Docker入门与实践

第一章：Docker基础篇

1、dotCloud成立
2、开发PaaS服务

1、核心引擎开源
2、程序猿惊呼叫绝
3、互联网大佬纷纷接入
4、引擎名为Docker，Go语言

1、8月把Paas出售给cloundControl公司
2、专心研发Docker维护相关社区

2010 2011 2012 2013 2014 2015

1、微软Azure
2、Amazon EC2
3、Google GAE
4、VMWare CloudFoundry
5、dotCloud举步维艰

1、10月dotCloud公司更名为Docker，Docker正式诞生

1、近10年来发展最快的开源框架
2、国内互联网大会专设Docker专场
3、国内社区活动场场爆满

我这里好好的，为什么到你那里不行了？

Docker解决了这个问题

程序跑起来的几个要素：

代码　　依赖服务　　运行环境　　配置

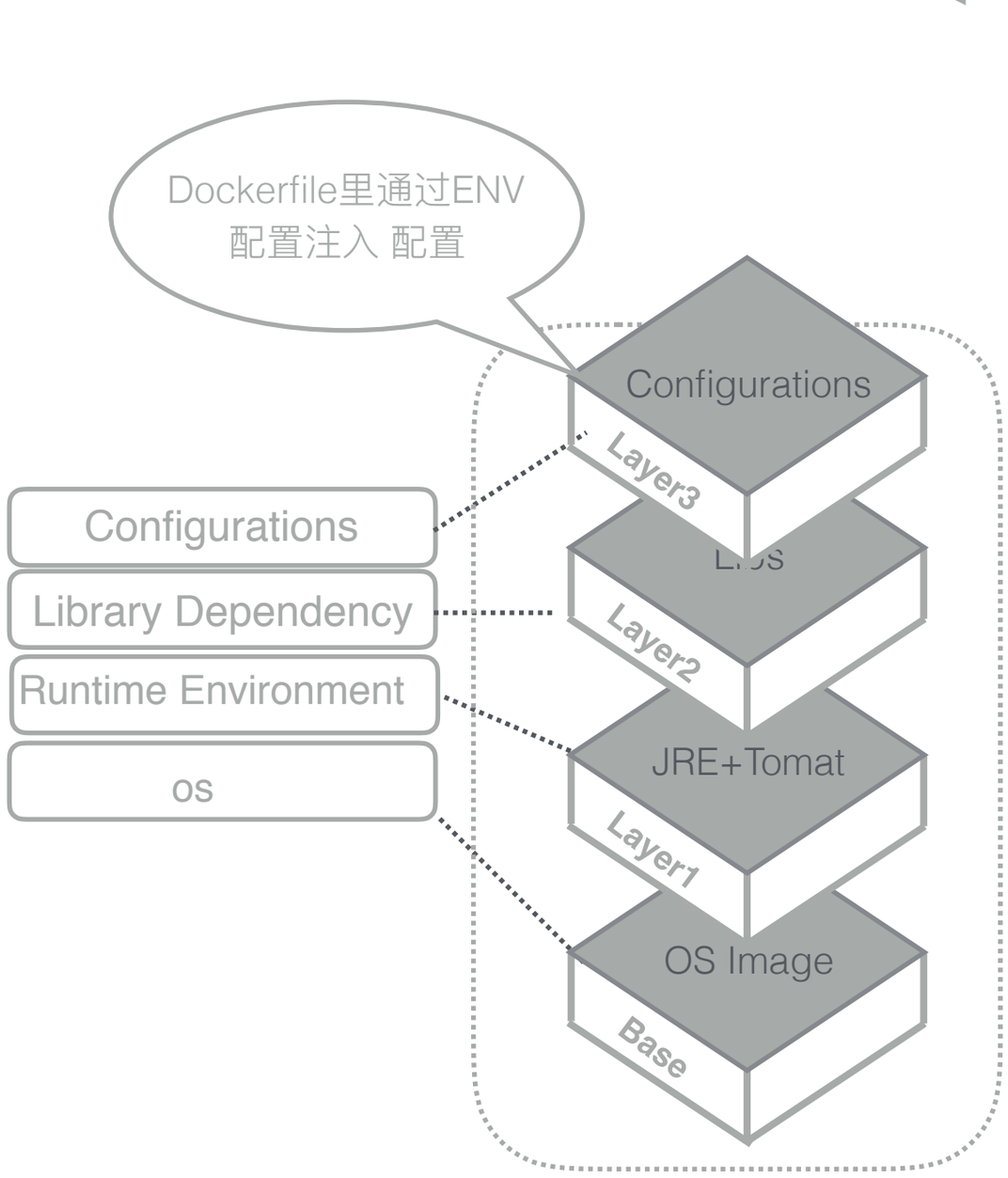
如果代码和依赖服务相同，那么程序跑不起来的原因是：

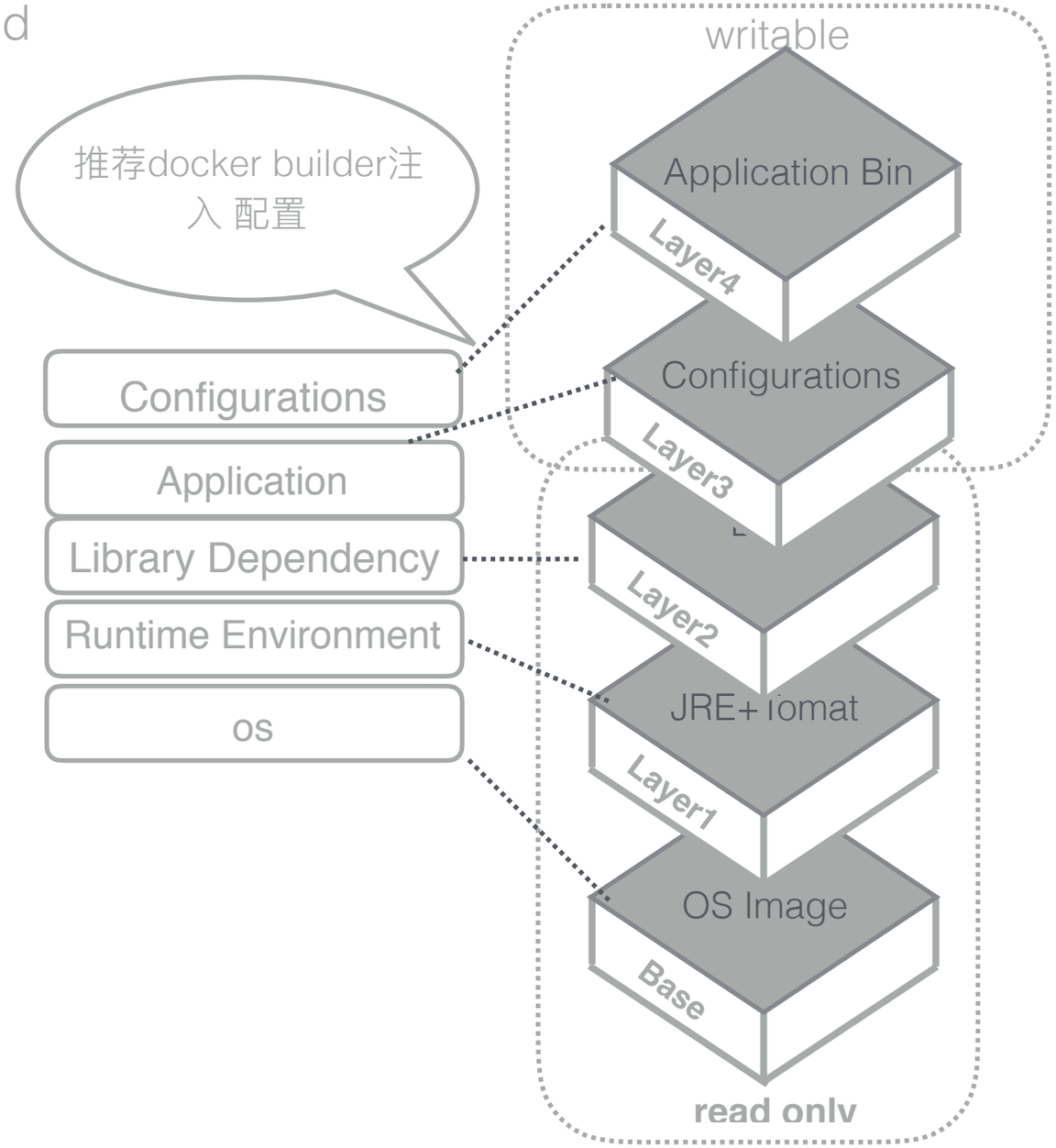
Docker镜像

run

build

Docker容器





- 镜像是个特殊的模版目录
  - 可传递
  - 可版本管理
  - 分层的
- 镜像用来创建容器

- 容器是镜像的运行实例
- 应用通过容器运行
  - 启动时,在所有镜像层之上创建一层可写层
  - 镜像是只读的
- 容器可转为镜像

Dockerfile文件

- 作用
  - 构建镜像
  - 配置注入
- 文件格式
  - 文本文件，包含自定义的指令和格式
  - 所有指令
    - FROM、MAINTAINER、RUN、COPY、ADD、ENV ENTROYPOINT、CMD、USER、EXPOSE、WORKDIR、ONBUILD
  - 一指令生成一缓冲层

