Infrastructure as Code with Examples using Chef and Ansible

Michael Bopp
@mwbopp

Configuration Management

- **How I got to now?**
- **Examples in Chef**
- **Examples in Ansible**

Learning the Ropes

- **Geocities?**
- **Accessible and easy to get started**
- **Super limited in capabilities**

Small Man in a Big Company

- “Production Team” and “Development Team”
- Server Configuration via scripts and manual incremental changes
- The idea of a “Golden Master” instance
- Opinionated Sys Admins
- Coding to fit the server

Web Development Contractor
using inexpensive solutions

- **Leveraging web hosting services using a configuration web app**
- **High Risk, and potentially Low Availability**
- **Manual Configuration and server setup**
  - **Not very repeatable, and not very fast**
- **Server Configuration is a moving target**
- **Live Instance doesn’t match development, or testing environment**

Repeatable Processes and Standards

- **Manually establishing a set of standards, enforced via documentation**
- **Using a single server for multiple, similar sites**
- **VCS Hooks**
- **Still not “High Availability”**

Site per Server

- **Dedicated Resources**
- **Somewhat Scalable**
- **One server that has just what it needs to host a website.**
- **The Problem: There is a lot of configuration, with lots of opportunities for failure.**

The Solution
Configuration as Code

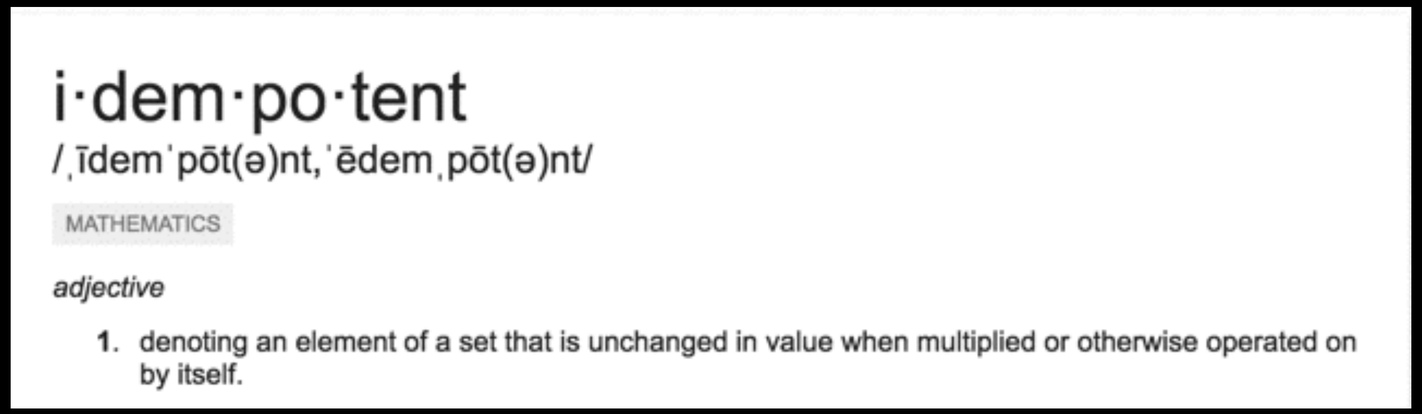
- **CFEngine**

- **Puppet**

- **Chef**

- **Ansible**

Idempotence



i·dem·po·tent

/ˌīdemˈpōt(ə)nt,ˈēdemˌpōt(ə)nt/

MATHEMATICS

adjective

1. denoting an element of a set that is unchanged in value when multiplied or otherwise operated on by itself.

