Chef vs. Salt

Martin Neiiendam

January 13, 2016

- Martin Neiiendam
- Lokalebasen.dk
- Chef vs. SaltStack

Who are you?

Who are you? - Experience?

Experience

- Linux/Unix?

Experience

- Linux/Unix?
- Chef?

Experience

- Linux/Unix?
- Chef?
- SaltStack?

Experience

- Linux/Unix?
- Chef?
- SaltStack?
- Ansible?

Mixed

- Chef/SaltStack/Ansible?

Mixed

- Chef/SaltStack/Ansible?
- Other experience?

Meet Chef

Meet Chef

Meet Chef

- Configuration Management

Meet Chef

- Configuration Management
- HTTP(S) Client/Server based

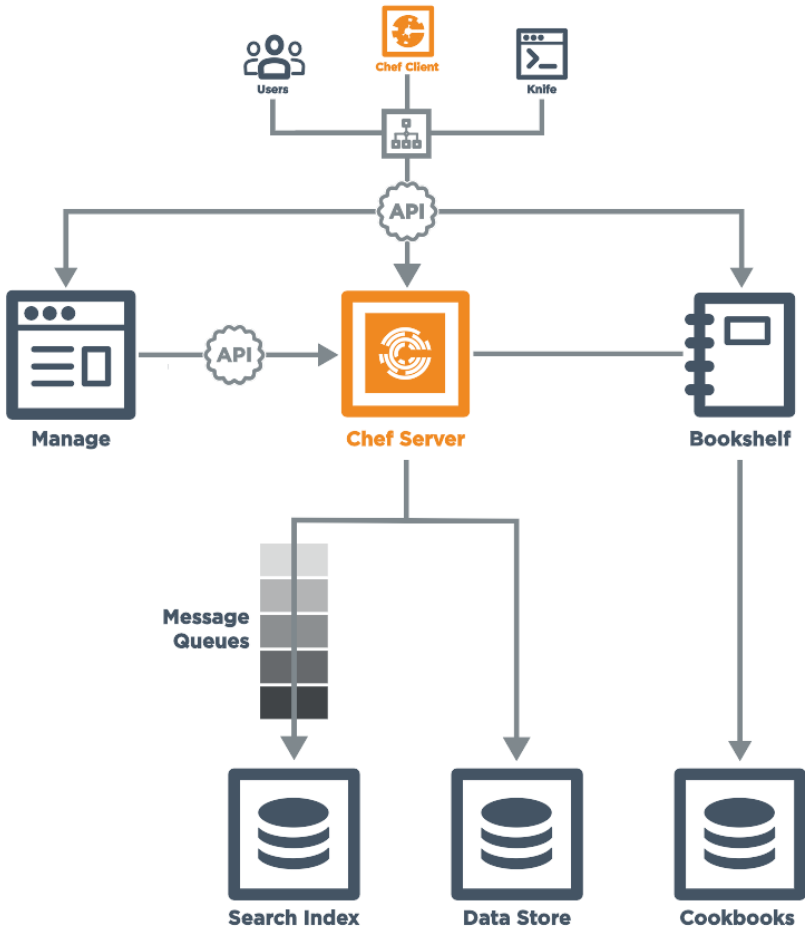
Meet Chef

- Configuration Management
- HTTP(S) Client/Server based
- Erlang Server

Meet Chef

- Configuration Management
- HTTP(S) Client/Server based
- Erlang Server
- Ruby client

Chef Infrastructure



Learning Chef

- Start with Chef Solo/Zero

Learning Chef

- Start with Chef Solo/Zero
- Ruby DSL

Learning Chef

- Start with Chef Solo/Zero
- Ruby DSL

```
1  template "/etc/nginx/sites-available/site.conf" do
2    source 'site.conf.erb'
3    mode '644'
4    user 'root'
5    group 'root'
6    variables(host: node.name)
7    notifies :reload, resources(service: 'nginx'), :delayed
8  end
9
```

