Broadview®
www.broadview.com.cn

腾讯众多专家工程师联袂力荐

SaltStack 运维实战

简单快速管理服务器，完成服务器集群基础架构的建设，更为高效地管理基础架构

刘英杰 编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

SaltStack 运维实战

刘英杰 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

SaltStack（简称 Salt）是由 Thomas Hatch 于 2011 年创建的一个开源项目，初衷只是想构建一个快速的远程执行系统。后来经过快速发展和新功能的不断加入，演变成了现在的 Salt，如今 Salt 已经成为了一套强大的自动化运维管理平台。

本书力求用简洁易懂的方式给读者展示 Salt 的核心功能和使用思想，系统地介绍 Salt 的主要功能，从安装和最基本的远程执行开始，循序渐进地讲解 Salt 的方方面面，涉及 Salt 的模块代码编写、状态系统编写、架构扩展和 Salt 的高级应用等主题。书中实例丰富，希望读者可以通过本书掌握 Salt 的本质和思想，在自己的工作中应用 Salt 来提高运维效率。

本书适合运维人员以及任何和服务器相关的工作人员阅读，包括研发人员或业余爱好者都可以通过本书掌握如何简单快速地管理服务器，完成服务器集群基础架构的建设。

图书在版编目（CIP）数据

SaltStack 运维实战 / 刘英杰编著. —北京：电子工业出版社，2016.5
ISBN 978-7-121-28639-1

Ⅰ. ①S… Ⅱ. ①刘… Ⅲ. ①数据处理软件 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字（2016）第 085896 号

责任编辑：陈晓猛
印　　刷：北京中新伟业印刷有限公司
装　　订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×980　1/16　印张：15.75　字数：300 千字
版　　次：2016 年 5 月第 1 版
印　　次：2016 年 5 月第 1 次印刷
印　　数：3000 册　定价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819　faq@phei.com.cn。

前　言

为什么要写本书

系统管理员和运维人员日常会进行大量的重复性操作，诸如安装软件、修改配置文件、创建用户、批量执行命令等。如果主机数量庞大，单靠人工来维护，单调冗繁的任务实在让人难以忍受。早期的运维人员会根据自己的生产环境写特定脚本来辅助完成这些大量重复性的工作。但是这些脚本不但复杂，难于维护，更为重要的是不可移植。总体上讲，系统管理员面临的问题主要分为两大类：一是系统状态维护（配置管理），二是远程执行命令。为了解决这些问题，诞生了很多开源软件，系统状态维护方面有 Puppet、Chef、CFEngine、Ansible、SaltStack 等，这些软件擅长维护系统状态，可根据定义使相应的主机达到某种状态。维护主机的整个生命周期，实现从系统安装初始化到下线整个过程的管理和控制。在远程命令执行方面的软件则有 pssh、Fabric、Func、Rundeck、Ansible、SaltStack 等，它们可以方便地对大量主机进行批量的命令执行操作。无论采用哪款软件，系统管理员所面临的问题都是一样的，即如何在多变复杂的环境中完成灵活的配置管理和命令执行。这就需要所用的软件本身足够强大，而且具备很好的可扩展性。SaltStack 在这方面表现得十分出色，SaltStack（简称 Salt）是由 Thomas Hatch 于 2011 年创建的一个开源项目。初衷只是想构建一个快速的远程执行系统。后来随着各种新功能的不断加入，演变成了现在的 Salt。如今 Salt 已成为世界上最流行的开源项目之一，同时也是最流行的基础架构管理平台之一。

Salt 灵活性强，既可进行大规模部署，也能进行小规模的系统部署。Salt 的设计架构适应于任意数量的服务器，从少量本地网络系统到跨越数个不同的数据中心，拓扑结构都是简单的服务器/客户端模型，配置简单，默认的配置几乎无须更改，只需要微调即可满足特定需求。不管你有几台、几百台甚至几千台服务器，都可以使用 Salt 在一个中心节点上对它们进行管控。使用 Salt 可以灵活定位任意服务器子集来运行命令或完成任务。也可以使用状态系统来定义被管理服务器需要达到的状态，并且只需要一条命令就可以在很短的时间内让对应的服务器变成你所定义的角色。由于 Salt 是用 Python 编写的，允许用户通过 Python 语言自定义功能模块，同时也为用户提供了大量的 Python API 接口，所以用户可以根据自己的需要进行简单快速的扩展。

