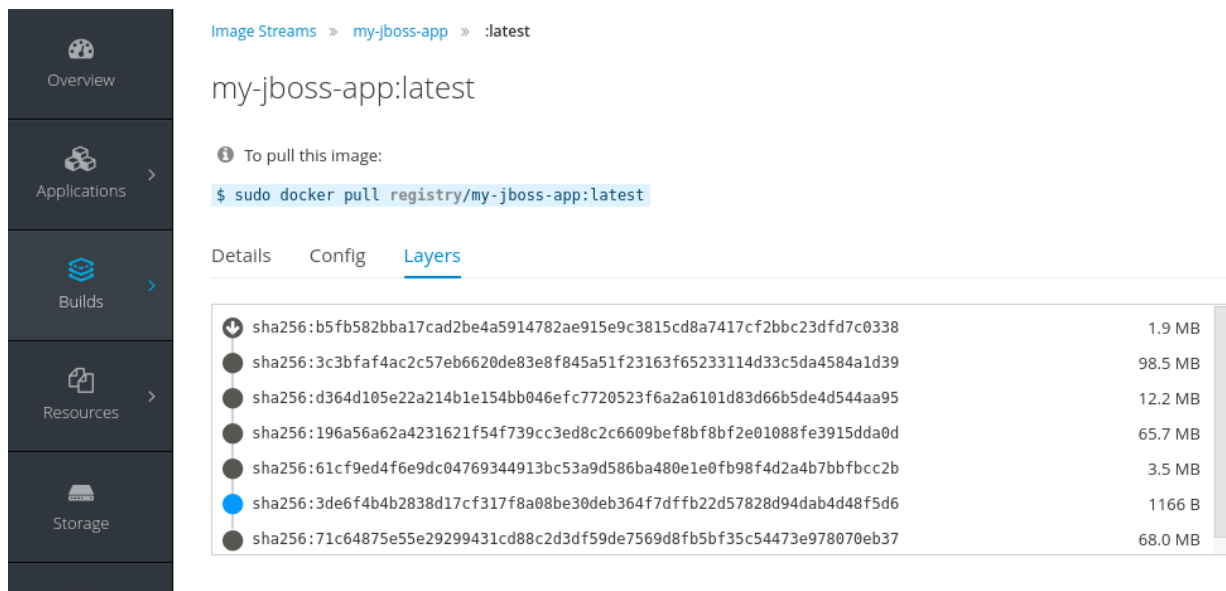
**To pull this image:**

```
$ sudo docker pull registry/my-jboss-app:latest
```

**Details** Config Layers

<b>Name</b>	jboss-webserver-3/webserver30-tomcat8-openshift
<b>Summary</b>	Provides the latest release of Red Hat Enterprise Linux 7 in a fully featured and supported base image.
<b>Description</b>	The Red Hat Enterprise Linux Base image is designed to be a fully supported foundation for your containerized applications. This base image provides your operations and application teams with the packages, language runtimes and tools necessary to run, maintain, and troubleshoot all of your applications. This image is maintained by Red Hat and updated regularly. It is designed and engineered to be the base layer for all of your containerized applications, middleware and utilities. When used as the source for all of your containers, only one copy will ever be downloaded and cached in your production environment. Use this image just like you would a regular Red Hat Enterprise Linux distribution. Tools like yum, gzip, and bash are provided by default. For further information on how this image was built look at the /root/anacanda-ks.cfg file.
<b>Author</b>	Unknown
<b>Built</b>	a month ago
<b>Digest</b>	sha256:a522b5413319bbe5c5192c85c3f7be08f48b10d803398569f5e9241535aca229
<b>Identifier</b>	
<b>Labels</b>	Architecture=x86_64 BZComponent=jboss-webserver-3-webserver30-tomcat8-openshift-docker Name=jboss-webserver-3/webserver30-tomcat8-openshift Release=10

Figure 2.2. Layers Tab



### 2.3.7.2. Support Additional Slashes in Image Tag Names

You can now use external docker distribution servers that support images with more than two path segments. For example:

```
exampleretry.net/project/subheading/image:tag
```

OpenShift Container Platform, however, is still limited to images of the form **\$namespace/\$name**, and cannot create multi-segment images.

## 2.3.8. Developer Experience

### 2.3.8.1. OpenShift Pipelines Fully Supported

OpenShift Pipelines, introduced in OpenShift Container Platform 3.3 as a [Technology Preview](#) feature, are now fully supported. OpenShift Pipelines are based on the [Jenkins Pipeline plug-in](#). By integrating Jenkins Pipelines, you can now leverage the full power and flexibility of the Jenkins ecosystem while managing your workflow from within OpenShift Container Platform.

See the following for more on pipelines:

- ✎ [Pipeline Concept](#)
- ✎ [Configuring Pipeline Execution](#)
- ✎ [Pipeline Strategy Option](#)

### 2.3.8.2. Jenkins 2.0 Image

OpenShift Container Platform users using integrated Jenkins CI and CD pipelines can now leverage Jenkins 2.0 with improved usability and other enhancements.

### 2.3.8.3. Automatically Log in to Integrated Jenkins Using OAuth

Users who deploy a OpenShift Container Platform integrated Jenkins server can now configure it to allow automatic logins from the web console based on an OAuth flow with the master instead of requiring the standard Jenkins authentication credentials.

See [OpenShift Container Platform OAuth Authentication](#) for configuration details.

#### 2.3.8.4. Designated Build Nodes

Cluster administrators can now designate nodes to be used for builds (i.e., Source-to-Image and/or Docker builds) so that build nodes can be scaled independently from the application container nodes. Build nodes can also be configured differently in terms of security settings, storage back ends, and other options.

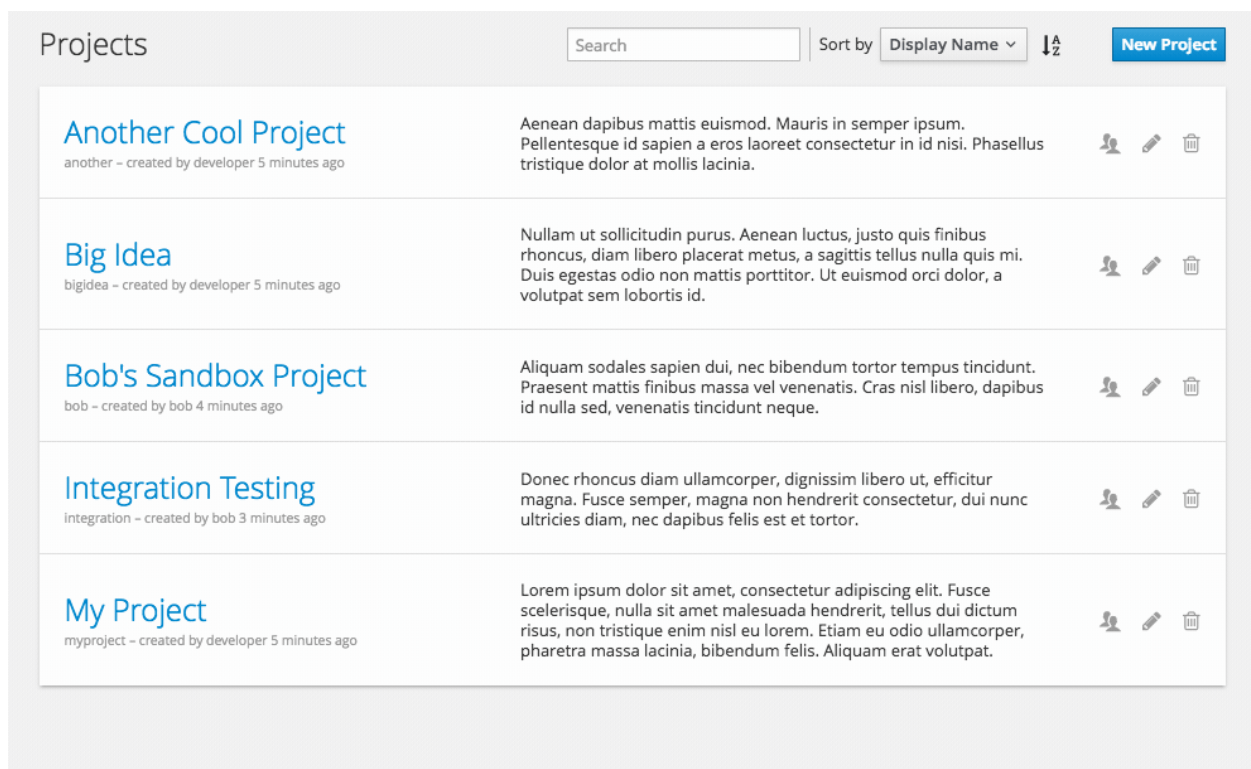
See [Configuring Global Build Defaults and Overrides](#) for details on setting `nodeSelector` to label build nodes, and [Assigning Builds to Specific Nodes](#) for details on configuring a build to target a specific node.

### 2.3.9. Web Console

#### 2.3.9.1. Filtering and Sorting the Projects List

To make navigation easier for users interacting with large numbers of projects, the **Projects** page now has a text filter by name, display name, description, and project creator. It also allows sorting on several of these attributes.

