K8s 在到喜啦的实践之路

基于Shell，Jenkins + Docker + kubernetes的持续集成与持续部署

汝林 到喜啦资深DevOps工程师

上海到喜啦信息技术有限公司

让每一个幸福时刻都有到喜啦

到喜啦正式成立于2010年5月，是国内首家垂直婚宴、喜宴预订电子商务平台，用O2O模式为用户提供省时、优惠、有保障的婚宴预订及结婚周边服务。

我们拥有中国最大最专业的结婚行业垂直网站，到喜啦结婚网(www.daoxila.com)和移动端领先的结婚应用，到喜啦结婚App、到喜啦 Web等，目前已成为国内结婚一站式服务平台的领导者。

目录

现状与需求

- 命令行管理的KVM 虚拟机，部署及迁移维护
- 更充分、更合理的充分利用资源
- 活动时突发流量下快速扩容
- 生产环境上线，需要更透明可控的流程及工具
- 测试环境部署，需要更少的人工参与操作
- 项目语言环境与版本复杂繁多，依赖性问题
- 多环境，开发、测试、预发布、生产，环境与权限控制

环境与版本

- 主机型号： HP GL365(8C/16G) 与 DELL R610(24C/32G) (15 台)
- 主机系统： CentOS 7
- 主机网络： 全千兆
- 存储系统： Ceph Gluster (DELL R610 * 3, 73G SAS * 2 RAID1 OS, 512GSSD*1 Journal, 600G SAS * 3 OSD)
- Registry： Dell R210 (8C/16G, 500G SATA * 2 RAID1)
- Kubernetes: 1.8.1
- 集群网络： Flannel

服务类型


node
JS
php
Java
python
MySQL
mongoDB
redis
MEMCACHED
Tokyo
Cabinet
8192PiB
hadoop
Stack
Apache
ACTIVEMQ
Solr
Grafana
Sonatype
Nexus
Apache
HTTP SERVER
Apache
Tomcat
NGINX

从哪里入手？

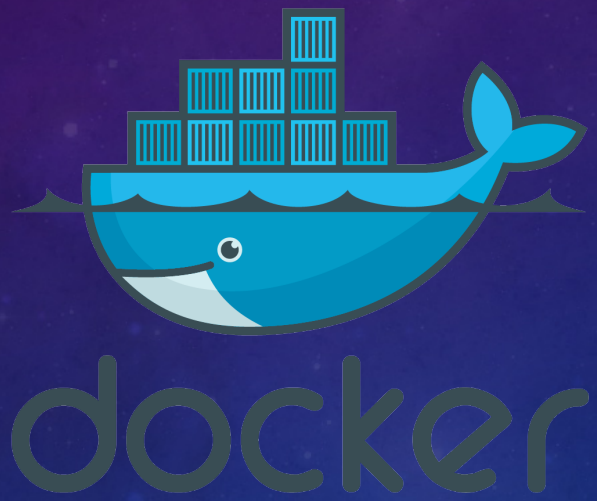
- 结合Jenkins做持续集成和部署
- 项目编译、镜像构建及推送、K8s 更新流程

- 看似很简单的PHP项目(> 40)
- 非常友好的Node JS项目(> 15)
- 数量庞大且调用关系错综复杂的支付(Java)项目(> 70)

如何迁移

- 根据业务特性和实际情况，制定合适方案
- 确保基本集群环境就绪，如存储服务、API server高可用等
- 同步运行，容器内项目与虚机上的一起通过Nginx upstream分配，逐渐加大容器比重，平滑过渡，在容器异常时可以随时回退至虚拟机运行环境
- 经过一段时间平稳的同步运行后，虚机逐步退出运行(备份虚机镜像)，腾出物理机加入K8s集群
- 扩容后的集群继续迁入更多服务

持续集成



Jenkins - 高效、可靠、一教就会(配置)不易出错的运维好伙伴

1. 拉代码
2. 预处理
3. 编译
4. 从模版生成 Dockerfile, 添加任务信息到 label 中，如构建者、代码版本、任务名称等等，以便日后追溯
5. 从模版生成 Deployment & Service Yaml 文件
6. Docker build & push，使用代码版本号或Jenkins任务编号作为 docker tag
7. kubectl 操作，存在则更新，否则就创建
8. 更新还是回滚？使用指定版本的镜像进行更新或回滚即可

Docker镜像与Registry

- VMware Harbor，优秀的开源镜像管理系统，支持LDAP认证及镜像漏洞扫描
- 包含漏洞镜像的自动更新

- 第一阶段使用centos作为基础镜像
- 第二阶段使用alpine等更小的镜像为基础，减少存储和网络传输压力，另外也能加快容器创建速度，同时更少的功能意味着相对更少的安全漏洞

