SaltStack在Ubuntu上的安装与使用

1. SaltStack简介

注意：本文档中，绿色命令是通用命令，红色命令是在主控端运行，蓝色命令是在受控端执行。

SaltStack 采用 C/S模式，master是server端（主控端），minion是client端（受控端），master与minion之间通过ZeroMQ消息队列通信。

minion上线后先与master端联系，把自己的pub key发过去，这时master端通过salt-key -L命令就会看到minion的key，接受该minion-key后， master则可以与minion互相通信，这时master可以发送任何指令让minion执行了，salt有很多可执行模块，比如说cmd模块，在安装minion的时候已经自带了，它们通常位于你的python库中，查看salt自带的所有东西：

locate salt | grep /usr/

这些模块是python写成的文件，里面会有好多函数，如cmd.run，当我们执行

salt '\*' cmd.run 'uptime'

的时候，master下发任务到匹配的minion上去，minion执行模块函数，并返回结果。master监听4505和4506端口，4505对应的是ZMQ的PUB system，用来发送消息，4506对应的是REP system，是用来接受消息的。

1.1 YAML语法

SaltStack的配置文件均采用YAML语法，必须严格按照YAML语法来编辑配置文件。

1.1.1 缩进

YAML中不要使用Tab，YAML推荐缩进为2个空格，’:’和’-’后面要缩进1个空格。

1.1.2 数字会解析成数字

如mode: 0644会解析成mode: 644,你可以用’括住，比如mode: ’0644′，以避免mode: 0644被解析成mode: 644

1.1.3 不允许双简写

1.1.4 字符集

YAML只支持ASCII，其它字符集最好不要使用，如果非要使用则按照以下格式:

•micro: '\u00b5 '

1.1.5. 下划线\_将会被删除

date: 2013\_05\_13 会被解析成 date: 20130513

通过'括住防止出现该问题

date: '2013\_05\_13'

2．下载与安装（本文档操作均在su下进行）

2.1 安装Python及相关包

需要先安装Python，如未安装Python请先安装，然后还要执行如下命令安装Python相关包：

sudo apt-get install python-software-properties

可能还要执行如下命令行：

sudo apt-get install software-properties-common

2.2 安装SaltStack PPA

sudo add-apt-repository ppa:saltstack/salt

你也可以指定PPA的发布版本，目前可以使用的有：salt16, salt17, salt2014-1, salt2014-7, salt2015-5。例如，安装2015.5.x版本就执行如下命令：

sudo add-apt-repository ppa:saltstack/salt2015-5

2.3 安装PPA key

执行如下命令（是的，你没有看错，就是这么长，复制运行就是了）：

echo deb http://ppa.launchpad.net/saltstack/salt/ubuntu `lsb\_release -sc` main | sudo tee /etc/apt/sources.list.d/saltstack.list

wget -q -O- "http://keyserver.ubuntu.com:11371/pks/lookup?op=get&search=0x4759FA960E27C0A6" | sudo apt-key add –

2.4 更新包数据

sudo apt-get update

2.5安装salt相关包

apt-get install salt-api

apt-get install salt-cloud

apt-get install salt-ssh

apt-get install salt-syndic

到目前为止，上面的操作是每一台机器都要执行的，现在的情况是，SaltStack分主控端和受控端，在你想作为主控端的Ubuntu上安装salt-master，运行如下命令：

