



OpenShift Enterprise

3.1

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

CHAPTER 1. OVERVIEW	3
CHAPTER 2. OPENSIFT ENTERPRISE 3.1 RELEASE NOTES	4
2.1. OVERVIEW	4
2.2. NEW FEATURES AND ENHANCEMENTS	4
2.3. BUG FIXES	7
2.4. TECHNOLOGY PREVIEW FEATURES	10
2.5. KNOWN ISSUES	10
2.6. ASYNCHRONOUS ERRATA UPDATES	10
2.6.1. OpenShift Enterprise 3.1.1	11
2.6.1.1. Enhancements	11
2.6.1.2. Technology Preview Features	13
2.6.1.3. Bug Fixes	13
2.6.1.4. Known Issues	15
2.6.2. OpenShift Enterprise 3.1.1.11	16
2.6.2.1. Upgrading	16
2.6.3. OpenShift Enterprise 3.1.1.11-2	16
2.6.3.1. Upgrading	16
2.6.4. OpenShift Enterprise 3.1.1.11-3	16
2.6.4.1. Upgrading	16
CHAPTER 3. XPAAS RELEASE NOTES	17
3.1. OVERVIEW	17
3.2. XPAAS IMAGE FOR RED HAT JBOSS EAP	17
3.3. XPAAS IMAGE FOR RED HAT JWS	17
3.4. XPAAS IMAGE FOR RED HAT JBOSS A-MQ	17
3.5. XPAAS IMAGE FOR RED HAT JBOSS FUSE (FUSE INTEGRATION SERVICES)	18
3.6. XPAAS IMAGE FOR RED HAT JBOSS BRMS (DECISION SERVER)	18
3.7. XPAAS IMAGE FOR RED HAT JBOSS DATA GRID	18
3.8. KNOWN ISSUES FOR XPAAS IMAGES	18
CHAPTER 4. COMPARING OPENSIFT ENTERPRISE 2 AND OPENSIFT ENTERPRISE 3	21
4.1. OVERVIEW	21
4.2. ARCHITECTURE CHANGES	21
4.3. APPLICATIONS	21
4.4. CARTRIDGES VS IMAGES	22
4.5. BROKER VS MASTER	23
4.6. DOMAIN VS PROJECT	23
CHAPTER 5. REVISION HISTORY: RELEASE NOTES	24
5.1. THU JUN 29 2017	24
5.2. THU MAY 18 2017	24
5.3. TUE APR 25 2017	24
5.4. THU MAR 17 2016	24
5.5. THU FEB 04 2016	24
5.6. THU JAN 28 2016	25
5.7. MON JAN 19 2016	25
5.8. THU NOV 19 2015	25

CHAPTER 1. OVERVIEW

The following release notes for OpenShift Enterprise 3.1 and xPaaS summarize all new features, major corrections from the previous version, and any known bugs upon general availability.

CHAPTER 2. OPENSIFT ENTERPRISE 3.1 RELEASE NOTES

2.1. OVERVIEW

OpenShift Enterprise by Red Hat 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 Enterprise 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 Enterprise provides a secure and scalable multi-tenant operating system for today's enterprise-class applications, while providing integrated application runtimes and libraries. OpenShift Enterprise 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. NEW FEATURES AND ENHANCEMENTS

OpenShift Enterprise version 3.1 is now available. Ensure that you follow the instructions on upgrading your OpenShift cluster properly, including steps specific to this release.



Important

For any release, always review [Installation and Configuration](#) for instructions on [upgrading your OpenShift cluster](#) properly, including any additional steps that may be required for a specific release.

For Administrators:

Service, Package, File, and Directory Names Changed

Previous Name	New Name
<code>openshift-master</code>	<code>atomic-openshift-master</code>
<code>openshift-node</code>	<code>atomic-openshift-node</code>
<code>/etc/openshift/</code>	<code>/etc/origin/</code>
<code>/var/lib/openshift/</code>	<code>/var/lib/origin/</code>
<code>/etc/sysconfig/openshift-master</code>	<code>/etc/sysconfig/atomic-openshift-master</code>

Previous Name	New Name
<i>/etc/sysconfig/openshift-node</i>	<i>/etc/sysconfig/atomic-openshift-node</i>

Docker Version Update Required

Docker version 1.8.2 is required. This contains the fix to let the */etc/groups* file use supplementary groups.

LDAP Synchronization

OpenShift now allows you to sync LDAP records with OpenShift, so that you can manage groups easily.

