

# 01 初识Docker

## 1.1 What is Docker

### 1.1.1 官网首页

<https://www.docker.com/>

Modernize your applications, accelerate innovation  
Securely build, share and run modern applications anywhere

### 1.1.2 Docs

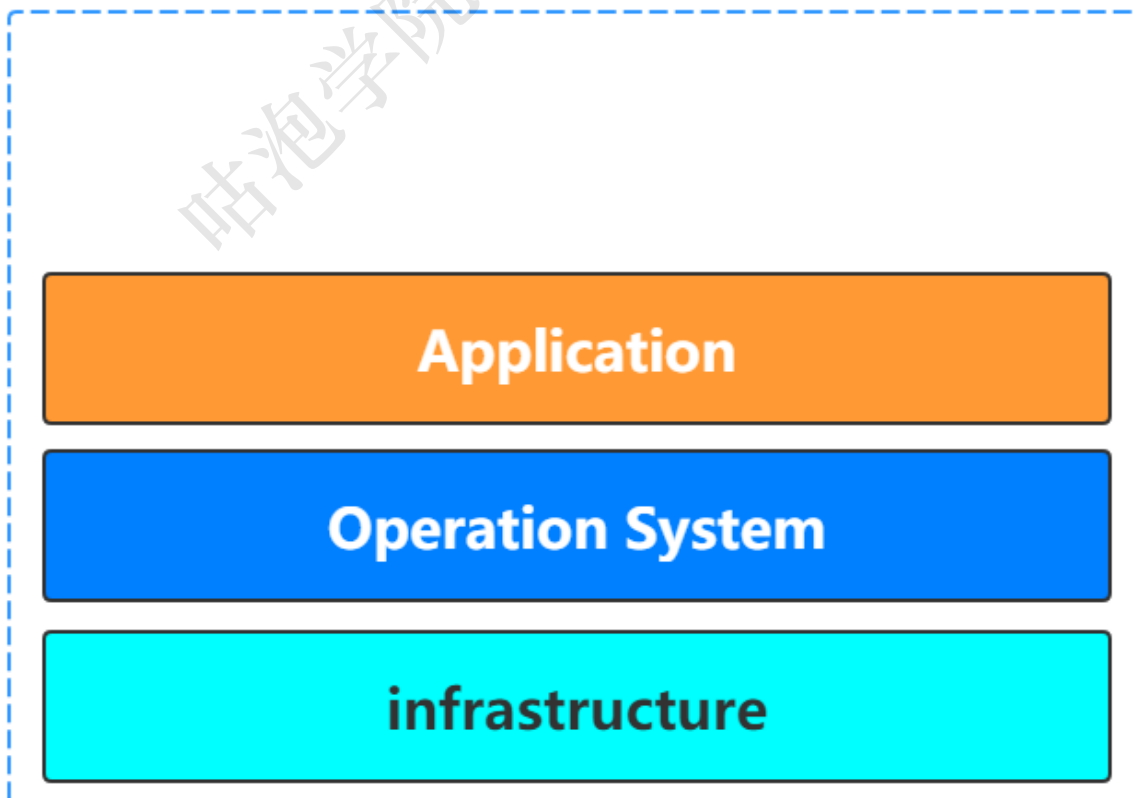
<https://docs.docker.com/get-started/>

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers. The use of Linux containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

