Docker@Alibaba

——超大规模Docker化的实战经验

毕玄
阿里巴巴研究员

目 录
content

扫码观看大会视频

一、Docker化前的阿里

1

虚拟化

ECS
T4：基于LXC的“VM”

2

标准化

内部自定义的基线

3

部署模式

多种多样，甚至不同语言不同部署系统

碰到的问题

- 标准化不好推进，不同业务模式接受度不同，标准化推进不力导致各种系统都很难做；
- 需要更强制的DevOPS模式；
- 多种部署模式导致的重复建设。

二、Docker化的目标

1

虚拟化

将Docker改造成更像“VM”。

2

标准化

有效推进DevOPS；
各种语言系统都走同样的部署链路。

3

双11核心买家链路全部Docker化

核心应用全部Docker化，快速推进。

扫码观看大会视频

Docker和DevOPS、标准化

- 更强制的DevOPS模式
  - dockerfile描述了整个运行环境；
- 更有效的推进标准化
  - 部署是销毁用镜像重新拉起；
    - 意味着目录的规范性非常重要。

三、Docker化碰到的问题

1

Swarm

规模化带来的挑战
稳定性

2

Docker Engine

更像“VM”
各种bug fix和功能增强

3

Docker生态

从编译到部署

扫码观看大会视频

Swarm

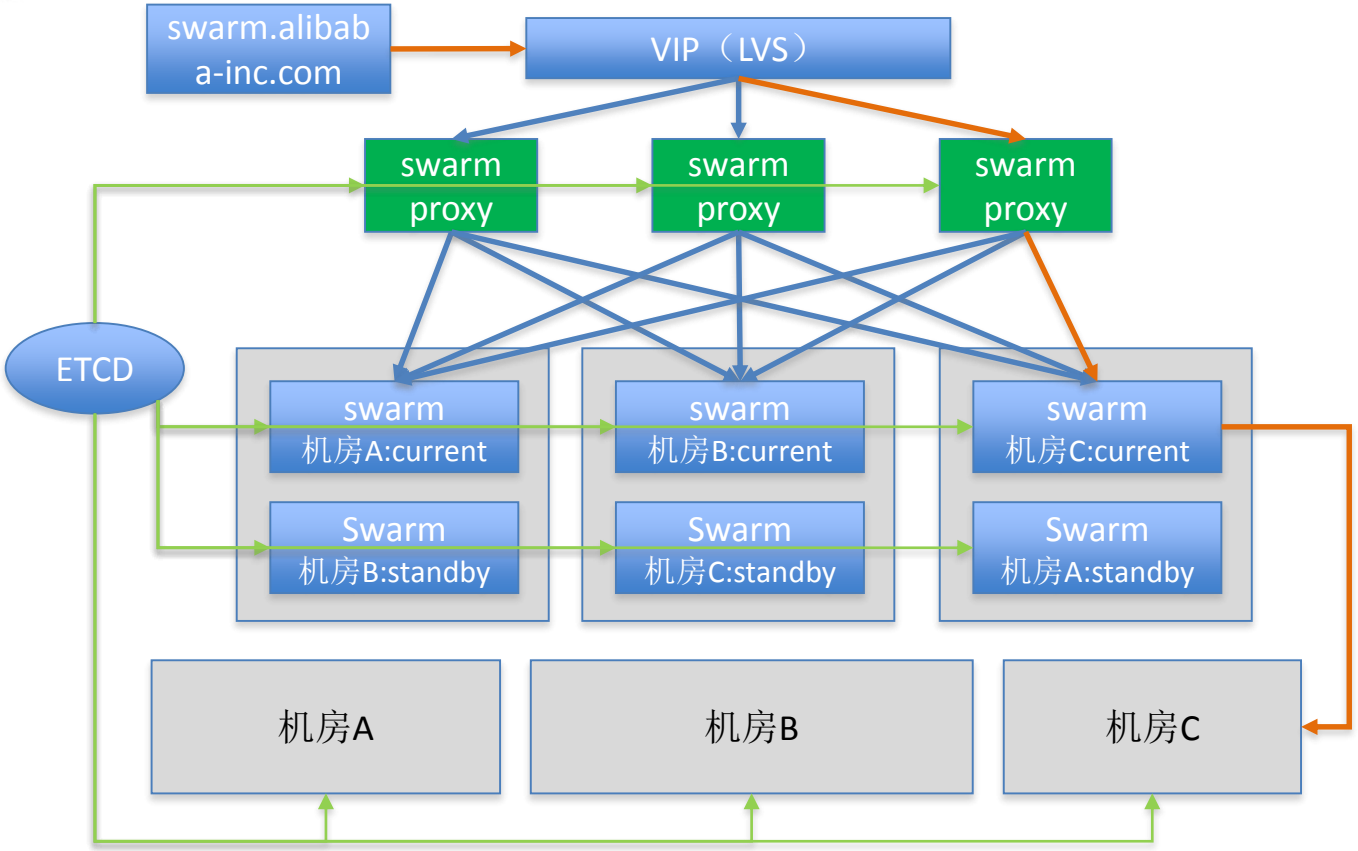
- 规模化的挑战
  - 官方
    - Swarm is production ready and tested to scale up to one thousand (1,000) nodes and fifty thousand (50,000) containers.
  - 我们改造后
    - 单Swarm实例health nodes稳定在2W+;