F5 Availability

You can now configure an F5 load-balancer for use with your OpenShift environment.

More Persistent Storage Options

Several persistent storage options are now available, such as Red Hat's GlusterFS and Ceph RBD, AWS, and Google Compute Engine. Also, NFS storage is now supplemented by iSCSI- and Fibre Channel-based volumes.

More Middleware Options

Several middleware services are now available, such as JBoss DataGrid, and JBoss BRMS, as well as a supported JBoss Developer Studio and Eclipse plug-in.

Job Controller Now Available

The job object type is now available, meaning that finite jobs can now be executed on the cluster.

Installer Updates

Multiple enhancements have been made to the Ansible-based installer. The installer can now:

- ✧ Perform container-based installations. (Fully supported starting in [OpenShift Enterprise 3.1.1](#))
- ✧ Install active-active, highly-available clusters.
- ✧ Uninstall existing OpenShift clusters.

Custom CA Certificates

You can now specify your own CA certificate during the install, so that application developers do not have to specify the OpenShift-generated CA to obtain secure connections.

DNS Service Name Change

The DNS name for service SRV discovery has changed. Services without search paths resulted in long load times to resolve DNS. The change reduces load times.

New Parameter Preventing Memory Overload

Excessive amounts of events being stored in etcd can lead to excessive memory growth. You can now set the **event-ttl** parameter in the master configuration file to a lower value (for example, **15m**) to prevent memory growth.

New Parameter for Port Destination

You can now specify the port to send routes to. Any services that are pointing to multiple ports should have the **spec.port.targetPort** parameter on the pod set to the desired port.

New Remote Access Command

The **oc rsync** command is now available, which can copy local directories into a remote pod.

Project Binding Command

Isolated projects can now be bound together using **oadm pod-network join-project**.

Host Configuration Validation Commands

New commands exist to validate master and node configuration files: **openshift ex validate master-config** and **openshift ex validate node-config**, respectively.

New Tag Deletion Command

You can now delete tags from an image stream using the **oc tag <tag_name> -d** command.

New Overcommit Abilities

You can now create containers that can specify compute resource requests and limits. Requests are used for scheduling your container and provide a minimum service guarantee. Limits constrain the amount of compute resource that may be consumed on your node.

CPU Limits Now Enforced Using CFS Quota by Default

If you wish to disable CFS quota enforcement, you may disable it by modifying your **node-config.yaml** file to specify a **kubeletArguments** stanza where **cpu-cfs-quota** is set to **false**.

For Developers:

v1beta3 no Longer Supported

Using **v1beta3** in configuration files is no longer supported:

- ✦ The **etcdStorageConfig.kubernetesStorageVersion** and **etcdStorageConfig.openShiftStorageVersion** values in the master configuration file must be **v1**.
- ✦ You may also need to change the **apiLevels** field and remove **v1beta3**.
- ✦ **v1beta3** is no longer supported as an endpoint. **/api/v1beta3** and **/osapi/v1beta3** are now disabled.

Web Console Enhancements

Multiple web console enhancements:

- ✧ Extended resource information is now available on the web console.
- ✧ The ability to trigger a deployment and rollback from the console has been added.
- ✧ Logs for builds and pods are now displayed on the web console in real time.
- ✧ When enabled, the web console will now display pod metrics.
- ✧ You can now connect to a container using a remote shell connection when in the **Builds** tab.

Aggregating Logging with the EFK Stack

Elasticsearch, Fluentd, and Kibana (together, known as the EFK stack) are now available for logging consumption.

Heapster Now Available

The Heapster interface and metric datamodel can now be used with OpenShift.

Jenkins Is Now Available

- ✧ A Jenkins image is now available for deployment on OpenShift.
- ✧ Integration between Jenkins masters and Jenkins slaves running on OpenShift has improved.

`oc build-logs` Is Now Deprecated

The `oc build-logs <build_name>` command is now deprecated and replaced by `oc logs build/<build_name>`.

`spec.rollingParams.updatePercent` Field Is Replaced

The `spec.rollingParams.updatePercent` field in deployment configurations has been replaced with `maxUnavailable` and `maxSurge`.

Images Now Editable

Images can be edited to set fields such as **labels** or **annotations**.

2.3. BUG FIXES

[BZ#1264836](#)

Previously, the upgrade script used an incorrect image to upgrade the HAProxy router. The script now uses the right image.

[BZ#1264765](#)

Previously, an upgrade would fail when a defined image stream or template did not exist. Now, the installation utility skips the incorrectly defined image stream or template and continues with the upgrade.

