Docker 技术介绍

黑洞　heidsoft@sina.com

内容简要

Docker生态系统

Community

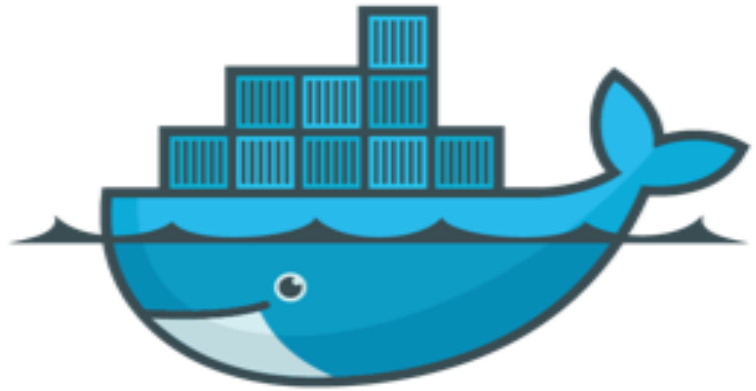
600+ Contributors
115+ Meetups on Docker
21M+ Downloads
13K+ Projects on GitHub

Partners

rackspace HOSTING · IBM · openstack · Atlassian · redhat · amazon webservices · ubuntu · Microsoft Azure · Chef · SALTSTACK · puppet labs

Users

ebay · Spotify · RelateIQ · yelp · mailgun · GILT · New Relic · auto.com · GROUPON · OpenTable · Google Cloud Platform



Support

Enterprise Support
Robust Documentation
Implementation, Integration, Training
Network of Partners

