The Saltstack Salt Stack

Presentation to RVALug • May 16, 2015

David Elkins • elkinsd@gmail.com

github.com/elkinsd/rvalug-salt

About me

I still love Slackware

Using Salt for about 2 years

20+ years experience with Linux administration, longtime Debian Consultant

VP R&D for Pica9, Inc. in New York City

Make frequent use of a mix of technologies, AMA:

- Perl, PDF, Pica9 Docquery, Elixir, Erlang/OTP
- Apache HTTPD, CouchDB, HTML/CSS
- JavaScript, Python, PHP, C
- PostgreSQL, MySQL, ElasticSearch
- Salt/ZeroMQ, Nagios, VMWare
- OSX and Linux: Slackware, Debian, Centos
- Imagemagick, GhostScript, XeTeX, FontLab

TOC

What is Salt?

Salt is a *configuration management* system which can be used to maintain remote nodes in defined states (ie, ensure package installed or specific service running)

Salt is a *distributed remote execution* system which can execute commands and query data on remote nodes, individually or by simple/complex specifications

What is Salt?

Salt provides secure parallel execution via

ZeroMQ / AES / msgpack

Salt supports Python / YAML natively with

API support for languages, and

XML/JSON for data format

How to install Salt

Package management, Debian:

sources: `deb http://debian.saltstack.com/debian jessie-saltstack main`

```
$ wget -q -O-
    "http://debian.saltstack.com/debian-salt-team-joehealy.gpg.key" |
    apt-key add -
$ apt-get update
$ apt-get install salt-master
$ apt-get install salt-minion
$ apt-get install salt-syndic
```

How to install Salt

Package management, CentOS:

```
$ rpm -Uvh epel-release-X-Y.rpm
$ yum install salt-master
$ yum install salt-minion
$ chkconfig salt-master on
$ service salt-master start
$ chkconfig salt-minion on
$ service salt-minion start
```

ie, epel-release-6-8.rpm for 6.8

How to install Salt

Where else does Salt work?

Arch, Fedora, FreeBSD, OSX, Ubuntu,

Solaris, Windows, Gentoo, and someday Slackware..

See Instructions and dependencies etc:

http://docs.saltstack.com/en/latest/topics/installation/index.html

How to find Salt Answers

Check out the documentation: http://docs.saltstack.com/en/latest/contents.html

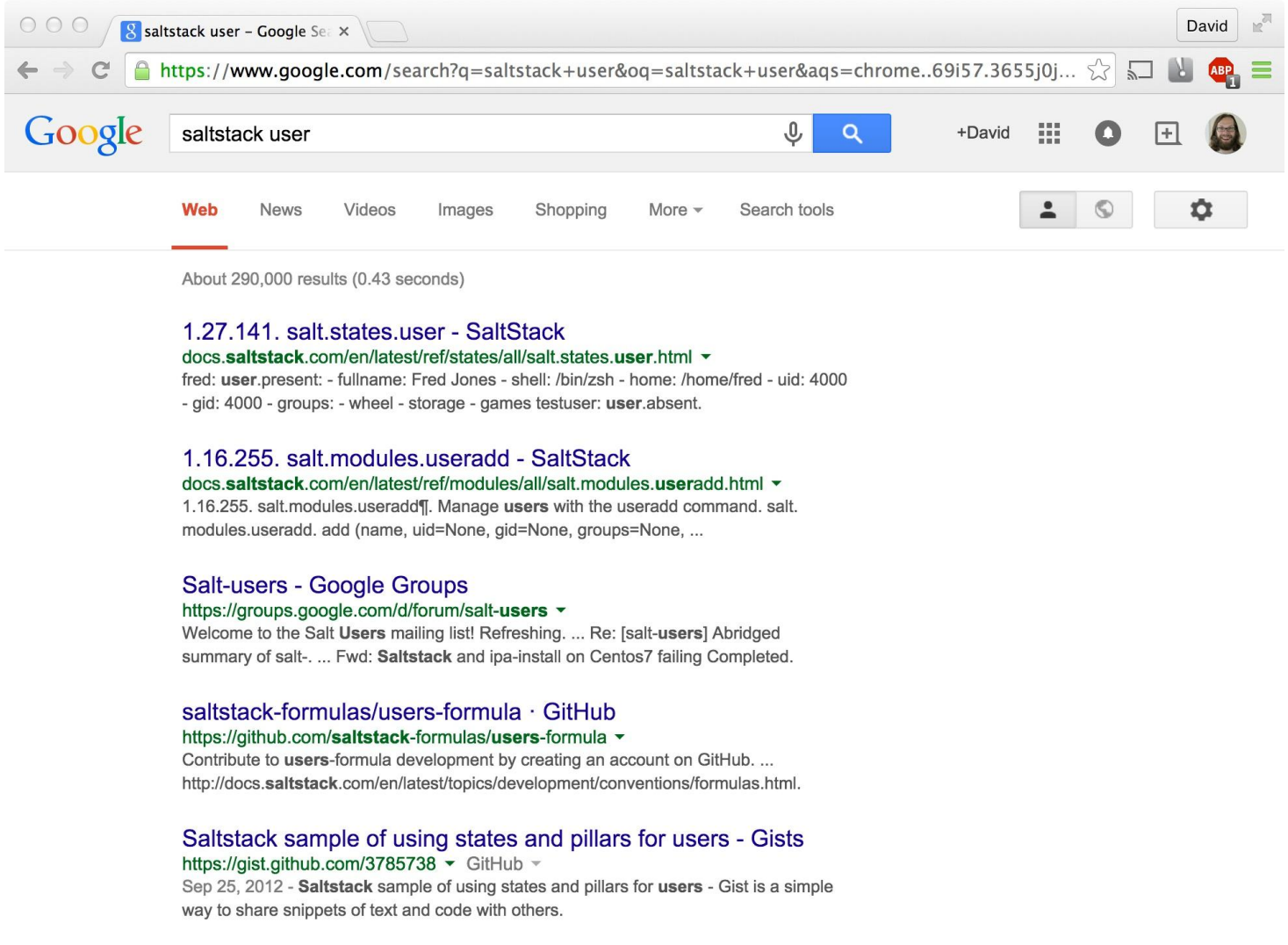
Check out the code: https://github.com/saltstack/salt

