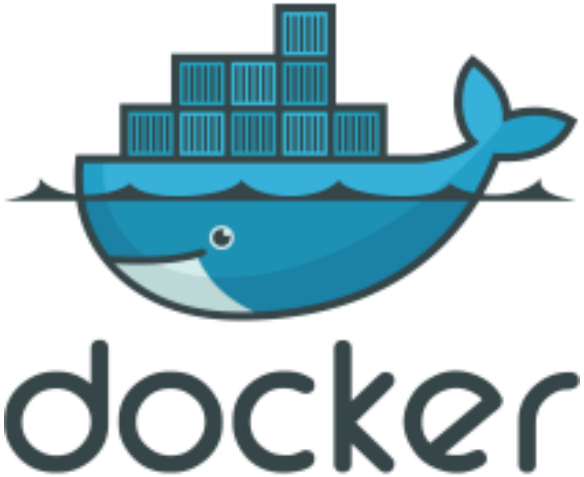
Introduction to



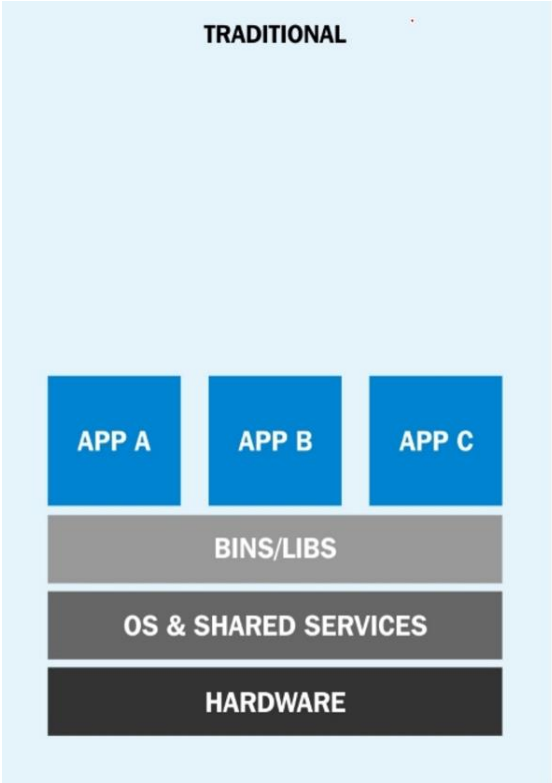
By:
Goutham Kumar

Agenda

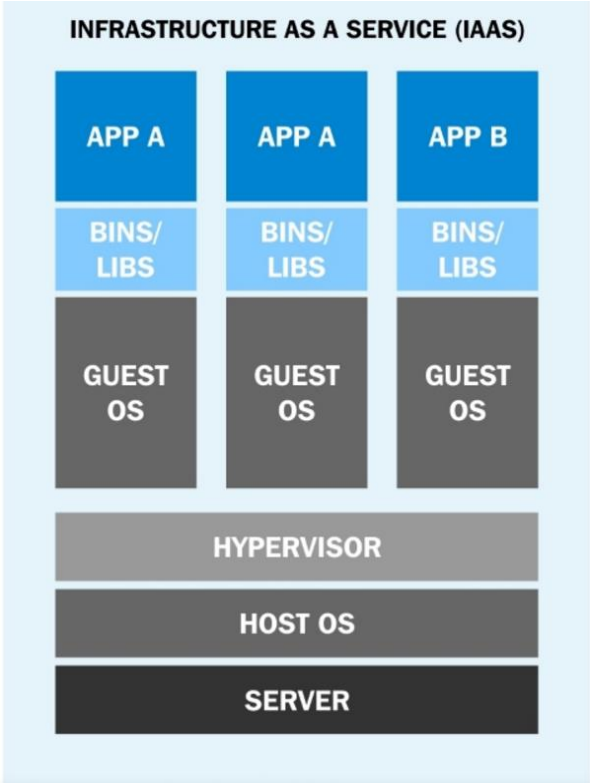
1. Why Docker
2. What is Docker
3. Underlying technology
4. Benefits of Docker
5. Docker Architecture
6. Components of Docker
7. Docker Objects
8. Docker Editions & Platform

