高磊
雪球（北京）技术开发有限公司
SRE团队高级工程师

雪球SRE团队高级工程师。2011年硕士毕业于北京邮电大学，从事缓存、数据等分布式系统的研发工作，现在在雪球负责底层基础设施的开发和运维，致力于探索Docker等基础设施在创业公司的最佳实践。

QClub Docker专场：聚焦国内Docker创业与企业实践

Docker 在雪球的技术实践

高磊
雪球SRE团队

1. 雪球对 Docker 的定位

2. 雪球对 Docker 的使用方式

3. 在使用 Docker 中遇到的问题和解决方案

4. 雪球对 Docker 的未来展望

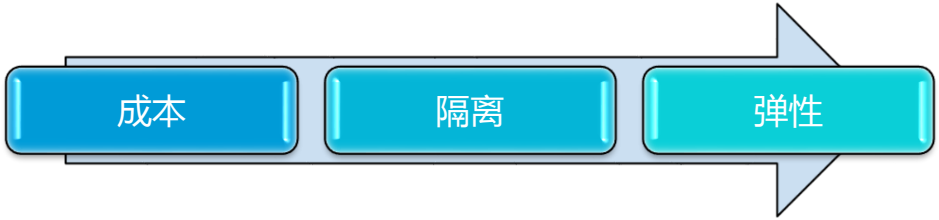
Part 1

雪球对Docker的定位

关于雪球

雪球对 Docker 的定位

- 雪球为什么选择虚拟化或容器技术



- 雪球为什么选择 Docker
  - 小、快、轻
  - 一次构建，到处运行

- 雪球对 Docker 的使用需求
  - 容器特性：轻量、只读
  - VM特性：交互

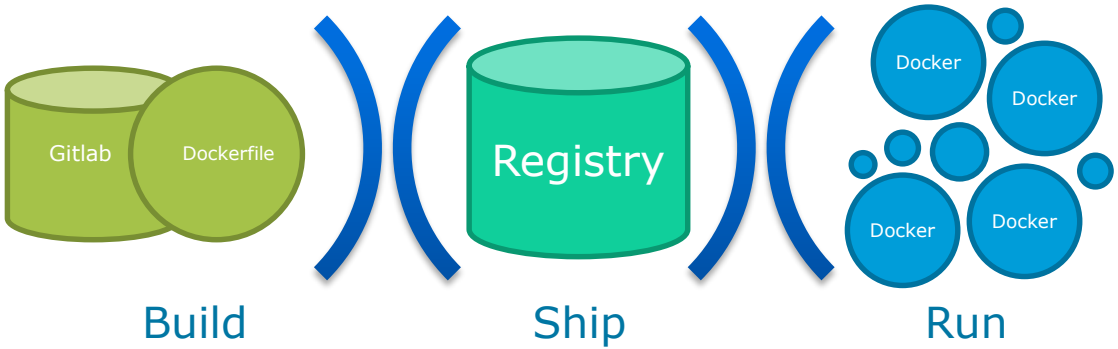
Part 2

雪球对Docker的使用方式

雪球对 Docker 的使用方式

- Linux 发行版
  - Ubuntu 14.04
- 服务类型
  - 有状态：LXC
  - 无状态：Docker
- Docker Version
  - 0.6.4
  - 1.2.0
  - 1.5.0