- Will only change if needs to
- Declarative
- Can run over and over again

“Idempotence is the ability to run an operation which produces the same result whether run once or multiple times”

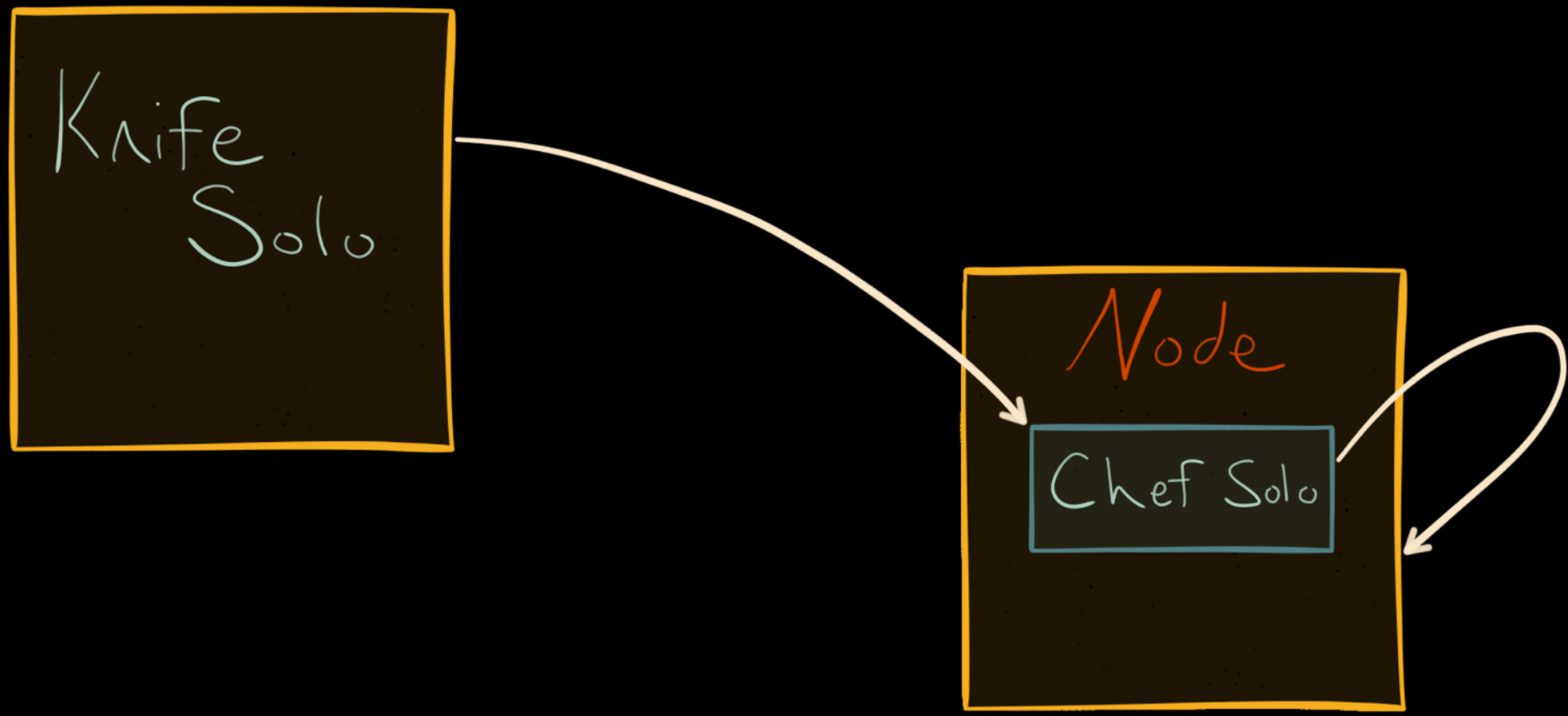
~ Excerpt From: Jeff Geerling. “Ansible for DevOps.” iBooks.

