OpenShift Dedicated 3

Release Notes

Last Updated: 2018-05-17

Legal Notice

Abstract

Table of Contents

CHAPTER 1. OVERVIEW

The following release notes for OpenShift Dedicated summarize key features upon general availability. OpenShift Dedicated uses the same code base as OpenShift Container Platform 3; for more detailed technical notes, see the OpenShift Container Platform 3.7 Release Notes.

1.1. VERSIONING POLICY

OpenShift Dedicated 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 Dedicated 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 Dedicated will continue to support older **oc** clients against newer servers. For example, a 3.4 **oc** will work against 3.3, 3.4, and 3.5 servers.

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

Table 1.1. Compatibility Matrix

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

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

1 Fully compatible.

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

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

CHAPTER 2. LATEST PRODUCT UPDATES

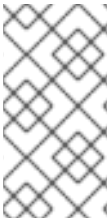
2.1. OVERVIEW

Red Hat OpenShift Dedicated provides managed, single-tenant OpenShift environments on the public cloud. Installed and managed by Red Hat, these clusters can provide additional resources as needed, use Red Hat JBoss® Middleware and partner services, integrate with an existing authentication system, and connect to a private datacenter.

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

See https://www.openshift.com/dedicated for more information.

2.2. ABOUT THIS UPDATE



NOTE

Cluster upgrades to a new update of OpenShift Dedicated are scheduled to begin soon, per recent Red Hat communications; customers can check the OpenShift Dedicated Status page for their scheduled upgrade date. This topic has been updated to reflect the upcoming new features and changes.

The latest update of Red Hat OpenShift Dedicated uses Red Hat OpenShift Container Platform version 3.7, which is based on OpenShift Origin 3.7. New features, changes, bug fixes, and known issues that pertain to the latest updates of OpenShift Dedicated are included in this topic.

2.3. NEW FEATURES AND ENHANCEMENTS

2.3.1. Dedicated Administrator Updates

2.3.1.1. Service Accounts

As of OpenShift Dedicated on 3.7, a reserved project called **dedicated-admin** will be created and maintained on each cluster. Any service account created within this project will have **dedicated-admin** permissions by default. Only **dedicated-admin** users will be able to manage service accounts within this project.

2.3.1.2. Oauthclientauthorizations

As of OpenShift Dedicated on 3.7, **dedicated-admin** users will be able to list and delete **`oauthclientauthorizations`**.

2.3.1.3. Scheduler statistics

As of OpenShift Dedicated on 3.7, **dedicated-admin** users will be able to access scheduler statistics for each node by using the **`oc describe node`** command.

2.3.2. Container Orchestration

2.3.2.1. 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.

2.3.3. Security

2.3.3.1. Documented Private and Public Key Configurations and Crypto Levels

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

OpenShift Dedicated 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 Dedicated.

2.3.3.2. 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`**.

2.3.4. Networking

2.3.4.1. Network Policy

Network Policy is now fully supported in OpenShift Dedicated using 3.7.

Network Policy is an optional plug-in specification of how selections 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.

After installing the Network Policy plug-in, an annotation that flips the namespace from **`allow all traffic`** to **`deny all traffic`** must first be set on the namespace. At that point, **`NetworkPolicies`** can be created that define what traffic to allow. The annotation is as follows:

```
$ oc annotate namespace ${ns} 'net.beta.kubernetes.io/network-policy=
{"ingress":{"isolation":"DefaultDeny"}}'
```

NOTE

The annotation is not needed when using the v1 API.

The allow-to-red policy specifies "all red pods in namespace **`project-a`** allow traffic from any pods in any namespace." This does not apply to the red pod in namespace **`project-b`** because **`podSelector`** only applies to the namespace in which it was applied.

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.

2.3.4.2. 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. Developer Experience

2.3.5.1. Template Instantiation API

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

See Template Instantiation for more information.

2.3.5.2. Chaining Builds

In OpenShift Dedicated on 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.