Chef Basic Concepts

- Cookbooks

Chef Basic Concepts

- Cookbooks
- Recipies

Chef Basic Concepts

- Cookbooks
- Recipies
- Nodes

Chef Basic Concepts

- Cookbooks
- Recipies
- Nodes
- Clients

Chef Basic Concepts

- Cookbooks
- Recipies
- Nodes
- Clients
- Roles

Chef Basic Concepts

- Cookbooks
- Recipies
- Nodes
- Clients
- Roles
- Environments

Setting up Chef

- Standalone/Hosted

Setting up Chef

- Standalone/Hosted
- HTTPS - Certificate

Setting up Chef

- Standalone/Hosted
- HTTPS - Certificate
- CLI tools serverside

Setting up Chef

- Standalone/Hosted
- HTTPS - Certificate
- CLI tools serverside
- CLI tools client side

Bootstrapping a node

- Install embedded ruby

Bootstrapping a node

- Install embedded ruby
- Install chef (gem)

Bootstrapping a node

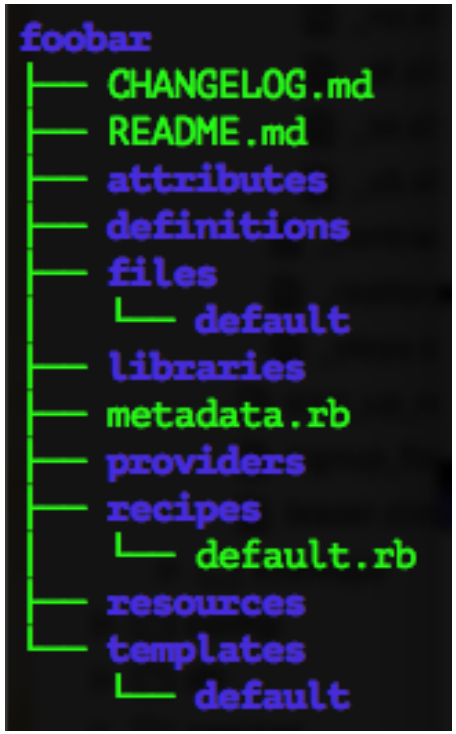
- Install embedded ruby
- Install chef (gem)
- Set up credentials

Bootstrapping a node

- Install embedded ruby
- Install chef (gem)
- Set up credentials
- Run chef-client first time

Cookbook structure

- knife cookbook create foobar



Applying a cookbook

- knife cookbook upload foobar
- knife node run_list add foobar.example.com 'recipe[foobar]'
- Wait for chef-client to run - defaults to every 30 minutes

Community Resources

- supermarket.chef.io
- Berkshelf
- Open Source chef cookbooks (Facebook, Riot Games, etc.)
- "Chef Style DevOps Kungfu"

Overall thoughts

- Rather complex

Overall thoughts

- Rather complex
- Convention over Configuration

Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary

Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary
- "Round Trip Time" is rather large

Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary
- "Round Trip Time" is rather large
- Well tested

Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary
- "Round Trip Time" is rather large
- Well tested
- Test Kitchen

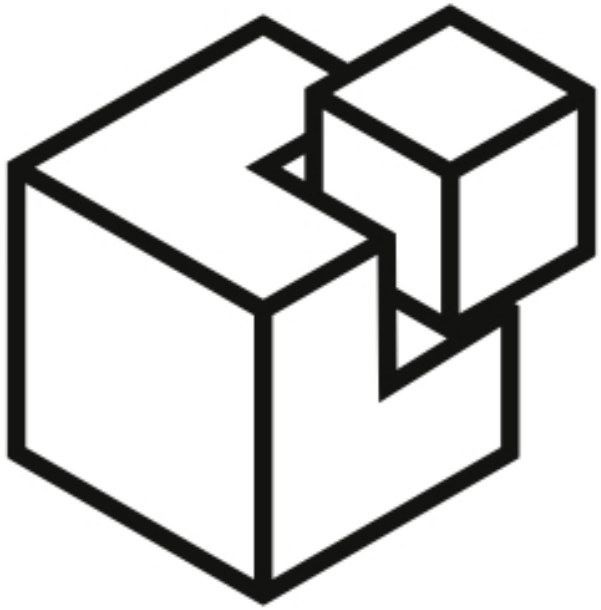
Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary
- "Round Trip Time" is rather large
- Well tested
- Test Kitchen
- Proven in large setups (30k+ nodes)

