

# Containerization with Docker

---

A transition from Hard work to Smart work.



# Today's Agenda

---

- Introduction to containers
- Containers vs Virtual Machines
- Docker, Docker images, DockerHub, Dockerfile
- Building an image out of a container
- Let's build an image with Dockerfile!
- Introduction to Kubernetes
- How do we use containers on real environments?



# What is Virtualization?

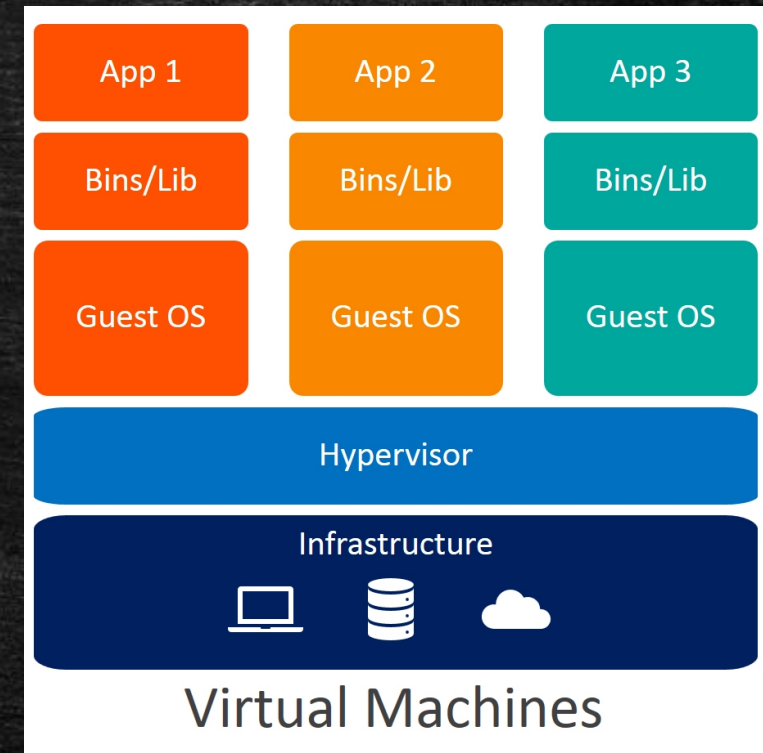
---

- **Virtualization** is the technique of importing a Guest operating system on top of a Host operating system.
- This technique was a revelation at the beginning because it allowed developers to **run multiple operating systems** in different virtual machines all running on the same host.
- This **eliminates** the need for **extra hardware resource**.



# What is a Virtual Machine?

- **Virtual machines** are heavy software packages that provide complete emulation of hardware devices like CPU, Disk and Networking devices.
- Virtual machines may also include a **complementary software** stack to run on the **emulated hardware**.
- These **hardware** and **software** packages combined produce a fully functional snapshot of a computational system.