2.3.6. Web Console

2.3.6.1. Initial Experience

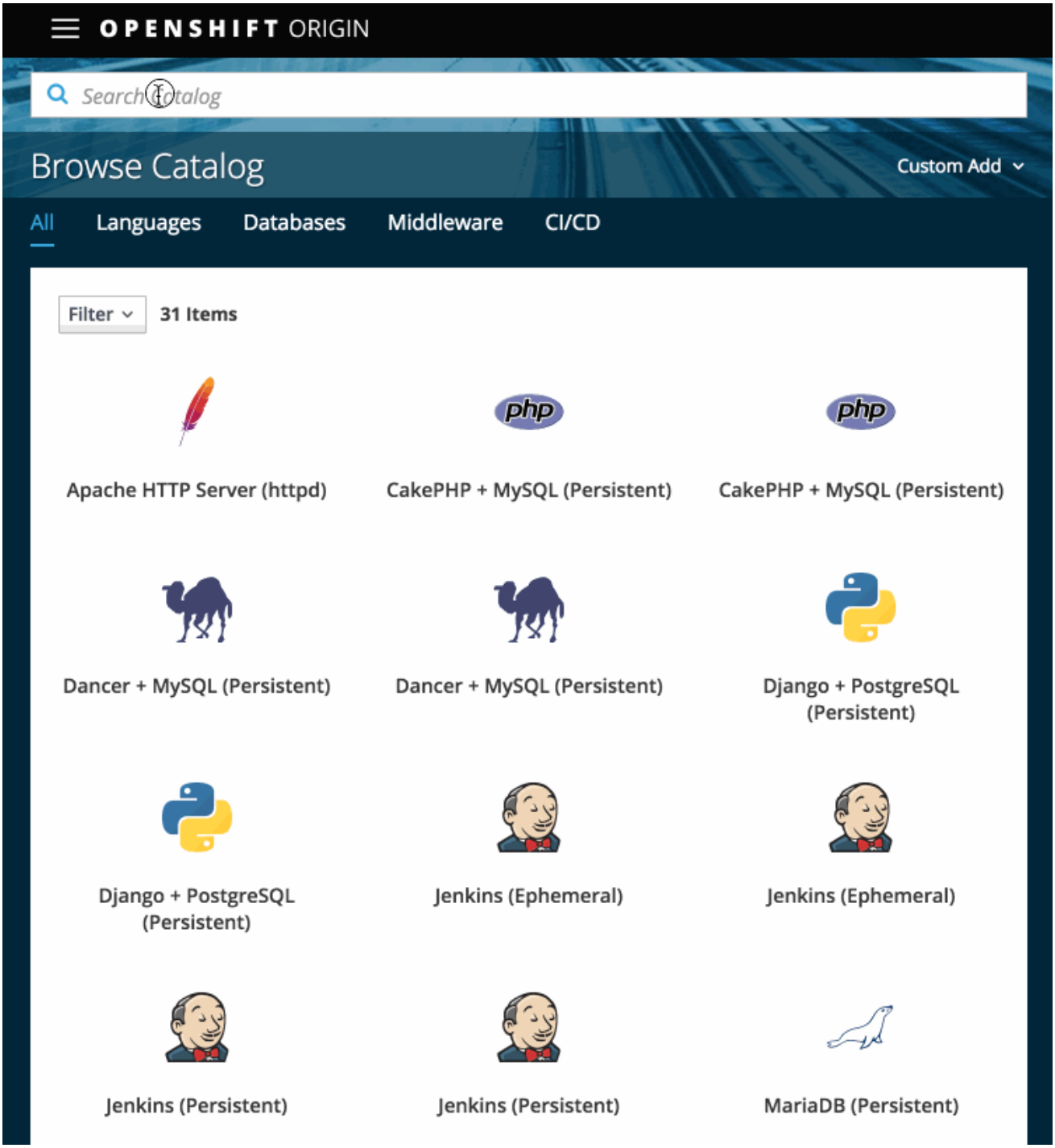
OpenShift Dedicated on 3.7 provides a better initial user experience with the Service Catalog. This includes:

- 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. These Service Catalog items are not yet available in OpenShift Dedicated.