Overall thoughts

- Rather complex
- Convention over Configuration
- Conventions are necessary
- "Round Trip Time" is rather large
- Well tested
- Test Kitchen
- Proven in large setups (30k+ nodes)
- Scaling is a "Black art"

SaltStack



SALTSTACK

Buzzwords

- ZeroMQ

Buzzwords

- ZeroMQ
- Event Based

Buzzwords

- ZeroMQ
- Event Based
- Python

Buzzwords

- ZeroMQ
- Event Based
- Python
- Remote Execution

The taste of Salt

The taste of Salt

Core Concepts

- Master/Minion/Syndic

Core Concepts

- Master/Minion/Syndic
- PKI

Core Concepts

- Master/Minion/Syndic
- PKI
- States

Core Concepts

- Master/Minion/Syndic
- PKI
- States
- Grains

Core Concepts

- Master/Minion/Syndic
- PKI
- States
- Grains
- Pillars

Core Concepts

- Master/Minion/Syndic
- PKI
- States
- Grains
- Pillars
- Beacons/Reactors

Core Concepts

- Master/Minion/Syndic
- PKI
- States
- Grains
- Pillars
- Beacons/Reactors
- Returners/Outputters

Learning Salt

- Master: Install 'salt-master'

Learning Salt

- Master: Install 'salt-master'
- Minion: Install 'salt-minion'

Learning Salt

- Master: Install 'salt-master'
- Minion: Install 'salt-minion'
- Minion: Point to master ip

Learning Salt

- Master: Install 'salt-master'
- Minion: Install 'salt-minion'
- Minion: Point to master ip
- Minion: Restart 'salt-minion'

Learning Salt

- Master: Install 'salt-master'
- Minion: Install 'salt-minion'
- Minion: Point to master ip
- Minion: Restart 'salt-minion'
- Master: Accept minion with 'salt-key'

Ping all

```
[15:31:33] fracklen@cm:~ $ sudo salt '*' test.ping
dev-right:
    True
kitchen:
    True
sales-left:
    True
dev-left:
    True
sales-right:
    True
```

Instant execution

Instant execution

Instant execution

```
[15:31:45] fracklen@cm:~ $ sudo salt '*' cmd.run 'whoami'
dev-right:
    root
kitchen:
    root
dev-left:
    root
sales-right:
    root
sales-left:
    root
```

Instant execution - Dev perspective

Switch perspective

Lets look at this from the Ops perspective

Instant execution - Ops perspective

- sudo salt '*' cmd.run 'rm -rf /'

Instant execution - Ops perspective

- sudo salt '*' cmd.run 'rm -rf /'
- sudo salt '*' cmd.run 'poweroff -f'

Instant execution - Ops perspective

Security

Secure the Salt Master!

Security

- M&M's Security

Security

- M&M's Security
- Protect your master - at all cost

Security

- M&M's Security
- Protect your master - at all cost
- Disallow 'cmd.run' if not needed

Top file

- Separates environments

Top file

- Separates environments
- Acts as an overview

Top file

- Separates environments
- Acts as an overview
- Targets minions

Top file

- Separates environments
- Acts as an overview
- Targets minions
- Defines which states apply

Top file

```
[16:25:45] frackle@cm:/srv/salt $ cat top.sls
base:
  '*':
    - dashboard
[16:25:51] frackle@cm:/srv/salt $ ls
dashboard  dashboard.sls  top.sls
```

State file

- Defines the content of the state

State file

- Defines the content of the state
- References a list of execution modules

State file

- Defines the content of the state
- References a list of execution modules
- YAML defines name, module and parameters

