B.M.S. COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to VTU
Estd. 1946

DEPARTMENT OF CSE

CTY Project Work In collaboration with HPE

Weekly report

<table>
<tr><td>Project Title</td><td colspan="3">Comparative study of container runtimes</td></tr>
<tr><td rowspan="2">Student Team</td><td>Name: Kizhakel Sharat Prasad
USN: 1BM19CS074
SEM:UG-6th sem</td><td colspan="2">Name: Sharan S Pai
USN: 1BM19CS146
SEM:UG-6th sem</td></tr>
<tr><td>Name: Disha N
USN: 1BM19CS051
SEM:UG-6th sem</td><td colspan="2">Name: Vallisha M
USN: 1BM19CS177
SEM:UG-6th sem</td></tr>
<tr><td>Faculty Mentor</td><td>Dr. Pallavi G B
(Assistant Professor)
Prof. Madhavi R P
(Associate Professor)</td><td>HPE Mentors</td><td>Ravi Mehta</td></tr>
<tr><td>Review for the Period</td><td>28-02-2022</td><td colspan="2">14-03-2022</td></tr>
<tr><td>Work done so far</td><td colspan="3">1. Understood evolution of containers from virtual machines.
2. Explored first popular container run time Docker in detail.
3. Understood the meaning of OCI and CRI, explored few open source runtimes at a high level.</td></tr>
<tr><td>Review</td><td colspan="3">INTRODUCTION:
<u>COMPARATIVE STUDY OF CONTAINER RUNTIMES</u>
What is containerisation?
Containerisation, an Operating system (OS)-level virtualiza-tion technology, simplifies the deployment and management of applications by providing a flexible, low-overhead virtualization solution.</td></tr>
</table>

In recent years, containers have been deployed as a replacement for VirtualMachine (VM)-based virtualization. Containers offer better performance and flexibility at the expense of isolation and security when compared to traditional hypervisor virtualization. Their low overhead allows cloud infrastructure providers to deploy more microservices on a single host.

What is virtualization?

Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.

What is a container runtime?

Container runtime is the engine that runs and manages the components required to run containers. Communicating with the kernel to start containerized processes, setting up cgroups, configure mount points and do many things to make your container work.

Different container runtimes:

- **containerd:** An open and reliable container runtime.
- **cri-o:** Open Container Initiative-based implementation of Kubernetes Container Runtime Interface.
- **Firecracker:** Secure and fast microVMs for serverless computing.
- **gVisor:** Application Kernel for Containers.
- **rkt:** rkt is a pod-native container engine for Linux. It is composable, secure, and built on standards.

What is Docker?

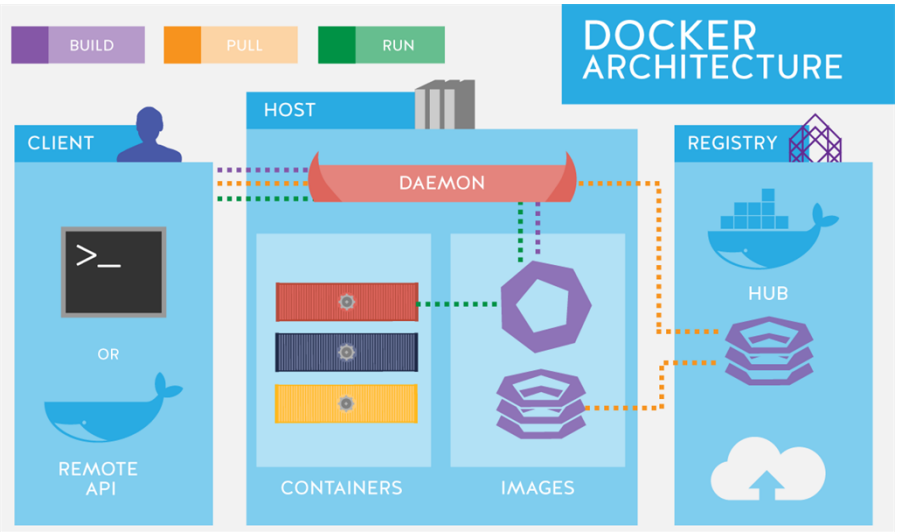
Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments.

