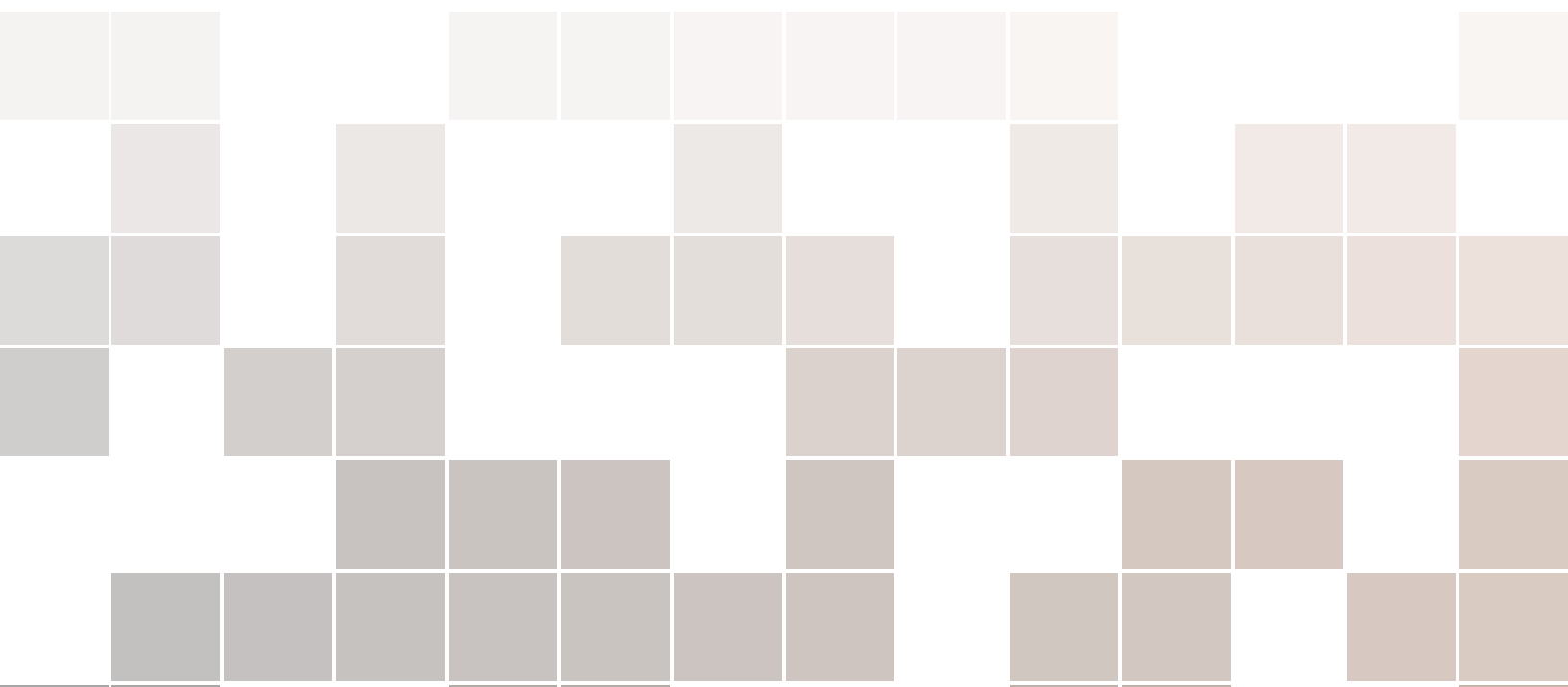


iTop 实施杂记

An He



Copyright © 2017 An He

Published by L^AT_EX

<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 12 月 29 日 16:45:00

前言

从事运维工作已经 3 年多，所见所闻，切身感受，当运维的规模大到一定程度的时候，资源的管控将变得很重要，主要体现在效率和成本两个方面。

举例来说，我接手了一个 URL 监控的工作，具体流程是：研发人员写一个监控配置文件提交到代码库，然后发邮件通知我，我执行一个脚本将监控加入 Zabbix。当规模很小的时候不会有什么问题，即使完全手动来配置都可以，我也能记住所有的 URL。一旦规模大到一定程度，比如说 50 个，我记不住了，或者每周每月都有几十的增减，需要经常投入时间去执行脚本，研发人员配置文件写错的时候还要来回邮件沟通。这个时候有一个合理的管控手段就非常重要了。要有一个系统来记录 URL 的监控信息以及状态，同步给监控系统，并且 URL 的增删改应该由研发自助完成，无需运维人肉参与，这样一来，监控变更速度（效率）提升了，运维也不用做这个重复的工作了。人力成本，沟通成本相应就下降了。

一个反例，我第二份工作，公司层面没有一个有效全面的资源管控的技术手段，没有全公司统一的业务线口径。服务器、数据库、域名、缓存等资源一般只能管理到部门粒度，就是说一个部门指定几个接口人，有事都找他们。这样的设计是很糟糕的，资源提供方把本该自己牢牢掌控的资源管理权下放给了各个部门，导致自己无法掌握资源的关联及使用情况，业务部门受限于人力或者技术，也常常不能有效的管控资源，常常是业务早下线了，数据库还长期占用着资源。在公司经历危机之后，拖欠了好几个机房的费用，这时开始大张旗鼓手忙脚乱的回收闲置资源或者关闭机房，统计资源的手段用的却是 Excel 表格、编辑 wiki 页面等形式。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，最终成本只会越来越高。

我认为正确的方向应该是用技术手段来减小资源管理的粒度，统一业务线名称，并用一个系统去搞清楚资源关联关系，而不是 Excel 表格。具体来说，业务要挂在具体的人名下，而资源要挂在业务下面，而不是某个人名下，这样就能容易的做到业务下线，相关资源一并下线。还带来一个额外的好处，人员离职只需要交接业务，因为资源关联的是业务而不是人，就不会出现人员离职找不到联系人的问题。这样不仅仅是减少了钱的支出，也提高了沟通的效率。只要粒度够细，关联关系够清晰，很多事情都会迎刃而解，比如报警，有这样一个 CI 关联数据库就不愁报警发给谁了，就不用去人肉维护 Zabbix 动作以及用户了。

针对以上问题，我希望引入 CMDB 系统来解决。调研过几个开源的 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，管理资源的关联关系，实现业务联系人查询接口供报警调用，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。实际效果是我部门的资源管理情况得到改善。由于没有机会在整个公司层面实施，有些观点可能并不全面。另外，

我对 ITIL 理论并不熟悉，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。

计划安排以下内容：iTop 简介，iTop 插件开发流程，常用插件介绍；然后通过几个案例说明 iTop 对运维效率提升机成本优化的帮助，包含一致性维护，工单流程，标准化，CI 关联关系，公共查询接口等主题。

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，自动化与权限控制实例（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**