Figure 2.3. Filtering and Sorting Projects



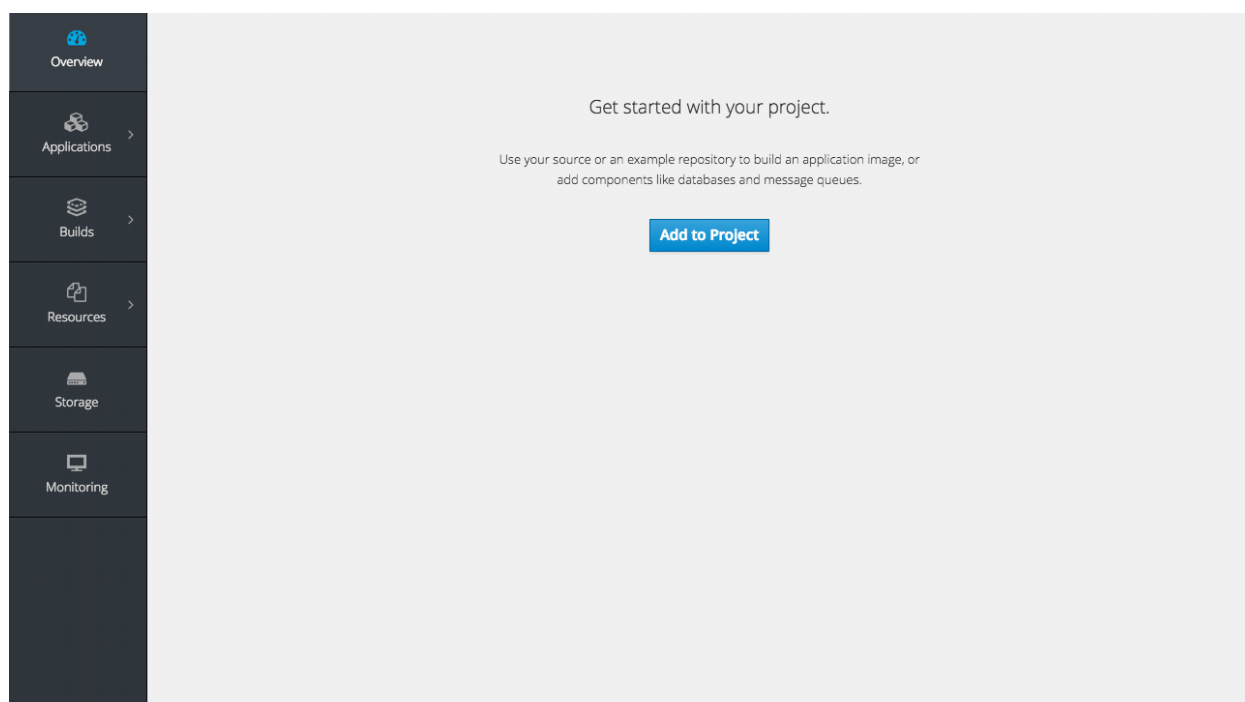
#### 2.3.9.2. Better Catalog Organization and Customizable Categories

The existing **Add to project** catalog could become cluttered when dealing with builder images with many versions or many templates with slight differences. Previously, the focus was minimizing the number of clicks to get an application running, however the updated layout now focuses on helping you find what you are actually looking for.

The main catalog page now only contains high-level categories **Languages** and **Technologies**, and underneath those are subcategories, such as **Java** and **Data Stores**. Clicking one of those shows redesigned tiles for build images and templates. Different versions of the same builder image now all roll-up into the same tile with the semantically **latest** version automatically selected.

All of the default image streams and templates have also now been updated with better display names, descriptions, and categorization.

**Figure 2.4. Catalog Organization**



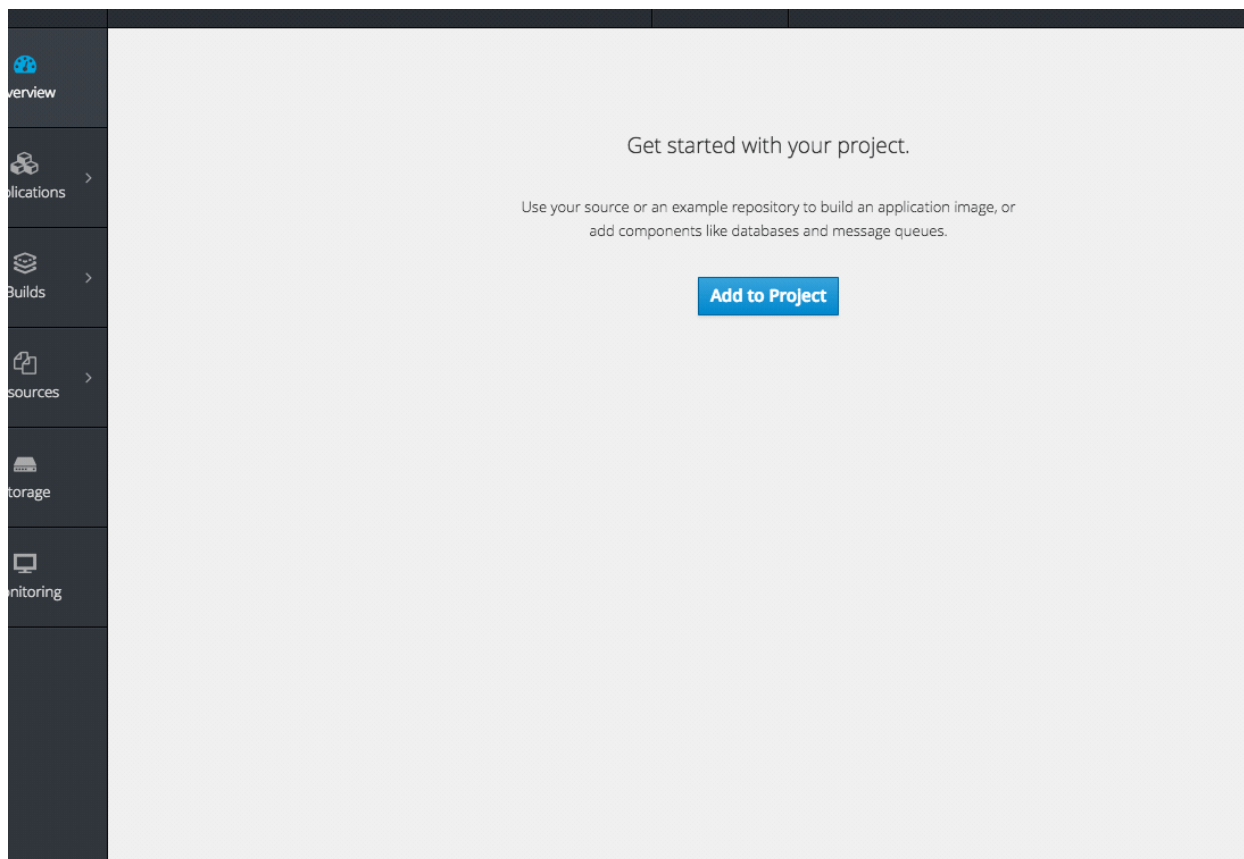
If you do not like the default categories and subcategories, you can also now customize those as well. See [Configuring Catalog Categories](#) for more details.

### 2.3.9.3. Creating and Adding Secrets for Build and Deployment Configurations

It was previously difficult to set up a build against a private Git repository from the web console. You had to import YAML or JSON to create your secret, then edit your build's YAML to make it use that secret.

You can now expand the advanced build options, create a user and password or SSH key-based secret, then specify that the build use that when cloning your source. If you already have your secret created in the project, you can also choose any of the existing ones.