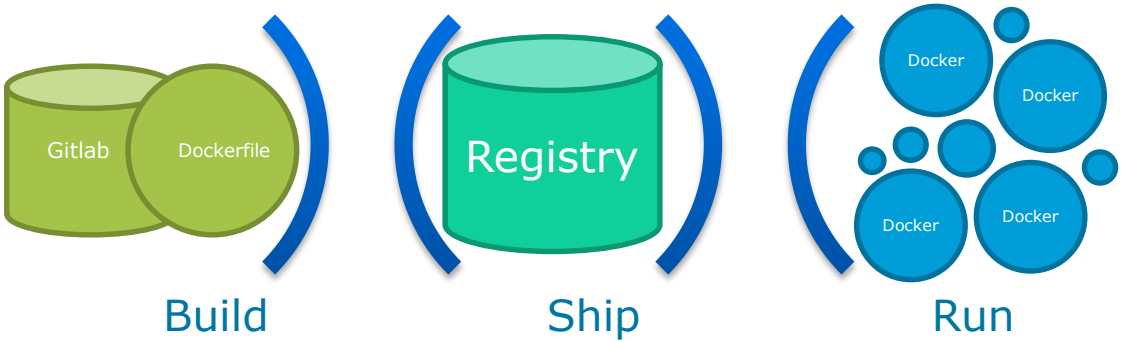
- Docker的构建、分发、状态维护



Part 3

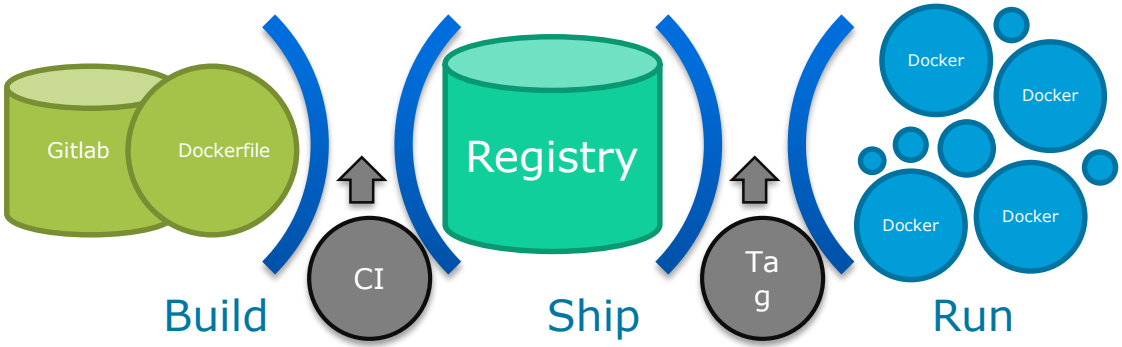
遇到的问题和解决方案

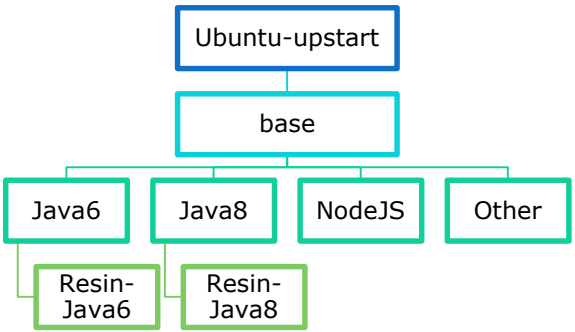
遇到的问题和解决方案



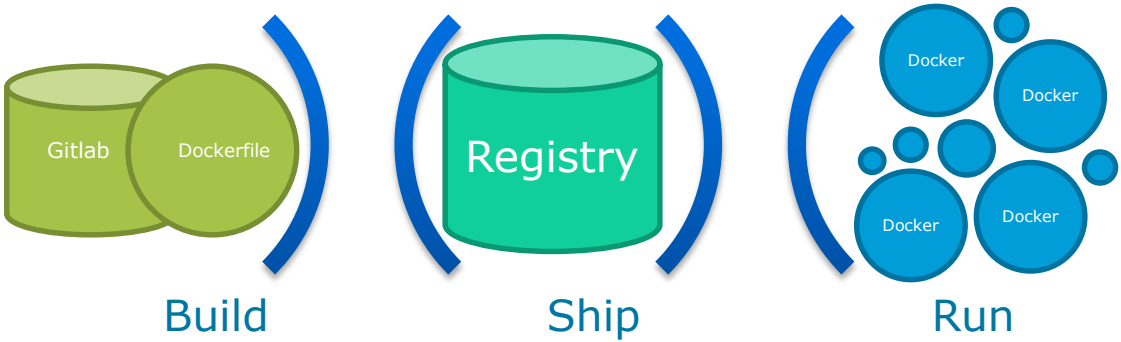
- 构建
  - 构建文件优雅统一
  - 临时容器
- 分发
  - CI
  - 部署
  - 镜像
- 运行
  - Daemon
  - Container
  - Monitor

分发：CI、Tag与镜像


Gitlab
Dockerfile
CI
Registry
Ta
g
Docker
Docker
Docker
Docker
Build
Ship
Run


Ubuntu-upstart
base
Java6
Java8
NodeJS
Other
Resin-Java6
Resin-Java8

构建



- 构建
  - 构建文件优雅统一
  - 临时容器
- 分发
  - CI
  - 部署
  - 镜像
- 运行
  - Daemon
  - Container
  - Monitor

构建：Dockerfile

```
FROM ubuntu:14.04

# Install required packages
RUN apt-get update -q \
    && DEBIAN_FRONTEND=noninteractive
      ca-certificates \
      openssh-server \
      wget

# Download & Install GitLab
# If the Omnibus package version below
# If you run GitLab Enterprise Edition
RUN TMP_FILE=$(mktemp); \
    wget -q -O $TMP_FILE https://downl
    && dpkg -i $TMP_FILE \
    && rm -f $TMP_FILE

# Manage SSHD through runit
RUN mkdir -p /opt/gitlab/sv/sshd/super
    && mkfifo /opt/gitlab/sv/sshd/supe
    && printf "#!/bin/sh\nexec 2>&1\nu
    && chmod a+x /opt/gitlab/sv/sshd/r
    && ln -s /opt/gitlab/sv/sshd /opt/
    && mkdir -p /var/run/sshd

# Expose web & ssh
EXPOSE 80 22

# Declare volumes
VOLUME ["/var/opt/gitlab", "/var/log/g

# Copy assets
COPY assets/gitlab.rb /etc/gitlab/
COPY assets/wrapper /usr/local/bin/

# Wrapper to handle signal, trigger ru
CMD ["/usr/local/bin/wrapper"]
103.245.222.133
```

