

# SaltStack Overview

Initial Commit

# About me

SaltStack Certified Engineer # 23

Presenter at SaltConf15

Various projects using SaltStack as a key component:

- hadoop - real clusters
- white hat botnet - mobile game company
- fast parallel lamp stack application deployment
- configuration management
- custom web application testing
- Phoenix - open source HA implementation on AWS

# My take on SaltStack

## Fast solutions platform

- parallel remote execution
- configuration management
- simple api to program custom solutions
- very flexible
- ideal for projects that require orchestration between servers
- active helpful open source community

The swiss army knife for DevOps!

# Major components of SaltStack

Master: the “controller” of servers

Minion: a controlled server

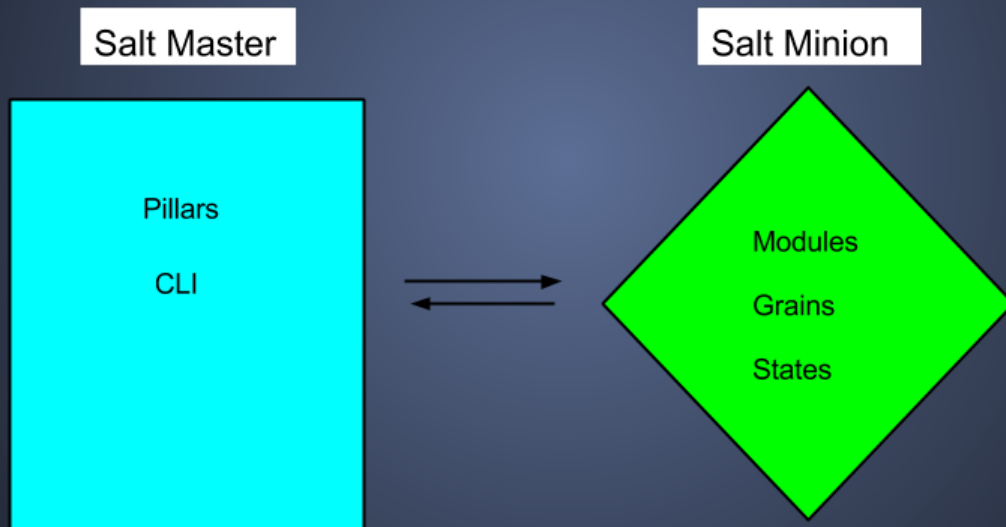
Grains: static info about a server

Pillars: key/value lookup dictionary (on master)

Modules: python execution modules (run on minions)

States: configuration management (built on top of modules)

# SaltStack Components



# Masters and Minions

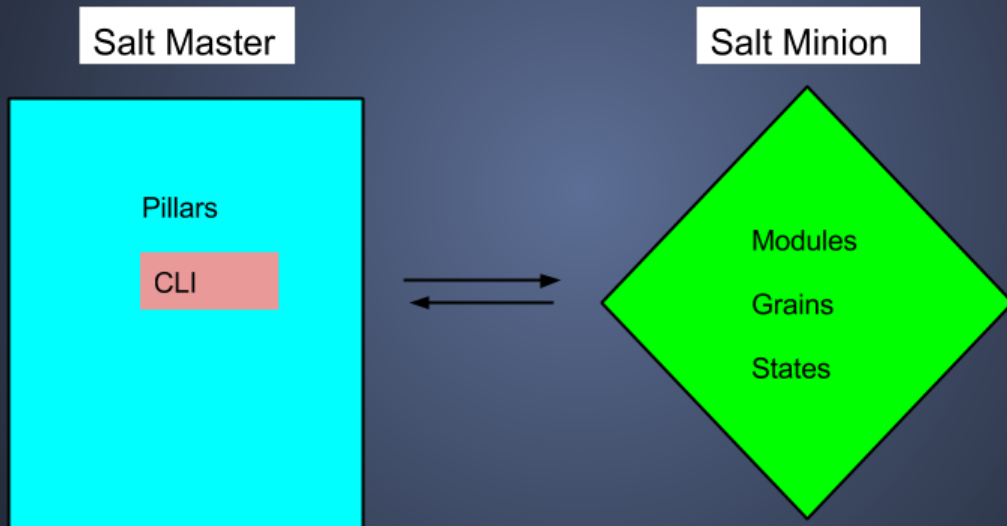
## Salt Master:

- handles key management
- agent based minions attach to master (zeromq) to receive / publish messages (tcp 4505/4506)
- ssh based minions must be accessible via ssh (tcp 22)
- code is transported from master to minions (e.g. modules/states/grains)
- pillar information resides on master

