Saltmaster - How to Build

Description

This document describes the full process to build a salt master from scratch Requires vm created in AWS with public ipaddr

Install Saltstack components

- ssh to public ip of new vm using ssh key specified when creating the vm in AWS UI
- sudo -s (need to be root for this setup)
- curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"
- python get-pip.py
- sudo yum install https://repo.saltstack.com/yum/redhat/salt-repo-latest-1.el7.noarch.rpm

> example to specify an older version:
>
> USE PINNED version:
>
> sudo yum install https://repo.saltstack.com/yum/redhat/salt-repo-2015.8-3.el7.noarch.rpm
>
> (If sudo yum install https://repo.saltstack.com/yum/redhat/salt-repo-latest-1.el7.noarch.rpm doesn't work, use sudo yum install https://repo.saltstack.com/yum/redhat/salt-repo-latest-2.el7.noarch.rpm)

- sudo yum clean expire-cache
- sudo yum install salt-master
- sudo yum install salt-minion
- sudo yum install salt-cloud

Enable auto-start for salt services

- systemctl enable salt-master.service
- systemctl enable salt-minion.service
- systemctl start salt-minion.service
- systemctl start salt-master.service

Configure gitfs backend file system

- sudo pip install gitpython
- sudo yum install git-all
- systemctl restart salt-master.service
- configure /etc/salt/master config with settings: reference master config
- create sshkey pair (this one used by salt to bootstrap minions)
- add pub key to github account
- assure keypair is chmod 400 and copy to /root/.ssh /home/centos/.ssh

> Optionally
>
> can also create /root/.ssh/config with the following or similar
>
> Host github.com
>
> IdentityFile /home/centos/.ssh/id_rsa
>
> StrictHostKeyChecking no

- restart services and verify
- systemctl stop salt-master.service
- systemctl start salt-master.service
- view salt-master log /var/log/salt/master

** gitfs should now be setup and functioning

Install other required packages

- pip install wget tree
- yum install telnet
- yum install gpg
- sudo yum install python-setuptools python-setuptools-devel
- pip install awscli simplejson boto3

Configure salt master for AWS api access

- create local file ~/.boto Add the following

```
[Credentials]
aws_access_key_id = AWS_ID
aws_secret_access_key = AWS_ACCESS_KEY

[Boto]
ec2_region_name = us-east-1
ec2_region_endpoint = ec2.us-east-1.amazonaws.com
```

> AWS_ID and AWS_ACCESS_KEY should be valid AWS credentials with api access

- create aws local file ~/.aws/credentials Add the following

```
[default]

aws_access_key_id = AWS_ID
aws_secret_access_key = AWS_ACCESS_KEY
```

- Test aws connectivity after creating .boto file and .aws/credentials
    - aws ec2 describe-instances --region us-east-1

Configure salt-master to be a minion of itself

edit /etc/salt/minion

- need to set these:

```
master: 127.0.0.1
environment: envX
pillarenv: envX
```

- save and close
- systemctl stop salt-minion.service
- systemctl start salt-minion.service
- systemctl stop salt-master.service
- systemctl start salt-master.service
- accept keys:
    - salt-key -A
- salt '*' test.ping (tests master->minion)
- salt-call grains.get saltpath
- (tests minion->master) create role grain for salt master
- salt 'whateverthenodename' grains.set role saltmaster
- Copy/upload saltmaster.pub key to aws

Install Conductor python egg

- pull from github and drop in /home/centos
- mkdir -p /srv/runners
- easy_install -s /srv/runners conductor-1.0-py2.7.egg
- run a conductor command

```
EXAMPLE ONLY:
salt-run conduct.group create group=salty role=activemq count=1
saltenv=dev region=us-east-1a
```

GPG setup on salt-master

- gpg --gen-key (doc says gpg gen-key)
- select 1024 (do not enter passphrase)
- when prompted for create User, enter the key name such as 'saltmaster_encrypt' or something
- when key is created, COPY ALL FILES FROM /etc/salt/gpgkeys/.gnupg to the root /etc/salt/gpgkeys
- then you are all set to encrypt:

```
echo -n 'This is encrypted!' | gpg --armor --encrypt -r
'saltmaster_encrypt'
```

Trouble shooting

- make sure ports 4505 and 4506 are open between salt master network and minions.
- If desired, install locate on saltmaster (for troubleshooting)

Change Hostname and ID of saltmaster

The salt-master would have id of ip-xxx-xxx-xxx.domain.com, we want to make it generic as it's used in the pre provision orchestration hooks.

Perform these steps:

- salt 'ip-xxx-xxx' grains.setval id saltmaster (change id grain to new desired name)
- salt 'ip-xxx-xxxx' grains.setval nodename saltmaster (change nodename grain to new desired name)
- salt-call network.mod_hostname saltmaster (run state on master to change the hostname)
- systemctl stop salt-minion.service
- systemctl start salt-minion.service
- vi /etc/salt/minion_id (change to saltmaster)
- systemctl stop salt-minion.service
- systemctl start salt-minion.service
- salt-key -A (accept the new key finger)
- salt-key -d 'ip-*eliza.com' to remove the old finger

Saltstack install reference

https://repo.saltstack.com/#rhel