Docker is composed of the following elements:

- a Daemon, which is used to build, run, and manage the containers
- a high-level API which allows the user to communicate with the Daemon,
- and a CLI, the interface we use to make this all available.

Advantages of using Docker:

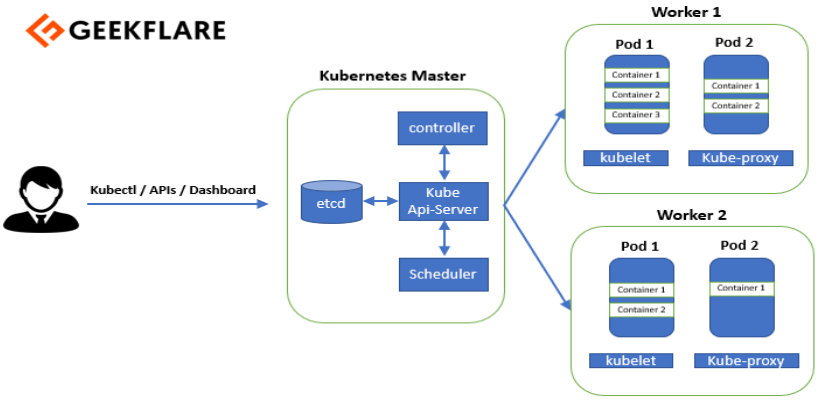
- Consistent and Isolated Environment
- Mobility – Ability to Run Anywhere
- Cost-effectiveness with Fast Deployment
- Flexibility
- Test, Roll Back and Deploy



What is Kubernetes?

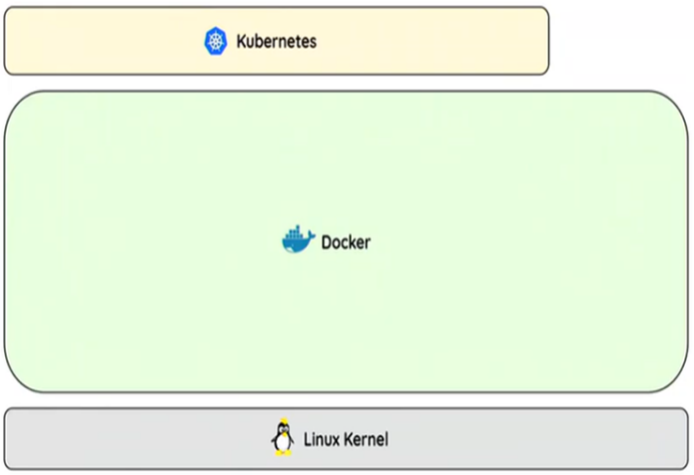
Kubernetes is an open-source system which allows you to run containers, manage them, automate deploys, scale deployments, create and configure ingresses, deploy stateless or stateful applications, and many other things.

- Improved resource utilization.
- Shortened software development cycles.
- Containerized monolithic applications.
- Enabled move to the cloud.
- Reduced public cloud costs.



Downfall of Docker as a container runtime and its dissection

Docker was widely used as container runtime. It was used by Kubernetes for managing, running, etc. containers. But some researchers were not favorable to the idea of Kubernetes relying only on docker. All independent companies started creating their own type of containers which led to a lot of difficulty in managing different types of containers. This led to formation of OCI (Open Container Initiative).



What is OCI?