Traditional Application Setup

1



2



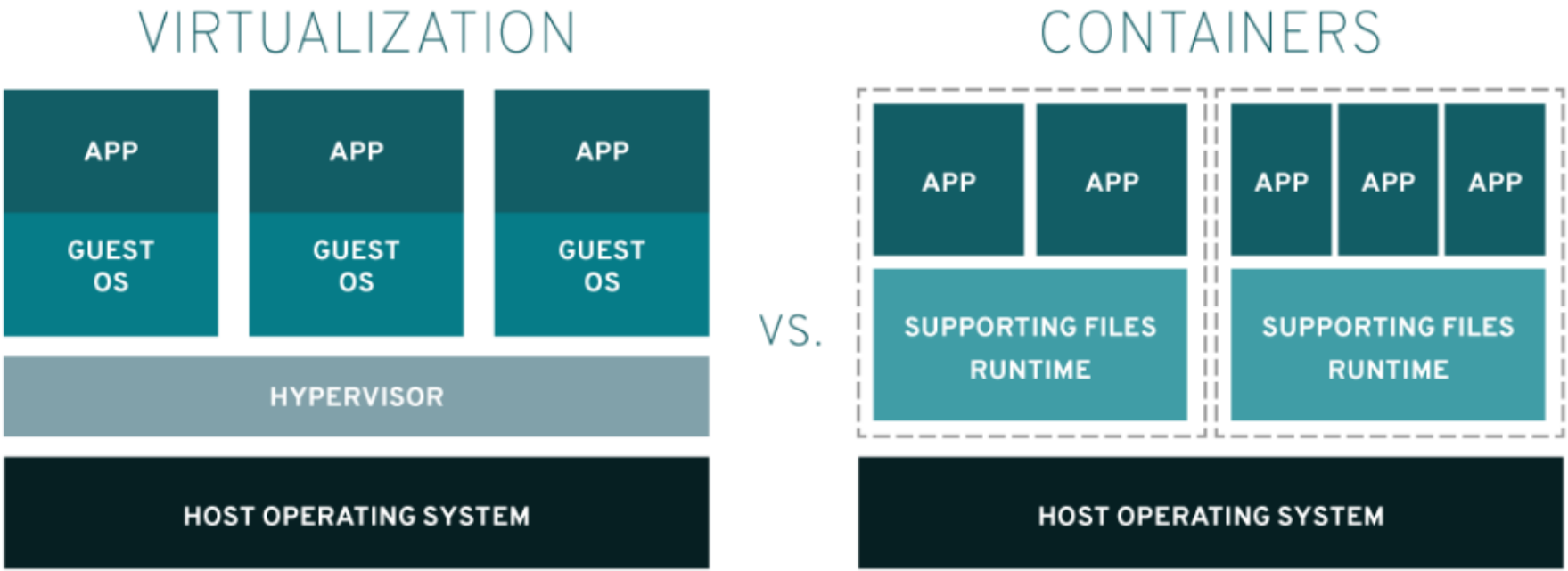
What is a Container

LXC - Linux containers are self-contained execution environments—with their own, isolated CPU, memory, block I/O, and network resources—that share the kernel of the host operating system.

The result is something that feels like a virtual machine, but sheds all the weight and startup overhead of a guest operating system.