[BZ#1274134](#)

When using the `oc new-app` command with the `--insecure-registry` option, it would not set if the Docker daemon was not running. This issue has been fixed.

BZ#1273975

Using the **oc edit** command on Windows machines displayed errors with wrapping and file changes. These issues have been fixed.

BZ#1268891

Previously, creating pods from the same image in the same service and deployment were not grouped into another service. Now, pods created with the same image run in the same service and deployment, grouped together.

BZ#1267559

Previously, using the **oc export** command could produce an error, and the export would fail. This issue has been fixed.

BZ#1266981

The recycler would previously fail if hidden files or directories would be present. This issue has been fixed.

BZ#1268484

Previously, when viewing a build to completion on the web console after deleting and recreating the same build, no build spinner would show. This issue has been fixed.

BZ#1269070

You can now use custom self-signed certificates for the web console for specific host names.

BZ#1264764

Previously, the installation utility did not have an option to configure the deployment type. Now, you can run the **--deployment-type** option with the installation utility to select a type, otherwise the type set in the installation utility will be set.

BZ#1273843

There was an issue with the **pip** command not being available in the newest OpenShift release. This issue has been fixed.

BZ#1274601

Previously, using the **oc exec** command was only available to be used on privileged containers. Now, users with permissions to create pods can use the **oc exec** command to SSH into privileged containers.

BZ#1267670

There was an issue with using the **iptables** command with the **-w** option to make the **iptables** command wait to acquire the **xtables** lock, causing some SDN initializations to fail. This issue has been fixed.

BZ#1272201

When installing a clustered etcd and defining variables for IP and etcd interfaces when using two network interfaces, the certificate would be populated with only the first network, instead of whichever network was desired. The issue has now been fixed.

BZ#1269256

Using the **GET fieldSelector** would return a 500 BadRequest error. This issue has been fixed.

BZ#1268000

Previously, creating an application from a image stream could result in two builds being initiated. This was caused by the wrong image stream tag being used by the build process. The issue has been fixed.

BZ#1267231

The **ose-ha-proxy** router image was missing the **X-Forwarded** headers, causing the Jenkins application to redirect to HTTP instead of HTTPS. The issue has been fixed.

BZ#1276548

Previously, an error was present where the HAProxy router did not expose statistics, even if the port was specified. The issue has been fixed.

BZ#1275388

Previously, some node hosts would not talk to the SDN due to routing table differences. A **1br0** entry was causing traffic to be routed incorrectly. The issue has been fixed.

BZ#1265187

When persistent volume claims (PVC) were created from a template, sometimes the same volume would be mounted to multiple PVCs. At the same time, the volume would show that only one PVC was being used. The issue has been fixed.

BZ#1279308

Previously, using a etcd storage location other than the default, as defined in the master configuration file, would result in an upgrade fail at the "generate etcd backup" stage. This issue has now been fixed.

BZ#1276599

Basic authentication passwords can now contain colons.

BZ#1279744

Previously, giving **EmptyDir** volumes a different default permission setting and group ownership could affect deploying the **postgresql-92-rhel7** image. The issue has been fixed.

BZ#1276395

Previously, an error could occur when trying to perform an HA install using Ansible, due to a problem with SRC files. The issue has been fixed.

BZ#1267733

When installing a etcd cluster with hosts with different network interfaces, the install would fail. The issue has been fixed.

BZ#1274239

Previously, when changing the default project region from **infra** to **primary**, old route and registry pods are stuck in the terminating stage and could not be deleted, meaning that new route and registry pods could not be deployed. The issue has been fixed.

BZ#1278648

If, when upgrading to OpenShift Enterprise 3.1, the OpenShift Enterprise repository was not set, a Python error would occur. This issue has been fixed.

2.4. 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 features are in Technology Preview:

- ✳ Binary builds and the Dockerfile source type for builds. (Fully supported starting in [OpenShift Enterprise 3.1.1](#))
- ✳ Pod autoscaling, using the **HorizontalPodAutoscaler** object. OpenShift compares pod CPU usage as a percentage of requested CPU and scales according to an indicated threshold. (Fully supported starting in [OpenShift Enterprise 3.1.1](#))
- ✳ Support for OpenShift Enterprise running on RHEL Atomic Host. (Fully supported starting in [OpenShift Enterprise 3.1.1](#))
- ✳ Containerized installations, meaning all OpenShift Enterprise components running in containers. (Fully supported starting in [OpenShift Enterprise 3.1.1](#))

2.5. KNOWN ISSUES

- ✳ When pushing to an internal registry when multiple registries share the same NFS volume, there is a chance the push will fail. [A workaround](#) has been suggested.
- ✳ When creating a build, in the event where there are not enough resources (possibly due to quota), the build will be pending indefinitely. As a workaround, free up resources, cancel the build, then start a new build.

