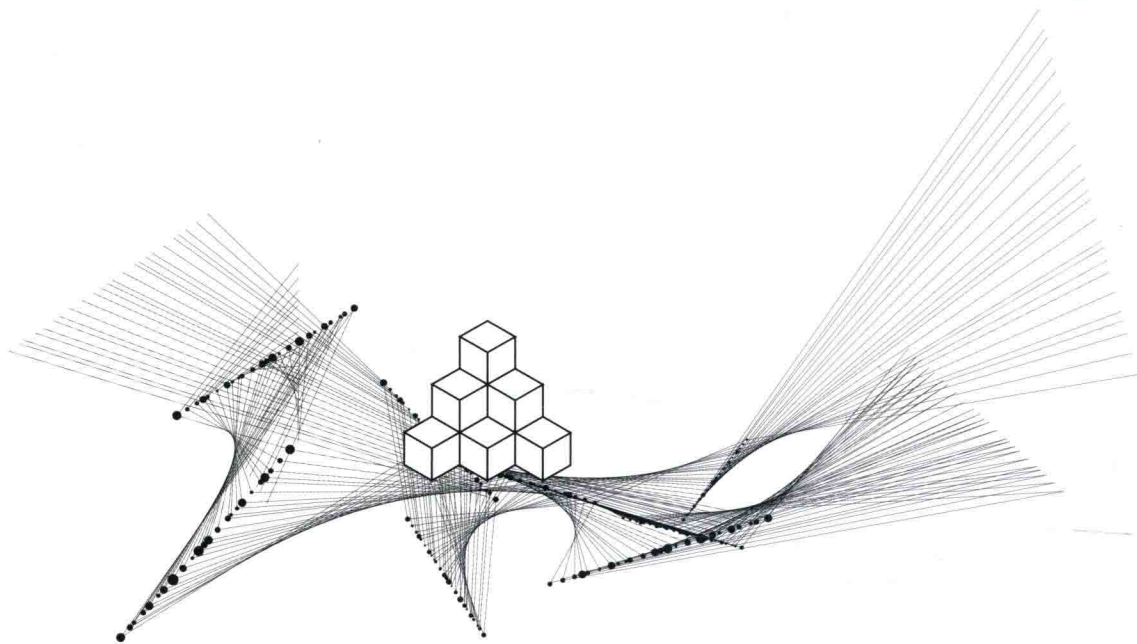


SaltStack公司创始人兼CTO诚摯作序推荐。

系统化地讲解 SaltStack 的概念、架构与工作原理，使读者对SaltStack有完整而深入的了解，可快速入门并融会贯通，进而搭建自己的自动化运维平台。



刘继伟 沈灿 赵舜东 编著

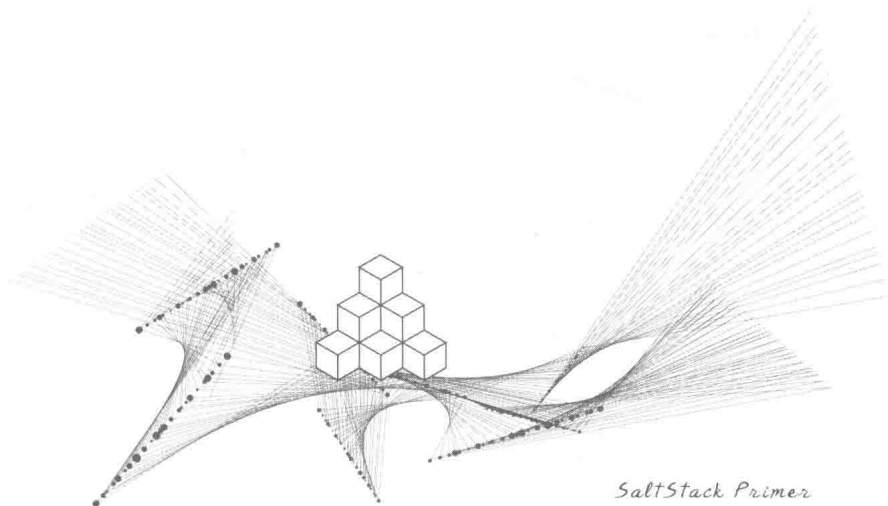
SaltStack Primer

SaltStack

技术入门与实战



机械工业出版社
China Machine Press



SaltStack Primer

SaltStack

技术入门与实战

刘继伟 沈灿 赵舜东 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

SaltStack 技术入门与实战 / 刘继伟, 沈灿, 赵舜东编著. —北京: 机械工业出版社, 2015.12

(实战)

