OpenShift Container Platform 3.7

Release Notes

Last Updated: 2019-04-10

Legal Notice

Abstract

Table of Contents

CHAPTER 1. OVERVIEW

The following release notes for OpenShift Container Platform 3.7 summarize all new features, major corrections from the previous version, and any known bugs upon general availability.

1.1. VERSIONING POLICY

OpenShift Container Platform provides strict backwards compatibility guarantees for all supported APIs, excluding alpha APIs (which may be changed without notice) and beta APIs (which may occasionally be changed in a non-backwards compatible manner).

The OpenShift Container Platform version must match between master and node hosts, excluding temporary mismatches during cluster upgrades. For example, in a 3.7 cluster, all masters must be 3.7 and all nodes must be 3.7. However, OpenShift Container Platform will continue to support older **oc** clients against newer servers. For example, a 3.4 **oc** will work against 3.3, 3.4, and 3.5 servers.

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

Table 1.1. Compatibility Matrix

	X.Y (**oc** Client)	**X.Y+N** [a] (**oc** Client)
X.Y (Server)	❶	❸
X.Y+N [a] (Server)	❷	❶

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

❶ Fully compatible.

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

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

CHAPTER 2. OPENSHIFT CONTAINER PLATFORM 3.7 RELEASE NOTES

2.1. OVERVIEW

Red Hat OpenShift Container Platform provides developers and IT organizations with a cloud application platform for deploying new applications on secure, scalable resources with minimal configuration and management overhead. OpenShift Container Platform supports a wide selection of programming languages and frameworks, such as Java, Ruby, and PHP.

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

2.2. ABOUT THIS RELEASE

Red Hat OpenShift Container Platform version 3.7 (RHSA-2017:3188) is now available. This release is based on OpenShift Origin 3.7. New features, changes, bug fixes, and known issues that pertain to OpenShift Container Platform 3.7 are included in this topic.

OpenShift Container Platform 3.7 is supported on RHEL 7.3, 7.4.2, 7.5, and Atomic Host 7.4.2 and newer with the latest packages from Extras, including Docker 1.12.

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

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

To upgrade to this release from a previous version, see the Upgrading 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 Upstream

Many core features Google announced in June for Kubernetes 1.7 were the result of OpenShift engineering. Red Hat continues to influence the product in the areas of storage, networking, resource management, authentication and authorization, multi-tenancy, security, service deployments, templating, and controller functionality.

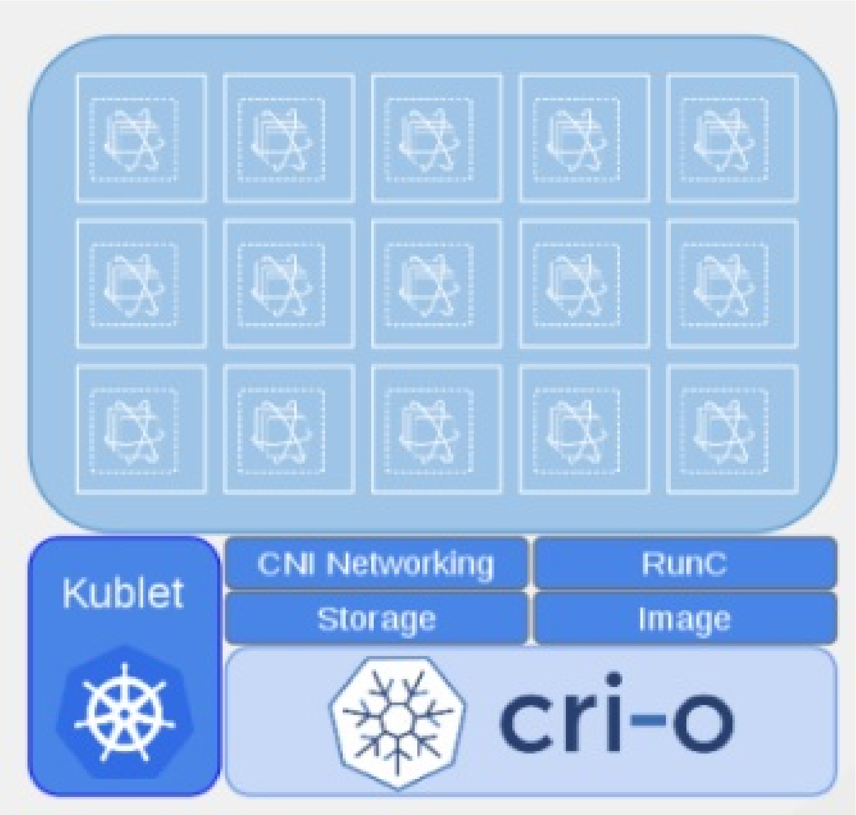
2.3.1.2. CRI-O (Technology Preview)

This feature is currently in Technology Preview and not for production workloads. CRI-O with builds will not yet work.

CRI-O v1.0 is a lightweight, native Kubernetes container runtime interface. By design, it provides only the runtime capabilities needed by the kubelet. CRI-O is designed to be part of Kubernetes and evolve in lock-step with the platform.

CRI-O brings:

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



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

```
openshift_use_crio=true
```

This setting pulls the **openshift3/cri-o** system container image from the Red Hat Registry by default. If you want to use an alternative CRI-O system container image from another registry, you can also override the default using the following variable:

```
openshift_crio_systemcontainer_image_override=<registry>/<repo>/<image>:
<tag>
```

NOTE

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

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

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

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

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

2.3.1.3. Cluster-wide Tolerations and Per-namespace Tolerations to Control Pod Placement

In a multi-tenant environment, you want to leverage administration controllers to help define rules that can help govern a cluster, should a tenant not set a toleration for placement.

The following is offered to administrators where the namespace setting will override the cluster setting:

- Cluster-wide and per-namespace default toleration for pods.
- Cluster-wide and per-namespace white-listing of toleration for pods.

Cluster-wide Off Example

```
admissionConfig:
  pluginConfig:
    PodTolerationRestriction:
      configuration:
        kind: DefaultAdmissionConfig
        apiVersion: v1
        disable: true
```

Cluster-wide On Example

```
admissionConfig:
  pluginConfig:
    PodTolerationRestriction:
      configuration:
        apiVersion: podtolerationrestriction.admission.k8s.io/v1alpha1
        kind: Configuration
        default:
         - key: key3
           value: value3
        whitelist:
         - key: key1
           value: value1
         - key: key3
           value: value3
```

Namespace-specific Example

```
apiVersion: v1
kind: Namespace
metadata:
  annotations:
    openshift.io/description: ""
    openshift.io/display-name: ""
    openshift.io/sa.scc.mcs: s0:c8,c7
    openshift.io/sa.scc.supplemental-groups: 1000070000/10000
    openshift.io/sa.scc.uid-range: 1000070000/10000
    scheduler.alpha.kubernetes.io/defaultTolerations: '[ { "key": "key1",
"value":"value1" }]'
    scheduler.alpha.kubernetes.io/tolerationsWhitelist: '[ { "key":
"key1", "value":
      "value1" }, { "key": "key2", "value": "value2" } ]'
  generateName: dma-
spec:
  finalizers:
  - openshift.io/origin
  - kubernetes
```

2.3.2. Security

2.3.2.1. Documented Private and Public Key Configurations and Crypto Levels

While OpenShift Container Platform is a secured by default implementation of Kubernetes, there is now documentation on what security protocols and ciphers are used.

OpenShift Container Platform leverages Transport Layer Security (TLS) cipher suites, JSON Web Algorithms (JWA) crypto algorithms, and offers external libraries such as The Generic Security Service Application Program Interface (GSSAPI) and libgpgme.

Private and public key configurations and Crypto levels are now documented for OpenShift Container Platform.

2.3.2.2. Node Authorizer and Node Restriction Admission Plug-in

Pods can no longer try to gain information from secrets, configuration maps, PV, PVC, or API objects from other nodes.

Node authorizer governs what APIs a kubelet can perform. Spanning read-, write-, and auth-related operations. In order for the admission controller to know the identity of the node to enforce the rules, nodes are provisioned with credentials that identify them with the user name **`system:node:<nodename>`** and group **`system:nodes`**.

These enforcements are in place by default on all new installations of OpenShift Container Platform 3.7. For upgrades from OpenShift Container Platform 3.6, they are not in place due to the **`system:nodes`** RBAC being granted from OCP 3.6. To turn the enforcements on, run:

```
# oc adm policy remove-cluster-role-from-group system:node system:nodes
```

2.3.2.3. Advanced Auditing

With Advanced Auditing, administrators are now exposed to more information from the API call within the audit trail. This provides a deeper traceability of what is occurring across the cluster. We also capture all login events at the default logging level and modifications to role binds and SCC.

OpenShift Container Platform now has an audit **`policyFile`** or **`policyConfiguration`** where administrators can filter in on what they want to capture.

See Advanced Audit for more information.

2.3.2.4. Complete Upstreaming of RBAC, Then Downstreaming it Back into OpenShift

The rolebinding and RBAC experience is now the same across all Kubernetes distributions.

Administrators do not have to do anything for this migration to occur. The upgrade process to OpenShift Container Platform 3.7 offers a seamless experience. Now, the user experience is consistent with upstream.

A role can be defined within a namespace with a **`Role`**, or cluster-wide with a **`ClusterRole`**.

A **`RoleBinding`** or **`ClusterRoleBinding`** binds a role to subjects. Subjects can be groups, users, or service accounts. A role binding grants the permissions defined in a role.

2.3.2.5. Issue Longer-lived API Tokens to OAuth Clients

Administrators now have the ability to set different token timeouts for the different ways users connect to OpenShift Container Platform (for example, via the **`oc`** command line, from a GitHub authentication, or from the web console).

Administrators can edit **`oauthclients`** and set the **`accessTokenMaxAgeSeconds`** to a time value in seconds that meets their needs.

There are three possible OAuth client types:

1. **`openshift-web-console`** - The client used to request tokens for the OpenShift web console.
2. **`openshift-browser-client`** - The client used to request tokens at ***/oauth/token/request*** with a user-agent that can handle interactive logins, such as using Auth from GitHub, Google Authenticator, and so on.
3. **`openshift-challenging-client`** - The client used to request tokens with a user-agent that can handle WWW-Authenticate challenges, such as the **`oc`** command line.

   - When **`accessTokenMaxAgeSeconds`** is set to **`0`**, tokens do not expire.
   - When left blank, OpenShift Container Platform uses the definition in **`master-config`**.
   - Edit the client of interest via:

     ```
     # oc edit oauthclients openshift-browser-client
     ```

   - Set **`accessTokenMaxAgeSeconds`** to **`600`**.
   - Check the setting via:

     ```
     # oc get oauthaccesstoken
     ```

See Other API Objects for more information.

2.3.2.6. Security Context Constraints Now Supports flexVolume

flexVolumes allow users to integrate with new APIs easily by being able to mount in the items needed for integration. For example, the ability to bind mount in certain files without overwriting whole directories to integrate with Kerberos.

Administrators are now able to grant access to users to use specific flexVolume driver names. Previously, the only way administrators could restrict flexVolumes was by setting them as **on** or **off**.

2.3.3. Storage

2.3.3.1. Local Storage Persistent Volumes (Technology Preview)

Local storage persistent volumes is a feature currently in Technology Preview and not for production workloads.

Local persistent volumes (PVs) now offer the ability to allow tenants to request storage that is local to a node through the regular persistent volume claim (PVC) process without needing to know the node. Local storage is commonly used in data store applications.

The administrator needs to create the local storage on the nodes, mount them under directories, and then manually create the persistent volume (PV). Alternatively, they can use an external provisioner and feed it the node configuration via **configMaps**.

Example persistent volume named **example-local-pv** that some tenants can now claim:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: example-local-pv
  annotations:
    "volume.alpha.kubernetes.io/node-affinity": '{
      "requiredDuringSchedulingIgnoredDuringExecution": {
        "nodeSelectorTerms": [
          { "matchExpressions": [
            { "key": "kubernetes.io/hostname",
              "operator": "In",
              "values": ["my-node"]
            }
          ]}
         ]}
        }'
spec:
  capacity:
    storage: 5Gi
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: local-storage
  local:
    path: /mnt/disks/vol1
```

See Configuring for Local Volume and Persistent Storage Using Local Volume for more information.

2.3.3.2. Tenant-driven Storage Snapshotting (Technology Preview)

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

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

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

1. The administrator runs an external provisioner for the cluster. These are images from the Red Hat Container Catalog.

2. The tenant made a PVC and owns a PV from one of the supported storage solutions.The administrator must create a new **`StorageClass`** in the cluster with:

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

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

```
$ oc create -f snapshot.yaml

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

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

```
$ oc create -f restore.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: snapshot-pv-provisioning-demo
  annotations:
    snapshot.alpha.kubernetes.io/snapshot: snapshot-demo
spec:
  storageClassName: snapshot-promoter
```

2.3.3.3. Storage Classes Get Zones

Public clouds are particular about not allowing storage to cross zones or regions, so tenants need an ability at times to specify a particular zone.

In OpenShift Container Platform 3.7, administrators can now leverage a zone's definition within the **StorageClass**:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: slow
provisioner: kubernetes.io/<provisioner>
parameters:
  type: pd-standard
  zones: zone1,zone2
```

See Dynamic Provisioning and Creating Storage Classes for more information.

2.3.3.4. Increased Persistent Volume Density Support by CNS

Container-native storage (CNS) on OpenShift Container Platform 3.7 now supports much higher persistent volume density (three times more) to support a large number of applications at scale. This is due to the introduction of brick-multiplexing support in GlusterFS.

Over 1,000 volumes in a 3-node cluster with 32 GB of RAM per node available to GlusterFS has been successfully tested. Also, 300 Block PVs are supported now on 3-node CNS.

2.3.3.5. CNS Multi-protocol (File, Block, and S3) Support for OpenShift

Container-native storage (CNS) is now extended support iSCSI and S3 back end for OpenShift Container Platform. Heketi is enhanced to support persistent volume (PV) expansion, volume option, and HA.

Block device-based RWO implementation is added to CNS to improve the performance of ElasticSearch, PostgreSQL, and so on. With OpenShift Container Platform 3.7, Elastic and Cassandra are fully supported.

2.3.3.6. CNS Full Support for Infrastructure Services

Container-native storage (CNS) now fully supports all OpenShift Container Platform infrastructure services: registry, logging, and metrics.

OpenShift Container Platform logging (with Elasticsearch) and OpenShift Container Platform metrics (with Cassandra) are fully supported on persistent volumes backed by CNS/CRS iSCSI block storage.

The OpenShift Container Platform registry is hosted on CNS/CRS by RWX persistent volumes, providing high availability and redundancy through Gluster architecture.

Logging and metrics were tested at scale with 1000+ pods.

2.3.3.7. Automated Container Native Storage Deployment with OpenShift Advanced Installation

OpenShift Container Platform 3.7 now includes an integrated and simplified installation of container-native storage (CNS) through the advanced installer. The advanced installer is enhanced for automated and integrated support for deployment of CNS including block provisioner, S3 provisioner, and files for correctly configured out-of-the-box OpenShift Container Platform and CNS. The CNS storage device details are added to the installer's inventory file. The installer manages configuration and deployment of CNS, its dynamic provisioners, and other pertinent details.

2.3.3.8. Official FlexVolume Support for Non-storage Use Cases

There is now a supported interface to allow you to bind and mount in content from a running pod. FlexVolume is a script interface that runs on the kubelet and offers five main functions to help you mount in content such as device drivers, secrets, and certificates as bind mounts to the container from the host:

- `init` - Initialize the volume driver.
- `attach` - Attach the volume to the host.
- `mount` - Mount the volume on the host. This is the part that makes the volume available to the host to mount it in ***/var/lib/kubelet***.
- `unmount` - Unmount the volume.
- `detach` - Detach the volume from the host.

2.3.4. Scale

2.3.4.1. Cluster Limits

Updated guidance around Cluster Limits for OpenShift Container Platform 3.7 is now available.

2.3.4.2. Updated Tuned Profile Hierarchy

The Tuned Profile Hierarchy is updated as of 3.7.

2.3.4.3. Cluster Loader

Guidance regarding use of Cluster Loader is now available with the release of OpenShift Container Platform 3.7. Cluster Loader is a tool that deploys large numbers of various objects to a cluster, which creates user-defined cluster objects. Build, configure, and run Cluster Loader to measure performance metrics of your OpenShift Container Platform deployment at various cluster states.

2.3.4.4. Guidance on Overlay Graph Driver with SELinux

In OpenShift Container Platform 3.7, guidance about the benefits of using the Overlay Graph Driver with SELinux is now available.

2.3.4.5. Providing Storage to an etcd Node Using PCI Passthrough with OpenStack

Guidance on Providing Storage to an etcd Node Using PCI Passthrough with OpenStack is now available.

2.3.5. Networking

2.3.5.1. Network Policy

Network Policy is now fully supported in OpenShift Container Platform 3.7.

Network Policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints. It provides fine-grained network namespace isolation using labels and port specifications.

NOTE

Network Policy is available only if you use the **`ovs-networkpolicy`** network plugin.

For example, the **allow-to-red** policy specifies all red pods in the same namespace as the NetworkPolicy object allow traffic from any pods in any namespace.

Policy applied to project

```
kind: NetworkPolicy
apiVersion: extensions/v1beta1
metadata:
  name: allow-to-red
spec:
  podSelector:
    matchLabels:
      type: red
  ingress:
  - {}
```

See Managing Networking for more information.

2.3.5.2. Cluster IP Range Now More Flexible

Cluster IP ranges are now more flexible by allowing multiple subnets for hosts. This provides the capability to allocate multiple, smaller IP address ranges for the cluster. This makes it easier to migrate from one allocated IP range to another.

There are multiple comma-delimited CIDRs in the configuration file. Each node is allocated only a single subnet from within any of the available ranges. You can not allocate different-sized host subnets, or use this to change the host subnet size. The **`clusterNetworkCIDRs`** can be different sizes, but must be equal to or larger than the host subnet size. It is not allowed to have some nodes use subnets that are not part of the **`clusterNetworkCIDRs`**. Nodes can allocate different-sized subnets by setting different **`hostSubnetLength`** values.

In regard to migration or edits, networks can be added to the list, CIDRs in the list may be re-ordered, and a CIDR can be removed from the list when there are no nodes that have an SDN allocation from that CIDR.

Example:

```
networkConfig:
  clusterNetworkCIDR: 10.128.0.0/24
  clusterNetworks:
  - cidr: 11.128.0.0/24
    hostSubnetLength: 6
  - cidr: 12.128.0.0/24
    hostSubnetLength: 6
  - cidr: 13.128.0.0/24
    hostSubnetLength: 4
  externalIPNetworkCIDRs:
  - 0.0.0.0/0
  hostSubnetLength: 6
```

2.3.5.3. Routes Allowed to Set Cookie Names for Session Stickiness

The HAProxy router can look for a cookie in a client request. Based on that cookie name and value, always route requests that have that cookie to the same pod instead of relying upon the client source IP, which can be obscured by an F5 doing load balancing.

A cookie with a unique name is used to handle session persistence.

1. Set a per-route configuration to set the cookie name used for the session.
2. Add an **env** to set a router-wide default.
3. Ensure that the cookie is set and honored by the router to control access.

Example scenario:

1. Set a default cookie name for the HAProxy router:

   ```
   $ oc env dc/router ROUTER_COOKIE_NAME=default-cookie
   ```

2. Log in as a normal user and create the project/pod/svc/route:

   ```
   $ oc login user1
   $ oc new-project project1
   $ oc create -f https://example.com/myhttpd.json
   $ oc create -f https://example.com/service_unsecure.json
   $ oc expose service service-unsecure
   ```

3. Access the route:

   ```
   $ curl $route -v
   ```

   The HTTP response will contain the cookie name. For example:

   ```
   Set-Cookie: default_cookie=[a-z0-9]+
   ```

4. Modify the cookie name using route annotation:

   ```
   $ oc annotate route service-unsecure
   router.openshift.io/cookie_name="route-cookie"
   ```

5. Re-access the route:

   ```
   $ curl $route -v
   ```

   The HTTP response will contain the new cookie name:

   ```
   Set-Cookie: route-cookie=[a-z0-9]+
   ```

See Route-specific Annotations for more information.

2.3.5.4. HSTS Policy Support

HTTP Strict Transport Security (HSTS) ensures all communication between the server and client is encrypted and that all sent and received responses are delivered to and received from the authenticated server.

An HSTS policy is provided to the client via an HTTPS header (HSTS headers over HTTP are ignored) using an **`haproxy.router.openshift.io/hsts_header`** annotation to the route. When the Strict-Transport-Security response in the header is received by a client, it observes the policy until it is updated by another response from the host, or it times-out (**`max-age=0`**).

Example using reencrypt route:

1. Create the pod/svc/route:

   ```
   $ oc create -f https://example.com/test.yaml
   ```

2. Set the Strict-Transport-Security header:

   ```
   $ oc annotate route serving-cert
   haproxy.router.openshift.io/hsts_header="max-
   age=300;includeSubDomains;preload"
   ```

3. Access the route using **`https`**:

   ```
   $ curl --head https://$route -k