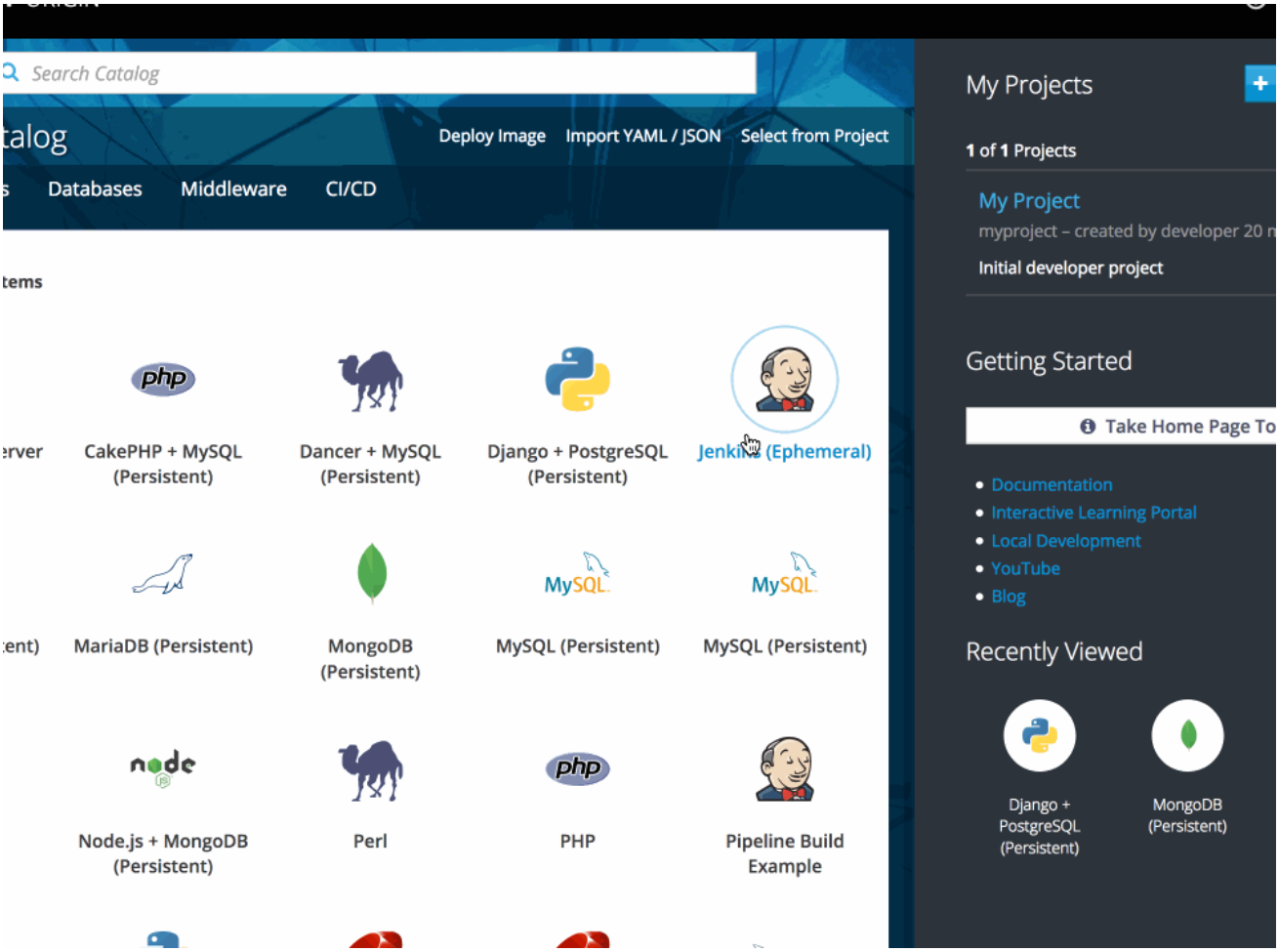
2.3.6.2. Search Catalog

OpenShift Dedicated on 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.



