

WHITE PAPER

RedHat OpenShift Container Platform

Abstract

- Applications are designed around smaller independent components called microservices.
- The decomposition of applications into smaller components has distinct advantages of developing applications faster and delivering them in a more resilient way—both of which are better aligned with today's business needs. However, this design adds complexity in the application layer for IT operations.
- For IT operations to support this new application architecture, the underlying middleware, runtimes and other software services needed for application, the lifecycle management should be highly automated and have their complexity abstracted away.
- Standardization and platform abstraction not only make operations more efficient in handling developer needs, but they also help streamline the deployment pipeline, thereby accelerating the delivery of microservice-based applications.
- The shift to modern application development and the need for rapid and continuous deployment make a strong DevOps-enabling platform, a key component in the IT arsenal.

1.1 Introduction

Cloud Computing

Cloud computing is the delivery of on-demand computing resources, everything from applications to data centers over the internet.

Benefits:

- **Elastic resources:** Scale up or down quickly and easily to meet a demand.
- **Metered service:** You only pay for what you use.
- **Self-service:** All the required IT resources with self-service access.

Based on a cloud location, we can classify cloud as:

- **Public:** A public cloud is where users don't need to purchase hardware, software, or supporting infrastructure, which is owned and managed by the providers.
- **Private:** A private cloud is pretty much same as public cloud and has similar advantages, but is provisioned for a single organization over private infrastructure.
- **Hybrid:** As name suggests hybrid cloud is a type of cloud computing where private/on-premises cloud infrastructure integrates with public cloud services.

Based on a service that the cloud is offering they can be classified as:

- **IaaS** (Infrastructure-as-a-Service)
- **PaaS** (Platform-as-a-Service)
- **SaaS** (Software-as-a-Service)

Containerization

- A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

- Linux containers are a key building blocks for the modern platform. Today, the most popular Linux container format to build a robust DevOps environment is based on the Docker project. Docker is an open-source software container platform that makes the deployment of application easier and also run applications using containers.
- Containerization based on virtualization allows any application bundled in a container which can be run without the hassles of any dependency like libraries.
- They are a kind of isolated partition inside a single operating system.

Orchestration

- While containers are effective at the heavy lifting while instantiating an image or application component on a single hosting environment, most modern applications are comprised of many components that span multiple hosts that in turn could be in various geographic regions. This is where orchestration and management become a question.
- Applications generally are built up of individually containerized components/micro services that must be managed for the application to perform as expected.
- Container orchestration is the process of managing or organizing multiple containers in this fashion. For example: Docker Swarm, Kubernetes, Mesos and Marathon
- Kubernetes is an open source platform initially developed by engineers at Google, which automates container operations. It allows you to build application services that span multiple containers, schedule those containers across a cluster, scale those containers, and manage the health of those containers over time. Kubernetes handles container deployments and orchestration as a cluster manager by using a declarative model that enables the user to define their application needs.

DevOps

- DevOps help organizations respond in a more agile manner to changing business requirements by:
 - Automating and monitoring the process of software creation, from integration, testing, releasing to deploying and managing it.
 - Reducing the development cycles.
 - Increasing the frequency of deployment.
 - Streamlining the development and release pipeline.
- E.g.: Docker, Puppet, Kubernetes, Ansible, Chef and so on.

PaaS

- The real benefit of Cloud computing is the operational agility provided by the Cloud platform in which we don't have to worry about IT infrastructure and we can just focus on building our apps

- In PaaS the capability provided to the consumer is to deploy applications created using programming languages, libraries, services, and tools supported by the provider onto the cloud infrastructure.
- The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment
- PaaS originated from shared IT service model wherein multiple tenants ran applications on shared systems
- Containerization is an ability to virtualize the Operating System (OS) resources instead of the underlying hardware while it sits directly on top of the bare metal.

For Example: Azure's Cloud Services, Amazon Web Services (AWS)'s Elastic Beanstalk, Google's App Engine

1.2 OpenShift Container Platform

Understanding OpenShift Container Platform:

Let us now understand what OpenShift Container platform is.

- RedHat OpenShift Container Platform is a PaaS offering from RedHat.
- It is a supported distribution of Kubernetes using Docker containers and DevOps tools for accelerated application development.
- Open Shift also allows you to have highly available, self-healing, and auto-scaling applications without any of the manual setup that would typically need to be done in a traditional environment whether they're on-premise or in the public cloud.
- OpenShift includes a full complement of open source programming languages giving polyglot choice to developers.

Benefits of OpenShift:

Having understood what OpenShift is, what are the benefits of this Container platform?

