OpenShift and Kubernetes: What's the difference?

February 5, 2019 Brian ‘Redbeard’ Harrington

SHARE

Back to all posts

Tags: *Containers*

Having worked with OpenShift for about six years now, I've received a lot of questions about what OpenShift is. With the 3.x shift to Kubernetes from the previous architecture this evolved from what is it, to what the differences between it and Kubernetes are. Today, we're going to dive into that a bit.

Kubernetes is the “kernel”

At CoreOS we considered Kubernetes to be the "kernel" of distributed systems. We recognized that a well designed job scheduler, operating across multiple machines, capable of reconciling the state of managed workloads would naturally foster collaboration much in the same way that the Linux kernel did for the scheduling workloads on a single host. Following this logic, we knew that products would differentiate themselves based on the concerns important to their users.

It’s the same Linux kernel running in many phones, laptops, servers, and even the Raspberry Pi, albeit with varying degrees of patches to support the hardware the kernel sits directly on.

It's within this same model that it's the same Kubernetes in the various Kubernetes distributions, albeit with varying degrees of patches to support the layer Kubernetes sits directly atop. The Linux distributions that each flavor of Kubernetes is running its workloads on.

OpenShift is the distribution

This is a powerful distinction. The team behind OpenShift has been proud to produce a distribution of Kubernetes focused on the experience of developers who have the need to develop the next generation of cloud native applications. The team behind Tectonic (the CoreOS distribution of Kubernetes) focused on the experience of administrators and operations teams who needed to quickly remediate issues with the operating system and Kubernetes itself. With the forthcoming release of OpenShift 4.0, we're providing interfaces for both audiences so that we have a platform catering to these specialized needs.

While anyone could build Linux from Scratch by choosing each piece and assembling them in the bespoke manner each user chooses, most do not. The level of abstraction that most users choose means that they don't derive a lot of value from managing (or even knowing about) the differences between Util-Linux version 2.31 and 2.33. To take this a step further, user care about a minimum level of functionality (e.g. they know which commands/APIs will be available as long as they surpass a minimum version number) and then a list of features provided.

This is much the same as OpenShift. We package Kubernetes and include additional tooling as features that we find important and our users demand. Much as CoreOS and CentOS contain different sets of tooling, catering to different users, so it is the same with Kubernetes distributions. At Red Hat we have focused on making available the tools that help make developers and operations teams successful. This is why, for example, we are including Istio as a technology preview in OpenShift now. We feel it is a tool many users may come to depend on, and thus should be included as table stakes in the base distribution.

OKD vs Red Hat OpenShift

Let's rewind though.

Is OpenShift open source software? Absolutely! All of the components within OpenShift are developed within the open source community and can be viewed on GitHub. You'll find a large number of repositories there spanning many of the concerns of keeping a Kubernetes cluster up and running.

We package the software components needed to run Kubernetes into a project. This project is a distribution of Kubernetes called OKD, previously called "Origin". In this way, Kubernetes and OKD are similar in that they are both open source projects, with Kubernetes being one of a number of upstream projects to OKD, similar to the Linux kernel, GNU Bash, GCC, and the Apache httpd server being upstreams of the Fedora Linux distribution. When we want to improve or add features to OpenShift, we do that work upstream if it should land in Kubernetes and we work from Kubernetes releases when creating OpenShift.

Red Hat then packages the project OKD, along with a number of other projects like Maistra, various operators, and other resources into Red Hat OpenShift Container Platform the product. After the work of cutting a release of Kubernetes ends, the work of packaging OKD and then later OpenShift begins.

Red Hat OpenShift builds on all of this, undergoing extensive internal testing to verify that all components are well integrated and teams are prepared to support the needs of our customers running the software in production. This is part of the reason why there is a gap between the release of the upstream and the subsequent enterprise ready release of OpenShift: internal enablement. Our customers want to stand on the shoulders of our expertise, knowing that we can provide end to end support for the components we ship within OpenShift.

In the same way you can build Linux from Scratch one can optimize to install Kubernetes the hard way, but this exercise is best left to the individuals with the time, patience, and level of risk that does not necessitate enterprise support. For those focused on their own applications who want to stand on the shoulders of Red Hat, we recommend you chose OpenShift Container Platform.