Dockerfile关键指令释义

- **FROM**

  语法： FROM <image> [:<tag>]

  解释： 设置要制作的镜像基于哪个基础镜像，FROM指令必须是整个Dockerfile的第一个指令，如果指定的镜像在本地不存在默认会自动从Docker Hub上下载

- **ENV**

  语法： ENV <key> <value>

  解释： ENV指令用于设置环境变量

- **RUN**

  语法： RUN <command> or RUN ["executable", "param1", "param2"]

  解释： RUN指令会在一个新的容器中执行任何命令

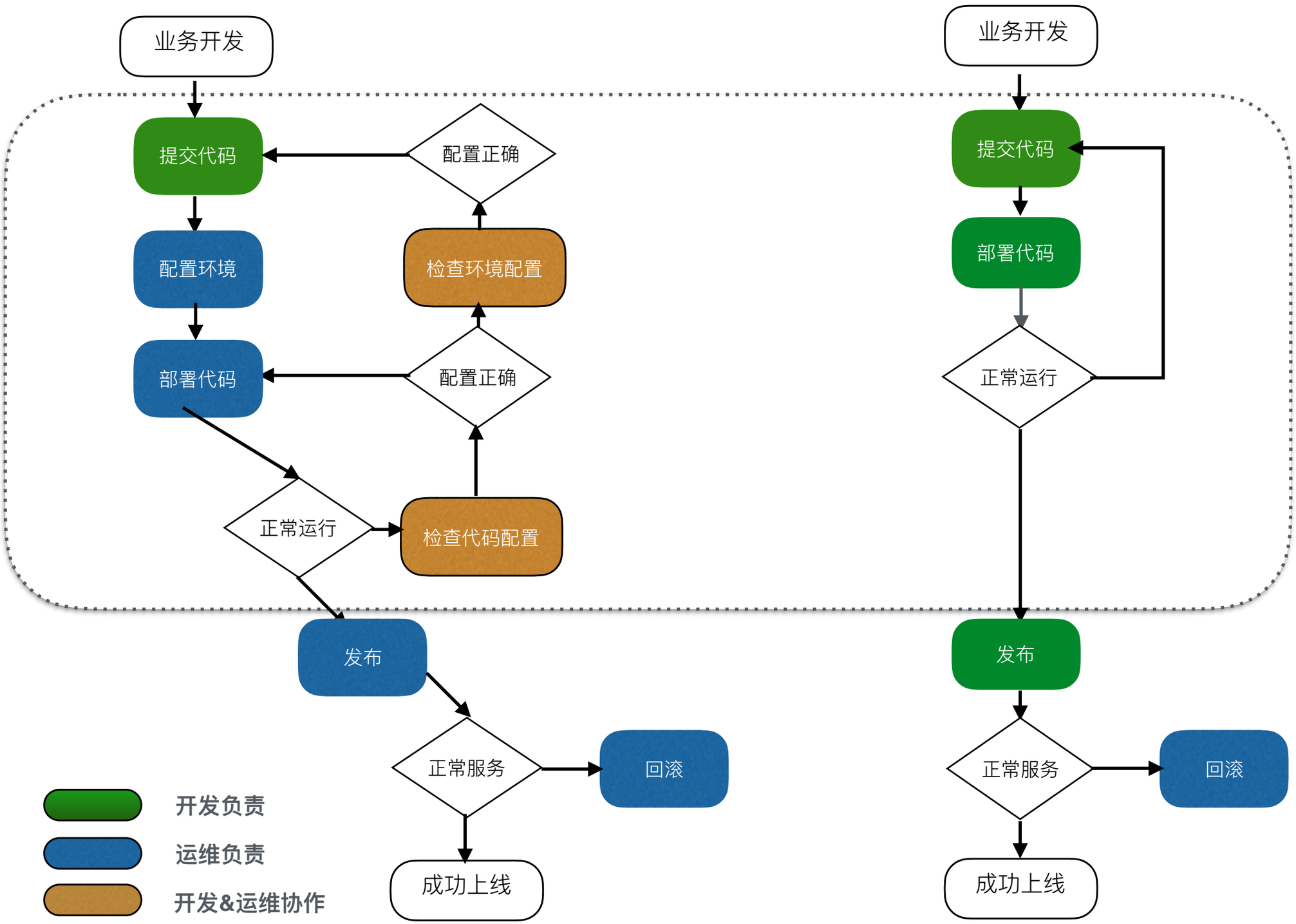
- **EXPOSE**

  语法： EXPOSE <port> [...]

