

Research Paper on Redhat Openshift Cloud Platform

Akshay Kushwaha

Department of Information Technology, KeraleeyaSamajam's Model College, Dombivli West, Mumbai, Maharashtra

Abstract: OpenShift is a cloud development Platform as a Service (PaaS) developed by Red Hat. It is an open-source development platform, which enables the developers to develop and deploy their applications on cloud infrastructure. It is very helpful in developing cloud-enabled services. This research paper will help you understand OpenShift and how it can be used in the existing infrastructure.

I. Introduction:

OpenShift is a cloud development Platform as a Service (PaaS) hosted by Red Hat. It's an open-source cloud-based user-friendly platform used to create, test, and run applications, and finally deploy them on cloud.

OpenShift is capable of managing applications written in different languages, such as Node.js, Ruby, Python, Perl, and Java. One of the key features of OpenShift is it is extensible, which helps the users support the application written in other languages.

OpenShift comes with various concepts of virtualization as its abstraction layer. The underlying concept behind OpenShift is based on virtualization.

II. History of OpenShift

OpenShift came into existence from its base named OpenShift V2, which was mainly based on the concept of gear and cartridges, where each component has its specifications starting from machine creation till application deployment, right from building to deploying the application.

Cartridges – They were the focal point of building a new application starting from the type of application the environment requires

to run them and all the dependencies satisfied in this section.

Gear – It can be defined as the bare metal machine or server with certain specifications regarding the resources, memory, and CPU. They were considered as a fundamental unit for running an application.

Application – These simply refer to the application or any integration application that will get deployed and run on OpenShift environment.

As we go deeper in the section, we will discuss on different formats and offerings of OpenShift. In the earlier days, OpenShift had three major versions.

OpenShift Origin – This was the community addition or open-source version of OpenShift. It was also known as upstream project for other two versions.

OpenShift Online – It is a public PaaS as a service hosted on AWS.

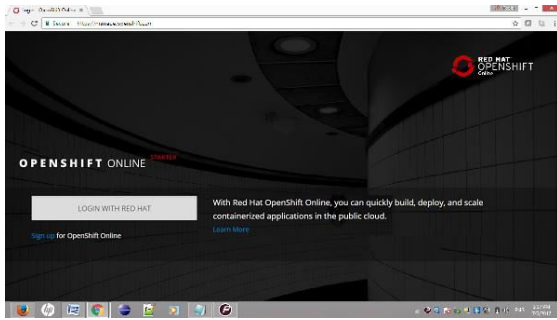
OpenShift Enterprise – is the hardened version of OpenShift with ISV and vendor licenses.

- **OpenShift Online**

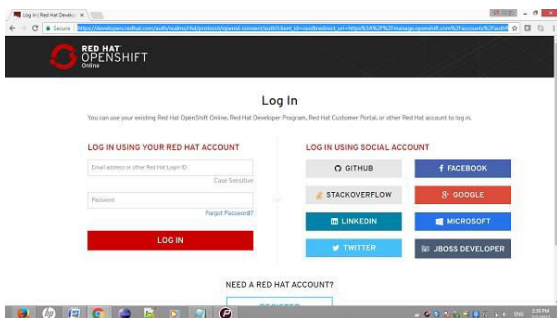
OpenShift online is an offering of OpenShift community using which one can quickly build, deploy, and scale containerized applications on the public cloud. It is Red Hat's public cloud application development and hosting platform, which enables automated provisioning, management and scaling of application which helps the developer focus on writing application logic.

- **Setting Up Account on Red Hat OpenShift Online**

Step 1 – Go to browser and visit the site <https://manage.openshift.com/>

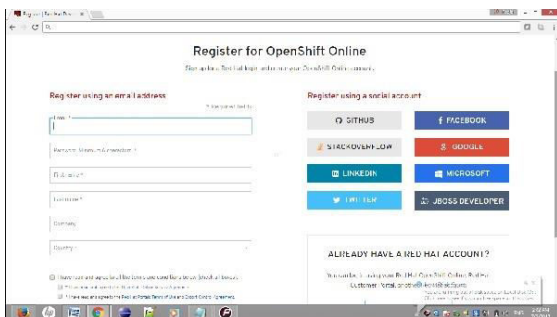


Step 2 – If you have a Red Hat account, login to OpenShift account using the Red Hat login ID and password using the following URL. <https://developers.redhat.com>

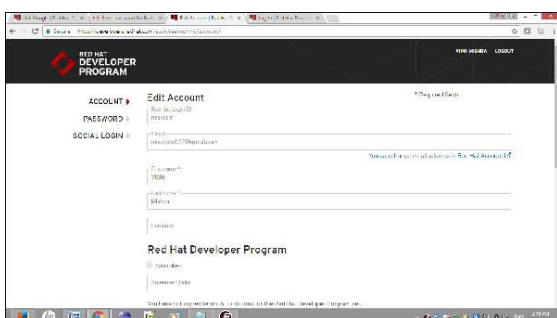


Step 3 – If you do not have a Red Hat account login, then sign up for OpenShift online service using the following link.