The Docker Platform

Docker Engine
Docker Hub

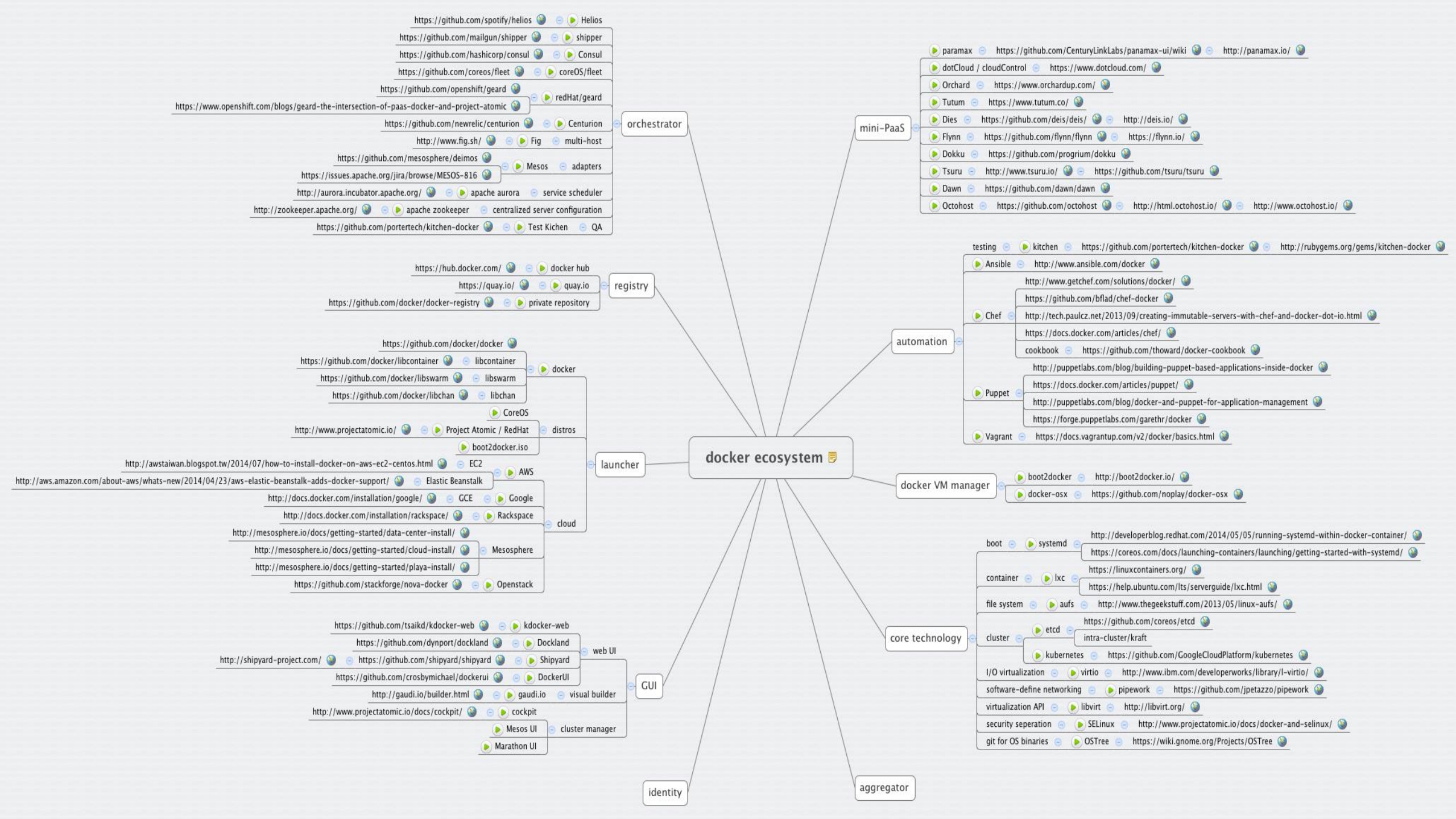
Build, Ship, and Run

Content

redis · NGINX

Official Repos &30 K Dockerized Apps

Java · Ruby · MySQL · RAILS · PostgreSQL · WordPress · node · mongoDB


docker ecosystem
orchestrator
https://github.com/spotify/helios
Helios
https://github.com/mailgun/shipper
shipper
https://github.com/hashicorp/consul
Consul
https://github.com/coreos/fleet
coreOS/fleet
https://github.com/openshift/geard
redHat/geard
https://www.openshift.com/blogs/geard-the-intersection-of-paas-docker-and-project-atomic
https://github.com/newrelic/centurion
Centurion
http://www.fig.sh/
Fig
multi-host
https://github.com/mesosphere/deimos
Mesos
adapters
https://issues.apache.org/jira/browse/MESOS-816
http://aurora.incubator.apache.org/
apache aurora
service scheduler
http://zookeeper.apache.org/
apache zookeeper
centralized server configuration
https://github.com/portertech/kitchen-docker
Test Kichen
QA
registry
https://hub.docker.com/
docker hub
https://quay.io/
quay.io
https://github.com/docker/docker-registry
private repository
launcher
https://github.com/docker/docker
https://github.com/docker/libcontainer
libcontainer
https://github.com/docker/libswarm
libswarm
https://github.com/docker/libchan
libchan
docker
CoreOS
http://www.projectatomic.io/
Project Atomic / RedHat
boot2docker.iso
distros
http://awstaiwan.blogspot.tw/2014/07/how-to-install-docker-on-aws-ec2-centos.html
EC2
http://aws.amazon.com/about-aws/whats-new/2014/04/23/aws-elastic-beanstalk-adds-docker-support/
Elastic Beanstalk
AWS
http://docs.docker.com/installation/google/
GCE
Google
http://docs.docker.com/installation/rackspace/
Rackspace
http://mesosphere.io/docs/getting-started/data-center-install/
http://mesosphere.io/docs/getting-started/cloud-install/
http://mesosphere.io/docs/getting-started/playa-install/
Mesosphere
https://github.com/stackforge/nova-docker
Openstack
cloud
GUI
https://github.com/tsaikd/kdocker-web
kdocker-web
https://github.com/dynport/dockland
Dockland
http://shipyard-project.com/
https://github.com/shipyard/shipyard
Shipyard
https://github.com/crosbymichael/dockerui
DockerUI
web UI
http://gaudi.io/builder.html
gaudi.io
visual builder
http://www.projectatomic.io/docs/cockpit/
cockpit
Mesos UI
Marathon UI
cluster manager
identity
mini-PaaS
paramax
https://github.com/CenturyLinkLabs/panamax-ui/wiki
http://panamax.io/
dotCloud / cloudControl
https://www.dotcloud.com/
Orchard
https://www.orchardup.com/
Tutum
https://www.tutum.co/
Dies
https://github.com/deis/deis/
http://deis.io/
Flynn
https://github.com/flynn/flynn
https://flynn.io/
Dokku
https://github.com/progrium/dokku
Tsuru
http://www.tsuru.io/
https://github.com/tsuru/tsuru
Dawn
https://github.com/dawn/dawn
Octohost
https://github.com/octohost
http://html.octohost.io/
http://www.octohost.io/
automation
testing
kitchen
https://github.com/portertech/kitchen-docker
http://rubygems.org/gems/kitchen-docker
Ansible
http://www.ansible.com/docker
Chef
http://www.getchef.com/solutions/docker/
https://github.com/bflad/chef-docker
http://tech.paulcz.net/2013/09/creating-immutable-servers-with-chef-and-docker-dot-io.html
https://docs.docker.com/articles/chef/
cookbook
https://github.com/thoward/docker-cookbook
Puppet
http://puppetlabs.com/blog/building-puppet-based-applications-inside-docker
https://docs.docker.com/articles/puppet/
http://puppetlabs.com/blog/docker-and-puppet-for-application-management
https://forge.puppetlabs.com/garethr/docker
Vagrant
https://docs.vagrantup.com/v2/docker/basics.html
docker VM manager
boot2docker
http://boot2docker.io/
docker-osx
https://github.com/noplay/docker-osx
core technology
boot
systemd
http://developerblog.redhat.com/2014/05/05/running-systemd-within-docker-container/
https://coreos.com/docs/launching-containers/launching/getting-started-with-systemd/
container
lxc
https://linuxcontainers.org/
https://help.ubuntu.com/lts/serverguide/lxc.html
file system
aufs
http://www.thegeekstuff.com/2013/05/linux-aufs/
cluster
etcd
https://github.com/coreos/etcd
intra-cluster/kraft
kubernetes
https://github.com/GoogleCloudPlatform/kubernetes
I/O virtualization
virtio
http://www.ibm.com/developerworks/library/l-virtio/
software-define networking
pipework
https://github.com/jpetazzo/pipework
virtualization API
libvirt
http://libvirt.org/
security seperation
SELinux
http://www.projectatomic.io/docs/docker-and-selinux/
git for OS binaries
OSTree
https://wiki.gnome.org/Projects/OSTree
aggregator