  解释： EXPOSE指令用来告诉Docker这个容器在运行时会监听哪些端口

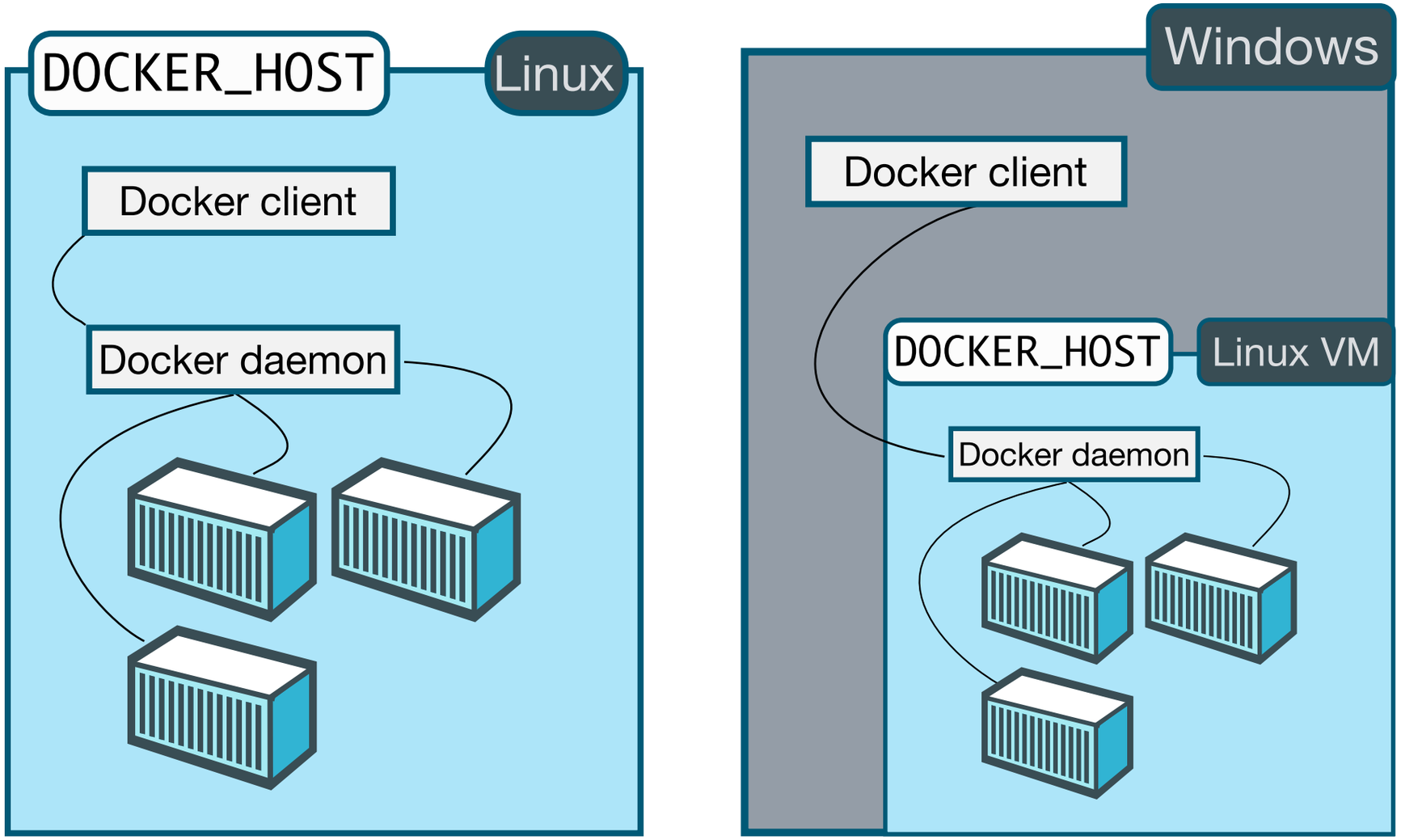
详见参考：https://hujb2000.gitbooks.io/docker-flow-evolution/content/cn/basis/dockerfiledetail.html