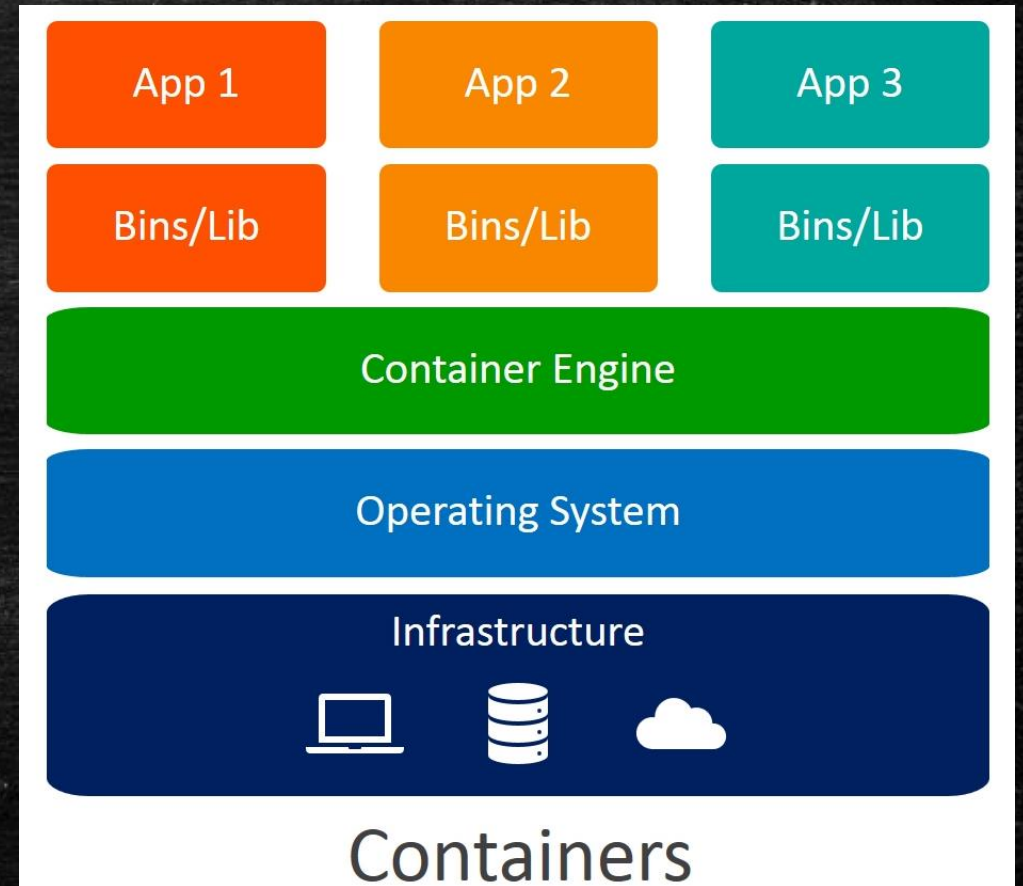
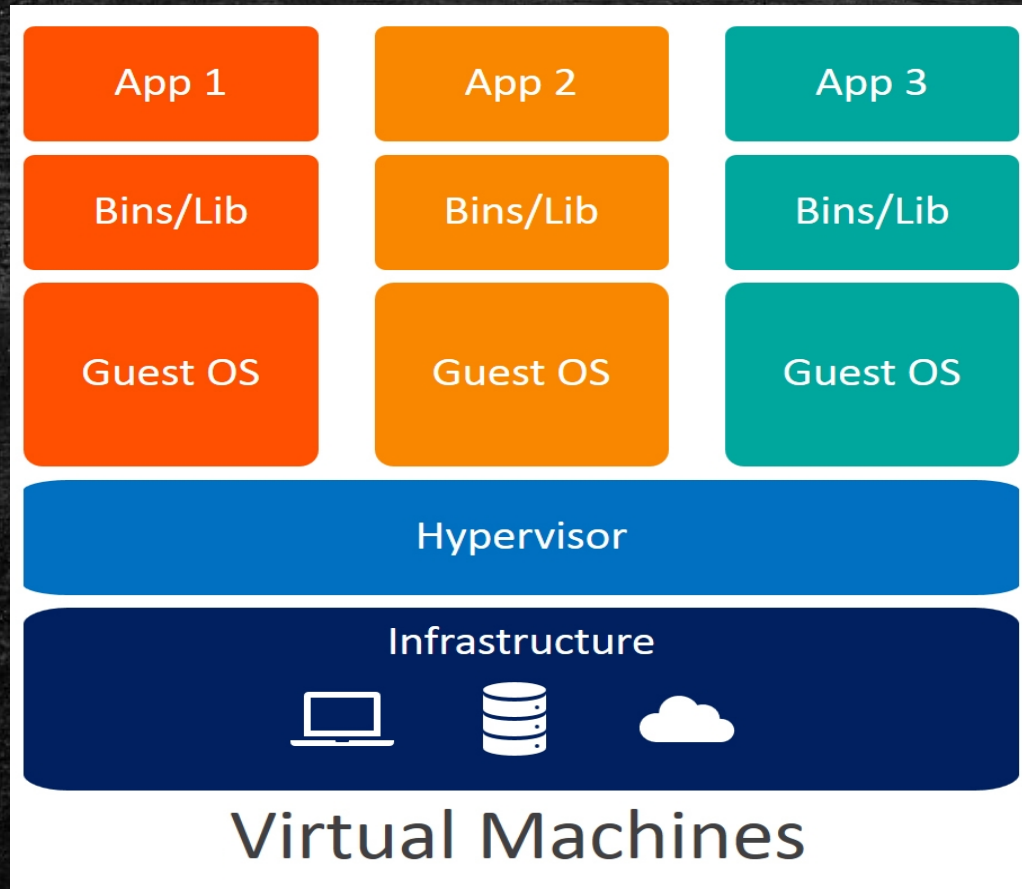
# Containerization