### 1.1.3 不理解，我太难了

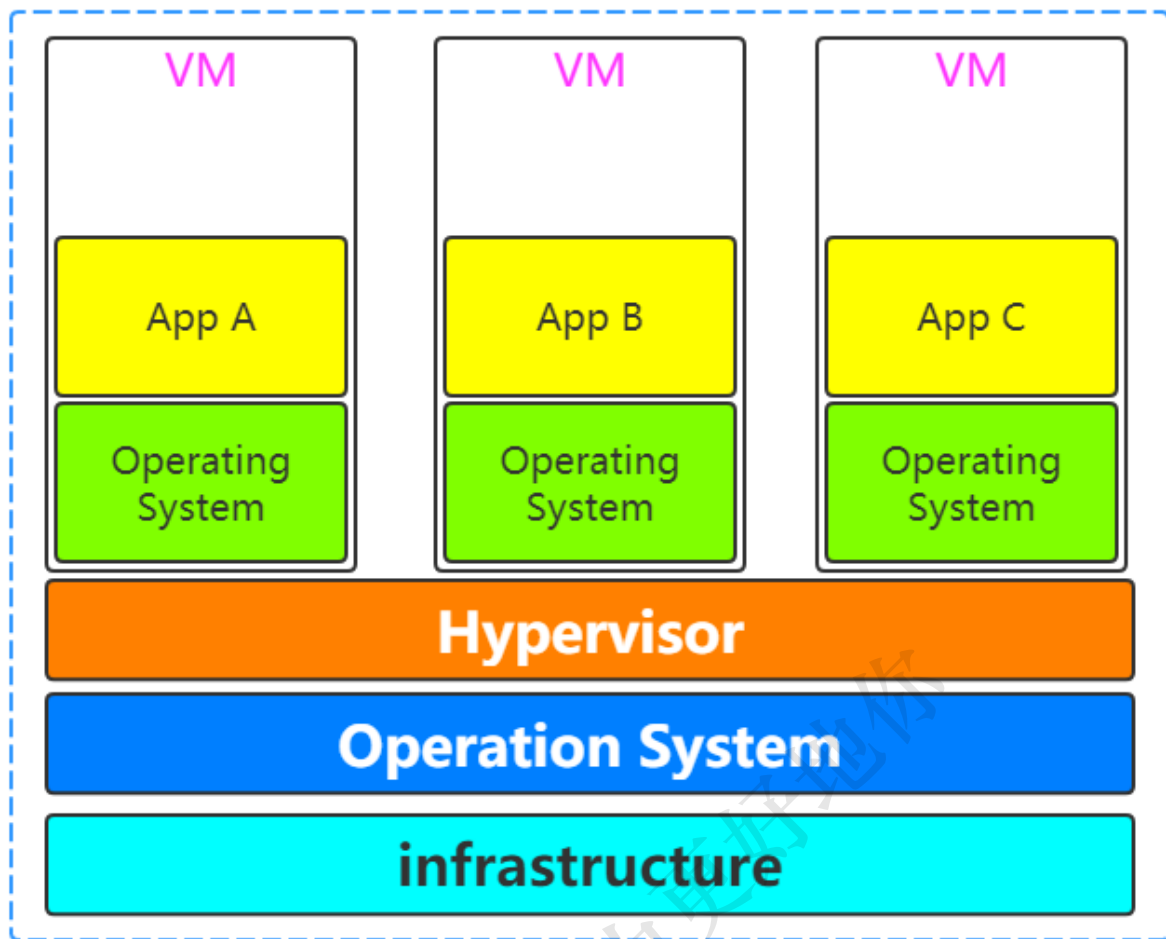
不妨从一个需求开始：开发好了一个项目shopping，部署上线

- 远古时代



问题：成本高、部署慢、浪费资源、硬件限制、不利于迁移扩展

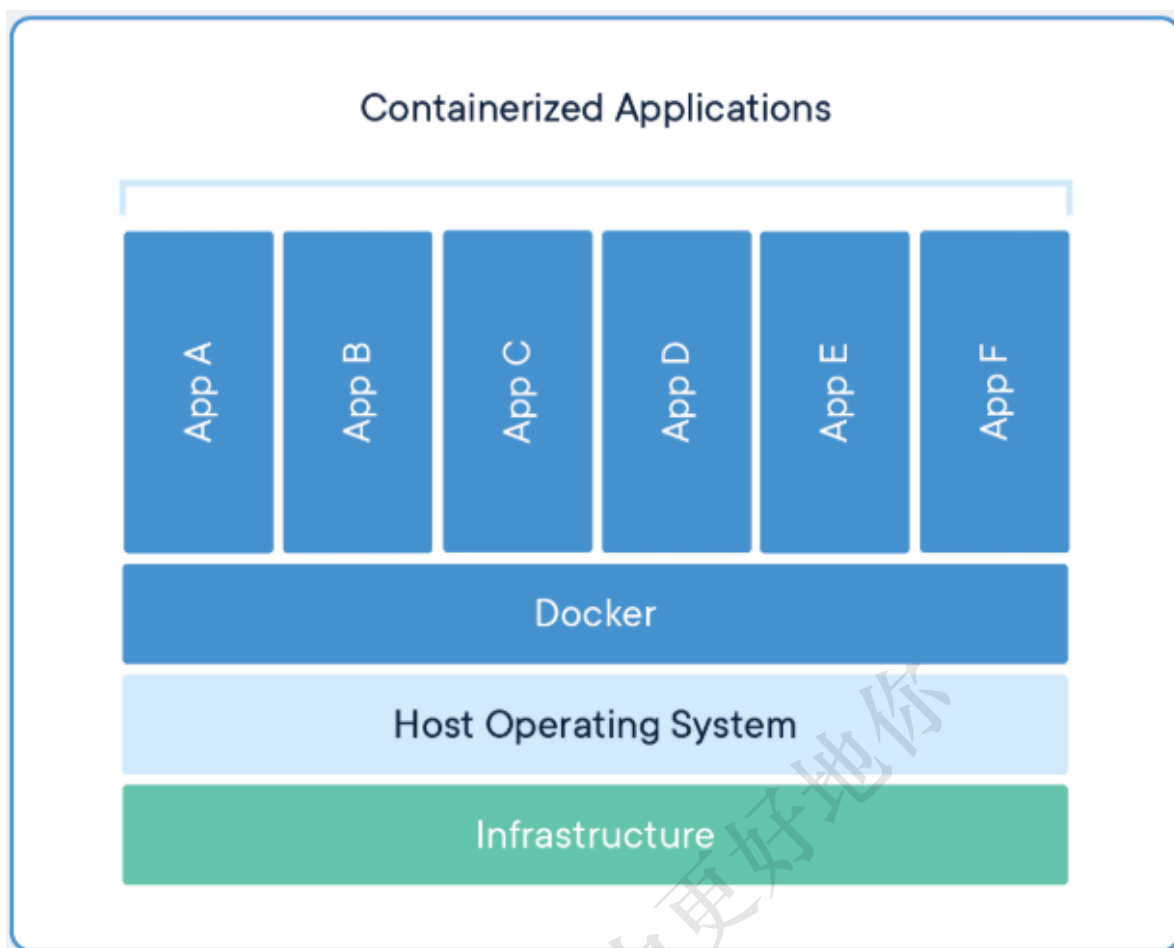
- 虚拟化时代



优点：相对利用好资源，相对容易扩展等。

缺点：虚拟机太重了，一上来占用较多物理资源，移植性差，资源利用率低等。

- 容器时代



### 1.1.4 再次理解Docker

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers. The use of Linux containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

发现还是比较容易理解的，但是这里有一句“Containers are not new”，也就是容器化技术很早就出现了，比如常见的容器化技术有OpenVZ，LXC，RKT等

### 1.1.5 Docker的优势和应用场景

[www.docker.com](http://www.docker.com) --->Solutions

- (1) 有助于Microservices的落地和部署
- (2) 充分利用物理机资源，同时能够整合服务器资源
- (3) 提高开发效率，测试效率，部署效率，有利于DevOps的落地，CICD
- (4) 云原生落地，应用更好地迁移
- ...

## 1.2 What is Image and Container?

### 1.2.1 What is Image?

Why is docker?->[What is a container](#)

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

## 1.2.2 What is Container?

Why is docker?->[What is a 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.

## 1.2.3 Relation between image and container

Container images become containers at runtime and in the case of Docker containers- images become containers when they run on Docker Engine.