Using Chef

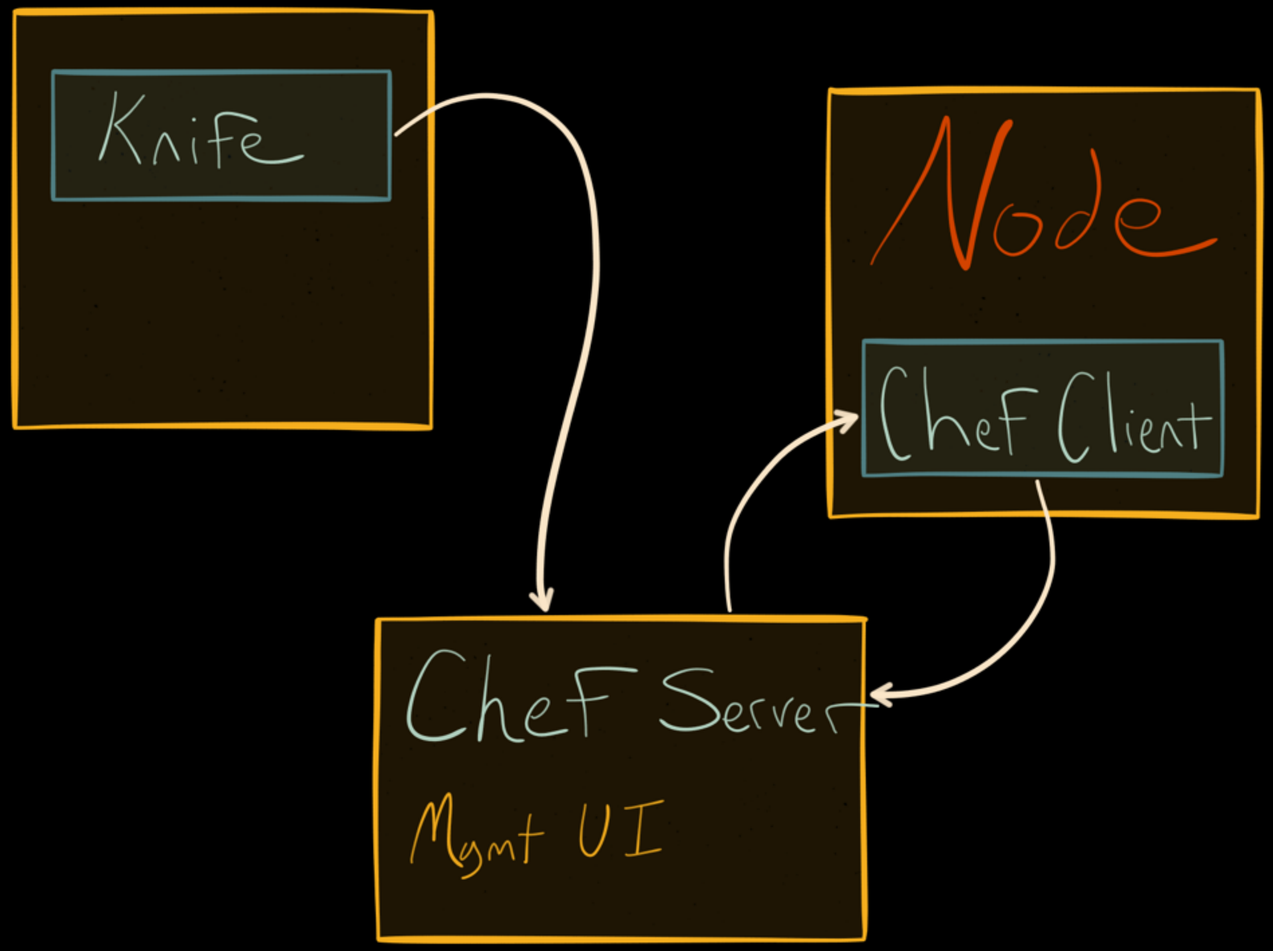
- Has open-source edition or pai
  “hosted” solution
- Hosted solution has a Web UI
- Can be done in either a **server**
  configuration or using **chef-solo**

Chef Patterns

Chef Patterns



Chef Patterns



Step 1 - Install

```
root@ubuntu:~# apt-get install curl
```

```
root@ubuntu:~# curl -L https://www.opscode.com/chef/install.sh | bash
```

```
root@ubuntu:~# chef-solo -v
 Chef: 12.11.18
```