什么是Docker?

English interpretation

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications. Consisting of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows, Docker enables apps to be quickly assembled from components and eliminates the friction between development, QA, and production environments. As a result, IT can ship faster and run the same app, unchanged, on laptops, data center VMs, and any cloud.

中文解释

Docker是Docker公司开源的一个基于轻量级虚拟化技术的容器引擎项目,整个项目基于Go语言开发，并遵从Apache 2.0协议。

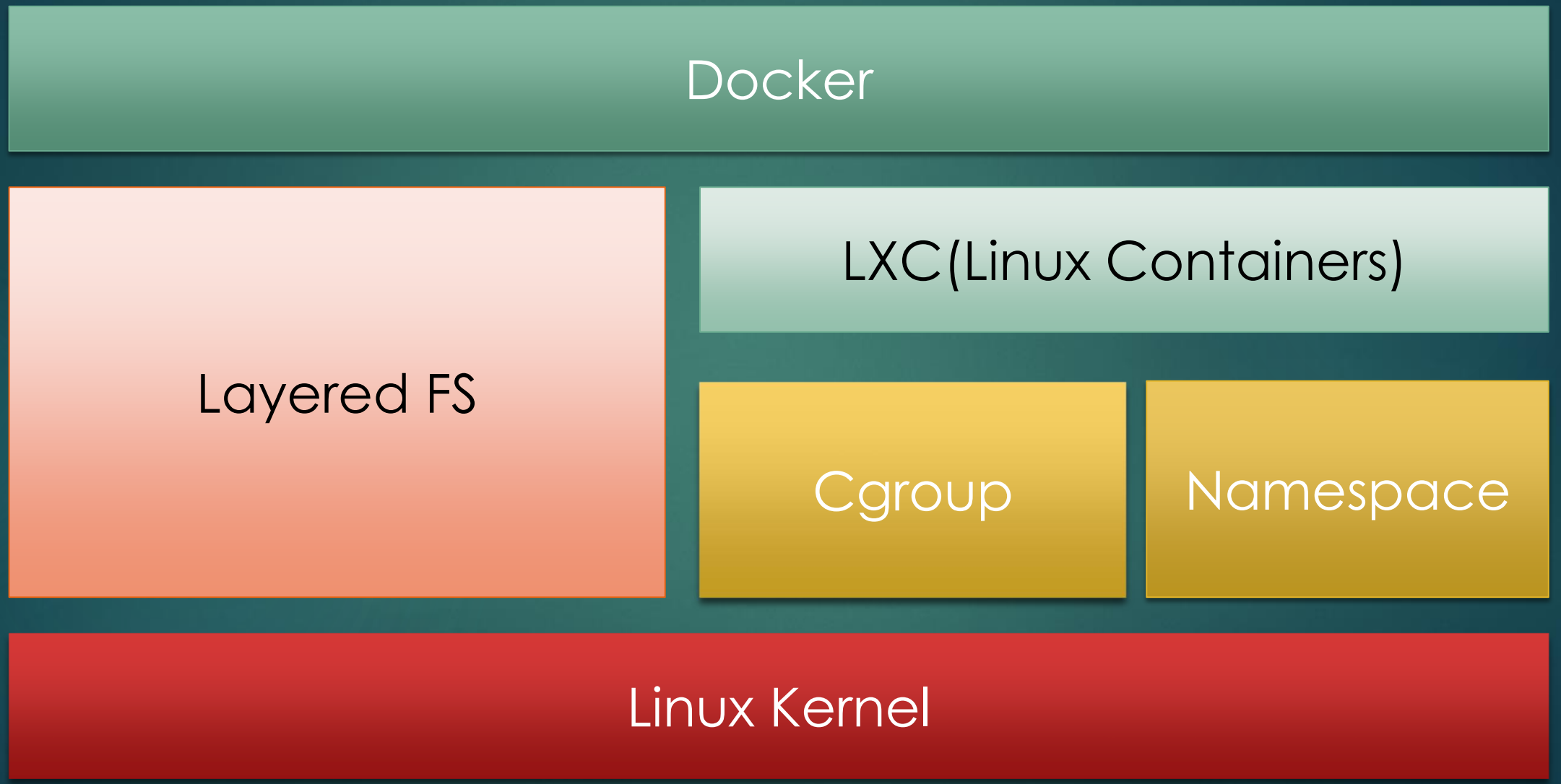
目前，Docker可以在容器内部快速自动化部署应用，并可以通过内核虚拟化技术（namespaces及cgroups等）来提供容器的资源隔离与安全保障等。