2.3.6.3. Add from Catalog

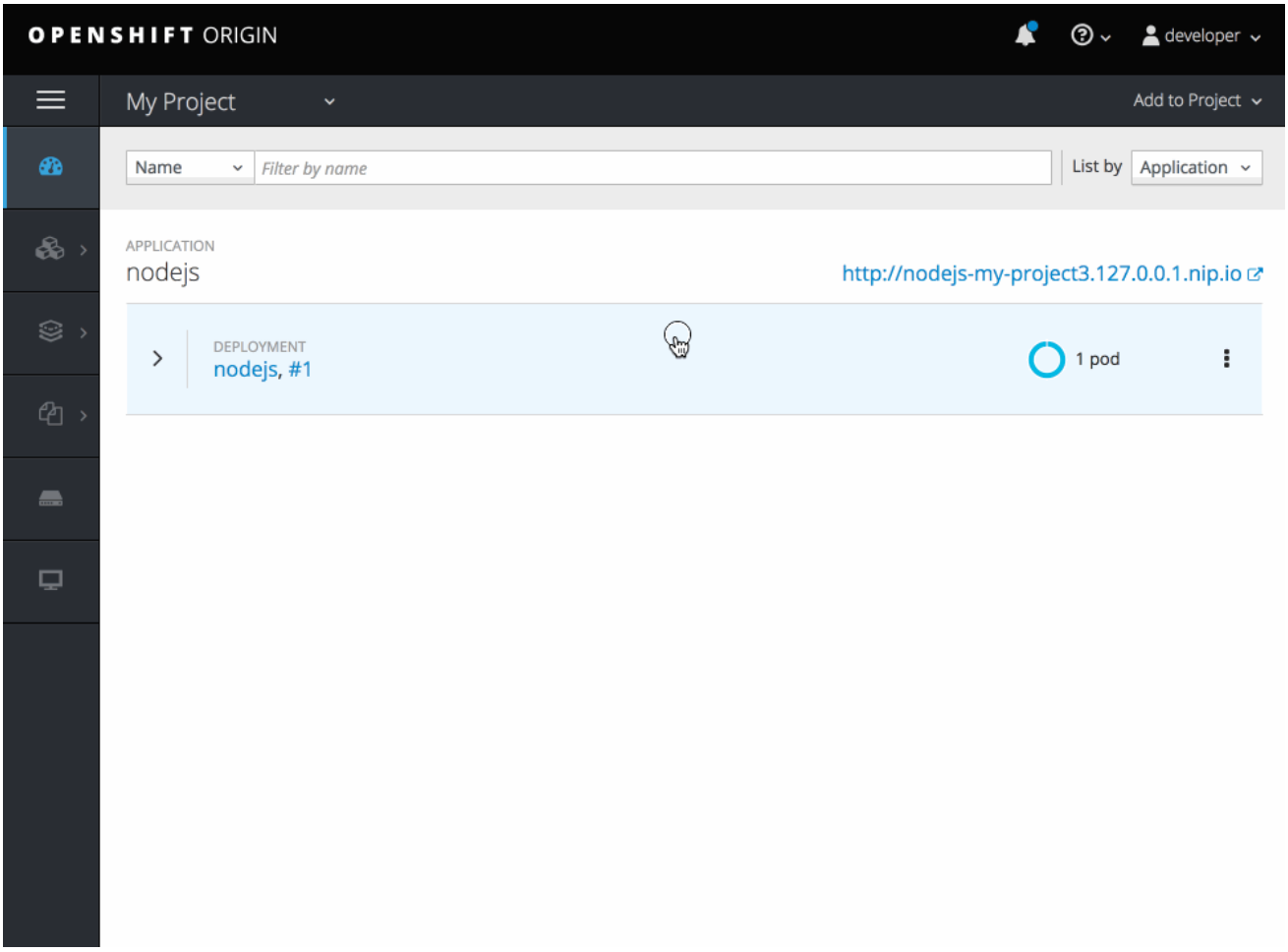
Provision a service from the catalog. Select the desired service and follow prompts for the desired project and configuration details.



