

# 公有云的自动化运维



# 背景

- 用户规模激增
- 硬件规模激增
- 用户业务种类繁多
- 人员增长速度跟不上业务增长速度

硬件数量		业务种类	IDC数量	运维人员数量
2012	三位数	2	2	2
2013	四位数	5	4	10
2014	五位数	10	7	20



# 遇到的问题及解决方法

问题：公司规模扩展太快，运维工具和平台的建设进度跟不上

解决办法：招人，使用开源工具+自主开发，使用第三方平台来作为补充



# 遇到的问题及解决方法

问题：作为云计算运维，既要负责内部运维，又要负担客户的运维工作

解决方法：内部运维团队和外部技术支持团队分离，各司其责。



# 遇到的问题及解决方法

问题：公司规模扩展太快，运维工具和平台的建设进度跟不上

解决办法：招人，使用开源工具+自主开发，使用第三方平台来作为补充



# 遇到的问题及解决方法

问题：物理机+虚拟机，os数量特别大，很多新问题浮现出来，arp表、mac表规模不足等等

解决方法：减少内部监控工具的arping、ping的频率，改变网络架构，采用SDN等技术，控制广播、组播。



# 遇到的问题及解决方法

问题：运营环境越来越复杂，人员水平参差不齐。

解决方法：CMDB保证运维数据的准确性，制定完善的运维流程，知识库建设。



# 遇到的问题及解决方法

问题：重复劳动太多，员工缺乏积极性，都不想做没技术含量的工作

解决方法：提高运营平台自动化水平，装机自动化，报表自动化，部署自动化，监控告警自动化。



# 自动化运维工具的选择及使用经验

2012: `omnitty/pssh`, 简单易用, 不需要安装agent, 可满足小规模服务器的批量维护。

2013: `ansible`, 易用, 不需要安装agent, 中等规模的服务器也能玩得转。

2014: `saltstack`, 轻量级, 定制性强, 可实现配置管理功能。



# 监控软件的选择及使用经验

2012: cacti, 易用, 无需agent, 配置简单, 功能少

2013至今: zabbix, 功能强大, 可定制agent, 分布式, 但基于mysql的存储, 磁盘IO压力大, 维护压力大。

2014: opentsdb, 基于hbase, 非常高的可定制性



*Thanks*