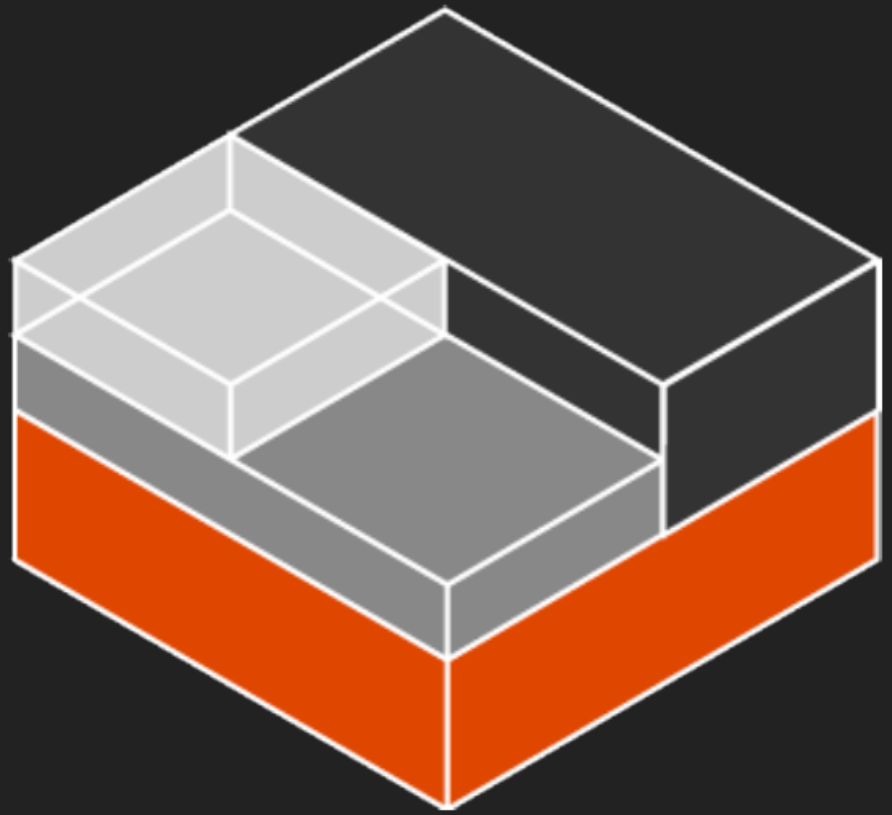
A Linux container is nothing more than a process that runs on Linux. It shares a host kernel with other containerized processes.

Virtualization vs Containers



THE SLOW RISE OF LINUX CONTAINERS

- chroot system call (1979)
  - Version 7 Unix
- Virtuozzo (2000)
- jail (2000)
  - FreeBSD 4.0
- Solaris Zones (2004)
  - Solaris 10
- OpenVZ (2005)
- Linux Containers - LXC (2008)
  - version 2.6.24 of the Linux kernel