2.3.6.4. Connect a Service

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

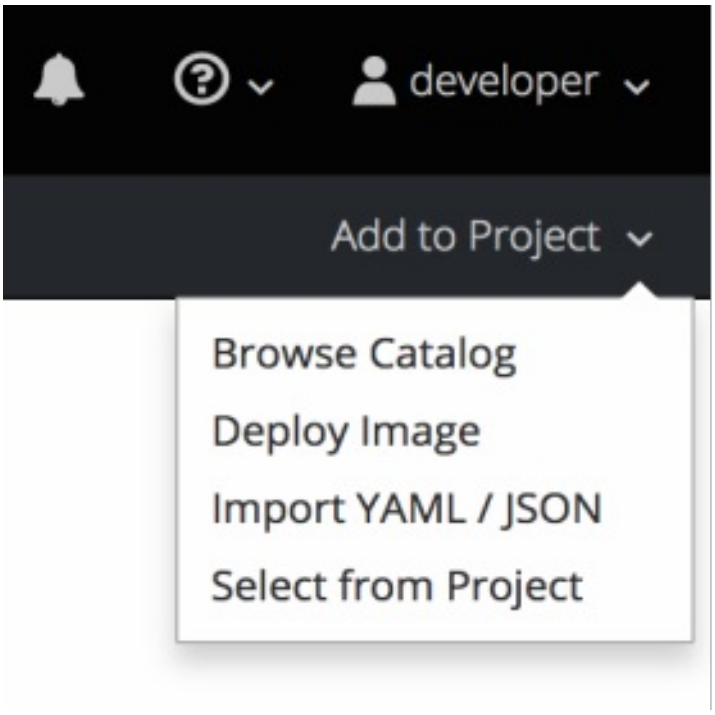
The broker returns a secret, which is stored in the project for use. You are guided through a process to update the deployment to inject a secret.



2.3.6.5. 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.