State file

```
[16:31:52] fracklen@cm:/srv/salt $ cat dashboard.sls

/etc/lightdm/lightdm.conf.d:
  file.directory:
    - user: root
    - group: root
    - makedirs: True
    - mode: 755

/etc/lightdm/lightdm.conf.d/50-login.conf:
  file.managed:
    - source: salt://dashboard/autologin.conf
    - user: user
    - group: user
    - template: jinja

/home/user/.config/autostart:
  file.directory:
    - user: user
    - group: user
    - makedirs: True
    - mode: 755
```

Using pillars

- Allows use of confidential data in states

Using pillars

- Allows use of confidential data in states
- Not searchable

Using pillars

- Allows use of confidential data in states
- Not searchable
- Not targetable

Using pillars

- Allows use of confidential data in states
- Not searchable
- Not targetable
- Only pushed to the relevant minions

Using pillars

```
/home/user/.config/autostart/chromium.desktop:
  file.managed:
    - source: salt://dashboard/chromium.desktop
    - template: jinja
    - user: user
    - group: user
    - context:
      url: {{ pillar['dashboard_url'] }}
```

```
[16:31:56] fracklen@cm:/srv/salt $ cat dashboard/chromium.desktop
[Desktop Entry]
Type=Application
Exec=google-chrome --kiosk '{{ url }}'
Hidden=false
NoDisplay=false
X-GNOME-Autostart-enabled=true
Name[C]=Geckoboard dashboard
Name=chrome
Comment[C]=
Comment=
```

Grains vs Pillars

- Grains are 'public'

Grains vs Pillars

- Grains are 'public'
- Pillars are 'private'

Grains vs Pillars

- Grains are 'public'
- Pillars are 'private'
- Grains can be targeted

Grains vs Pillars

- Grains are 'public'
- Pillars are 'private'
- Grains can be targeted
  - salt -G 'os:CentOS' system.reboot

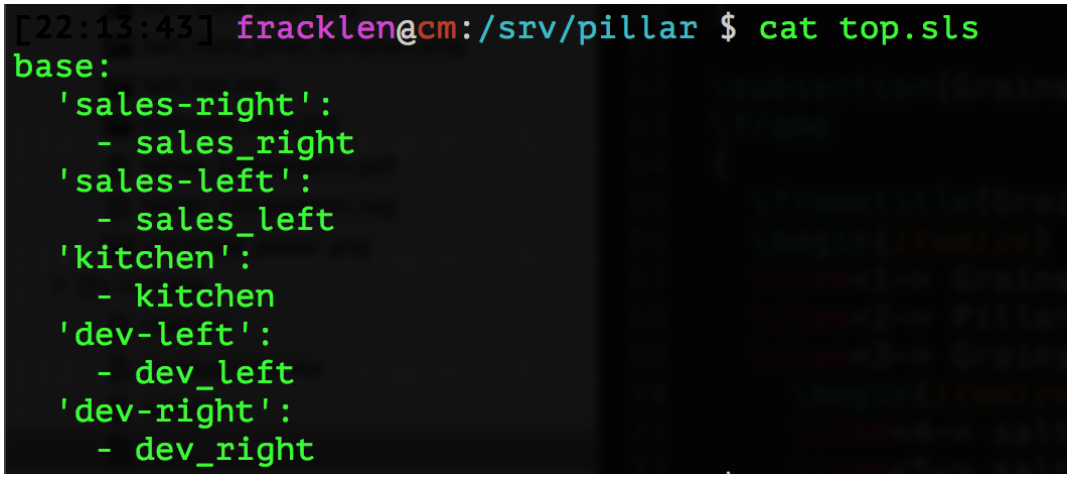
Grains vs Pillars

- Grains are 'public'
- Pillars are 'private'
- Grains can be targeted
  - salt -G 'os:CentOS' system.reboot
  - salt -G 'num_cpus:32' cmd.run 'some-task'

Grains vs Pillars

- Grains are 'public'
- Pillars are 'private'
- Grains can be targeted
  - salt -G 'os:CentOS' system.reboot
  - salt -G 'num_cpus:32' cmd.run 'some-task'
  - salt '*' grains.ls