ISBN 978-7-111-51822-8

I. S… II. ①刘… ②沈… ③赵… III. 数据处理软件 IV. TP274

中国版本图书馆 CIP 数据核字 (2015) 第 247225 号

SaltStack 是管理基础设施的强大工具, 为远程管理服务器提供了高扩展性、高速度、高效的轻量级解决方案。本书系统化地讲解 SaltStack 的概念、架构与工作原理, 使读者对 SaltStack 有完整而深入的了解; 通过典型案例讲解 SaltStack 的使用场景、技巧及注意事项, 使读者学以致用; 还介绍了 SaltStack 与运维相关的业务整合方案, 使读者基于 SaltStack, 但不拘泥于 SaltStack, 真正将 SaltStack 融会贯通, 作为自动化运维生态的连接点。主要内容包 括: SaltStack 的基础概念、组件和实例, 扩展 SaltStack 组件, 第三方调用 SaltStack, 扩展 SaltStack 架构, 开发 SaltStack Web 平台, 对 SaltStack 进行小的改造, 以及 SaltStack 排错与运维等。

SaltStack 技术入门与实战

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴 怡

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2016 年 1 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 14.25

书 号: ISBN 978-7-111-51822-8

定 价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

华章IT
HZBOOKS | Information Technology



Preface 序 言

When I started writing SaltStack I wanted to create a system that would be useful in many vast scenarios. A system that would be able to scale to any need, small or large, and a system that would allow people to better automate processes far beyond the datacenter.

To my knowledge, this is the first book on Salt to be published in China. I hope the contents of this book will prove valuable to those in China seeking to learn Salt Stack and that this book will be a resource to them.

China SaltStack User Group has been lead for many years by talented individuals who I would like to express my thanks and gratitude to. The Chinese users and contributors to Salt have helped the project move forward in immeasurable ways!

【译文】

当我开始 SaltStack 项目的时候，其实是想构建一个多种复杂环境均可用的（运维配置管理）系统，可以做到按需扩展，环境可大可小，能够真正帮助数据中心更好地实现自动化运维流程。

据我所知，本书是中国地区的关于 SaltStack 的第一本书籍。我希望本书的出版可以帮助中国地区那些正在苦寻 SaltStack 学习之路的朋友们。

中国 SaltStack 用户组（<http://www.saltstack.cn/>）自成立以来，持续地为社区做出了杰出贡献，来自中国的用户和参与者为推动 SaltStack 项目的发展起到了不可估量的作用，在此，我深表感谢和感激！

Thomas S. Hatch

SaltStack 公司创始人兼 CTO

前言 *Introduction*

为什么用 SaltStack

常规运维很大一部分工作涉及业务的配置管理和状态维护，目前，基于状态（系统状态、代码状态、配置状态和进程状态）的配置管理已经得到极大发展，并且让运维有了很大的进步；出现了各种工具和平台，从最早的 CFEngine 到后来的 Puppet、Chef，以及最近的 SaltStack、Ansible，每一种工具的出现都是为了满足新的场景以及解决之前解决不了的问题。新工具的出现层出不穷，令人眼花缭乱，在实际应用中，这些工具之间到底是替换还是结合，在每个特定的场景以及选型的理解上也会有所不同，最终可能以完全不同的形态进行展示。

但是人的基本特点是不会变的，就是我们都很懒，我们不想维护过多的平台，我们都希望学习过程尽可能简单，使用工具强大，二次开发的成本也低，从这几方面讲，SaltStack 可能是一个很好的选择。

本书定位以及读者对象

本书定位为了一本工具书，介绍一些 SaltStack 实用的功能，帮助想了解和入门 SaltStack 的朋友在较短的时间内，全面并且稍微深入地了解 SaltStack，进而自己进行探索深入。

勘误与支持

由于作者水平有限，时间比较紧张，书中难免会出现错误以及不准确的地方，希望读者能够批评指正。对于本书的任何问题请反馈到 SaltStack 中国用户组 QQ 群：294953305，我们会及时进行处理。