运维&开发驱动 VS 开发驱动



开发驱动 => 简化流程降低协作成本

Docker Client & Docker Daemon



- Docker Client
  - 作用

    向Docker daemon发起请求，执行相应的管理操作
  - 存在

    可以是命令行工具

    可以是任何遵循Docker API的客户端

- Docker Daemon
  - 作用

    负责响应来自Docker Client的请求，

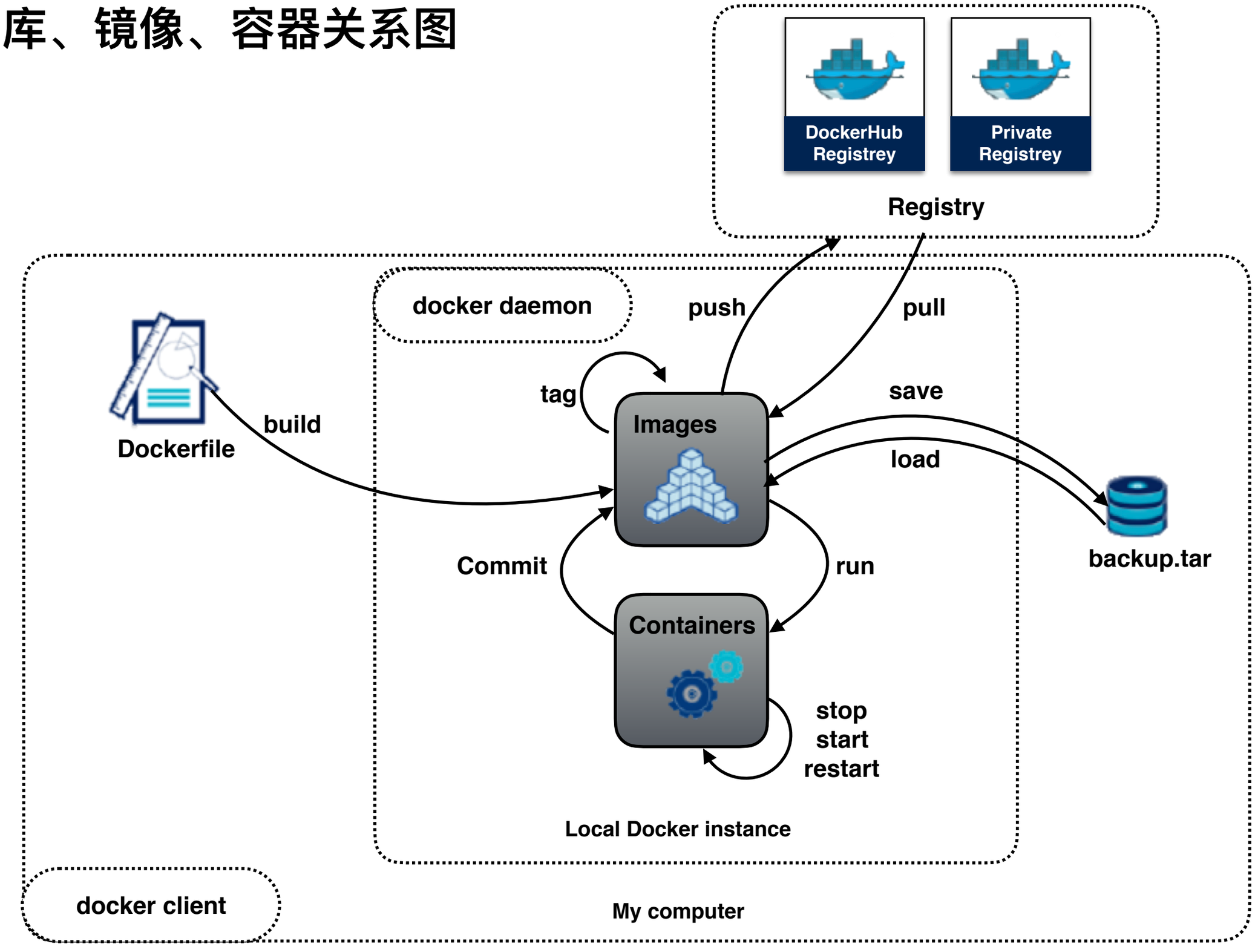
    将请求翻译成系统调用完成容器管理操作
  - 存在

    Docker最核心的后台进程

Docker仓库

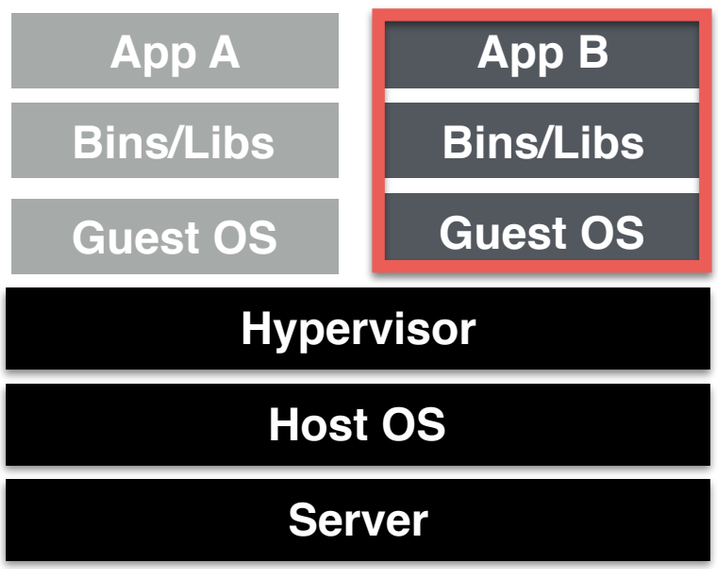
- 集中存放镜像文件的场所
  - 公开仓库
    - 国外: Docker Hub
    - 国内: Docker Pool
  - 私有仓库
    - 蜂巢，希云
    - .....

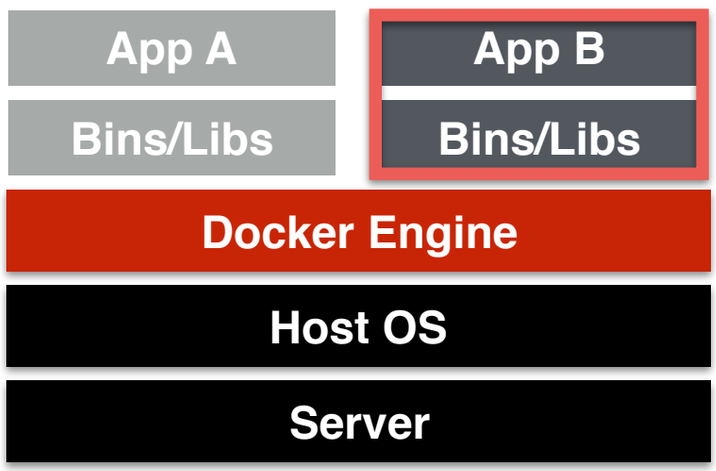
仓库、镜像、容器关系图


DockerHub Registrey
Private Registrey
Registry
docker daemon
push
pull
tag
save
Images
build
Dockerfile
load
backup.tar
Commit
run
Containers
stop
start
restart
Local Docker instance
docker client
My computer

VM VS Docker

对比项	容器技术	虚拟机技术
占用磁 盘空间	小，甚至几十KB(镜像层情况)	非常大，上GB
启动速度	快，几秒种	慢，几分钟
运行状态	运行于宿主机的内核， 容器共享同一个Linux内核	运行于Hypervisior上
并发性	一台宿主机可以启动成千上百个容器	最多几十个虚拟机
性能	接近宿主机本地进程	逊于宿主机
IO/CPU/MEM资源利用率	高	低





天下武功，唯快不破 => Docker完胜

Docker安装

