The Differences Between Kubernetes and OpenShift

By James Buckett (/author/james-buckett)   February 22, 2017

I'm a Cloud Consultant at Levvel (https://levvel.io) that specializes in Kubernetes. We are partners with Red Hat, so I was asked to skill up on Red Hat Openshift. Red Hat Openshift is a Platform as a Service based on Kubernetes, but I was largely unaware of the different terms in use in the OpenShift ecosystem.

I wrote this to assist people who are familiar with Kubernetes but unfamiliar with Openshift, so that they can understand the different verbiage used in each ecosystem. I'll cover some of the infrastructure differences between Kubernetes and OpenShift—specifically, the differences between route/router and ingress/ingress controllers and between namespaces and projects.

Let's get started.

OpenShift is a platform as a service (PaaS) from Red Hat that is built on Docker and Kubernetes. Kubernetes is an open source, container as a service (CaaS) project originating from Google.

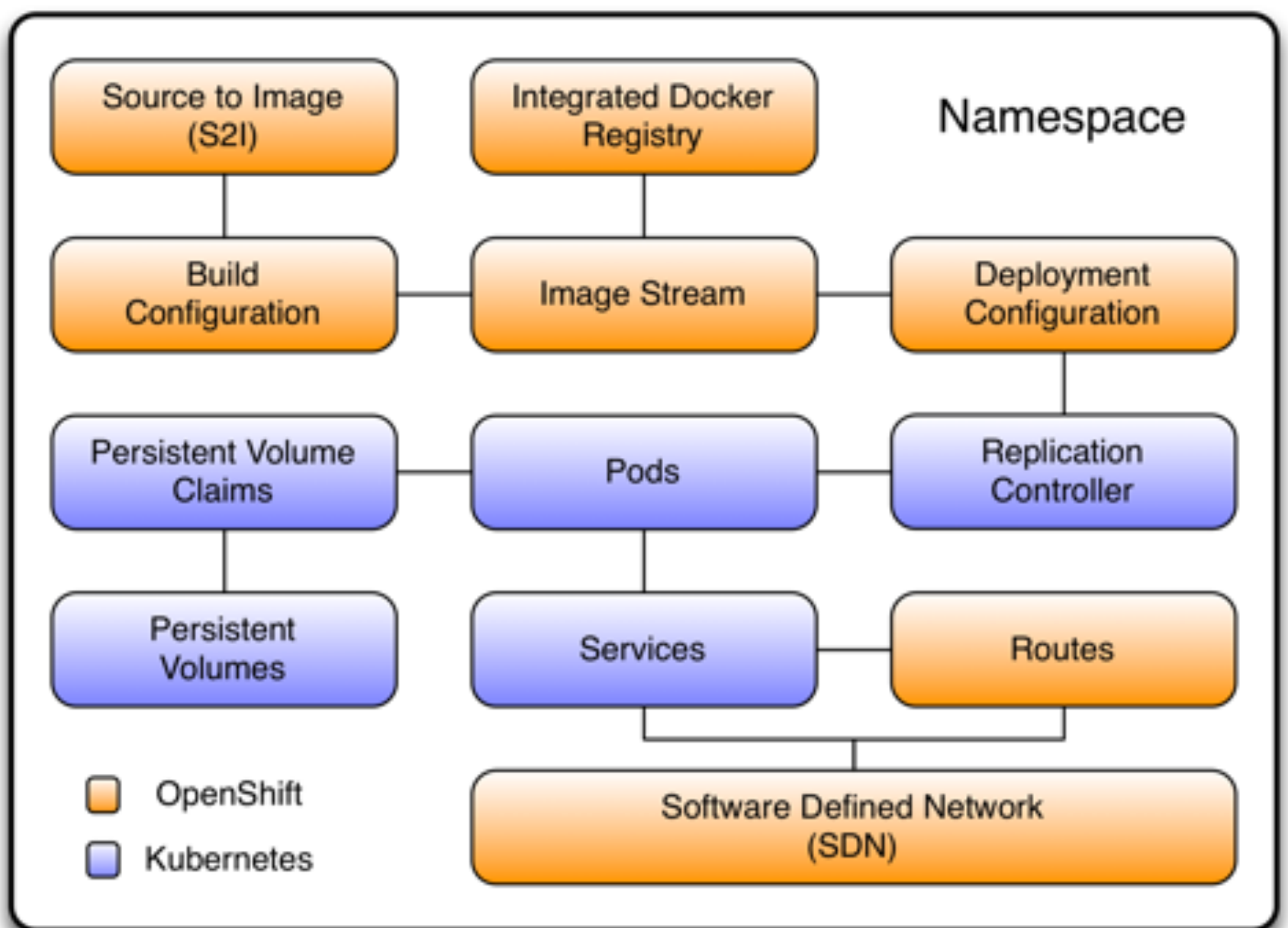Figure 1. below shows Kubernetes components in purple and OpenShift components in orange.

