1. 构建监控知识体系

作者：bestyyl006
原文链接：https://ld246.com/article/1537764497559
来源网站：链滴

1.构建监控思维

1）一般运维的监控思路

面试过程中，问你“之前公司的监控怎么做的？”

得到的类似答案：“就是公司用的什么软件做的，再继续并不清楚如何回答”

答案没有问题，反应出运维人员的模糊的思路。

2）监控的目标

想一想，监控的目标是什么？

终极目标就是为了保证业务的持续和稳定运行。就是要让运维站在业务的角度的开始规划监控体系，不是某个技术范畴。

出发点都是日常运维监控

监控对象的理解：要监控的对象你是否了解呢？比如CPU到底是如何工作的？

监控对象的指标：要监控这个东西的什么属性？比如CPU的CPU使用率、负载、上下文切换。

确定报警基准线：怎么样才算是故障，要报警呢？比如CPU的负载到底多少算高？

如果上述的条件不满足，那就先不要开始实施监控了

3）监控的项目

1.监控的项目

硬件监控、系统监控、应用服务监控、流量分析、网络监控、安全监控、业务监控、日志监控、文件

控

2.引入Zabbix

那么多工具，选择最强大的zabbix

硬件监控：Zabbix IPMI Interface

系统监控：Zabbix Agent Interface

Java监控：Zabbix JMX Interface

网络设备监控：Zabbix SNMP Interface

应用服务监控：Zabbix Agent UserParameter

MySQL数据库监控：percona-monitoring-plulgins

URL监控：Zabbix Web 监控

4）运维监控思路

1.故障自动处理有没有相关的思路?

网上有相关的例子，如腾讯蓝鲸，他们实现了故障自愈的功能。

2.运维需要懂业务吗?

我个人的观点是懂业务能让运维走的更远，运维服务的对象不一定是其它部门，为什么不能是终端用呢?

可以站在终端用户的视角来做运维，比如有用户反映访问慢，为什么慢，是架构的原因，Nginx配置原因，还是数据库的原因。

不要等着领导来安排，运维能带来的价值是需要运维自己做出来的，思想有多远，运维就能走多远!

那么监控在这里发挥的作用就是：让数据说话!

3.做监控也需要CMDB吗?

CMDB（-Configuration Management Database 配置管理数据库）是基础，以业务为导向、以CMB为中心、以API为基础来进行运维体系的建设。尤其在自动化运维的模式下，一台服务器放在哪个位、配置是什么？什么时候购买的？什么时候过保？都运行了哪些服务、开放了什么端口、版本是什么升级时使用）。

2.监控目标

监控是整个运维乃至整个产品生命周期中最重要的一环，事前及时预警发现故障，事后提供详实的数用于追查定位问题。

选择一款开源的监控系统，是一个省时省力，效率最高的方案。

什么是监控，监控的重要性以及监控的目标，监控是需要站在公司的业务角度去考虑，而不是针对某监控技术的使用。

1.对系统不间断实时监控

2.实时反馈系统当前状态

3.保证服务可靠性安全性

4.保证业务持续稳定运行

3.监控方法

1.了解监控对象:如要监控cpu，那CPU到底是如何工作的?

2.性能基准指标:如监控，那CPU的使用率、负载、用户态、内核态、上下文切换。

3.报警阈值定义:如CPU的负载到底多少算高，用户态、内核态分别跑多少算高?

4.故障处理流程:收到了故障报警，要怎么处理，处理的流程是什么样的

4.监控核心

1.发现问题:发现问题怎么处理，报警或者报警邮件发送

2.定位问题:如一台服务器连不上，是网络问题、还是负载太高导致长时间无法连接，又或者某开触发了防火墙禁止的相关策略等，需要去分析故障原因。

3.解决问题:有故障通过优先级去解决故障。

4.总结问题:总结归纳，避免以后重复出现。

5.监控工具

1）老牌监控:

MRTG（Multi Route Trffic Grapher）：一套可用来绘制网络流量图的软件，由瑞士奥尔滕的Tobias Oetiker与Dave Rand所开发，以GPL授权。 用perl语言写成，可跨平台使用，数据采集用SNMP议，MRTG将手机到的数据通过Web页面以GIF或者PNG格式绘制出图像。