---

- **Containerization** is the packaging together of software code with all its necessary components like libraries, frameworks, and other dependencies so that they are isolated in their own "container."
- A **container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

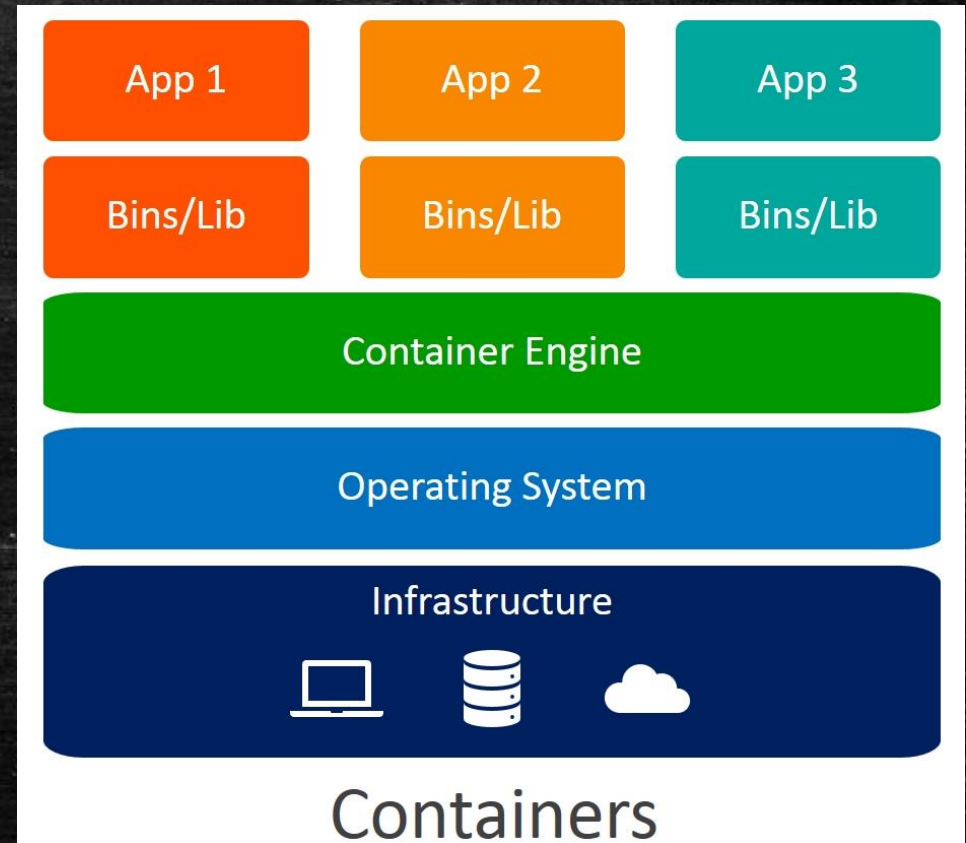


# Containerization over Virtualization:



# Containerization over Virtualization:

- Containers on the same OS kernel are **lighter and smaller**
- **Better resource utilization** compared to VMs
- **Boot-up** process is **short** and takes few seconds





# When to use Containers?

---

Consider containers in particular if the following is a priority

- Start time
- Efficiency
- Licensing
- Code reuse



It works on my machine!!!

---





It works on my machine!!!

---





**WHAT IF I TOLD YOU WE COULD REALLY**

**HAVE THE SAME CODE WORK EVERYWHERE**



# Docker

---

- **Docker** is a containerization platform that packages your application and all its dependencies together in the form of Containers to ensure that your application works seamlessly in any environment.
- Docker containers that run on Docker Engine:
  - Standard
  - Lightweight
  - Secure