       ...
       Strict-Transport-Security: max-age=300;includeSubDomains;preload
       ...
   ```

2.3.5.5. Enabling Static IPs for External Project Traffic (Technology Preview)

As a cluster administrator, you can assign specific, static IP addresses to projects, so that traffic is externally easily recognizable. This is different from the default egress router, which is used to send traffic to specific destinations.

Recognizable IP traffic increases cluster security by ensuring the origin is visible. Once enabled, all outgoing external connections from the specified project will share the same, fixed source IP, meaning that any external resources can recognize the traffic.

Unlike the egress router, this is subject to **`EgressNetworkPolicy`** firewall rules.

See Managing Networking for more information.

2.3.5.6. Re-encrypt Routes are Now Supported for Exposing the OpenShift Container Platform Registry

You can gain external access to the OpenShift Container Platform registry using a re-encrypt route. This allows you to log in to the registry from outside the cluster using the route address and to tag and push images using the route host.

2.3.6. Master

2.3.6.1. Public Pull URL Provided for Images

A public pull URL is provided for images versus being able to know the internal in-cluster IP or DNS of the service.

A new API field for the image stream with the public URL of the image was added, and a public URL is configured in the ***master-config.yaml*** file. The web console will understand this new field and generate the public pull specifications automatically to users (so users can just copy and paste the pull URL).

Example:

1. Check the **`internalRegistryHostname`** setting in the ***master-config.yaml*** file:

    ```
    ...
    imagePolicyConfig:
      internalRegistryHostname: docker-registry.default.svc:5000
    ...
    ```

2. Delete the **`OPENSHIFT_DEFAULT_REGISTRY`** variable in both:

    ```
    /etc/sysconfig/atomic-openshift-master-api
    /etc/sysconfig/atomic-openshift-master-controllers
    ```

3. Start a build and check the push URL. It should push the new build image with **`internalRegistryHostname`** to the **`docker-registry`**.

2.3.6.2. Custom Resource Definitions

A *resource* is an endpoint in the Kubernetes API that stores a collection of API objects of a certain kind (for example, pod objects). A *custom resource definition* is a built-in API that enables the ability to plug in your own custom, managed object and application as if it were native to Kubernetes. Therefore, you can leverage Kubernetes cluster management, RBAC and authentication services, PI services, CLI, security, and so on, without having to know Kubernetes internals or modifying Kubernetes itself in any way.

Custom Resource Definitions (CRD) deprecates Third Party Resources in Kubernetes 1.7.

How it works:

1. Define a CRD class (your custom objects) and register the new resource type. This defines how it fits into the hierarchy and how it will be referenced from the CLI and API.

2. Define a function to create a custom client, which is aware of the new resource schema.

3. Once completed, it can be accessed from the CLI. However, in order to build controllers or custom functionality, you need API access to the objects, and so you need to build a set of CRUD functions (library) to access the objects and the event-driven listener for controllers.

4. Create a client that:

    - Connects to the Kubernetes cluster.
    - Creates the new CRD (if it does not exist).
    - Creates a new custom client.
    - Creates a new test object using the client library.
    - Creates a controller that listens to events associated with new resources.

See Extending the Kubernetes API with Custom Resources for more information.

2.3.6.3. API Aggregation

There is now Kubernetes documentation on how API aggregation works in OpenShift Container Platform 3.7 and how other users can add third-party APIs:

- Set up an extension **`api-server`** to work with the aggregation layer
- Kubernetes aggregation layer

2.3.6.4. Master Prometheus Endpoint Coverage

Prometheus endpoint logic was added to upstream components so that monitoring and health indicators can be added around deployment configurations.

2.3.7. Installation

2.3.7.1. Migrate etcd Before OpenShift Container Platform 3.7 Upgrade

Starting in OpenShift Container Platform 3.7, the use of the etcd3 v3 data model is required.

OpenShift Container Platform gains performance improvements with the v3 data model. In order to upgrade the data model, an embedded etcd configuration option in no longer allowed. Embedded is not co-located. Migration scripts will convert the v3 data model and allow you to move an embedded etcd to an external etcd either on the same host or a different host than the masters. In addition, there is a new scale up ability for etcd clusters.

See Migrating Embedded etcd to External etcd for more information.

2.3.7.2. Modular Installer to Allow Playbooks to Run Independently

The installer has been enhanced to allow administrators to install specific components. By breaking up the roles and playbooks, there is better targeting of ad hoc administration tasks.

2.3.7.3. New Installation Experience Around Phases

When you run the installer, OpenShift Container Platform now reports back at the end what phases you have gone through.

If the installation fails during a phase, you will be notified on the screen along with the errors from the Ansible run. Once you resolve the issue, rather than run the entire installation over again, you can pick up from the failed phase. This results in an increased level of control during installations and results in time savings.

2.3.7.4. Increased Control Over Image Stream and Templates

With OpenShift Container Platform 3.7, there is added control over whether or not your cluster automatically upgrades all the content provided during cluster upgrades.

Edit the **`openshift_install_examples`** variable in the hosted file or set it as a variable in the installer.

```
RPM = /etc/origin/examples /etc/origin/hosted
```

```
Container = /usr/share/openshift/examples /usr/share/openshift/hosted

openshift_install_examples=false
```

Setting **`openshift_install_examples`** to **`false`** will cause the installer to not upgrade the imagestream and templates. **`True`** is the default behavior.

2.3.7.5. Installation and Configuration of CFME 4.6 from the OpenShift Installer

Red Hat CloudForms Management Engine (CFME) 4.6 is now fully supported running on OpenShift Container Platform 3.7 as a set of containers.

IMPORTANT

CFME 4.6 is not yet released. Until it is available, this role is limited to installing ManageIQ (MIQ), the open source project that CFME is based on. The following is provided mainly for informational purposes. The OpenShift Container Platform 3.7 documentation will be updated with more complete instructions on deploying CFME 4.6 after it has been released.

CFME is an available API endpoint on all OpenShift Container Platform clusters that choose to use it. More cluster administrators are now able to leverage CFME and begin experiencing the insight and automations available to them in OpenShift Container Platform.

To install CFME 4.6:

```
# ansible-playbook -v -i <YOUR_INVENTORY> \
    playbooks/byo/openshift-management/config.yml
```

NOTE

There is a known issue with this playbook.

To configure CFME 4.6 to consume the OpenShift Container Platform installation it is running on:

```
# ansible-playbook -v -i <YOUR_INVENTORY> \
    playbooks/byo/openshift-management/add_container_provider.yml
```

You can also automate the configuration of the provider to point to multiple OpenShift clusters:

```
# ansible-playbook -v -e container_providers_config=/tmp/cp.yml \
    playbooks/byo/openshift-management/add_many_container_providers.yml
```

NOTE

The ***/tmp/cp.yml*** file requires some manual configurations to create and use it correctly.

See Multiple Container Providers for more information.

2.3.8. Diagnostics

2.3.8.1. Additional Health Checks

More health checks are now available for administrators to run after installations and upgrades. Administrators need the ability to run tests periodically to help determine the health of the framework components within the cluster. OpenShift Container Platform 3.7 offers test functionality via Ansible playbooks that can be run and output can be sent as file-based output.

```
$ ansible-playbook playbooks/byo/openshift-checks/adhoc.yml
                curator
                diagnostics
                disk_availability
                docker_image_availability
                docker_storage
                elasticsearch
                etcd_imagedata_size
                etcd_traffic
                etcd_volume
                fluentd
                fluentd_config
                kibana
                logging
                logging_index_time
                memory_availability
                ovs_version
                package_availability
                package_update
                package_version

$ ansible-playbook playbooks/byo/openshift-checks/adhoc.yml -e
openshift_checks=fluentd_config,logging_index_time,docker_storage
```

Alternatively, they are included in the health playbook:

```
$ ansible-playbook playbooks/byo/openshift-checks/health.yml
```

To capture the output:

```
$ ansible-playbook playbooks/byo/openshift-checks/health.yml -e
openshift_checks_output_dir=/tmp/checks
```

2.3.9. Metrics and Logging

2.3.9.1. Docker Events and API Calls Aggregated to EFK as Logs

Fluentd captures standard error and standard out from the running containers on the node. With this change, fluentd collects all the errors and events coming from the docker daemon running on the node and sends it to Elasticsearch (ES).

Enable this via the OpenShift Container Platform installer:

```
openshift_logging_fluentd_audit_container_engine=true
```

The collected information is in operation indices of ES and only cluster administrators have visual access. The event message includes action, pod name, image name, and user time-stamp.

2.3.9.2. Master Events are Aggregated to EFK as Logs

The **eventrouter** pod scrapes the events from kubernetes API and and outputs to**STDOUT**. The **fluentd** plug-in transforms the log message and sends it to Elasticsearch (ES).

Enable **`openshift_logging_install_eventrouter`** by setting it to **`true`**. It is off by default. **Eventrouter** is deployed to the default namespace. Collected information is in operation indices of ES and only cluster administrators have visual access.

See the design documentation for more information.

2.3.9.3. Kibana Dashboards for Operations Are Now Shareable

This allows OpenShift Container Platform administrators the ability to share saved Kibana searches, visualizations, and dashboards.

When **`openshift_logging_elasticsearch_kibana_index_mode`** is set to **`shared_ops`**, one **`admin`** user can create queries and visualizations for other **`admin`** users. Other users can not see those same queries and visualizations.

When **`openshift_logging_elasticsearch_kibana_index_mode`** is set to **`unique`**, users can only see saved queries and visualizations they created. This is the default behavior.

See Aggregating Container Logs for more information.

2.3.9.4. Removed ES_Copy Method for Sending Logs to External ES

`ES_Copy` was replaced with the **secure_formard** plug-in for fluentd to send logs from fluentd to external fluentd (that can then ingest into ES). **`ES_COPY`** is removed from the installer and the documentation.

When **`openshift_installer`** is run for logging to upgrade to 3.7, the installer now checks for **`ES_COPY`** in the inventory and fails the upgrade with:

```
msg: The ES_COPY feature is no longer supported. Remove the variable from
your inventory
```

See Aggregating Container Logs for more information.

2.3.9.5. Expose Elasticsearch as a Route

By default, Elasticsearch (ES) deployed with OpenShift aggregated logging is not accessible from outside the logging cluster. This enables a route for external access to ES for those tools that want to access its data.

You now have direct access to ES using only your OpenShift token and have the ability to provide the external ES and ES Ops hostnames when creating the server certificate (similar to Kibana). Ansible tasks now simplify route deployment.

2.3.9.6. Removed Metrics and Logging Deployers

The metrics and logging deployers bare now replaced with **`playbook2image`** for **`oc cluster up`** so that **`openshift-ansible`** is used to install logging and metrics:

```
$ oc cluster up --logging --metrics
```

Check metrics and pod status:

```
$ oc get pod -n openshift-infra
$ oc get pod -n logging
```

2.3.9.7. Prometheus (Technology Preview)

OpenShift Container Platform operators deploy Prometheus (currently in Technology Preview and not for production workloads) on a OpenShift Container Platform cluster, collect Kubernetes and infrastructure metrics, and get alerts. Operators can see and query metrics and alerts on the Prometheus web dashboard, or bring their own Grafana and hook it up to Prometheus.

See Prometheus on OpenShift for more information.

2.3.9.8. Integrated Approach to Adding Hawkular OpenShift Agent (Tecnhology Preview)

Hawkular OpenShift Agent (HOSA) remains in Technology Preview and not for production workloads. It is packaged and can now be installed with the **`openshift_metrics_install_hawkular_agent`** option in the installer by setting it to **`true`**.

See Enabling Cluster Metrics for more information.

2.3.10. Developer Experience

2.3.10.1. AWS Service Broker

Users can seamlessly configure, deploy, and scale AWS services like Amazon RDS, Amazon Aurora, Amazon Athena, Amazon Route 53, and AWS Elastic Load Balancing directly within the OpenShift Container Platform console. For installation instructions, see AWS Service Broker and Getting Started Guide.

2.3.10.2. Template Instantiation API

Clients can now easily invoke a server API instead of relying on client logic.

2.3.10.3. Metrics

OpenShift Container Platform now includes:

- Prometheus metrics that show you the health of builds in the system (number running, failing, failure reasons, and so on).

- Timing information on build objects themselves to show how long they spent in various steps (not exposed as Prometheus metrics).

2.3.10.4. CLI Plug-ins (Technology Preview)

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

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

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

2.3.10.5. Chaining Builds

In OpenShift Container Platform 3.7, Chaining Builds is a better approach for producing runtime-only application images, and fully replaces the Extended Builds feature.

Benefits of Chaining Builds include:

- Supported by both Docker and Source-to-Image (S2I) build strategies, as well as combinations of the two, compared with S2i strategy only for Extended Builds.
- No need to create and manage a new assemble-runtime script.
- Easy to layer application components into any thin runtime-specific image.
- Can build the application artifacts image anywhere.
- Better separation of concerns between the step that produces the application artifacts and the step that puts them into an application image.

2.3.10.6. Default Hard Eviction Thresholds

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

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

See Handling Out of Resource Errors for more information.

2.3.11. Web Console

2.3.11.1. OpenShift Ansible Broker

In OpenShift Container Platform 3.7, Open Service Broker API is implemented, enabling users to leverage Ansible for provisioning and managing services from the Service Catalog. This is a standardized approach for delivering simple to complex multi-container OpenShift services via Ansible. It works in conjunction with Ansible Playbook Bundle (APB) for lightweight application definition. APBs can be used to deliver and orchestrate on-platform services, but could also be used to provision and orchestrate off-platform services (from cloud providers, IaaS, and so on).

OpenShift Ansible Broker supports production workloads and multiple service plans. There is now secure connectivity between Service Catalog and Service Broker.

You can interact with the Service Catalog to provision and manage services while the details of the broker remain largely hidden.

2.3.11.2. Ansible Playbook Bundles

Ansible Playbook Bundles (APBs) are short-lived, lightweight container image consisting of:

- a simple directory structure with named action playbooks.
- metadata (required and optional parameters, as well as dependencies).
- an Ansible runtime environment.

Developer tooling is included, providing a guided approach to APB creation. There is also support for the ***test*** playbook, allowing for functional testing of the service.) Two new APBs are introduced for MariaDB (SCL) and MySQL DB (SCL).

When a user provisions an application from the Service Catalog, the Ansible Service Broker will download the associated APB image from the registry and run it.

Developing APBs can be done in one of two ways: Creating the APB container image manually using standardized container creation tooling, or with APB tooling that Red Hat will deliver, which provides a guided approach to creation.

2.3.11.3. OpenShift Template Broker

The OpenShift Template Broker exposes templates through a Open Service Broker API to the Service Catalog.

The Template Broker matches the lifecycles of **`provision`**, **`deprovision`**, **`bind`**, and **`unbind`** with existing templates. No changes are required to templates, unless you expose **`bind`**. Your application will get injected with configuration details.

2.3.11.4. Initial Experience

OpenShift Container Platform 3.7 provides a better initial user experience with the Service Catalog. This includes:

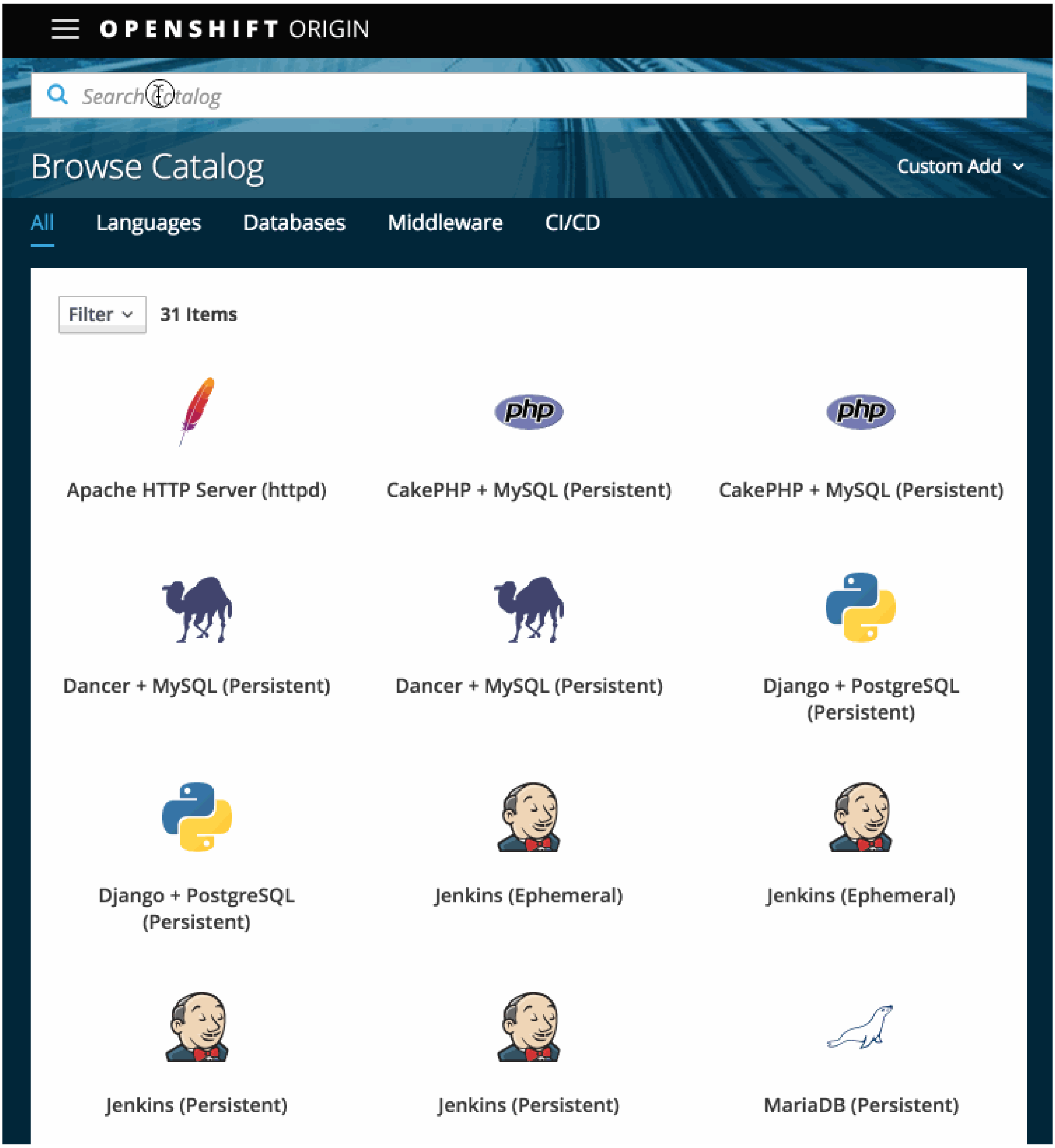
- A task-focused interface
- Key call-outs
- Unified search
- Streamlined navigation

The new user interface is designed to really streamline the getting started process, in addition to incorporating the new Service Catalog items. It shows the existing content (for example, builder images and templates) as well as catalog items (if the catalog is enabled).

> **NOTE**
>
> The new user experience can be enabled as a Technology Preview feature without the Service Catalog to be active. A cluster with this user interface (UI) would still be supported. Running the catalog UI without the Service Catalog enabled will work, but access to templates without the catalog will require a few extra steps.

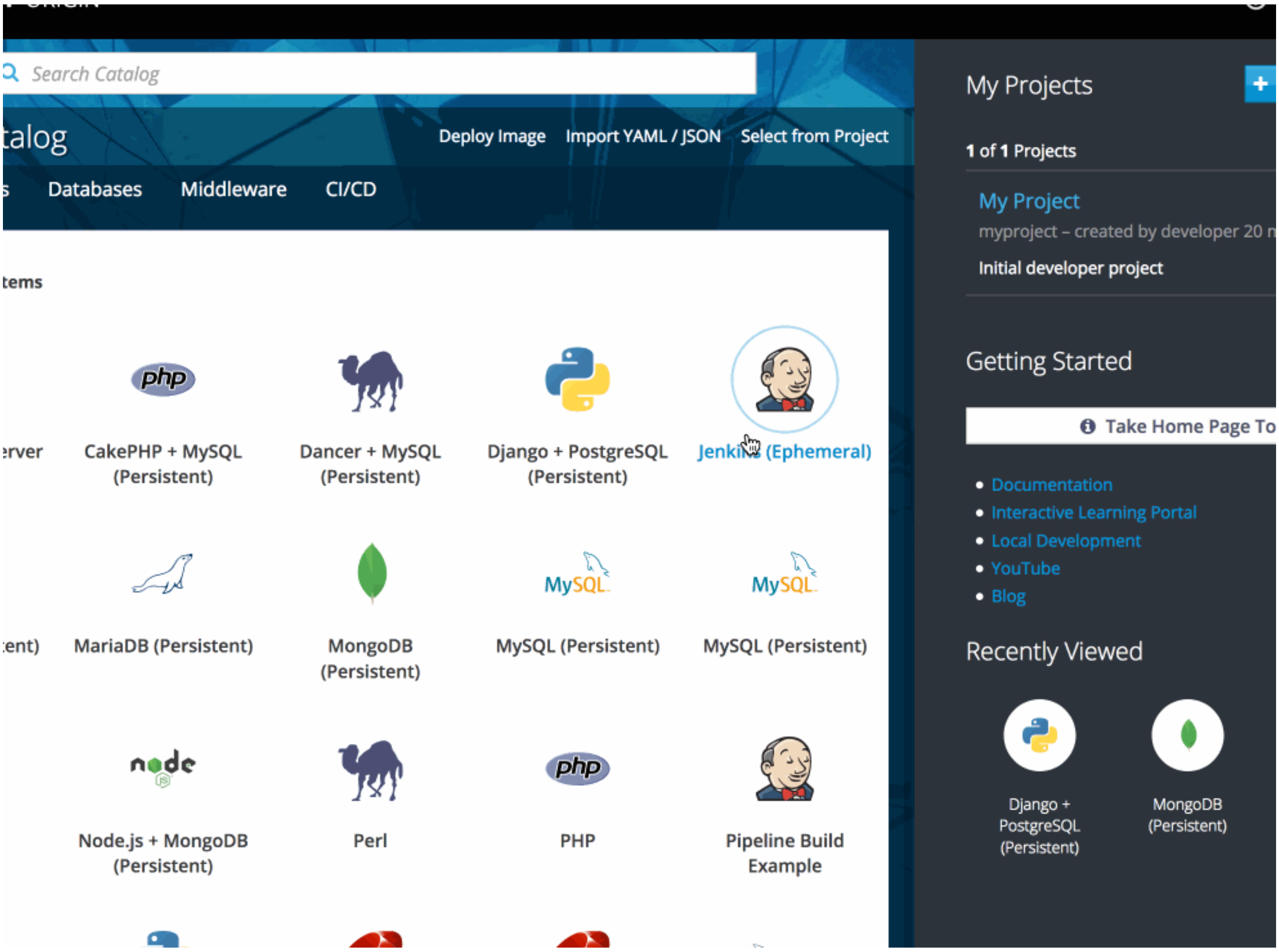
2.3.11.5. Search Catalog

OpenShift Container Platform 3.7 provides a simple way to quickly get what you want The new Search Catalog user interface is designed to make it much easier to find items in a number of ways, making it even faster to find the items you are wanting to deploy.



2.3.11.6. Add from Catalog

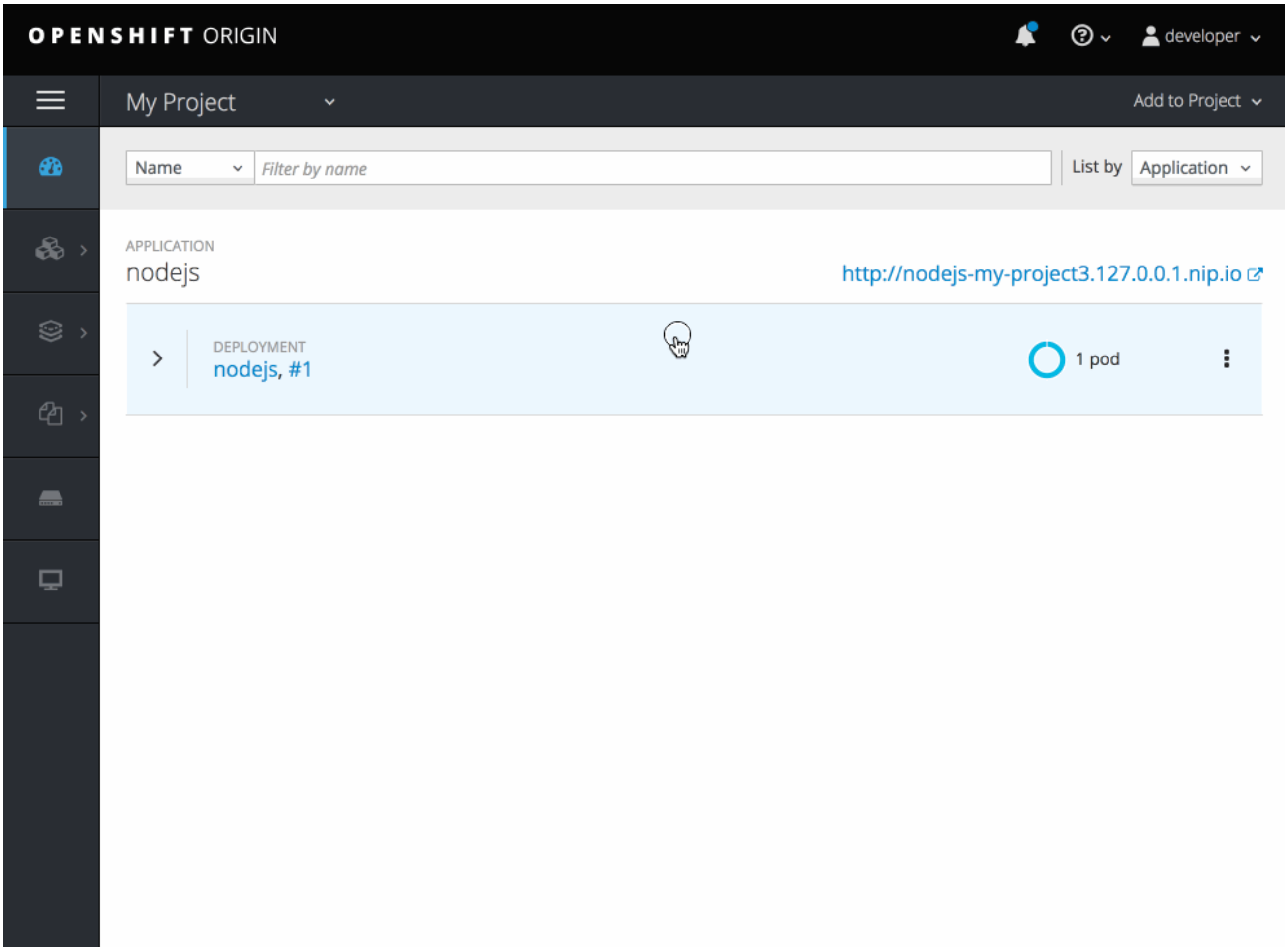
Provision a service from the catalog. Select the desired service and follow prompts for the desired project and configuration details.



2.3.11.7. Connect a Service

Once a service is deployed, get coordinates to connect the application to it.

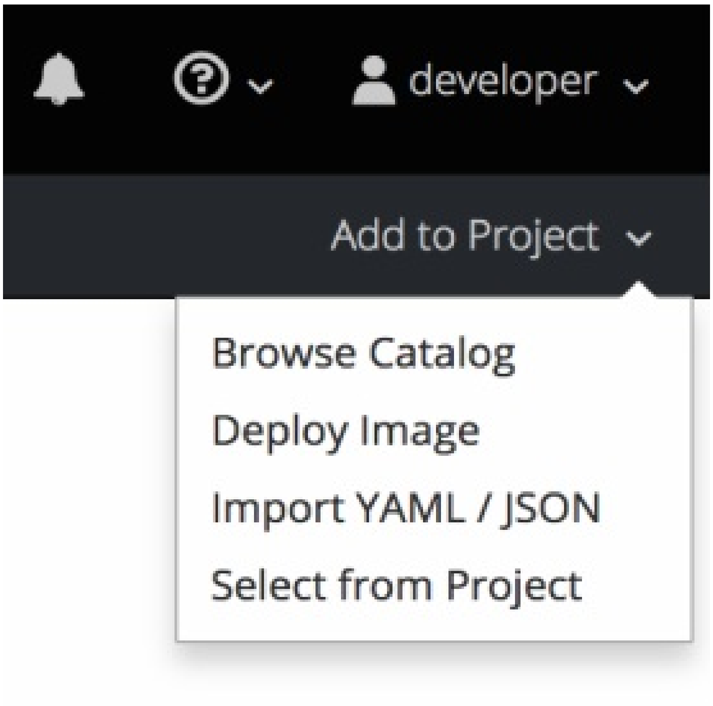
The broker returns a secret, which is stored in the project for use. You are guided through a process to update the deployment to inject a secret.



2.3.11.8. Include Templates from Other Projects

Since templates are now served through a broker, there is now a way for you to deploy templates from other projects.

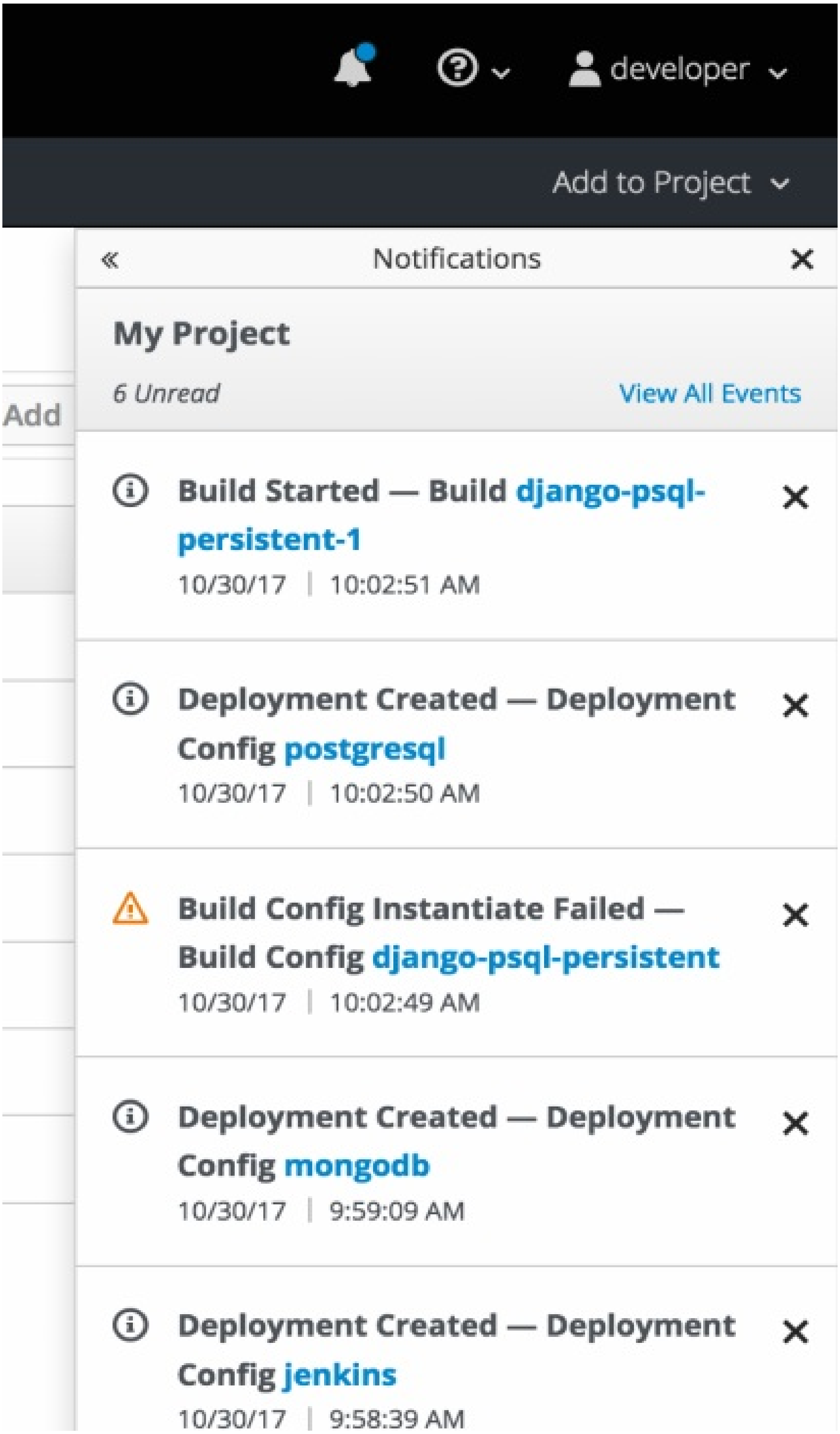
Upload the template, then select the template from a project.



2.3.11.9. Notifications

Key notifications are now under a single UI element, the notification drawer.

The bell icon is decorated when new notifications exist. You can mark all read, clear all, view all, or dismiss individual ones. Key notifications are represented with the level of information, warning, or error.



My Project

6 Unread

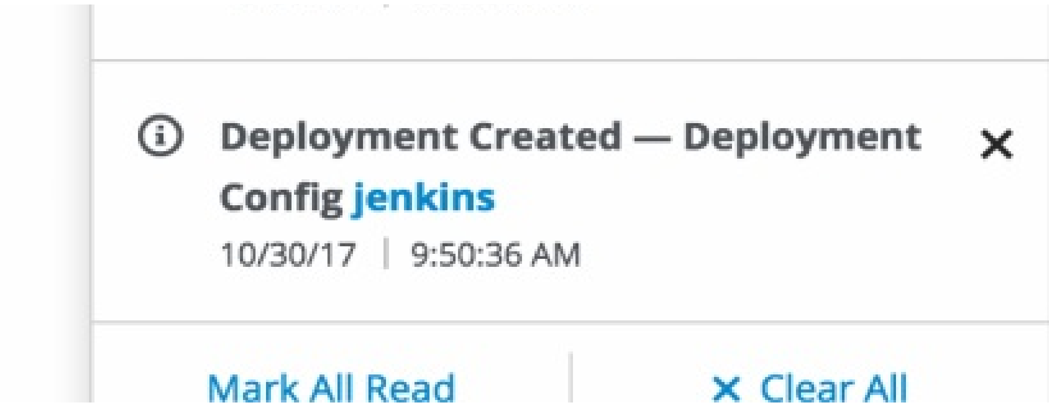
Build Started — Build django-psql-persistent-1
10/30/17 | 10:02:51 AM

Deployment Created — Deployment Config postgresql
10/30/17 | 10:02:50 AM

Build Config Instantiate Failed — Build Config django-psql-persistent
10/30/17 | 10:02:49 AM

Deployment Created — Deployment Config mongodb
10/30/17 | 9:59:09 AM

Deployment Created — Deployment Config jenkins
10/30/17 | 9:58:39 AM

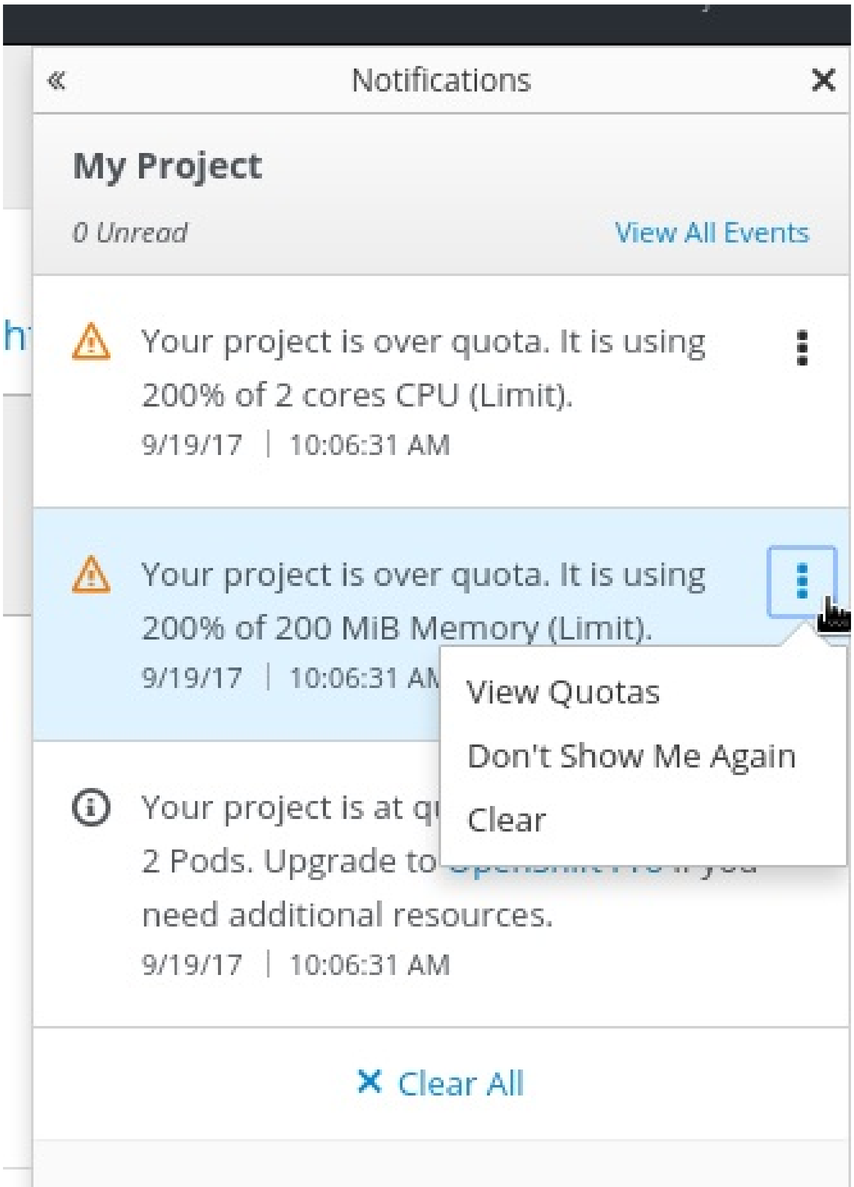


Deployment Created — Deployment Config jenkins

10/30/17 | 9:50:36 AM

2.3.11.10. Improved Quota Warnings

Quota notifications are now put in the notification drawer and are less intrusive.



Your project is over quota. It is using 200% of 2 cores CPU (Limit).
9/19/17 | 10:06:31 AM

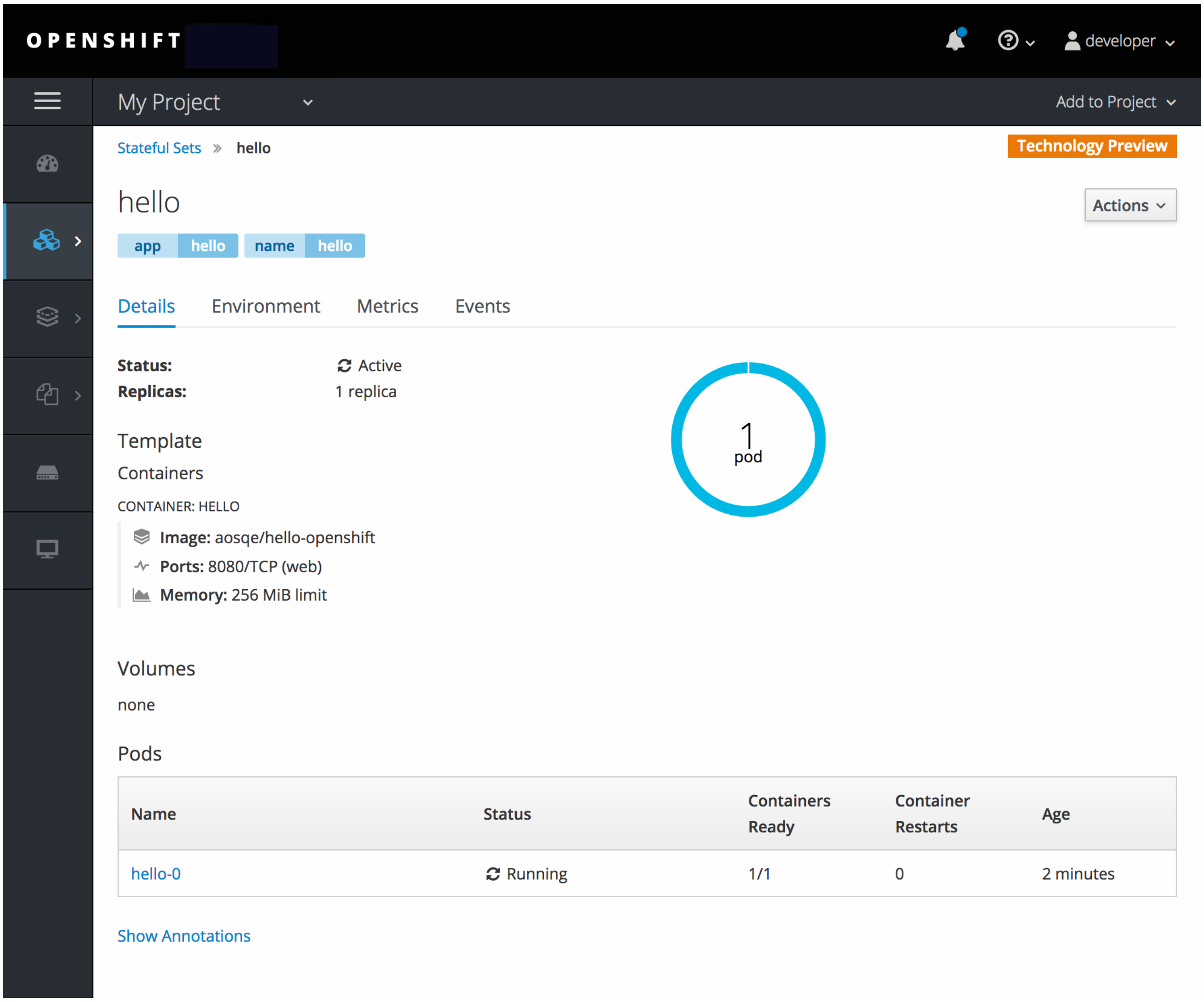
Your project is over quota. It is using 200% of 200 MiB Memory (Limit).
9/19/17 | 10:06:31 AM

Your project is at q… 2 Pods. Upgrade to … need additional resources.
9/19/17 | 10:06:31 AM

There are now separate notifications for each quota type instead of one generic warning. When at quota and not over quota, this is displayed as an informative message. Usage and maximum is displayed in the message. You can mark **Don’t Show Me Again** per quota type. Administrators can create custom messages to the quota warning.

2.3.11.11. Environment Variable Editor Added to the Stateful Sets Page

An environment variable editor is now added to the **Stateful Sets** page.



Status: Active
Replicas: 1 replica

Template

Containers

CONTAINER: HELLO

Image: aosqe/hello-openshift
Ports: 8080/TCP (web)
Memory: 256 MiB limit

Volumes

none

Pods

Name	Status	Containers Ready	Container Restarts	Age
hello-0	Running	1/1	0	2 minutes

2.3.11.12. Support for the EnvFrom Construct

Anything with a pod template now supports the `EnvFrom` construct that lets you break down an entire configuration map or secret into environment variables without explicitly setting `env name` to `key mappings`.

2.4. NOTABLE TECHNICAL CHANGES

OpenShift Container Platform 3.7 introduces the following notable technical changes.

Use the Ansible Version Shipping with OpenShift Container Platform
OpenShift Container Platform 3.6 and 3.7 were developed and tested using Ansible 2.3, which ships in the OpenShift Container Platform channels. Subsequently, the RHEL 7 Extras channel added Ansible 2.4, which has known issues with OpenShift Container Platform 3.6 and 3.7. If you experience any problems with the installer, downgrade to Ansible 2.3 by running `yum downgrade ansible-2.3\*` See BZ#1575063 for additional information.

API Connectivity Variables OPENSHIFT_MASTER and KUBERNETES_MASTER Are Now Deprecated

OpenShift Container Platform deployments using a custom strategy or hooks are provided with a container environment, which includes two variables for API connectivity:

- **OPENSHIFT_MASTER**: A URL to the OpenShift API .
- **KUBERNETES_MASTER**: A URL to the Kubernetes API exposed by OpenShift.

These variables are now deprecated, as they refer to internal endpoints rather than the published OpenShift API service endpoints. To connect to the OpenShift API in these contexts, use service DNS or the automatically exposed **KUBERNETES**service environment variables.

The **OPENSHIFT_MASTER** and **KUBERNETES_MASTER** environment variables are removed from deployment container environments as of OpenShift Container Platform 3.7.

openshift_hosted_{logging,metrics}_* Ansible Variables for the Installer Are Now Deprecated

The **openshift_hosted_{logging,metrics}_*** Ansible variables used by the installer have been deprecated. The installation documentation has been updated to use the newer variable names. The deprecated variable names are planned for removal in the next minor release of OpenShift Container Platform.

See Configuring Cluster Metrics and Configuring Cluster Logging for more information.

Removed generatedeploymentconfig API Endpoint

The **generatedeploymentconfig** API endpoint is now removed

Deprecated Policy Related APIs and Commands

A large number of policy related APIs and commands are now deprecated. In OpenShift Container Platform 3.7, the policy objects are completely removed and native RBAC is used instead. Any command trying to directly manipulate a policy object will fail. Roles and rolebindings endpoints are still available, and they proxy the operation to create native RBAC objects instead. The following commands do not work against a 3.7 server:

```
$ oc adm overwrite-policy
$ oc adm migrate authorization
$ oc create policybinding
```

NOTE

A 3.7 client will display an error message when trying these command against a 3.7 server, but will still work against a previous server version, and old client will just fail hard against a 3.7 server.

Red Hat Enterprise Linux Atomic Host Version 7.4.2.1 or Newer Required for Containerized Installations

In OpenShift Container Platform 3.7, containerized installations require Red Hat Enterprise Linux Atomic Host version 7.4.2.1 or newer. You can also perform containerized installations on Red Hat Enterprise Linux.

Labeling Clusters for Amazon Web Services

Starting with 3.7 versions of the installer, if you configured AWS provider credentials, you must also ensure that all instances are labeled. Then, set the **openshift_clusterid** variable to the cluster ID. See Labeling Clusters for Amazon Web Services (AWS) for more information.

Stricter Security Context Constraints (SCCs)

With the release of OpenShift Container Platform 3.7, there are now some stricter security context constraints (SCCs). The following capabilities are now removed:

- **nonroot** drops **`KILL`**, **`MKNOD`**, **`SETUID`**, and **`SETGID`**.
- **hostaccess** drops **`KILL`**, **`MKNOD`**, **`SETUID`**, and **`SETGID`**.
- **hostmount-anyuid** drops **`MKNOD`**.

It is possible that the pods that previously were admitted by these SCCs, and were using such capabilities, will fail after upgrade. In these rare cases, the cluster administrator should create a custom SCC for such pods.

CloudForms Management Engine (CFME) Support Changes

OpenShift Container Platform 3.7 now fully supports Installation and Configuration of CFME 4.6 from the OpenShift Installer. As previously stated, CFME 4.6 is not currently released. The current CFME installer implementation in OpenShift Container Platform 3.7, however, is incompatible with the Technology Preview deployment process of CFME 4.5 as described in the OpenShift Container Platform 3.6 documentation.

The OpenShift Container Platform 3.7 documentation will be updated with more complete instructions on deploying CFME 4.6 after it has been released.

Node Authorizer and Admission Plug-in for Managing Node Permissions

In OpenShift Container Platform 3.7, the node authorizer and admission plug-in are used to manage and limit a node’s permissions. Therefore, nodes should be removed from the group that previously granted them broad permissions across the cluster:

```
$ oc adm policy remove-cluster-role-from-group system:node system:nodes
```

In OpenShift Container Platform 3.8, this step should be performed automatically via Ansible as a post-upgrade step.

The kube-service-catalog Namespace Is Global

The **`kube-service-catalog`** namespace is now made global by Ansible. Therefore, if you want multicast to work in vnid 0, you must set the **`netnamespace.network.openshift.io/multicast-enabled=true`** annotation on both namespaces (**`default`** and **`kube-service-catalog`**).

Migration to Kubernetes Role-based Access Control (RBAC)

Steps Taken During the 3.6 Release

A custom migration controller was created to automatically migrate OpenShift authorization policy resources to the equivalent RBAC resources:

1. If an OpenShift authorization policy resource was created or modified or deleted, the action was automatically mirrored to the corresponding RBAC resource
2. Changes directly applied to RBAC resources were, generally, automatically rolled back and forced to match the corresponding OpenShift authorization policy resource. If no corresponding resource existed, the RBAC resource would be deleted.

In essence, OpenShift authorization policy objects were the source of truth, and the RBAC objects were forced into matching these objects.

Release 3.6 Pre-upgrade Steps Before Upgrading to 3.7

There is a small set of configurations that are possible in OpenShift authorization policy resources that are not supported by RBAC. Such configurations require manual migration based on the use case. To guarantee that all Openshift authorization policy objects are in sync with RBAC, the **`oc adm migrate authorization`** command has been added. This read-only command emulates the migration controller logic, and reports if any resource is out of sync. It is run as a pre-upgrade step via an Ansible playbook and will cause the upgrade to fail if the objects are not in sync.

During a Rolling Upgrade from Release 3.6 to 3.7
The following scenario describes a rolling upgrade

1. One master is upgraded and starts proxying OpenShift authorization policy resources and authorizing against RBAC objects.

2. Old masters are still running the migration controller and one of them holds the controller leader election lock (either because it already had it or because it gained it by the first master being upgraded).

3. The new master cannot modify any RBAC or proxied OpenShift authorization policy objects because the migration controller will undo all changes.

4. Old masters can change OpenShift authorization policy resources and the migration controller will sync these to RBAC, making the changes visible to the new master.

5. The new master does not have the migration controller.

6. Controllers only speak to their local masters in OpenShift installed via Ansible, thus the migration controller is guaranteed to only communicate with the old masters.

7. There is a small chance that a 3.7 controller process will become the leader once two masters have been upgraded (meaning no migrations of policy objects will occur after this point).

8. Once all masters have been upgraded from 3.6 to 3.7, OpenShift authorization policy objects will be always proxied to RBAC objects.

9. The migration controller will be gone and it will be possible to make changes to RBAC objects directly.

Considerations for Administrators During Rolling Upgrade

Avoid actions that require changes to OpenShift authorization policy resources such as the creation of new projects. If a project is created against a new master, the RBAC resources it creates will be deleted by the migration controller since they will be seen as out of sync from the OpenShift authorization policy resources. If a project is created against an old master and the migration controller is no longer present due to a 3.7 controller process being the leader, then its policy objects will not be synced and it will have no RBAC resources. After the 3.7 upgrade is complete, the following read-only script can be used to determine what namespaces lack RBAC role bindings (it is up to the cluster administrator to decide how to remediate these namespaces):

```
#!/bin/bash

set -o errexit
set -o nounset
set -o pipefail

for namespace in $(oc get namespace -o name); do
   ns=$(echo "${namespace}" | cut -d / -f 2)
   rolebindings_count=$(oc get rolebinding.rbac -o name -n "${ns}" | wc -
l)
```