- [ ] Docker安装主要参考官网
  - Mac OS X下安装
    - Docker Toolbox
    - https://docs.docker.com/engine/installation/mac
  - Windwos下安装
    - Docker Toolbox
    - https://docs.docker.com/engine/installation/windows/
  - Debian下安装
    - Docker Toolbox
    - https://docs.docker.com/engine/installation/debian/

镜像常用操作命令

- 拉取镜像
  - Usage:　docker pull [OPTIONS] NAME[:TAGl@DIGEST]

```
hujiabaos-MBP:~ hujiabao$ docker pull mysql:5.7.10
5.7.10: Pulling from library/mysql
523ef1d23f22: Pulling fs layer
140f9bdfeb97: Pulling fs layer
2df110ab8d10: Pulling fs layer
8ce18aa992e7: Pulling fs layer
19aa914a4bb3: Pulling fs layer
46a35d16da7c: Pulling fs layer
60ce6e65025d: Pulling fs layer
56080edc0174: Pulling fs layer
e7be56140f8b: Pulling fs layer
3666f062674e: Pulling fs layer
3c3357b44a48: Pulling fs layer
bd3d56d4cb20: Pulling fs layer
01e70da302a5: Pulling fs layer
d7ef45ff532e: Pulling fs layer
8eb845d8d7f6: Pulling fs layer
d39c3fa09ced: Pulling fs layer
```