## 1.2.4 View from Docs

从帮助文档的角度看

[docker官网](#)->Resources->Docs->Get started->Get started with Docker->Orientation->Images and containers

A container is launched by running an image. An image is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.

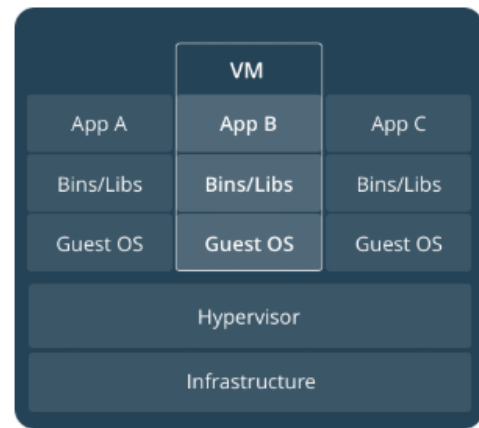
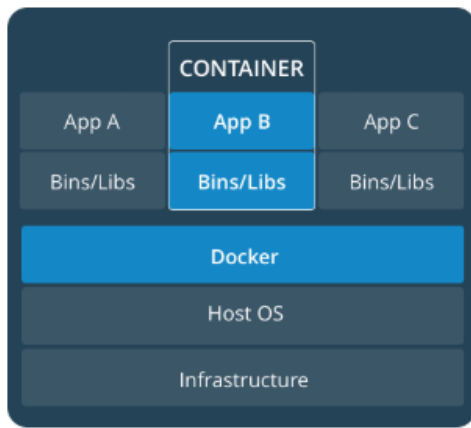
A container is a runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process). You can see a list of your running containers with the command, `docker ps`, just as you would in Linux.

## 1.3 Containers and virtual machines

[docker官网](#)([www.docker.com](http://www.docker.com))->Resources->Docs->Get started->Get started with Docker->Orientation->Containers and virtual machines

A container runs natively on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking no more memory than any other executable, making it lightweight.

By contrast, a virtual machine (VM) runs a full-blown "guest" operating system with virtual access to host resources through a hypervisor. In general, VMs provide an environment with more resources than most applications need.



