Salt

Configuration management

What is it for?

- Configuring lots of machines
- Doing general sysadmin on machines

Two parts

States

- Specify the configuration of the boxes

Modules

- Do things on the boxes.
- Run commands, check ip address
- Shutdown!

Network

Uses ZeroMQ encrypted with AES

Either setup key on box when installing, or acknowledge key on server.

Can now use ssh with no client

Master and minion

Server and client

Getting started on ec2

Use cloudinit - paste it in when you're creating the minion

```
#cloud-config
# minion
apt_sources:
  - source: "ppa:saltstack/salt"
apt_update: true
apt_upgrade: true

packages:
  - salt-minion
ssh_authorized_keys:
  - ssh-rsa AAAAAAAAAAAAAAAAAADSADSADSADSADSA....
salt_minion:
  conf:
    master: ec2-....compute.amazonaws.com
```

Now get them talking

Use `salt-key` to manage the keys

Install something

Select which states to install in `top.sls`

```
salt machine-name state.highstate
```

The anatomy of a state file

```
apache:                    # id declaration
  pkg:                     # state declaration
    - installed            # functions to run
  service:                 # state declaration
    - running              # functions to run
    - require:             # requisite statement
      - pkg: apache
```

Jinja templates

```
upstream noentropy {
    server localhost:{{ pillar['noentropy']['port'] }};
}

server {
    listen 0.0.0.0:{{ pillar['noentropy']['external-port'] }};
    server_name noentropy;
    location / {
        proxy_pass http://noentropy/;
        proxy_set_header X-Forwarded-Host $http_host;
        proxy_set_header X-Forwarded-Port $http_port;
        proxy_set_header X-Forwarded-For  $proxy_add_x_forwarded_for;
    }
```

Data

- Pillar - data held on the master in parallel .sls files
- Grains - data about each machine
- Grains can be set manually
- Pillar data can be passed in via the command line too

Debugging

- Crank up the logging
- salt-call
- Use python to dump the yaml/interpret the jinja

The catch all

- cmd.run and cmd.wait allow you to fill in the gaps
- Install perl modules with cpanm like this
- Crude but effective
- Be careful to keep it idempotent

Doing things

Useful commands

```
salt '*' test.version
salt '*' system.halt
salt '*' sys.doc
salt '*' network.interfaces # ip_addrs often useful
salt '*' pillar.items
```

Gotchas

Ensure all ids are unique. Use name to help avoid duplicates

The reporting of missing jinja variables can be inaccurate if you have more than 1 variable in use.

Learning Salt

The documentation is pretty good

As you go through the tutorial look at the corresponding modules in the list of all the states - http://docs.saltstack.com/ref/states/all/

Summary

- Saves a lot of time
- Still a bit rough around the edges

Alternatives

- Puppet - ruby + dsl + ssl <- most established
- Ansible - python + yaml + ssh, no client
- Chef - ruby + ruby