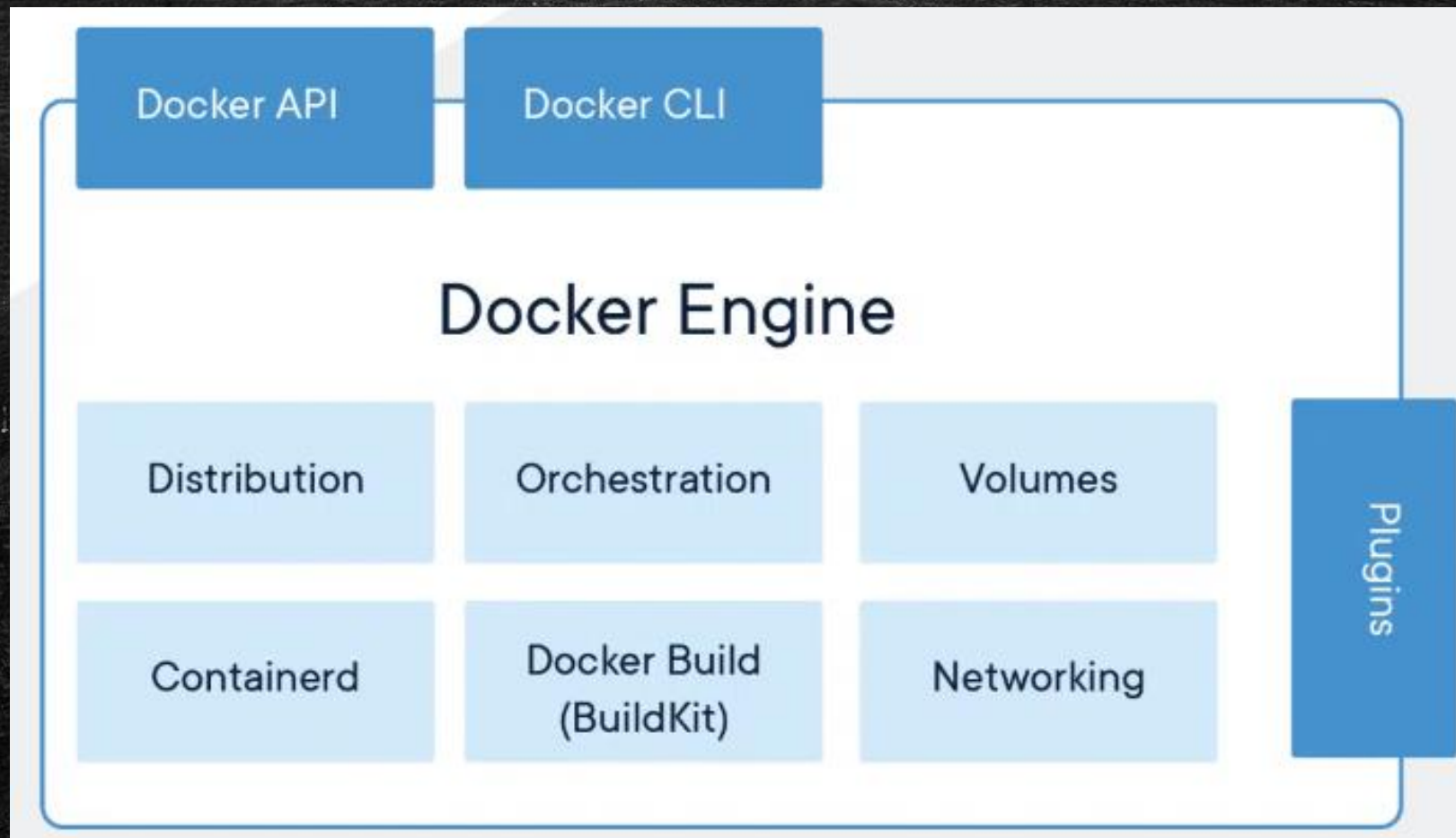
# Why Docker

---

- Docker creates **simple tooling** and a **universal packaging** approach that bundles up all application dependencies inside a container which is then run on Docker Engine.
- **Docker Engine** enables containerized applications to run anywhere consistently on any infrastructure, solving “dependency issues” for developers and operations teams, and eliminating the “**it works on my laptop!**” problem.