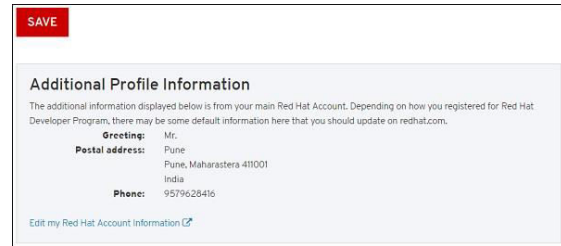
https://developers.redhat.com/auth/realms/rhd/login-actions/registration?code=G4w-myLd3GCH_QZCqMUmIOQIU7Dif_gfIvGu38nnzZQ.cb229a9d-3cff-4c58-b7f6-7b2c9eb17926



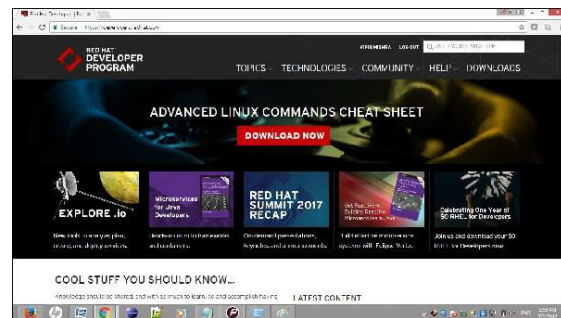
After login, you will see the following page.



Once you have all the things in place, Red Hat will show some basic account details as shown in the following screenshot.



Finally, when you are logged in, you will see the following page.



• **OpenShift Container Platform**

OpenShift container platform is an enterprise platform which helps multiple teams such as development and IT operations team to build and deploy containerized infrastructure. All the containers built in OpenShift uses a very reliable Docker containerization technology, which can be deployed on any data center of publically hosted cloud platforms.

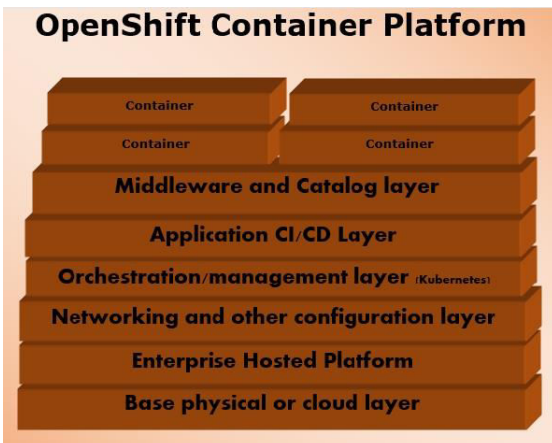
OpenShift container platform was formally known as OpenShift Enterprises. It is a Red Hat on-premise private platform as service, built on the core concept of application containers powered by Docker, where orchestration and administration is managed by Kubernetes.

In other words, OpenShift brings Docker and Kubernetes together to the enterprise level. It is a container platform software for enterprise units to deploy and manage applicants in an infrastructure of own choice. For example, hosting OpenShift instances on AWS instances.

OpenShift container platform is available in **two package levels**.

OpenShift Container Local – This is for those developers who wish to deploy and test applications on the local machine. This package is mainly used by development teams for developing and testing applications.

OpenShift Container Lab – This is designed for extended evaluation of application starting from development till deployment to pre-prod environment.



- **OpenShift Dedicated**

This is another offering added to the portfolio of OpenShift, wherein there is a customer choice of hosting a containerized platform on any of the public cloud of their choice. This gives the end user a true sense of multi-cloud offering, where they can use OpenShift on any cloud which satisfies their needs.

This is one of the newest offering of Red Hat where the end user can use OpenShift to build test deploy and run their application on OpenShift which is hosted on cloud.