## Salt Minion:

- listens to messages on master for execution, config management
- data is compiled on minion
- actions take place (generally) on minion, results are returned to the master
- config file on minion “points” to master (ip/dns)

# SaltStack CLI



# salt command line

## general syntax

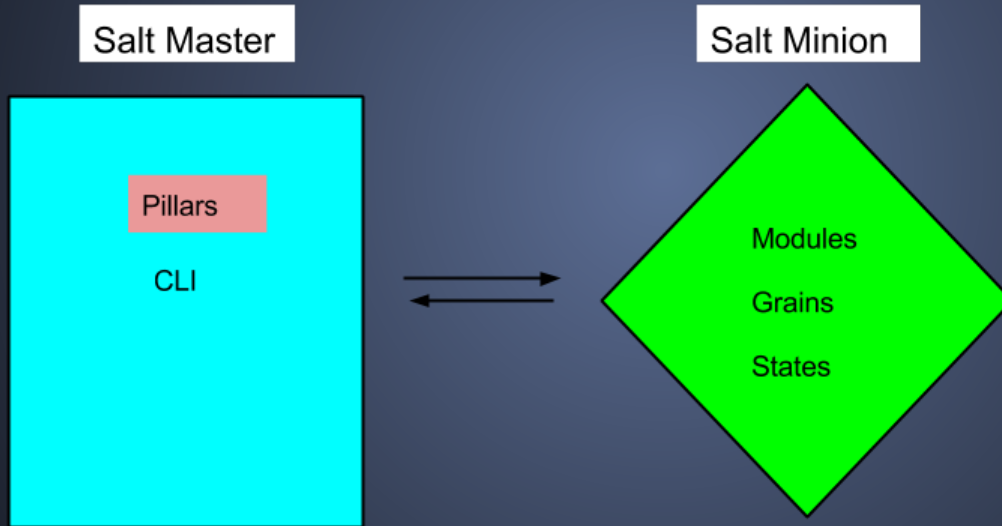
(remote) salt [options] '<target>' <function> [arguments]

(remote) salt-ssh [options] '<target>' <function> [arguments]

(local) salt-call [options] '<target>' <function> [arguments]