2.6. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Enterprise 3.1 are released as asynchronous errata through the Red Hat Network. All OpenShift Enterprise 3.1 errata is [available on the Red Hat Customer Portal](#). See the [OpenShift Enterprise 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 Enterprise entitlements for OpenShift Enterprise errata notification emails to generate.

The following sections provide notes on enhancements and bug fixes for each asynchronous errata release of OpenShift Enterprise 3.1.



Important

For any release, always review the instructions on [upgrading your OpenShift cluster](#) properly.

2.6.1. OpenShift Enterprise 3.1.1

OpenShift Enterprise release 3.1.1 ([RHSA-2016:0070](#)) is now available. Ensure that you follow the instructions on [upgrading your OpenShift cluster](#) to this asynchronous release properly.

This release includes the following enhancements and bug fixes.

2.6.1.1. Enhancements

Containerized Installations Now Fully Supported

Installation of OpenShift Enterprise master and node components as containerized services, added as Technology Preview in OpenShift Enterprise 3.1.0, is now fully supported as an alternative to the standard RPM method. Both the [quick](#) and [advanced installation](#) methods support use of the containerized method. See [RPM vs Containerized](#) for more details on the differences when running as a containerized installation.

RHEL Atomic Host Now Fully Supported

Installing OpenShift Enterprise on Red Hat Enterprise Linux (RHEL) Atomic Host 7.1.6 or later, added as Technology Preview in OpenShift Enterprise 3.1.0, is now fully supported for running containerized OpenShift services. See [System Requirements](#) for more details.

Binary Builds and Dockerfile Sources Now Fully Supported

[Binary builds](#) and the [Dockerfile source type](#) for builds, added as Technology Preview in OpenShift Enterprise 3.1.0, are now fully supported.

Pod Autoscaling Now Fully Supported

[Pod autoscaling](#) using the **HorizontalPodAutoscaler** object, added as Technology Preview in OpenShift Enterprise 3.1.0, is now fully supported. OpenShift compares pod CPU usage as a percentage of requested CPU and scales according to an indicated threshold.

Web Console

- » When creating an application from source in the web console, you can independently specify build environment variables and deployment environment variables on the creation page. Build environment variables created in this way also become available at runtime. ([BZ#1280216](#))
- » When creating an application from source in the web console, all container ports are now exposed on the creation page under "Routing". ([BZ#1247523](#))
- » Build trends are shown on the build configuration overview page.
- » Individual build configurations and deployment configurations can be deleted.

- ✳ Any object in the web console can be edited like **oc edit** with a direct YAML editor, for when you need to tweak rarely used fields.
- ✳ The experience around web console scaling has been improved with more information.
- ✳ Empty replication controllers are shown in the Overview when they are not part of a service.
- ✳ Users can dismiss web console alerts.

Command Line

- ✳ **oc status** now shows suggestions and warnings about conditions it detects in the current project.
- ✳ **oc start-build** now allows **--env** and **--build-loglevel** to be passed as arguments.
- ✳ **oc secret** now allows custom secret types to be created.
- ✳ Secrets can be created for Docker configuration files using the new **.docker/config.json** format with the following syntax:

```
$ oc secrets new <secret_name> .dockerconfigjson=  
[path/to/].docker/config.json
```

([BZ#1282426](#))

- ✳ **oc new-build** now supports the **--to** flag, which allows you to specify which image stream tag you want to push a build to. You can pass **--to-docker** to push to an external image registry. If you only want to test the build, pass **--no-output** which only ensures that the build passes.

Security

- ✳ The user name of the person requesting a new project is now available to parameterize the initial project template as the parameter **PROJECT_REQUESTING_USER**.
- ✳ When creating a new application from a Docker image, a warning occurs if the image does not specify a user that administrators may have disabled running as root inside of containers.
- ✳ Add a new role **system:image-pusher** that allows pushing images to the integrated registry.
- ✳ Deleting a cluster role from the command line now deletes all role bindings associated to that role unless you pass the **--cascade=false** option.

