Alibaba Group
阿里巴巴集团

云上应用Docker化持续交付实践

刘昕（唐容）

自我介绍

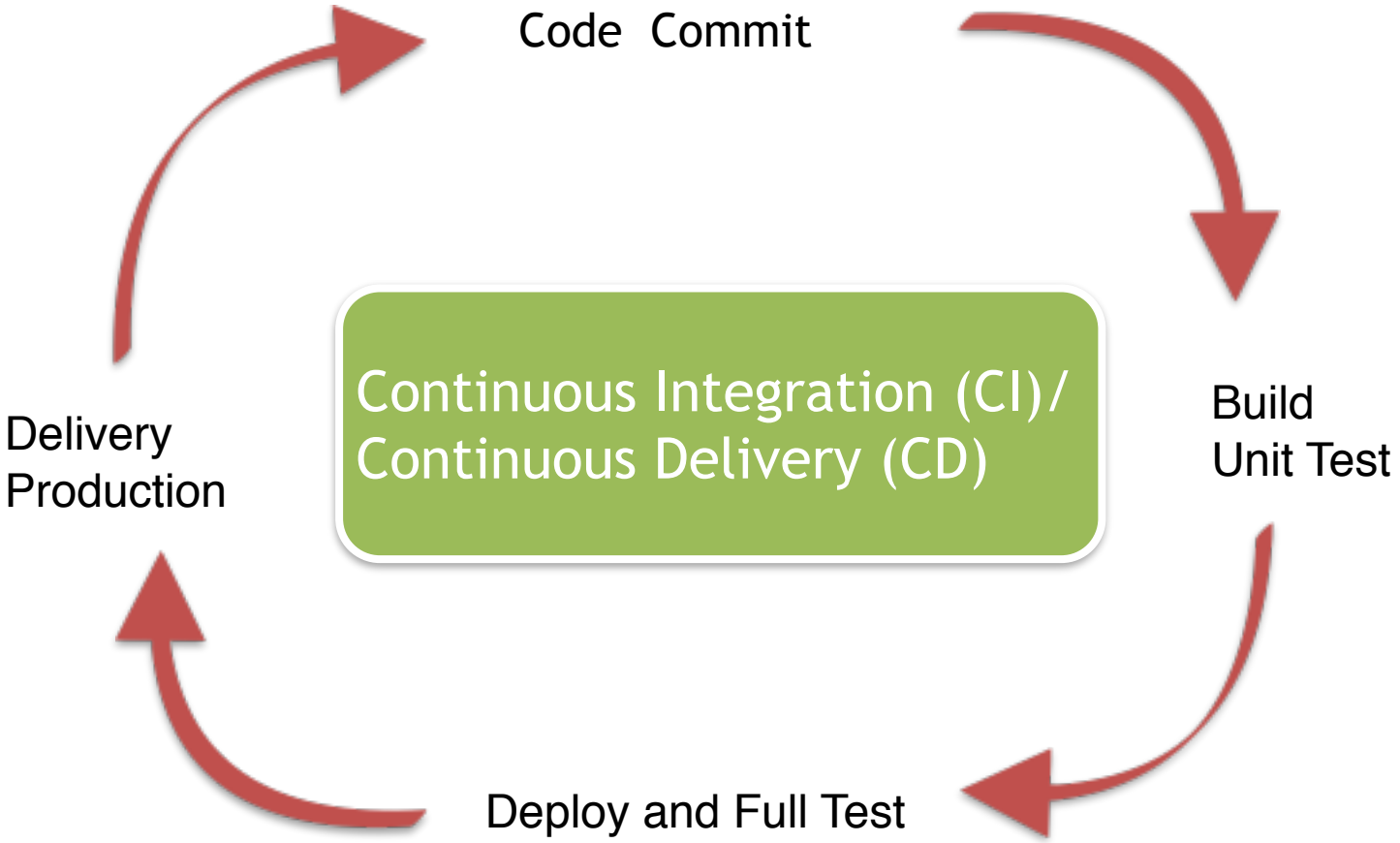
刘昕（唐容）

- 阿里巴巴技术专家。目前负责工程效能团队的产品开发工作，主要包含阿里云持续交付平台（CRP）及云Code 平台。
- 曾负责淘宝日常测试环境，集团源码管理，编译系统等基础设施的建设，以及阿里云云计算等部门的配置管理工作。