# Docker Engine

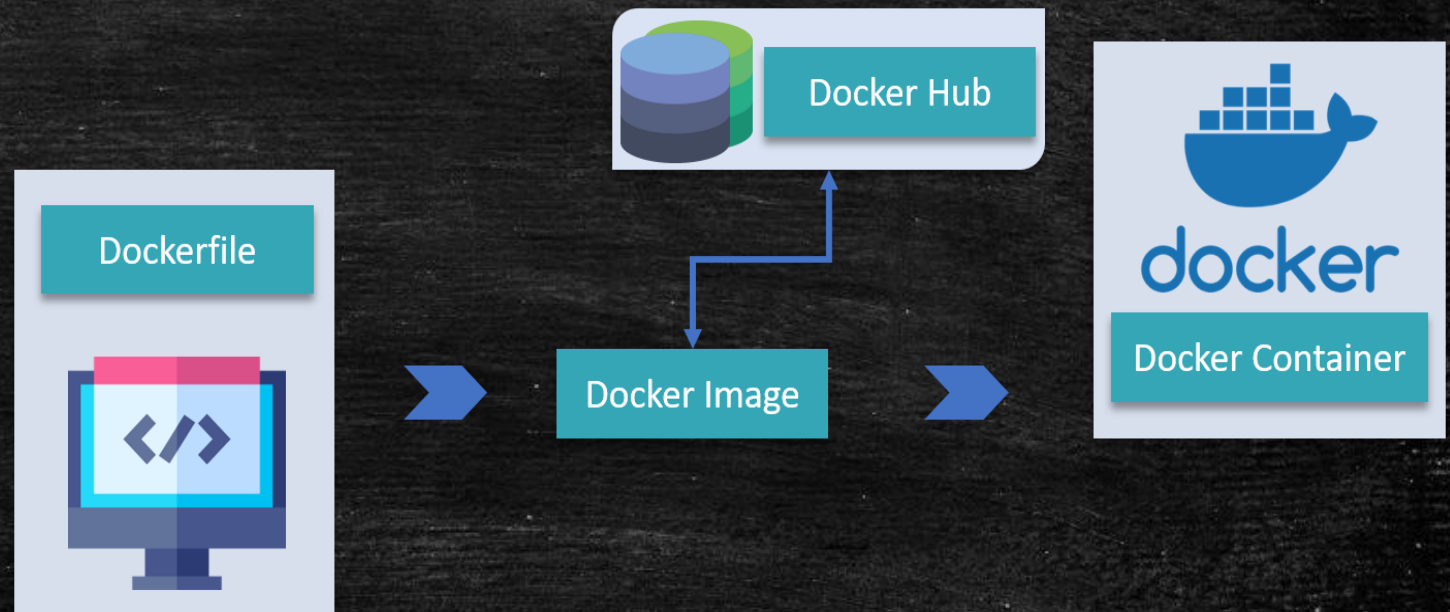




# Terminologies of Docker

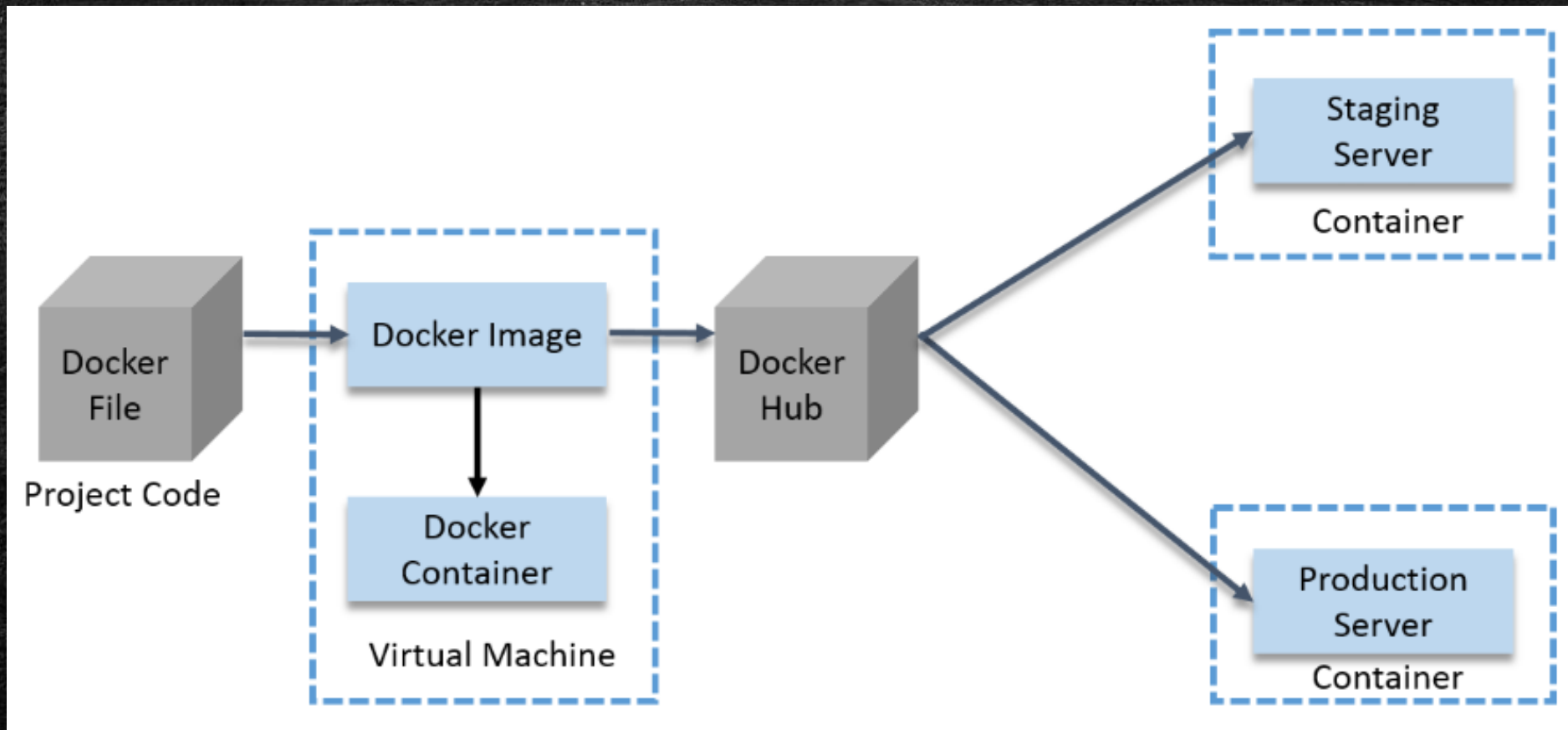
---

1. Docker Image
2. Dockerfile
3. Docker Containers
4. Docker Hub
5. Docker Compose
6. Docker Client
7. Docker Daemon





# How a Docker Container Works?





# Introduction to Kubernetes

---

- Kubernetes is a **container management technology** developed in Google lab to **manage** containerized applications in different kind of environments such as **physical, virtual, and cloud infrastructure**.
- It is an **open source system** which helps in creating and managing containerization of application.
- Kubernetes comes with a capability of **automating deployment, scaling of application, and operations of application containers across clusters**.



# Features of Kubernetes

---

## Features

Continues **development, integration**  
and **deployment**

Containerized infrastructure