由于Docker通过操作系统层的虚拟化实现隔离，所以Docker容器在运行时，不需要类似虚拟机（VM）额外的操作系统开销，提高资源利用率，并且提升诸如IO等方面的性能。

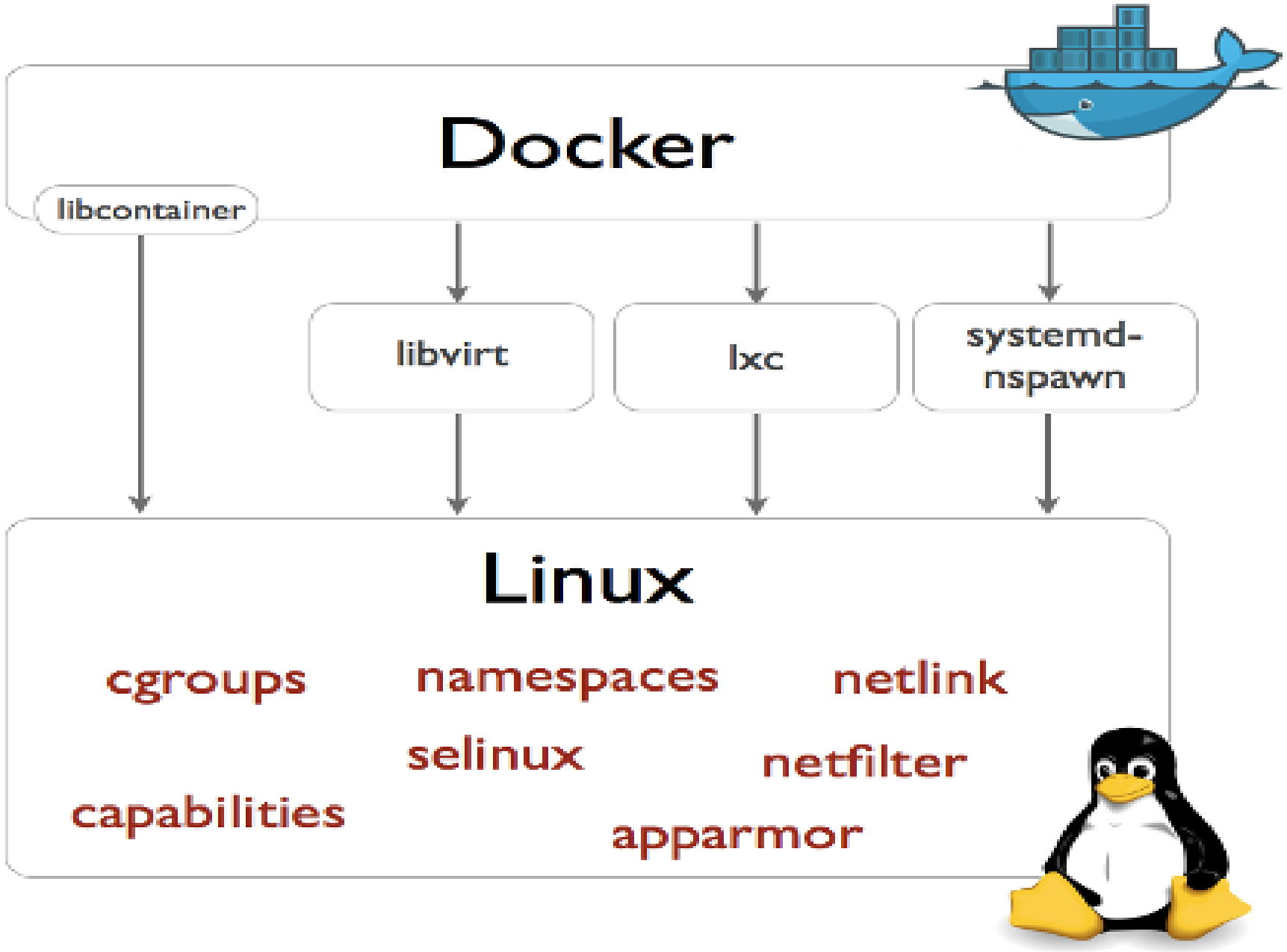
Docker应用场景

- web应用的自动化打包和发布
- 自动化测试和持续集成、发布
- 在服务型环境中部署和调整数据库或其他的后台应用
- 从头编译或者扩展现有的OpenShift或Cloud Foundry平台来搭建自己的PaaS环境。