目录

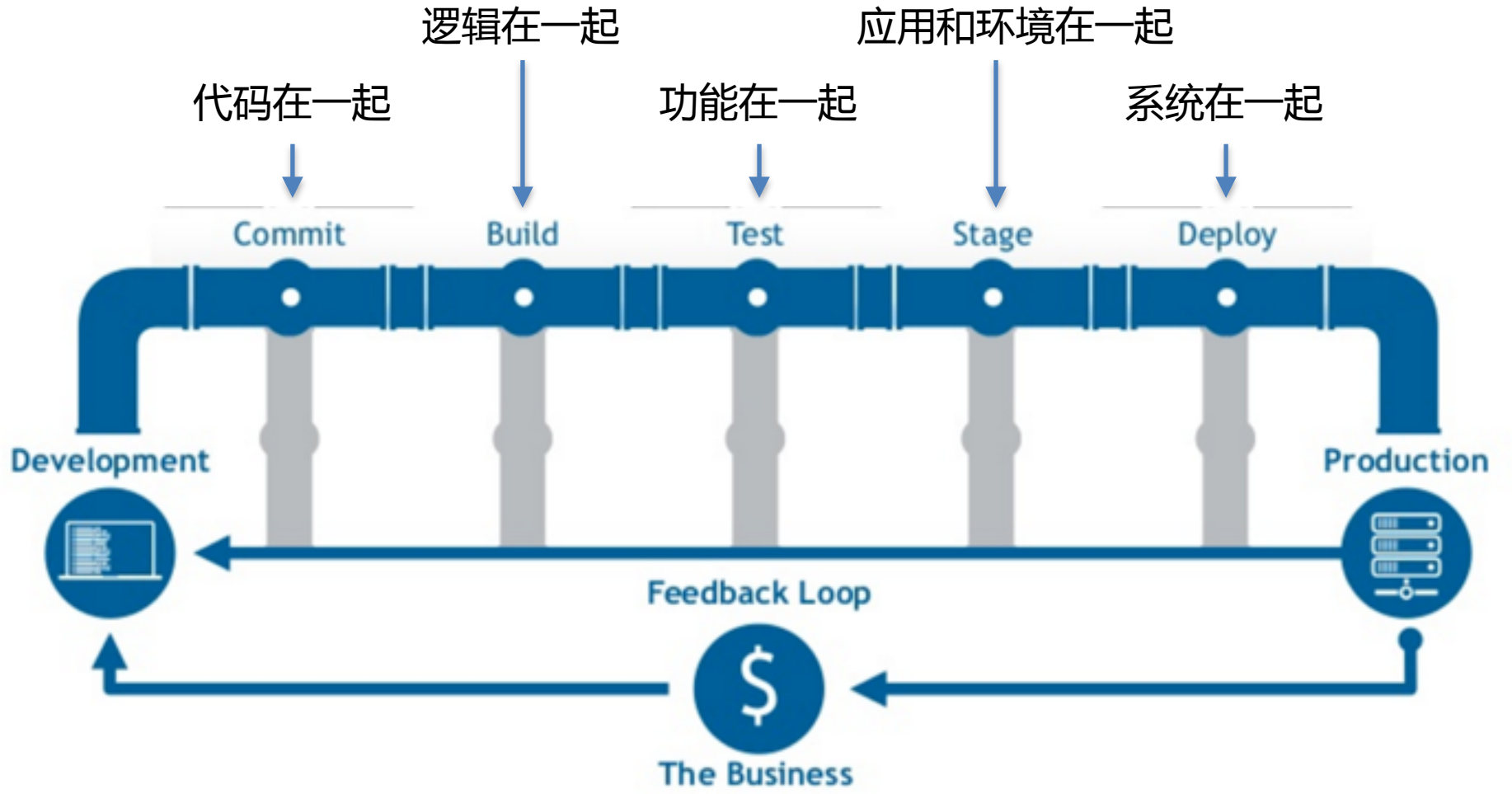
研发过程的困境

Solution

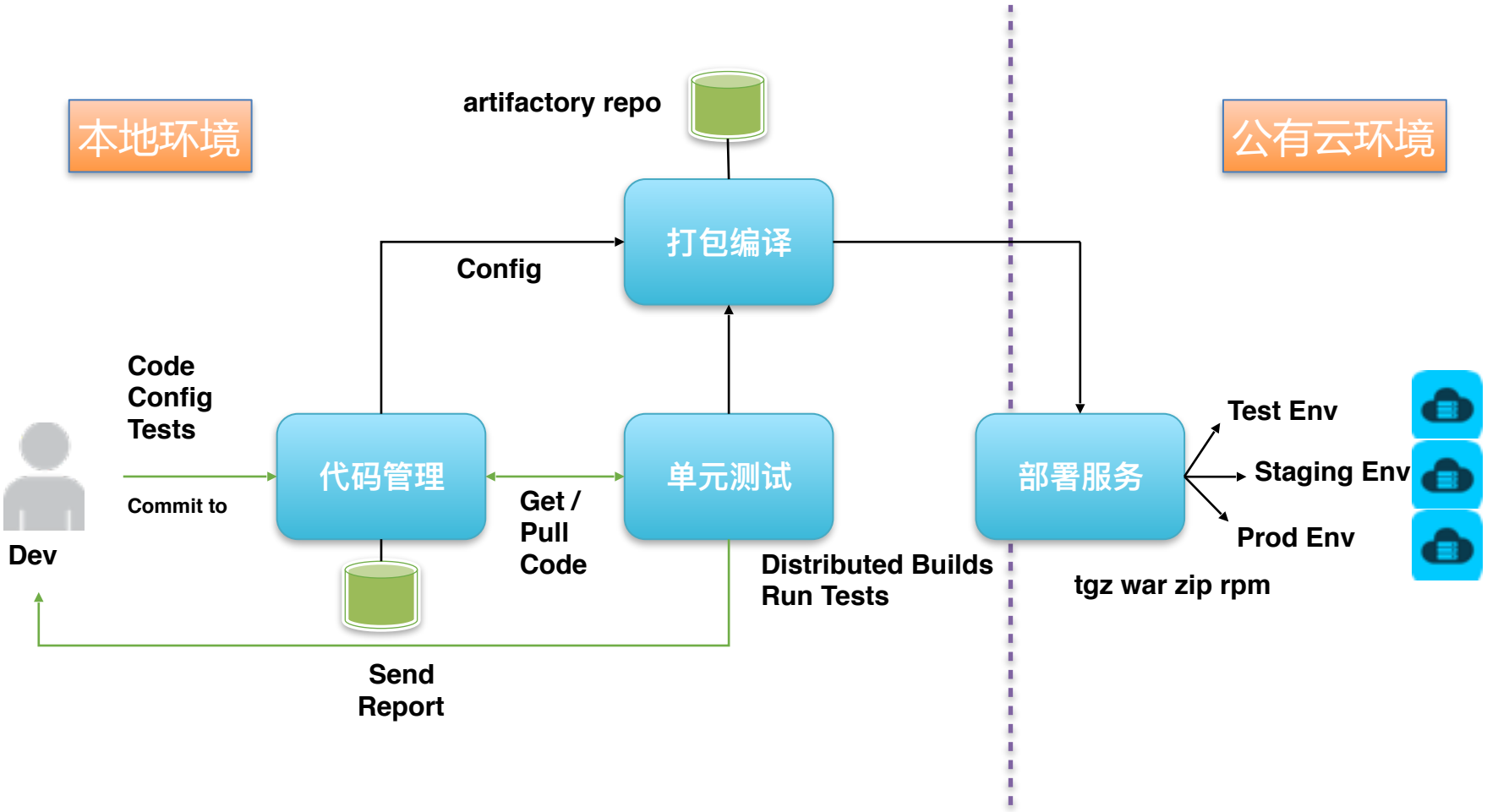


- **在一起**就是集成，每次集成都应该有**反馈**。
- 只有不停的集成才是持续集成。越少持续，每次反馈**代价越大**。
- **多次集成**产生一次交付。