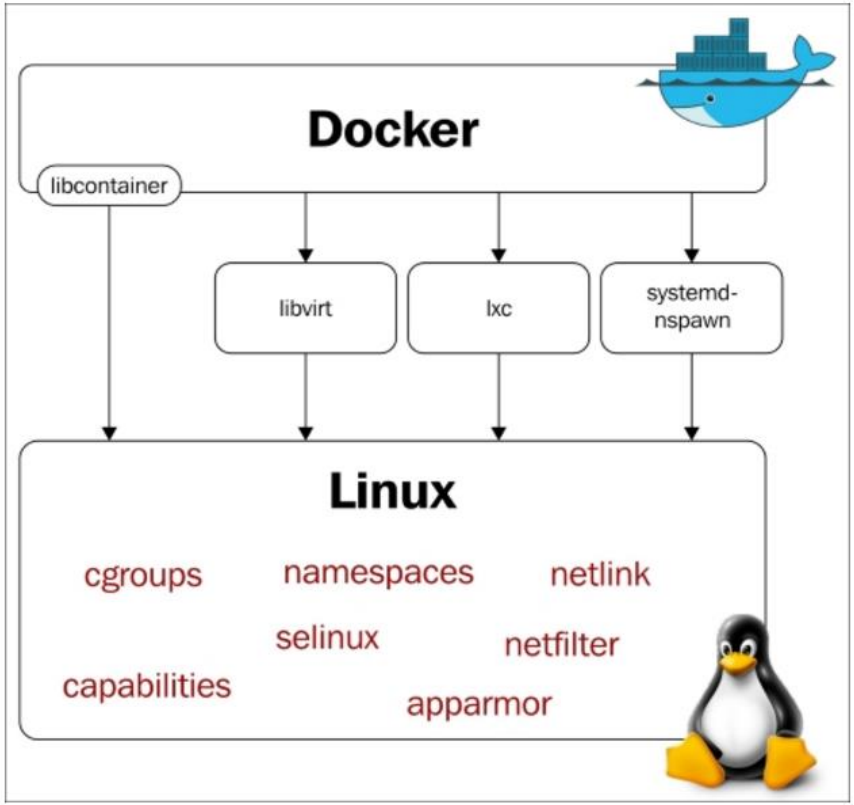


Understanding Linux Containers

To understand containers, we have to start with Linux cgroups and namespaces, the Linux kernel features that create the walls between containers and other processes running on the host.

1. Namespace
2. Control Groups

Building Blocks of LXC



NAMESPACE

Kernel namespaces provide a virtualized world for the container processes to run in. For example, the "PID" namespace causes a containerized process to only see other processes inside of that container, but not processes from other containers on the shared host.

They wrap a set of system resources and present them to a process to make it look like they are dedicated to that process.

PİD Namespace, NET Namespace, MNT Namespace, UTS Namespace

CONTROL GROUPS

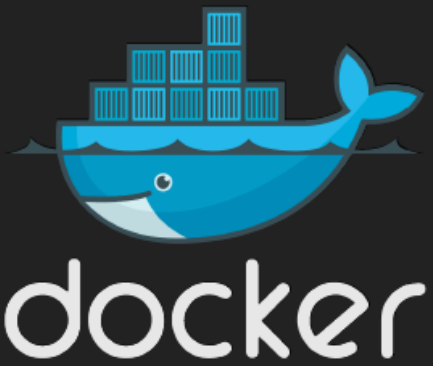
Linux cgroups, originally developed by Google, govern the isolation and usage of system resources, CPU and memory, for a group of processes.

For example, if you have an application that takes up a lot of CPU cycles and memory, such as a scientific computing application, you can put the application in a cgroup to limit its CPU and memory usage.

.

DOCKER ENGINE ISN'T A...

- virtualization platform (VMware, KVM, etc.)
- cloud platform (AWS, Azure, etc.)
- configuration management tool (Chef, Puppet, etc.)
- deployment framework (Capistrano, etc.)
- workload management tool (Mesos, Kubernetes, etc.)
- development environment (Vagrant, etc.)