Figure 1. OpenShift project (Namespace)

Routes/Routers

The first block that I didn't recognize in the diagram is the orange "Routes" block. Looking at the Red Hat documentation:

"An OpenShift Enterprise route exposes a service at a host name, like www.example.com, so that external clients can reach it by name."

"An OpenShift Enterprise administrator can deploy routers to nodes in an OpenShift Enterprise cluster, which enable routes created by developers to be used by external clients."

To clarify, the default Router in Openshift is an actual HAProxy container providing reverse proxy capabilities.

The route is an Openshift construct that defines the rules you want to apply to incoming connections.

This matches the Ingress (https://kubernetes.io/docs/user-guide/ingress/) and Ingress Controller (https://kubernetes.io/docs/user-guide/ingress/#ingress-controllers) resources in Kubernetes.

Where in Kubernetes the Ingress Controller could be a NGINX container providing reverse proxy capabilities, and the Ingress Resource defines the connection rules.

In short, an Ingress is a collection of rules that allow inbound connections to reach the cluster services.

Before you start using the Ingress resource, you need to create an Ingress controller to satisfy an Ingress—creating the Ingress resource will have no effect without an Ingress controller.

The table below maps the Kubernetes name to the OpenShift name:

Kubernetes	OpenShift
Ingress resource (rules)	Route (rules)
Ingress Controller (NGINX container)	Router (HAProxy container)

Project

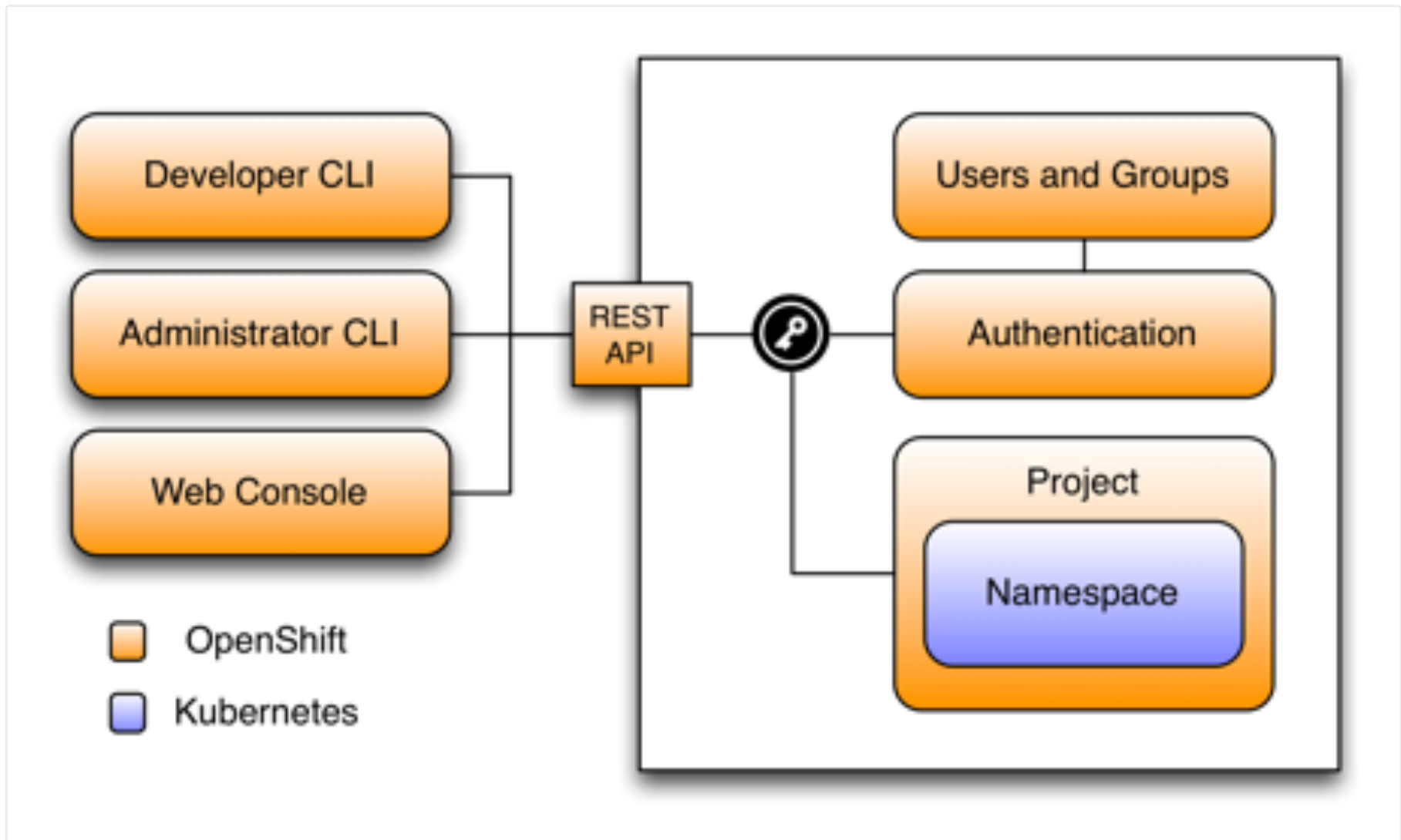Figure 2. shows the next construct in OpenShift, the project, which I did not understand:

Figure 2. OpenShift access and control

I found the answer in the book *OpenShift for Developers, A Guide for Impatient Beginners* by Grant Shipley and Graham Dumpleton.

The following is quoted from the book verbatim:

namespace

"The primary grouping concept in Kubernetes is the namespace. Namespaces are also a way to divide cluster resources between multiple uses. That being said, there is no security between namespaces in Kubernetes; if you are a "user" in a Kubernetes cluster, you can see all the different namespaces and the resources defined in them."

project

"The first new concept OpenShift adds is project, which effectively wraps a namespace, with access to the namespace being controlled via the project. Access is controlled through an authentication and authorization model based on users and groups. Projects in OpenShift therefore provide the walls between namespaces, ensuring that users, or applications, can only see and access what they are allowed to."

So, projects are namespaces with security annotations.

Summary

OpenShift route exposes a service at a host name, like www.example.com, so that external clients can reach it by name in a mature fashion. The route function is mature in OpenShift and easy to implement.

Ingress and Ingress Controller are still a beta resource at time of writing and should not be considered stable for production use.

Projects enables multitenant use of an OpenShift cluster with access privileges determined by the identity of the user or the team they belong to.

Links:

- Docs, OpenShift (https://docs.openshift.com/enterprise/3.2/architecture/core_concepts/routes.html)

- Kubernetes, Ingress User Guide (http://kubernetes.io/docs/user-guide/ingress/)

James Buckett (/author/james-buckett)

Cloud Consultant

James recently joined Levvel as a Cloud Consultant with a particular focus on Kubernetes and associated technologies. Before joining Levvel, James worked for IBM Professional Services in the IBM Cloud Division.

🐦 (http://twitter.com//jamesbuckett)**in** (https://www.linkedin.com/in/jamesbuckett/)

Related Posts

NOVEMBER 29, 2017

My Journey with Red Hat CloudForms 4.2, a Cloud Management Platform (/our-ideas/Red-Hat-CloudForms-Journey)

AUGUST 21, 2017

Webinar: A Macro View on Microservices (/our-ideas/Macro-View-on-Microservices-Webinar)

(/our-ideas/hipaa-phi-software)

AUGUST 16, 2016

[HIPAA, PHI, and Software (/our-ideas/hipaa-phi-software)](/our-ideas/hipaa-phi-software)

© Levvel, LLC 2017

(/)

About Us (/about-us)

What We Do (/what-we-do)

Our Work (/our-work)

Our Ideas (/our-ideas)

Happenings (/happenings)

Contact (/contact)

 Twitter (https://twitter.com/getlevvel)

 Facebook (https://www.facebook.com/getlevvel)

 Linkedin (https://www.linkedin.com/company/levvel.io/)

M [Medium (https://medium.com/levvel-consulting)](https://medium.com/levvel-consulting)

hello@levvel.io

(980) 278-3065