运行CD的Pipeline



构建出CD过程所需的环境



环境 + 存储 + 网络

CD过程中遇到的问题

编译环境维护困难

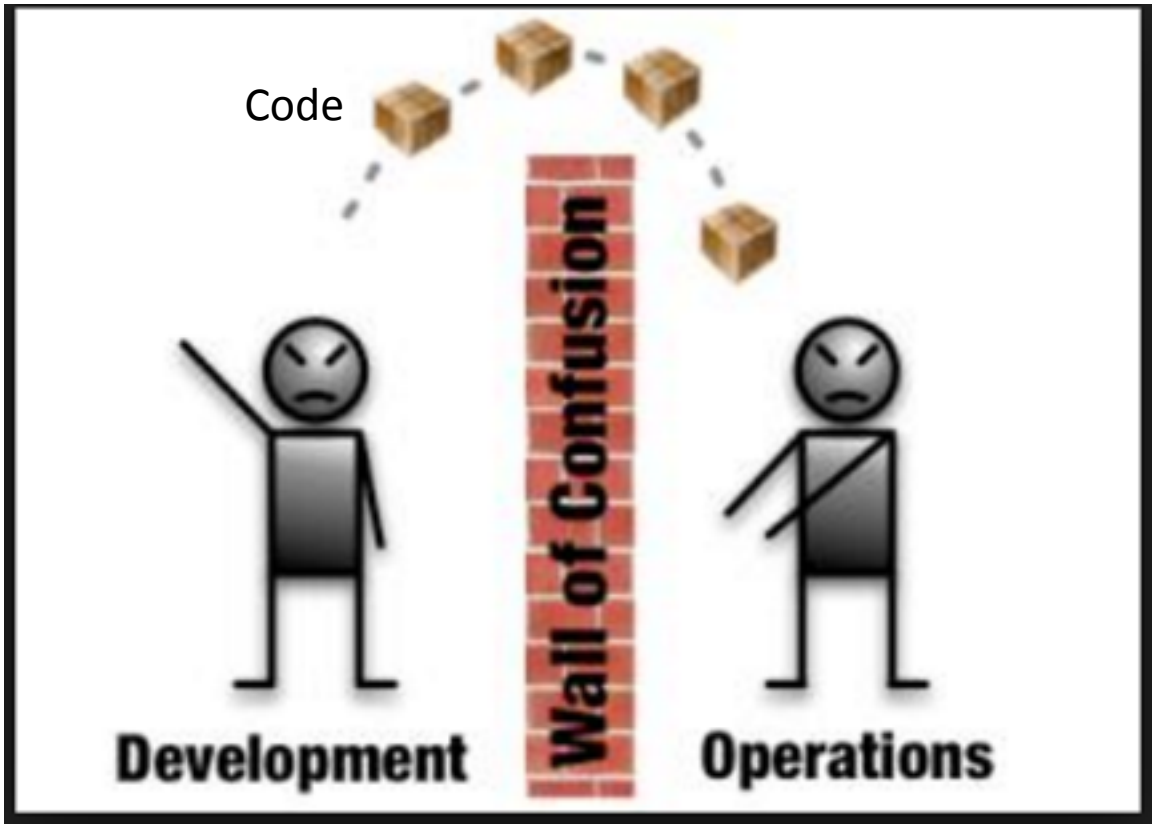
依赖环境维护困难

切换环境调试困难

运行包的版本维护困难

统一环境标准，环境回溯，难上加难

问题的根源：交付的只有代码



- Developer 交付的只有Code ，以及Code的依赖
- 而Keep Site Running 需要除了Code之外的运行环境，以及运行环境之间的依赖

变革软件交付方式的技术： Docker

交付方式变革改变了全球经济格局

“没有集装箱，不可能有全球化。” ——《经济学家》

变革软件交付方式的技术：Docker

An open platform for distributed applications for developers and sysadmins

如果我们能轻松交付整个软件运行栈

coding

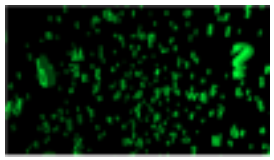
Code
… src/main
… src/test
… pom.xml
… readme.md
… .gitignore
… 一个描述文件

building

environment define
… Ubuntu 14.04
… CentOS 7
environment dependencies
… tomcat
… Jre
environment describe
… PATH
… configs
package.war
… src.jar
… dependencies.jars