**Figure 2.5. Secrets in Web Console**



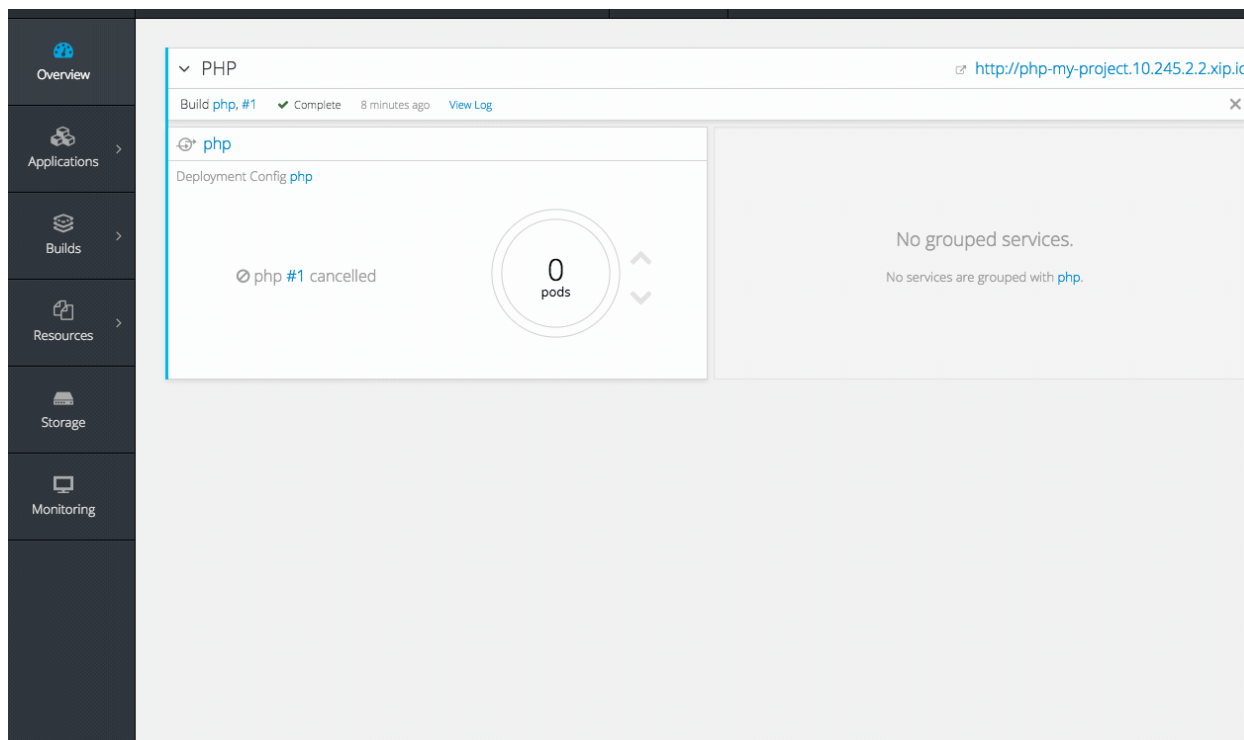
Setting up push and pull against private image registries has also been improved. The build configuration editor now allows you to set up a push or pull secret in case the image you are building from, or the image stream you are pushing to, is in a secure registry. Similarly, the new deployment configuration editor allows you to specify a pull secret.

#### 2.3.9.4. Editing Deployment Configuration Strategy, Hooks, and Secrets

A deployment configuration editor has been added to the web console, similar to the existing build configuration editor. With this new editor, you can:

- ✎ Switch your deployment strategy
- ✎ Modify advanced deployment settings like the maximum number of pods that can be unavailable during the deployment
- ✎ Add, edit, or remove deployment lifecycle hooks
- ✎ Change the image being deployed
- ✎ Set a pull secret for the registry your image is being pull from
- ✎ Add, edit, or remove environment variables for the pods that will be deployed

**Figure 2.6. Deployment Configuration Editor**



Many of the existing editing actions still exist as separate actions, such as editing health checks, or configuring different resource limits. If you want to make a number of changes without triggering a deployment for each change, you can now pause your deployment, make all the changes you want, and then resume it. Pausing prevents any deployment from happening, no matter whether it was automatically or manually triggered.

### 2.3.9.5. Quota Warnings

Users working within quota constraints had a hard time knowing when they had run out of quota, unless they went to check the **Quota** page. To address this, checks have been added for the most common scenarios where people have problems with quota. You now get quota warnings:

- ✎ On the **Overview** as a generic warning if anything in your quota is at its limit.
- ✎ On the **Overview** pod count visualizations when you are unable to reach your scale target due to quota.
- ✎ If you try to create something and you are out of quota for that resource.
- ✎ If you try to create something and it will cause you to exceed quota for a resource.

**Figure 2.7. Quota Warnings**



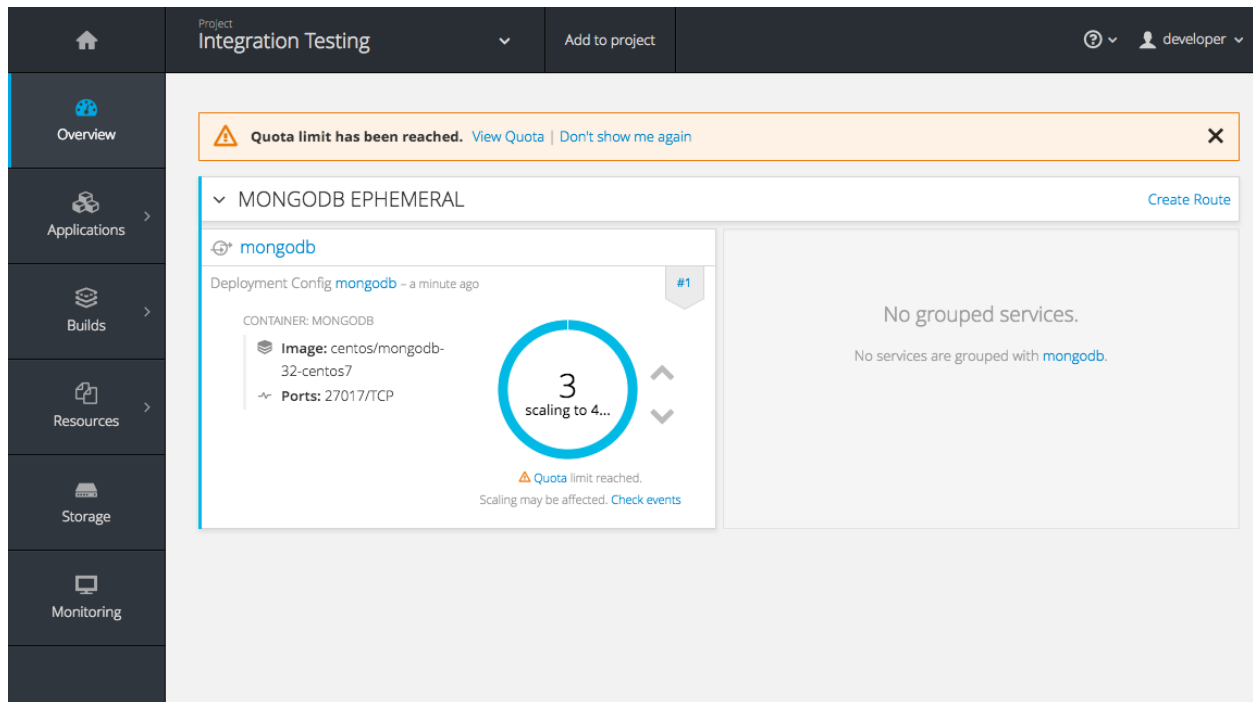
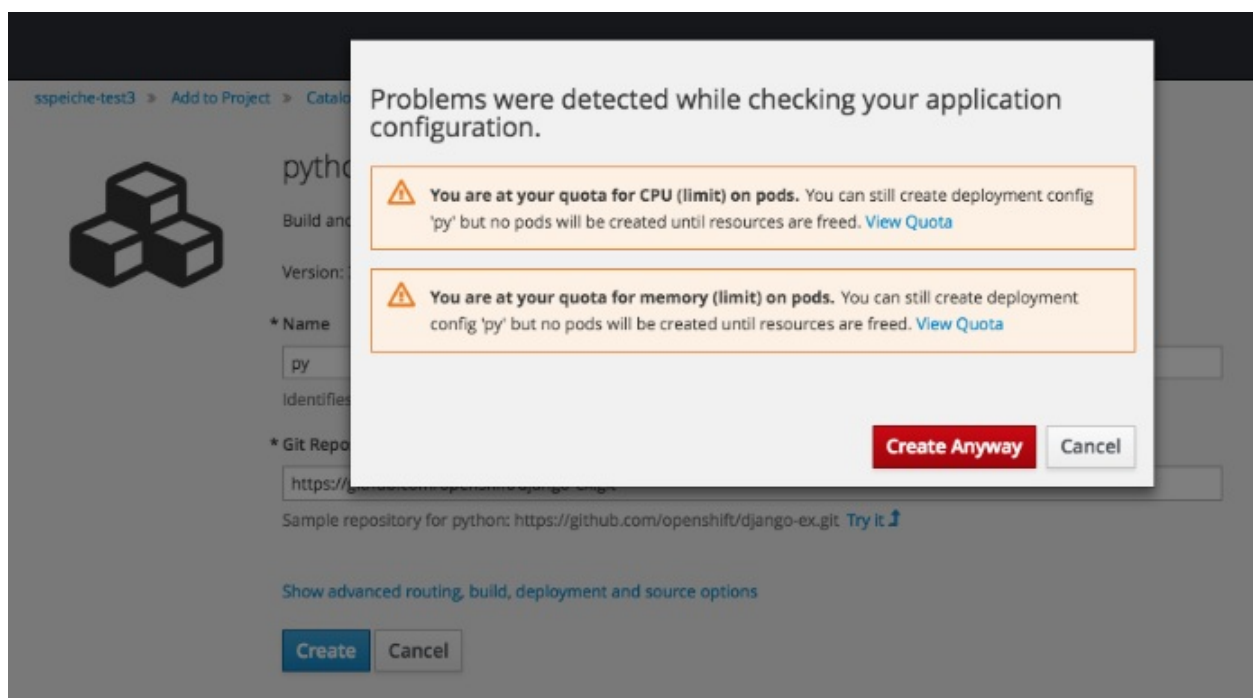