apt-get install salt-master

在你想作为受控端的Ubuntu上安装salt-minion，运行如下命令：

apt-get install salt-minion

3. 相关配置

3.1 主控端配置

在主控端执行如下命令：

vi /etc/salt/master

分别找到如下相关字段，去掉注释符号，修改后保存退出：

interface: 0.0.0.0

user: root

log\_file: /var/log/salt/master # 记录主控端运行日志

key\_logfile: /var/log/salt/key # 记录认证证书日志

3.2受控端配置

先在主控端执行ifconfig查看主控端IP

可以知道，主控端IP是129.188.33.253，则在受控端执行：

vi /etc/salt/minion

分别找到如下相关字段，去掉注释符号，修改后保存退出：

master: 29.188.33.253 # 设置主控端IP

user: root

id: ubuntu-minion-001 # 设定受控端编号

log\_file: /var/log/salt/minion # 记录受控端运行日志

key\_logfile: /var/log/salt/key # 记录认证证书日志

4. 运行与测试

4.1 运行受控端

受控端执行如下命令，在前台运行则执行如下命令：

salt-minion

或者让minion在后台运行：

salt-minion -d

这样，受控端就跑起来了。

4.2 运行主控端

主控端执行如下命令，在前台运行则执行如下命令：

salt-master

或者让master在后台运行，则执行以下命令：

salt-master -d

主控端就起来了。

4.3 测试

在主控端执行如下命令：

salt-key –L

可以看到受控端情况，如下，你可以看到未接受的受控端列表：

这时，你可以接受特定的受控端，主控端运行

salt-key -a ubuntu-minion-001

也可以一次性接受所有的受控端，主控端运行

salt-key –A

为了测试受控端受控情况，主控端运行：

salt ubuntu-minion-001 test.ping

OK，可以正常运作。你也一次性测试所有的受控端受控情况，在主控端运行：

salt '\*' test.ping

5. 文件同步

5.1 主控端配置同步根目录

在主控端执行如下命令：

vi /etc/salt/master

如下修改配置文件后保存退出：

file\_roots:

base:

- /srv/salt/

dev:

- /srv/salt/dev/

主控端执行：

mkdir -p /srv/salt/dev

ls -ld /srv/salt/dev

然后主控端要重启一下master服务：

service salt-master restart

5.2 介绍cp.get\_file

使用cp.get\_file可以从master端下载文件到minion的指定目录下。

5.3 在主控端创建测试文件

主控端运行如下命令：

echo 'I am test file. ' > /opt/test.txt

运行如下命令查看测试文件的内容：

cat /opt/test.txt

5.4 文件分发

主控端运行如下命令，将test.txt复制到到我们在4.1中指定的同步目录下：

cp /opt/test.txt /srv/salt/

接着在主控端运行如下命令，将文件同步到受控端ubuntu-minion-001的/tmp/目录下：

salt 'ubuntu-minion-001' cp.get\_file salt://test.txt /tmp/test.txt

在受控端执行如下命令，可以查看刚才同步的文件test.txt的内容：

cat /tmp/test.txt

5.5 附加的知识

5.5.1 压缩方式gzip

使用gzip的方式进行压缩，数字越大，压缩率就越高，9代表最大的压缩率。

salt 'ubuntu-minion-001' cp.get\_file salt://test.txt /tmp/test.txt gzip=9

5.5.2 创建目录makedirs

设置该属性为true，当分发的位置在目标主机上不存在时，自动创建该目录。

salt 'ubuntu-minion-001' cp.get\_file salt://test.txt /tmp/srv/test.txt makedirs=True

6. 目录同步

cp.get\_dir与cp.get\_file的用法十分相似，用来将整个目录分发到minions

创建测试文件。

6.1 准备

主控端执行如下命令新建一个测试目录test\_dir：

mkdir /srv/salt/test\_dir

在该目录下新建两个文本文档：

echo 'hello word !!' >>/srv/salt/test\_dir/hello1.txt

echo 'hello2 word !!' >>/srv/salt/test\_dir/hello2.txt

查看一下是否创建成功：

ll /srv/salt/test\_dir/

没有问题。

6.2 目录分发

执行目录文件的分发，并使用压缩传输：

salt 'ubuntu-minion-001' cp.get\_dir salt://test\_dir /tmp gzip=9

6.3 受控端查看分发情况

登录到目标节点查看分发状态：

ll /tmp/test\_dir/

OK，目录同步成功。

7. 远程批量执行

格式：

salt '<target>' <function> [argument]

注： function是salt带的或自己写的可执行模块里面的function，自带的所有列表http://docs.saltstack.com/ref/modules/all/index.html?highlight=full%20list%20builtin 实例：

salt '\*' test.ping