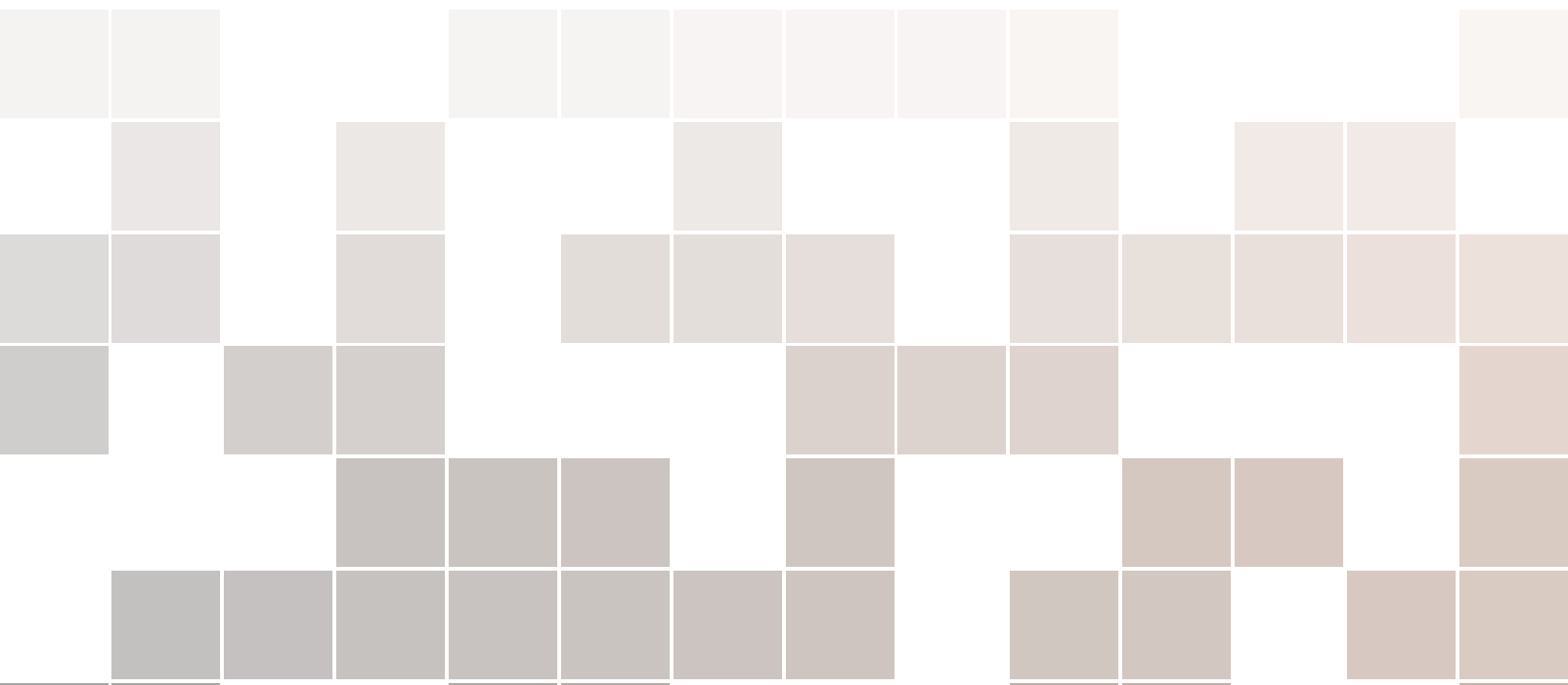


# iTop 实施杂记

An He



Copyright © 2017 An He

Published by L<sup>A</sup>T<sub>E</sub>X

<https://github.com/annProg/itop-book>

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “as is” basis, without warranties or conditions of any kind, either express or implied. See the License for the specific language governing permissions and limitations under the License.

最后编译日期, 2017 年 9 月 30 日 16:07:00

# 前言

笔者是互联网运维行业从业者，2014 年毕业后就进入一个知名互联网公司做基础运维工作，当时是很兴奋的，然而工作没几天就蔫了。每天做的事不外乎处理研发部门提过来的故障工单，给机房外包发重启重装工单等等。最讨厌的是改 IP，改完之后还要手动去 CMDB 更新 IP 以及上联交换机信息。总之就是复制粘贴。修修补补写了几个半自动化的脚本，依然感觉很无力。

第二份工作进入一个规模较大研发部门做运维。部门业务最小单元称为 APP，APP 会使用域名、数据库等资源。部门有 400 多个 APP，近 1000 的域名，1000 多台服务器，以及数百个 Mongo，MySQL 数据库。最直接的是问题仅凭记忆搞不清楚资源的归属，经常要在部门大群里问，这个域名是谁负责啊？这个数据库是我们的业务在用吗？这个 APP 报警了，zabbix 里没设置联系人，谁负责啊？如何有效的管控这些资源是个问题。