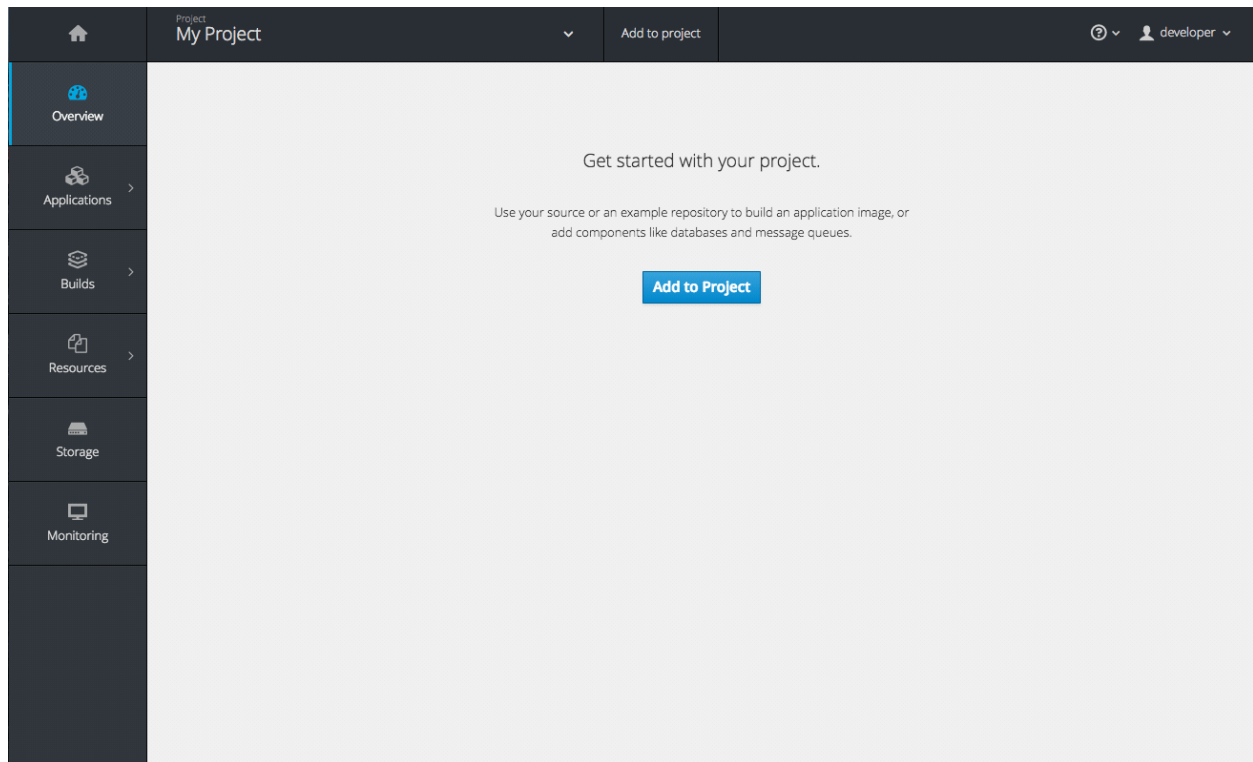
Figure 2.8. Quota Warnings



### 2.3.9.6. Managing Project Membership

An important feature for users that want to collaborate with the same projects, the new membership management interface allows you to add and remove roles to users, groups, and service accounts within your project.

Figure 2.9. Managing Project Membership



Project administrators have access to view and modify the project's membership. Membership management is the only difference between an administrator and an editor in the default OpenShift Container Platform roles. Cluster administrators can add a description to any role to provide extra information for end users about what that role actually allows.

### 2.3.9.7. Bookmarkable Page States

Tab selection, label filters, and several other options that change page state are now persisted to the URL throughout the web console. This allows you to bookmark specific pages and share with others.

### 2.3.9.8. Support for New Kubernetes Features

Support for the following new Kubernetes features have been added to the web console:

#### ✎ Create storage using storage classes

- If your cluster administrator sets up storage classes, they will be available for you to pick from in the **Create Storage** page.

#### ✎ Deployments and ReplicaSets

- Fit in seamlessly on the **Overview** page alongside your existing deployment configurations.
- Appear on the **Applications** → **Deployments** page.
- Support many of the actions already supported for deployment configurations (excluding the new editor).

#### ✎ Roll-up of **PetSet** pods on the **Overview** page

- Pods for a **PetSet** roll up into a single card with a pod count visualization like the other controllers.
- Metrics viewable on the overview for the pods in the **PetSet**.

## 2.4. NOTABLE TECHNICAL CHANGES

OpenShift Container Platform 3.4 introduces the following notable technical changes.

### Updated Infrastructure Components

- ✳ Kubernetes has been updated to v1.4.
- ✳ OpenShift Container Platform 3.4 requires Docker 1.12.
- ✳ etcd has been updated to 3.1.0-rc.0.

While etcd has been updated from etcd 2 to 3, OpenShift Container Platform 3.4 continues to use an etcd 2 data model and API for both new and upgraded clusters.

### Updated Logging Components and Common Data Model

The latest EFK stack has been updated to:

- ✳ Elasticsearch 2.4
- ✳ Kibana 4.5
- ✳ Fluentd 0.12

This stack also now uses a common data dictionary and format for how Red Hat names components, systems, capabilities, and more when referring to them in a log message. As a result, search queries will be able to be reused across other Red Hat products.

This means Fluentd sends logs to Elasticsearch with a new indexing pattern for projects. The pattern is:

```
project.{namespace_name}.{namespace_id}.YYYY.MM.DD
```

For example:

```
project.logging.5dad9bd0-a7a1-11e6-94a0-5254000db84b.2016.11.14
```

The pattern for the **operations** logs remains the same.



#### Important

Downgrading from Elasticsearch 2.4 to Elasticsearch 1.x is not possible due to migration to a new data structure.

### OpenShift SDN Converted to Kubernetes CNI Plug-in

The default OpenShift SDN has been modernized and converted to a Kubernetes CNI plug-in. OpenShift SDN presents itself to Kubernetes as a network plug-in and assumes responsibility for IPAM instead of Docker.

As a side effect, Docker is no longer used for pod IP address management, so running the **docker inspect** command will no longer show the pod's IP address and other network details. Pod IP details are still (and have always been) available through **oc describe pod** command output.

### Miscellaneous Changes

- ✧ The `deploymentConfig.spec.strategy.rollingParams.updatePercent` field is removed in favor of `deploymentConfig.spec.strategy.rollingParams.maxUnavailable` and `deploymentConfig.spec.strategy.rollingParams.maxSurge`.
- ✧ The pre-OpenShift Origin 1.0 compatibility fields for service `spec.portalIP` and pod `spec.host` have been removed from the API. Use `spec.clusterIP` for services and `spec.nodeName` for pods. Clients that send these fields to the server will have those values ignored.
- ✧ The `oc whoami --token` command is deprecated in favor of `oc whoami -t`, and `oc whoami --context` is deprecated in favor of `oc whoami -c`. The `--token` and `--context` options will be removed in a future release.
- ✧ Support for OpenShift Container Platform 3.1 clients for deployment configurations is dropped. More specifically, the `oc scale` command will not work as expected.
- ✧ It is no longer possible to set multiple environment variables or template parameters by passing a comma-separated list to single a `--env`, `--param`, or `--value` option. For example:

```
$ oc new-app mysql --param MYSQL_USER=user,MYSQL_PASSWORD=password
```

will not work, and:

```
$ oc new-app mysql --param MYSQL_USER=user --param  
MYSQL_PASSWORD=password
```

should be used instead.

## 2.5. BUG FIXES

This release fixes bugs for the following components:

### Authentication

- ✧ Project visibility calculation failed if it encountered a role binding that referenced a missing role. Projects containing a role binding that referenced a missing role would not appear when listing projects via the API. This bug fix skips role bindings with invalid role references when evaluating project visibility. As a result, projects with invalid role bindings still appear in the projects list if another valid role binding exists that grants access. ([BZ#1382393](#))

### Builds

- ✧ Pipeline strategies now support run policies: serial and parallel. Previously, pipeline builds were executed independently of the requested run policy associated with the build configuration, which resulted in confusion. With this enhancement, pipeline jobs running in Jenkins now respect the run policy that was specified by the OpenShift build configuration. ([BZ#1356037](#))
- ✧ Parameter references are now supported in non-string template fields. Previously, parameter references could not be used in non-string API fields such as replica count or port. With this enhancement, this is now supported by using the `${{PARAMETER}}` syntax to reference a parameter within the template. ([BZ#1383812](#))
- ✧ When creating a build object through the REST API, the type of the `from` image was not checked and was assumed to be `DockerImage`. Build objects created with a Custom strategy referencing an `ImageStreamTag` as its `from` image resulted in failure or, potentially, a build

using the wrong image. This bug fix checks the type of builder image when creating build objects, and if it is not **DockerImage**, the request is rejected as invalid. As a result, Custom builds with builder images specified as **ImageStreamTag** are rejected. ([BZ#1384973](#))

