



OpsWorld 运维世界 大会·深圳站

DevOps之又拍云实践

运维的八荣八耻

- **DevOps**的八荣八耻
- **Heroku PaaS**的**12**要素宣言
- **OpenStack IaaS**的设计指导思想
- **Amazon Cloud**的超级接口

以可配置为荣 以硬编码为耻

程序生成程序

提炼应用参数



`/etc/upyun.cfg`

```
NGINX_UPSTREAM="  
192.168.251.163#1  
192.168.251.164#2  
192.168.251.165#3  
192.168.251.166#4  
192.168.251.167#5"
```

`/etc/init.d/nginx config`

动态生成配置文件



`/opt/nginx/conf/upstream.conf`

```
upstream hot.yupoo.com {  
    server 192.168.251.163:8100 weight=1;  
    server 192.168.251.164:8100 weight=2;  
    server 192.168.251.165:8100 weight=3;  
    server 192.168.251.166:8100 weight=4;  
    server 192.168.251.167:8100 weight=5;  
  
    consistent_hash $uri 2;  
}
```

一

以可配置为荣 以硬编码为耻

- 各种开关变量：`enable/disable true/false`
- 可调节参数：`max/min timeout weight`
- 上下游关系：`upstream`

Heroku十二要素之：

依赖 - 显式声明依赖关系

配置 - 在环境中存储配置

一

以可配置为荣 以硬编码为耻

- 本地配置,程序生成 (txt/ini/cfg)
- 集中配置, 动态生成(Yaml/Json)
- 环境变量(代码无侵入&语言无关性)
- 服务自动发现,自动注册(etcd/consul)

经济原则 (宁花机器一分,不费程序员一秒)

二

以互备为荣 以单点为耻

- LVS + Keepalived + VRRP (CDN/API/FTP)
- OSPF + Zebra + Quagga (CDN 二代)
- Nginx + Haproxy / LVS (SRS)
- Mysql master/slave/Galen MMM
- rabbitmq/redis/kafka
- consul / etcd / doozer / zookeeper
- hadoop/elk

墨菲定律：“凡事只要有可能出错，那就一定会出错。”

二

以互备为荣 以单点为耻

- 无状态/负载均衡(F5,LVS,Haproxy...)
- 无共享/消息队列(Redis,Kafka...)
- 松耦合/异步处理(Gearman,Celery...)
- 分布式/集群协作(Hadoop/Ceph...)

Heroku十二要素之:

进程模型方便并发

要做端口绑定

进程必须无状态而且无共享

三

以随时重启为荣 以不能迁移为耻

- **Pet -> Cow** 故障是常态
- **OpenStack** 虚拟机编排
- **LXC/RKT Docker** 导入导出

十二要素之：

进程必须无状态而且无共享

四

以整体交付为荣 以部份交付为耻

- **OpenStack** - 云计算, 云网络, 云存储
统一调度分配资源
任何资源都是可度量
在线使用,回收,备份,重载
- **Docker - Build, Ship, and Run**
统一配置,统一入口
开发,测试,运维三位一体
随时随地调用

Heroku十二要素之:

构建, 发布, 运行 - 严格分离构建和运行

开发环境与线上环境等价 - 尽可能的保持开发、预发布、线上环境相同

五

以无状态为荣
以有状态为耻

- 资源竞争
- 锁竞争
- 横向扩展差
- 不能重启/互备

五

以无状态为荣 以有状态为耻

- 可靠中间件(生产级/工业级/分布式)
- 公共资源池 (云实例/云数据库/云消息队列)
- 能够被计算(ceph/paxos, etcd/raft, hash++)

Heroku十二要素之:

后端服务 - 把后端服务当作附加资源

六

以标准化为荣 以特殊化为耻

- 统一输入输出(**gitlab / cmdb / ansible**)
- 统一流程控制(**git-ci / yaml / playbook**)
- 抽象底层设计 -> 复用组件 (函数库)
- **PC** , 容器服务, **server-less** = 微服务

香农-信息熵理论:

变量的不确定性越大, 搞清楚所需要的信息量也就越大。

因此一个孤立的系统, 始终会趋向于越来越乱(无序化)的方向发展。

七

以自动化工具为荣 以手动 + 人肉为耻

- **bash/sed/awk/rsync/fabric(业务模型标准化)**
- **puppet/ansible playbook**
- **tgz/rpm/pkg 打包部署**
- **集成测试/自动测试/发布**
- **ELK 集中日志分析/大数据分析**
- **不要图形,不要交互,不要终端**

十二要素之:

基准代码 - 一份基准代码, 多处部署

八

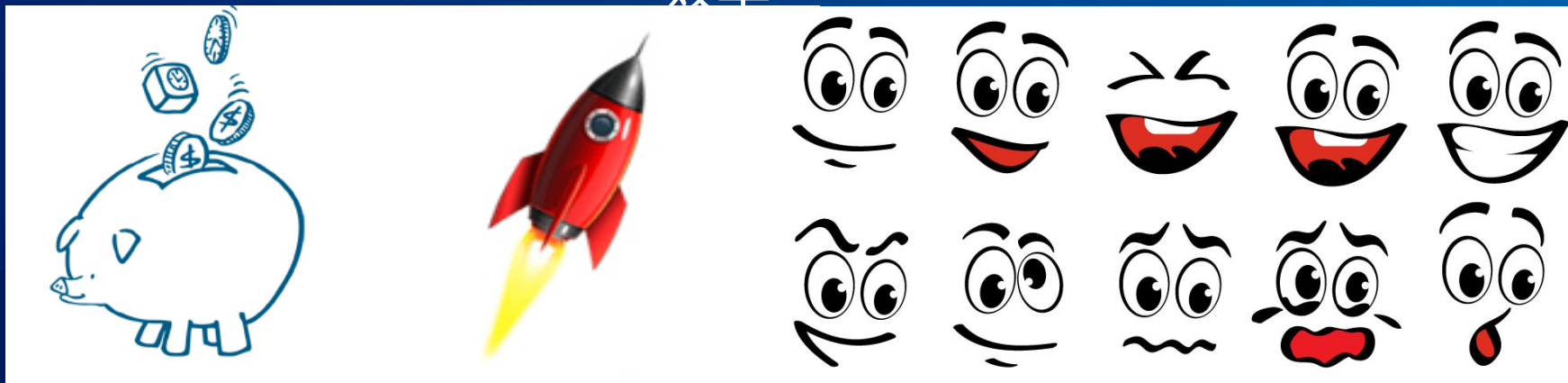
以无人值守为荣 以人工介入为耻

- 运维自动化(标准化组件,标准化流程)
- 监控常态化(同构大数据分析/处理)
- 性能可视化(数据的有效展示)

节省时间和精力

从容应付业务增长
效率

保持稳定的执行



Heroku PaaS的12要素宣言

- I. 基准代码 - 一份基准代码, 多处部署(标准化, 复用组件, 函数库)
- II. 依赖 - 显式声明依赖关系(**nodejs/go/maven**)
- III. 配置 - 在环境中存储配置
- IV. 后端服务 - 把后端服务当作附加资源(**url/服务发现**)
- V. 构建, 发布, 运行 - 严格分离构建和运行 (统一性, 标准化, 模板/容器)
- VI. 进程 - 以一个或多个进程, 必须无状态且无共享 (任何需要持久化的数据都存储在后端服务内)
- VII. 端口绑定 - 通过端口绑定提供服务(**Restful/http**)
- VIII. 并发 - 通过进程模型进行扩展 (方便扩展/分布式/弹性)
- IX. 易处理 - 快速启动和优雅终止可最大化健壮性(雅虎准则)
- X. 开发环境与线上环境等价 - 尽可能的保持开发、预发布、线上环境相同
- XI. 日志 - 把日志当作事件流(**rsyslog/heka/logstash**流式收集/过滤/处理/展示)
- XII. 管理进程 - 后台管理任务当作一次性进程运行(**server-less**, 微服务)

DevOps的本质 - 无关



- 与硬件无关 (硬件冗余, 数量冗余, 网络冗余...)
- 与人无关, 与己无关 (标准化操作, 自动化脚本...)
- 与程序无关 (随时启停, 热加载, 无硬编码, 弹性扩展, 负载均衡,...)
- 与激增无关 (大数据日志监控, API调度, 外部服务, 编排回滚...)

谢谢！