# SaltStack Pillar



# Salt Pillar

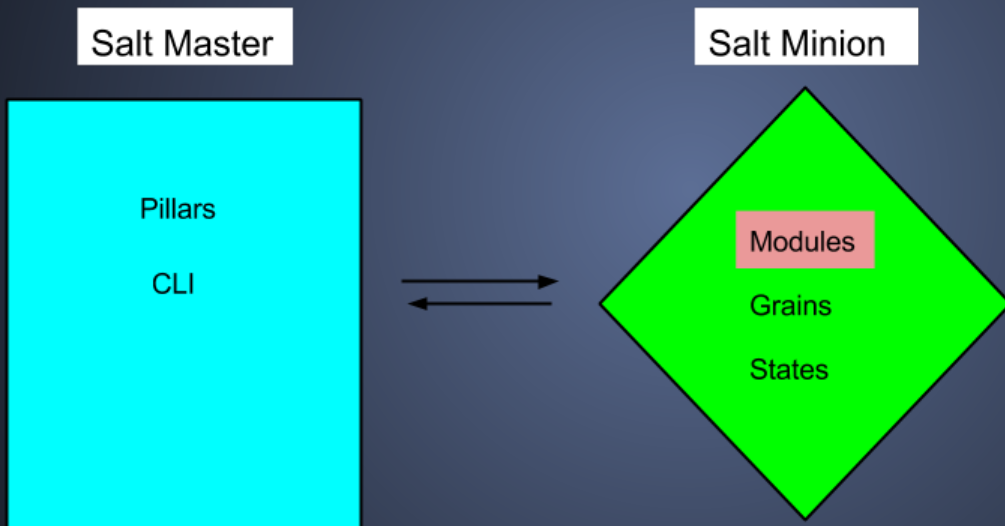
Key / Value pairs, can be nested

Python dictionary

Handy for secure distribution of information

Nice way to override values in config mgmt

# SaltStack Modules

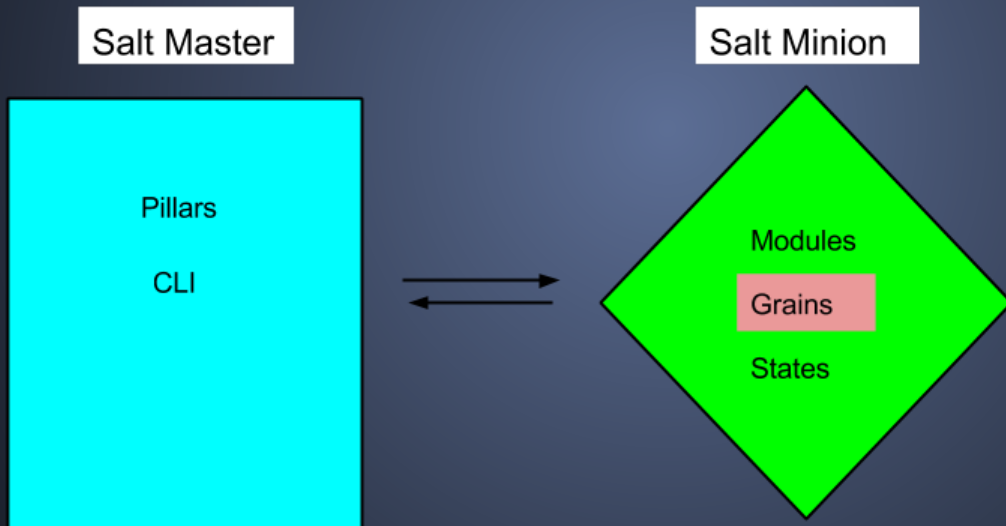


# salt modules

The heart of salt is remote execution

- done by standard python modules
- modules have access to salt dictionaries (e.g. grains/pillars)
- a module can execute other modules
- a module can execute code on other servers

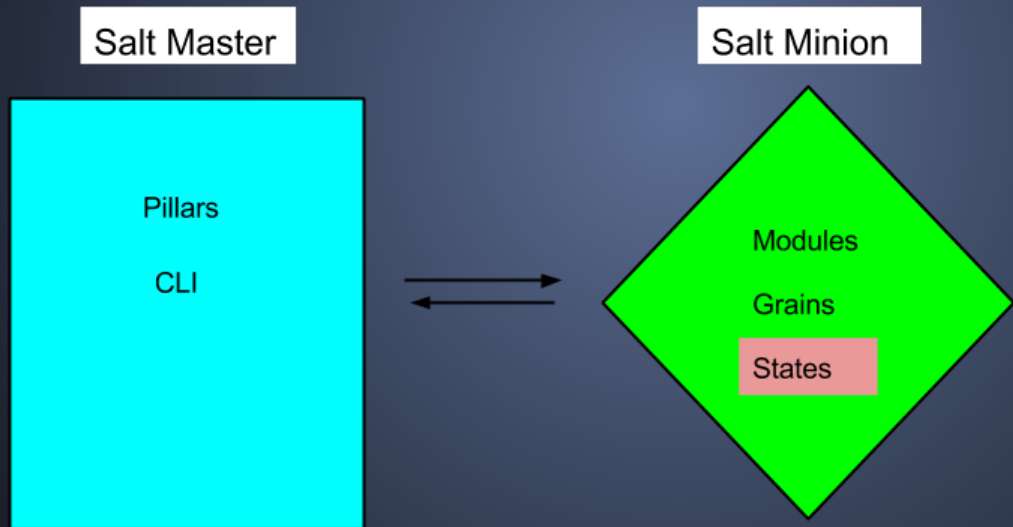
# SaltStack Grains



# salt grains

- key / value pairs
- data is gathered when minion starts up
- standard grains include information about a server
- custom grains are very useful, especially on cloud servers e.g. targeting

# SaltStack States



# salt states

- used for config management
- conceptually a state is how / what you want something to be
- a state is transformed into a run list on the minion
- states = data, usually created by combining jinja + yaml on the minion
- a state is transformed into a run list on the minion
- salt states use execution modules to determine what needs to be done, and to perform the actions
- states return an expected (python) dictionary back to the master showing what changed, what failed and etc.



# Other Components and Additions

- reactors
- salt-cloud
- engines
- beacons
- scheduler
- proxy minion
- multi-master

# Next Meeting Topic?

- salt agentless / salt-ssh in depth
- salt cloud
- orchestration of servers - how to do it with examples
- writing custom grains
- writing custom execution modules
- writing custom states
- salt-mine
- the salt reactor -- when + why + how
- salt formulas - what, how, why
- best practices for pillars, states, what to back up, key mgmt and etc
- running saltstack in a highly available / fault tolerant way

# Reference Links:

General Information: <http://docs.saltstack.com/en/latest/>

Full list of execution modules: <http://docs.saltstack.com/en/latest/ref/modules/all/index.html>

Full list of state modules: <http://docs.saltstack.com/en/latest/ref/states/all/>

Phoenix - SaltStack HA in AWS: <https://github.com/wcannon/saltconf2015.git>

List of returners: <http://docs.saltstack.com/en/latest/ref/returners/>

Salt over SSH: <http://docs.saltstack.com/en/latest/topics/ssh/>

Thanks for attending the “initial  
commit”