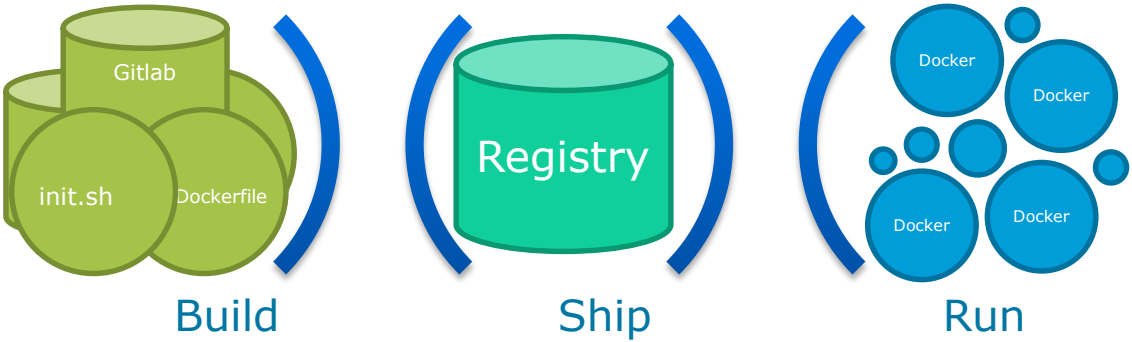
Gitlab 的 Dockerfile

```
FROM docker.snowballfinance.com:5000/ubuntu-upstart:14.04
MAINTAINER gaolei@xueqiu.com

ADD dockerinit/base /data/tools/dockerinit/base
ADD dockerinit/base.sh /data/tools/dockerinit/base.sh
ADD dockerinit/init.sh /data/tools/dockerinit/init.sh

RUN /data/tools/dockerinit/base.sh
```

```
base
├── apt
│   ├── sources.list
│   └── trusted.gpg
├── kerberos
│   ├── app.k5login
│   ├── krb5.conf
│   ├── op.k5login
│   └── root.k5login
├── limits
│   └── limits.conf
├── ntp
│   └── ntp.conf
├── resolv
│   └── resolv.conf
├── salt
│   └── minion
├── ssh
│   ├── authorized_keys.app
│   ├── ssh_config
│   └── sshd_config
├── sysctl
│   └── sysctl.d
│       └── 10-ptrace.conf
├── timezone
│   └── timezone
└── zabbix
    ├── scripts
    │   ├── listen_port_discovery.sh
    │   └── zabbix_low_discovery.sh
    ├── sudoer_zabbix
    ├── zabbix_agentd.conf
    └── zabbix_agentd.d
```

雪球的 Dockerfile 与 base.sh

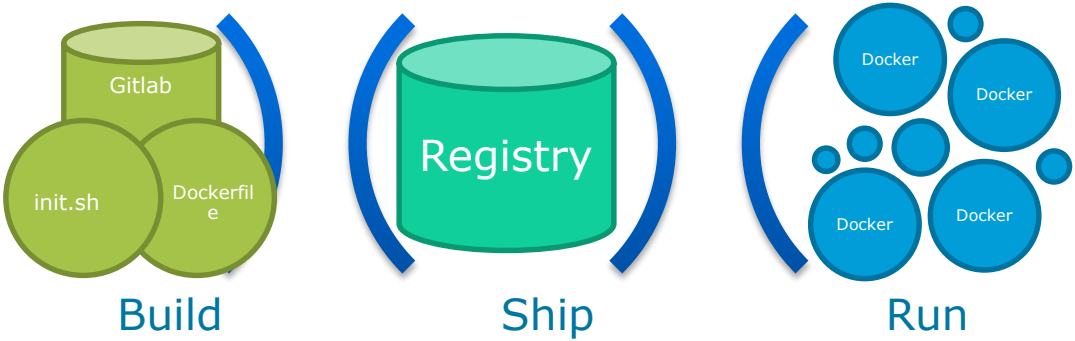
构建——init.sh


Gitlab
init.sh
Dockerfile
Registry
Docker
Docker
Docker
Docker
Build
Ship
Run

构建：临时容器

- 问题
  - Docker build 时无法访问外网
  - Docker build 时无法安装 Cron 等包
- 原因分析
  - 临时容器
  - Default Gateway
  - No Support for upstart
- 解决方案
  - 宿主机打开 ip_forward=1 转发
  - 传入 --ip 和 --gateway 参数
  - 传入 --entrypoint 并支持 upstart