Join the IRC and ask a question!: #salt @ Freenode

Search terms: “saltstack $terms”

Results break down broadly into two categories: “saltstack user”

states	“salt.states.user”
modules	“salt.modules.useradd”


saltstack user - Google Se
David
https://www.google.com/search?q=saltstack+user&oq=saltstack+user&aqs=chrome..69i57.3655j0j...
Google
saltstack user
+David
Web
News
Videos
Images
Shopping
More
Search tools
About 290,000 results (0.43 seconds)
1.27.141. salt.states.user - SaltStack
docs.saltstack.com/en/latest/ref/states/all/salt.states.user.html
fred: user.present: - fullname: Fred Jones - shell: /bin/zsh - home: /home/fred - uid: 4000 - gid: 4000 - groups: - wheel - storage - games testuser: user.absent.
1.16.255. salt.modules.useradd - SaltStack
docs.saltstack.com/en/latest/ref/modules/all/salt.modules.useradd.html
1.16.255. salt.modules.useradd¶. Manage users with the useradd command. salt. modules.useradd. add (name, uid=None, gid=None, groups=None, ...
Salt-users - Google Groups
https://groups.google.com/d/forum/salt-users
Welcome to the Salt Users mailing list! Refreshing. ... Re: [salt-users] Abridged summary of salt-. ... Fwd: Saltstack and ipa-install on Centos7 failing Completed.
saltstack-formulas/users-formula · GitHub
https://github.com/saltstack-formulas/users-formula
Contribute to users-formula development by creating an account on GitHub. ... http://docs.saltstack.com/en/latest/topics/development/conventions/formulas.html.
Saltstack sample of using states and pillars for users - Gists
https://gist.github.com/3785738 GitHub
Sep 25, 2012 - Saltstack sample of using states and pillars for users - Gist is a simple way to share snippets of text and code with others.