- Provides a container management platform for your complex IT infrastructure.
- Kubernetes is the industry leading open source container orchestration framework and Red Hat OpenShift Container Platform is the leading enterprise distribution of Kubernetes.
- Enables your development team to focus on designing and testing applications rather than spending excessive time in managing and deploying containers.
- It includes containerization for multitenancy, automatic provisioning, automatic application scaling, continuous integration, and self-service for developers.

- Accelerate application development, test that application throughout the IT architecture without being hindered by conflicts of framework, any deployment issue or language inconsistencies.
- OpenShift accelerates development and delivery of applications from Java EE to Node. JavaScript for web, mobile and enterprise applications, and can be a key enabling technology in a DevOps transformation.

PaaS = Platform as a Service

A Cloud Application Platform



Analyzing the advantages of OpenShift over others:

OpenShift allows a variety of options to deploy application source codes. They are available in 3 forms, namely:

- OpenShift Online
- OpenShift Enterprise
- OpenShift Origin

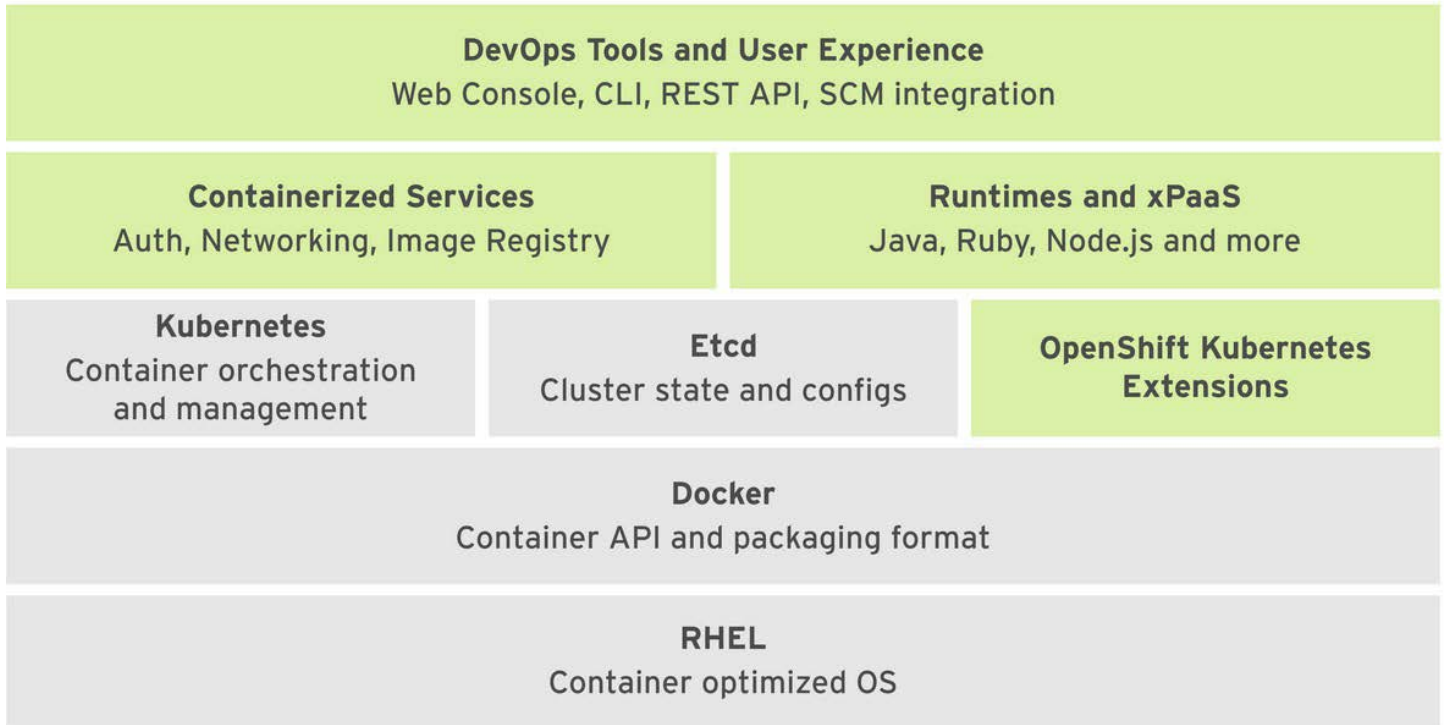
OpenShift is developed and supported by Red Hat with Kubernetes at its core. So, from a vendor support perspective, enterprises are on solid ground with OpenShift.

OpenShift has the edge in management and automation whereas other container management platforms shine with broad application support and ease of use.