API Changes

- ✳ You can delete a tag using **DELETE /oapi/v1/namespaces/<namespace>/imagestreamtags/<stream>:<tag>**.
- ✳ It is no longer valid to set route TLS configuration without also specifying a termination type. A default has been set for the type to be **terminate** if the user provided TLS certificates.
- ✳ Docker builds can now be configured with custom **Dockerfile** paths.

Miscellaneous

- ✳ The integrated Docker registry has been updated to version 2.2.1.
- ✳ The LDAP group prune and sync commands have been promoted out of experimental and into **oadm groups**.
- ✳ More tests and configuration warnings have been added to **openshift ex diagnostics**.
- ✳ Builds are now updated with the Git commit used in a build after the build completes.
- ✳ Routers now support overriding the host value in a route at startup. You can start multiple routers and serve the same route over different wildcards (with different configurations). See the help text for **openshift-router**.

2.6.1.2. Technology Preview Features

The following features have entered into [Technology Preview](#):

- ✳ [Dynamic provisioning](#) of persistent storage volumes from Amazon EBS, Google Compute Disk, OpenStack Cinder storage providers.

2.6.1.3. Bug Fixes

BZ#1256869

Deleting users and groups cascades to delete their role bindings across the cluster.

BZ#1289603

In clustered etcd environments, user logins could fail with a 401 Unauthorized error due to stale reads from etcd. This bug fix updates OpenShift to wait for access tokens to propagate to all etcd cluster members before returning the token to the user.

BZ#1280497

OpenShift Enterprise now supports DWARF debugging.

BZ#1268478

Builds can now retrieve sources from Git when providing the repository with a user other than **git**.

BZ#1278232

When a build fails to start because of quota limits, if the quota is increased, the build is now handled correctly and starts.

BZ#1287943

When canceling a build within a few seconds of entering the running state, the build is now correctly marked "Cancelled" instead of "Failed".

BZ#1287414

The example syntax in the help text for **oc attach** has been fixed.

BZ#1284506

The man page for the **tuned-profiles-atomic-openshift-node** command was missing, and has now been restored.

BZ#1278630

An event is now created with an accompanying error message when a deployment cannot be created due to a quota limit.

BZ#1292621

The default templates for Jenkins, MySQL, MongoDB, and PostgreSQL incorrectly pointed to CentOS images instead of the correct RHEL-based image streams. These templates have been fixed.

BZ#1289965

An out of range panic issue has been fixed in the OpenShift SDN.

BZ#1277329

Previously, it was possible for core dumps to be generated after running OpenShift for several days. Several memory leaks have since been fixed to address this issue.

BZ#1254880

The Kubelet exposes statistics from cAdvisor securely using cluster permissions to view metrics, enabling secure communication for Heapster metric collection.

BZ#1293251

A bug was fixed in which service endpoints could not be accessed reliably by IP address between different nodes.

BZ#1277383

When the **ovs-multitenant** plug-in is enabled, creating and deleting an application could previously leave behind OVS rules and a veth pair on the OVS bridge. Errors could be seen when checking the OVS interface. This bug fix ensures that ports for the deleted applications are properly removed.

BZ#1290967

If a node was under heavy load, it was possible for the node host subnet to not get created properly during installation. This bug fix bumps the timeout wait from 10 to 30 seconds to avoid the issue.

BZ#1279925

Various improvements have been made to ensure that OpenShift SDN can be installed and started properly.

BZ#1282738

The MySQL image can now handle if **MYSQL_USER=root** is set. However, an error is produced if you set **MYSQL_USER=root** and also **MYSQL_PASSWORD** and **MYSQL_ROOT_PASSWORD** at the same time.

BZ#1283952

The default HAProxy "503" response lacked response headers, resulting in an invalid HTTP response. The response headers have been updated to fix this issue.

BZ#1290643

HAProxy's "Forwarded" header value is now RFC 7239 compliant.

BZ#1279744

The default strategies for cluster SCCs have been changed to **RunAsAny** for **FSGroup** and **SupplementalGroups**, to retain backwards compatible behavior.

BZ#1273739

When creating a PV and PVC for a Cinder volume, it was possible for pods to not be created successfully due to a "Cloud provider not initialized properly" error. This has been fixed by ensuring that the related OpenShift instance ID is properly cached and used for volume management.

2.6.1.4. Known Issues**BZ#1293578**

There was an issue with OpenShift Enterprise 3.1.1 where hosts with host names that resolved to IP addresses that were not local to the host would run into problems with liveness and readiness probes on newly-created HAProxy routers. This was resolved in [RHBA-2016:0293](#) by configuring the probes to use **localhost** as the hostname for pods with **hostPort** values.