```
   if [[ "${rolebindings_count}" == "0" ]]; then
       echo "Namespace ${ns} has no role bindings which may require
further investigation"
   else
       echo "Namespace ${ns}: ok"
   fi
done
```

RBAC and OpenShift Authorization Policy in Release 3.7
In 3.7, the RBAC objects become the source of truth. The OpenShift authorization policy objects no longer exist as real objects; the APIs are proxied to the RBAC resources. Therefore, creating, modifying, or deleting OpenShift authorization policy resources seamlessly results in actions against RBAC objects. The API master handles the conversion between these resources and legacy clients will continue to work as if nothing has changed. The RBAC objects also support watches, unlike the OpenShift authorization policy resources.

Policy-based resources have been removed in 3.7. However, RBAC role and binding objects are available and provide equivalent functionality.

Non-production Installations
The recommended way for installing non-production environments may change significantly in the next minor release of OpenShift Container Platform. Administrators should avoid tight coupling to the **`atomic-openshift-installer`** tool as part of the quick installer installation and upgrade processes.

2.5. BUG FIXES

This release fixes bugs for the following components:

Authentication

- The secret for the private browser OAuth client was not correctly initialized. Therefore, the request token endpoint did not work. This bug fix correctly initializes the browser OAuth client on server start. The request endpoint can now be used to request tokens. (**BZ#1491193**)

- The LDAP sync/prune command did not take into account the use of **`groupUIDNameMapping`** with a whitelist. The sync/prune command would fail with "group not found" errors because it would query for the wrong group name. With this bug fix, the command was updated to take **`groupUIDNameMapping`** into account when using a whitelist. Now, the command queries for the correct group name when **`groupUIDNameMapping`** and a whitelist are used together. (**BZ#1484831**)

- **`RoleBinding`** objects can now be created without first creating a **`PolicyBinding`** object. (**BZ#1477956**)

Builds

- **`ImageStream`** output references and their corresponding secrets were resolved during build creation time. If the output imagestream did not exist yet, no push secret would be be computed, resulting in a build failure during push. With this bug fix, the **`ImageStream`** output and push secret will be computed when preparing to run the build, under logic which will retry until the **`imagestream`** is available. Builds that are started before the output **`imagestream`** exists will no longer fail during the push phase. (**BZ#1443163**)

- Build, delete, and watch events, and the current Jenkins job being canceled were not handled when a build was canceled in OpenShift. Various negative, inconsistent Jenkins job results occurred along with many exception stack traces in the Jenkins system log. With this bug fix, Jenkins jobs are halted as soon as the build watch event detects that a build was deleted as the result of a build cancel action taken within OpenShift. There is now consistent, sensible behavior for the Jenkins users when builds are canceled or deleted. (BZ#1473329)

- Source-to-image was not closing stdin/out/err pipes correctly in some error cases, causing a hang to occur. This was causing some OpenShift builds to hang in **running** status. (BZ#1442875)

- The **openshift jenkins sync** plug-in was updating Jenkins pipeline build status annotations every second, regardless of whether the status changed. The frequency of updates would put unnecessary stress on the etcd instance backing openshift master. Now, Jenkins pipeline build status annotations are only updated if the status actually changes, or 30 seconds have passed. (BZ#1475867)

- Permissions on directories injected as a build input via the image source input mechanism have user-only access permissions. The resulting application image cannot access the content when run as a random user ID. The directories will now be injected with group permissions, which allows the container user to access the directories. The directories will now be accessible at runtime as desired. (BZ#1480312)

- When no tag is explicitly set, docker pulls all images. Builds would pull more images than necessary and take longer than needed. With this bug fix, a default tag will be set when the user does not supply a tag. Only a single image will be pulled for the build. (BZ#1498178)

- The BitBucket build trigger webhook did not handle older versions of the webhook payload. Builds could not be triggered by older versions of the BitBucket server. This bug fix adds support for the older payload format. Builds can now be triggered by older versions of BitBucket. (BZ#1500731)

- A regression bug was reported whereby source-to-image builds would fail if the source repository file system contained a broken symlink (pointing to a non-existent item). This is now resolved. (BZ#1506173)

Command Line Interface

- The **oc** binary for macOS is not signed. Some of the customer's company policies do not allow users to install unsigned binaries. This bug fix signs the **oc** binary using a Red Hat certificate. The **oc** binary is now trusted by companies that restrict the installation of unsigned binaries. (BZ#1436093)

- The **`git clone`** command was being run without a timeout. Therefore, the **`oc new-app`** command was timing out. With this bug fix, **`oc new-app`** now uses **`git ls-remote`** with a timeout and the **`oc new-app`** command will not timeout. (BZ#1488283)

Containers

- The **`POOL_META_SIZE`** configuration item is now added. The thin pool metadata size was set to .1% of free space of volume group. **`POOL_META_SIZE`** allows the operator to customize the size of thin pool metadata volume size to meet their workload. (BZ#1451769)

Deployments

- Shortly after OpenShift starts, the caches might not yet be synchronised. Asa result, scaling the replication controllers might fail. Retry the scaling when there is a cache miss. With this bug fix, the replication controllers are scaled properly. (BZ#1427992)

Images

- A .NET jenkins slave image for performing .NET CI/CD flows is now offered. This makes it easier to build and test .NET code bases using Jenkins. A .NET slave image is provided and configured out of the box in the Jenkins master image. (**BZ#1451403**)

- Jenkins now installs all plug-ins via one RPM, and the missing plug-in is now included. (**BZ#1481010**)

- **`importPolicy.insecure`** is ignored in **`oc import-image <imagestream:tag>`** As a result, re-import from an insecure registry fails because it expects a valid SSL certificate. When the image stream tag exists, use its **`importPolicy.insecure`**. With this bug fix, re-import succeeds. (**BZ#1494231**)

Image Registry

- Images younger than the threshold are not added to the dependency graph. A blob that is used by a young image and by a prunable image is deleted because it has no references in the graph. Add young images to the graph and mark them as non-prunable. With this bug fix, the blob has references and is not deleted. (**BZ#1487408**)

- The image pruning algorithm would consider only managed images for pruning. As a result, mirrored blobs for not managed images could not be pruned. External images could not be removed using pruning. With this bug fix, the pruning algorithm evaluates all the images, not just managed images. External images and their blobs can now be pruned. (**BZ#1441028**)

- Previously, a bug in a regulator of concurrent file system access could cause a routine to hang. This caused many builds to hang during the registry push.This bug fix corrects the regulator. As a result, concurrent pushes no longer hang. (**BZ#1436841**)

- Previously, the **`oc adm prune images`** command would print confusing errors (such as operation timeout). This bug fix enables errors to be printed with hints. As a result, users are able to prune images, including images outside of the OpenShift cluster. (**BZ#1469654**)

- The registry previously appended forwarded target ports to redirected location URLs. The client’s new request to the target location lacked credentials, and as a result, image push failed due to an authorization error. This bug fix rebased the registry to a newer version that fixes forwarding processing logic. As a result, clients can push images successfully to the exposed registry using arbitrary TLS-termination. (**BZ#1471707**)

- Previously, **`imagestreamtags`** were not checked for dangling image references. This caused references to deleted images to be retained. This bug fix removes references to deleted images. As a result, deleting an image should allow references to the image to be deleted from **`imagestreamtags`**. (**BZ#1386917**)

- Documentation and command help are now updated to include information on troubleshooting insecure connections to the secured registry. Error messages are now printed with hints, and new flags have been added to allow for insecure fall-back. As a result, users can now easily enforce both secure and insecure connections. (**BZ#1448595**)

Installer

- Previously, the installation would fail when creating the Heketi secret because the key file did not copy on the first master host. This bug fix enables the installer to copy the SSH private key to the master node. (**BZ#1477718**)

- The Ansible quick install would previously fail if the hostname was manually defined containing an uppercase letter. As a result, Kubernetes converted the names of the nodes to lowercase and did not recognize a node name with an uppercase letter. This bug fix ensures that hostnames for node objects are created with lowercase letters. (**BZ#1396350**)

- Previously, the node service was not restarted when Open vSwitch was restarted, which could result in a misconfigured networking environment. This bug fix updates the services to ensure that the node service is restarted whenever Open vSwitch is restarted. (**BZ#1453113**)

- Previously, Ansible facts added the **`svc`** domain to the **`NO_PROXY`** settings. As a result, users behind proxies were not able to push to registry by DNS. This bug fix adds the **`svc`** domain to the Ansible facts code. As a result, users behind a proxy can now push to registry by DNS. (**BZ#1467776**)

- The flannel network was previously defined using the same subnet as the Kubernetes services subnet. This caused a conflict between services and SDN networks. The flannel network is now correctly defined by the osm_cluster_network_cidr variable. (**BZ#1473858**)

- The necessary role for role binding in openshift_metrics was missing due to being processed out of order in the role. The role binding creation would fail and the role would fail to install. This bug fix updates the metrics to create the role immediately. As a result, role binding can be created during installation. (**BZ#1476195**)

- The etcd scaleup playbook had an error where it attempted to run commands on hosts other than the host that was currently being scaled up resulting in an error if the other hosts did not yet have certain dependencies met. The playbooks now properly target only the host currently being scaled up. (**BZ#1490739**)

- The stand-alone entry point for the **`openshift_storage_nfs`** task did not have the **`os_firewall`** role included. This resulted in the firewall not being properly installed and configured. The **`os_firewall`** has been added to the play. (**BZ#1491657**)

- The etcd quota backend was set to 2GB by default. This resulted in a cluster going into a hold state, blocking all writes into the etcd storage. The default quota backend was increased to 4GB by default to encompass the storage needs of bigger clusters. (**BZ#1492891**)

- When a company CA is added as a named certificates, the CA is added to **`ca-bundle.crt`** as well. This can cause client certificate popups when using IE,Safari or Chrome if the user has client certs configured via the browser. The code has been changed to not use the **`ca-bundle.crt`** and use the internal CA for client cert CA. (**BZ#1493276**)

- As part of deprecating the use of **`openshift_hosted_{logging,metrics}_*`** variables, a default size for the storage volume wasn’t set for an NFS installation. As a result, the playbook would fail that the variable was not defined at runtime. The code was changed to use the default '10Gi' if not specified. The installer runs as expected. (**BZ#1495203**)

- The disconnected installer did not have a way to specify a username/password to login to the docker repository to access downloaded images, requiring the user to disable authentication. The installation script now includes a mechanism for entering credentials. (**BZ#1500642**)

- A new Docker option **`--signature-enabled`** that was introduced in a recent Docker release is set to **`False`** by default. The OpenShift Container Platform installation removes the parameter during the installation and Docker would get the default value of **`True`**. The Ansible scripts have been changed to include this option. (**BZ#1502560**)

- Upgrading the logging component from 3.4.1 to 3.5.0 using Ansible failed with a **`No Elasticsearch pods found running`** error. The logging upgrade has been disabled as the EFK stack used for 3.4 and 3.5 is the same. The upgrade functionality is not necessary. (**BZ#1435144**)

- When using ansible to configure the openID-connect provider for the OpenID and GitLab providers resulted in an error when setting **`challenge`** to true. This happens because of the validate function did not allowing this. The Ansible validate function was removed for OpenID and GitLab providers. The installation can complete successfully, and login succeeds. (**BZ#1444367**)

- Docker 1.12.6-34 uses ***/etc/containers/registries.conf*** to define registries, but OpenShift Container Platform installer uses ***/etc/sysconfig/docker***. As a result, system containers were reading registry information from the incorrect file. The code was changed to duplicate the registries in both locations to ensure additional/blocked/insecure registries are honored. (**BZ#1460930**)

- A containerized installation with system containers enabled (**`use_system_containers=true`**) failed due to missing mounts. The code was updated so that the install performs as expected. (**BZ#1463574**)

- The OpenShift Container Platform would correctly fail is the public host name was 64 characters or greater. However, the error message displayed did not report the source of the failure. The installer has been changed to report if the installation failed due to hostname length. (**BZ#1467790**)

- When installing the service catalog, the template service broker (TSB) was not getting created. As a result, the TSB had to be created manually. The code has been changed so that the TSB is created automatically. (**BZ#1470623**)

- Input for **`include_granted_scopes`**, which was expected to become a single quoted boolean string, was instead being interpreted and written to the file incorrectly. The resulting configuration file could have the wrong value for **`include_granted_scopes`** and removal of a code block attempted to interpret the input for **`include_granted_scopes`**. Input that is expected to land via **`include_granted_scopes`** now passes to the ***master-config.yml*** as expected. (**BZ#1488505**)

- Because the Docker image availability health check does not support authenticated registries, checks failed when running against an authenticated registry. The code was changed to allow Docker to health check authenticated registries. (**BZ#1488833**)

- Running the **`redeploy-router-certificates.yml`** playbook caused the router pod to fail (**`CrashLoopBackOff`**). The code was changed so that after running the **`redeploy-router-certificates.yml`** playbook, the router pod runs as expected. (**BZ#1490186**)

- With Ansible 2.3, warnings are issued when using Jinja delimiters in 'when' conditions. The delimiters have been removed from the code base to avoid these warnings. (**BZ#1490268**)

- Due to an earlier code change, the installation failed when giving a wildcard certificate to the installer. The code has been changed to properly copy a wildcard certificate during installation. (**BZ#1492786**)

- Because of internal refactoring, the list of hostnames in the **`NO_PROXY`** file was empty. The facts have been restored The list of NO_PROXY names is correctly defined. (**BZ#1495142**)

- When **`openshift_docker_use_system_container`** was set to **`false`**, the installer was incorrectly attempting to start the container engine, resulting in the installation failing. The installer code was changed and the installation proceeds as expected. (**BZ#1496725**)

- The installer can now use an inventory specified as a directory rather than just a single file. This adds a parameter `INVENTORY_DIR` to the openshift-ansible image such that the user can indicate that ansible-playbook should use a mounted inventory directory. (**BZ#1498908**)

- The logic for selecting the Enterprise registry was moved to a location that which was never read when installing system containers. Enterprise installs using system containers would fail as the openshift-ansible image could not be found in the Docker hub registry. Moved the enterprise registry logic into a high level playbook so that it is set for all runtime set ups. The enterprise images can be found and installation works. (**BZ#1503860**)

- Due to recent simplification and refactoring there was a possibility of ***/etc/atomic.conf*** not being updated with proxy values before the first Atomic command was executed. Proxy use with the Atomic command did not work during the install. A new openshift_atomic role has been created for atomic specific tasks. The first task added is proxy which handles updating /etc/atomic.conf to ensure the proper proxy configuration is configured. This task file is then included (via include_role) in system container related task files. The atomic command always is able to use the properly defined proxy settings. (**BZ#1503903**)

- An undefined variable was used in a task. The undefined variable caused a jinja template evaluation error which would crash the installation. The undefined variable has been removed and replaced with more informative error text. The playbook does not error out for external NFS storage class installations. (**BZ#1504535**)

- The OpenShift Health Checker was not part of an Installer Phase and was not reported after playbook execution. The OpenShift Health Checker section of the primary installer path has been moved to its own section and an installer 'phase' has been added to report on installer status. (**BZ#1504593**)

- When updating the `openshift-ansible` package, all subpackages are now updated in order to keep them in sync. (**BZ#1506971**)

- The NetworkManager dispatcher script responsible for configuring a host to use dnsmasq operated in a non-atomic manner, resulting in failed DNS queries during boot up. The script has been refactored to ensure that required services are verified before ***/etc/resolv.conf*** is reconfigured. (**BZ#1410288**)

- Using the Ansible installer to install metrics with dynamic storage failed. Installation now fails if the parameter storage kind = 'dynamic' is set without enabling dynamic provisioning. (**BZ#1415297**)

- An error occurred from the yum module during the upgrade process. Yum transactions are now retried. (**BZ#1479533**)

- The 'registry-console' image stream did not have a source tag specified, causing it to be improperly imported.The source tag has been added to the image stream ensuring that it imports properly. (**BZ#1480442**)

- When enabling API aggregation with the ovs-multitenant SDN driver, creating a global project failed due to a performance latency issue. While creating a global project, the netnamespace is now checked to ensure availability and the Ansible Playbook Bundle finishes the operation. (**BZ#1487959**)

- The device mapper kernel modules may not have been loaded on a host if `overlay2` storage was used, which prevented the gluster storage system from working properly. With this fix, the installer now ensures that when gluster is used the `dm_thin_pool`, `dm_snapshot`, and `dm_mirror` modules are loaded. (**BZ#1490905**)

- Previously, if there was no DNS search path in **/etc/resolv.conf**, then the NetworkManager dispatcher would omit adding **`cluster.local`** to the search path. With this bug fix, the dispatcher script was updated to ensure that a search path is created if one did not already exist. (**BZ#1496593**)

- The example inventories have been updated to clearly indicate that the NFS export directory must only consist of lowercase alphanumeric characters, hyphens or periods, and must start and end with an alphanumeric character. (**BZ#1488366**)

- The quick installer tool, **`atomic-openshift-installer`**, was initially blocked for use with OpenShift Container Platform 3.7 due to a bug. This has now been fixed in the latest update. (**BZ#1509112**)

Logging

- Messages were read into Fluentd's memory buffer and were lost if the pod was restarted because Fluentd considered them read, but they were not pushed to storage. This caused the loss of any message not stored, but already read by Fluentd. This fix replaced the memory buffer with a file based buffer. As a result, the file buffered messages are pushed to storage once Fluentd restarts. (**BZ#1460749**)

- Kibana visualizations and dashboard for monitoring container and pod logs allows administrator users, cluster-admin or cluster-reader, to view logs by deployment, namespace, pod, and container. The script **`es_load_kibana_ui_objects`** is used to load dashboards and other Kibana UI objects for the given user. To use, run **`oc exec $espod — es_load_kibana_ui_objects user-name`**. It exists inside the Elasticsearch and ES-OPS pod, and must be run inside those pods. Additionally, it requires some indices and other objects set up by the OpenShift Elasticsearch plug-in, so the user must login to Kibana or Elasticsearch before using this script. This will also add an index pattern for project.* and load the necessary index pattern file. Kibana visualizations and dashboard gives administrators an easier way to view Kubernetes/OpenShift related logs in the cluster, allowing admin users have graphs and a dashboard to use to view logs from OpenShift pods and containers. (**BZ#1467963**)

- The execute bit in the downstream repo was previously not set for **`run.sh`**. (**BZ#1474715**)

- The value of the **`buffer_chunk_limit`** is now configurable, and defaults to 1M. To configure the **`buffer_chunk_limit`**, set the value to the environment variable **`BUFFER_SIZE_LIMIT`** or **`openshift_logging_fluentd_buffer_size_limit`** in the Ansible inventory file. To cover various types of input, **`buffer_chunk_limit`** needs to be configurable. The "size of the emitted data exceeds buffer_chunk_limit" can be fixed by configuring **`buffer_chunk_limit`**. (**BZ#1413147**)

- Role permissions were generated based upon the project, causing queries to be disallowed if they involved multiple indices. This fix generates role permissions based on the user and not the project, allowing users to query across multiple indices. (**BZ#1445425**)

- The **`openshift-elasticsearch-plugin`** was creating ACL roles based on the provided name, which could include slashes and commas. This caused the dependent **`lib`** to not properly evaluate roles. This fix hashes the name when creating ACL roles so they no longer contain the invalid characters. Now, users can use kibana and logging. (**BZ#1456584**)

- The **`ansible`** parameter name is confusing to use and does not properly reflect how it is consumed by Fluentd. This fix removed the parameter, allowing Fluentd to consistently collect logs based on the source it detects. (**BZ#1466152**)

- Elasticsearch was logging to console logs, resulting Elasticsearch ending up in a feedback loop ingesting its own logs. This fix turned off console logs in favor of file logs. As a result, the feedback loop is broken but users will need to setup Elasticsearch log volume with file rotation to get ES logs. Additionally, **oc logs** against an Elasticsearch pod will no longer be sufficient to retrieve Elasticsearch pod logs. (**BZ#1432607**)

- Elasticsearch default value for sharing storage between Elasticsearch instances was wrong. This caused the incorrect default value to be allowed an Elasticsearch pod starting up (when another Elasticsearch pod was shutting down) to create a new location on the PV for managing the storage volume, duplicating data, and in some instances, potentially causing data loss. With this fix, all Elasticsearch pods now run with **node.max_local_storage_nodes** set to **1**. As a result, the Elasticsearch pods starting up and shutting down will no longer share the same storage and prevent the data duplication and data loss. (**BZ#1460564**)

- Use underscores when providing memory switches to the Nodejs runtime instead of dashes. As a result, the Nodejs interpreter understands the request. (**BZ#1464020**)

- The **openshift_logging_purge_logging** Ansible variable was introduced to purge logging persistent data. Because the **openshift_logging_install_logging=false** will keep persistent data, there was a need for a complete uninstall. As a result, there are no changes to **openshift_logging_install_logging**, with the additional variable **openshift_logging_purge_logging** for complete uninstall. (**BZ#1467265**)

- In the configuration for the Fluentd systemd input plug-in, the **read_from_head** parameter was not set properly based on the environment variable **JOURNAL_READ_FROM_HEAD** or its corresponding Ansible parameter **openshift_logging_fluentd_journal_read_from_head**. Due to the problem, the full contents of pre-existing logs were indexed instead of the latest logs captured by “tail” when a **pos_file** does not exist, which happens when the logging system is initially deployed or a **pos_file** is deleted. With this bug fix, the parameter is correctly set. And based on the setting, if **JOURNAL_READ_FROM_HEAD=true**, all the logs are indexed; if **JOURNAL_READ_FROM_HEAD=false**, logs read from "tail" are indexed when a **pos_file** does not exist. (**BZ#1488941**)

- When deploying **logging-fluentd** with **secure-forward** to send the collected logs to **logging-mux**, it requires **openshift_logging_mux_client_mode=maximal** with **openshift_logging_use_mux=True** in the ansible inventory if the Fluentd container and the **mux** container are on the same node. If **openshift_logging_mux_client_mode=maximal** is set without **openshift_logging_use_mux=True**, the **mux** secret directory **/etc/fluent/muxkeys** is mounted in the Fluentd container although the secret directory does not exist. It makes Fluentd hang when it tries to access the **mux** secrets at the startup time. This patch checks the value of **openshift_logging_mux_client_mode** and **openshift_logging_use_mux** in the Ansible playbook and if the former is true while the latter is false, then it does not mount the **mux** secret directory in the Fluentd container. Also, if the Fluentd start script finds the **mux** secret directory does not exist, it disables **openshift_logging_mux_client_mode** even if it is enabled. (**BZ#1490647**)

- The **json-file** parser was assuming the "time" field was a Time object instead of a String object, which does not have a "utc" method, causing the logs to fill with the error. This fix checks the type of object for the "time" field, and convert the String to a Time object if necessary. As a result, **json-file** read time values are parsed correctly with no errors. (**BZ#1491405**)

- The **openshift-elasticsearch-plugin** was creating ACL roles based on the provided name which could include slashes and commas. This caused the dependent **lib** to not properly evaluate roles. This fix hashes the name when creating ACL roles so they no longer contain the invalid characters. As a result, users can use Kibana and logging. (**BZ#1494239**)

Web Console

- Previously in the web console pod terminal, you could not enter third-level characters using the AltGr key such as ‘|’ (pipe) in some keyboard layouts. Now Alt+Gr-<character> combinations work properly in the web console pod terminal. (**BZ#1292507**)

- In the web console, copying and pasting content from the terminal could result in extra spaces being added to the end of each line. Now when you copy content from the terminal, no extra spaces are added. (**BZ#1395564**)

- The left navigation column did not support vertical scrolling. When the browser viewport was less than 440 pixels tall and wider than 768 pixels the bottom left navigation link was not accessible. The new left navigation column markup supports vertical scrolling. Now, all left navigation links are accessible at all browser viewport sizes and zoom levels. (**BZ#1375134**)

- Previously, on iOS Safari, number inputs used the full keyboard rather than the number input. Now inputs that accept only numbers show the iOS number pad for easier entry. (**BZ#1470976**)

- Previously, some requests for templates in the web console could timeout or take a long time to complete over high latency network connections. This could cause an error when loading the **Add to Project** page. The web console can now load templates using much less data, which fixes the problem. (**BZ#1471033**)

- Clarifies help text on the Route creation and editing pages to make it clear that the CA certificates should be certificate chains. (**BZ#1471155**)

- A known bug in Internet Explorer resulted in the layout of pod charts overflowing their containers on the overview page. As a result, the pod charts looked mis-aligned in the UI. The fix involved increasing the specificity on some CSS declarations so that they only apply when they are needed, which is during a deployment when the pod charts are being animated. As a result, the pod charts appear correctly aligned in Internet Explorer. (**BZ#1473512**)

- A known bug in Internet Explorer resulted in the layout of catalog items taking up too much space. As a result, not all the catalog items were visible in Internet Explorer. The fix involved adding an additional CSS declaration as a workaround for IE. As a result, the catalog items now take up the correct space in IE. (**BZ#1473615**)

- The code was using an empty **`envFrom`** entry when creating/editing the environment variable, causing a validation failure when adding or editing an environment variable using **Deployment Configuration** page of the web console. The user would receive an error that the deployment configuration is invalid. The **`envFrom`** entry is now properly submitted and the user can add or edit environment variables from the web console. (**BZ#1502914**)

- Various errors were present in the source code that prevented Config maps were not available from the drop-down menu on the **Edit Deployment Config** page for pre and post-hooks when using **Add Value from Config Map or Secret**. These errors have been corrected. Config maps appear in the appropriate drop-downs. (**BZ#1502914**)

- Previously, secrets with null values would display incorrectly when values were revealed on the secret details page. Now the web console will correctly display the secret key as having no value. (**BZ#1510346**)

- Previously there was a quirk in the drag-and-drop behavior of the key value editor. While reordering an env var it might jump more than a single node at a time. This bug fix ensures that the drag-and-drop behavior will behave as expected. (**BZ#1428991**)

- On the project overview, the **Application** drop-down menu was incorrectly set to

`overflow:hidden`. As a result, when the application row is collapsed, the menu did not display fully. The **`overflow: hidden`** parameter has been removed and the menu is now fully visible. (**BZ#1460153**)

- Previously, deleting a service account would ignore the SAs namespace. This means that the delete action from the web UI could delete multiple service account rolebindings under the service account tab if service accounts from different namespaces had the same name. The delete action on the SA tab will now respect the namespace and only delete the specified SA rolebinding from the correct namespace. (**BZ#1507730**)

- The **Configuration** tab of the **Deployment** page in the web console was laid out in such a way that a large gap could appear when the right column contents were longer than the left column contents. The fix involved changing the layout markup so the gap does not appear. The result is there is no longer a gap between Volumes and Triggers when the right column content is longer than the left column content. (**BZ#1505255**)

Master

- Ansible installs with a caBundle on the service catalog API service resulting in a *500 Internal Server Error* on the product overview page in the web console. The installer was changed to install with **`insecureSkipTLSVerify`** flag set to **`true`**. As a result, the product overview page works as expected. (**BZ#1473523**)

- CronJobs are placed in batch/v2alpha1 group, whereas other batch resources are placed in batch/v1. Due to this fact, some API machinery does not handle multiversioning problems properly. The restmapper, which is responsible for matching resource with appropriate api group version to handle multi-versioned apis, was updated. Describing resources works as expected. (**BZ#1480453**)

- The installer was configured to watch specific resources that do not support watching. As a result, the ***/var/log/messages*** file was reporting errors and warnings related to the issue. The installer has been corrected to not watch these resources and the errors/warnings are not generated. (**BZ#1452206**)

- Creating project using project template does not use the substituted project name, but the namespace name. As a result, the user is not able to use parametrized name as a project name as the generated suffix or prefix might be dropped. The code was changed to allow the use of substituted project name when creating the namespace. (**BZ#1454535**)

- Node status information was getting rate limited during heavy traffic causing some nodes to fall into not ready status. The code was changed to use a separate connection for node healthiness. As a result, node status is reported without any problems. (**BZ#1464653**)

- Running multiple clusters in a single authorization zone in AWS requires resources be tagged. If the clusters are not tagged, the clusters will not work properly. The master controllers process will require a ClusterID on resources in order to run. Existing resources will need to be tagged manually. Multiple clusters in one az will work properly once tagged. (**BZ#1468579**)

- An upstream patch caused an error with the **`oc apply`** command. The patch deleted an element from an array (eg. env) and then reordered or modified another array (eg. volumeMounts). The **`kubectl apply`** fails with the _unable to find api field in struct Container for the json field "$setElementOrder/env". The algorithm was updated so that it continues operation under described condition. The **`oc apply`** works without any problems. (**BZ#1497325**)

Metrics

- When either a certificate within the chain at **`serviceaccount/ca.crt`** or any of the certificates within the provided truststore file contain white space after the **`BEGIN CERTIFICATE`** declaration, the Java keytool rejects the certificate with an error, causing Origin Metrics to fail to start. As a workaround, Origin Metrics will now attempt to remove the spaces before feeding the certificate to the Keytool. Admins should ensure their certificates don’t contain such spaces. (**BZ#1503450**)

- When deleting a large number of pods, the **hawkular-metrics** pod log reports *Pool is busy* errors. The condition was fixed upstream in Cassandra and clusters with a large number of pods should not report the *Pool is busy* error. (**BZ#1451209**)

- When opening the metrics page in a disconnected environment, Hawkular attempted to connect to external web sites, such asfonts.googleapis.com. Because the cluster cannot connect to Internet, the metrics page loaded slowly. Changes were made upstream so that Hawkular does not attempt to connect to external web sites when there is no access to the Internet. As a result, in a disconnected environment, the metrics page loads properly. (**BZ#1466403**)

- In Cassandra, it is possible that new generation objects (with the **`-Xmn`** flag) can exceed the maximum size of the Java memory heap (with the **`-Xmx`** flag). If that happens, the JVM will log a warning at start up, but Cassandra still starts. The code was changed to set the size of new generation objects at ¼ of the maximum heap size. (**BZ#1471239**)

- Cassandra metrics would not start up if the commit log exceeded the limit applied to the log. An out-of-memory (OOM) condition would cause metrics to constantly start and stop. The commit log size is now based on total available memory. Also, log compression is no longer used, which will reduce the demand on resources. As a result, large logs should not affect metrics operation. (**BZ#1473013**)

Networking

- When changes are made to software defined network (SDN) plugin, the master controller will fail to start when there are headless services in the cluster. As a result, when initializing OpenShift Container Platform, SDN fails to allow a nil service IP and OpenShift Container Platform was unable to start. The code was changed to allow nil as a valid value of **`srv.Spec.ClusterIP`**. OpenShift Container Platform SDN properly starts after changing network with headless service. (**BZ#1451881**)

- The nodes local IP address is not part of the Open vSwitch (OVS) rules. If you deny 0.0.0.0/0 and allow a DNS name in the egress network policy, the node will not be able to reach that allowed address because DNS name resolution is blocked Adding the local node IP to the ovs allowed rule so that the name resolution will not be blocked. Also adding a note to the docs for the case when dns resolution does not happen on the node. OpenShift Container Platform can successfully block 0.0.0.0/0 as a **`cidrSelector`** and allow specific DNS names through. (**BZ#1458849**)

- If the **`service network restart`** command is executed on a machine while the OpenShift Container Platform node process is running, a **`stop()`** function properly disables IP forwarding. However, the **`start()`** function was not re-enabling it. The code was changed to persist IP forwarding on nodes during network restarts. (**BZ#1477716**)

- While upgrading nodes, if any invalid network CIDRs are detected, nodes might be unable to upgrade and will fail. The code was changed to not fail with invalid CIDRs. (**BZ#1506017**)

- The Kubernetes CNI (Container Network Interface) plug-in generates errors if **`hostNetwork=true`** is configured for pods. This issue has been fixed. (**BZ#1507257**)

- Because of upstream issues in Kubernetes, vSphere had networking problems when used with OpenShift Container Platform. The periodic resync of Kubernetes into OpenShift Container Platform included the required changes. vSphere now works correctly. (**BZ#1433236**)

- Because of changes with upstream Kubernetes, the **`oc adm join-projects`**, **`oc adm isolate-projects`** and other commands that depend on the pod update operation will not work. The code was changed to fetch some required elements from the Container Runtime Interface (CRI) directly. As a result, the pod update operation works correctly and the commands work as expected. (**BZ#1453190**)

- Because of default authorization, project administrators (standard user) were not able to manage network policies for their own projects. Changes to the code now allow project admins to create, delete, list the network polices in their own projects. (**BZ#1461208**)

- An invalid HostSubnet could not be fixed. As a result, if a node with an invalid HostSubnet is restarted, the node assigned to the HostSubnet, would fail to start. The code has been changed to allow an invalid HostSubnet to be changed, using commands such as **`oc edit hostsubnet`**. (**BZ#1466239**)

- Adding an IPv6 address to a host subnet as an egress resulted in a panic error. The code has been changed to better handle IPv6 addresses with a meaningful error message. (**BZ#1500664**)

- Using ipfailover when a node fails ensures that a second node receives traffic. Previously, traffic went back to the first node once it is back up, potentially causing traffic imbalance. Now, using the **`--preemption-strategy="nopreempt"`** option, allows the administrator to control the default strategy, meaning that the strategy to switch to a higher priority node is suppressed. (**BZ#1465987**)

- A log message similar to the following was repeatedly appearing:

  ```
  LoadBalancerRR: Removing endpoints for ops-health-monitoring/pull-
  07062050z-ie:8080-tcp
  ```

  This caused the logs to be filled with information not deemed important. The message has been hidden from the logs. (**BZ#1468420**)

- Previously, the image for the default network diagnostics pod was mismatched, causing the diagnostics to fail. The image checking has been fixed, and the network diagnostics works without errors. (**BZ#1481147**)

- Previously, conntrack entries for UDP traffic were not erased when an endpoint was added for a service that previously had no endpoints. This meant that the system could end up incorrectly caching a rule that would cause traffic to that service be dropped rather than being send to the new endpoint. The relevant conntrack entries have been changed to be deleted at the right time, meaning that the UDP services work correctly when endpoints are added and removed. (**BZ#1487438**)

Pod

- Previously, network debug tests were showing errors regarding not being able to read stats from a changing pod. This was because, even though the container process had exited, but the cgroup wasn’t removed, leading to a Docker container with no tasks. The log spam has been reduced. (**BZ#1328913**)

- Because of an outdated Go format, kubemarl-scale was consistently failing. The version of Golang was updated, stopping the failures. (**BZ#1454239**)

- Previously, the HPA V1 was unable to get the metrics from the resource CPU. This was due to the custom setup of the HPA controller changing. The settings have been restored. (**BZ#1458663**)

- Previously, multi-node environments produced “Failed to watch” errors. This was because the controller didn’t have permission to watch resources, which meant its behaviour was to retry every second by default. The controller has been given the permission to watch resources. (**BZ#1465361**)

- Previously, the OpenShift master failed to start when using Openstack integration without Neutron LBaaS, which is not available in OpenShift. The issue now gives a warning instead of a failure, which mean the master will start successfully even if the LBaaS is not available. (**BZ#1465722**)

- Previously, project volumes were not included in security context constraints, meaning that pods could not be used with projected volumes. The projected volumes have been added to the correct SCCs, and the projected volumes can be used as expected. (**BZ#1448816**)

- Init containers with resource requests or limits were producing error messages. This was due to a mismatch in the sum of a pod’s container resources, resulting in the parent cgroup choosing the incorrect resource. The issue has been fixed upstream and the correct resources are being chosen. (**BZ#1459826**)

- Previously, when a deployment configuration was created without any memory information when quota restrictions were in place, no error message would appear. The expected results were a “FailedCreate” event, much like with replication controllers. The “FailedCreate” event now appears when the pod immediately fails. (**BZ#1465801**)

- A design limitation in previous versions does not account for memory-backed volumes against the pod’s cumulative memory limit. So, it is possible for a user to exhaust memory on the node by creating a large file in an memory-backed volume, regardless of the memory limit. Now, pod-level cgroups have been added to, among other things, enforce limits on memory-backed volumes, resulting in memory-backed volume sizes now being bound by cumulative pod memory limits. (**BZ#1422049**)

- Previously, upgrading to 3.4 gave a “insufficient pods” error. This was due to a change in configuration from a **`max-pods`** variable to the smaller of 250 or 10 pods per core. The upgrade broke installations with fewer pods. The change has been made so that the **`max-pods`** variable has become the limiting variable. (**BZ#1430484**)

- Previously, error messages in the status field of failed builds said “error” instead of an actual error message. This was because the status was showing the message from the Docker daemon returning the failed pod message. The message now returns a more helpful error message. (**BZ#1449820**)

- Previously, registry pods were occasionally reporting liveness and readiness probe failures with the message **`http2: no cached connection was available`**. This was due to an upstream issue where the liveness and readiness probes get in the way of each other. The problem has been fixed upstream, and updated for OpenShift Container Platform version 3.7. (**BZ#1454858**)

- Large clusters with a large amount of HPAs or unhealthy pods sent a large number of events if an object was unable to reach its desired state. This bug fix updates the event client to protect against spamming master components. As a result, this controls traffic to the masters and reduces writed to etcd. (**BZ#1466933**)

- For all resources other than pod or PVCs, the quota controller would make a LIST call per namespace to determine current usage counts. This caused quota recalculation to take an extended period of time. This bug fix reduces LIST calls made by the resource quota controller by using shared informer caches. As a result, LIST operations made to the master were reduced and information was pulled from a shared cache in the controller. (**BZ#1473370**)

- Previously, users were not able to to look up PVC information for the Drupal database without receiving scheduler log spam. This bug fix prevents unnecessary logging of a harmless error from a PVC-related scheduler predicate. (**BZ#1475558**)

- Previously, messages originating from the AWS SDK were causing partial log entries due to new lines in the message itself. Error messages are now properly quoted so all messages are (**BZ#1462445**)

Routing

- Previously, the help information included a redundant example. This bug fix removed the redundant example. As a result, the help information is now more concise. (**BZ#1440620**)

- Previously, the code path automatically prepended the partition name to the vserver name. If the vserver was in a path of length more than 1, then the path was lost because only the partition name was prepended. This bug fix prepends the entire path of vserver instead of just concatenating the partition name and vserver name. (**BZ#1465304**)

- Previously, if you had a router of a previous version of OpenShift Container Platform a 403 http status resulted when the router stats were accessed without credentials. This web browser did not prompt the user for a password so the stats were inaccessible. The code has been updated to return a 403 when no credentials are passed and the browser now prompts the user for a password, so the router stats are visible in a web browser. (**BZ#1467257**)

- Previously, the IP failover keepalived image did not support IPV6 addresses or ranges, as well as IP address validation. Adding IPV6 addresses to the `oc adm ipfailover` command resulted in a new vrrp section pertaining to the wrong address. The code has been updated, and inputting invalid IPV4 and IPV6 addresses now return an error as expected. (**BZ#1459960**)

- Previously, the x-forwarded header and its associated information, displayed the IPV6 form in IPV4 form. The `ROUTER_IP_V4_V6_MODE` environment variable has been created to control which form is displayed. (**BZ#1471255**)

- Previously, the locking was overly broad, causing events to not be processed while an HAProxy reload was happening. This meant that route changes would take hours to process. The locking has been made more fine-grained, so that events can be processed in parallel. And changes are now processed within the time of two reloads of the router. (**BZ#1471899**)

- An error in the router code caused by a missing locking around a router data structure was causing errors causing the router pod to occasionally crash and restart. The locking has been fixed, and the router now works as expected. (**BZ#1473031**)

- When running the `oc adm router --expose-metrics` command, the router deployment failed because the generated deployment configuration object was not compatible. This was due to a background change upstream. A change has been made with the `oc adm router` command, and the command can now handle `--expose-metrics`. (**BZ#1488954**)

- Previously, multiple service catalog objects named “default” were not a problem, but a change made them all top level. This bug fixes the object names to be unique. (**BZ#1420543**)

Service Broker

- Previously, a fresh installation using the **`openshift-ansible`** method and with a **`service-catalog`** resulted in the service class being empty, resulting in the stage registry giving a bad response. The administrator would need to see the ASB logs and trigger a manual bootstrap. Now, if the bootstrap fails, the broker fails, and the kubelet retries the process until it works correctly. (**BZ#1468173**)

- This bug fixes running the **`service-catalog`** binaries for the apiserver and controller manager when used with the **`--version`** option, which previously reported **`UNKNOWN`**, but now reports the correct value. (**BZ#1476134**, **BZ#1475251**)

- Previously, when deleting a namespace, the Ansible Service Broker (ASB) attempted to execute deprovision playbook actions using a namespace in a "terminating" state. This led to the APB actions being rejected, because of the namespace terminating. As a result, deprovision fails, and both the APB deprovision sandbox and target namespace were not deleted. Now, instead of executing APB actions on namespace deletion, the records of the services to be deprovisioned are cleaned up, allowing kubernetes to delete the resources normally, meaning the target namespace is properly deleted by Kubernetes. (**BZ#1476173**)

- The error message returned when a user does not have permission to modify a TemplateInstance is updated. (**BZ#1460145**)

- Previously, only one annotation returned when both *expose* and *base64-expose* annotations were defined in template (per bind request). This issue is fixed in the latest release. (**BZ#1463570**)

- Previously, Ansible Playbook Bundles (APB) that have been removed from their container catalog, appeared in Ansible Service Broker (ASB) as valid options even after **bootstrap** was performed. This issue is fixed now. (**BZ#1463798**)

- Previously, there were inconsistency between the serviceclass and the server-broker. After creating a broker, the controller-manager only fetched the catalog once. This resulted in inability to updates the serviceclass unless the broker was recreated. This is fixed now. (**BZ#1469448**)

- Previously, the Ansible service broker would fail on provisioning because of incorrect permissions. This is now fixed and Ansible service broker now has the required permissions for creating new namespaces and dynamic service account in these new namespace to run APBs. (**BZ#1469485**)

- The **`oc version`** command did not get OpenShift version against the ansible deployed service catalog environment. The version information is added the command now reports correct information. (**BZ#1471717**)

- Previously, when the Ansible Service Broker started it could not communicate to the configured registry, and therefore got no information about APBs. This was because of a missing setting in the ansible service broker configuration. The **`broker: bootstrap_on_startup: true`** setting is now added in the configuration which resolves this issue. (**BZ#1471973**)

- Previously, the ansible service broker container would fail if the dockerhub credentials were not supplied because the encryption script required them. It is now reconfigured to use RHCC adapter and the dockerhub credentials are optional. (**BZ#1464222**)

- Previously, bad data was being returned from the bootstrapped registry. This was because the broker failed to bootstrap and it used to error out due to a null pointer de-reference. The broker now has logic to avoid de-referencing null pointers if the data is corrupted. This issue is now resolved and the broker skips image with bad data and continues with next one. (**BZ#1467905**)

- The Service Broker Installer was setting incorrect configuration values for **launchapbonbind**, this is fixed and configuration value is now set as **launch_apb_on_bind**. (**BZ#1467948**)

- The role for Service Accounts used by the Ansible Service Broker is updated. The Broker runs under **asb** service account set to **admin** through a *ClusterRoleBinding* and APBs run under a temporary service account granted **edit** through a *RoleBinding* in the target namespace. (**BZ#1470824**)

Storage

- Creating a new persistent volume claim (PVC) using OpenStack Cinder storageclass resulted in the PVC being stuck in **Pending** state. This bug fix re-configured the cloud provider openstack.conf to use OpenStack Keystone V3. As a result, dynamic provisioning of new Cinder volumes works as documented. (**BZ#1491331**)

- Previously, the Gophercloud library used by OpenShift to communicate with the OpenStack API did not accept HTTP status 300 in pagination. It was not possible to dynamically provision OpenStack Cinder volumes. This bug fix upgrades the Gophercloud library in the OpenShift vendor directory. As a result, dynamic provisioning of new Cinder volumes works as documented. (**BZ#1490768**)

- Previously, the default bootstrap policy allowed basic users to “get” storage classes, but not “list” storage classes. Basic users would receive an error message after issuing the **oc get storagelcass storageclass_name** command. This bug fix modified the bootstrap policy. As a result, basic users can now issue the **oc get storagelcass storageclass_name** command to receive specific storage classes. (**BZ#1449608**)

- Previously, the lack of cloud provider configuration in the admission plug-in caused persistent volume (PV) creation to fail when attempting to create the PV in a zone other than master. This bug fix enables static PV provisioning in multizone environments. As a result, users can now statically provision PVs in zones other than master. (**BZ#1454601**)

- Previously, when creating storage classes, users could not specify the **fstype**. This bug fix allows specifying the desired **fstype** when dynamically provisioning volumes with storage classes. As a result, storage classes now support file system configuration when creating dynamically provisioned volumes. (**BZ#1469001**)

- Previously, it was not possible to dynamically provision ScaleIO volumes if the ScaleIO volume plug-in was not enabled. This bug fix enables the ScaleIO volume plug-in in OpenShift Container Platform 3.7. As a result, it is now possible to dynamically provision ScaleIO volumes. (**BZ#1482274**)

- When trying to mount/unmount, the FlexVolume plug-in’s file system previously assumed that SELinux was supported. This assumption instructed docker to relabel the volume. If the FlexVolume plugin’s file system did not support file system relabeling, the container using the FlexVolume would fail to start. This bug fix added the **selinuxRelabel** capability, which allows FlexVolume plug-ins to report in their **init** call. As a result, FlexVolume plug-ins can now be configured to opt out of SELinux relabeling. (**BZ#1484899**)

Templates

- Previously, the service catalog could not provide authentication when invoking the template service broker, which meant the template service broker API had to allow calls from unauthenticated clients. This bug fix allows the service catalog to use proper authentication to invoke the template service broker when issuing the `oc cluster up` command to run both. As a result, the template service broker APIs will now be secured, and will only be invokable by the service catalog (or another client with appropriate credentials). (**BZ#1470628**)

Upgrade

- Previously, the master node upgrade took more disk space than was initially estimated. This caused the etcd member to report a `no space left on device` error message. This bug fix increased the estimation of disk space needed before the master node upgrade can start. As a result, a master node is properly upgraded with enough disk space left after the upgrade finishes. (**BZ#1489182**)

- Previously, the upgrade playbooks incorrectly overwrote nondefault `admissionConfig` parameters while setting specific values required of the upgrade process. This bug fix removed this task as it is no longer necessary after upgrading from OpenShift Container Platform 3.4 to OpenShift Container Platform 3.5. (**BZ#1486054**)

- Previously, the etcd v3 data migrated prior to the first etcd v2 snapshot being written. Without a v2 snapshot, the v3 data was not propagated properly to the remaining etcd members, which resulted in a loss of some v3 data. This bug fix checks to see if there is at least one v2 snapshot before etcd data migration proceeds. As a result, etcd v3 data is now properly distributed among all etcd members. (**BZ#1501752**)

- When trying to upgrade OpenShift Container Platform with dedicated etcd from v3.6 to v3.7, the upgrade failed at the [Stop atomic-openshift-master-controllers] task due to the wrong hosts group. This bug fix corrected the host group to specify the masters group for controller restart. As a result, the upgrade now succeeds. (**BZ#1504515**)

- Previously, if Ansible tags were used to evaluate some of the tasks in a set of playbooks, the conditional for including a task file was not properly evaluated. This caused the upgrade to fail. This bug fix allows the conditional to evaluate properly and skip running the task. (**BZ#1464025**)

- Ansible playbooks now exit immediately when health checks fail. Previously, in some instances, a host failure would not result in the playbook exiting during failed health checks. This bug fix sets the `any_errors_fatal` play option to `true`, ensuring that the playbook exits as expected. (**BZ#1484324**)

- Upgrades that made use of system reboots to restart services may have failed if hosts took longer than 5 minutes to restart. This bug fix increases the timeout to 10 minutes. As a result, the shutdown process is now faster. (**BZ#1455836**)

2.6. TECHNOLOGY PREVIEW FEATURES

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

Technology Preview Features Support Scope

The following new features are now available in Technology Preview:

- Prometheus Cluster Monitoring

- Local Storage Persistent Volumes

- CRI-O

- Tenant-driven Storage Snapshotting
- CLI Plug-ins
- Enabling Static IPs for External Project Traffic

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

- Service Catalog
- Template Service Broker
- OpenShift Ansible Broker
- Ansible Playbook Bundles
- Network Policy
- Initial Experience
- Add from Catalog and Add to Project
- Search Catalog
- Automated installation of CloudForms Inside OpenShift

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

- Cron Jobs (formerly called Scheduled Jobs)
- Kubernetes Deployments Support
- `StatefulSets`, formerly known as `PetSets`
- Require Explicit Quota to Consume a Resource
- Mount Options
- Installation of etcd, Docker Daemon, and Ansible Installer as System Containers
- Running OpenShift Installer as a System Container
- Integrated Approach to Adding Hawkular OpenShift Agent
- Bind in Context
- `mux`

2.7. KNOWN ISSUES

- When users are redirected to an external domain after successfully logging in, users should only be redirected to specific URLs. This is a known issue and will be fixed in OpenShift Container Platform 3.9. (**BZ1525538**)
- The installer can not deploy system container-based installations when the specified registry

requires authentication credentials in order to pull the required system container images. The fix for this depends on an update to the `atomic` command, which will be updated after OpenShift Container Platform 3.7 GA. (**BZ#1505744**)

- A OpenShift Container Platform 3.7 master will return an unstructured response instead of structured JSON when an action is forbidden. This is a known issue and will be fixed in OpenShift Container Platform 3.8.
- The volume snapshot Technology Preview feature may not be available to non-administrator users by default due to API RBAC settings. When the volume snapshot controller and provisioner are installed and run, the cluster administrator needs to configure the API access to the VolumeSnapshot objects by creating roles and cluster roles, then assigning them to the desired users or user groups. (**BZ#1502945**)
- OpenShift Container Platform is unable to list known health checks. (**BZ#1509157**)
- The current format of audit logs is difficult to consume. Some keys are duplicates and some are misleading in that they match wrong keys in the linux-audit dictionary. (**BZ#1496176**)
- Limit NFS provisioning of the Quick Installer to hosts that are part of the cluster. If a host is outside of the cluster, where Ansible will not run, use the Advanced Installation method.

2.8. ASYNCHRONOUS ERRATA UPDATES

Security, bug fix, and enhancement updates for OpenShift Container Platform 3.7 are released as asynchronous errata through the Red Hat Network. All OpenShift Container Platform 3.7 errata is available on the Red Hat Customer Portal. See the OpenShift Container Platform Life Cycle for more information about asynchronous errata.

Red Hat Customer Portal users can enable errata notifications in the account settings for Red Hat Subscription Management (RHSM). When errata notifications are enabled, users are notified via email whenever new errata relevant to their registered systems are released.

NOTE

Red Hat Customer Portal user accounts must have systems registered and consuming OpenShift Container Platform entitlements for OpenShift Container Platform errata notification emails to generate.

This section will continue to be updated over time to provide notes on enhancements and bug fixes for future asynchronous errata releases of OpenShift Container Platform 3.7. Versioned asynchronous releases, for example with the form OpenShift Container Platform 3.7.z, will be detailed in subsections. In addition, releases in which the errata text cannot fit in the space provided by the advisory will be detailed in subsections that follow.

IMPORTANT

For any OpenShift Container Platform release, always review the instructions on upgrading your cluster properly.

2.8.1. RHBA-2017:3464 - OpenShift Container Platform 3.7.14 Bug Fix and Enhancement Update

Issued: 2017-12-18

OpenShift Container Platform release 3.7.14 is now available. The packages and bug fixes included in the update are documented in the RHBA-2017:3464 advisory. The container images included in the update are provided by the RHBA-2017:3465 advisory.

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

2.8.1.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.1.2. Bug Fixes

- The `_source` was not disabled for metrics records being stored in Elasticsearch. Therefore, the records were taking up much more resources (CPU, RAM, and disk space) than necessary. Completely disable the `_source` for `project.ovirt-metrics*` records. With this bug fix, metrics records are much smaller and require fewer resources. (**BZ#1494912**)

- `docker_image_availability` casted `openshift_docker_additional_registries` to a list using the `list()` function. If a string was returned (only a single registry added) the result would be the string split up by component characters. Force a string result from get_var to be placed inside a list. If the result is anything but a string, the original `list()` function is called on the result. With this bug fix, docker_image_availability passes when it should. (**BZ#1497274**)

- The Ansible role accepted the CPU request and limit without additional evaluation. Creating the resource would fail if the request was greater than the limit. Evaluate the facts, inventory, and defaults and lower the request to the limit value if needed. With this bug fix, the resource is created without issue. (**BZ#1506073**)

- Logging deployments were not maintaining the association of deployment configuration (DC) to persistent volume claim (PVC) for Elasticsearch, causing deployment failures. This bug fix addresses the regression introduced with 3.7 regarding reuse of the PVC that was previously specified within a DC. (**BZ#1511110**)

- When upgrading to a new version of logging, if the deployment does not use a separate OPS logging cluster and there is no `@OUTPUT` label in the ***fluent.conf***, fluentd does not know how to route the operations logs. The operations logs are not logged. The user will see no new operations (`.operations.*`) logs after the upgrade. Change the fluentd configuration to correctly handle the case both with and without the `@OUTPUT` label, and for the OPS and non-OPS cases. With this bug fix, operations logs flow uninterrupted after upgrade. (**BZ#1511719**)

- The OpenShift Docker builder invoked the Docker build API without the `ForceRmTemp` flag. Containers from failed builds remained on the node where the build ran. These containers are not recognized by the kubelet for garbage collection and are, therefore, accumulated until the node runs out of space. This bug fix modified the Docker build API call from the OpenShift Docker builder to force the removal of temporary containers. Failed containers no longer remain on the node where a Docker build ran. (**BZ#1512679**)

- On slow network connections, the **Next** button could sometimes take several secrets to enable when opening the create from builder image dialog on the OpenShift web console catalog landing page with the service catalog enabled. This problem has been fixed, and the **Next** button should now be enabled immediately when the dialog is displayed. (**BZ#1512858**)

- This bug fix brings new features to the GlusterFS deployments, specifically the ability to provision gluster-block and gluster-s3 volumes. (**BZ#1512978**)

- The **`assign-macvlan`** pod annotation can now take the name of a network interface rather than just **`true`** or **`false`**. The egress-routers can now be run on a non-default network interface. (**BZ#1513767**)

- The configuration management for build defaults attempts to remove environment variables that were previously defined but have since been removed from the configuration. In situations where no environment variables have been configured, this was failing because the **`env`** key did not exist. The process has now been updated to skip the cleanup when the **`env`** key does not exist. (**BZ#1515457**)

- An error in the gluster-block installation playbooks is now corrected. (**BZ#1516530**)

- Previously, deploying gluster-block storage would fail due to a missing file. The file name has been corrected, ensuring proper deployment. (**BZ#1517225**)

- Jenkins images that were not backwards compatible to older clusters were pushed as **`latest`**. Clusters referencing Jenkins master and slave images via the **`latest`** tag picked up images that might not be compatible with the cluster, due to the version of the **`oc`** client binary embedded in the Jenkins images. This results in **`oc`** client operations failing when run from within the Jenkins (and Jenkins slave) containers, against the cluster. With this bug fix, the **`latest`** tag now points to a compatible image that will work with older clusters. Older clusters referencing the **`latest`** Jenkins images now work properly. (**BZ#1518501**)

- Changes in the destination for the RPM installation of the list of plug-ins included in the **`openshift-jenkins-rhel7`** image changed between versions 3.6 and 3.7. After an upgrade of the **`openshift-jenkins-rhel76`** image from 3.6 to 3.7 when persistent volumes are maintained, Jenkins would not start successfully, as it could not read in the needed plug-in ***jpi/hpi*** files. With this bug fix, updates are made to the 3.7 **`openshift-jenkins-rhel7`** image so that additional links are established so that the real ***jpi/hpi*** files can be found. Jenkins in an OpenShift Container Platform pod with PVs will now appear successfully after a 3.6 to 3.7 upgrade. (**BZ#1519034**)

2.8.2. RHBA-2018:0076 - OpenShift Container Platform 3.7.14-9 Images Update

Issued: 2018-01-10

OpenShift Container Platform release 3.7.14-9 is now available. The list of container images included in the update are documented in the RHBA-2018:0076 advisory.

The container images in this release have been updated using the latest base images.

2.8.2.1. Images

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