running

running Images
… container
… container



Docker提供的能力：

Build Once， Run EveryWhere

1 描述环境的能力

提供了描述运行栈，并且自定义Build 过程的能力。

Code 中的描述文件就是Dockerfile

2 分层文件系统

Image可以像Git一样进行管理，并且每一层都是只读的，对环境的每个操作都会被记录，并且可回溯

3 Docker Registry

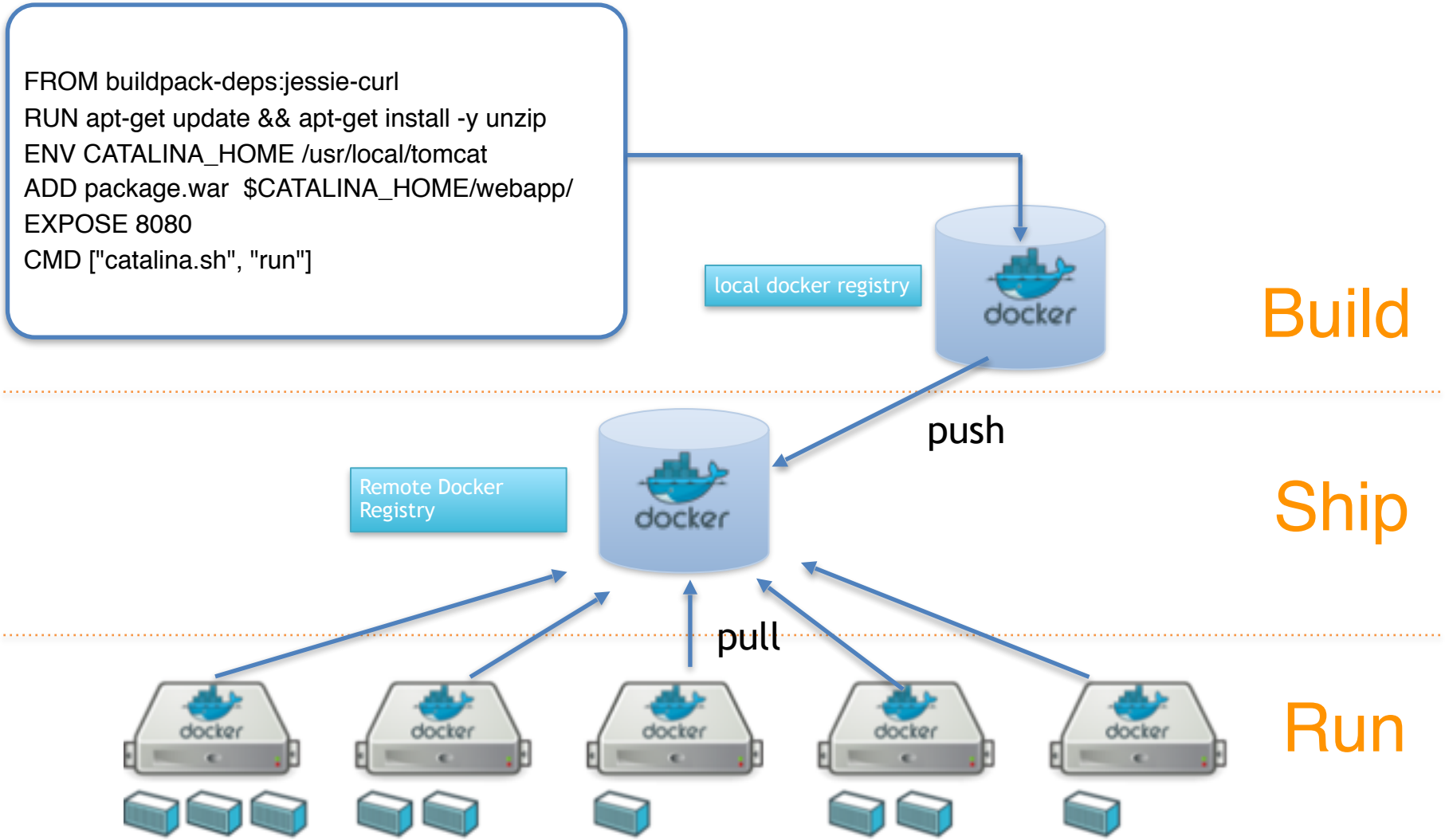
提供了管理Image 存储系统，可以存储，传递，并且对Image进行版本管理

4 屏蔽Host OS差异

解决了环境差异，保证在任何环境下的运行都是一致的（只要满足运行docker的linux 内核）

Docker 传递的过程

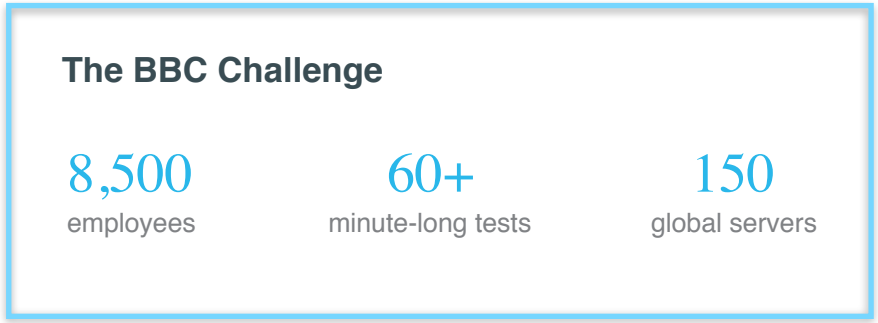
```
FROM buildpack-deps:jessie-curl
RUN apt-get update && apt-get install -y unzip
ENV CATALINA_HOME /usr/local/tomcat
ADD package.war  $CATALINA_HOME/webapp/
EXPOSE 8080
CMD ["catalina.sh", "run"]
```



应用Docker化交付的过程

一个官方案例： BBC News

Before：
- 10种CI环境，26000 Jobs，500Dev
- 任务需要等待，无法并行



After：

- **Slash job time from 1 hour to 10 minutes:**
  不需要等待定时任务，并行任务提升60%速度
- **Language Flexibility:**
  自定义不同编译语言的编译Job
- **Eliminate Multi-Day Sideloading Process:**
  安全的开放CI环境
- **Empower Developers:**
  让开发能够根据自己的应用架构，采用合适的语言版本。
- **Standardization:**
  开放，可复制，可定制，可扩展，高可用，最佳实践

Step 1: 安装Docker 运行环境

Docker Machine

- 配置安装
  - 安装Docker Toolkit
  - 安装云驱动
    - ECS driver for Docker Machine
    - AWS, GCE, etc.
- 创建Docker运行环境

```