If you created a router under the affected version, and your liveness or readiness probes unexpectedly fail for your router, then add **host: localhost**:

```
# oc edit dc/router
```

Apply the following changes:

```
spec:
  template:
    spec:
      containers:
      ...
      livenessProbe:
        httpGet:
          host: localhost 1
          path: /healthz
          port: 1936
          scheme: HTTP
        initialDelaySeconds: 10
        timeoutSeconds: 1
      ...
      readinessProbe:
        httpGet:
          host: localhost 2
          path: /healthz
          port: 1936
          scheme: HTTP
        timeoutSeconds: 1
```

1

Add **host**: **localhost** to your liveness probe.

2

Add **host**: **localhost** to your readiness probe.

2.6.2. OpenShift Enterprise 3.1.1.11

OpenShift Enterprise release 3.1.1.11 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:0989](#) advisory. The list of container images included in the update are documented in the [RHBA-2017:0990](#) advisory.

The container images in this release have been updated using the **rhel:7.3-74** base image, where applicable.

2.6.2.1. Upgrading

To upgrade an existing OpenShift Enterprise 3.0 or 3.1 cluster to the latest 3.1 release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

2.6.3. OpenShift Enterprise 3.1.1.11-2

OpenShift Enterprise release 3.1.1.11-2 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:1235](#) advisory. The list of container images included in the update are documented in the [RHBA-2017:1236](#) advisory.

2.6.3.1. Upgrading

To upgrade an existing OpenShift Enterprise 3.0 or 3.1 cluster to the latest 3.1 release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

2.6.4. OpenShift Enterprise 3.1.1.11-3

OpenShift Enterprise release 3.1.1.11-3 is now available. The list of packages and bug fixes included in the update are documented in the [RHBA-2017:1665](#) advisory.

2.6.4.1. Upgrading

To upgrade an existing OpenShift Enterprise 3.0 or 3.1 cluster to the latest 3.1 release, use the automated upgrade playbook. See [Performing Automated In-place Cluster Upgrades](#) for instructions.

CHAPTER 3. XPAAS RELEASE NOTES

3.1. OVERVIEW

Starting in OpenShift Enterprise 3.0, xPaaS images are provided for the following:

- ✦ Red Hat JBoss Enterprise Application Platform
- ✦ Red Hat JBoss Web Server
- ✦ Red Hat JBoss A-MQ

Starting in OpenShift Enterprise 3.1, xPaaS images are also provided for the following:

- ✦ Red Hat JBoss Fuse (Fuse Integration Services)
- ✦ Red Hat JBoss BRMS (Decision Server)
- ✦ Red Hat JBoss Data Grid

See enterprise.openshift.com/middleware-services for additional information.

3.2. XPAAS IMAGE FOR RED HAT JBOSS EAP

Red Hat JBoss EAP is available as a [containerized xPaaS image](#) that is designed for use with OpenShift Enterprise 3.0 and later.

However, there are [significant differences](#) in supported configurations and functionality in the JBoss EAP xPaaS image compared to the regular release of JBoss EAP. Documentation for other JBoss EAP functionality not specific to the JBoss EAP xPaaS image can be found in the [JBoss EAP documentation on the Red Hat Customer Portal](#).

3.3. XPAAS IMAGE FOR RED HAT JWS

The Apache Tomcat 7 and Apache Tomcat 8 components of Red Hat JBoss Web Server 3 are available as [containerized xPaaS images](#) that are designed for use with OpenShift Enterprise 3.0 and later.

However, there are [significant differences](#) in the functionality between the JBoss Web Server xPaaS images and the regular release of JBoss Web Server. Documentation for other JBoss Web Server functionality not specific to the JBoss Web Server xPaaS images can be found in the [JBoss Web Server documentation on the Red Hat Customer Portal](#).

3.4. XPAAS IMAGE FOR RED HAT JBOSS A-MQ

Red Hat JBoss A-MQ is available as a [containerized xPaaS image](#) that is designed for use with OpenShift Enterprise 3.0 and later. It allows developers to quickly deploy an A-MQ message broker in a hybrid cloud environment.

However, there are [significant differences](#) in supported configurations and functionality in the JBoss A-MQ image compared to the regular release of JBoss A-MQ. Documentation for other JBoss A-MQ functionality not specific to the JBoss A-MQ xPaaS image can be found in the [JBoss A-MQ documentation on the Red Hat Customer Portal](#).