本书相关代码可以在 Github 上获取：<https://github.com/cssug>。

致谢

首先感谢 SaltStack 团队创造这么好的工具。特别感谢机械工业出版社以及吴怡编辑能够给我们一个机会来出版本书。感谢为本书做出卓越贡献的姚炫伟（微博 ID：绿小小肥）、曹正（微博 ID：CooCla）、张会源（微博 ID：神奇的魔法师）、吴兆松（微博 ID：itnihao）、白少华。最后感谢我们的家人和朋友一路给予的支持和帮助。

刘继伟

2015 年秋于北京

目 录 *Contents*

序言

前言

第1章 SaltStack入门	1
1.1 SaltStack 服务架构	2
1.2 SaltStack 架构安装	2
1.2.1 SaltStack 软件依赖	3
1.2.2 yum 方式安装 (采用 EPEL 源)	4
1.2.3 pip 方式安装	4
1.2.4 源码安装	4
1.2.5 salt-bootstrap 安装	5
1.3 开始 SaltStack 之旅	6
1.3.1 证书管理	6
1.3.2 日常命令参数	7
1.4 熟悉 SaltStack 配置文件	10
1.4.1 Master 配置文件	10
1.4.2 Minion 配置文件	14
第2章 SaltStack组件	17
2.1 从管理对象说起	17
2.2 管理对象属性	20

2.2.1 通过 Minion 配置文件定义 Grains	21
2.2.2 通过 Grains 模块定义 Grains	22
2.3 数据管理中心	23
2.4 针对管理对象操作	25
2.5 配置管理从这里开始	27
2.6 执行结果处理	31
2.7 Job 管理	33
2.8 Event 和 Reactor 系统	37
2.9 Renderer 组件	40
2.10 其他组件	43
2.10.1 Mine	43
2.10.2 Peer	44
第3章 SaltStack实践案例	49
3.1 环境规划	50
3.1.1 实验环境设置	51
3.1.2 SaltStack 环境设置	51
3.2 YAML 编写技巧	52
3.2.1 什么是 YAML	52
3.2.2 规则一：缩进	52
3.2.3 规则二：冒号	52
3.2.4 规则三：短横杠	53
3.3 Jinja 使用技巧	54
3.3.1 什么是 Jinja	54
3.3.2 如何区分模板文件	54
3.3.3 Jinja 基本使用	55
3.3.4 Jinja 逻辑关系	55
3.4 系统初始化	56
3.4.1 DNS 配置	56
3.4.2 History 记录时间	56

3.4.3	命令操作审计	57
3.4.4	内核参数优化	57
3.4.5	epel 仓库	58
3.4.6	Zabbix Agent 安装	58
3.4.7	初始化环境引用	60
3.5	Haproxy 配置管理	61
3.5.1	pkg 配置	62
3.5.2	Haproxy 服务配置	62
3.5.3	编写 Haproxy 安装 SLS	63
3.5.4	Haproxy 业务引用	65
3.5.5	执行 Haproxy 状态	67
3.5.6	查看 Haproxy 状态	67
3.6	Keepalived 配置管理	68
3.6.1	软件包准备	68
3.6.2	编写 Keepalived 安装 SLS	69
3.6.3	Keepalived 业务引用	70
3.6.4	执行 Keepalived 状态	72
3.6.5	Haproxy+Keepalived 测试	73
3.7	Memcached 配置管理	74
3.7.1	www 用户配置	75
3.7.2	Libevent 配置	76
3.7.3	Memcached 配置	76
3.7.4	Memcached 服务	77
3.7.5	执行 Memcached 状态	78
3.8	Nginx 配置管理	78
3.8.1	PCRE 模块	79
3.8.2	Nginx 模块	80
3.8.3	Nginx 配置文件	81
3.8.4	编写 Nginx 服务 SLS	82
3.8.5	PHP (FastCGI) 配置管理	83