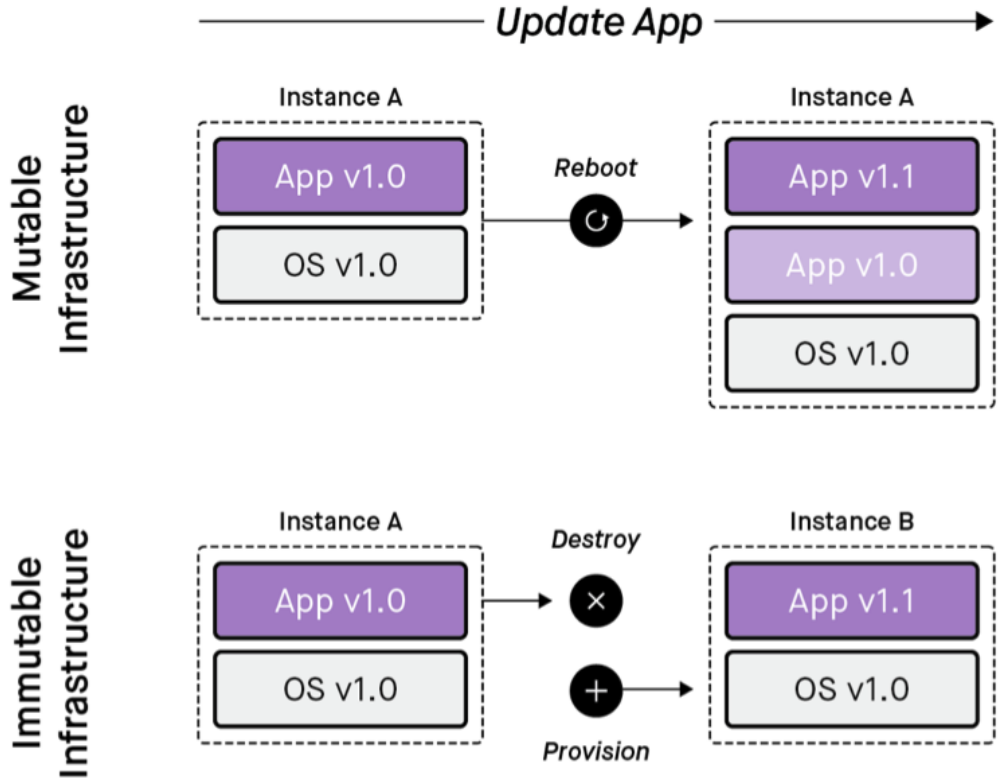
What is Docker

The docker container engine manages the configuration of Linux kernel namespaces, additional security features, and cgroups, and introduced a layered packaging format for content that runs inside containers.

Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

The runtime for those containers isn’t docker, it’s Linux.

Mutable vs Immutable



Benefits of Containers

Speed and Size

Reproducible and portable builds

Isolation

Security

Multi-Cloud Platforms

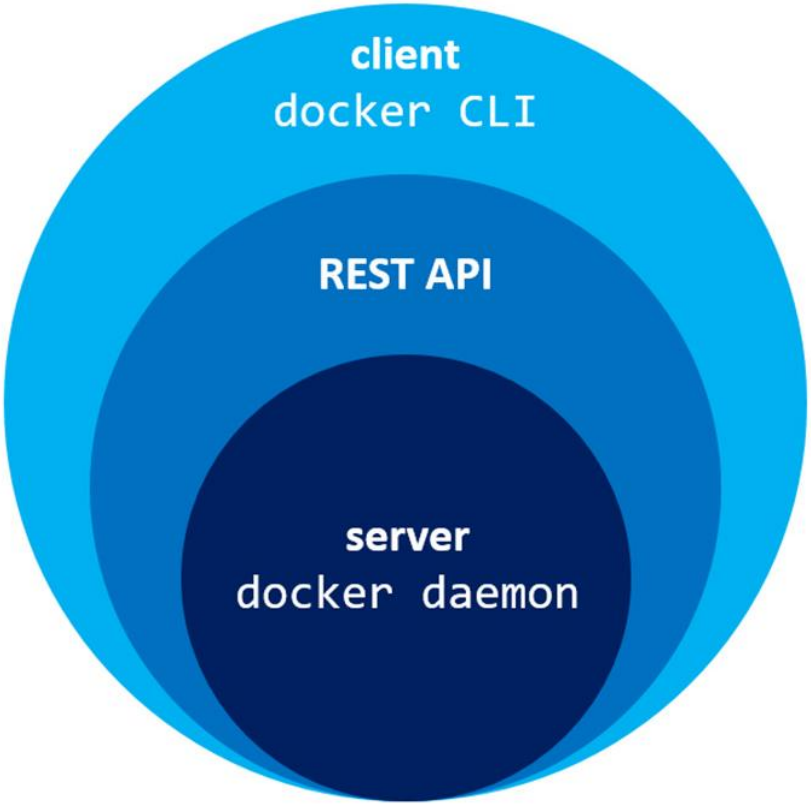
Running more workloads on the same hardware

Docker Architecture

The Docker daemon is a service that runs on your host operating system.