```
openshift3/jenkins-slave-nodejs-rhel7:v3.7.14-10
openshift3/logging-kibana:v3.7.14-9
openshift3/node:v3.7.14-9
openshift3/openvswitch:v3.7.14-10
```

2.8.2.2. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.5 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.3. RHBA-2018:0113 - OpenShift Container Platform 3.7.23 Bug Fix and Enhancement Update

Issued: 2018-01-22

OpenShift Container Platform release 3.7.23 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:0113 advisory. The container images included in the update are provided by the RHBA-2018:0114 advisory.

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

2.8.3.1. Bug Fixes

- A regression bug was reported whereby source-to-image builds would fail if the source repository filesystem contained a broken symlink (pointing to a non-existent item). This was resolved with this bug fix. (**BZ#1519822**)

- Categories like **all** were moved to the server, but some of them were only moved after the upstream cut for the rebase, causing an incomplete list of resources. Therefore, some resources could not be found in **`oc get all`** and some other **`oc get`** calls. With this bug fix, the remaining upstream commits were picked to include all needed resources and **`oc get all`** and the other problematic calls were fixed. (**BZ#1515878**)

- Node selectors were incorrectly set on template service broker daemonset object. Consequently, the looping failed the deployment of template service broker pods and there was excessive CPU usage on master and nodes. The node selectors are now set correctly on the template service broker daemonset object and the template service broker pods now deploy correctly. (**BZ#1524219**)

- Fluentd fails to properly process messages when it is unable to determine the namespace and pod UUIDs.The logging pipeline outputs a lot of messages and sometimes blocks log flow to Elasticsearch. Check for the missing fields and orphan the record if needed. With this bug fix, logs continue to flow and orphaned records end up in an orphaned namespace. (**BZ#1494612**)

- Elasticsearch clusters that take a long time recovering data do not reach the **YELLOW** state fast enough. The OpenShift Container Platform cluster restarts the pod because the readiness probe fails, which starts the Elasticsearch node recovery again. Check only for the Elasticsearch cluster to be listening on the desired port. The OpenShift Container Platform cluster does not terminate the Elasticsearch node early, which allows it to complete its recovery. The cluster may be in the **RED** state at this time, but is able to accept queries and writes. (**BZ#1510697**)

- When trying to alias an index that does not exist, the bulk alias operation failed. Only alias ***.operations*** if there is at least one ***.operations*** index. As a result, there will be an alias for all indices. (**BZ#1519705**)

- The **remote-syslog** plug-in in fluentd takes a configuration parameter **`tag_key`** to use the field specified in **`tag_key`** from the record to set the syslog key. When a field specified in **`tag_key`** does not exist, it caused a Ruby exception, which was not caught. With this bug fix, the field no longer exists and **`tag_key`** is ignored and the default tag is used. (**BZ#1519213**)

- There was a logic error in the fluentd startup script. When an ops cluster was first disabled then enabled, the proper ops configuration file was not enabled. As a result, Sub-configuration files starting with **`output-ops-extra-`** did not have a chance to be called from the ops configuration file. The logic error s now fixed When an ops cluster is first disabled then enabled, the proper ops configuration file is enabled and its sub-configuration files are also enabled. (**BZ#1519679**)

- The annotation to identify the proper Kibana instance to use for ops namespaces was being set regardless of if an ops logging cluster is used or not. Users were directed to non-existent service from the web console when trying to view operations logs. For existing deployments, manually run **`oc annotate ns $NAMESPACE openshift.io/logging.ui.hostname-`** for each affected namespace. New deployments will only have this annotation set by the installer if the **`openshift_logging_use_ops`** variable is set to **`true`**. With this bug fix, users will be directed to the correct version of Kibana when viewing ops logs from the web console. (**BZ#1519808**)

- Previously, when importing a template as YAML in the web console, then clicking **Back** in the wizard, the **Next** button would stop working. The problem has been fixed so that the **Next** button works correctly after clicking **Back** in the **Import YAML** dialog. (**BZ#1526215**)

- The Kubernetes resource quota controller had a fatal a race condition. Therefore, the master controller process occasionally crashes, writes a stack dump, and restarts. With this bug fix, the race condition is resolved and the crash no longer occurs. (**BZ#1519277**)

- When a Network Egress DNS policy was used, a bug may have prevented further correct operation of the proxy, resulting in new pods not handling service requests. That bug is fixed and Egress DNS policies can now be used without triggering this bug. (**BZ#1502602**)

- A bug in the node container garbage collection and network setup prevented pod sandboxes from being properly garbage collected. Nodes could exhaust the available pool of pod IP addresses, especially if they are restarted and/or containers were removed while the node was not running. Nodes now properly garbage collect and tear down pod sandboxes, ensuring that IP addresses are released to the pool for subsequent re-use. Newly installed nodes should no longer experience IP address exhaustion due to pod sandbox teardown errors. Upgraded nodes should remove all files in ***/var/lib/cni/networks/openshift-sdn/*** during the upgrade, or after upgrade when no pods are running on the node. (**BZ#1516782**)

- This bug fix corrects an interaction between runc and systemd where the sandbox creation fails when the pod specifies CPU limit in 1000s of millicores and the last digit is not 0. (**BZ#1509467**)

- This bug fix corrects an issue where **`oc logs`** exits with error **unexpected stream type**. (**BZ#1521026**)

- When running the etcd v2 to v3 migration playbooks as included in the OpenShift Container Platform 3.7 release, the playbooks incorrectly assumed that all services were HA services (**`atomic-openshift-master-api`** and **`atomic-openshift-master-controllers`** rather than **`atomic-openshift-master`**), which is the norm on version 3.7. However, the migration playbooks would be executed prior to upgrading to version 3.7, so this was incorrect. The migration playbooks have been updated to start and stop the correct services ensuring proper migration. (**BZ#1523814**)

2.8.3.2. Enhancements

- Elasticsearch **`deploymentconfigs`** are now modified to disable OpenShift Container Platform rollback behavior. Upgrades of the logging stack that do not result in the readiness probe succeeding are rolled back to the last previous successful deployment. This can result in the deployed instances being out-of-date with their configuration and old images. This change will make it so the deployments will not be rolled back and operators can manually intervene without the image and configuration mismatches. (**BZ#1519622**)

- An ***.operations*** index-mapping in a non-ops Elasticsearch cluster is no longer displayed because operations indices will never exist in a non-ops Elasticsearch cluster. (**BZ#1519706**)

2.8.3.3. Images

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

```
openshift3/apb-base:v3.7.23-3
openshift3/container-engine:v3.7.23-3
openshift3/cri-o:v3.7.23-3
openshift3/image-inspector:v3.7.23-3
openshift3/jenkins-2-rhel7:v3.7.23-3
openshift3/jenkins-slave-base-rhel7:v3.7.23-3
openshift3/jenkins-slave-maven-rhel7:v3.7.23-3
openshift3/jenkins-slave-nodejs-rhel7:v3.7.23-3
openshift3/local-storage-provisioner:v3.7.23-3
openshift3/logging-auth-proxy:v3.7.23-3
openshift3/logging-curator:v3.7.23-3
openshift3/logging-elasticsearch:v3.7.23-3
openshift3/logging-eventrouter:v3.7.23-3
openshift3/logging-fluentd:v3.7.23-3
openshift3/logging-kibana:v3.7.23-3
openshift3/mariadb-apb:v3.7.23-3
openshift3/mediawiki-123:v3.7.23-3
openshift3/mediawiki-apb:v3.7.23-3
openshift3/metrics-cassandra:v3.7.23-3
openshift3/metrics-hawkular-metrics:v3.7.23-3
openshift3/metrics-hawkular-openshift-agent:v3.7.23-3
openshift3/metrics-heapster:v3.7.23-3
openshift3/mysql-apb:v3.7.23-3
openshift3/node:v3.7.23-3
openshift3/oauth-proxy:v3.7.23-3
openshift3/openvswitch:v3.7.23-3
openshift3/ose-ansible-service-broker:v3.7.23-3
openshift3/ose-ansible:v3.7.23-3
openshift3/ose-base:v3.7.23-3
openshift3/ose-cluster-capacity:v3.7.23-3
openshift3/ose-deployer:v3.7.23-3
openshift3/ose-docker-builder:v3.7.23-3
openshift3/ose-docker-registry:v3.7.23-3
openshift3/ose-egress-http-proxy:v3.7.23-3
openshift3/ose-egress-router:v3.7.23-3
openshift3/ose-f5-router:v3.7.23-3
openshift3/ose-haproxy-router:v3.7.23-3
openshift3/ose-keepalived-ipfailover:v3.7.23-3
openshift3/ose-pod:v3.7.23-3
openshift3/ose-recycler:v3.7.23-3
openshift3/ose-service-catalog:v3.7.23-3
openshift3/ose-sti-builder:v3.7.23-3
openshift3/ose-template-service-broker:v3.7.23-3
openshift3/ose:v3.7.23-3
```

```
openshift3/postgresql-apb:v3.7.23-3
openshift3/prometheus-alert-buffer:v3.7.23-3
openshift3/prometheus-alertmanager:v3.7.23-3
openshift3/prometheus:v3.7.23-3
openshift3/registry-console:v3.7.23-3
openshift3/snapshot-controller:v3.7.23-3
openshift3/snapshot-provisioner:v3.7.23-3
```

2.8.3.4. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.4. RHBA-2018:0636 - OpenShift Container Platform 3.7.42-2 Bug Fix and Enhancement Update

Issued: 2018-04-05

OpenShift Container Platform release 3.7.42-2 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:0636 advisory. The list of container images included in the update are documented in RHBA-2018:0637 advisory.

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

2.8.4.1. Bug Fixes

- When a pod is admitted by "nonroot" SCC, the **`securityContext.runAsUser`** parameter is not set, and the user is able to provide an arbitrary image. This made it possible to escalate permissions within a container. This bug fixes the check in the kubelet, and now such a container will fail during its creation and will not run. (**BZ#1518889**)

- Image validation used to validate old image objects, and the image signature import controller used to generate such an image. This could cause invalid images to be pushed to etcd. This bug fix changes the validation to validate new image objects and introduces logic to fix some invalid images. As a result, the controller no longer generates invalid images and it is no longer possible to upload an invalid image object. (**BZ#1557607**)

- With the image registry, concurrent writes to the cache could cause a panic to occur. This bug fix protects writes to the cache with mutex. As a result, the cache is safe to use concurrently. (**BZ#1549917**)

- Previously, when scaling up containerized masters, the registered **`install_result`** was not available when the install task was skipped for containerized hosts, causing scale up to fail. This bug fix defaults the value so that scale up can succeed. (**BZ#1511357**)

- The **`openshift-ansible docker_image_availability`** check did not utilize the **`osm_use_cockpit=false`** variable. This check could fail because the **registry-console** (Cockpit) image was not available, even if it would never be used. The check now consults this variable. As a result, if **`osm_use_cockpit=false`** is set, then the **openshift-ansibledocker_image_availability** check will not report a missing **registry-console** image. (**BZ#1511616**)

- Amazon EC2 C5 instances use different identifiers for **`bios_vendor`**. Code which uses this information for identifying the host as an AWS instance was not being run. This bug fix adds logic to use the new **`bios_vendor`** identifier. As a result, AWS C5 instances are properly identified by **openshift-ansible** code. (**BZ#1538781**)

- The ***haproxy.cfg*** template in the **load-balancer** role was not updated to reflect changes in new versions of HAProxy, causing the service to fail to start. This bug fix updates the configuration file template to work for newer versions of HAProxy. (**BZ#1538789**)
- In order for the ***adhoc.yml*** playbook to list health checks, they are all loaded, and one failed to load in that environment. Listing health checks failed with an error. This bug fix adjusts the problematic health check, and as a result the listing works as expected. (**BZ#1509157**)
- Single host failures for installing packages were not resulting in a failed playbook run. This caused the playbook to continue and fail at a later point. This fix adds a play directive to cause the playbook to fail immediately if there are any host failures. As a result, if package installation fails for any host, the playbook will mpw fail immediately. (**BZ#1512810**)
- The Jenkins image stream tags have been updated to allow for OpenShift Container Platform version-specific image streams. (**BZ#1525659**)
- An error message on etcd group validation has been updated to reflect the required configurations to better inform the user of the failure state. (**BZ#1538795**)
- A variable defined in an inventory file was being interpreted as a string, not a bool. This caused tasks to not be conditionally run as expected. This bug fix casts the string to a bool for a proper conditional check, and as a result tasks run as expected based on the inventory variable setting. (**BZ#1538816**)
- The **`OPTIONS`** value in the ***/etc/sysconfig/atomic-openshift-node*** file now works properly when multiple options are specified with containerized installations. (**BZ#1539094**)
- When the **`openshift_management_flavor*`** variable was set in ***vars/main.yml***, the variable was not dependent on the user inventory variable for overriding the deployment type. This bug fix updates the **`openshift_management_flavor*`** logic to be based on **`openshift_management_app_template`** from the inventory file. As a result, the requested management template can be successfully deployed. (**BZ#1540246**)
- Alternative names in certificates were not being properly parsed. Alternatives with **`email:`** were being added as additional host names. This bug fix updates the logic to only add alternative names which begin with **`DNS:`**. As a result, **`namedCertificates`** are properly parsed and updated. (**BZ#1538896**)
- The **docker_image_availability** check did not take into account variables that specify a proxy to be used for pulling container images. The check could incorrectly report failure looking for images that are only available via proxy. This bug fix ensures the check now uses a proxy for looking up images if specified in the Ansible variables. The check should accurately report whether the necessary images are available. (**BZ#1539146**)
- The **docker** daemon was incorrectly restarted when redeploying node certificates. This is only necessary when deploying a new CA and can safely be skipped, which ensures that running pods are not restarted when updating node certificates. (**BZ#1542162**)
- A security fix to **openshift-elasticsearch-plugin** caused metrics requests to be rejected because the **oauth-proxy** does not pass the bearer token. This bug fix modifies the plug-in to accept a user name and password from the **oauth-proxy** and deploy with a randomly-generated password. As a result, data and metrics are correctly secured and able to be retrieved based on authorization. (**BZ#1510320**)

- As part of an enhancement, configuration triggers are no longer used for Elasticsearch (ES), and it instead rolls out in a handler. The handler was watching the old ES pod to see if the ES cluster was green or yellow. (**BZ#1533313**)

- Attempting to use an Ansible 2.4 feature in Ansible 2.3 caused the maximum recursion depth to be exceeded in cmp errors. This bug fix makes sure to use the right Ansible features with the correct version of Ansible. As a result, **`mux`** can be installed correctly with Ansible. (**BZ#1540893**)

- A task assumed the **`oc`** binary was in the usable path of Ansible, causing the task to fail when the binary was not found. This bug fix modifies the task to allow the binary to be provided as an Ansible fact. As a result, the task completes successfully. (**BZ#1541403**)

- Messages for which the unique namespace ID could not be determined could not be properly indexed. Messages could be lost and the error message appears in the logs. This bug fix modifies the cache algorithm to provide the necessary data or default the value to **`orphaned`**. As a result, the error message is resolved and messages are stored in an **`orphaned`** index when a unique namespace ID can not be determined. (**BZ#1493022**)

- The multi-tenancy plug-in in Elasticsearch was inadvertently changed, while fixing another bug, not to look up projects for the user upon every login. This caused the list of projects to not be displayed properly. This bug fix changes the multi-tenancy plug-in in Elasticsearch back to look up projects for the user upon every login. As a result, the list of projects is displayed properly. (**BZ#1511432**)

- Fluentd was adding the level field with a value of **`3`** or **`6`**, overwriting any existing level field. The level field set by the application was being removed, and the **`3`** or **`6`** value was not useful. With this bug fix, if there is already a **`level`** field in the record, then see if it is a "close" match to one of the canonical **`level`** field values. For example, if **`level`** is **`CRITICAL`**, convert to **`crit`**; if level is **`WARN`**, convert to **`warning`**. Otherwise, if it cannot be used directly or normalized, convert it to its string representation (ruby **`to_s`** method) and store the string value in the **`level`** field. As a result, if the record already has a level field, the value is normalized or preserved, otherwise, a value like **`info`** or **`err`** is used. (**BZ#1515448**)

- A script that parsed user-supplied configuration produced an incorrect regular expression. This caused no impact on users; the non-escaped period in the regular expression matched the period in the project name as "any character". This bug fix properly escapes characters that should not be expanded as special characters in regular expressions. For example, regular expressions produced after applying this patch would look like **`^\.operations\..*$`** instead of **`.operations.`** (**BZ#1524644**)

- The load on the Elasticsearch cluster is great enough to cause a request from Kibana to timeout before it is able to get a response. This causes no results to be returned and Kibana to display an error. This bug fix modifies the Kibana image to allow configuration of the request timeout. As a result, Kibana is able to retrieve data from Elasticsearch. (**BZ#1538171**)

- When creating an Elasticsearch (ES) cluster of size 3+, the node quorum and recovery settings prevent the first ES node from ever reaching a ready and green state in time during a fresh install. This causes the playbook to time out waiting for the first ES node to be ready. With this bug fix, when new ES nodes are created, OpenShift no longer waits for them to be healthy since the recovery settings and quorum would have changed and will need all nodes to be running at the same time. As a result, the playbook no longer times out when creating large clusters of ES nodes. (**BZ#1544243**)

- The Kibana container image was not built with the required header image. This bug fix replaces the OpenShift Origin header image with the OpenShift Container Platform header image. As a result, the Kibana page displays the desired header. (**BZ#1547263**)

- The link generation code assumes all project logs are written to indices that have a common naming pattern. This can cause users to be linked to non-existent indices. With this bug fix, project logs that will be archived to different indices are annotated with the required information to properly build the link. As a result, users are routed using a link that will query the data store correctly and return data. (**BZ#1547348**)

- Previously, the ClusterResourceOverride admission controller configuration was not passed through to the web console. This meant that the web console was showing fields for resource requests and limits in its forms that were calculated values and should have been hidden in the UI. The problem has been fixed, and the calculated request and limit fields are correctly hidden in the web console. (**BZ#1501164**)

- On slow network connections, the Next button could sometimes take several secrets to enable when opening the create from builder image dialog on the OpenShift web console catalog landing page with the service catalog enabled. This problem has been fixed, and the Next button should now be enabled immediately when the dialog is displayed. (**BZ#1512858**)

- Previously, the sample Git reference and context dir for building images sample repositories was not used when clicking the "Try Sample Repository" link in the web console add to project dialog. The sample reference and context dir are now properly set in the build config that is created when the sample is used. (**BZ#1519096**)

- Previously, provisioned services that reference service classes or service plans that do not exist would not display correctly in the web console overview. This problem has been fixed. The web console now shows provisioned services even when its referenced service class doesn’t exist. (**BZ#1532182**)

- Previously, image stream tag aliases would not show up in the add to project dialog version dropdown when the alias was specified in "name:tag" format in the **`from.name`** property. The **`oc tag --alias`** uses this format. The web console now handles aliases in this format correctly, and they will show up as an alias in the version dropdown with the tag they reference. (**BZ#1518497**)

- Previously, the web console would not correctly update a route host name in the route editor for users with authority to update custom hosts. This problem has been fixed, and the web console now correctly applies the changes when changing a route’s host name. (**BZ#1540783**)

- Use of javascript Array/String methods not supported by IE11 (Array.find, Array.findIndex, String.startsWith, String.endsWith) caused an error that prevents some A-MQ pages to show content when the user clicks on a tree node. This bug fix replaces the unsupported methods with the Lodash library equivalent methods. As a result, A-MQ pages show their content as expected. (**BZ#1543467**)

- Previously, when the service catalog was installed and then uninstalled using the Ansible playbooks, the web console still incorrectly showed the **Provisioned Services** menu item, which did not load any content when clicked. The bug is now fixed and the web console no longer shows the **Provisioned Services** menu after uninstalling the service catalog. (**BZ#1549097**)

- The default **ui-select** selection field had a lower z-index value than the project bar. The**add to project** selection could be partially hidden at small browser viewport size because it was beneath the project bar. With this bug fix, the **ui-select** dropdown field a z-index was made 1 greater than the project bar. The **select** field will now appear over the project bar if needed. (**BZ#1479114**)

- An OpenShift Container Platform node was not waiting long enough for the VNID while master assigned VNID. It could take a while to propagate to the node, resulting in failed pod creation. With this bug fix, the timeout increased from 1 to 5 seconds for fetching VNID on the node and pod creation succeeds. (**BZ#1540606**)

- When using the "static per-project egress IPs" feature, egress IPs may stop working in some circumstances if an egress IP is moved from one project to another, or from one node to another. Additionally, if the same egress IP is assigned to two different projects, or two different nodes, then it may not work correctly even after the duplicate assignment is removed. This bug fix addresses these issues and static per-project egress IPs now works more reliably. (**BZ#1553297**)

- Certain initialization code was only run when doing a full SDN setup, and not when OpenShift Container Platform was restarted and found the SDN was already up and running. If the **atomic-openshift-node** service was restarted, that node would be unable to create new per-project static egress IPs (`HostSubnet.EgressIPs`). The egress IP initialization code is now run in all circumstances. Per-project static egress IPs work correctly, even after a node restart. (**BZ#1535658**)

- In some circumstances, iptables rules could become reordered in a way that would cause the "per-project static IP address" feature to stop working for some IP addresses. (For most users, egress IP addresses that ended with an even number would continue to work, but egress IP addresses ending with an odd number would fail.) External traffic from pods in a project that was supposed to use a per-project static IP address would end up using the normal node IP address instead. The iptables rules have been changed so that they now have the expected effect even when they get reordered. With this bug fix, the per-project static egress IP feature now works reliably. (**BZ#1540611**)

- In some circumstances, nodes were apparently receiving a duplicate out-of-order HostSubnet "deleted" event from the master. When processing the duplicate event, the node could end up deleting OVS flows corresponding to an active node, causing pods on the two nodes to be unable to communicate with each other. This was most noticeable when it happened to a node hosting the registry. With this bug fix, the HostSubnet event-processing code will now notice that the event is a duplicate and ignore it. As a result, OVS flows are not deleted, and pods can communicate. (**BZ#1546170**)

- Previously, a "transport endpoint is not connected" error occurred when deleting projects with pods using CNS backed PVs. This bug fix corrects the issue. (**BZ#1546156**)

- When using the vSphere cloud provider, the `InternalIP` information was not populated for nodes. This would cause issues with Heapster since it uses the `InternalIP` for gathering metrics. With this bug fix, the `InternalIP` information is now populated and the issues are resolved. (**BZ#1527315**)

- The installer has been modified to turn on API aggregation for upgrades to OpenShift Container Platform 3.7, which is a required dependency for service catalog to work properly. (**BZ#1523298**)

- Previously, the Ansible installer was not updating the API service definition with newly generated certificate data. Also, the service catalog API server was not being restarted to pick up the new certificates. Using mismatched CAs causes x509 errors in the API server logs and, with this bug fix, is now corrected. (**BZ#1523625**)

- OpenShift Container Platform 3.7 upgrades were incorrectly using an old default etcd port value of `4001`. This bug fix modifies the value to the number, `2379`. (**BZ#1535206**)

- Unbinding a template service instance throws an error if the template service instance was deleted. It becomes impossible to unbind a service instance if the template service instance was manually deleted, including if the project containing the TSI was deleted. The template service broker will return **success/gone** in cases where the unbind refers to a non-existent template service instance. With this bug fix, the unbind can proceed even if the TSI no longer exists. (**BZ#1543044**)

- The broker overwrote the bind credentials on subsequent bind endpoint calls, causing bind credentials to be saved incorrectly. With this bug fix, bind credentials are now saved for every call. (**BZ#1501512**)

- The update strategy in the controller manager template was incorrectly set. Previously, any parameters that were modified in the controller manager template would fail to cause the pod to redeploy with the updated changes. With this bug fix, the issue is corrected. (**BZ#1537227**)

- OpenShift Container Platform checked mounted NFS volumes with **squash** (running as root). Permissions of OpenShift Container Platform (running as root) were squashed to user **nobody**, which did not have permissions to access mounted NFS volumes. As a result, OpenShift Container Platform checks failed and NFS volumes were not unmounted. OpenShift Container Platform does not access mounted NFS volumes at all and checks for mounts by parsing the ***/proc*** filesystem. With this bug fix, NFS volumes with the root **squash** option are unmounted. (**BZ#1536019**)

- Incorrect batch size calculations in Ansible 2.4.1 would cause playbooks to fail when using **max_fail_percentage**. The batch calculations were updated in Ansible 2.4.2 to correctly account for failures in each batch. (**BZ#1538807**)

- During the control plane upgrade, etcd is now upgraded to etcd 3.2.x. (**BZ#1495486**)

2.8.4.2. Enhancements

- To avoid the unplanned restart of Elasticsearch pods, which could cause loss of data, Elasticsearch deployment configurations are now created without configuration change triggers. When a configuration change is made to an Elasticsearch deployment configuration, it is no longer automatically rolled out, and can instead be done manually at a scheduled time. See Manual Elasticsearch Rollouts for steps on how to do so. (**BZ#1498989**)

- Regular expressions are now allowed in Curator settings. A special tag can be used in Curator settings to specify custom regular expressions. (**BZ#1541948**)

2.8.4.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.5. RHSA-2018:1231 - OpenShift Container Platform 3.7.44 Security and Bug Fix Update

Issued: 2018-04-29

OpenShift Container Platform release 3.7.44 is now available. The packages, security fixes, and bug fixes included in the update are documented in the RHSA-2018:1231 and RHBA-2018:1261 advisories. The list of container images included in the update are documented in the RHBA-2018:1230 and RHBA-2018:1262 advisories.

2.8.5.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.6. RHBA-2018:1576 - OpenShift Container Platform 3.7.46 Bug Fix and Enhancement Update

Issued: 2018-05-18

OpenShift Container Platform release 3.7.46 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:1576 advisory. The list of container images included in the update are documented in the RRHBA-2018:1577 advisory.

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

2.8.6.1. Bug Fixes

- Previously, the value of **`openshift_master_extension_scripts`** was not preserved during installations when the service catalog was enabled. This bug fix ensures that the scripts in **`openshift_master_extension_scripts`** are now correctly set during installation. (**BZ#1541129**)

- When verifying various API endpoints, a timeout was not set. This meant that the verification would wait 120 seconds for the connection to fail and then wait the prescribed delay before retrying. This would have led to certain tasks waiting up to two hours rather than moving forward with the installation. With this bug fix, a connection timeout of two seconds has now been set. (**BZ#1556679**)

- The EFS Provisioner image tag was incorrectly set to match the tag of all other OpenShift Container Platform images. This has been updated and now installations will pull from the **`latest`** image tag, which should be compatible with all versions of OpenShift Container Platform. (**BZ#1566300**)

- The number of fluentd outputs is computed in the startup script run.sh. When the forward plug-in was configured, it was not correctly counted. This caused a number of output to be missed when the forward plug-in was configured. The number of output is used for calculating the file buffer size and the size grew larger than expected. This bug fix updates the logic of the calculation to include the forward plug-in case. As a result, the correct output number then the correct file buffer size is now derived when the forward plug-in is configured. (**BZ#1569544**)

- The default write operation for fluentd to Elasticsearch is index. Writes can trigger unnecessary **`delete`** operations for Elasticsearch causing extra load that affects performance. This bug fix utilizes the **`create`** operation which will avoid 'delete' when duplicates are introduced. (**BZ#1568415**)

- Due to incorrect management of OVS flows, if two nodes rebooted and swapped IP addresses when they came back up, then other nodes might become unable to send traffic to pods on one or both of those nodes. With this bug fix, the code that manages OVS flows is now more careful to make the correct changes in cases of node IP reassignment. As a result, pod-to-pod traffic should continue to work correctly even after nodes swap IP addresses. (**BZ#1570396**)

- Updating egress policy required blocking outgoing traffic, patching OVS flows, and then re-enabling traffic. However, the OVS flow generation for DNS names was slow, resulting in a few seconds egress traffic downtime, which may not be acceptable. This bug fix changes update egress policy handling to pre-populate all new OVS flows before blocking the outgoing traffic. As a result, the downtime during egress policy updates is now reduced. (**BZ#1571433**)

- When using per-namespace static egress IPs, all external traffic is routed through the egress IP. "External" means all traffic which is not directed to another pod, and so includes traffic from the pod to the pod's node. When pods are told to use the node's IP address for DNS, and the pod is using a static egress IP, then DNS traffic will be routed to the egress node first, and then back to the original node, which might be configured to not accept DNS requests from other hosts, causing the pod to be unable to resolve DNS. With this bug fix, pod-to-node DNS requests now bypass the egress IP and go directly to the node, and as a result DNS now works as expected. (**BZ#1570400**)

- This update fixes an issue where a node can stop reporting status if the connection to the Azure API is terminated uncleanly, resulting in a long timeout before the connection is re-established, blocking the status update loop. (**BZ#1573122**)

- Resources consumed by pods whose node is unreachable or have past their termination grace period are no longer counted against quota. (**BZ#1455743**)

2.8.6.2. Enhancements

- You may now define a set of hooks to run arbitrary tasks during the node upgrade process. To implement these hooks, set **`openshift_node_upgrade_pre_hook`**, **`openshift_node_upgrade_hook`**, or **`openshift_node_upgrade_post_hook`** to the path of the task file you wish to execute. The **`openshift_node_upgrade_pre_hook`** hook is executed after draining the node and before it has been upgraded. The **`openshift_node_upgrade_hook`** hook is executed after the node has been drained and packages updated but before it is marked schedulable again. The **`openshift_node_upgrade_post_hook`** hook is executed after the node has been marked schedulable immediately before moving on to other nodes. (**BZ#1572798**)

- Previously, OpenShift Container Platform defaulted the indexing request timeout for fluentd to 600 seconds, or about 10 minutes. In certain situations, this timeout was not long enough. If we make this essentially infinite, we'll avoid fluentd pods re-submitting requests unnecessarily. With this enhancement, the fluentd timeout is now 2147483648 and will wait for a very long time before resubmitting a request to Elasticsearch due to a timeout failure. As a result, this avoids fluentd pods being resubmitted unnecessarily. (**BZ#1569548**)

2.8.6.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.7. RHBA-2018:1798 - OpenShift Container Platform 3.7.52 Bug Fix and Enhancement Update

Issued: 2018-06-07

OpenShift Container Platform release 3.7.52 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:1798 advisory. The list of container images included in the update are documented in the RHBA-2018:1799 advisory.

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

2.8.7.1. Bug Fixes

- Jenkins **`no_proxy`** processing could not handle suffixes like **`".svc"`**. As a result, communication between a Jenkins k8s agent pod and the Jenkins master would attempt to go through a configured http_proxy and fail. With this bug fix, the OpenShift Container Platform jenkins agent images are updated to automatically include the jenkins master and jnlp hosts in the **`no_proxy`** list. The Jenkins limitation for **`no_proxy`** processing is now circumvented. (**BZ#1578987**)

- This bug fix addresses an issue where HPA will scale above **`maxReplicas`** in some situations. (**BZ#1578422**)
- This bug fix addresses an issue where **`cfs_quota`** might still be set on a pod even when **`cpu-cfs-quota`** is set to **`false`** on the node. (**BZ#1581862**)

2.8.7.2. Images

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