Docker总架构图


Docker
Layered FS
LXC(Linux Containers)
Cgroup
Namespace
Linux Kernel

Docker On LinuX


Docker
libcontainer
libvirt
lxc
systemd-
nspawn
Linux
cgroups
namespaces
netlink
selinux
netfilter
capabilities
apparmor

Docker 的特征

Docker有不少有趣的功能，Docker特性主要包括以下几点：

- 速度飞快以及优雅的隔离框架
- 物美价廉
- CPU/内存的低消耗
- 快速开/关机
- 跨云计算基础架构

Docker 组件与元素

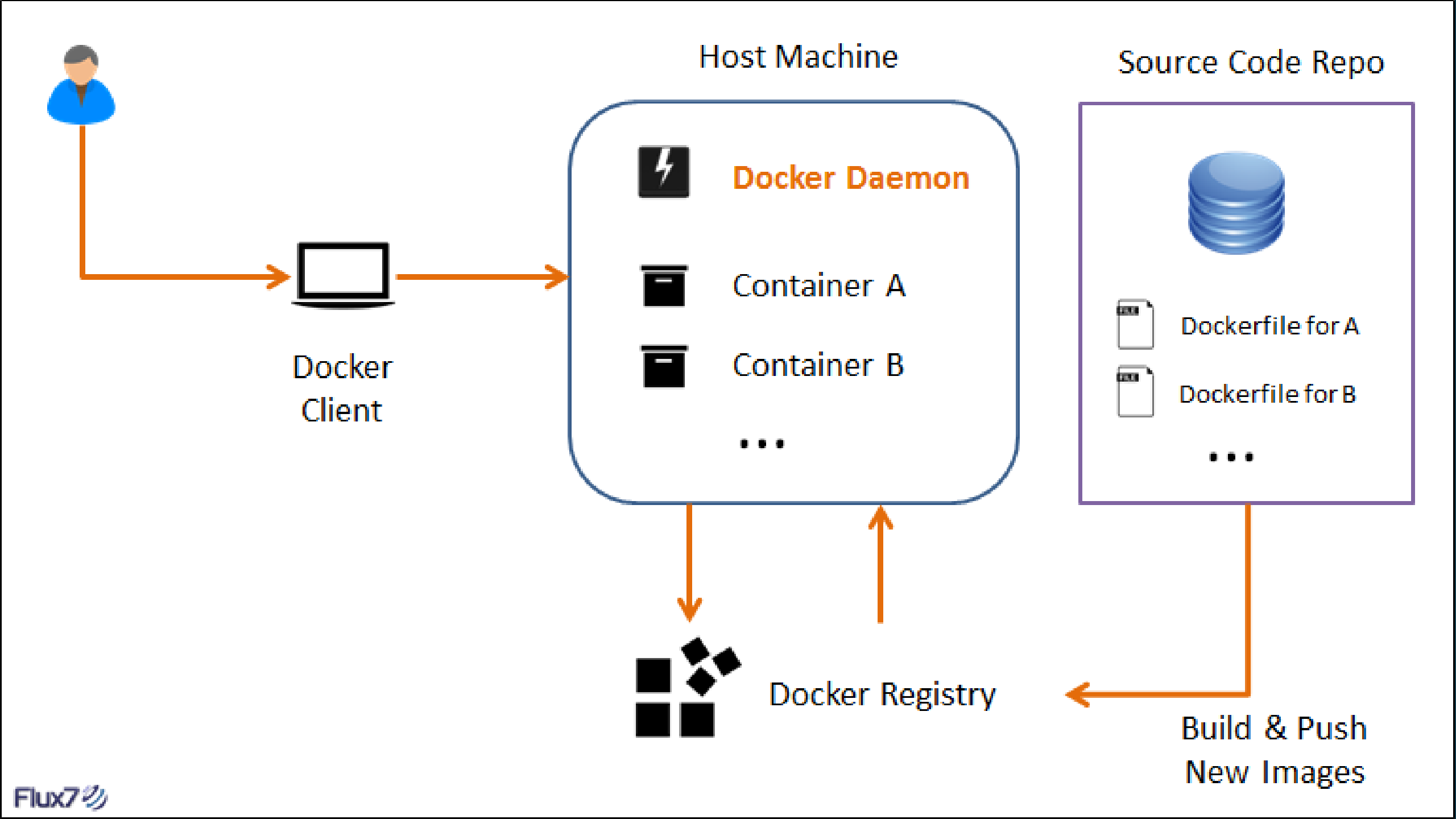
三个基本组件

- Docker Client 是用户界面，它支持用户与Docker Daemon之间通信。
- Docker Daemon运行于主机上，处理服务请求。
- Docker Index是中央registry，支持拥有公有与私有访问权限的Docker容器镜像的备份