本书希望以一种简洁的方式引导读者掌握 Salt 的核心功能和理念。学完之后，相信您会对简洁强大的 Salt 爱不释手！

一些建议

如果读者之前没有接触过配置管理类的软件，习惯了用脚本编译部署服务器环境，需要转变一下管理服务器的理念，不要再用编译源码包的方式部署服务器，请习惯用描述式的语言来部署服务器环境，配合自建的 yum 仓库把需要编译的软件制作成 rpm 包的形式进行管理，这样既便于管理软件版本也不需要每次部署都在服务器上对软件进行编译。

学习本书所需的环境

您需要几台 Linux 机器来运行本书中所涉及的示例，可以用主流的虚拟机软件搭建一个或多个 Linux 系统环境。操作系统最好是 CentOS6.5，Salt 版本是 2015.5.5，当然大多数 Linux 主流版本都可以。如果您使用的操作系统不是 CentOS6.5，那您得到的输出结果可能和本书中的结果会略有不同。

读者对象

本书最主要的读者是系统管理员和 Linux 运维人员。此外，任何和服务器相关的工作人员，包括研发人员或业余爱好者都可以通过本书掌握如何简单快速地管理服务器，完成服务器集群基础架构的建设。本书适用于任何想使用 Salt 更为高效地管理基础架构的技术人员。

致谢

特别感谢我的朋友罗庆昌，没有他的帮助就不可能完成这本书。还得感谢我在新浪微博工作期间的技术领导王春生，在春生的帮助和指引下让我有机会更深入地接触配置管理领域，另外还要感谢我现在的技术领导 alanshao（邵宗文）和 shawnding（丁晓坤）以及我们运维中心副总监 zedanli（李震东），在他们的帮助下，让我对海量业务运维有了重新的认识。

让我们开启学习之旅吧！

刘英杰

2016.3.2

目　录

第 1 章　开始使用 Salt

SaltStack 是由 Thomas Hatch 于 2011 创建的一个开源项目，底层网络架构采用 ZeroMQ 实现。SaltStack 项目的设计初衷是为了实现一个快速的远程执行系统，后来在研发过程中不断加入新的功能，逐渐形成今天强大的 Salt。如今 Salt 的功能已远不止配置管理和远程执行这么简单，Salt 已演变为一个功能强大的平台，在这个平台上，既可以使用 Salt 提供的各种工具来管理服务器基础架构，也可以自己创建工具来满足特定需求。但所有这些强大功能实现的根基仍是 Salt 的远程执行系统（这也正是 SaltStack 的设计哲学）。这一切还是源于当时的设计初衷，所以我们的学习也应该从远程执行开始。

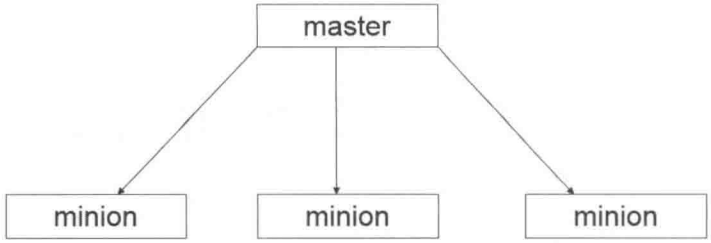
本章学习内容包括：

- Salt 部署的基本架构；
- 安装 Salt——包括软件包安装方式、脚本方式、源码方式及不同平台的安装；
- 配置 Salt——配置 master 服务器和 minion 服务器，建立连接关系；
- 执行第一条远程执行命令。