- ✳ The code that launches the build container in Source-to-Image was waiting indefinitely when an error occurred that was not a timeout error. This caused failed builds to hang indefinitely in Running state. This bug fix updates Source-to-Image to no longer wait for containers once an error is received. As a result, builds now fail as expected and no longer hang in Running state. ([BZ#1390749](#))
- ✳ Multiple Jenkins builds were being triggered for a single OpenShift build. This caused build details to appear to sync inconsistently. This bug fix ensures only a single Jenkins build is triggered for each OpenShift build. As a result, build details sync properly and the web console displays the pipeline properly. ([BZ#1390865](#))
- ✳ The **oc start-build --follow** command could return a timeout error if there were delays in scheduling the build. With this bug fix, **oc start-build --follow** now blocks until the build completes. ([BZ#1368581](#))
- ✳ **NO\_PROXY** values can now be set for **git clone** operations in builds. Previously, cluster administrators could set **HTTP\_PROXY** and **HTTPS\_PROXY** values that would be used for all builds. Certain builds needed to access domains that could not be reached when going through those default proxies. Adding a **NO\_PROXY** field allows the cluster administrators to set domains for which the default proxy value will not be used. With this enhancement, default proxies can now be bypassed when performing **git clone** operations against specific domains. ([BZ#1384750](#))
- ✳ The generic webhook build trigger would cause builds to run even when invalid content was POSTed in the request body. This behavior has been maintained for backwards compatibility reasons, but this bug fix adds a warning to make the situation clearer to whoever is calling the trigger. ([BZ#1373330](#))

## Command Line Interface

- ✳ During builds, comparison of the master host and port with that specified by the user failed when the user-specified URL did not contain the default port (when using 443). This caused builds to fail to trigger. This bug fix updates the comparison of the host and port to account for the default port. As a result, starting builds works when the master port is 443 and is using a self-signed certificate. ([BZ#1373788](#))
- ✳ The **oc new-app --search** command expected that the cluster could always reach **registry-1.docker.io**. When **registry-1.docker.io** was unreachable, as is the case when running a disconnected cluster, the command would always fail. With this bug fix, the command now prints a warning when **registry-1.docker.io** is unreachable and no longer fails with an error. As a result, the command is now usable in disconnected environments or in other circumstances when **registry-1.docker.io** is unreachable. ([BZ#1378647](#))
- ✳ An extra line of information caused invalid JSON or YAML output when using the **oc set** command. With this bug fix, the extra line of information is now output through stderr. As a result, valid JSON or YAML is now printed via the **oc set** command. ([BZ#1390140](#))
- ✳ The **oc convert** command failed to produce a YAML file with valid syntax when converting from multiple files in a directory. When converting from multiple files in a directory and piping the output to **oc create**, it would only create the first file converted. This bug fix updates the YAML syntax in the output of **oc convert** when converting multiple files. As a result, the output of **oc convert** can feed **oc create** properly. ([BZ#1393230](#))

- ✎ The **oc adm prune images|builds|deployments** commands ignored the **--namespace** parameter. This made cluster administrators unable to limit the scope of prune commands to particular namespaces. This bug fix makes the **oc adm prune** command aware of the **--namespace** parameter and limits the scope of pruning to the given namespace. As a result, cluster administrators are now able to limit the scope of the command to single namespace. When applied to images, none of the images will be removed, because images are non-namespaced. ([BZ#1371511](#))

## Containers

- ✎ Docker versions earlier than 1.12 required IPv6, which made it impossible to run the docker daemon on a kernel with IPv6 disabled. This bug fix modifies the docker daemon to no longer require IPv6. ([BZ#1354491](#))

## Deployments

- ✎ The **oc deploy --latest** command previously updated **latestVersion** directly from the API, which made it impossible to separate between manual and automatic updates. This enhancement adds an instantiate endpoint for deployment configurations, allowing for distinction between these types of updates. As a result, the API call for a manual deployment is now distinguishable. ([BZ#1371403](#))
- ✎ A deployment configuration with multiple containers using the same **ImageChangeTrigger** would not be updated by the image change controller. This bug was fixed as part of redesigning the triggering mechanism, which removed the image change controller. ([BZ#1381833](#))
- ✎ The pause and resume operations are now handled using the PATCH method, which ensures the operation always succeeds for the user. ([BZ#1388832](#))
- ✎ When **Autodeploy when: New image is available** was unchecked in the web console's **Add to project** page, the web console would not create an image change trigger on the new deployment configuration. This meant that users had to manually set an image using the **oc set image** command before deployments. Otherwise, all deployments would fail with image pull back-off errors.
- ✎ This bug fix updates the web console to add an image change trigger with **automatic: false**. This prevents deployments from happening automatically when the image stream tag is updated, but allows users to run **oc rollout** commands, or use the **Deploy** action in the web console, without any additional configuration. ([BZ#1383804](#))
- ✎ It was impossible to specify when to start a deployment with the latest image. Triggers would cause each build to deploy. So triggers had to be disabled, then enabled once a deploy is desired. With this bug fix, a new endpoint and **oc rollout latest** that uses the endpoint and supersedes **oc deploy --latest** were added in OpenShift Container Platform 3.4 to enable manual deployments without the need to enable triggers. ([BZ#1303938](#))

## Image

- ✎ Various OpenShift sample templates included an expired, self-signed X.509 certificate and key for **www.example.com**. These unnecessary certificates and keys have been removed from the templates. ([BZ#1312278](#))
- ✎ The Jenkins Sync plug-in failed to consistently sync build changes from the OpenShift cluster. Builds created in OpenShift were therefore not observed and executed by the Jenkins server. This bug fix makes sync logic more robust to ensure changes are not missed. As a result, builds are now properly processed by the sync plug-in and executed in Jenkins. ([BZ#1364948](#))
- ✎ API server restarts caused the Jenkins sync plug-in to lose its connection to OpenShift. This

caused pipeline builds to not be properly executed in the Jenkins server. This bug fix updates the sync plug-in to handle connection loss when the API server is restarted. As a result, builds are now properly processed by the sync plug-in and executed in Jenkins if the API server is restarted. ([BZ#1364949](#))

- ✳ New build configuration events were missed, causing associated Jenkins jobs to not be created. This bug fix ensures the order of resource watches is correct and periodically resyncs to prevent missing events. As a result, associated Jenkins jobs are now always created. ([BZ#1392353](#))
- ✳ The pipeline plug-in did not use an optimal endpoint for scaling. This made scaling beyond one replica problematic. This bug fix updates the pipeline plug-in to use an optimal endpoint, and users can now scale a deployment configuration's replication controller beyond one replica. ([BZ#1392780](#))
- ✳ Failure to use overrides methods in one area of the Jenkins plug-in caused job failures when **namespace** parameter was not set. This bug fix updates the plug-in, and **namespace** is now an optional parameter. ([BZ#1396022](#))

## Image Registry

- ✳ This enhancement updates OpenShift Container Platform to allow multiple slashes in Docker image names and allows using external registries that support them. ([BZ#1373281](#))
- ✳ When importing a Docker image from a remote registry that is insecure, the pull-through capability did not work, causing pull failures. This bug fix ensures that these pulls now succeed for insecure registries. ([BZ#1385855](#))
- ✳ Previous versions of docker only checked for the existence of one layer digest in remote repositories before falling back to the full blob upload. However, each layer can have multiple digests associated depending on the docker version used to push images to a source registry. During an image push, the docker daemon could have picked up the wrong layer digest associated to a particular image layer, which did not exist in remote repository. It would then fall back to the full blob upload, even though the daemon knew another digest existing in the remote repository. With this bug fix, the docker daemon now sorts candidate layer digests by their similarity with the remote repository and iterates over a few of them before falling back to full blob re-upload. As a result, docker pushes are now faster when layers already exist in the remote registry. ([BZ#1372065](#))

## Installer

- ✳ The installer generated a flannel configuration that was not compatible with the latest version of flannel available in Red Hat Enterprise Linux 7. The installer has been updated to produce configuration files compatible with both the new and old versions of flannel. ([BZ#1391515](#))
- ✳ Previously, openshift-ansible did not configure environments using Google Compute Engine (GCE) as multizone clusters. This prevented nodes from different zones registering against masters. With this bug fix, GCE-based clusters are multizone enabled, allowing nodes from other zones to register themselves. ([BZ#1390160](#))
- ✳ This enhancement moves the node scale-up workflow in the quick installer out of the **install** subcommand and into a separate **scaleup** subcommand. Users reported that having the scaleup workflow inside install was confusing, and a result scale-up now lives in its own space and users can access it directly. ([BZ#1339621](#))
- ✳ This feature provides the ability to add persistent node-labels to hosts. Rebooting hosts (such as in cloud environments) would not have the same labels applied after reboot. As a result, node-labels persist across reboot. ([BZ#1359848](#))
- ✳ The **openshift-ansible** NetworkManager configuration script was unconditionally restarting



the dnsmasq service every time it ran. As a result, host name resolution would fail temporarily while the dnsmasq service restarted. The **openshift-ansible** NetworkManager configuration script now only restarts the dnsmasq service if a change was detected in the upstream DNS resolvers. As a result, host name resolution will continue to function as expected. (BZ#1374170)