- **Features of OpenShift Dedicated**

OpenShift dedicated offers customized solution application platform on public cloud and it is inherited from OpenShift 3 technology.

- **Extensible and Open** – This is built on the open concept of Docker and deployed on cloud because of which it can expend itself as and when required.

- **Portability** – As it is built using Docker, the applications running on Docker can easily be shipped from one place to the other, where Docker is supported.

- **Orchestration** – With OpenShift 3, one of the key features of container orchestration and cluster management is supported using Kubernetes which came into offering with OpenShift version 3.

- **Automation** – This version of OpenShift is enabled with the feature of source code management, build automation, and deployment automation which makes it very popular in the market as a Platform as a Service provider.

- **Competitors of OpenShift**

Google App Engine – This is Google's free platform for developing and hosting web applications. Google's app engine offers fast development and deployment platform.

Microsoft Azure – Azure cloud is hosted by Microsoft on their data centers.

Amazon Elastic Cloud Compute – They are built-in services provided by Amazon, which help in developing and hosting scalable web applications on cloud.

Cloud Foundry – is an open source PaaS platform for Java, Ruby, Python, and Node.js applications.

CloudStack – Apache's CloudStack is a project developed by Citrix and is designed to become a direct competitor of OpenShift and OpenStack.

OpenStack – Another cloud technology provided by Red Hat for cloud computing.

Kubernetes – It is a direct orchestration and cluster management technology built to manage Docker container.

III. Why use openshift over Kubernetes?

- **User Interface**

OpenShift offers a default feature-rich graphical interface that allows you to build, deploy, scale, monitor and implement almost every Kubernetes task from a web browser, both as an administrator and developer.

This makes OpenShift simple and easy to use, and time-saving for every developer considering the short learning curve.

Using Kubernetes, the dashboard UI is not deployed by default. You have to deploy and access it by running different commands manually. It will require you to learn some commands before you're able to use them, tiring with the longer learning curve.

- **Cloud Platform Agnostic**

Even though Kubernetes is cloud-agnostic (that is, you can use it with various cloud providers), it isn't as cloud agnostic as OpenShift when compared.

To improve productivity and even protect your interests as a developer, you would want the freedom to migrate your application between different cloud service providers without having to tweak or make changes to your application infrastructure.

Different cloud service providers, AWS, IBM Cloud, Google Cloud Platform, Azure, all offer different Kubernetes managed services whose sets of add-ons, plugins, and usage instructions are specific to each cloud platform. This will require you to get familiar with each cloud service's managed Kubernetes service to understand how things are done before moving between them.

OpenShift is a "build once and deploys anywhere" container platform that allows you to build, deploy and manage your application with the same web interface across all cloud service platforms.

- **Automated Workflow**

When deploying your application, Red Hat OpenShift offers several automated workflows which are not offered in Kubernetes.

OpenShift supports several programming languages, including Python, Java, PHP, Go, Node.js, and Ruby. It provides pre-created application templates in many of these languages and frameworks to quickly kickstart your application development. In the past, Kubernetes templates are way better than OpenShift templates. It has in its new version. It provided two templates you can use when deploying an app, Kubernetes Helm Charts and OpenShift template. This gives the developer a nicer range of template options.

OpenShift also has an in-built image registry that uses the source-to-image feature that automatically pairs OpenShift with DockerHub or Red Hat and allows you to manage container images using image streams easily. You can also use your private image registry with OpenShift.

Kubernetes, on the other hand, does not offer this kind of integrated image management, and it also requires manual configuration for many of its workflows.

In October 2020, Red Hat announced the integration of Ansible Automation with OpenShift. This will improve automation and also amenable enterprises to use Ansible for deployment and management of your application.

Red Hat's OpenShift does a lot of heavy lifting and automation in the background, making the development, testing, deployment, and production environments of your application much easier with the release of OpenShift 4, an improved version of OpenShift.

- **Continuous Integration/Continuous Delivery (CI/CD) Solution**

Coupled with automation at various levels of application, OpenShift offers seamless integration with CI/CD pipeline building tools like Jenkins.

Even though it's not a complete CI/CD solution, OpenShift provides a certified Jenkins container integration that can be used with other tools to set up a complete CI/CD infrastructure on the OpenShift platform.

Below is an image of a continuous integration and deployment infrastructure built on OpenShift using Jenkins, Gogs, and other tools.

