OpenShift Overview

OpenShift is a platform that allows you to run containerized applications and workloads, and its powered by Kubernetes under the covers.

There are different flavors that are available with OpenShift

The open source project that actually powers OpenShift is called OKD, or Original Kubernetes Distribution, and as it is open source we can start with it for free today.

OpenShift on the other hand has multiple flavors and its an offering that comes with Red Hat support regardless of where you choose to run your applications and workloads.

Lets take a look at the architecture of OpenShift.

One of the biggest advantages is being able to take advantage of public or private resources for running OpenShift.

That includes bare-metal or virtualized hardware, whether its on-premises or on a cloud provider.

On top of that we are gonna have the OS, which is generally RHEL or CentOS (Working with OKD).

On top of that we are gonna have Kubernetes.

Finally on top of K8 we'll have OpenShift and makes working with it easier.

**OpenShift takes lot of the difficult tasks like deploying app and doing things like day-to-day operations, easier by building a web console in CLI, as well as a façade on top of K8's to make all of those tasks just a little bit more streamlined and easy to do.**

2 types of engineers who really benefit from taking advantage of OpenShift.

Developer: Day-to-day job

Write applications, Create changes, test them out, deploy them into cluster. Any kind of distractions will slow a developer

OpenShift is the number 1 Enterprise Kubernetes platform right now.

OpenShift is built on top of several components and setting up a cluster is bit difficult.

Especially for those without infrastructure knowledge.

What if I need to learn OpenShift and Test case functionalities.

RedHat + Kata Koda platform (Redhat interactive learning portal is available at free of cost)

KataKoda is a platform for creating real life lab environments where

<https://miro.com/app/board/uXjVKESciOk=/>

OpenShift Vs Kubernetes

One of the key features that OpenShift is going to give developers is "Speed of development".

Particularly with the steps of creating and building the cloud-native application, which is manifested as containers.

And the feature in OpenShift that makes this possible is "Source-to-Image"(S2I).

A diagram of a software project

Description automatically generated

Key Feature of OpenShift = Speed of development

Where is this observed? = particularly with the steps of creating and building the cloud-native application, which is manifested as containers.

Who makes this possible? = Source to Image(in OpenShift). S2I.

Lets consider from the above diagram, A developer writes code and pushes into repository (GIT), Once the code is pushed into repo now S2I will pitch in. S2I is going to look into the repository and it is able to determine the language we wrote the code in (Java, python etc..), Referring the repository it is now smart enough to know the type of base image that we need to pick.

And S2I is going to pull that determined image from the image catalog. Now S2I is going to take the code we wrote and its going to combine with the base image that was appropriate for the code, whether we wrote java or python code etc.. And with this, its going to build a new image. And its going to put that new image inside a registry.

And then it pushes it for deploying.

Why was there such a great feature? What is its use?

Lot of the steps in the above mentioned process like, selecting an image, merging that image with code and building new image, typically one have to write a container script-building files. Sometimes called docker files, to do all these steps. For a seasoned developer these steps are not hard, However if the developer is not familiar with these steps or if he is less experienced or some one who is not familiar with cloud native development experience or how to do container type of activities, building containers and picking an image and putting it and merging it with the code they wrote and then pushing it to registry, Having OpenShift automate all these steps for us is a huge help.

So these are the key features that OpenShift provides for new developers.

Another feature that OpenShift provide is Easy networking through the use of routes.

In vanilla Kubernetes networking is very complicated.

OpenShift takes care of networking piece for us and removes this complication in Kubernetes.

It gives us easy to read URL that maps deployed application.

It also provides integration with external load balancers.

It'll take care of load balancing integration with external load balancer and it also gives different load balancing options.

To do all these networking steps in Kubernetes its very complicated and we need to very proficient in that.

Open Shift Vs Kuberneetes

OpenShift is a platform to run our containerized applications, and its built on top of open source K8S engine.

Installation:

Open shift can be installed in one of the below platforms and it cannot be installed in other linux distros.

* 1. Red Hat enterprise Linux (RHEL) or Red Hat Atomic on OpenShift 3.
  2. Red Hat CoreOS (required by control plane - master and infra server, the default for compute nodes) and optionally RHEL for compute nodes only on OpenShift 4.
  3. RHEL on CentOS for OKD.

On Contrary K8S can be installed on most of the platforms! There are many tools available for the same too.

Kubeadm being the simplest and most used one, there are few other tools like Kube-spray, kops, Booktube etc.

Kubernetes Vs OpenShift

Imagine yourself as a tech-head of e-Commerce start up and your team has been working overtime to package/ bundle- up your main apps into containers and now is the time to pick a platform to handle and manage them smoothly.

You have 2 options:

K8S :- Open source container management system.

OpenShift: The enterprise Kubernetes distribution from RedHat

How to choose the right one that fits our requirement?

This notes is targeted towards someone new to the world of containers and cloud native technologies.

This also highlights the key difference between these 2 and further understanding the main differences between K8S and OpenShift platforms.