运行



- 构建
  - 构建文件优雅统一
  - 临时容器
- 分发
  - CI
  - 部署
  - 镜像
- 运行
  - Daemon
  - Container
  - Monitor

运行：Daemon

- 网络模式
  - 桥接模式
  - 清理 docker0，桥接至 br0

```
1  ip link set dev docker0 down
2  brctl delbr docker0
3  docker --bridge=br0
```

- IPForward，IPTables
  - 问题：毛刺流量，服务丢包
  - 分析：
    - Daemon 默认暴力开启 ipforward 并添加 iptables 规则
    - TCP Retransmission
    - nf_contract 不稳定
    - 网卡混杂模式
  - 解决方案：
    - 禁用 ipforward / iptables
    - 清理 x_*、xt_*、nf_*、iptables_* 模块并列入黑名单

运行：Container

- 固定 IP
  - 必要性：监控报警关联、可运维
  - 解决方案：
    - Docker run 传入 --ip 参数
    - 运行时修改 eth0 网卡 IP

- 固定 MAC 地址
  - 问题：ARP 缓存，网络丢包
  - 解决方案：

```
IP="XX.XX.XX.XX"
MAC_TMP=`echo "$IP" | awk -F'.' '{print "0x02 0x42 "$1" "$2" "$3" "$4}'`
MAC=`printf "%.2x:%.2x:%.2x:%.2x:%.2x:%.2x" $MAC_TMP`
```

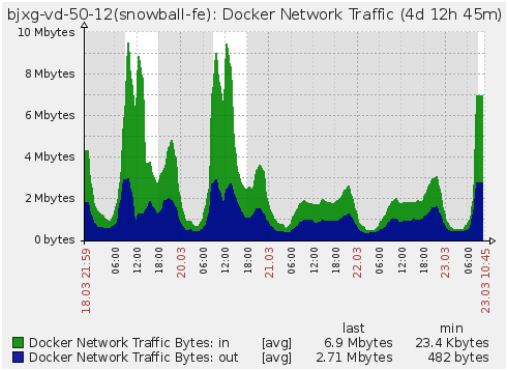
- IPv6 DNS 搜索域
  - 问题：域名解析超时
  - 问题分析：
    - IPv4 查询压力大时，IPv6 查询会以 Hostname 后缀为搜索域
  - 解决方案：
    - Daemon 传入 DNS 搜索域为 “.” ——失败
    - Container 传入 DNS 搜索域为 “.”——失败
    - Container 的 resolv.conf 加入搜索域 “.”

运行：Container

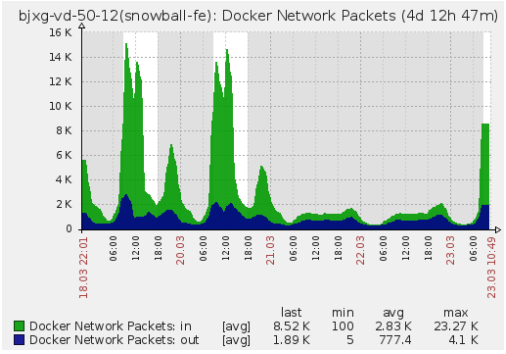
- Upstart
  - 问题
    - sshd、zabbix-agentd、cron、logrotate 的后台启动支持
  - 分析：
    - /sbin/init 要存在，且可执行
    - /sbin/init 要以 PID=1 启动
  - 解决方案：
    - 使用带 upstart 的镜像
    - Docker run 的 entrypoint 参数仅包含 “/sbin/init”
    - Docker run 不添加任何 CMD 参数
    - 自启动程序放入 /etc/rc.local

运行：Monitor

- 监控 Memory
  - 问题：
    - /sys/fs/cgroup/memory/docker/<container_id>/memory.usage_in_bytes 不准确
  - 解决方案：
    - Docker run 传入 --memory 参数



- 监控 Block IO
  - 问题：
    - Docker stats API 读到的信息不易解读
  - 分析：
    - 宿主机分区不一致
    - major:minor <-> /dev 设备 <-> 分区
  - 解决方案：
    - 统一宿主机硬盘和分区
    - 统一宿主机 Docker Graph 目录
    - 对所有磁盘的 Block IO信息汇总



Part 4

雪球对 Docker 的未来展望

雪球对 Docker 的未来展望

- 从 Container 的角度
  - 填坑，特别是网络方面
  - 对 Cgroups 隔离的更加细致使用
- 从 Host 的角度
  - 统一的部署系统
  - Remote API over HTTPs
  - 弹性扩容缩容

QClub Docker专场：聚焦国内Docker创业与企业实践

Thank you!

雪球

高磊
微信： lostleon
邮箱：gaolei@xueqiu.com