- ✳ Previously, the installer would re-run the metrics deployment steps if the configuration playbook was re-run. The playbooks are now updated to only run the metrics deployment tasks once. If a previous installation of metrics has failed, the administrator must manually resolve the issue or remove the metrics deployment and re-run the configuration playbook. See the [cleanup instructions](#). (BZ#1383901)
- ✳ The Ansible **quiet output** configuration was not set for non-install runs of **atomic-openshift-installer**. As a result, users would see full Ansible output rather than abbreviated step-by-step output. The Ansible **quiet output** configuration is now set as the default for all **atomic-openshift-installer** runs. With this fix, users see abbreviated output and can toggle back to verbose output with **-v** or **--verbose**. (BZ#1384294)
- ✳ Previously, the quick installer would unnecessarily prompt for the name of a load balancer for non-HA installations. This question has been removed for single master environments. (BZ#1388754)
- ✳ The **a-o-i** package was considering extra hosts when determining if the target HA environment is a mix of installed and uninstalled hosts. As a result, the comparison failed and incorrectly reported that a fully installed environment was actually a mix of installed and uninstalled. With this fix, non-masters and non-nodes were removed from the comparison and installed HA environments are correctly detected. (BZ#1390064)
- ✳ Previously, the dnsmasq configuration included **strict-order**, meaning that dnsmasq would iterate through the host's nameservers in order. This meant that if the first nameserver had failed, a lengthy timeout would be observed while dnsmasq waited before moving on to the next nameserver. By removing the **strict-order** option, dnsmasq prefers nameservers that it knows to be up over those that are unresponsive, ensuring faster name resolution. If you wish to add this or any other option, use the advanced installer option **openshift\_node\_dnsmasq\_additional\_config\_file**, which allows you to provide the path to a dnsmasq configuration file that will be deployed on all nodes. (BZ#1399577)
- ✳ Previously, the NetworkManager dispatcher script did not correctly update **/etc/resolv.conf** after a host was rebooted. The script has been updated to ensure that **/etc/resolv.conf** is updated on reboot, ensuring proper use of dnsmasq. (BZ#1401425)
- ✳ The openshift-ansible advanced install method now alters the Registry Console's **IMAGE\_PREFIX** value to match the **oreg\_url** prefix when **openshift\_examples\_modify\_imagestreams=true**, allowing users to install from a registry other than **registry.access.redhat.com**. (BZ#1384772)
- ✳ **openshift\_facts** was parsing full package versions from **openshift version**. The parsed versions do not match actual **yum** package versions. With this fix, **openshift\_facts** is updated to remove **commit offset** strings from parsed versions. Parsed versions now match actual **yum** package versions. (BZ#1389137)
- ✳ Previously, if hosts defined in the advanced installation inventory had multiple inventory names defined for the same hosts, the installer would fail with an error when creating **/etc/ansible/facts.d**. This race condition has been resolved, preventing this problem from happening. (BZ#1385449)

## Kubernetes



- ✳ This feature adds the ability to define eviction thresholds for **imagefs**. Pods are evicted when the node is running low on disk. As a result, the disk is reclaimed and the node remains stable. ([BZ#1337470](#))
- ✳ This bug fixes an issue with the OpenShift master when the OpenStack cloud provider is used. If the master service controller is unable to connect with the LBaaS API, it prevents the master from starting. With this fix, the failure is treated as non-fatal. Services with type **LoadBalancer** will not work, as the master is able to create the load balancer in the cloud provider, but the master functions normally. ([BZ#1389205](#))
- ✳ This feature adds the ability to detect local disk pressure and reclaim resources. To maintain stability of the node, the operator is able to set eviction thresholds that, when crossed, will cause the node to reclaim disk resource by pruning images, or evicting pods. As a result, the node is able to recover from disk pressure. ([BZ#1352390](#))
- ✳ Previously, it was possible to configure resource (CPU, memory) eviction thresholds (hard and soft) to a negative value and the kubelet started successfully. As eviction thresholds can not be negative, this erroneous behavior is now fixed. The kubelet now fails to start if a negative eviction threshold is configured. ([BZ#1357825](#))
- ✳ The pod container status field **ImageID** was previously populated with a string of the form **docker://SOME\_ID**. This displayed an image ID, which was not usable to correlate the image running in the pod with an image stored on a registry. Now, the **ImageID** field is populated with a string of the form **docker-pullable://sha256@SOME\_ID**. This image ID may be used to identify and pull the running image from the registry unambiguously. ([BZ#1389183](#))
- ✳ The **oc logs** command was using a wrapped word writer that could, in some cases, modify input such that the length of output was not equal to the length of input. This could cause a **ErrShortWrite** (short write) error. This change restores **oc logs** to use Golang's standard output writer. ([BZ#1389464](#))
- ✳ The default directory for the location of Seccomp profile JSON files on the node was not set properly. As a result, there was an issue when using the Seccomp profile annotation in a pod definition. With this fix, the default Seccomp profile directory is appropriately set to **/var/lib/kubelet/seccomp**. ([BZ#1392749](#))
- ✳ OpenShift uses **fsGroup** in the pod specification to set volume permissions in unprivileged pods. The **S\_ISGID** bit is set on all directories in the volume so that new files inherit the group ID. However, the bit is also set for files, for which it has a different meaning of **mandatory file locking, see stat(2)**. This fix ensures that the **S\_ISGID** bit is now only set on directories. ([BZ#1387306](#))
- ✳ This bug fix corrects an issue on the OpenShift master when using the Openstack cloud provider. The LBaaS version check was done improperly, causing failures when using v2 of the LBaaS plug-in. This fix corrects the check so that v2 is detected properly. ([BZ#1391837](#))
- ✳ While autoscaling, the reason for the failed `--max`` flag validation was unclear. This fix divides reasons into `* value not provided or too low*` or **value of max is lower than value of min**. ([BZ#1336632](#))

## Logging

- ✳ Piping to **oc volume** from **oc process** would not create the deployment configuration (DC) as it did before. As a result, the deployer would provide output stating that the DC that would be generated did not exist, and would fail. With this fix, the output of **oc volume** to **oc create** is properly piped. As a result, you can create the missing DC with the PVC mount when you have the deployer attaching PVC to ES upon creation. The deployer no longer fails. ([BZ#1396366](#))

## Web Console

- ✧ A JavaScript bug caused the HTML page to not refresh after deleting the route in Camel. This fix addresses the JavaScript bug and the HTML page is refreshed after deleting the route. ([BZ#1392416](#))
- ✧ Tables with label filters will persist the current filter into the URL. Clicking directly into a pre-filtered pod list, clicking somewhere else, and then hitting **Back** took you back to the entire pod list instead of the filtered one. This behavior was not expected. Now, the latest filtering state a page is on will be persisted into the URL and work with browser history. ([BZ#1365304](#))
- ✧ Previously, the deployment configuration on the **Overview** page was not shown when it had not yet run a deployment. With this update, a tile is shown for the deployment configuration. If the deployment configuration has an image change trigger, a link to the image stream of the tag it will trigger on is shown. ([BZ#1367379](#))
- ✧ The web console would not show any errors on the **Overview** page when metrics were configured, but not working. It would quietly fall back to the behavior when metrics were not set up. The web console now shows an error message with a link to the metrics status URL to help diagnose problems such as invalid certificates. The alert can be permanently dismissed for users who do not want to see it. ([BZ#1382728](#))
- ✧ In some cases, the Y-axis values would not adjust to fit the data when looking at metrics for a pod. The Y-axis now scales appropriately to fit the data as usage increases. ([BZ#1386708](#))
- ✧ If you deleted a pod and created a new pod with the same name, you would see metrics for the previous pod when viewing metrics. Only metrics for the new pod are now shown. ([BZ#1386838](#)\*)
- ✧ When a pod had more than one container, the web console was incorrectly showing total memory and CPU usage for all containers in the pod on the metrics page rather than only the selected container. This could make it appear that memory usage exceeded the limit set for the container. The web console now correctly shows the memory and CPU usage only for the selected container. ([BZ#1387274](#))
- ✧ The logo and documentation links must be changed for each release. This was not yet completed, so the logo and documentation links represented OpenShift Origin instead of OpenShift Container Platform. The appropriate logo and links for the release were added and are now correct. ([BZ#1388798](#))
- ✧ Previously, you could select **Push Secret** and **Pull Secret** on the DC editor page and on the **Create From Image** page. These options are not helpful on these pages because they are using integrated registry. Therefore, the **Push Secret** and **Pull Secret** select boxes are now removed from the DC editor and **Create From Image** pages and users can no longer select these options. ([BZ#1388884](#))
- ✧ Routes popover warning messages were being truncated at the end of the string. Before the relevant portion of the warning message could be displayed, the certificate content results in the warning message were being truncated. After the bug fix, the truncation of the warning message was changed from truncating at the end of the string to truncating in the middle of the string. As a result, the relevant portion of the warning message is now visible. ([BZ#1389658](#))
- ✧ Camel route diagrams had a typo that, on hover, route component showed **Totoal**. As a result of this bug fix, on hover the route component shows **Total**. ([BZ#1392330](#))
- ✧ The password field was set as type **text**, and therefore the password was visible. In this bug fix, the password field type was set to **password**. As a result, the password is not visible. ([BZ#1393290](#))