3.5. XPAAS IMAGE FOR RED HAT JBOSS FUSE (FUSE INTEGRATION SERVICES)

Red Hat JBoss Fuse is available as a [containerized xPaaS image](#), known as Fuse Integration Services, that is designed for use with OpenShift Enterprise 3.1. It allows developers to quickly deploy applications in a hybrid cloud environment. In Fuse Integration Services, application runtime is dynamic.

However, there are [significant differences](#) in supported configurations and functionality in the Fuse Integration Services compared to the regular release of JBoss Fuse. Documentation for other JBoss Fuse functionality not specific to the Fuse Integration Services can be found in the [JBoss Fuse documentation on the Red Hat Customer Portal](#).

3.6. XPAAS IMAGE FOR RED HAT JBOSS BRMS (DECISION SERVER)

Red Hat JBoss BRMS is available as a [containerized xPaaS image](#), known as Decision Server, that is designed for use with OpenShift Enterprise 3.1 as an execution environment for business rules. Developers can quickly build, scale, and test applications deployed across hybrid environments.

However, there are [significant differences](#) in supported configurations and functionality in the Decision Server xPaaS image compared to the regular release of JBoss BRMS. Documentation for other JBoss BRMS functionality not specific to the Decision Server xPaaS image can be found in the [JBoss BRMS documentation on the Red Hat Customer Portal](#).

3.7. XPAAS IMAGE FOR RED HAT JBOSS DATA GRID

Red Hat JBoss Data Grid is available as a [containerized xPaaS image](#) that is designed for use with OpenShift Enterprise 3.1. This image provides an in-memory distributed database so that developers can quickly access large amounts of data in a hybrid environment.

However, there are [significant differences](#) in supported configurations and functionality in the JBoss Data Grid xPaaS image compared to the full, non-PaaS release of JBoss Data Grid. Documentation for other JBoss Data Grid functionality not specific to the JBoss Data Grid xPaaS image can be found in the [JBoss Data Grid documentation on the Red Hat Customer Portal](#).

3.8. KNOWN ISSUES FOR XPAAS IMAGES

The following are the current known issues along with any known workarounds:

JWS

- ✎ <https://issues.jboss.org/browse/CLOUD-57>: Tomcat's access log valve logs to file in container instead of stdout

Due to this issue, the logging data is not available for the central logging facility. To work around this issue, use the **oc exec** command to get the contents of the log file.

- ✎ <https://issues.jboss.org/browse/CLOUD-153>: **mvn clean** in JWS STI can fail

Cleaning up after a build in JWS STI is not possible, because the Maven command **mvn clean** fails. This is due to Maven not being able to build the object model during startup.

To work around this issue, add Red Hat and JBoss repositories into the **pom.xml** file of the application if the application uses dependencies from there.

- ✉ <https://issues.jboss.org/browse/CLOUD-156>: Datasource realm configuration is incorrect for JWS

It is not possible to do correct JNDI lookup for datasources in the current JWS image if an invalid combination of datasource and realm properties is defined. If a datasource is configured in the **context.xml** file and a realm in the **server.xml** file, then the **server.xml** file's **localDataSource** property should be set to **true**.

EAP

- ✉ <https://issues.jboss.org/browse/CLOUD-61>: JPA application fails to start when the database is not available

JPA applications fail to deploy in the EAP OpenShift Enterprise 3.0 image if an underlying database instance that the EAP instance relies on is not available at the start of the deployment. The EAP application tries to contact the database for initialization, but because it is not available, the server starts but the application fails to deploy.

There are no known workarounds available at this stage for this issue.

- ✉ <https://issues.jboss.org/browse/CLOUD-158>: Continuous HornetQ errors after scale down "Failed to create netty connection"

In the EAP image, an application not using messaging complains about messaging errors related to HornetQ when being scaled.

Since there are no configuration options to disable messaging to work around this issue, simply include the **standalone-openshift.xml** file within the source of the image and remove or alter the following lines related to messaging:

```
Line 18:
<!-- ##MESSAGING_EXTENSION## -->

Line 318:
<!-- ##MESSAGING_SUBSYSTEM## -->
```

- ✉ <https://issues.jboss.org/browse/CLOUD-161>: EAP pod serving requests before it joins cluster, some sessions reset after failure

In a distributed web application deployed on an EAP image, a new container starts serving requests before it joins the cluster.

There are no known workarounds available at this stage for this issue.

EAP and JWS

- ✉ <https://issues.jboss.org/browse/CLOUD-159>: Database pool configurations should contain validation SQL setting

In both the EAP and JWS images, when restarting a crashed database instance, the connection pools contain stale connections.