Salt Terminology

Master	Salt server
Minion	Salt client
States	Salt state file declarations, modules, functions
Modules	State and execution, virtual and OS-specific
Grain	Static information about minions, from minion
Pillar	Private information about minions, from master
Renderer	Translates data format (YAML) into Salt structure
Returner	Returns command results to external datastore (DB)
Outputter	Formats Salt command output
Runners	Master-only commands executed by `salt-run`

Salt Terminology

Jobs	Command execution tasks, return information
Highstate	Master state from top.sls
Lowstate	Evaluated highstate after handling requisites
Environment	State file directory tree
Jinja	Template engine/language
Salt Mine	Make arbitrary grain data avail to master, all minions
Salt Syndic	Tiered Master forwarder
Proxy Minions	Minion controlling devices that can't run `salt-minion`
Salt Virt	Manages virtual machines, using libvirt
Salt Cloud	Manages virtual machines, using libcloud

Salt Terminology

Events	Notice emitted on actions
Reactor	Interface for event listening and actions
Salt SSH	Use SSH as transport, doesn't use 'salt-minion' etc
Salt Rosters	Flat file list of target hosts (used by salt-ssh only)
Halite	Salt GUI
REST	HTTP REST API

Salt Terminology

Minion Salt client

/etc/salt/minion

/etc/salt/minion_id

/etc/salt/minion.d/grains.conf

```
$ salt-call state.highstate
$ salt-call state.sls statename
$ salt-call state.sls httpd.server
```

Salt Terminology

States Salt state file declarations, modules, functions

/srv/salt/top.sls:

```
  base:
    '*':
      - base
```

```
$ salt-call state.highstate
$ salt-call state.sls statename
$ salt-call state.sls base
```

Salt Terminology

Modules — State and execution modules, some virtual that are implemented by OS-specific functions

```
$ salt '*' test.ping
$ salt '*' disk.percent
$ salt '*' sys.doc
$ salt '*' saltutil.sync_all
$ salt '*' cmd.run “ls -l /etc”
$ salt '*' network.interfaces
$ salt '*' -b 5 pkg.install vim
```

Salt Terminology

Grains Static information about and from minions

/etc/salt/minion.d/grains.conf:

```
  grains:
    node_type: db
    services:
      - ssh
```

```
$ salt-call grains.ls
$ salt-call grains.get nodename
```

Salt Terminology

Pillars Private information about minions from master, may be encrypted.

/srv/pillar/top.sls:

```
  'base'
    - '*'
      - base          -> /srv/pillar/base.sls or /srv/pillar/base/init.sls
      - pkgs          -> /srv/pillar/pkgs.sls or /srv/pillar/pkgs/init.sls
```

```
$ salt-call pillar.items
$ salt-call pillar.get pillarname
```

Salt Terminology

Pillars

pkgs.sls:

```
 pkgs:
    {% if grains.os == 'CentOS' %}
    apache: httpd

$ salt-call pillar.get pkgs

$ salt-call state.highstate pillar='{"foo": "bar"}'
$ salt-call state.sls mystate pillar='["foo","bar"]'
```

Targeting Minions

Minions can be targeted based on minion_id/hostname, grains, pillars..

```
$ salt 'target' <function> [arguments]
```

```
$ salt '*' test.ping
$ salt -G 'os:Debian' state.highstate
$ salt -L 'foo,bar' cmd.run "ls /etc/hosts"
$ salt -E 'web-(1|2|3)*' test.ping
$ salt -C 'hostname and G@os:Debian'
$ salt-call sys.doc
```

Targeting Minions

Minion match types

G	Grains glob	G@os:Ubuntu
E	PCRE Minion ID	E@web-(1|2|3).*
P	Grains PCRE	P@os:(Redhat|CentOS)
L	List of minions	L@minion2,minion2,minion3
I	Pillar glob	I@pdata:foobar
S	Subnet/IP addresses	S@192.168.1.0/24
N	Node group	N@groupname

Targeting Minions

example of matching in state files

```
base:
  'web-(1|2|3)*':
    - match: pcre
    - webserver
```

example of matching against grain data:

```
base:
  'node_type:db':
    - match: grain
    - database
```

Salt States

Salt states — Data that defines state declarations, functions and arguments

```
httpd:
 pkg:
  - installed
```

ID declaration, must be unique

state declaration

function declaration

```
apache:
  pkg.installed:
    -name: {{ salt['pillar.get’](‘pkgs:apache', 'httpd') }}
```

Salt States

httpd/server.sls:

```
apache:
  pkg.installed: []
  service.running:
    - watch:
      - pkg: apache
      - file: /etc/httpd/conf/httpd.conf

/etc/httpd/conf/httpd.conf:
  file.managed:
    - source: salt://apache/httpd.conf
```

Salt States

ssh/init.sls:

```
openssh-client:
  pkg.installed

/etc/ssh/ssh_config:
  file.managed:
    - user: root
    - group: root
    - mode: 644
    - source: salt://ssh/ssh_config
    - require:
      - pkg: openssh-client
```