Grnglia：一个跨平台的、可扩展的、高性能的分布式监控系统，开销低，适合自动化监控，能处2000以上的环境。

Cacti：（英文含义为仙人掌）一套基于PHP、MySQL、SNMP和RRDtool开发的网络流量监测图形析工具，需要结合nagios使用。

Nagios：一个企业级监控系统，侧重于监控服务的可用性，能根据监控指标状态触发告警。 画能力不强，结合cacti使用。

Smokeping：主要用于监视网络性能，包括常规的ping、www服务器性能、DNS查询性能、SS性能等。

Smokeping的站点为：http://tobi.oetiker.cn/hp

OpenTSDB：用Hbase存储所有时序（无须采样）的数据，来构建一个分布式、可伸缩的时间序数据库。它支持秒级数据采集，支持永久存储，可以做容量规划，并很容易地接入到现有的告警系统。

2）王牌监控

***Zabbix**是一个分布式监控系统，支持多种采集方式和采集客户端，有专用的Agent代理，也支持SNP、IPMI、JMX、Telnet、SSH等多种协议，它将采集到的数据存放到数据库，然后对其进行分析整，达到条件触发告警。其灵活的扩展性和丰富的功能是其他监控系统所不能比的。相对来说，它的总

功能做的非常优秀。 ***

从以上各种监控系统的对比来看，Zabbix都是具有优势的，其丰富的功能、可扩展的能力、二次开发能力和简单易用的特点，读者只要稍加学习，即可构建自己的监控系统。

小米的监控系统：open-falcon。open-falcon的目标是做最开放、最好用的互联网企业级监控品。

OWL是TalkingData公司推出的一款开源分布式监控系统OWLgithub地址

3）三方监控:

如：监控宝、监控易、听云、还有很多云厂商自带监控。

4）Zabbix、Nagios、Open-Falcon比较

6.为什么要监控

1）实时提醒或者报告业务运行状态

2）有问题可以找到根源

3）掌握基础环境及业务应用系统的可用性

在软件系统的高可靠性（也称为可用性，英文描述为HA，High Available）里有个衡量其可靠性的准——X个9，这个X是代表数字3~5。X个9表示在软件系统1年时间的使用过程中，系统可以正常使时间与总时间（1年）之比，我们通过下面的计算来感受下X个9在不同级别的可靠性差异。

1个9：(1-90%)*365=36.5天，表示该软件系统在连续运行1年时间里最多可能的业务中断时间是36.天

2个9：(1-99%)*365=3.65天， 表示该软件系统在连续运行1年时间里最多可能的业务中断时间是3.6天

3个9：(1-99.9%)*365*24=8.76小时，表示该软件系统在连续运行1年时间里最多可能的业务中断时是8.76小时。

4个9：(1-99.99%)*365*24=0.876小时=52.6分钟，表示该软件系统在连续运行1年时间里最多可能业务中断时间是52.6分钟。

5个9：(1-99.999%)*365*24*60=5.26分钟，表示该软件系统在连续运行1年时间里最多可能的业务断时间是5.26分钟。

6个9：(1-99.9999%)*365*24*60*60=31秒， 示该软件系统在连续运行1年时间里最多可能的业务断时间是31秒

.....

7.监控系统的实现

一个监控系统的组成分为两部分：数据采集部分（客户端）和数据存储分析告警展示部分（服务器端，这两部分构成了监控系统的基本模型。

数据采集的工作模式可以分为被动模式（服务端到客户端采集数据）和主动模式（客户端主动上报数到服务器端）。大多数监控系统应该能同时支持这两种模式。被动模式对服务器的开销较大，适合小模的监控环境：主动模式对服务器的开销较小，适合大规模的监控环境。

采集数据的协议方式可以分为两种：专用客户端采集和共用协议采集（SNMP、SSH、Telnet等）

对于采集到的监控数据，可以将其存储到数据库或者文本或者其他方式，具体采用哪一种，应根据实需求来决定。