To work around this issue, restart all instances in case of a database failure.

Fuse Integration Services

- ✎ <https://issues.jboss.org/browse/OSFUSE-112>: karaf /deployments/karaf/bin/client CNFE org.apache.sshd.agent.SshAgent

Attempting to run the karaf client in the container to locally SSH to the karaf console fails.

Workaround: Adding both **shell** and **ssh** features make the client work. It will log the warning errors in the logs.

```
$ oc exec karaf-shell-1-bb9zu -- /deployments/karaf/bin/client  
osgi:list
```

These warnings are logged when trying to use the JBoss Fuse bin/client script to connect to the JBoss Fuse micro-container. This is an unusual case, since the container is supposed to contain only bundles and features required for a micro-service, and hence does not need to be managed extensively like a traditional JBoss Fuse install. Any changes made using commands in the remote shell will be temporary and not recorded in the micro-service's docker image.

- ✎ <https://issues.jboss.org/browse/OSFUSE-190>: cdi-camel-jetty S2I template incorrect default service name, breaking cdi-camel-http

The cdi-camel-http quickstart expects the cdi-camel-jetty service to be named **qs-cdi-camel-jetty**. In the cdi-camel-jetty template however, the service is named **s2i-qs-cdi-camel-jetty** instead by default. This causes the cdi-camel-http to output this error when both are deployed using the S2I with default values.

Workaround: Set the cdi-camel-jetty SERVICE_NAME template parameter to **qs-cdi-camel-jetty**.

- ✎ <https://issues.jboss.org/browse/OSFUSE-193>: karaf-camel-rest-sql template service name too long

oc process karaf-camel-rest-sql template fails with the following error: The Service "s2i-qs-karaf-camel-rest-sql" is invalid. SUREFIRE-859: metadata.name: invalid value 's2i-qs-karaf-camel-rest-sql', Details: must be a DNS 952 label (at most 24 characters, matching regex [a-z]([-a-z0-9]*[a-z0-9])?): e.g. "my-name" deploymentconfig "s2i-quickstart-karaf-camel-rest-sql" created

Workaround: Set SERVICE_NAME template parameter to **karaf-camel-rest-sql**.

- ✎ <https://issues.jboss.org/browse/OSFUSE-195>: karaf-camel-amq template should have parameter to configure A-MQ service name

The application template for A-MQ deployments uses a suffix for every transport type to distinguish. Hence there should be a configurable parameter for setting the service name as environment parameter **A_MQ_SERVICE_NAME**.

A-MQ

There are no known issues in the A-MQ image.

CHAPTER 4. COMPARING OPENSIFT ENTERPRISE 2 AND OPENSIFT ENTERPRISE 3

4.1. OVERVIEW

OpenShift version 3 (v3) is a very different product than OpenShift version 2 (v2). Many of the same terms are used, 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 JUN 29 2017

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Added release notes for RHBA-2017:1665 - OpenShift Enterprise 3.1.1.11-3 Bug Fix Update .

5.2. THU MAY 18 2017

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Added release notes for RHBA-2017:1235 - OpenShift Enterprise 3.1.1.11-2 Bug Fix Update .

5.3. TUE APR 25 2017

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Added release notes for RHBA-2017:0989 - OpenShift Enterprise 3.1.1.11 Bug Fix Update .

5.4. THU MAR 17 2016

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Changed a known issue to a fix regarding liveness and readiness probes.

5.5. THU FEB 04 2016

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Updated the OpenShift Enterprise 3.1.1 Enhancements section to more clearly identify the features that came out of Technology Preview.
	Updated the OpenShift Enterprise 3.1.1 Technology Preview Features section to add a link to the new Dynamically Provisioning Persistent Volumes topic.

5.6. THU JAN 28 2016

OpenShift Enterprise 3.1.1 release.

Affected Topic	Description of Change
OpenShift Enterprise 3.1 Release Notes	Added the Asynchronous Errata Updates section and included release notes for OpenShift Enterprise 3.1.1 , detailing the various enhancements, technology preview features, bug fixes, and known issues.

5.7. MON JAN 19 2016

Affected Topic	Description of Change
Comparing OpenShift Enterprise 2 and OpenShift Enterprise 3	New topic discussing changes in architecture, concepts, and terminology between OpenShift Enterprise 2 and OpenShift Enterprise 3.

5.8. THU NOV 19 2015

OpenShift Enterprise 3.1 release.