docker-machine create \
    —driver aliyunecs mytest \
    eval "$(docker-machine env mytest)"
docker run -d nginx
```

Step 2: 用Dockerfile 描述你的应用环境

举例：Java 运行环境包含什么?

- linux某操作系统
- 基础软件
- openjdk 7 && 配置 Java Home 等环境变量
- Tomcat 7 && 配置 环境变量
- 应用包 target.war
- 应用包 启动参数 JVM
- Web Server 指定端口 8080
- 启动tomcat

```
FROM buildpack-deps:jessie-curl
RUN apt-get update && apt-get install -y unzip  \
        openjdk-7-jre-headless="$JAVA_DEBIAN_VERSION" \
       &&  rm -rf /var/lib/apt/lists/*
ENV LANG C.UTF-8

ENV JAVA_VERSION 7u91
ENV JAVA_DEBIAN_VERSION 7u91-2.6.3-1~deb8u1

ENV CATALINA_HOME /usr/local/tomcat
ENV PATH $CATALINA_HOME/bin:$PATH
RUN mkdir -p "$CATALINA_HOME"
WORKDIR $CATALINA_HOMEENV TOMCAT_VERSION 7.0.68
ENV TOMCAT_TGZ_URL  \
      https://xxxx/apache-tomcat-$TOMCAT_VERSION.tar.gz

RUN set -x \
          && curl -fSL "$TOMCAT_TGZ_URL" -o tomcat.tar.gz \
          && curl -fSL "$TOMCAT_TGZ_URL.asc" -o tomcat.tar.gz.asc \
          && gpg --batch --verify tomcat.tar.gz.asc tomcat.tar.gz \
          && tar -xvf tomcat.tar.gz --strip-components=1 \
          && rm bin/*.bat \
          && rm tomcat.tar.gz*

EXPOSE 8080
CMD ["catalina.sh", "run"]
```

Think ?