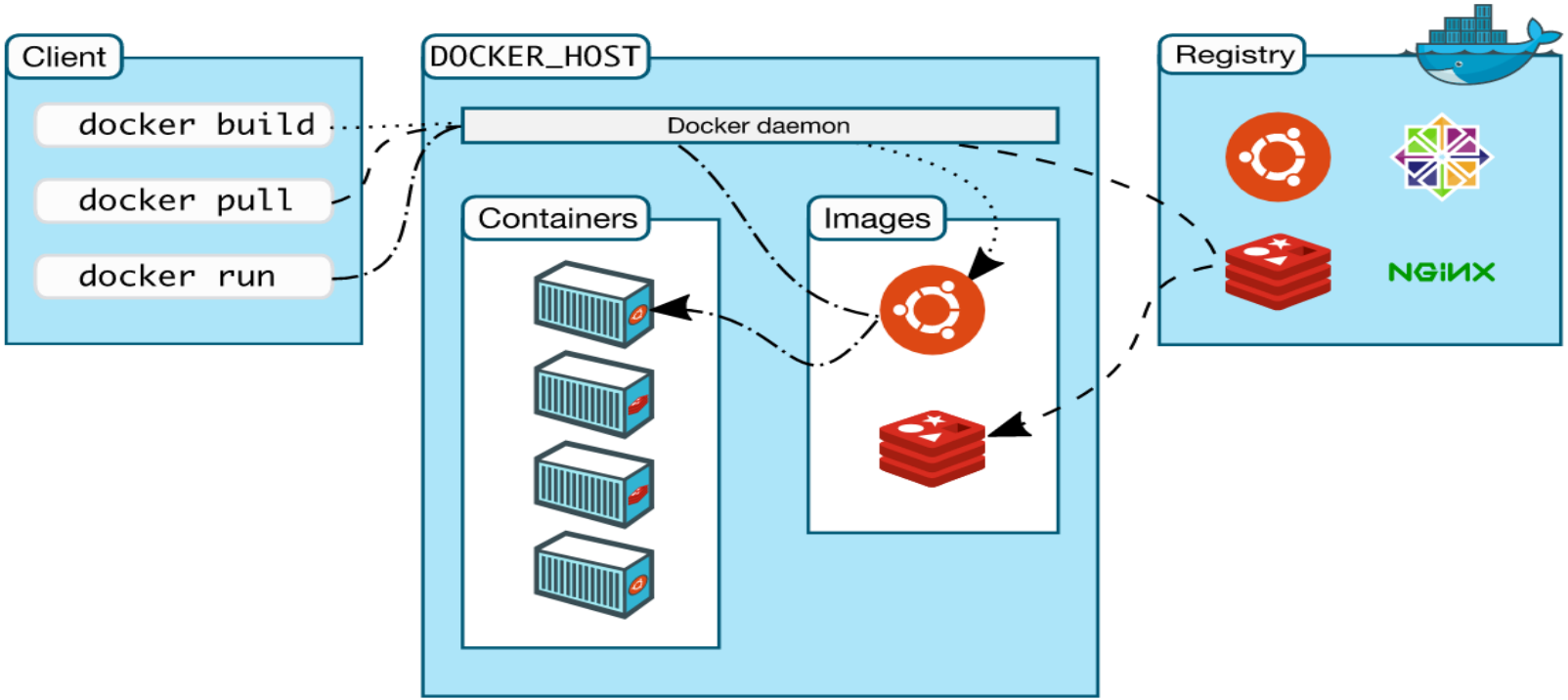
The Docker daemon itself exposes a REST API. From here, a number of different tools can talk to the daemon through this API.

The Docker CLI. It is a command line tool that lets you talk to the Docker daemon.