另外一个比较鸡肋的工作是监控报警的维护工作。报警以 APP 为最小单元，APP 负责人的变化是比较频繁的，业务调整，人员离职都会有负责人的变化。因此经常要做的事情是：业务交接人员变更后，去 zabbix 改报警接收人。对于 Url 监控，需要将研发人员提交的监控配置文件手动加入监控。还有一个讨厌的事情，充当 HUB 代研发人员向集团运维部门提交资源申请工单。还是复制粘贴的活。非常希望有一个自动化的手段来处理这些事情。

针对这些情况，笔者调研了几个开源 CMDB 系统，包括 yourcmdb，i-doit 等。最终选择了文档功能都较为完善，开发也较为活跃的 iTop 作为 CMDB 工具进行定制，并且实现了以下功能：APP，数据库，服务器，域名等资源入库；人员与 APP 关联，APP 与资源的关联，开放人员关联查询的接口给 zabbix，避免用 zabbix 维护报警联系人；定制 iTop 权限让研发人员可以编辑自己负责的 APP，相当于自助订阅报警；将 iTop 作为 Url 监控配置的前端表单，实现研发自助添加 Url 监控；定制工单系统，自动指派工单，自动录入新申请资源。实施之后，之前混乱低效的情况有所好转。但是距离理想状态还差很远，理想状态是我这个职位就不该存在，公司层面的运维提供良好的资源管控与监控报警系统，研发人员直接操作资源与报警，就像各种云计算平台那样，这样一来，我这个中间层 HUB 就可以失业了，运维的效率也可以提高了。

遗憾的是，公司层面并没有这样的平台。随着公司爆发的危机，才发现我这个职位作为一个部门的对接人在这种混乱环境里可能还少不了，否则可能会更混乱。为什么这么说呢？因为公司缺少一个标准化运维的体系。就拿业务线来说，没有标准统一的业务线名称（工单系统申请资源的时候业务线那一栏是文本，随便填），很多时候要靠部门来区分资源归属，这样确实需要一个部门接口人。数据库、域名等资源没有纳入公司 CMDB，也没有其他有效的管理方案，哪些在用，哪些下线，哪个业务在用，联系人是谁等等都不清晰。多次用群发 Excel 表格这种低效又没准头的手段来统计使用信息。梳理域名时发现我部门名下有 1200 多个域名，很多我没见过也 ping 不通的域名在列表里。数据库也有大量已经不再使用的。服务器更是有相当一部分的闲置。这种混乱带来的不仅仅是效率低下，还有高昂的运维成本

---

以及负反馈带来的恶性循环：资源无法得到有效的管控，不能及时回收，需要不断的购入新的服务器资源，最终只会越来越混乱。危机爆发后，在付不出钱的情况下，经常整个机房的关停，业务方也要中断正常工作忙着迁移业务，最严重的一次机房在没有通知的情况下直接断电，核心业务中断几个小时。

可以想见，如果有一个高效的标准化运维体系，有效的管控资源，即使在危机爆发的情况下，也不至于那样狼狈。这些经历，使我看到 **CMDB** 对 **IT** 组织的必要性，也感受到 **CMDB** 实施是一个范围很大也很复杂的工程。笔者从业时间不长，经验不丰富，对 **ITSM** 等理论也不熟悉。因此不打算从理论上探讨 **CMDB**，只从 **iTop** 的实施经历，谈谈 **iTop CMDB** 的定制与运维自动化的一些经验。

本书计划安排以下内容：第一章简单介绍一下 **iTop**，**iTop** 的插件开发流程，**CI** 定制，本地化等等 第二章 **CI** 属性约束，唯一性、**profile** 权限，只读，隐藏等等 第三章 **SSO** 集成方法 第四章 **iTop** 简介 **iTop** 插件开发流程 **CI** 定制（删除、新增、修改）**menunode** 定制本地化 **CI** 属性约束（唯一性，只读，隐藏）**SSO** 集成方法 **action-shell-exec trigger CI** 生命周期（**lifecycle**）**cmdbapi rest** 扩展 **api-client request-template**，工单自动指派，资源自动入库 **custom-pages iframe** 嵌入 **AttributeClassCustom** 其他常用插件介绍审计 **Profile**，自助化与权限控制实例（**URL** 监控）**api** 相关的可以放在系统集成部分

# 目录

封面	i
前言	iii
目录	v
表格列表	vii
插图列表	ix
第一章 SSO 集成方法	1
第二章 CI 校验	3
第三章 action-shell-exec	5



# 表格





## 插图



# 第一章 **SSO** 集成方法



## 第二章 **CI** 校验



## 第三章 **action-shell-exec**