OCI was a collaborative project with Linux foundation. Its primary objective was:

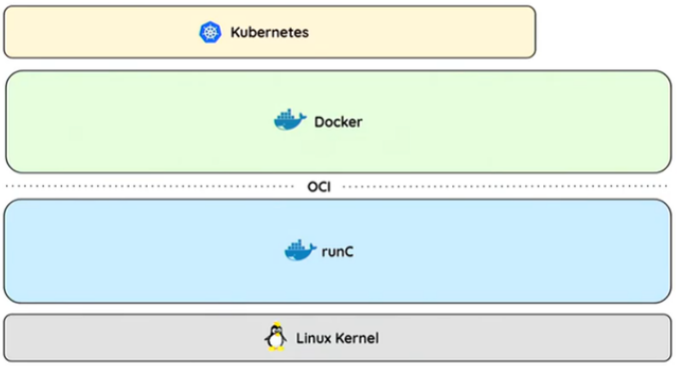
1. To create specification for containers and container runtimes.

2. To create a standard for image format used for distribution.

3. A reference runtime. (runC implemented in Docker).

NOTE: Docker used runC with libContainers as a container runtime.

Advantage of having OCI compliance is, it helped in interoperability, adaptability. Basically, for Kubernetes developers, container runtime became an abstraction.



Rkt, an alternative to Docker:

Docker was a more generalized. We could have used it independently or along with Kubernetes. Rkt, came up more specific idea. We could use rkt only with

Kubernetes to manage pods of containers. But it was not OCI compliant. Initially, the code for running Docker or rkt was integrated into Kubelets. But, to adjust with changing scenarios they had to create a new interface called CRI (Container Runtime Interface). This created a standard for any future runtime interface to come. Docker adapted to this rkt didn't. Rkt faded and had lost steam.

Dealing with 2 types of environments?

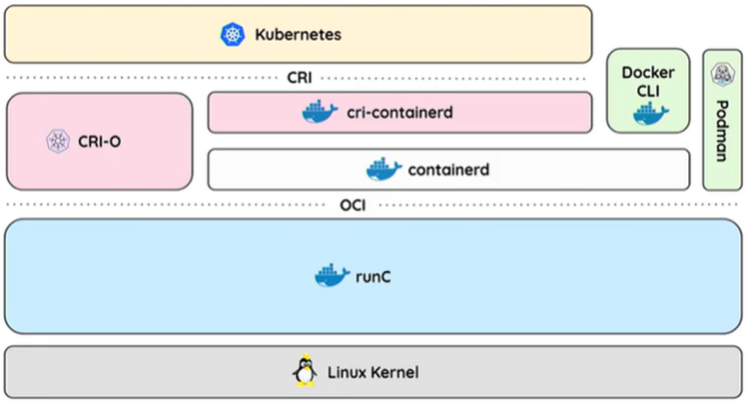
NOTE: Both can still be managed by Docker.

1. Kubernetes environment dependent

CRI-O is an implementation of the Kubernetes CRI (Container Runtime Interface) to enable using OCI (Open Container Initiative) compatible runtimes. It is a lightweight alternative to using Docker as the runtime for Kubernetes. It allows Kubernetes to use any OCI-compliant runtime as the container runtime for running pods. Today it supports runc and Kata Containers as the container runtimes but any OCI-conformant runtime can be plugged in principle.

2. Kubernetes environment independent

Podman is a daemonless container engine for developing, managing, and running OCI Containers on your Linux System.



Workload Isolation:

Researchers found out a big flaw in containers at least for sensitive workloads. So, in real world (public clouds), containers are run on VM. This is done to provide an extra layer of isolation to the server kernel. This is not present in runc. This is the reason they had to find a new runtime engine.

Public clouds services had their own way of handling this (was not open source yet).