三个基本元素

- Docker Containers负责应用程序的运行，包括操作系统、用户添加的文件以及元数据。
- Docker Images是一个只读模板，用来运行Docker容器。
- DockerFile是文件指令集，用来说明如何自动创建Docker镜像。

Docker 工作方式


Host Machine
Source Code Repo
Docker Daemon
Container A
Container B
...
Docker
Client
Dockerfile for A
Dockerfile for B
...
Docker Registry
Build & Push
New Images

Docker的支柱

Docker通过作系统如下功能来提高容器技术效率

- Namespaces 充当隔离的第一级。确保一个容器中运行一个进程而且不能看到或影响容器外的其它进程。
- Control Groups是LXC的重要组成部分，具有资源核算与限制的关键功能。
- UnionFS（文件系统)）作为容器的构建块。为了支持Docker的轻量级以及速度快的特性，它创建层与用户。

Docker如何运行APP

- 构建一个镜像。
- 运行容器。

这些步骤的都是从Docker Client的命令开始的。Docker Client使用的是Docker二进制文件。在基础层面上，Docker Client会告诉Docker Daemon需要创建的镜像以及需要在容器内运行的命令。当Daemon收到创建镜像的信号后，会进行如下操作

第1步：构建镜像

如前面所述，Docker Image是一个构建容器的只读模板，它包含了容器启动所需的所有信息，包括运行哪些进程和配置数据。

所有的镜像都会基于一个基本镜像构建，紧接着会根据Dockerfile中的指令创建模板，对于每个指令，在镜像上创建一个新的层。

一旦镜像创建完成，就可以将它们推送到中央registry：Docker Index，以供他人使用。然而，Docker Index为镜像提供了两个级别的访问权限：公有和私有访问。您可以将镜像存储在私有仓库。Docker官网有私有仓库的套餐可以供你选择。总之，公有库是可搜索和可重复使用的，而私有库只能给拥有权限的成员访问。Docker Client可用于Docker Index内的镜像搜索。

第2步：运行容器

运行容器源于我们在第一步中创建的镜像。当一个容器被启动后，一个读写层会被添加到镜像的顶层。当分配合适的网络和IP地址后，最应用程序就可以在容器中运行了。

如果你还是有点不解，先别急，在接下来的内容中我们会和你分享很多的实战案例。

目前为止，我们已经介绍了Docker的基本概念，接下来，让我们一起安装Docker！

安装Docker

Ubuntu

- 1、更新Ubuntu内核
  使用如下命令行更新内核至3.8.0-25
  sudo apt-get install linux-image-3.8.0-25-generic
  sudo apt-get install linux-headers-3.8.0-25-generic
  完成后重启电脑，通过命令 “uname -r” 来查看内核是否成功更新。
- 2、安装lxc-docker
  root@ubuntu: sudo apt-get install software-properties-common #增加 add-apt-repository 命令
  root@ubuntu: sudo apt-get install python-software-properties
  root@ubuntu: sudo add-apt-repository ppa:dotcloud/lxc-docker #增加一个ppa源，如：ppa:user/ppa-name
  root@ubuntu: sudo apt-get update #更新系统
  root@ubuntu: sudo apt-get install lxc-docker

安装Docker

CentOS7

Docker 软件包已经包括在默认的 CentOS-Extras 软件源里。因此想要安装 docker，只需要运行下面的 yum 命令：

[root@localhost ~]# yum install docker

启动 Docker 服务

安装完成后，使用下面的命令来启动 docker 服务，并将其设置为开机启动：

[root@localhost ~]# service docker start

[root@localhost ~]# chkconfig docker on

（LCTT 译注：此处采用了旧式的 sysv 语法，如采用CentOS 7中支持的新式 systemd 语法，如下：

[root@localhost ~]# systemctl start docker.service

[root@localhost ~]# systemctl enable docker.service

Docker 启动

```
[root@heidsoft ~]# docker -d
2015/01/04 23:31:16 docker daemon: 1.2.0 2a2f26c/1.2.0; execdriver: native; graphdriver:
[c472645d] +job serveapi(unix:///var/run/docker.sock)
[info] Listening for HTTP on unix (/var/run/docker.sock)
[c472645d] +job init_networkdriver()
[c472645d.init_networkdriver()] creating new bridge for docker0
[c472645d.init_networkdriver()] getting iface addr
[c472645d] -job init_networkdriver() = OK (0)
[info] Loading containers:
..[error] driver.go:141 Warning: error unmounting device 3ec11cd42081cbfa89ddb9bdc6b6369e3160b74ded5e64
b0f7772923029751f5: UnmountDevice: device not-mounted id 3ec11cd42081cbfa89ddb9bdc6b6369e3160b74ded5e64
b0f7772923029751f5

[info] : done.
[c472645d] +job acceptconnections()
[c472645d] -job acceptconnections() = OK (0)
[info] GET /v1.14/images/json
[c472645d] +job images()
[c472645d] -job images() = OK (0)
```