We'll also see how the deployment process differs the tooling and interfaces provided for managing applications and monitoring and we'll also see the security and compliance features that each platform offers.

In this notes we'll also cover pros and cons of K8S vs OpenShift and be able to make more informed decision on which platform might be better fit for our particular needs and requirements.

What is Kubernetes:

Kubernetes is just like a master conductor (orchestrator) for the containers, think of it as this super smart system that helps you manage all the containerized apps smoothly, especially when you got loads of them, its open source which means its free to use and has become the go to choice for handling containers on a big scale.

With K8s we get this awesome toolkit for automatically deploying scaling and looking after all those containerized workloads.

To deploy an app on k8s you'd create a yaml manifest files that describe the different components the container image scaling requirements, networking configuration and so on.

si 
Introduction to Kubernetes 
Deployment 
kubernetes 
To deploy an app on 
Kubernetes we need to 
create YAML files that 
describe the different 
components 
Container Image 
Scaling requirements 
Networking Configuration 
Kubernetes is an open-source container orchestration platform that 
simplifies managing and scaling containerized applications 

This flexibility is great as it allows you to customize K8S to our hearts content but it also means there's fair amount of complexity involved especially when it comes to setting up supporting tools for monitoring, logging, CICD pipelines and more.

Case study or real-time example:

Lets say we want to deploy our e-Commerce Web application on K8S.

Introduction to Kubernetes 
Cl / CD Workflow 
Prometheus 
simplilearr 
Grafana 
Kubernetes dashboard gives you visibility into the state of your deployment, but 
to really monitor and manage your apps, you'd likely need to integrate additional 
tools like Prometheus, Grafana, and custom CI/CD workflows 

Step 1:

Provision a K8S cluster either on-Prem or on a cloud provider.

Step 2:

Write necessary YAML manifest to describe your app and apply those to the cluster.

Step 3:

K8S dashboard gives visibility into the state of your deployment, but to really monitor and manage your apps you likely need to integrate additional tools like Prometheus, Grafana and Custom CICD workflows

Difference between OpenShift and K8S in terms of deployment and Automation.

Deployment and Automation 
simplilea 
Kubernetes 
Kubernetes requires more hands-on work to 
set up CI/CD pipelines 
Configuring networking in Kubernetes can 
be more complex 
Developers and DevOps teams often need 
to write more configurations 
Developers need to integrate various tools 
and services e.g., Jenkins and Tekton 
OpenShift 
OpenShift, which is built on top of 
Kubernetes, provides a lot of deployment 
Includes a user-friendly web console and 
command-line interface 
Automation features such as image builds, 
and scaling can reduce the manual effort 
Deployment and automation tools e.g., 
Jenkins, Tekton are pre-integrated 

1. K8S requires more hands-on work to setup CICD pipelines and streamline the deployment process Whereas OpenShift which is built on top of K8S, provides a lot of deployment and automation functionality out of the box.
2. Configuring n/w ing in K8s can be more complex as we need to setup load balancers and other n/w ing components to expose our applications to the outside world. On the other hand OpenShift includes a user-friendly web console and command line interface that simplify the management of CICD pipelines n/w ing and other deployment related tasks.
3. In case of K8S developer teams often need to write more custom scripts and configurations to automate the deployment process. Whereas in OpenShift many of the commonly used deployment and automation tools like Jenkins, tekton are pre-integrated and configured in open-shift making it easier to setup a robust software delivery pipeline
4. Setting up and managing a robust CICD pipeline in K8S requires more effort as we need to integrate various tools and services (Ex: Jenkins, Argo CD and Tekton to handle the different stages of the software delivery lifecycle. Whereas OpenShifts deployment automation features such as image builds automatic scaling can significantly reduce manual effort required to manage the deployment process.

Difference between K8S and OpenShift in-terms of application management.

Application Management 
simplilea 
Kubernetes 
Provides basic visibility and management 
capabilities for your applications 
Allows you to view running pods, services 
and perform some basic administrative tasks 
To understand application performance, we 
need to integrate external monitoring tools 
The dashboard has limited functionality 
and doesn't provide a comprehensive view of 
your application portfolio 
OpenShift 
Provides a more robust and unified view of 
your application management 
Offers a comprehensive dashboard 
displaying information of your applications 
Includes features like monitoring, 
deployment and security scanning 
Provides advanced features such as ability t 
view application dependencies 

Security and Compliance 
simplilek 
Kubernetes 
It made progress with features like Role- 
Based Access Control and Network Policies 
Allows you to define fine-grained 
permissions for users and service accounts 
Provides a way to set security-related 
constraints on pods 
Network Policies allow you to control the 
network traffic improving the overall security 
OpenShift 
Inherits all these security features adding 
enterprise-grade security capabilities 
Provides more secure container defaults, suc 
as running containers as a non-root user 
Integrates with enterprise identity 
management systems and Active Directory 
Includes security scanning helping you 
identify and address potential security risks 