

运维自动化平台白皮书

目录

一、	概述	3
二、	功能介绍	3
1.	平台整体功能	3
2.	安装部署	4
3.	配置更新	4
4.	任务执行	4
5.	监控报警	5
6.	巡检管理	5
三、	技术特点	6
1.	Python 语言开发.....	6
2.	融合云计算平台.....	6
3.	规则知识库.....	6
4.	标准 RESTful API	6
5.	运维控制台	6

一、概述

本产品为运维自动化平台，集安装部署、配置更新、任务执行、监控报警、巡检管理等功能为一体，将运维管理员的经验和运维工具有效的结合，引入丰富的运维规则库，辅助管理员完成日常运维工作。

运维自动化平台立足于传统的数据中心架构，也能更好的支持 Openstack 等框架下的私有云平台 and 公有云平台，做到传统运维和云运维的结合。其设计原则是“平台化、模块化、松耦合、全开放”，以平台化、模块化实现工具集成、功能聚合，改变原有运检工具分散独立运行的现状，将运维工作全部整合在统一的平台中，并且各模块均提供标准化接口，满足模块化、松耦合的原则，可以与其他系统的功能模块方便地集成；其核心是从配置管理着手，配合监控工具，对各类应用系统进行从基础资源的部署到应用发布，再到运行维护的全生命周期的管理，最终实现运维的自动化、可视化、智能化。

二、功能介绍

1. 平台整体功能

(1) 权限管理

目前的权限管理主要指对平台的普通用户可使用的运维功能模块进行管理，由管理员统一进行权限的管理。如用户 A 只拥有安装部署的权限，则其他的权限对用户 A 来说是隐藏的。

(2) 用户管理

管理员对平台的普通用户进行增加、修改和删除的操作，也可以由使用者自己注册平台用户，并申请权限。注册功能可以启用或者禁用。

(3) 通知管理

用户可以接收到平台运行中发生较严重的事件，在平台使用界面的菜单栏中可以查看。

(4) 规则库管理

平台中的每个模块都需要建立规则库，以支撑运维操作的执行。目前规则库分散到各个模块中独立管理。

2. 安装部署

本功能主要分为两部分，一是实现对物理机的操作系统的推送和自动化安装，二是实现在目标操作系统上实现对中间件、数据库及其他软件的自动化安装、更新及卸载。

平台可以自动发现需要安装操作系统的物理服务器，然后再根据预先在模板库中定义的系统镜像来安装操作系统，支持主流的操作系统，如 Ubuntu、Redhat、CentOS 等。

安装软件需要从“软件商店”中选择，将软件和主机进行关联，实现对应操作系统的软件的安装及卸载。安装方式支持从软件源中安装，也支持自定义软件包安装。对传统的物理环境、Openstack 私有云以及公有云提供全面的支撑，有效减少企业内部基础设施环境的差异化。

3. 配置更新

平台提供丰富的配置更新管理功能，实现自动化修改目标主机的配置文件并使之生效。支持操作系统的配置文件，例如：hosts；也支持应用程序的配置文件，例如：sshd_config。

配置更新包括两个角度的实现：配置参数的修改以及配置文件的替换。配置参数的修改可以对多数的配置文件进行修改，完成参数的新增、更新以及删除；而复杂格式的配置文件目前只能通过配置文件的替换完成，该功能将替换掉原有的配置文件。

4. 任务执行

统一运维自动化平台能通过简单的操作在平台上完成对指定节点的命令行或脚本任务的创建与执行。任务执行模块主要分为三个子模块完成相应的功能，分别是：规则库、运行记录与统计分析。

用户在规则库中自定义名称、规则描述和需要执行的脚本命令，定义好后在平台上创建一个任务，选择指定的节点和指定的脚本模板，提交后后端程序立即执行或周期性执行任务，执行结束后平台会将任务执行的结果和自动化处理的结果反馈给用户。

用户在查询任务执行结果时，能鲜明的查看到结果与对应的脚本详情，包

括名称、状态、命令、节点和创建时间等，并且，如果一个脚本对多个节点执行，平台可以让用户选择查看全部结果或者是单条结果，单条结果即为某一个节点上的任务执行情况，可以根据用户的需求进行相应的选择。

在任务执行模块，每一个任务在执行后，后端都会对其进行统计分析，平台会实时将所有的任务执行的数量情况，以天为单位，反映到折线图上，用户可以在网页上随时进行查看与分析。

5. 监控报警

运维自动化平台采集服务器节点的性能数据和状态数据，以及数据库、中间件等的运行数据，生成监控图表，并根据预设策略，以短信、邮件方式发出报警信息。

对数据的监控，首先需要定义监控规则、监控项及监控模板。监控项定义了需要采集的监控数据，在定义的过程中需要设置一些属性。监控模板定义了监控项的集合。

对监控功能的管理，包括管理服务器节点的监控项，监控模板及触发器，可以关闭对某个节点的监控功能。当应用监控项后，就可以采集节点上这个监控项的数据。监控模板采集节点上多个监控项的数据。触发器可以根据监控项的数据在某种情况下触发事件。

平台还需要维护报警策略和报警对象。报警策略的定义，主要通过触发器的关联来实现。告警方式包括短信告警和邮件告警，需要有邮件网关和短信网关，才能发出告警信息。报警对象包括联系人及联系人组。

6. 巡检管理

对主机的巡检管理，获取巡检对象的状态和性能数据，对系统的运行状态进行检查，并生成巡检报告。包括以下内容：

巡检规则。用户在执行巡检任务前，需要自定义巡检规则，即如何对软件或硬件进行巡检。比如定义巡检的项目和阈值、巡检的策略、需要配置部署的软件等。规则是由脚本组成，支持 shell 脚本和 python 脚本。

巡检任务制定。巡检任务分为手动执行和自动执行两种。手动巡检在创建完巡检任务之后立即执行，而自动巡检在巡检执行周期自动执行，用户可以根据需要灵活设定。

巡检报告。手动巡检和自动巡检在结束后将巡检结果存储在数据库中，用户可以在界面中查看到结果，并输出巡检报告。

三、 技术特点

1. Python 语言开发

平台的前端和后端统一采用 Python 语言进行开发。严格来说，Python 是一门脚本语言，但由于其拥有非常多的模块以及优秀的 Web 框架，使其成为设计开发运维平台的首选。

2. 融合云计算平台

本产品在设计技术选择上和架构的设计上，考虑了与 Openstack、Cloudstack 等目前流行的开源云计算平台的融合，可以很好的支持云平台上的虚拟主机的自动化运维工作，可以将资源申请和运维工作更加的流程化、自动化。

3. 规则知识库

运维平台需要维护规则知识库，将运维人员的实际运维经验总结成规则知识库，并通过自动化的方式，完成运维操作。

规则知识库以 Shell 脚本或 Python 脚本的方式规定了实际动作，如通过哪种方式完成软件的安装，怎么收集巡检数据等。随着平台使用的深入以及知识库中规则的积累，运维自动化平台的运行会更加稳定，功能会得到更多的实现。

4. 标准 RESTful API

平台提供的运维功能均提供标准的 RESTful API，方便与其他系统进行集成，以及做二次开发工作。

5. 运维控制台

运维自动化平台的控制台采用与开源云计算平台 Openstack 相同的技术，即 Python 的 Django 框架、Bootstrap 样式和 AngularJS 库实现，充分考虑运维人员的操作习惯和方式，