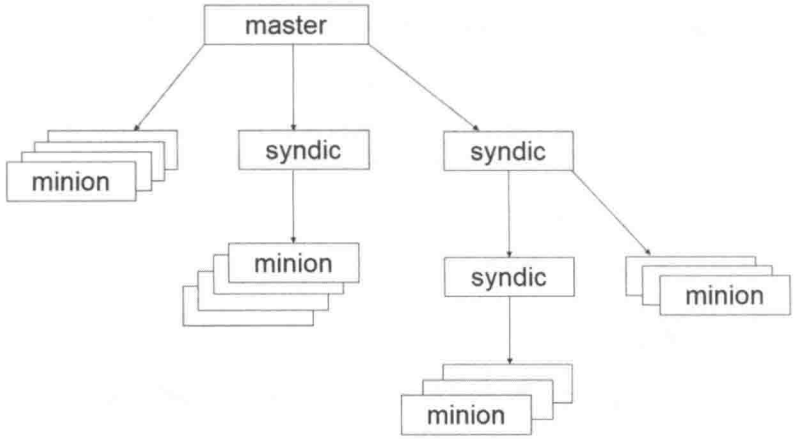
1.1　Salt 部署的基本架构

安装 Salt 前，首先要理解 Salt 基本架构中的各个角色。Salt 架构中最主要的角色是 Salt master 和 Salt minion（另外一种角色是 syndic，将在后面章节详细介绍）。顾名思义，master 是中心控制系统，而 minion 是被管理的客户端。Salt 部署架构可以分成三种。

第一种：master→minion，这种架构中 master 和所有 minion 都直接连接，minion 接收来自 master 的指令，完成命令执行或配置管理，如下图所示。



第二种：master→syndic→minion，这种架构中 master 通过 syndic 对 minion 进行管理，同时该架构可以进行多级扩展，如下图所示。



第三种：无 master 的 minion，这种架构中 minion 不受任何 master 控制，通过本地运行即可完成相关功能。

Salt 的两个主要设计理念是远程执行和配置管理。在远程执行系统中，Salt 用 Python 通过函数调用来完成任务。Salt 中的配置管理系统可以称作 state，也是基于远程执行系统之上，通过 master 的定义可以让对应的 minion 达到想要的系统状态。

理解上面这些基础知识之后，下面我们来学习如何安装 Salt。

1.2　安装 Salt

运行 Salt 所需的环境条件如下所示。

```
python >=2.6 <3.0
zeromq >=2.1.9
pyzmp >=2.1.9
pycrypto
msgpack-python
yaml
jinja2
```

通常解决软件包依赖的最简单方法就是用系统自带的安装包管理器，在 CentOS 系统上为 yum，Ubuntu 系统则为 apt；另外也可以使用 Salt Bootstrap 脚本来安装，Salt Bootstrap 是一个 shell 脚本，该脚本可在各种平台下自动正确完成 Salt 的安装，可自动判断系统类型并采用对应的软件包管理器完成 Salt 的安装和配置。还有一种安装方式就是源码安装，Salt 本身是 Python 程序，可以按照 Python 程序包的标准安装方法进行安装，下面将分别介绍这几种安装方式。

1.2.1　软件包安装方式

操作系统：CentOS 6.5 64 位。

首先安装 EPEL 的 yum 源，默认的仓库是不包含 SaltStack 的，执行如下命令：

rpm-ivh：http://mirrors.zju.edu.cn/epel/6/x86_64/epel-release-6-8.noarch.rpm

安装 epel-release 之后，执行如下命令，可以看到仓库中 SaltStack 相关的包：

```
# yum list|grep salt
python-salttesting.noarch          2015.7.10-1.el6          epel
salt.noarch                        2015.5.5-1.el6           epel
salt-api.noarch                    2015.5.5-1.el6           epel
salt-cloud.noarch                  2015.5.5-1.el6           epel
salt-master.noarch                 2015.5.5-1.el6           epel
salt-minion.noarch                 2015.5.5-1.el6           epel
```

```
salt-ssh.noarch                          2015.5.5-1.el6                 epel
salt-syndic.noarch                       2015.5.5-1.el6                 epel
```

安装完 EPEL 源之后就可以开始安装 Salt 了，CentOS 系统下安装 SaltStack 非常简单，借助 yum 命令系统可自动解决软件包的依赖问题。

安装 Salt master 时，如果不是 root 用户，命令前请加 sudo，命令如下：

```
# yum -y install salt-master
```

安装 Salt minion 时，如果不是 root 用户，命令前请加 sudo，命令如下：

```
#yum -y install salt-minion
```