- 如果有一个安装好Java的环境 ？
- 如果有一个安装好Java和Tomcat的环境 ？
- 如果是微服务，对环境只依赖Java/Node基础环境，是不是所有应用都可以共用1个环境?

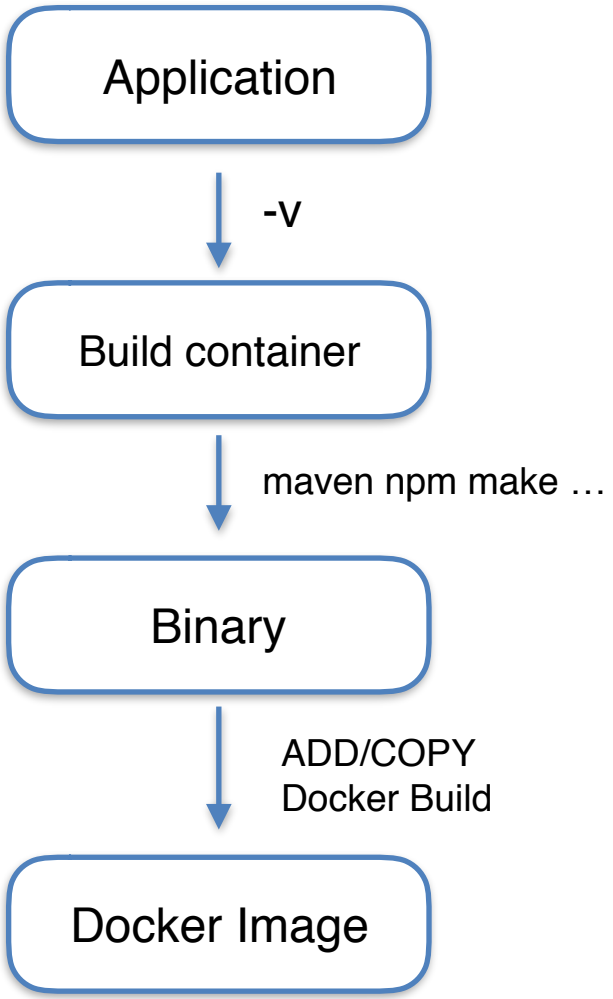
Step 3: 用Docker 描述你的编译/UT环境

用docker作为编译/UT环境的好处：

- 自定义，可扩展，可复制的编译环境。

Point：

- build app 和 build docker image 分开进行
  - 分层概念导致源码泄露的可能
  - 镜像最小化原则
- Docker file 不要放到代码根目录下
  - 避免大量文件传给docker deamon
- 通过运行容器直接运行测试脚本



Step 4: 用Docker-Compose 描述依赖环境

Write your dockerfile

```
WORKDIR /code
ADD requirements.txt
/code/
RUN pip install -r
requirements.txt
ADD . /code
CMD python app.py
```

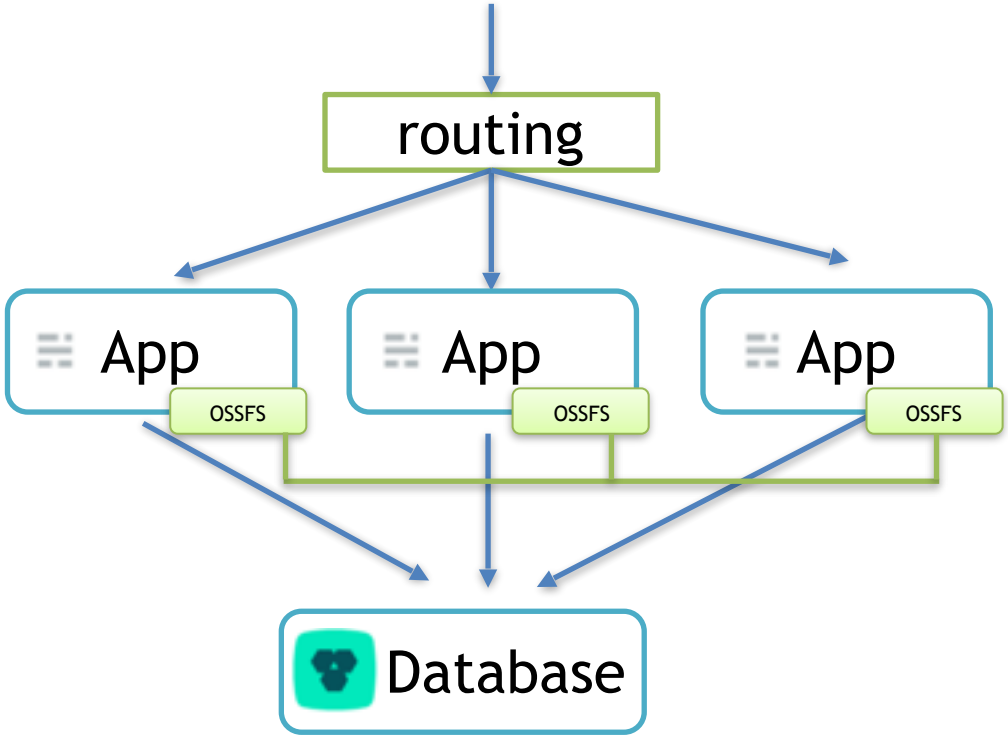
Write your compose.yml file

```
web:
  build: .
  links:
  - db
  ports:
  - "8000:8000"
db:
  image: postgres
```