```
openshift3/container-engine:v3.7.52-1
openshift3/cri-o:v3.7.52-1
openshift3/image-inspector:v3.7.52-1
openshift3/jenkins-2-rhel7:v3.7.52-1
openshift3/jenkins-slave-base-rhel7:v3.7.52-1
openshift3/jenkins-slave-maven-rhel7:v3.7.52-1
openshift3/jenkins-slave-nodejs-rhel7:v3.7.52-1
openshift3/local-storage-provisioner:v3.7.52-1
openshift3/logging-auth-proxy:v3.7.52-1
openshift3/logging-curator:v3.7.52-1
openshift3/logging-elasticsearch:v3.7.52-1
openshift3/logging-eventrouter:v3.7.52-1
openshift3/logging-fluentd:v3.7.52-1
openshift3/logging-kibana:v3.7.52-2
openshift3/mariadb-apb:v3.7.52-1
openshift3/mediawiki-123:v3.7.52-1
openshift3/mediawiki-apb:v3.7.52-1
openshift3/metrics-cassandra:v3.7.52-1
openshift3/metrics-hawkular-metrics:v3.7.52-1
openshift3/metrics-hawkular-openshift-agent:v3.7.52-1
openshift3/metrics-heapster:v3.7.52-1
openshift3/mysql-apb:v3.7.52-1
openshift3/node:v3.7.52-1
openshift3/oauth-proxy:v3.7.52-1
openshift3/openvswitch:v3.7.52-1
openshift3/ose-ansible-service-broker:v3.7.52-1
openshift3/ose-ansible:v3.7.52-1
openshift3/ose-cluster-capacity:v3.7.52-1
openshift3/ose-deployer:v3.7.52-1
openshift3/ose-docker-builder:v3.7.52-1
openshift3/ose-docker-registry:v3.7.52-1
openshift3/ose-egress-http-proxy:v3.7.52-1
openshift3/ose-egress-router:v3.7.52-1
openshift3/ose-f5-router:v3.7.52-1
openshift3/ose-haproxy-router:v3.7.52-1
openshift3/ose-keepalived-ipfailover:v3.7.52-1
openshift3/ose-pod:v3.7.52-1
openshift3/ose-recycler:v3.7.52-1
openshift3/ose-service-catalog:v3.7.52-1
```

```
openshift3/ose-sti-builder:v3.7.52-1
openshift3/ose-template-service-broker:v3.7.52-1
openshift3/ose:v3.7.52-1
openshift3/postgresql-apb:v3.7.52-1
openshift3/prometheus-alert-buffer:v3.7.52-1
openshift3/prometheus-alertmanager:v3.7.52-1
openshift3/prometheus:v3.7.52-1
openshift3/registry-console:v3.7.52-1
openshift3/snapshot-controller:v3.7.52-1
openshift3/snapshot-provisioner:v3.7.52-1
```

2.8.7.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.8. RHBA-2018:2010 - OpenShift Container Platform 3.7.54 Bug Fix Update

Issued: 2018-06-27

OpenShift Container Platform release 3.7.54 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2009 advisory. The list of container images included in the update are documented in the RHBA-2018:2010 advisory.

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

2.8.8.1. Bug Fixes

- When performing pullthrough, even when the internal registry is the source of the image, the registry will attempt to pull from itself if a manifest cannot be found. The pullthrough attempt fails both because the manifest does not exist and potentially because the registry is not secured. This bug fix adds a check to prevent attempting manifest pullthrough when the image source is the internal registry. As result, pullthrough will not be attempted and the failure will not occur. (**BZ#1557399**)

- In certain cases, an existing etcd installation may not have updated configuration variables, causing services to fail. This bug fix ensures the etcd.conf file is verified during upgrades to ensure all variables are set as expected. (**BZ#1563376**)

- When OpenShift Container Platform (OCP) 3.7 introduced a second container in the Elasticsearch pod, the related role did not account for the fact that upgrading to 3.7 would mean there was only one container. This caused the playbook run to fail because it does not have a second container. This bug fix expands the check to also see if there is more than one container in the pod to account for upgrades from pre-3.7 to 3.7. As a result, the playbook is able to run to completion when upgrading from 3.6 to 3.7. (**BZ#1557044**)

- Certain upgrade tasks used the default kubeconfig file, which may have been updated by the administrator in such a way that it prevented upgrade success. With this bug fix, the upgrade playbooks now use an administrator-specific kubeconfig file which is not prone to being altered. As a result, this now ensures proper upgrade process. (**BZ#1576527**)

- Certain incoming data to Elasticsearch has a field **`record["event"]`** that is a String value and not the Hash value expected by the **`transform_eventrouter`** code. This caused the code to throw an error and fluentd to emit an error like:

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

This bug fix changes the `transform_eventrouter` code to only process the `record["event"]` field if it is a Hash. As a result, records can flow through to Elasticsearch again. (**BZ#1588827**))

- Curator gave up waiting for Elasticsearch to become ready after 60 seconds, causing Curator to exit with an error status. With this bug fix, Curator waits for Elasticsearch with no time limit; exponential wait is capped at 300 seconds. As a result, Curator is passivated if Elasticsearch is not reachable, instead of crashing. Elasticsearch status is polled every x seconds. (**BZ#1557483**)

- Previously, clicking "Back" from the Template Configuration step in the web console "Import YAML" wizard would prevent users from clicking "Next" again. The problem has been fixed, and the "Next" button works properly in this wizard now after clicking "Back". (**BZ#1530507**)

- Network policy objects cannot be upgraded between OCP 3.7.z versions before 3.7.23. The documentation has been changed to explain how to work around the problem, and the error has been fixed in 3.7.23. As a result, you can now upgrade as expected. (**BZ#1570777**)

- The APB tooling was not looking in the **openshift-ansible-service-broker** namespace for the broker route, and the `--broker` flag misbehaved with improper suffixes. This caused commands like `apb list` to not behave as expected. This bug fix backports some newer fixes to OCP 3.7 so that the tooling automatically tries to find the correct resources even if the user does not provide all of the information. As a result, the `apb list` command works, even without the `--broker` flag. (**BZ#1554138**, **BZ#1555390**)

2.8.9. RHBA-2018:2234 - OpenShift Container Platform 3.7.57 Bug Fix Update

Issued: 2018-07-24

OpenShift Container Platform release 3.7.57 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2234 advisory. The list of container images included in the update are documented in the RHBA-2018:2235 advisory.

2.8.9.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.10. RHBA-2018:2337 - OpenShift Container Platform 3.7.61 Bug Fix and Enhancement Update

Issued: 2018-08-09

OpenShift Container Platform release 3.7.61 is now available. The packages and bug fixes included in the update are documented in the RHBA-2018:2337 advisory. The list of container images included in the update are documented in the RHBA-2018:2338 advisory.

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

2.8.10.1. Bug Fixes

- Due to corrupt file chunk buffers, Fluentd was blocked processing messages until the buffer was removed. This bug fix introduces a handler to remove corrupt messages from the processing work flow. Corrupt messages are now sent to a dead letter queue while continuing to process other messages and the pipleline is no longer blocked. (**BZ#1511116**)

- **`fluent-plugin-elasticsearch`** improperly handled book-keeping of submitted records. Fluentd was stuck processing a chunk even though there was a valid request and response. With this bug fix, records submitted to Elasticsearch are properly accounted for and the pipeline is no longer stuck. (**BZ#1593312**)

- Node labeling for Fluentd was done via a script run from the ***/tmp*** directory. If ***/tmp*** was mounted with the **`noexec`** option, then Ansible could not execute the script and fails to complete the playbook. The execution is now moved out of a script and into a **`shell`** Ansible task. The playbook run is now completed successfully. (**BZ#1600685**)

- There was an undefined variable in the Fluentd role and the deployment failed when trying to label the nodes. With this bug fix, the variable is replaced with the correct one to the **`oc`** client binary. Logging deployment is now possible and properly labels the nodes. (**BZ#1601307**)

- Recently, **`cloudResourceSyncManager`** was implemented, which continuously fetched node addresses from cloud providers. The kubelet then received node addresses from the **`cloudResourceSyncManager`**. At the time of node registration or kubelet start, the kubelet fetches node addresses in a blocking loop from **`cloudResourceSyncManager`**. The issue was that **`cloudResourceSyncManager`** was not started before the kubelet had started fetching node addresses from it for the first time. Due to this, the kubelet got stuck in the blocking loop and never returned. It caused node failures at the network level, and no node could be registered. Also, as the kubelet was blocked early, the **`cloudResourceSyncManager`** never got a chance to start. The **`cloudResourceSyncManager`** is now started early in the kubelet startup process so that the kubelet does not get blocked on it and **`cloudResourceSyncManager`** is always started. (**BZ#1601378**)

2.8.10.2. Enhancements

- This feature allows the number of indices and replicas to be configured using environment variables. The logs collected for infra services consume a large portion of the available disk space. Spreading the data across available nodes by modifying the replica and shard settings allow Elasticsearch to better support these large amounts of data. This feature results in improved performance in Elasticsearch when there are large amounts of data from infra services. (**BZ#1552977**)

2.8.10.3. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.11. RHBA-2018:2547 - OpenShift Container Platform 3.7.62 Bug Fix Update

Issued: 2018-08-28

OpenShift Container Platform release 3.7.62 is now available. The list of packages and bug fixes included in the update are documented in the RHBA-2018:2547 advisory. The list of container images included in the update are documented in the RHBA-2018:2546 advisory.

2.8.11.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.12. RHBA-2018:2656 - OpenShift Container Platform 3.7.64 Bug Fix Update

Issued: 2018-09-24

OpenShift Container Platform release 3.7.64 is now available. The list of packages and bug fixes included in the update are documented in the RHBA-2018:2656 advisory. The list of container images included in the update are documented in the RHBA-2018:2657 advisory.

2.8.12.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.13. RHSA-2018:2908 - OpenShift Container Platform 3.7.72 Security and Bug Fix Update

Issued: 2018-11-19

OpenShift Container Platform release 3.7.72 is now available. The list of packages and security fixes included in the update are documented in the RHSA-2018:2906 advisory. The container images included in the update are provided by the RHBA-2018:2905 advisory.

2.8.13.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

2.8.14. RHSA-2019:0617 - OpenShift Container Platform 3.7.108 Bug Fix Update

Issued: 2019-04-10

OpenShift Container Platform release 3.7.108 is now available. The list of packages and security fixes included in the update are documented in the RHSA-2019:0617 advisory. The container images included in the update are provided by the RHBA-2019:0616 advisory.

2.8.14.1. Upgrading

To upgrade an existing OpenShift Container Platform 3.6 or 3.7 cluster to this latest release, use the automated upgrade playbook. See Performing Automated In-place Cluster Upgrades for instructions.

CHAPTER 3. XPAAS RELEASE NOTES

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

CHAPTER 4. COMPARING WITH OPENSHIFT ENTERPRISE 2

4.1. OVERVIEW

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

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

4.2. ARCHITECTURE CHANGES

Gears Versus Containers

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

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

Kubernetes

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

4.3. APPLICATIONS

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

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

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

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

Flexible linking means you can link any two arbitrary components together. As long as one component can export environment variables and the second component can consume values from those environment variables, and with potential variable name transformation, you can link together any two components without having to change the images they are based on. So, the best containerized implementation of your desired database and web framework can be consumed directly rather than you having to fork them both and rework them to be compatible.

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

4.4. CARTRIDGES VERSUS IMAGES

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

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

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

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

Dependencies

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

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

Collection

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

Source Code

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

Build

In OpenShift v2, builds occurred in application gears. This meant downtime for non-scaled applications due to resource constraints. In v3, builds happen in separate containers. Also, OpenShift v2 build results used rsync to synchronize gears. In v3, build results are first committed as an immutable image and published to an internal registry. That image is then available to launch on any of the nodes in the cluster, or available to rollback to at a future date.

Routing

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

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

4.5. BROKER VERSUS MASTER

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

4.6. DOMAIN VERSUS PROJECT

A project is essentially a v2 domain.

CHAPTER 5. REVISION HISTORY: RELEASE NOTES

5.1. MON JAN 22 2018

Affected Topic	Description of Change
OpenShift Container Platform 3.7 Release Notes	Added release notes for RHBA-2018:0113 - OpenShift Container Platform 3.7.23 Bug Fix and Enhancement Update.

5.2. WED JAN 10 2018

Affected Topic	Description of Change
OpenShift Container Platform 3.7 Release Notes	Added release notes for RHBA-2018:0076 - OpenShift Container Platform 3.7.14-9 Images Update.

5.3. MON DEC 18 2017

Affected Topic	Description of Change
OpenShift Container Platform 3.7 Release Notes	Added release notes for RHBA-2017:3464 - OpenShift Container Platform 3.7.14 Bug Fix and Enhancement Update.

5.4. WED NOV 29 2017

OpenShift Container Platform 3.7 Initial Release