安装完毕后可以执行如下命令来查看 Salt 安装信息：

```
salt --versions-report
           Salt: 2015.5.5
         Python: 2.6.6 (r266:84292, Jul 10 2013, 22:43:23)
         Jinja2: 2.2.1
        M2Crypto: 0.20.2
  msgpack-python: 0.4.4
    msgpack-pure: Not Installed
       pycrypto: 2.0.1
        libnacl: Not Installed
         PyYAML: 3.10
          ioflo: Not Installed
          PyZMQ: 14.3.1
           RAET: Not Installed
            ZMQ: 3.2.4
           Mako: Not Installed
        Tornado: 2.2.1
        timelib: Not Installed
       dateutil: 1.4.1
```

1.2.2 脚本安装方式

Salt 官方推荐用 Salt Bootstrap 进行脚本安装，首先下载 bootstrap-salt.sh 脚本：

```
    # wget -c https://raw.githubusercontent.com/saltstack/salt-bootstrap/stable/bootstrap-salt.sh --no-
check-certificate
    # sh bootstrap-salt.sh -h 可以查看帮助
    安装 salt-master 和 salt-minion
    # sh bootstrap-salt.sh -M 自动安装 salt-master 和 salt-minion
    单独安装 salt-master
    # sh bootstrap-salt.sh -M -N
    单独安装 salt-minion
    # sh bootstrap-salt.sh
```

1.2.3　源码方式安装

Salt 本身是一个标准的 Python 程序，所以可以通过 pip 命令进行安装，pip 命令可以自动解决 Python 软件包的依赖问题，命令如下：

```
    # pip install salt
```

另一种方式是下载软件包后用 python setup.py install 命令进行安装，首先需要安装官方文档要求的所有依赖软件包，然后下载 Salt 软件包，执行解压后用 python setup.py install 命令进行安装，过程比较烦琐，而且实际中几乎不会采用源码方式安装，所以不再详细介绍，有兴趣的读者可以自己动手实践下。

1.2.4　其他发行版 Linux 系统安装 Salt

有关其他主要发行版本 Linux 上部署 Salt 的方法，可以查看官方文档，都有详细说明。当然，大多数情况下，直接使用 salt-bootstrap 的脚本来安装，更简单方便。参照上面的脚本安装方式相信读者都可以轻易完成不同发行版本的 Linux 下部署 Salt，下面是 salt-bootstrap 脚本支持的 Linux 发行版本：

- Amazon Linux AMI 2012.09；
- Arch Linux；
- CentOS 5/6；
- Debian 6.x/7.x/8（git installations only）；
- Fedora 17/18；

- FreeBSD 9.1/9.2/10；
- Gentoo Linux；
- Linaro；
- Linux Mint 13/14；
- OpenSUSE 12.x；
- Oracle Linux 5/6；
- RHEL 5/6；
- Scientific Linux 5/6；
- SmartOS；
- SuSE 11 SP1 and 11 SP2；
- Ubuntu 10.x/11.x/12.x/13.x/14.x。

小知识：EPEL 是企业版 Linux 附加软件包的简称，是一个由特别兴趣小组创建、维护并管理的，针对红帽企业版 Linux（RHEL）及其衍生发行版（比如 CentOS、Scientific Linux、Oracle Enterprise Linux）的一个高质量附加软件包项目。

1.3 配置 Salt

完成 Salt 的安装后就可以开始配置 Salt 了，Salt master 启动后默认会监听两个端口：4505（publish_port）为 Salt Master pub 接口，提供远程执行命令发送功能；4506（ret_port）为 Salt Master Ret 接口，支持认证（auth），文件服务，结果收集等功能。要确保客户端能跟服务端的这 2 个端口通信，需要保证防火墙对于这两个端口是放开的。在本书的测试环境中防火墙是完全关闭的，读者可以执行 service iptables stop 来关闭防火墙，或者在 iptables 中对这个两个端口放开。Salt master 的配置文件是/etc/salt/master，minion 的配置文件是/etc/salt/minion，这两个配置文件中囊括了大量可调整的参数，这些参数可以控制 master 和 minion 的各个方面，在第 7 章中将详细介绍这些参数。现在我们需要用最简单的配置来完成 master 和 minion 的连接并执行第一条远程命令。

1.3.1　Salt minion 配置

Salt minion 和 master 通信时最好使用域名进行连接，通常在内网环境可以用轻量级的 DNS 软件 dnsmasq 搭建一个域名解析系统，根据 master 和 minion 的主机名和 IP 地址的对应关系设置 dnsmasq，然后将 master 服务器和 minion 服务器的 DNS 地址都设置为装有 dnsmasq 的服务器 IP 地址，这样就可以用域名进行连接。本书中为简便实验环境，直接修改 master 和 minion 的/etc/hosts 文件，步骤如下。