- ✎ Previously, the **BuildConfig** editor displayed a blank section. The **BuildConfig** editor now shows a message when there are no editable source types for a **BuildConfig**. (BZ#1393803)
- ✎ A bug in the communication between the **Web console** and **Jolokia endpoint** caused an error on the server when activating tracing. This bug fix changed the default value of Apache Camel configuration. As a result, the error is resolved. (BZ#1401509)
- ✎ A bug in the processing of Apache Camel routes defined in XML caused an error in the Apache Camel application. This bug fix corrected the XML by adding expected namespaces, resolving the error in the Apache Camel application. (BZ#1401511)
- ✎ On the Web Console's **BuildConfig** edit screen, the **Learn more** link next to **Triggers** gave a 404 Not Found error. The help link in the console contained the .org suffix instead of .com, therefore the build triggers help link would return a 404 because the requested page did not exist under the <https://docs.openshift.org> website. In the bug fix, the help link was updated to the correct URL. The help link now loads the correct help documentation for OpenShift Container Platform. (BZ#1390890)
- ✎ A bug in the JavaScript code prevented the profile page from showing expected content. The bug was fixed and the profile page displays the expected content. (BZ#1392341)
- ✎ A bug in the JavaScript code prevented message from changing after the Camel route source update. The bug was fixed and the message changes after the Camel route source update. (BZ#1392376)
- ✎ A bug in the JavaScript code prevented the delete header button from functioning. The bug fix enabled the delete header button. (BZ#1392931)
- ✎ A bug in the JavaScript code prevented content from being displayed in the **OSGi Configuration** tab. As a result of the bug fix, content is displayed appropriately on the **OSGi Configuration** tab. (BZ#1393693)
- ✎ A bug in the JavaScript code prevented content from being displayed in the **OSGi Server** tab. As a result of the bug fix, content is displayed appropriately on the **OSGi Server** tab. (BZ#1393696)
- ✎ The **OSGi Bundles** tab showed “TypeError: t.bundles.sortBy is not a function”. The error was a result of the function **sortBy** of Sugar JavaScript library not being included in the application. This bug fix changed the reference to Sugar JavaScript library to an equivalent function in Lodash library. As a result, content is displayed appropriately on the **OSGi Bundles** tab. (BZ#1393711)

## Metrics

- ✎ The scripts used to check if a deployment was successful did not properly handle the situation with dynamically provisioned storage and would cause an error message to be displayed after the metric components were deployed. The deployer would exit in an error status and display an error message in the logs. The metrics components would still deploy and function properly, it did not affect any functionality. In this bug fix, the scripts used to check if the deployment was successfully deployed were updated to support dynamically provisioned volumes when used on GCE. As a result, new deployments to GCE with **DYNAMICALLY\_PROVISIONED\_STORAGE** set to **true** will no longer result in an error message. (BZ#1371464)

## Networking

- ✎ Previously, nodes in an OpenShift cluster using **openshift-sdn** would occasionally report readiness and start assigned pods before networking was fully configured. Nodes now only report readiness after networking is fully configured. (BZ#1384696)

- ✳ When trying to merge the network between different projects, the wrong field was passed to UpdatePod. The network namespace was not correctly merged because the string passed was invalid. With this bug fix, the field to be passed was corrected. The network namespaces are now correctly merged. ([BZ#1389213](#))
- ✳ The Docker service adds rules to the iptables configuration to support proper network functionality for running containers. If the service is started before the iptables, these rules are not properly created. Ensure iptables are started prior to starting Docker. ([BZ#1390835](#))
- ✳ Sometimes with the presence of a pod, OpenShift would perform unnecessary cleanup steps. However the default networking plugin assumed it would only be called to do cleanup when there was cleanup to be done. This would occasionally cause Nodes to log the error "Failed to teardown network for pod" when there was no actual error. Typically, this error would only be noticed in the logs by users who were trying to find the cause of a pod failure. With this bug fix, the default networking plugin now recognizes when it has been called after the pod networking state has already been cleaned up successfully. And as a result, no spurious error message is logged. ([BZ#1359240](#))

## Quick Starts

- ✳ The Python image was overly restrictive about allowing host connections by default, causing readiness probes to fail because they could not connect from **localhost**. With this bug fix, the defaults were changed to allow connections from any host, including **localhost**. As a result, the readiness probe is able to connect from **localhost** and the readiness probe will succeed. ([BZ#1391145](#))

## REST API

- ✳ Because the finalization mechanism only read the preferred resources available in cluster, **ScheduledJobs** were not removed during project deletion. This bug fix enforces read all resources for finalization and garbage collection, not just the preferred. **ScheduledJobs** are now removed during project deletion. ([BZ#1391827](#))
- ✳ Active jobs were mistakenly counted during synchronization. This caused the active calculation to be wrong, which led to new jobs not being scheduled when **concurrencyPolicy** was set to **Replace**. This bug fix corrected how active jobs for a ScheduledJob are calculated. As a result, **concurrencyPolicy** should work as expected when set to **Replace**. ([BZ#1386463](#))

## Routing

- ✳ Generated hostnames with more than 63 characters caused DNS to fail. This bug fix added more stringent validation of the generated names. As a result, the error is caught for the user when the route is processed by the router, and provide a clear explanation why the route will not work. ([BZ#1337322](#))
- ✳ By default extended certificate validation was not enabled, so bad certificates in routes could crash the router. In this bug fix, the default in **oc adm router** was changed to turn on extended validation when a router is created. Now bad certificates are caught and the route they are associated with is not used, and an appropriate status is set. ([BZ#1379701](#))
- ✳ The **clusterrole** has always been able to list the services in a cluster. With this bug fix the role was enabled cluster-wide. The tests that were using this role in limited scope have been fixed to use it across the cluster. ([BZ#1380669](#))
- ✳ The extended certificate validation code would not allow some certificates that should be considered valid. Self-signed, expired, or not yet current certificates that were otherwise well-formed would be rejected. The extended validation was changed to allow those cases. Those types of certificates are now allowed. ([BZ#1389165](#))

## Storage

- ✳ When a volume fails to detach for any reason, the delete operation is retried forever, whereas the detach operation does not seem to try to detach more than once. This causes the delete to fail each time with a “VolumeInUse” error. OpenShift makes requests to delete volumes without any sort of exponential back off. Making too many requests to the cloud provider can exhaust the API quota. This bug fix implemented exponential backoff when trying to delete a volume. OpenShift now uses exponential backoff when it tries to delete a volume, and it does not overshoot the API quota easily. ([BZ#1399800](#))
- ✳ Using hostPath for storage could lead to running out of disk space, causing OpenShift root disk could become full and unusable. This bug fix added support for pod eviction based on disk space. As a result, a pod using hostPath consumes too much space it may be evicted from the node. ([BZ#1349311](#))
- ✳ The cloud provider was not initializing properly, causing features that require cloud provider API access, such as **PersistentVolumeClaim** creation, to fail. With this bug fix, the cloud provider is initialized in node. Features that require cloud provider API access no longer fail. (BZ#1390758) ([BZ#1379600](#))

## Upgrade

- ✳ Previously the upgrade playbook would inadvertently upgrade etcd when it should not have. If this upgrade triggered an upgrade to etcd3 then the upgrade would fail as etcd would become unavailable. With this bug fix, etcd no longer updates when it is not necessary ensuring upgrades proceed successfully. ([BZ#1393187](#))
- ✳ An error in the etcd backup routine run during upgrade could incorrectly interpret an external etcd host as embedded. The etcd backup would fail and the upgrade would exit prematurely, before making any changes on the cluster. This bug fix changed the variable to correctly detect embedded versus external etcd. The etcd backup will now complete successfully allowing the upgrade to proceed. ([BZ#1398549](#))
- ✳ The metrics deployer image shipped in OpenShift Container Platform 3.3.0 had an outdated version of the client included in the image. As a result the the deployer failed with an error when run in the refresh mode. That image has been rebuilt and the deployer no longer fails. ([BZ#1372350](#))

## 2.6. TECHNOLOGY PREVIEW FEATURES

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

### Technology Preview Features Support Scope

The following new features are now available in Technology Preview:

- ✳ [Kubernetes Deployments Support](#)
- ✳ [Pod Disruption Budgets](#)

The following features that were formerly in Technology Preview from a previous OpenShift Container Platform release are now fully supported:

- ✳ [OpenShift Pipelines](#)
- ✳ [Extended Builds](#)

- ✳ [Service Serving Certificate Secrets](#)

- ✳ [Dynamic Storage Provisioning](#)

The following features that were formerly in Technology Preview from a previous OpenShift Container Platform release remain in Technology Preview:

- ✳ [Scheduled Jobs](#)

## 2.7. KNOWN ISSUES

The following are known issues for the OpenShift Container Platform 3.4 initial GA release.

### Upgrade

- ✳ Previously, upgrading from OpenShift Container Platform 3.3 to 3.4 caused all user identities to disappear, though they were still present in etcd, and OAuth-based users could no longer log in. New 3.4 installations were also affected. This was caused by an unintentional change in the etcd prefix for user identities; egressnetworkpolicies were similarly affected.

This bug has been fixed as of the [OpenShift Container Platform 3.4.0.40 release](#). The bug fix restores the previous etcd prefix for user identities and egressnetworkpolicies, and as a result users can log in again successfully.