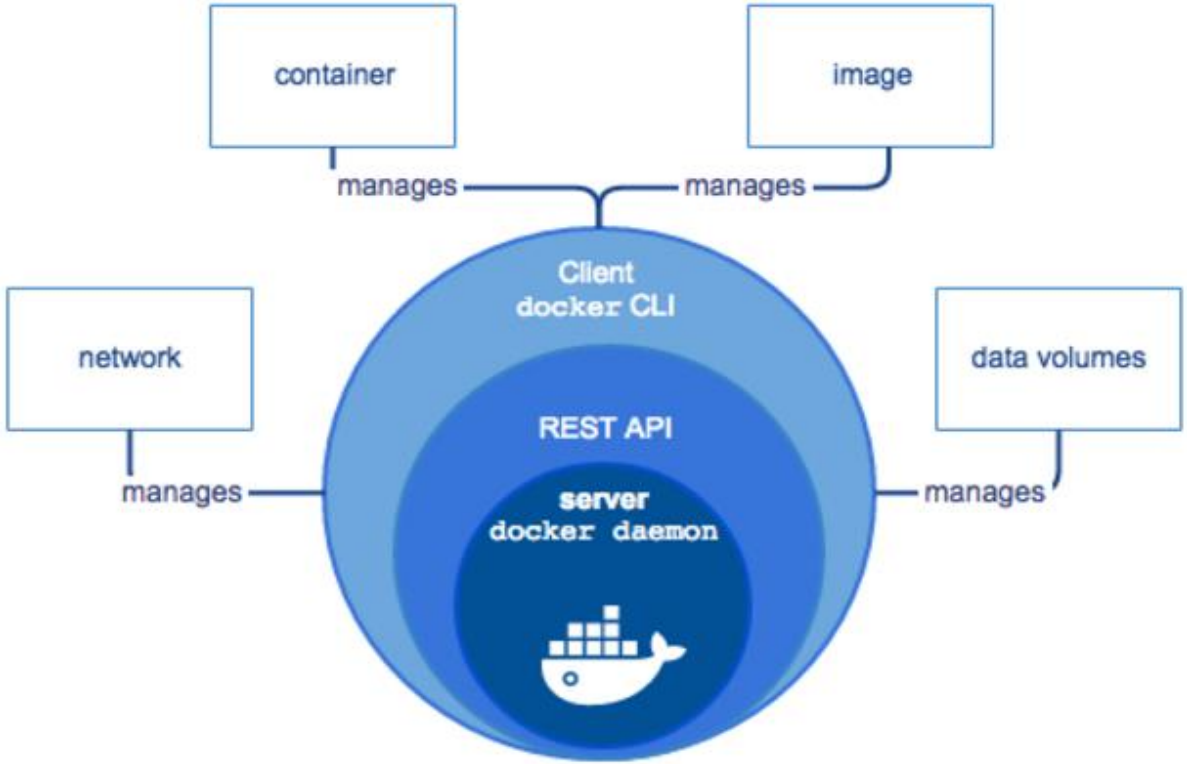


When you install Docker, you get both the Docker daemon and the Docker CLI tools together.

Docker Components



Docker Objects



Docker Editions

Docker is available in two editions:

Docker Enterprise Edition (Docker EE) is designed for enterprise development and IT teams who build, ship, and run business critical applications in production at scale.

Docker Community Edition (Docker CE) is ideal for developers and small teams looking to get started with Docker and experimenting with container-based apps

Docker CE has both **stable** and **edge** channels.
Stable builds are released once per quarter and are supported for 4 months.
Edge builds are released once per month

Platform Support

Platform	Docker EE	Docker CE x86_64	Docker CE ARM
Ubuntu	✔	✔	✔
Debian		✔	✔
Red Hat Enterprise Linux	✔		
CentOS	✔	✔	
Fedora		✔	
Oracle Linux	✔		
SUSE Linux Enterprise Server	✔		
Microsoft Windows Server 2016	✔		
Microsoft Windows 10		✔	
macOS		✔	
Microsoft Azure	✔	✔	
Amazon Web Services	✔	✔	

Q & A