Run your app

```
$ docker-compose up
```

Step 4: 用Docker 描述集成/运行环境



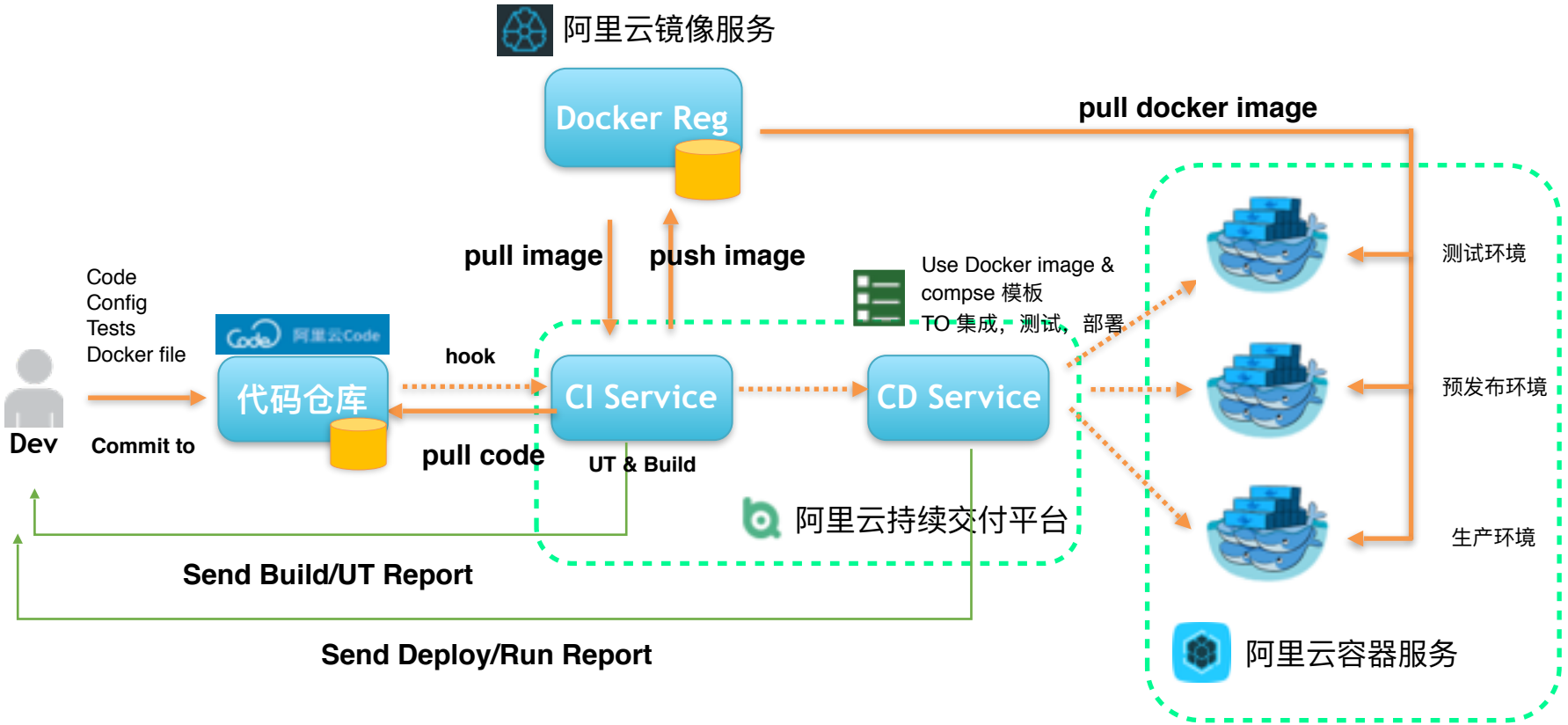
声明观念：

- I need 负载均衡（haproxy，Nginx）
- I need 数据库（mysql)
- I need 文件存储(通过-v，ossfs)
- I need 缓存服务(redis,kv-store)
- …