镜像常用操作命令

- 列出本地镜像
  - docker images [OPTIONS] [REPOSITORY[:TAG]]

```
^Chujiabaos-MBP:~ hujiabao$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
icp                                 0.0.1               854e7f8c0fe3        8 days ago          888.8 MB
<none>                              <none>              902740558c7f        8 days ago          714 MB
<none>                              <none>              3e2b6b8a7710        8 days ago          714 MB
<none>                              <none>              7ccfcf035821        9 days ago          714 MB
easynode                            0.0.3               e52f92564758        9 days ago          698.4 MB
registry.hz.netease.com/easynode    0.0.3               e52f92564758        9 days ago          698.4 MB
easynode                            0.0.2               5678f4fb5471        9 days ago          682.8 MB
php                                 latest              3df8db6425db        5 weeks ago         484.8 MB
mysql                               latest              a07681abeb6c        5 weeks ago         360.3 MB
java                                latest              d4849089125b        7 weeks ago         642 MB
```

镜像常用操作命令

- 运行镜像
  - Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

```
hujiabaos-MBP:~ hujiabao$ docker run --name mysql3 -e MYSQL_ROOT_PASSWORD='aa' mysql
Initializing database
2016-01-27T16:19:04.723677Z 0 [Warning] InnoDB: New log files created, LSN=45790
2016-01-27T16:19:04.761579Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2016-01-27T16:19:04.826476Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has bee
2016-01-27T16:19:04.828323Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2016-01-27T16:19:04.829073Z 1 [Warning] root@localhost is created with an empty password ! Please consider switching off the --initialize-
2016-01-27T16:19:05.278181Z 1 [Warning] 'user' entry 'root@localhost' ignored in --skip-name-resolve mode.
2016-01-27T16:19:05.278741Z 1 [Warning] 'user' entry 'mysql.sys@localhost' ignored in --skip-name-resolve mode.
2016-01-27T16:19:05.279259Z 1 [Warning] 'db' entry 'sys mysql.sys@localhost' ignored in --skip-name-resolve mode.
2016-01-27T16:19:05.279535Z 1 [Warning] 'proxies_priv' entry '@ root@localhost' ignored in --skip-name-resolve mode.
2016-01-27T16:19:05.279932Z 1 [Warning] 'tables_priv' entry 'sys_config mysql.sys@localhost' ignored in --skip-name-resolve mode.
Database initialized
MySQL init process in progress...
2016-01-27T16:19:06.826662Z 0 [Note] mysqld (mysqld 5.7.10) starting as process 39 ...
2016-01-27T16:19:06.830857Z 0 [Note] InnoDB: PUNCH HOLE support available
2016-01-27T16:19:06.831025Z 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2016-01-27T16:19:06.831559Z 0 [Note] InnoDB: Uses event mutexes
2016-01-27T16:19:06.831958Z 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
2016-01-27T16:19:06.832256Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.8
2016-01-27T16:19:06.832590Z 0 [Note] InnoDB: Using Linux native AIO
```

镜像常用操作命令

- 删除镜像
  - docker rmi [OPTIONS] IMAGE [IMAGE...]

```
hujiabaos-MacBook-Pro:~ hujiabao$ docker rmi -f 902
Deleted: 902740558c7ffa198bca4aa0de8b18410ef452a40bb0bb2f0fdad2854a440958
Deleted: fc9042b635f6036b30ae36db1a24685563d7ee4281f4f328bcea8f9828fabe8f
```

容器常用操作命令

- 启动容器
  - Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

    -t 选项让Docker分配一个伪终端（pseudo-tty）
    并绑定到容器的标准输入上，
    -i 则让容器的标准输入保持打开

```
hujiabaos-MacBook-Pro:~ hujiabao$ docker run -t -i mysql /bin/bash
root@1486897f0658:/#
```

容器常用操作命令

- 进入容器
  - Usage: docker exec -it [containerId|containerName] /bin/bash

```
root@zhapn1:/home/hjb/scripts# docker exec -it db95 /bin/bash
root@db95bc8e5b3a:/#
```