3.8.6	PHP Redis 模块	86
3.8.7	PHP Memcached 模块	86
3.9	业务引用模块	87
3.9.1	BBS 论坛案例	88
3.9.2	修改 top.sls	89
3.9.3	执行状态	89
3.9.4	案例扩展的思路	92
第4章	扩展SaltStack组件	93
4.1	扩展 Grains	93
4.1.1	理解扩展 Grains 流程	94
4.1.2	扩展 Grains 案例	96
4.2	扩展 Module	96
4.2.1	理解扩展 Module 原理	97
4.2.2	扩展 Module 案例	99
4.3	扩展 state	102
4.3.1	理解扩展 state 原理	102
4.3.2	扩展 state 案例	104
4.4	ext_pillar 与 ext_nodes	108
4.4.1	理解 ext_pillar 流程	108
4.4.2	理解 ext_nodes 流程和案例	110
4.4.3	ext_pillar 实际案例	115
4.4.4	扩展 ext_pillar	119
4.5	SaltStack git 文件服务器	120
第5章	第三方调用SaltStack	127
5.1	通过 Python API 调用	127
5.1.1	Python API 工作原理	128
5.1.2	Python API 介绍	133
5.2	通过 RESTful API 调用	138

5.2.1 Restful API 配置环境部署	139
5.2.2 通过 Restful API 实现日常操作	141

第6章 SaltStack架构扩展

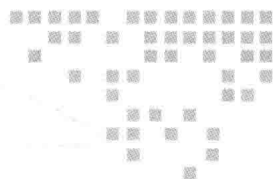
6.1 无 Master 架构	145
6.1.1 设置无 Master 运行	146
6.1.2 关闭 salt-minion	146
6.1.3 使用 salt-call 执行状态	146
6.2 多 Master 架构	147
6.2.1 配置多 Master	147
6.2.2 Master 数据存储	147
6.2.3 设置 Minion	148
6.3 Salt Syndic	149
6.3.1 Syndic 配置	149
6.3.2 高级 Master 配置	150
6.3.3 Syndic 测试	150
6.3.4 Syndic 是如何工作的	151
6.3.5 Syndic 的优缺点	151
6.4 Salt SSH	152
6.4.1 Salt Rosters	153
6.4.2 使用 Salt SSH	154
6.4.3 基本使用	154
6.4.4 生产实践	155

第7章 SaltStack Web 平台开发

7.1 SaltStack Web 平台开发流程	161
7.1.1 确定技术栈	162
7.1.2 做原型图	163
7.1.3 UI 设计	166
7.2 与 SaltStack 联动	166

7.3 实践	168
7.3.1 框架搭建	168
7.3.2 功能开发	170
7.3.3 效果演示	174
第8章 Hack SaltStack	177
8.1 实现命令执行结果的自动采集与监控	177
8.1.1 MySQL 建表	178
8.1.2 自定义 returner	179
8.1.3 通过 fluent 采集数据	180
8.1.4 结果查询	182
8.2 实现系统信息变更的自动上报	182
8.2.1 上报系统的需求与结构	183
8.2.2 软件选型	183
8.2.3 系统构建	184
8.2.4 基于 Event 的扩展	188
第9章 SaltStack排错	193
9.1 排错思路	193
9.1.1 确定问题范围	194
9.1.2 确保环境正常	194
9.1.3 定位问题	195
9.1.4 求助	198
9.1.5 Bug 修复	200
9.2 排错实战	203
第10章 SaltStack与运维	205
10.1 SaltStack 与自动化运维	205
10.1.1 运维的工作内容	205
10.1.2 运维自动化发展历程	207

10.1.3 运维标准化、工具化	207
10.1.4 运维 Web 化、平台化	208
10.1.5 运维服务化、API 化	209
10.1.6 运维智能化、自动化	210
10.2 SaltStack 与 CMDB	210
10.2.1 信息采集 & 存储	212
10.2.2 信息匹配	212
10.2.3 资产展示	213



第 1 章 *Chapter 1*

SaltStack 入门

SaltStack 是基于 Python 开发的一套 C/S 架构配置管理工具，它的底层使用 ZeroMQ 消息队列 pub/sub 方式通信，使用 SSL 证书签发的方式进行认证管理。

号称世界上最快的消息队列 ZeroMQ 使得 SaltStack 能快速在成千上万台机器上进行各种操作，而且采用 RSA Key 方式确认身份，传输采用 AES 加密，这使得它的安全性得到了保障。