Alibaba Group 阿里巴巴集团

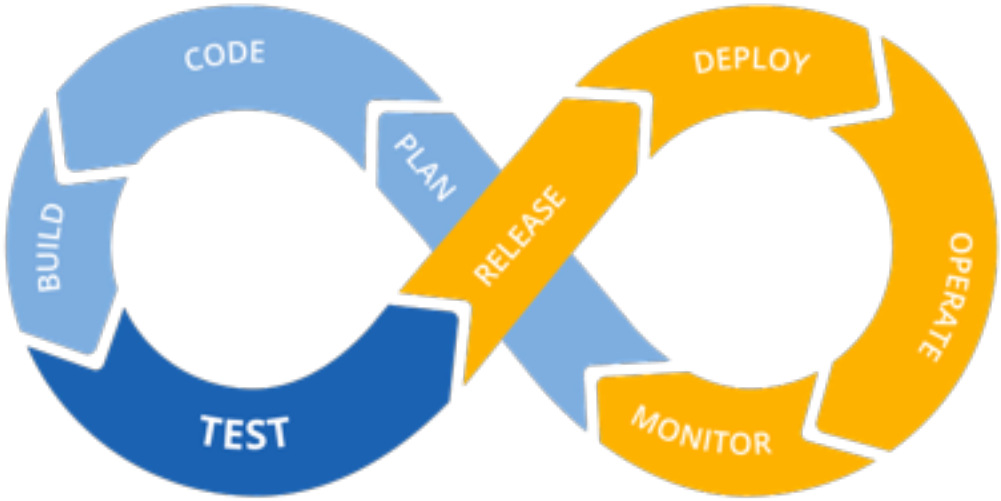
完整的容器持续交付流程

Put it All together



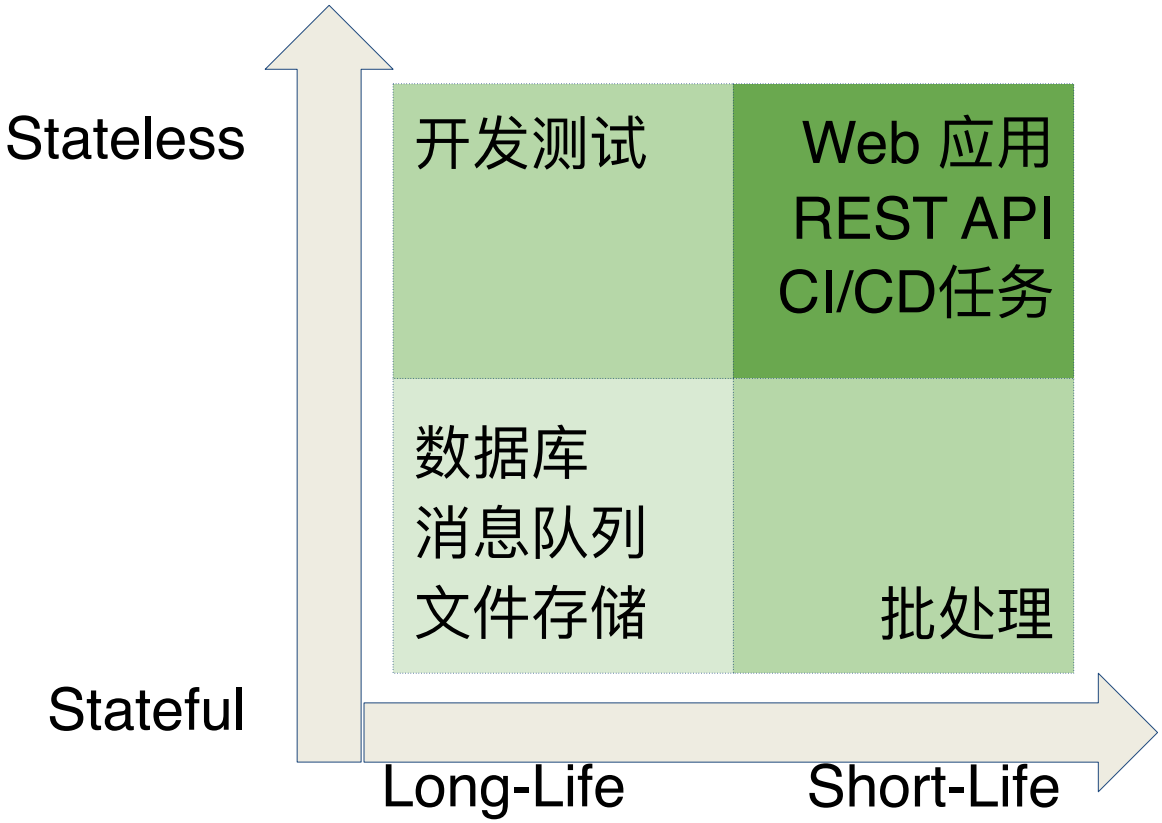
Docker化交付带来的

- 研发： 更清楚，更灵活的掌控自己的软件运行环境。
- 运维： 再也不用为应用软件依赖栈的变更碎片化自己的时间。
- 资源：每人每项目每环境。
- **DevOps 的最好诠释: 将编程的思想应用到运维领域**



最后：我们该把什么场景Docker化？

Let’s Think out together ……



Thanks!