If you had previously already upgraded to 3.4.0.39 (the GA release of OpenShift Container Platform 3.4), after upgrading to the 3.4.0.40 release you must also then perform a data migration using a data migration tool. See the following Knowledgebase Solution for further details on this tool:

<https://access.redhat.com/solutions/2887651>  
([BZ#1415570](#))

- ✳ An etcd performance issue has been discovered on new and upgraded OpenShift Container Platform 3.4 clusters. See the following Knowledgebase Solution for further details:

<https://access.redhat.com/solutions/2916381>  
([BZ#1415839](#))

## 2.8. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Container Platform 3.4 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 3.4 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 3.4. Versioned asynchronous releases, for example with the form OpenShift Container Platform 3.4.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-2017:0186 - OpenShift Container Platform 3.4.0.40 Bug Fix Update

OpenShift Container Platform release 3.4.0.40 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:0186](#) advisory. The list of container images included in the update are documented in the [RHBA-2017:0187](#) advisory.

### 2.8.1.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.3 or 3.4 cluster to this latest release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

If you had previously already installed or upgraded to 3.4.0.39 (the GA release of OpenShift Container Platform 3.4), after upgrading to the 3.4.0.40 release you must also then perform a data migration using a data migration tool. See the following Knowledgebase Solution for further details on this tool:

<https://access.redhat.com/solutions/2887651>

## 2.8.2. RHBA-2017:0218 - OpenShift Container Platform 3.4.1.2 Bug Fix Update

OpenShift Container Platform release 3.4.1.2 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:0218](#) advisory. The list of container images included in the update are documented in the [RHBA-2017:0219](#) advisory.

### 2.8.2.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.3 or 3.4 cluster to this latest release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

### 2.8.2.2. Bug Fixes

This release fixes bugs for the following components:

#### Build

- ✱ Proxy value validation prevented the use of default cluster proxy settings with SSH Git URLs. This caused build configurations that used SSH Git URLs in a cluster with default proxy settings to get a validation error unless the proxy value was explicitly set to empty string in the build configuration. This bug fix ensures that validation no longer rejects build configurations that use

SSH Git URLs and have a proxy value set. However, the proxy value will not be used when an SSH Git URL is supplied. ([BZ#1397475](#))

- ✳ The builds controller iterates through all builds in the system and processes completed builds to see if there are other builds that need to be started. It will continue iterating through completed builds regardless of when the build was completed. Scheduling a new build can take a long time when there is a great number of completed builds for the controller to process, for example more than 1000. To address this build controller performance issue, this bug fix ensures that a build is processed once only on completion to decide whether other builds should be started; they are ignored in the regular build controller loop. As a result, builds start quickly regardless of how many completed builds exist in the system. ([BZ#1400132](#))

## Command Line Interface

- ✳ A race condition was found when updating a batch of nodes to schedule or unschedulable with **oc adm manage-node --schedulable=<true|false>**. This made several nodes unable to update and show an "object has been modified" error. This bug fix uses a patch on the **unschedulable** field of the node object instead of a full update. As a result, all nodes can now be properly updated schedulable or unschedulable. ([BZ#1416509](#))

## Kubernetes

- ✳ The **us-east-2c**, **eu-west-2**, **ap-south-1**, and **ca-central-1** AWS regions have been added to OpenShift Container Platform, enabling cloud provider support for those regions. ([BZ#1400746](#))

## Web Console

- ✳ Code was ported from hawtio v1 to v2, and the method in which the links are specified has changed. This caused some broken links on the OSGi pages, for example the Bundles table and Packages table. This bug fix changes the links to the correct method in hawtio v2, which includes the relative path and navigation information. As a result, the broken links are not longer broken. ([BZ#1411330](#))
- ✳ The path for the OpenShift Container Platform 3.4 documentation links in the web console was incorrect. A redirect was added to the documentation site so the incorrect paths would resolve until the path could be fixed. This bug fix updates the documentation links in the web console to have the correct path. As a result, the documentation links go directly to the correct paths without needing the redirect. ([BZ#1414552](#))

## Metrics

- ✳ When authenticating users, Hawkular Metrics was not properly handling error responses back from the master for a subjectaccessreview. If the authentication token passed was invalid, the connection to Hawkular Metrics would stay open until a timeout. This bug fix ensures Hawkular Metrics now properly handles these error responses and closes the connection. As a result, if a user passes an invalid token, their connection now closes properly and no longer remain open until a timeout. ([BZ#1410899](#))
- ✳ In some rare circumstances, Hawkular Metrics would start to consume too much CPU resources. This could cause the Hawkular Metrics pod to stop responding and cause metrics to no longer be collected. The root of the problem appears to be with a Netty library used by the Cassandra driver. This bug fix configures the pod to use a different mechanism other than Netty. As a result, the Hawkular Metrics pod should no longer fail in this manner due to high CPU usage. ([BZ#1411427](#))
- ✳ When using Hawkular Metrics with AutoResolve triggers in a clustered environment, a trigger defined with **AUTORESOLVE** conditions fired correctly in **FIRING** mode but did not fire in **AUTORESOLVE** mode. This bug fix updates Hawkular Metrics to ensure the triggers fire correctly



in both modes. ([BZ#1415833](#))

## Networking

- ✳ In OpenShift SDN, the IP addresses for a node were not sorted. When the first IP was chosen, it may be different from the last one used, so the IP address appeared to have changed. OpenShift Container Platform would then update the node-to-IP mapping, causing problems with everything moving from one interface to another. This bug fix updates OpenShift SDN to sort the addresses, and as a result the traffic flows correctly and the addresses do not change. ([BZ#1410128](#))
- ✳ When the admission controller that adds security contexts is disabled, the node can crash. The node crashed trying to process a security context that was not present. This bug fix ensures that the pointer is checked to be defined before dereferencing it. As a result, the node no longer crashes. ([BZ#1412087](#))

## Routing

- ✳ Previously, the router would not reload HAProxy after the initial sync if the last item of the initial list of any of the watched resources did not reach the router to trigger the commit. This could be caused by a route being rejected for any reason, for example specifying a host claimed by another namespace. The router could be left in its initial state (without any routes configured) until another commit-triggering event occurred, such as a watch event. This bug fix updates the router to always reload after initial sync. As a result, routes are available after the initial sync. ([BZ#1383663](#))
- ✳ This release adds an option to allow HAProxy to expect incoming connections on port 80 or port 443 to use the **PROXY** protocol. The source IP address can pass through a load balancer if the load balancer supports the protocol, for example Amazon ELB. As a result, if the **ROUTER\_USE\_PROXY\_PROTOCOL** environment variable is set to **true** or **TRUE**, HAProxy now expects incoming connections to use the **PROXY** protocol. ([BZ#1410156](#))

## Storage

- ✳ The **ceph-common** client tools were missing from the containerized node image. This prevented containerized environments from mounting Ceph volumes. This bug fix adds the **ceph-common** package, enabling containerized environments to mount Ceph volumes. ([BZ#1411244](#))

## Upgrade

- ✳ An error in the **atomic-openshift-docker-excluder** package led to packages being removed from the exclusion list when upgraded. This bug fix ensures that the proper packages are excluded from yum operations. ([BZ#1404193](#))

### 2.8.3. RHBA-2017:0268 - OpenShift Container Platform 3.4.1.5 Bug Fix Update

OpenShift Container Platform release 3.4.1.5 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:0268](#) advisory. The list of container images included in the update are documented in the [RHBA-2017:0267](#) advisory.

#### 2.8.3.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.3 or 3.4 cluster to this latest release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

This release delivers the migration tool mentioned in the above [OpenShift Container Platform 3.4.0.40 release notes](#). See the following Knowledgebase Solution for instructions on running the script:

<https://access.redhat.com/solutions/2887651>

## 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 OPENSIFT 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 vs 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 **OPENSIFT\_MYSQL\_DB\_HOST** and **OPENSIFT\_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 PaaS 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 VS 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 VS 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 VS PROJECT

A [project](#) is essentially a v2 domain.

## CHAPTER 5. REVISION HISTORY: RELEASE NOTES

### 5.1. THU FEB 09 2017

Affected Topic	Description of Change
<a href="#">OpenShift Container Platform 3.4 Release Notes</a>	Added release notes for <a href="#">RHBA-2017:0268 - OpenShift Container Platform 3.4.1.5 Bug Fix Update</a> .
	Added an etcd performance issue to <a href="#">Known Issues</a> .

### 5.2. TUE JAN 31 2017

Affected Topic	Description of Change
<a href="#">OpenShift Container Platform 3.4 Release Notes</a>	Added release notes for <a href="#">RHBA-2017:0218 - OpenShift Container Platform 3.4.1.2 Bug Fix Update</a> .

### 5.3. TUE JAN 24 2017

Affected Topic	Description of Change
<a href="#">OpenShift Container Platform 3.4 Release Notes</a>	Added release notes for <a href="#">RHBA-2017:0186 - OpenShift Container Platform 3.4.0.40 Bug Fix Update</a> .
	Added <a href="#">BZ#1415570</a> to <a href="#">Known Issues</a> .

### 5.4. WED JAN 18 2017

OpenShift Container Platform 3.4 initial release.

Affected Topic	Description of Change
<a href="#">OpenShift Container Platform 3.4 Release Notes</a>	Added release notes for initial release.