Docker version

Docker系统有两个程序：docker服务端和docker客户端。其中docker服务端是一个服务进程，管理着所有的容器。docker客户端则扮演着docker服务端的远程控制器，可以用来控制docker的服务端进程。大部分情况下，docker服务端和客户端运行在一台机器上。

```
[root@heidsoft ~]# docker version
Client version: 1.2.0
Client API version: 1.14
Go version (client): go1.3.3
Git commit (client): 2a2f26c/1.2.0
OS/Arch (client): linux/amd64
2015/01/04 23:18:32 Get http:///var/run/docker.sock/v1.14/version: dial unix /var/run/docker.sock: no s
uch file or directory
```

Docker images

列出当前可用的镜像

```
[root@heidsoft ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
centos              latest              ae0c2d0bdc10        8 weeks ago
[root@heidsoft ~]#
```

Docker 下载容器镜像

下载镜像的命令非常简单，使用docker pull命令即可。(译者按：docker命令和git有一些类似的地方）。在docker的镜像索引网站上面，镜像都是按照**用户名/镜像名**的方式来存储的。有一组比较特殊的镜像，比如ubuntu这类基础镜像，经过官方的验证，值得信任，可以直接用**镜像名**来检索到。

docker容器的hello world!

在docker容器中运行hello world!

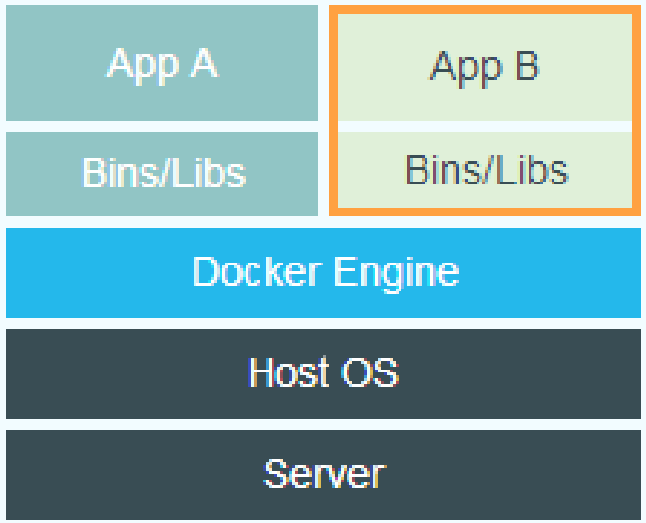
docker容器可以理解为在沙盒中运行的进程。这个沙盒包含了该进程运行所必须的资源，包括文件系统、系统类库、shell 环境等等。但这个沙盒默认是不会运行任何程序的。你需要在沙盒中运行一个进程来启动某一个容器。这个进程是该容器的唯一进程，所以当该进程结束的时候，容器也会完全的停止。

Docker on LXC

LXC is a userspace interface for the Linux kernel containment features.
Through a powerful API and simple tools, it lets Linux users easily create and manage system or application containers.

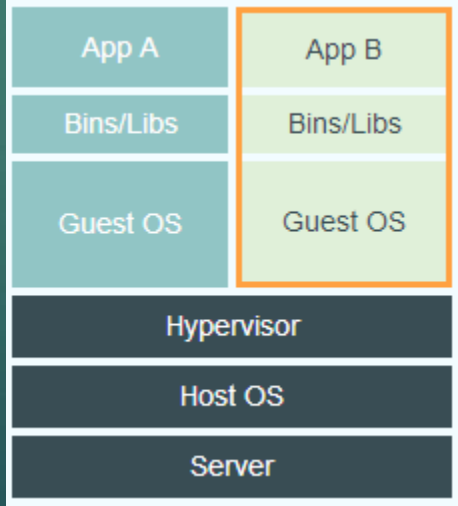
LXC是Linux内核的用户空间接口控制功能。通过一个强大的API和简单的工具,它允许Linux用户轻松地创建和管理系统或应用程序的容器。