Salt States

Include: include other state files

ssh/server.sls:

```
include:
  - ssh                    -> includes ssh/init.sls

sshd:
  service.running: []

/etc/ssh/banner:
  file.managed:
    - source: salt://ssh/banner
```

Salt States

Extend: overwrite state data

ssh/custom-server.sls:

```
include:
  - ssh.server
extend:
  /etc/ssh/banner:
    file:
      - source: salt://ssh/custom-banner
  sshd:
    service:
      - watch:
        - file: /etc/ssh/banner
```

Salt States

Environments: merge multiple file_roots, top-most match “wins”

/etc/salt/master:

```
file_roots:
  base:
    - /srv/salt
  dev:
    - /srv/salt-dev
    - /srv/salt
```

Salt States

Official example salt states:

https://github.com/SS-archive/salt-states

Community example salt states:

http://docs.saltstack.com/en/latest/topics/index.html#example-salt-states

Salt Modules

Modules State modules versus execution modules

```
salt.state.pkg
 =>
  salt.module.pkg
   =>
    salt.modules.aptpkg
    salt.modules.brew
    salt.modules.freebsdpkg
    salt.modules.pacman
    salt.modules.win_pkg
    salt.modules.yumpkg
```

Salt Modules

Examples of some of the available salt.state.* and salt.modules.*:

aliases alternatives apache archive at cloud cmd config cp cron disk dnsutil file	git grains hosts htpasswd iptables locate logrotate match mine mount network nginx pillar	ps postgres quota rsync saltutil selinux service shadow ssh state status system test virt

Renderers and Templates

yaml_jinja

salt, grains, pillars in {% %} functions, context/variables in {{ }}

```
apache:
  pkg.installed:
    {% if grains.os == 'RedHat' %}
    - name: httpd
    {% endif %}
  service.running:
    {% if grains.os == 'RedHat' %}
    - name: httpd
    {% endif %}
```

Renderers and Templates

yaml_jinja

```
{% salt['pillar.get']('keyname','') %}

{% for key,value in salt['pillar.get']('keyname',{}).items() %}
{{ key }}
{% endfor %}

{% set name = salt['pillar.get']('keyname','altval') %}
{{ name }}
```

Renderers and Templates

gpg

encrypted pillar data, requires master have 'python-gnupg' package

```
$ mkdir /etc/salt/gpgkeys
$ gpg --gen-key --homedir /etc/salt/gpgkeys

$ echo -n “supersecret” | gpg --homedir /etc/salt/gpgkeys
    --armor --encrypt -r <KEYNAME>

$ gpg --homedir --armor --encrypt -r <KEYNAME> FILENAME
```

Renderers and Templates

gpg

/srv/pillar/secrets.sls:

```
#!yaml|gpg
a-secret: |
  -----BEGIN PGP MESSAGE-----
  Version: GnuPG v1.4.12 (GNU/Linux)
  ...
  -----END PGP MESSAGE-----
```

Jinja: `{{ salt['pillar.get']('a-secret','') }}`

Salt Returners

Salt returners — Python functions that can send the return data to databases, etc, in addition to the normal shell/stdout return

```
$ salt-call saltutil.sync_returners
$ salt-call test.ping --return cdb
$ salt '*' state.highstate --return cdb
$ salt-call --local --metadata test.ping --out=pprint
```