虽然这里说 SaltStack 是一种配置管理工具，在工作中我们可能会使用它去做一些配置管理工作，但是 Saltstack 功能不止这些。在 SaltStack 官方的介绍中，SaltStack 不只是一个配置管理工具，还是一个做云计算与数据中心架构编排的利器。目前 Salt-cloud 项目也已经合并到 SaltStack 主项目里面，SaltStack 已经支持 Docker 相关模块。在友好地支持各大云平台之后，配合 SaltStack 的 Mine 实时发现功能就可以实现各种云平台业务自动扩展。在本书后面的章节中也会详细介绍 SaltStack 在各个领域中的应用。本章将介绍 SaltStack 的服务架构、安装、配置，你会发现，使用 SaltStack 如此简便。

1.1 SaltStack 服务架构

由于 SaltStack 是一种基于 C/S 架构的服务模式，可以简单地理解为如果我们想使用 SaltStack 就需要在现有的环境下引入与维护一套 C/S 架构。在 SaltStack 架构中服务器端叫作 Master，客户端叫作 Minion，在我们理解的传统 C/S 架构中，客户端发送请求给服务器端，服务器端接收到请求并且处理完成后再返回给客户端。在 SaltStack 架构中不仅有传统的 C/S 架构服务模式，而且有消息队列中的发布与订阅 (pub/sub) 服务模式。这使得 SaltStack 应用场景更加丰富。目前在实际环境中一般使用 SaltStack 的 C/S 架构进行配置管理。

在 Master 和 Minion 端都是以守护进程的模式运行，一直监听配置文件里面定义的 `ret_port` (接受 minion 请求) 和 `publish_port` (发布消息) 的端口。当 Minion 运行时会自动连接到配置文件里面定义的 Master 地址 `ret_port` 端口进行连接认证。默认客户端请求 id 是 `socket.getfqdn()` 取到的值，也可以在 Minion 启动之前修改 Minion 的 id 值。关于整个启动通信过程，大家可以使用 `debug` 查看详细记录。

C/S 架构如下：

□ Master 端：

```
SaltStack@Master: salt-master -l debug
SaltStack@Master: ss -a|egrep '4505|4506'
```

□ Minion 端：

```
SaltStack@Minion: salt-minion -l debug
```

SaltStack 除了传统的 C/S 架构外，其实还有 Masterless 架构，如果采用 Masterless 架构，我们就不需要单独安装一台 SaltStack Master 机器，只需要在每台机器上安装 Minion，然后采用本机只负责对本机的配置管理工作机制服务模式。

1.2 SaltStack 架构安装

经过对 SaltStack 的一些了解后，我们开始通过实际操作进一步学习 SaltStack。

下面分别介绍如何安装一个 SaltStack 服务架构。目前 SaltStack 支持很多平台的安装部署。具体每个平台的安装部署教程大家可以参照官方文档。在安装之前我们需要先了解下 SaltStack 的一些软件依赖，只有了解了安装和运行 SaltStack 的依赖，我们才能判断 SaltStack 到底适不适合我们的生产环境，以及能不能大规模地使用它。

1.2.1 SaltStack 软件依赖

SaltStack 对 Python 版本和 Python 模块有一定的要求。目前 SaltStack 有两种消息系统，一种是 RAET，另一种是 ZeroMQ，默认使用 ZeroMQ。目前 RAET 还处于测试阶段，所以本书不多介绍。软件依赖列表如下：

- ❑ Python 版本大于 2.6 或版本小于 3.0：对 Python 版本要求
- ❑ msgpack-python：SaltStack 消息交换库
- ❑ YAML：SaltStack 配置解析定义语法
- ❑ Jinja2：SaltStack states 配置模板
- ❑ MarkupSafe：Python unicode 转换库
- ❑ apache-libcloud：SaltStack 对云架构编排库
- ❑ Requests：HTTP Python 库
- ❑ ZeroMQ：SaltStack 消息系统
- ❑ pyzmq：ZeroMQ Python 库
- ❑ PyCrypto：Python 密码库
- ❑ M2Crypto：Openssl Python 包装库

下面通过一个案例简单介绍在 CentOS 6.5 上进行 SaltStack 部署。安装方式有四种，部署环境见表 1-1。

表 1-1 案例部署环境

角色	系统版本	主机名
Master	CentOS 6.5 x86_64	Master
Minion	CentOS 6.5 x86_64	Minion