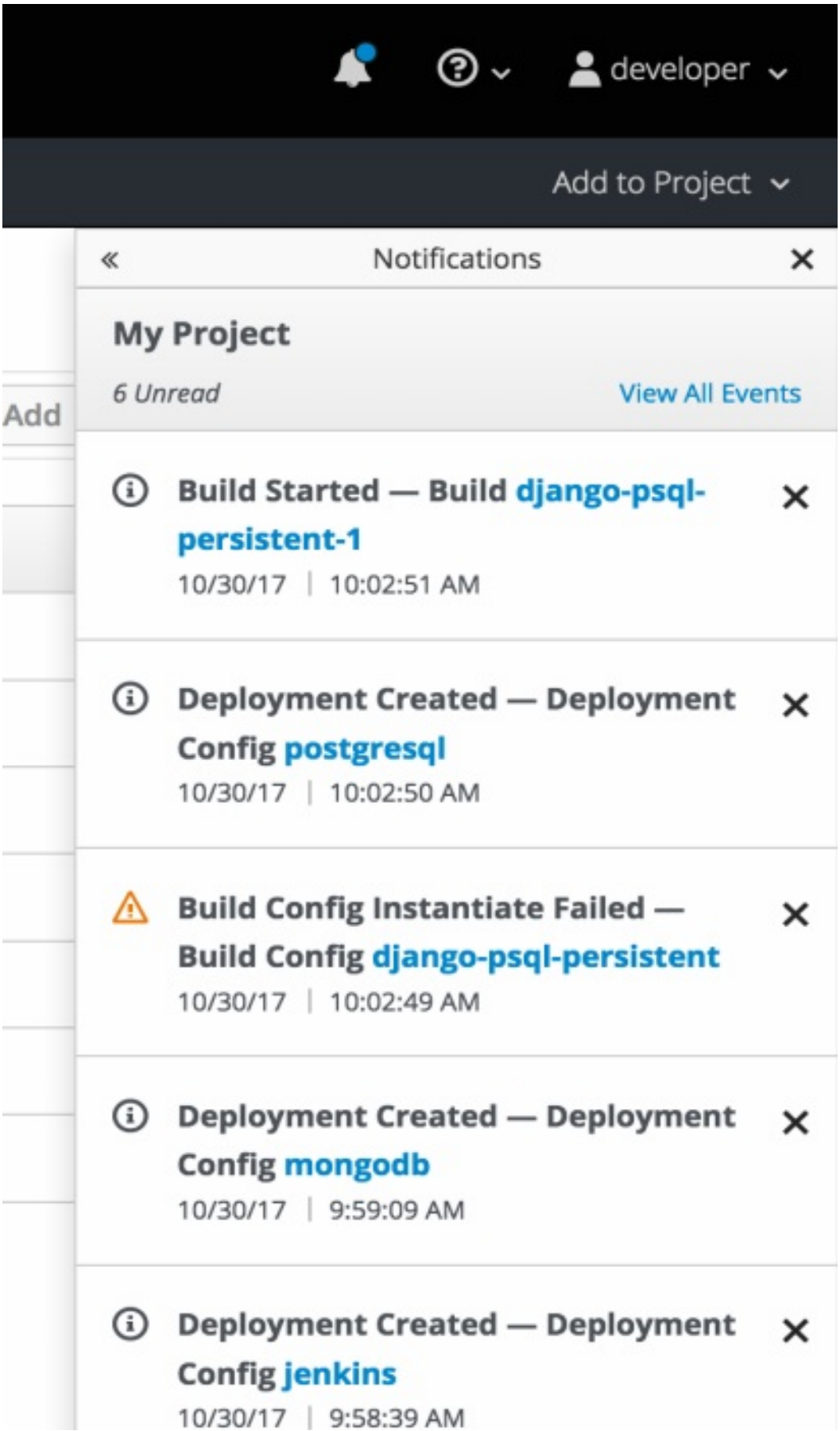
Upload the template, then select the template from a project.