In the case of Kubernetes, it doesn't offer an integrated CI/CD solution. Still, you can use different third-party automated monitoring, testing, and CI/CD tools and plugins like CircleCI to build a CI/CD pipeline on Kubernetes.

- **Security**

When working on a team of developers with different permissions on an application, there is a need to properly secure each team member's access and permission to the containerization platform to avoid possible attacks.

By default, OpenShift offers the role-based access control (RBAC) that helps ensure that each developer only gets permission to the features they need to avoid the issue of account compromise.

When you create a project with OpenShift, other security rules like IAM and OAuth are all created by default. You only need to add user permissions as required. This makes your application environment easier to set up and saves you time.

On the Kubernetes side, even though it now offers Kubernetes RBAC, you still have to set

everything up by yourself, including other security protocols like IAM and OAuth.

OpenShift also has a kind of stricter security compared to Kubernetes. It forbids running a container as root and restricts Docker containers' running as simple images, all to avoid compromise in security.

Recently, Red Hat announced that it had acquired StackRox - a Kubernetes native security platform, to bring more advanced security to OpenShift.

- **Node Configuration**

It is only normal that you have many virtual machines in your cluster, whether virtualized or bare metal. When an application requires an additional virtual machine, Kubernetes' handling the process is a little complicated and time-consuming.

It requires you to develop scripts to set up self-registration or different cloud automation before creating new VMs then bringing them into the cluster. This process can be a little tiring and time-consuming.

Red Hat's OpenShift addresses this in a much easier way. You can use the provided Ansible playbooks and installers to bring new virtual machines into the cluster easily.

It also provides auto-scaling, which enables the automatic addition of new virtual machines into the cluster to increase workload.

- **Operating System Support**

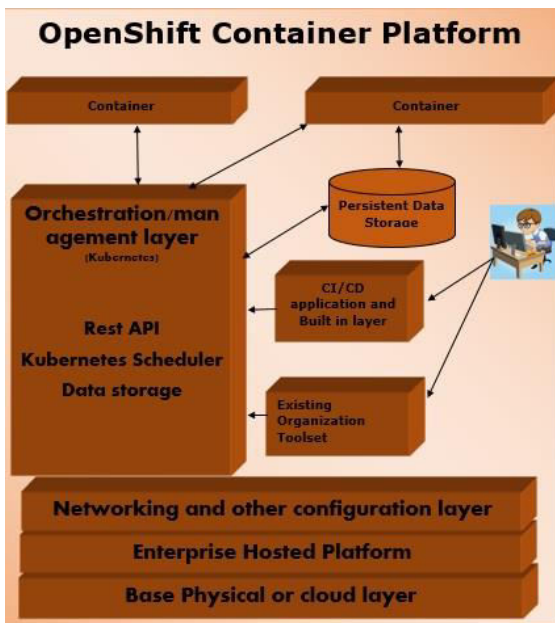
This is pretty much not an advantage of OpenShift over Kubernetes. In fact, Kubernetes supports all major operating systems, while OpenShift used to be limited to Linux-based containers.

But things changed when Red Hat announced in December 2020 that its OpenShift platform would be supporting Windows-based containers from 2021. This will allow

companies to run both Linux and Windows environments as the DevOps industry adopts a hybrid cloud.

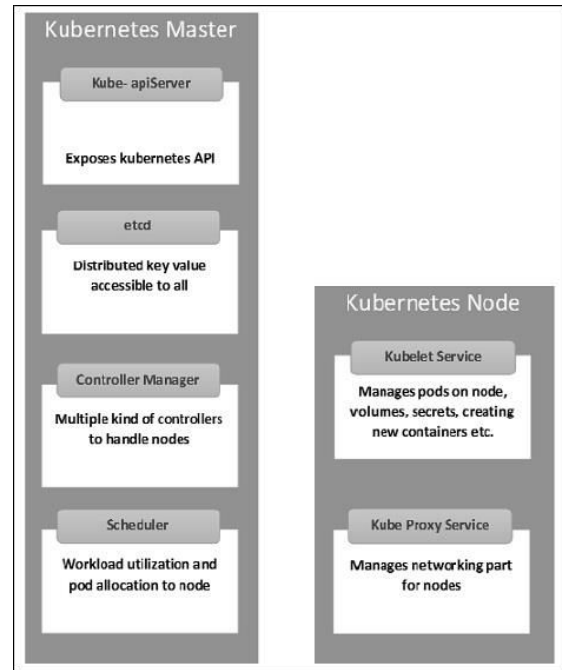
IV. How OpenShift works?