Docker VS VM



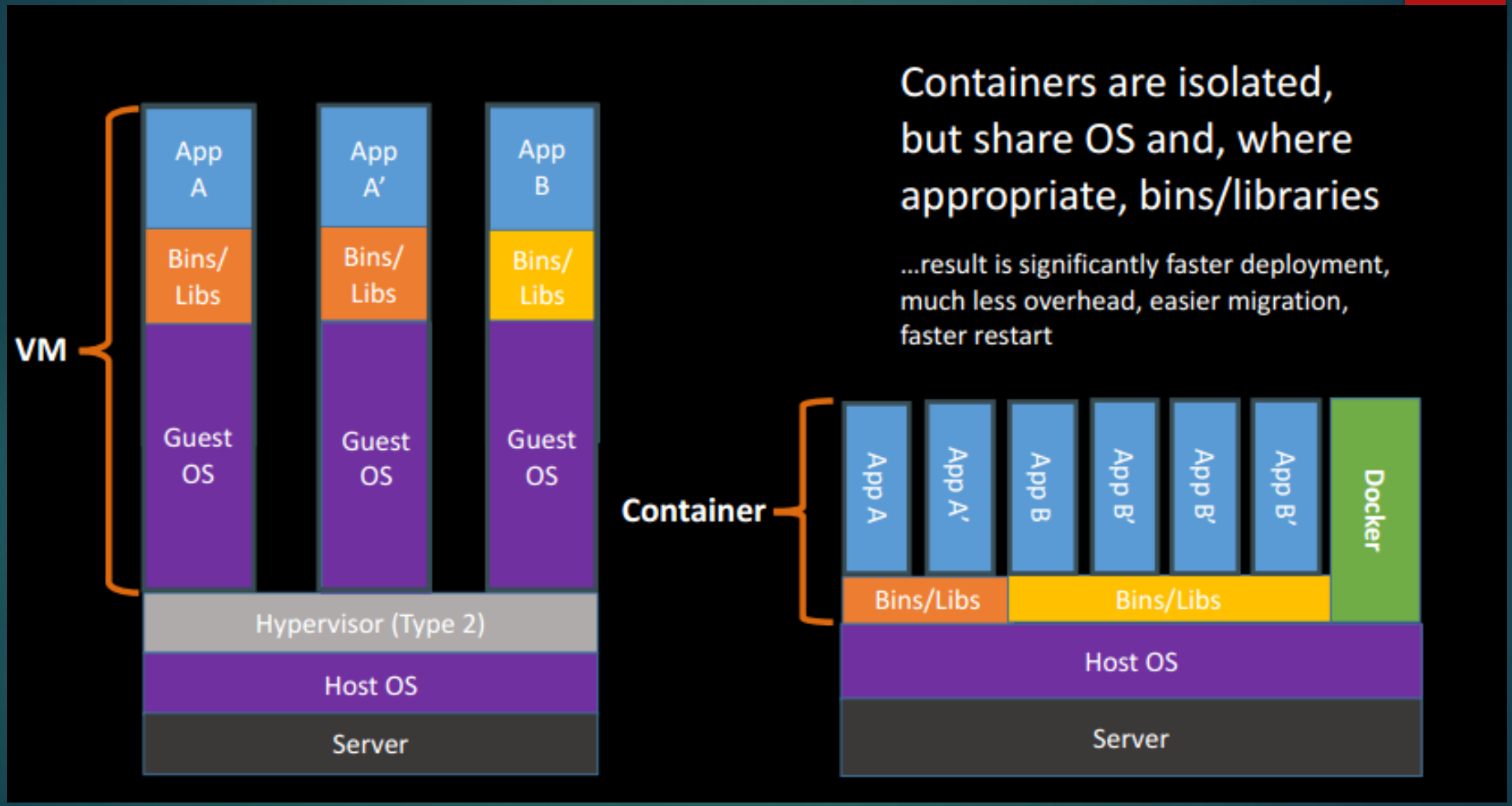
Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.


VM
App A
App A'
App B
Bins/ Libs
Bins/ Libs
Bins/ Libs
Guest OS
Guest OS
Guest OS
Hypervisor (Type 2)
Host OS
Server
Containers are isolated, but share OS and, where appropriate, bins/libraries
...result is significantly faster deployment, much less overhead, easier migration, faster restart
Container
App A
App A'
App B
App B'
App B'
App B'
Docker
Bins/Libs
Bins/Libs
Host OS
Server

Docker下的开发模式

- 共享基础容器
- 共享卷（Volume）开发容器
- 开发工具容器
- 不同环境下的测试容器
- 构建容器
- 安装容器
- 整合默认服务（Default-Service-In-A-Box）的容器和基础设施/粘合（Glue）容器

http://www.hokstad.com/docker/patterns

参考资料

https://www.docker.io/

https://github.com/dotcloud/docker

http://bit.ly/dockersources

http://lwn.net/Articles/199643/

http://lwn.net/Articles/236038/

http://en.wikipedia.org/wiki/Operating_system-level_virtualization

https://linuxcontainers.org/

http://en.wikipedia.org/wiki/Cgroups

http://en.wikipedia.org/wiki/Aufs