Features of RedHat OpenShift:

- Software Defined Network.
- Persistent Storage.
- Container Native Storage (CNS / SDS).
- Log Aggregation and Analysis.
- Monitoring | Telemetry.
- Capacity Management.
- Egress Routing for Enterprise integration.
- Router Sharding.
- Full Stack Support.
- System Certifications and Patching.





1.3. Overview of OpenShift Architecture

OpenShift Container Platform is a set of modular components and services built on top of Red Hat Enterprise Linux, Docker, and Kubernetes. OpenShift adds capabilities such as remote management, multitenancy, increased security, application life-cycle management and self-service interfaces for developers.

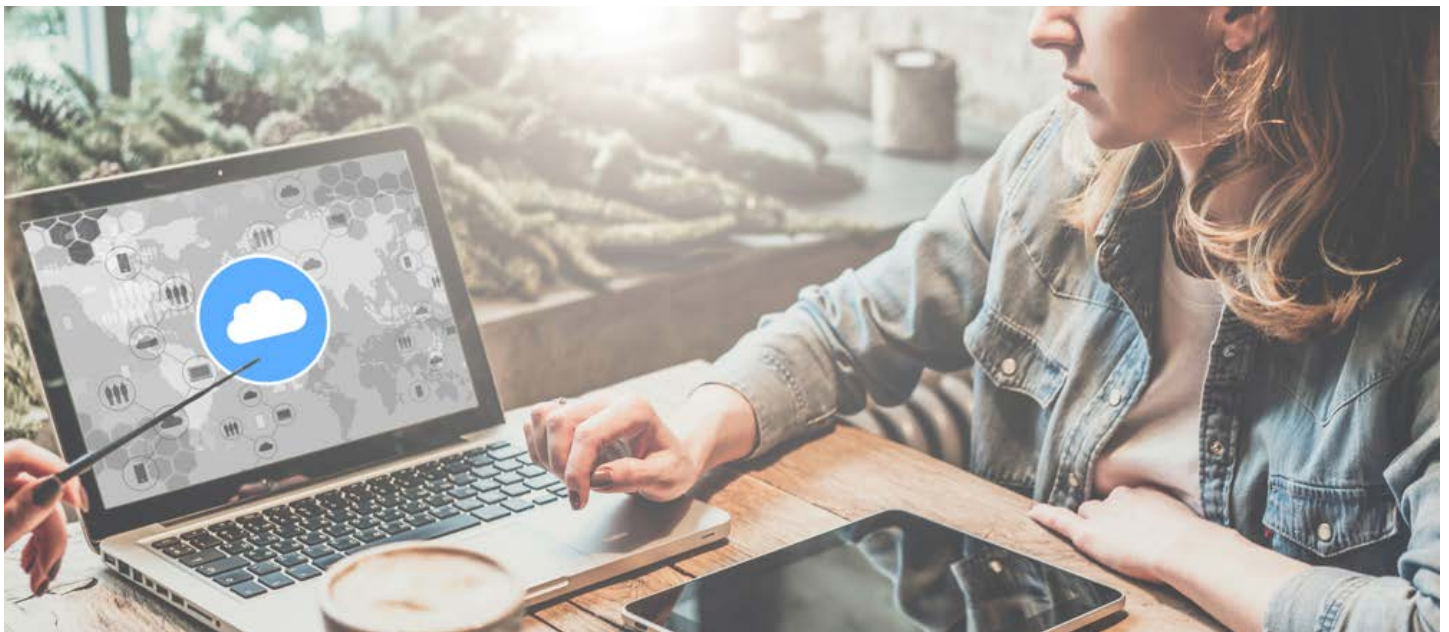
In the above figure, going from bottom to top, and from left to right, the basic container infrastructure is shown, integrated and enhanced by Red Hat:

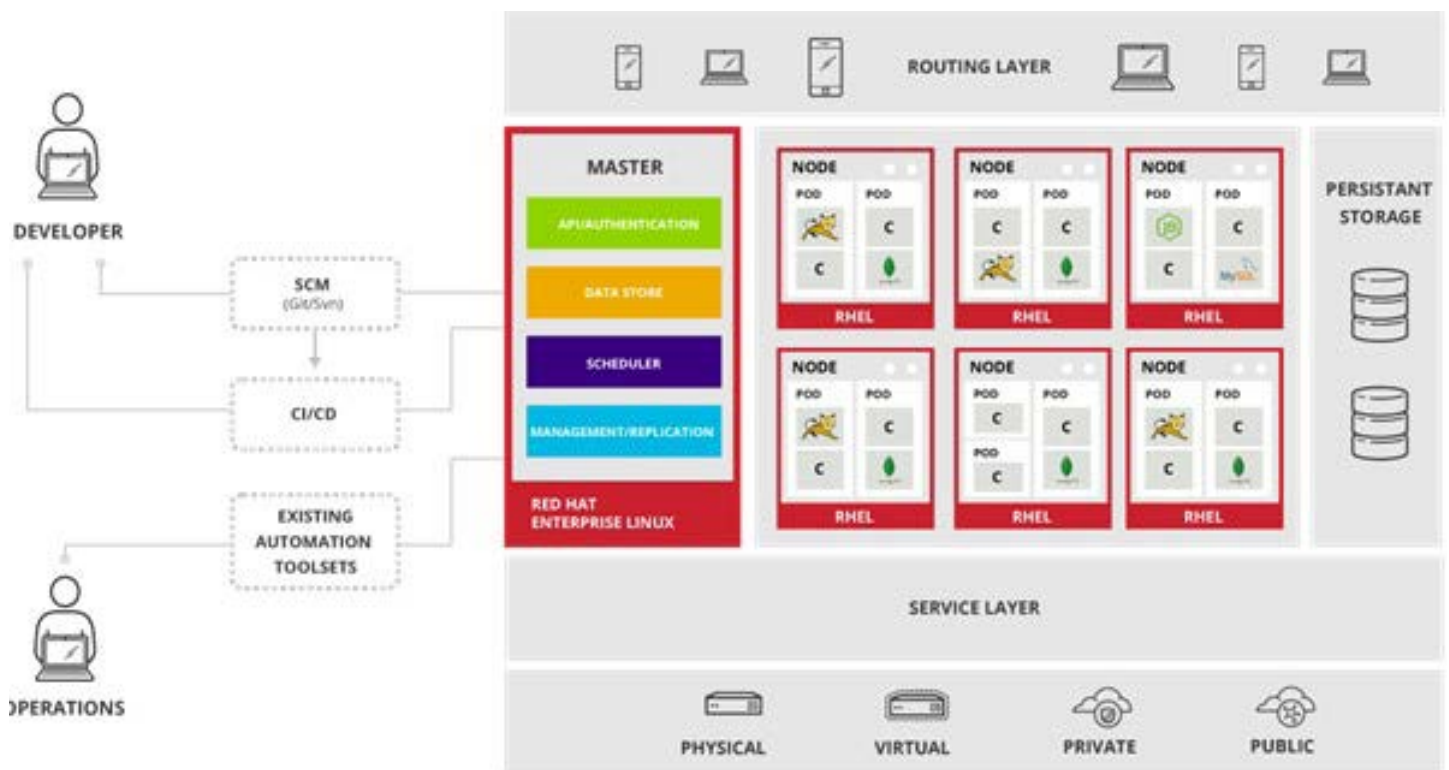
- The base OS is Red Hat Enterprise Linux (RHEL).
- Docker provides the basic container management API and the container image file format.

- Kubernetes manages a cluster of hosts (physical or virtual) that run containers. It works with resources that describe multi-container applications composed of multiple resources, and how they interconnect.
- Etcd is a distributed key-value store, used by Kubernetes to store configuration and state information about the containers and other resources inside the OpenShift cluster.

The working of RedHat Openshift:

An OpenShift cluster is a set of node servers that run containers and are centrally managed by a set of master servers. A server can act as both a master and a node, but those roles are usually segregated for increased stability.





Master: The master runs OpenShift core services such as authentication and provides the API entry point for administration.

Nodes: The nodes run applications inside containers, which are in turn grouped into Pods. This division of labor comes from Kubernetes, which uses the term 'minions' for nodes.

ETCD: OpenShift masters run the Kubernetes master services and etcd daemons, while the nodes run the Kubernetes kubelet and kube-proxy daemons. Scheduler and management/Replication in the figure are Kubernetes master services, while Data Store is the Etcd daemon.

Pods: The Kubernetes scheduling unit is the Pod, which is a grouping of containers sharing a virtual network device, internal IP address, TCP/UDP ports, and persistent storage. A Pod can be anything from a complete enterprise application, including each of its layers as a distinct container, to a single microservice inside a single container. Kubernetes manages replicas to scale pods. A replica is a set of pods sharing the same definition.

Project: A project groups Kubernetes resources so that the access rights can be assigned to users. A project can also be assigned a quota, which limits its number of defined pods, volumes, services, and other resources.

Images: The Source-to-Image (S2I) process in OpenShift pulls code from an SCM repository, automatically detects what kind of runtime that source code needs and starts a pod from a base image specific to that kind of runtime. Inside this pod, OpenShift builds the application the same way that the developer would.