（1）在 master 和 minion 的 hosts 文件最后一行增加下面内容。

```
192.168.1.10 master
192.168.1.11 minion-one
```

（2）用 vi 打开/etc/salt/minion，我们将对该文件进行修改。

首先找出配置选项 master 的命令行，如下所示。

```
#master: salt
```

去掉#，并将 salt 更改为 master，如下所示。

```
master: master
```

如果在文档中找不到相应的命令行，则在该文件的尾部部添加一行。

然后找到 ID 行：

```
#id:
```

去掉#，并将其设置为 minion-one。

```
id: minion-one（这个名字不一定和主机名一样，但最好设置成一样）
```

再次强调下，如果找不到文件中的相应的行，则在该文件的尾部部添加一行，保存并关闭文件。

注意：如果未手动指定 minion ID，minion 将在启动时尝试智能化判断其 minion ID。对于大多数系统，minion ID 将设置为全域名（FQDN）。

1.3.2　启动 Salt master 和 Salt minion

现在我们需要启动（或重启）Salt master 和 Salt minion。如果不是 root 用户，执行下

面的命令时请加上 sudo 命令：

```
# service salt-minion start
# service salt-master start
```

命令执行成功后在 Salt master 上会出现 master 对应进程：

```
/usr/bin/python /usr/bin/salt-master -d
```

在 minion 上会出现 minion 对应进程：

```
/usr/bin/python /usr/bin/salt-minion -d
```

1.3.3 在 master 上接受 minion 秘钥

距离运行第一条 Salt 远程命令，我们仅有一步之遥。minion 在启动后连接 master 会请求 master 为其签发证书，待证书签发完成后，代表 master 可以信任该 minion，并且 Salt 和 master 之间的通信是经过加密的。Salt 配备了 salt-key 命令来帮助我们管理 minion 秘钥。在 Salt master 上执行：

```
# salt-key -L
Accepted Keys:
Unaccepted Keys:
minion-one
Rejected Keys:
```

请注意这时我们的 minion 即 minion-one 这台主机列已经出现在了 Unaccepted Keys 中。这表示该 minion 已经与 master 联系，并且 master 已经获取了 minion 的公钥，正在等待更多进一步有关是否接受该 minion 的指令。

我们可以查看秘钥的指纹码来确保其与 minion 的秘钥相匹配，在 master 上查询的命令如下所示。

```
# salt-key -f minion-one
Unaccepted Keys:
minion-one:  23:63:12:36:3a:9e:e2:55:5a:5d:11:38:91:8c:e1:41
```

我们可以在 minion 上运行 salt-call 命令来获取 minion 的秘钥，在 minion 上查询如下所示。

```
# salt-call --local key.finger
local: 23:63:12:36:3a:9e:e2:55:5a:5d:11:38:91:8c:e1:41
```

指纹码是相互匹配的，所以我们可以在 master 接受该秘钥，如下所示。

```
# salt-key -a minion-one
The following keys are going to be accepted:
Unaccepted Keys:
minion-one
Proceed? [n/Y] Y
Key for minion minion-key accepted.
```

我们可以检验该 minion 秘钥是否已被接受，如下所示。

```
# sudo salt-key -L
Accepted Keys:
minion-one
Unaccepted Keys:
Rejected Keys:
```

对于有很多 minion 的情况，可以在/etc/salt/master 配置文件中查找如下行：

```
#auto_accept: True
```

去掉#，然后保存重启 salt-master，让 master 完成自动签发，另外也可以用 salt-key -A 命令统一接受，这部分在后面章节讲解配置文件和 Salt 常用命令时会有详细说明。

现在准备工作都做好了，可以运行我们的第一条 Salt 命令了！

1.4　第一条命令测试

下面是我们的第一条命令：

```
# sudo salt '*' test.ping
minion-one:
True
```

看上去十分简单，这是一个简单的探测主机是否存活的命令。不用着急，我们很快就会学习更多更有意义的命令。首先来看这条命令，刚才我们执行的这条命令叫远程执行命令。