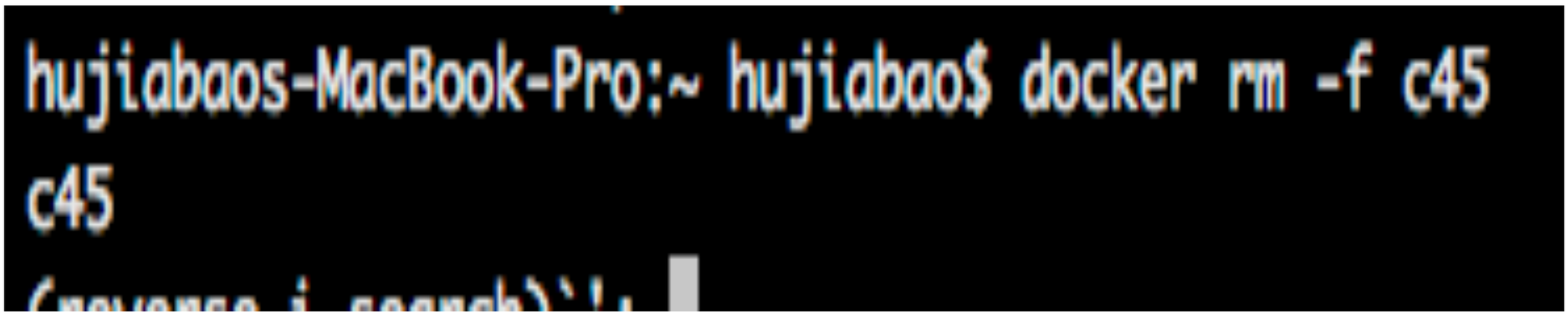
容器常用操作命令

- 查看容器
  - Usage: docker ps -a

```
root@zhapn1:/home/hjb/scripts# docker ps -a
CONTAINER ID        IMAGE                       COMMAND                  CREATED             STATUS              PORTS                                                                                                    NAMES
c3cf68af8713        registry@2.1.1              "/bin/registry /etc/d"   3 weeks ago         Up 3 weeks          0.0.0.0:5000->5000/tcp                                                                                   registry
ba003bd1db40        csphere/csphere:1.0.0       "/bin/csphere-init -i"   8 weeks ago         Up 8 weeks                                                                                                                   csphere-agent
c4be82855ee0        csphere/csphere:1.0.0       "/bin/csphere-init -i"   8 weeks ago         Up 8 weeks          8086/tcp, 27017/tcp, 0.0.0.0:1016->80/tcp                                                                csphere-controller
db95bc8e5b3a        mysql                       "/entrypoint.sh mysql"   8 weeks ago         Up 12 days          0.0.0.0:3306->3306/tcp                                                                                   mysql
e614b75e1a38        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:6674->5672/tcp, 0.0.0.0:16674->15672/tcp              rabbitmqc
b5f76d4979f2        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:6673->5672/tcp, 0.0.0.0:16673->15672/tcp              rabbitmqb
3d1120d8aefb        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:6672->5672/tcp, 0.0.0.0:16672->15672/tcp              rabbitmqa
530391f01c8a        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:5674->5672/tcp, 0.0.0.0:15674->15672/tcp              rabbitmq3
4f5bd0cb6e50        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:5673->5672/tcp, 0.0.0.0:15673->15672/tcp              rabbitmq2
fae7e5e5b26b        rabbitmq:3.5-management     "/docker-entrypoint.s"   8 weeks ago         Up 8 weeks          4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:15672->15672/tcp              rabbitmq1
92e433dd824f        phensley/docker-dns         "./dockerdns --domain"   8 weeks ago         Up 8 weeks                                                                                                                   dns
```

容器常用操作命令

- 删除容器
  - docker rm [OPTIONS] CONTAINER [CONTAINER...]



容器常用操作命令

- 容器转为镜像
  - Usage:　　docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

```
hujiabaos-MacBook-Pro:books hujiabao$ docker commit -m "hello, mysql" -a "hujiabao" 6cb mesql:5.7.10
d9f5096549424db6b5fa876f3b4dc34d1438e7430e801d22c742ac225be74dc2
```

Dockerfile使用举例

```
FROM node@5.5.0-wheezy                                  # 基于哪个基础镜像构建

MAINTAINER hujiabao                                     # 设置作者信息

RUN mkdir -p /usr/src/app                               # 运行命令建立目录

COPY . /usr/src/app                                     # 拷贝宿主机当前目录到容器/usr/src/app目录

ENV VERSION 1.0.0                                       # 设置环境亦是VERSION=1.0.0

RUN npm install                                         # 安装依赖包

WORKDIR /usr/src/app/netease/bin                        # 设置当前工作目录

USER  node                                              # 设置用户或uid来运行生成的镜像和执行RUN指令

EXPOSE 80                                               # 告诉Docker daemon这个容器在运行时会监听哪些端口

ONBUILD echo 'I lurking in the parent image'            # 设计父镜像时，在子镜像构建时被触发的父镜像中的指令

ENTRYPOINT ["./start.sh"]                               # 容启动时执行的最后一条命令
```

谢谢

THANK YOU