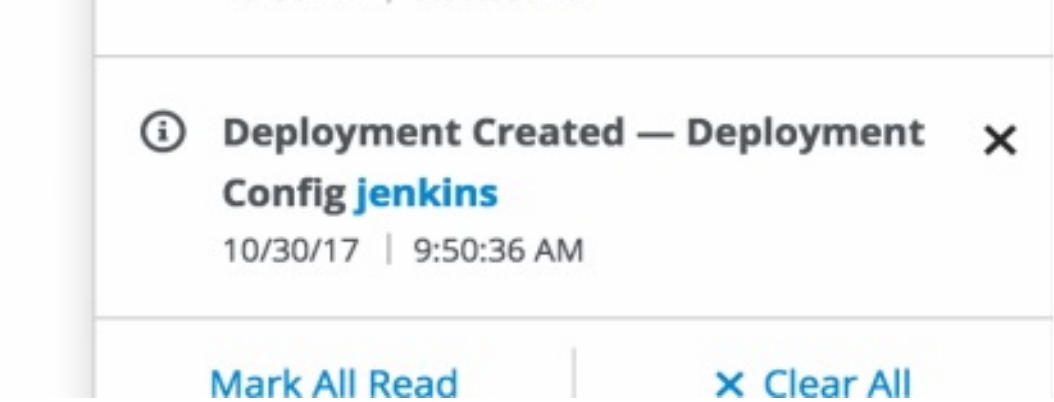


2.3.6.6. Notifications

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

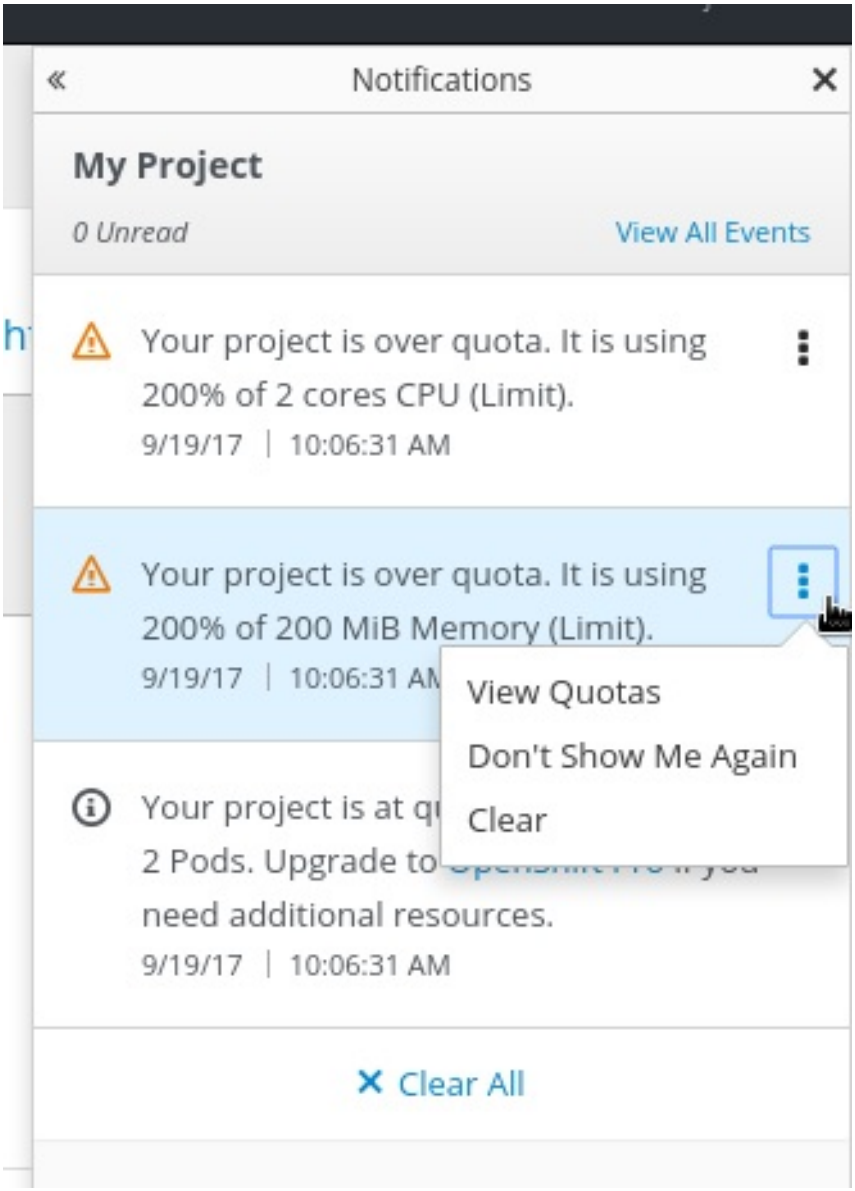
The bell icon is decorated when new notifications exist. You can mark all read, clear all, view all, or dismiss individual ones. Key notifications are represented with the level of information, warning, or error.





2.3.6.7. Improved Quota Warnings

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



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.

2.3.6.8. 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 introduced several notable technical changes to OpenShift Dedicated. Refer to the OpenShift Container Platform 3.7 Release Notes for more information on technical changes to the underlying software.

2.5. BUG FIXES

Refer to the OpenShift Container Platform 3.7 Release Notes for more information on bug fixes.