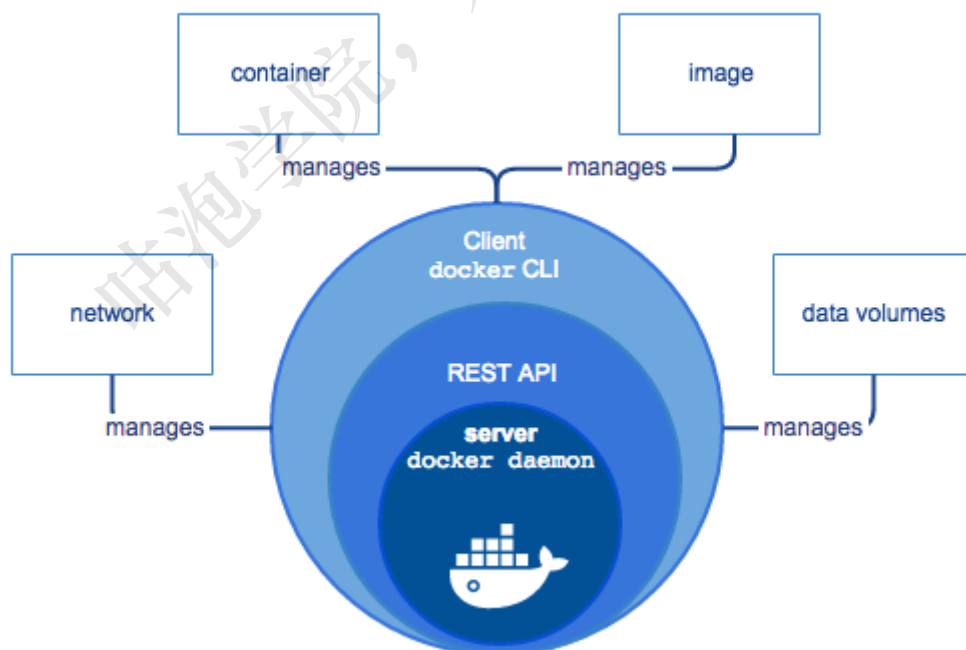
## 1.4 Docker Engine and Architecture

<https://docs.docker.com/engine/docker-overview/>

### 1.4.1 Docker Engine

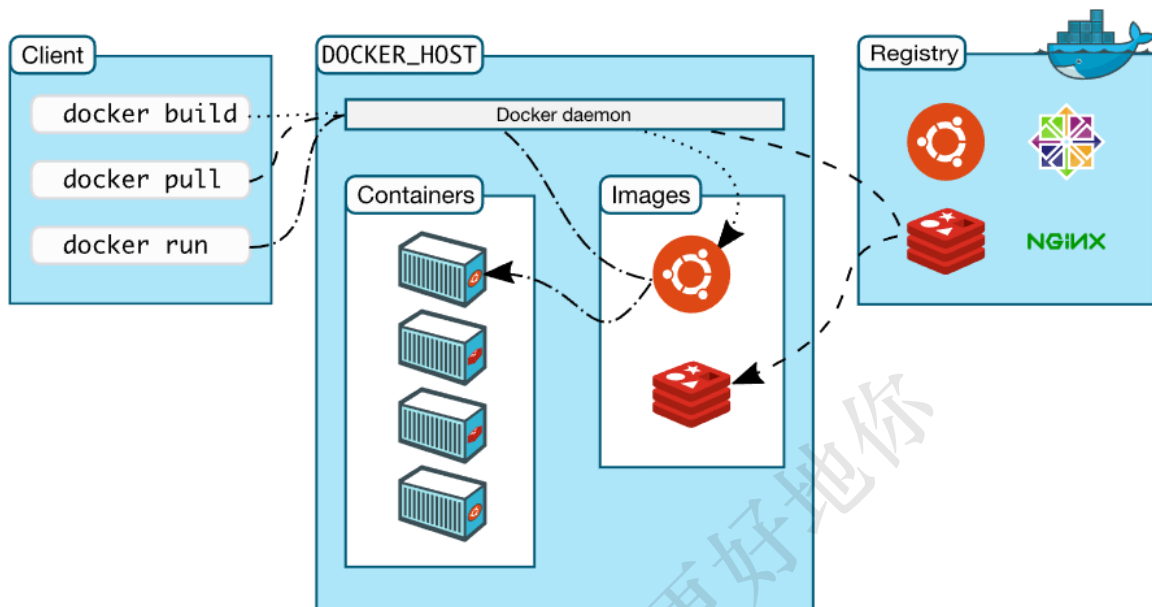
*Docker Engine* is a client-server application with these major components:

- A server which is a type of long-running program called a daemon process (the `dockerd` command).
- A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
- A command line interface (CLI) client (the `docker` command).



### 1.4.2 Docker Architecture

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface.



## 1.5 Install and Experience

见“上课课件”目录下的install.md文件

<https://docs.docker.com/install/linux/docker-ce/centos/>