If the build is successful, another image is created, layering the application binaries over its runtime, and this

image is pushed to an image registry internal to OpenShift. A new pod can then be created from this image, running the application. S2I can be viewed as a complete CI/CD pipeline already built into OpenShift.

OpenShift resources, such as images, containers, pods, services, builders, templates, and others, are stored on Etcd and can be managed by the OpenShift CLI, the web console, or the REST API. These resources can be viewed as JSON or YAML text files and shared or retrieved on an SCM system like Git or Subversion.

Networking: Docker creates a virtual kernel bridge and connects each container network interface to it. Docker itself does not provide a way to allow a pod on one host to connect to a pod on another host or to external world. Kubernetes service and route resources need external help to perform their functions.

A service needs software-defined networking (SDN) which will provide visibility between pods on different hosts, and a route needs something that forwards or redirects packets from external clients to the service internal IP. OpenShift provides an SDN based on Open vSwitch, and routing is provided by a distributed HAProxy farm.

Storage: Kubernetes recognizes a persistent Volume resource, which can define either local or network storage. A pod resource can reference a PersistentVolumeClaim resource in order to access storage of a certain size from a Persistent Volume.

HA: High Availability (HA) on an OpenShift Container Platform cluster has two distinct aspects: HA for the OpenShift infrastructure itself (that is, the masters); and HA for the applications running inside the OpenShift cluster. OpenShift provides a fully supported native HA mechanism for masters by default.

1.4. POC (Installation of OpenShift Enterprise on AWS)

Minimum Requirements:

Master	Etcid	Node
1	1	1

Prerequisites

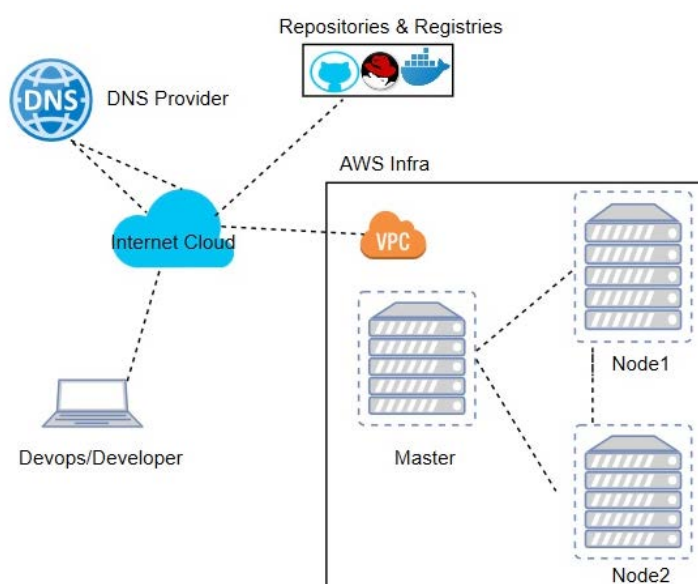
Name	CPU	Memory	Disk
Master	2	8GB	15GB / + /var 10G /home 25GB Unallocated (for setting up docker storage later)
Node 1 and 2	2	8GB	15GB / + /var 10G /home 25GB Unallocated (for setting up docker storage later)

This POC was designed with the following:

- AWS for EC2 instances as the compute resource.
- AWS VPC for the private communication.
- Domain registered in GoDaddy for URL access of OpenShift.
- RedHat Enterprise Linux 7 as the OS.
- RedHat subscription for enabling repos to install Atomic Host.

Capability of POC

- Installation and configuration of RedHat OpenShift Container platform.
- Managing pods and image streams.
- Manage access privileges and quota for system resources.
- Troubleshooting.



What are your thoughts on RedHat OpenShift? Share your comments with us at info@mindtree.com

References

- <https://www.openshift.com/>
- <https://www.redhat.com/en/technologies/cloud-computing/openshift>
- <https://docs.docker.com/registry/>
- <https://github.com/openshift/openshift-ansible>
- <https://www.redhat.com/en/resources/enterprise-linux-atomic-host-datasheet>

ABOUT MINDTREE

Mindtree [NSE: MINDTREE] delivers digital transformation and technology services from ideation to execution, enabling Global 2000 clients to outperform the competition. "Born digital," Mindtree takes an agile, collaborative approach to creating customized solutions across the digital value chain. At the same time, our deep expertise in infrastructure and applications management helps optimize your IT into a strategic asset. Whether you need to differentiate your company, reinvent business functions or accelerate revenue growth, we can get you there. Visit www.mindtree.com to learn more.