**Application-centric** management

**Auto-scalable** infrastructure

## Features

**Environment consistency** across  
development testing and production

**Loosely coupled infrastructure**, where  
each component can act as a separate  
unit

Higher density of resource utilization

**Predictable infrastructure** which is  
going to be created



# What can Kubernetes do for you?

---

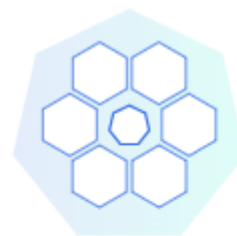
- Kubernetes helps you make sure that the containerized applications run **where** and **when you want**, and helps them find the resources and tools they need to work.
- Kubernetes is a **production-ready, open source platform** designed with Google's accumulated experience in container orchestration.



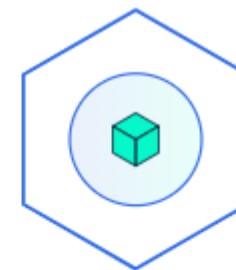
# Kubernetes Basics Modules



1. Create a Kubernetes cluster



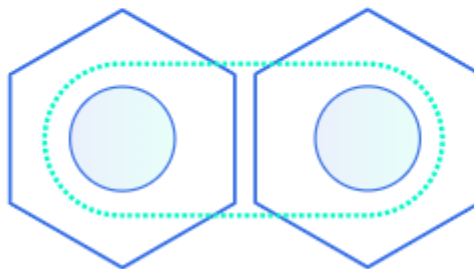
2. Deploy an app



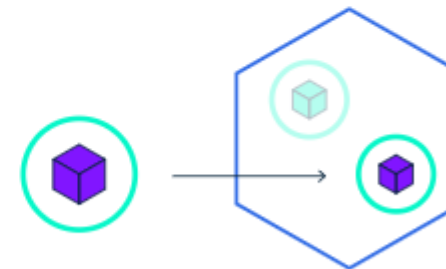
3. Explore your app



4. Expose your app publicly



5. Scale up your app



6. Update your app





*That's all Folks!*



---

THANK YOU