

1、 演讲主题；

中通物流的上云实践

2、 演讲主题简介；

简介：

本次分享中通物流演进历程。主要讲述我们通过机器资源池化集约管理，提高资源利用率、简化环境部署过程；融入云原生：资源配额、缩容扩容、滚动更新、弹性伸缩等；通过容器化部署提升机器应用部署密度 3 倍。

通过应用容器化来解决我们现阶段申请资源麻烦，环境初始化复杂，资源利用率低僵尸机多，重点会分享我们在容器化过程中遇到过哪些坑及解决方案，如加了应用资源画像优化调度，根据资源实际使用量优化调度，自研多集群纳管，和公司内发布流程打通，以及我们应用上容器中遇到的一些问题。

分享大纲：

1、 背景介绍（现有背景，刚开始容器化 从虚拟机到混布到容器化）

1，生产无状态服务虚机部署，资源申请响应速度慢，多一层硬件虚机化资源开销，资源利用率低，磁盘消耗大，逐渐累积了不少“僵尸机器”。

2，环境现状•扩容速度慢尤其是应急流量扩容滞后，没有自动扩缩容能力，至少需要十几分钟，且不能及时回收资源。中心随着业务大规模增长速度下，现有虚机资源响方式难以为继。

3，运维成本高，框架有心跳检测机制模块，但是运营上没有持续的心跳检测以及解决机制

2、 容器化过程中我们做了什么？遇到了什么问题？

1，配置问题

1.1: 用户配置一个 ingress 域名使集群内 ingress-nginx-controller 服务挂掉,导致集群内所有域名无法访问。

1.2: kubernetes 原生的 HPA 的 cpu 设置是 request 的百分比，用户设置了一个值之后再

去修改 request 或 limit，导致应用不自动扩缩容问题。设置 HPA 需要慎重

2，外网权限

2.1: Calico 网络策略排查。白名单制，排查每个 namespace 下的应用访问外网的情况。防止出现已发布应用无法访问外网问题

3，cpu 上限

3.1 应用在容器内 dubbo 服务 延迟比虚拟机高了 4 倍，（当时虚拟机 30ms 左右。容器 120ms 左右）还报有获取数据库连接错误，导致数据挤压，通过监控观察资源使用情况以及应用的 yaml 文件发现，deployment limit 设置为 4c，cpu 实际使用量超过了 4c，怀疑是 cpu 跑到了上限导致了。随即又把 cpu 调整到 8c。接口延迟 以及 cpu 使用情况都得到了有效的解决。

4，单一问题

4.1: 慢 SQL 问题导致的创建数据库连接超时和 dubbo 线程池耗尽

4.2：消息队列消费不均匀，导致局部流量压在某个副本上，水平扩容无效

4.3: dubbo 线程池运行一段时间会耗尽，应用日志中也有 dubbo 耗尽报错 “Thread pool is EXHAUSTED”

3、成果展示

1，核心产品线流量部分或者全部接入容器，整体流量占比 44%；

2，大大节省物理机资源，密度提高 3 倍；预计无状态全部容器化可节约 400 台物理机，12000 台虚机；

3，新增资源无需走申请流程，立建可得秒级相应，平均每一个实例节约 0.5 个资源准备时间；

4，通过机器资源池化集约管理，提高资源利用率、简化环境部署过程；融入云原生：资源配额、缩容扩容、滚动更新、弹性伸缩等；

5，双 11 各产品线无需热备大量虚机资源。

4、未来规划（粘性 ip , 容器画像 自定义调度 容器存储）