OpenShift is a layered system wherein each layer is tightly bound with the other layer using Kubernetes and Docker cluster. The architecture of OpenShift is designed in such a way that it can support and manage Docker containers, which are hosted on top of all the layers using Kubernetes. Unlike the earlier version of OpenShift V2, the new version of OpenShift V3 supports containerized infrastructure. In this model, Docker helps in creation of lightweight Linux-based containers and Kubernetes supports the task of orchestrating and managing containers on multiple hosts.



- Components of OpenShift

One of the key components of OpenShift architecture is to manage containerized infrastructure in Kubernetes. Kubernetes is responsible for Deployment and Management of infrastructure. In any Kubernetes cluster, we can have more than one master and multiple nodes, which ensures there is no point of failure in the setup.



- Kubernetes Master Machine Components

Etcd – It stores the configuration information, which can be used by each of the nodes in the cluster. It is a high availability key value store that can be distributed among multiple nodes. It should only be accessible by Kubernetes API server as it may have sensitive information. It is a distributed key value Store which is accessible to all.

API Server – Kubernetes is an API server which provides all the operation on cluster using the API. API server implements an interface which means different tools and libraries can readily communicate with it. A kubeconfig is a package along with the server side tools that can be used for communication. It exposes Kubernetes API”.

Controller Manager – This component is responsible for most of the collectors that regulate the state of the cluster and perform a task. It can be considered as a daemon which runs in a non-terminating loop and is responsible for collecting and sending information to API server. It works towards getting the shared state of the cluster and then make changes to bring the current status of the server to a desired state. The key controllers are replication controller, endpoint controller, namespace controller, and service account

controller. The controller manager runs different kind of controllers to handle nodes, endpoint, etc.

Scheduler – It is a key component of Kubernetes master. It is a service in master which is responsible for distributing the workload. It is responsible for tracking the utilization of working load on cluster nodes and then placing the workload on which resources are available and accepting the workload. In other words, this is the mechanism responsible for allocating pods to available nodes. The scheduler is responsible for workload utilization and allocating a pod to a new node.

• Kubernetes Node Components

Following are the key components of the Node server, which are necessary to communicate with the Kubernetes master.

Docker – The first requirement of each node is Docker which helps in running the encapsulated application containers in a relatively isolated but lightweight operating environment.

Kubelet Service – This is a small service in each node, which is responsible for relaying information to and from the control plane service. It interacts with etcd store to read the configuration details and Wright values. This communicates with the master component to receive commands and work. The kubelet process then assumes responsibility for maintaining the state of work and the node server. It manages network rules, port forwarding, etc.

Kubernetes Proxy Service – This is a proxy service which runs on each node and helps in making the services available to the external host. It helps in forwarding the request to correct containers. Kubernetes Proxy Service is capable of carrying out primitive load balancing. It makes sure that the networking environment is predictable and accessible but at the same time it is isolated as well. It manages pods on node, volumes, secrets, creating new containers health checkup, etc.

• Integrated OpenShift Container Registry

OpenShift container registry is an inbuilt storage unit of Red Hat, which is used for storing Docker images. With the latest integrated version of OpenShift, it has come up with a user interface to view images in OpenShift internal storage. These registries are capable of holding images with specified tags, which are later used to build containers out of it.

V. Conclusion

Kubernetes and OpenShift are independent container technologies that are related to each other in many ways and complement each other. Kubernetes can deploy Docker images and orchestrate. Similarly, OpenShift seamlessly manages the Kubernetes cluster. With Kubernetes, one can manage hundreds of containers.

Kubernetes allows auto-scaling, load balancing and storage management features to containers. Plus, Kubernetes has extremely powerful community support. OpenShift powered with Kubernetes takes Kubernetes further and enables more simplicity in the management of workloads along with different types of tools.

ACKNOWLEDGEMENT

I would like to thank KeraleeyaSamajam's Model College for providing me with an opportunity to present this research paper. And also, I would also like to thank Divya Ma'am for assistance and comments that greatly improved the manuscript.

REFERENCES

- [1] GRANT SHIPLEY AND GRAHAM DUMPLETON, OPENSIFT FOR DEVELOPERS: A GUIDE FOR IMPATIENT BEGINNERS
- [2] RED HAT OPENSIFT A COMPLETE GUIDE - 2020 EDITION BY GERARDUS BLOKDYK