Step 2 - File Structure

```
root@ubuntu:~# wget http://github.com/opscode/chef-repo/tarball/master
```

```
root@ubuntu:~# tar xzvf master
root@ubuntu:~# mv chef-chef-repo* chef
```

```
chefignore
cookbooks
data_bags
environments
LICENSE
README.md
roles
```

```
root@ubuntu:~/chef# mkdir .chef
root@ubuntu:~/chef# echo "cookbook_path [ '/root/chef/cookbooks' ]" >
.chef/knife.rb
```

@mwbopp

Step 3 - Custom Cookbook

```
root@ubuntu:~/chef# knife cookbook create demo-cookbook
```

```
attributes
CHANGELOG.md
definitions
files
libraries
metadata.rb
providers
README.md
recipes
  default.rb
resources
templates
```

Where our contributed cookbook dependencies need to be listed

Our custom instructions

Step 4 - Get Cookbooks

https://supermarket.chef.io/cookbooks

```
root@ubuntu:~/chef/cookbooks# knife cookbook site download apt
Downloading apt from Supermarket at version 4.0.1 to /root/chef/cookbooks/
apt-4.0.1.tar.gz
Cookbook saved: /root/chef/cookbooks/apt-4.0.1.tar.gz
root@ubuntu:~/chef/cookbooks# tar xzf apt-4.0.1.tar.gz
root@ubuntu:~/chef/cookbooks# rm apt-*.gz
```

```
root@ubuntu:~/chef/cookbooks# ls
apt  demo-cookbook  README.md
```

Step 5 - Config

solo.rb

```
file_cache_path "/root/chef-solo"
cookbook_path "/root/chef/cookbooks"
```

web.json

```
{
  "run_list": [ “recipe[apt]”, “recipe[demo-cookbook]” ]
}
```

Step 6 - Execute!

```
root@ubuntu:~/chef# chef-solo -c solo.rb -j web.json
```

Apache

- Add `depends 'httpd', '~> 0.3.5'` to cookbook `metadata.rb`

Install Apache

```
httpd_service 'demo' do
  mpm 'prefork'
  action [:create, :start]
end
```

Create Web Directory

```
directory ‘/var/www/vhosts/demo/' do
  recursive true
end
```

Create Site

```
httpd_config 'demo' do
  instance 'demo'
  source 'demo.conf.erb'
  notifies :restart, 'httpd_service[demo]'
end
```

LWRPs

Lightweight Resources & Providers

@mwbopp

MySQL

Add Client

```
mysql_client 'default' do
  action :create
end
```

Add Service

```
mysql_service 'default' do
  initial_root_password ‘initial_root_passwd’
  action [:create, :start]
end
```

Create Database

```
# Create the database instance.
mysql_database ‘chef_demo_db’ do
  connection(
    :host => '127.0.0.1',
    :username => 'root',
    :password => 'mysql_root_password'
  )
  action :create
end
```

Add User

```
mysql_database_user 'db_admin' do
  connection(
    :host => '127.0.0.1',
    :username => 'root',
    :password => 'mysql_root_password'
  )
  password 'mysql_admin_password'
  database_name ‘chef_demo_db’
  host '127.0.0.1'
  action [:create, :grant]
end
```

@mwbopp

PHP

Install PHP Apache Mod

```
httpd_module 'php5' do
  instance 'demo'
end
```

PHP MySql Library

```
package 'php5-mysql' do
  action :install
  notifies :restart, 'httpd_service[demo]'
end
```

Deploying Code Options

Create File

```
file ‘/var/www/vhosts/demo/index.html' do
  content ‘<html>Very Site, Much Deployment</html>’
  mode '0644'
  owner 'deploy'
  group 'deploy'
end
```