多环境运行

- 部分项目需要较长测试周期，因此需要多套测试环境
- 使用预发布环境进行功能(数据)与界面验证
- 按环境分配NAMESPACE
- Nginx监听多个IP，不同IP对应不同环境，相应环境容器启动时自动下载应用hosts文件
- 客户端切换hosts文件即可变更环境进行连接

监控

- 容器是否需要监控？
- 容器内运行的服务是否需要监控？该监控哪些目标以及如何监控？
- Zabbix
- Prometheus
- Heapster
- LivenessProbe & ReadinessProbe

Zabbix ：容器之外的监控

- 硬件：磁盘，RAID，温度，风扇，供电
- 系统：CPU与内存使用
- 网络：网卡流量，网络质量(稳定性)
- 服务：如URL，数据库性能指标
- 集群：重要服务，如 etcd, api-server, controller-manager, scheduler 等
- 其它：服务实例放心的交由k8s健康检查自动管理

容器监控

- 对于k8s 中的容器来说，本来就处于集群的管理和监控中
- 容器中的应用状态，响应速度
- Heapster & Prometheus

Liveness & READINESS

- httpGet, exec, socket

- 可灵活定义的监控，通过httpGet、command或tcp socket进行
- (更细更)精确的粒度，相对于服务的API(URL)监控，liveness 可以精确到每个实例，通过更高频次的检查，先人一步发现故障
- 故障自动恢复(重启)，神奇的自愈功能

- 故障容器即时离线，故障恢复后自动上线

Httpget: Jenkins 中的设置, 通过变量传递信息

Execute shell

Command

```
load payment

APP_FILE="web-war/target/$APP_SUB"

LIVENESS_URL="/$APP_SUB/server/"

PaymentController
```

httpget: 模版定义

```
livenessProbe:
  httpGet:
    path: LIVENESS_URL
    port: APP_PORT
    scheme: HTTP
  initialDelaySeconds: LIVENESS_IDS
  timeoutSeconds: LIVENESS_TIMEOUT
  periodSeconds: LIVENESS_PERIOD
```

Httpget: 使用sed替换 (计划删除)

```
sed -i "s#LIVENESS_URL#$LIVENESS_URL#"              $f_lists
sed -i "s/LIVENESS_IDS/$LIVENESS_IDS/"              $f_lists
sed -i "s/LIVENESS_TIMEOUT/$LIVENESS_TIMEOUT/"      $f_lists
sed -i "s/LIVENESS_PERIOD/$LIVENESS_PERIOD/"        $f_lists
```

并没有高科技

存储：容器迁移与文件同步

数据、配置文件、引用项目、日志

- 数据库数据文件
- 配置及上传文件
- 调用的公共项目文件
- 日志文件

- 第一阶段，NFS
- 第二阶段，GlusterFS
- 第三阶段，Ceph

Ceph的使用，rbd & cephfs

- ceph-fuse 挂载，节点系统上挂载，容器挂载本地目录形式使用(脱离中)
- rbd，数据库及Hadoop等使用了这种方式挂载

Hadoop on K8s

- 某个项目使用HDFS存放文件，为确保环境一致性而创建
- 由于是第一次安装配置Hadoop，且又是直接运行与k8s之上，安装调试难度较大
- 经过无数次的尝试(其实也就七八十次)，终于使用StatefulSet成功运行了Hadoop
- 使用 cdh5 版本，计划后期改为官方版本
- Zookeeper 使用基于k8s文档中的模版运行

在K8s运行体系下的运维

- 日志查看，有些时候可能需要查看指定容器内日志
- 命令执行，某些情况下开发需要进入容器执行命令等等

- 使用 RBAC 控制权限，为开发人员提供受限的基于命令行与Dashboard的操作

Dashboard

- 新版本的Dashboard提供了执行命令的功能，配合RBAC权限控制，可以为指定用户或群组配置合适的权限，比如按照项目对应namespace设置权限，使用小组成员能且仅能访问本项目内容器资源。确保了集群内其它项目或namespace安全稳定运行

踩过的坑

- 1.5.2 升级到 1.8.1
- 存储挂载及性能问题拖垮集群
- 项目文件放在镜像里还是挂载进去？
- 如往常一样，开发想进入容器内操作，如何控制权限及定位容器？

展望未来

- 容器减负，小型化，持续改进
- 推进生产环境容器化
- 集中管理Ceph和K8s
- 基于LDAP认证、群组权限控制的容器管理方案