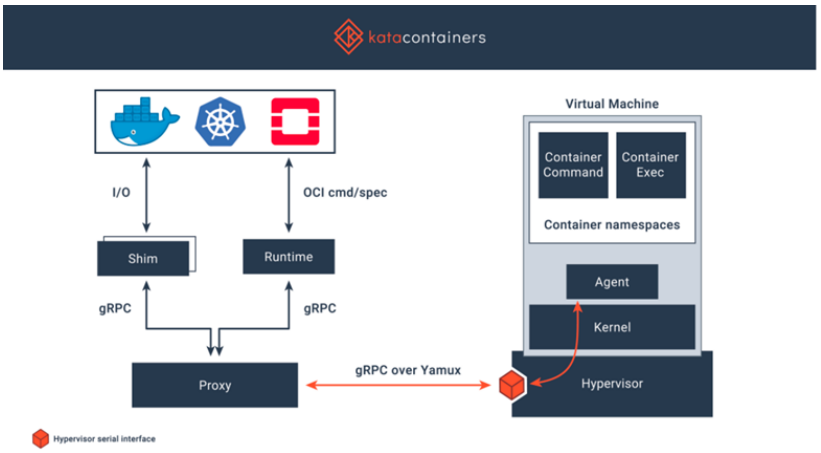
QEMU: **Quick EMUlator** (Open-source software for creating emulation and

virtual machine environments)

KVM: **Kernel Virtual Machine** (open-source virtualization technology built into Linux®. Specifically, KVM lets you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines)

KATA Containers:

Kata Containers is an open source community working to build a secure container runtime with lightweight virtual machines that feel and perform like containers, but provide stronger workload isolation using hardware virtualization technology as a second layer of defense. This is OCI compliant container engine. So, it can be used with Docker CLI/Podman as well. This was an open-source solution to workload isolation. This forced the public cloud services to release their respective technologies.



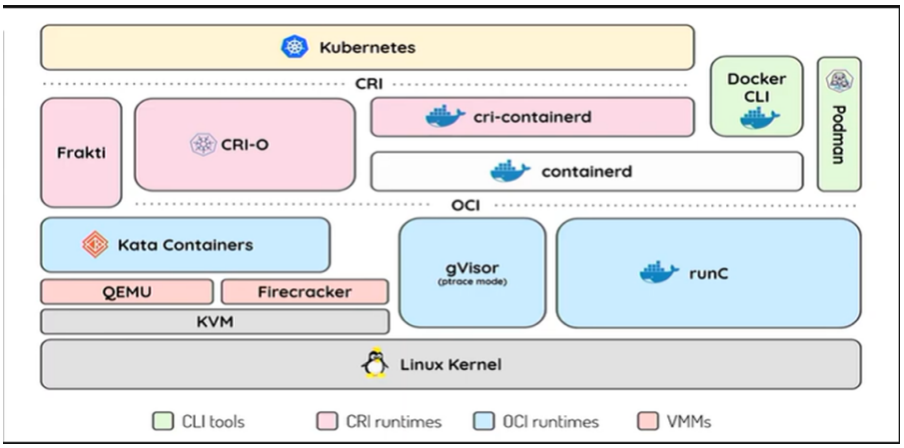
gVisor:

gVisor is an application kernel, written in Go, that implements a substantial portion of the Linux system call interface. It provides an additional layer of isolation between running applications and the host operating system. gVisor includes an Open Container Initiative (OCI) runtime called runsc that makes it easy to work with existing container tooling.

Firecracker:

Firecracker is an open-source virtualization technology that is purpose-built for

creating and managing secure, multi-tenant container and function-based services. QEMU was too wide and heavy, atleast that's what AWS felt. Firecracker enables you to deploy workloads in lightweight virtual machines, called microVMs, which provide enhanced security and workload isolation over traditional VMs, while enabling the speed and resource efficiency of containers.



Action plan for next week	1. Learning deployment/running application of choice using the above mentioned container runtimes especially rocket and containerd
References:	**1.** https://www.aquasec.com/cloud-native-academy/container-security/container-runtime-interface/ **2.** https://searchcloudsecurity.techtarget.com/feature/Cloud-containers-what-they-are-and-how-they-work **3.** https://www.ibm.com/in-en/cloud/learn/docker **4.** https://www.freecodecamp.org/news/what-is-docker-used-for-a-docker-container-tutorial-for-beginners/ **5.**