Salt Returners

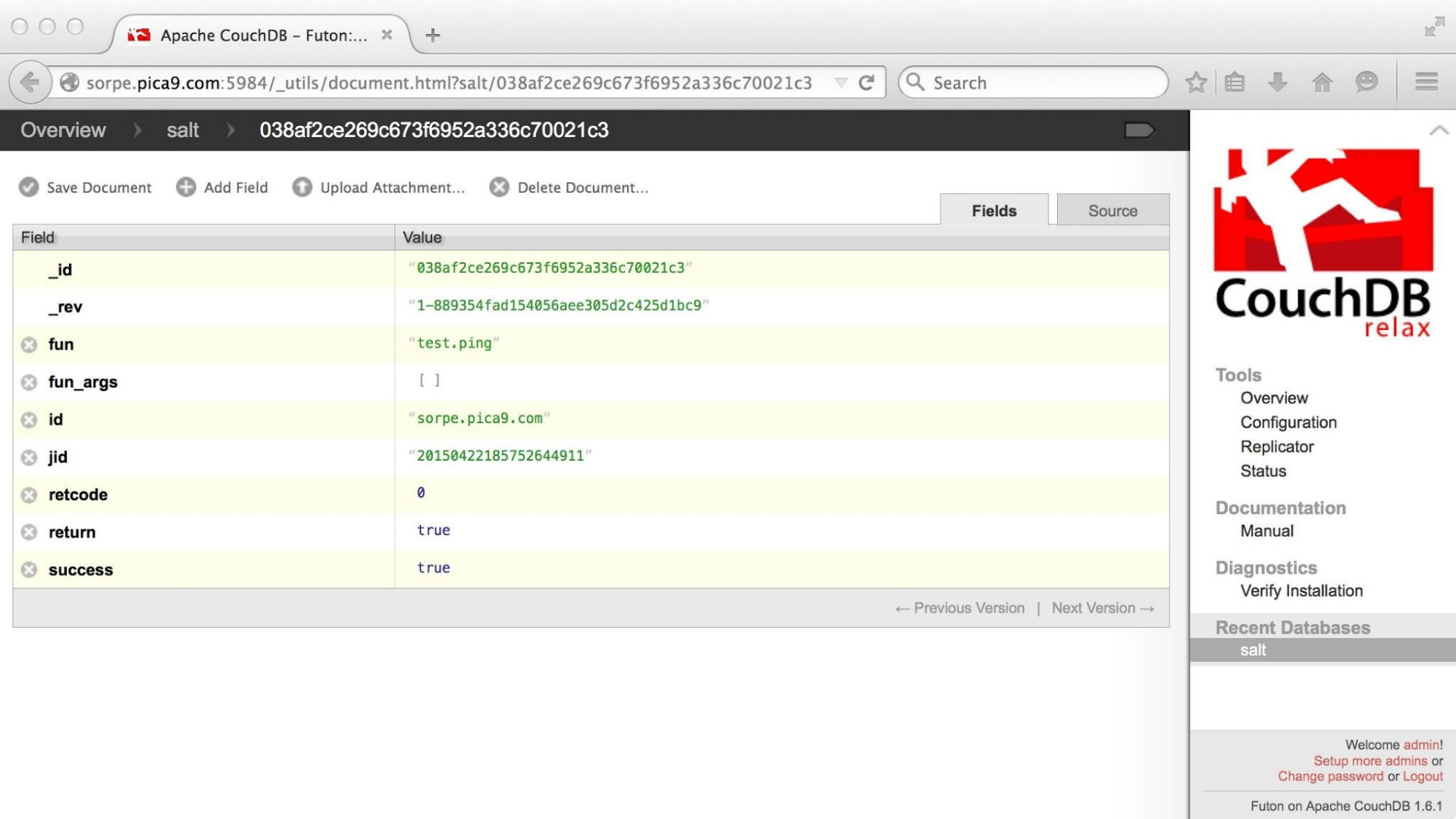
Example returner, sending data as JSON to remote CouchDB:

/srv/salt/_returners/cdb.py:

```
import couchdb
def returner(ret):
  couchdb_url = 'http://localhost:5984'
  couch = couchdb.Server(couchdb_url)
  couch.resource.credentials = ('user','pass')
  db = couch['salt']
  doc_id, doc_rev = db.save(ret)
```

$ salt-call test.ping --return cdb

```
local:
  True
```


Apache CouchDB – Futon:...
sorpe.pica9.com:5984/_utils/document.html?salt/038af2ce269c673f6952a336c70021c3
Search
Overview
salt
038af2ce269c673f6952a336c70021c3
Save Document
Add Field
Upload Attachment...
Delete Document...
Fields
Source
Field
Value
_id
"038af2ce269c673f6952a336c70021c3"
_rev
"1-889354fad154056aee305d2c425d1bc9"
fun
"test.ping"
fun_args
[ ]
id
"sorpe.pica9.com"
jid
"20150422185752644911"
retcode
0
return
true
success
true
← Previous Version | Next Version →
CouchDB
relax
Tools
Overview
Configuration
Replicator
Status
Documentation
Manual
Diagnostics
Verify Installation
Recent Databases
salt
Welcome admin!
Setup more admins or
Change password or Logout
Futon on Apache CouchDB 1.6.1

Salt Terminology

Master Salt server

/etc/salt/master

/etc/salt/master.d

```
$ salt-key -L
$ salt-key -a hostname.example.com
$ salt '*' test.ping
$ salt 'hostname*' test.ping
```