Defining pillars



Defining pillars

```
[22:20:17] fracklen@cm:/srv/pillar $ cat test.sls
dashboard_url: https://some.secret.url/
```

Pros and Cons

- Testing story

Pros and Cons

- Testing story
- Provisioning

Pros and Cons

- Testing story
- Provisioning
- React to changes

Pros and Cons

- Testing story
- Provisioning
- React to changes
- Ruby vs Python

Pros and Cons

- Chef 60 builtin resources

Pros and Cons

- Chef 60 builtin resources
- SaltStack 330 builtin modules

Pros and Cons

- Chef 60 builtin resources
- SaltStack 330 builtin modules
- Chef Community Cookbooks

Pros and Cons

- Chef 60 builtin resources
- SaltStack 330 builtin modules
- Chef Community Cookbooks
- SaltStack zipped modules

Pros and Cons

- Chef 60 builtin resources
- SaltStack 330 builtin modules
- Chef Community Cookbooks
- SaltStack zipped modules
- Write your own recipe/module

Why the mix?

Why mix Chef and SaltStack?

Why the mix?

- Instant remote execution

Why the mix?

- Instant remote execution
- *knife ssh 'chef_environment:staging' 'echo foobar'*

Why the mix?

- Instant remote execution
- *knife ssh 'chef_environment:staging' 'echo foobar'*
  - Doesn't scale well

Why the mix?

- Instant remote execution
- *knife ssh 'chef_environment:staging' 'echo foobar'*
  - Doesn't scale well
  - Requires ssh access

Why the mix?

- Instant remote execution
- *knife ssh 'chef_environment:staging' 'echo foobar'*
  - Doesn't scale well
  - Requires ssh access
  - Ssh using user or root?

Why the mix?

- Instant remote execution
- *knife ssh 'chef_environment:staging' 'echo foobar'*
  - Doesn't scale well
  - Requires ssh access
  - Ssh using user or root?
- Chef Push Jobs?

Chef Push Jobs

The pushy Chef

Chef Push Jobs

Chef Push Jobs

- Separate Cookbook

Chef Push Jobs

- Separate Cookbook
- Two new ports (10000 and 10003) on Chef Server

Chef Push Jobs

- Separate Cookbook
- Two new ports (10000 and 10003) on Chef Server
- Needs a plugin on the Chef Server

Chef Push Jobs

- Separate Cookbook
- Two new ports (10000 and 10003) on Chef Server
- Needs a plugin on the Chef Server
- Uses ZeroMQ

Chef Push Jobs

- Separate Cookbook
- Two new ports (10000 and 10003) on Chef Server
- Needs a plugin on the Chef Server
- Uses ZeroMQ
- Doesn't feel as elegant as Salt

Both sides are open

Mix and match

- salt.modules.chef

Mix and match

- salt.modules.chef
- https://supermarket.chef.io/cookbooks/salt

Mix and match

- salt.modules.chef
- https://supermarket.chef.io/cookbooks/salt
- Transition phase

Mix and match

- salt.modules.chef
- https://supermarket.chef.io/cookbooks/salt
- Transition phase
- Test Kitchen vs Python Unit Tests

Salt and Chef

Discussion

Salt and Chef

- Can you cook without Salt?

Salt and Chef

- Can you cook without Salt?
- Is Salt is needed in a Chef kitchen?

Salt and Chef

- Can you cook without Salt?
- Is Salt is needed in a Chef kitchen?
- Is Chef obsolete?

Pros and Cons

- Both are self sufficient

Pros and Cons

- Both are self sufficient
- Ruby vs Python

Pros and Cons

- Both are self sufficient
- Ruby vs Python
- YAML vs Ruby DSL

Pros and Cons

- Both are self sufficient
- Ruby vs Python
- YAML vs Ruby DSL
- Maturity

Pros and Cons

- Both are self sufficient
- Ruby vs Python
- YAML vs Ruby DSL
- Maturity
- Community

Pros and Cons

- Both are self sufficient
- Ruby vs Python
- YAML vs Ruby DSL
- Maturity
- Community
- ???