Swarm

- 规模化的挑战
  - 优化连接管理
    - 保证同一个engine各种极端情况下同一时间只建立一个连接
    - 避免一个连接起一个系统线程，从而解决了连接暴增后系统crash的问题
    - 优化后单实例管理2W+个node只需要40个系统线程
  - 优化锁管理
    - 严格控制锁的使用, 最小化锁粒度，解决了极端情况下swarm实例夯住的问题
  - 优化大批量node的加入删除
    - 修改diff算法，减少node刷新时的cpu开销
    - 避免实例重启时大批量node并行建立连接对系统的冲击

Swarm

- HA--自己实现了一套swarm-proxy
  - 证书和多实例多集群管理
    - 不同证书路由到不同集群的swarm实例，没有独立集群的证书路由到公共集群
    - 公共集群上不同证书创建的实例互相不可见（修改swarm实现做了过滤）
  - 单实例热备管理—故障自动切换，平滑升级回滚
    - 见下页图


swarm.alibaba-inc.com
VIP（LVS）
swarm proxy
swarm proxy
swarm proxy
ETCD
swarm 机房A:current
swarm 机房B:current
swarm 机房C:current
Swarm 机房B:standby
Swarm 机房C:standby
Swarm 机房A:standby
机房A
机房B
机房C

Docker Engine

- 更像“VM”的AliDocker
  - 延续在T4的经验，完善namespace；
    - top/load/free/dirquota/df/netstat等；
  - 很多改造都可在这里获取
    - https://github.com/alibaba/ali_kernel

Docker Engine

- 各种bug fix和功能增强
  - Daemon升级或crash后，所有容器被自动销毁的问题
  - cpuset、cpuacct和cpu子系统mount到一起时cgroup操作错误的bug
  - 支持基于目录的磁盘配额功能（依赖内核patch）
  - 支持指定IP启动容器，支持通过DHCP获取IP
  - 支持启动容器前后执行特定脚本
  - 支持镜像下载接入各种链式分发和内部mirror的机制
  - 增加docker build时的各种参数优化效率和适应内部运维环境
  - 优化Engine和registry的交互

Docker生态

- 从编译到部署，在没做优化前，整个速度比以前慢了大概两倍多，最差情况甚至慢了有四五倍；
    - 编译打包优化
        - cache— 三级cache：mirror、超级节点、单机
        - 应用镜像分层

Docker生态

- 从编译到部署，在没做优化前，整个速度比以前慢了大概两倍多；
  - 镜像分发优化
    - 基于地域部署，基于机房路由的mirror
    - 每一个image layer的BT模式下载加超级节点cache

Docker生态

- 从编译到部署，在没做优化前，整个速度比以前慢了大概两倍多；
  - 发布流程优化
    - 流式分发
    - 镜像分批预热

Docker生态

- 多种部署模式需求
  - Docker自身的部署机制
  - 增加了hotfix部署机制
    - Velocity模板类型
    - 本地cache

四、未来

1

全面AliDocker化

直接跑在物理机上的应用也迁移到AliDocker。

2

Docker生态体系输出到阿里云

把阿里大规模Docker实践的经验体系化、工具化的输出到阿里云，为云上Docker用户谋福利。

2016 The Computing Conference

THANKS