Deploying Code Options

Checkout Code

```
git “/var/www/vhosts/demo“ do
  repository 'git@github.com:mbopp/demo-repo.git'
  revision ‘master’
  action :sync
end
```

Deploying Code Options

Deploy

```
deploy ‘demo_repo' do
  repo 'git@github.com:mbopp/demo-repo.git'
  user 'deploy'
  deploy_to ‘/var/www/vhosts/demo’
  notifies :restart, 'service[demo]'
  action :deploy
end
```

Run It!

```
# chef-solo -c solo.rb -j web.json
```

Ansible

- Performs all operation of SSH (Agent-less)
- Can be done from any *nix your MacBook*
- Much shorter learning cu
- Easy to use on a smaller scale with

Ansible

- It ***CAN*** do a lot more than other solutions
  - Configuration Management (Chef, Puppet)
  - Deployment (Capistrano, Fabric)
  - Ad-Hoc Tasks (Plain SSH, at scale)

Downside

No Windows Support

Ansible Set Up

Install Python (in the unlikely case that it isn't already there)

```
# brew install python
```

Install Ansible

```
# sudo pip install ansible
```

Inventory Files

- **A list of hosts you plan to use**
- **Can contain groups of hosts**

```
/etc/ansible/hosts
[demo]
ansible-demo.grdevnight.org
second-example.grdevnight.org
…
```

Execute a command

```
# ansible demo -m ping -u root

# ansible demo -m “free -m” -u root
```

Playbooks

- Metaphor for a series of operations
- Written in YAML
- Easily converted to from shell scripts

Playbooks

playbook.yml

```
---
- hosts: all

  tasks:
    - name: Update Package Manager
      command: apt-get update
    - name: Install Apache
      command: apt-get -y install apache2
    - name: Copy configuration files.
      command: cp httpd.conf /etc/apache2/conf/apache.conf
```

Run it

```
# ansible-playbook playbook.yml -i hosts –u root
```

Roles

Groupings of Tasks, Playbooks, Handlers, Variables

playbook.yml

```
- hosts: demo

  roles:
    - server
    - php
    - mysql
    - site
```

Project Structure

```
playbook.yml
hosts
- roles
  - server
    - tasks
      - main.yml
  - php
    - tasks
      - main.yml
  - mysql
    - tasks
      - main.yml
  - site
    - handlers
      - main.yml
    - tasks
      - main.yml
```

Server Role

```
---
- name: Update apt cache
  apt: update_cache=yes cache_valid_time=3600
  sudo: yes

- name: Install required software
  apt: name={{ item }} state=present
  sudo: yes
  with_items:
    - apache2
    - mysql-server
    - php5-mysql
    - php5
    - libapache2-mod-php5
    - php5-mcrypt
    - python-mysqldb
```

PHP Role

```
---
- name: Install php extensions
  apt: name={{ item }} state=present
  sudo: yes
  with_items:
    - php5-gd
    - libssh2-php
```

MySql Role

```
---
- name: Create mysql database
  mysql_db: name=ansible_demo_db state=present

- name: Create mysql user
  mysql_user:
    name=db_admin
    password=mysql_admin_password
    priv=*.*:ALL
```

Site Role

```
---
- git: repo=git@github.com:mbopp/demo-repo.git
      dest=/var/www/vhosts/demo
      version=master

- name: Update default Apache site
  sudo: yes
  lineinfile:
    dest=/etc/apache2/sites-enabled/000-default.conf
    regexp="(.)+DocumentRoot /var/www/html"
    line="DocumentRoot /var/www/vhosts/demo“
  notify:
    - restart apache
  sudo: yes
```

Site Role - Handler

```
---
- name: restart apache
  service: name=apache2 state=restarted
  sudo: yes
```

Run it

```
# ansible-playbook playbook.yml -i hosts –u root
```

https://galaxy.ansible.com

```
# ansible-galaxy install geerlingguy.apache
```

@mwbopp

Questions

Michael Bopp

@mwbopp

Rapid Development Group