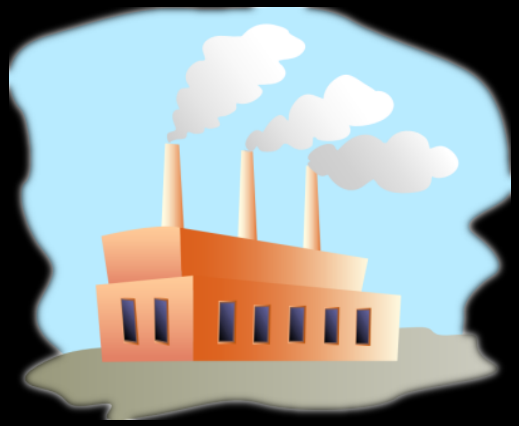
Chef solo

The beginning

Background



It is efficient to automate things that we do again and again, even it takes some time

Background

Server setup is one of the most tedious and boring task, which we have to do again and again.

Look at your screen to see if the mysql password request has come or not.

Why chef

- Idempotent: Safe to re run the script
- Thick Clients, Thin Server
- A level of platform independencies
- Rich collection of recipes (800+)
- Integration with Leading Cloud Providers

Terminology

Chef: Cooks a **recipe** for you,
they way you want

Recipe: A set of instructions for preparing
a particular dish

Cookbooks: Manages your **recipe**

Node: Configuration of your **recipe**

Knife: A tool to help you cook

Getting started

Root /
 -- cookbooks
 -- node.json
 -- run_recipies.rb

Run the script

```
$ sudo chef-solo -c run_recipies.rb -j node.json
```

Getting started

- **cookbooks**
  - mysql
    - recipes
      - default.rb
      - another_recipie.rb
    - templates
      - my.cnf
    - files

Using existing repo

Target: setup mysql with chef solo

- Pull the mysql cookbook in the **cookbooks** folder
- Change the **node.json** to serve our need

Using existing repo

Sample node.json

```
{  "run_list": [ "recipe[mysql::server]",
                 "recipe[mysql::client]" ],
   "mysql": {  "server_root_password": "",
               "server_repl_password": "",
               "server_debian_password": "",
               "bind_address": "0.0.0.0",
               "allow_remote_root": true
         }
}
```

Using existing repo

Run the script

$ sudo chef-solo -c run_recipies.rb -j node.json

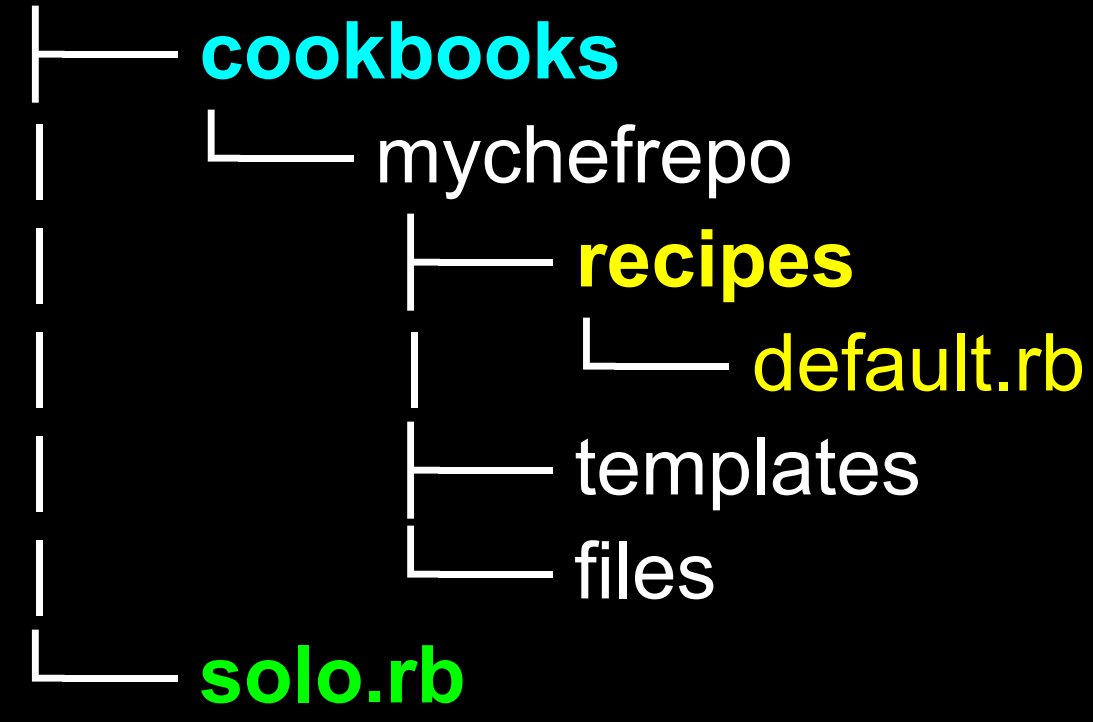
RE Run the script any number of time

$ sudo chef-solo -c run_recipies.rb -j node.json

And it is safe :D

Creating own recipe

Create folder structure as follows



default.rb

```
1 begin
2   require 'mysql'
3 rescue LoadError
4   Chef::Log.info("Missing gem 'mysql'")
5 end
6
7 ruby_block "Create replication user and database" do
8   block do
9     m = Mysql.new('localhost', "root", node[:mysql][:server_root_password])
10
11     command = %Q{
12    GRANT ALL ON *.*
13      TO '#{node[:mysql][:server_replication][:database_user]}'@'%'
14      IDENTIFIED BY
15      '#{node[:mysql][:server_replication][:database_password]}';
16    }
17     m.query(command)
18     m.query('FLUSH PRIVILEGES')
19   end
20   # only execute if mysql is running
21   only_if "pgrep 'mysqld$'"
22 end
```

default.rb

```
1 gem_package "bundler"
2 begin
3   require 'mysql'
4 rescue LoadError
5   Chef::Log.info("Missing gem 'mysql'")
6 end
```

default.rb

```
 7 ruby_block "Create replication user and database" do
 8   block do
 9     m = Mysql.new('localhost', "root", node[:mysql][:server_root_password])
10
11     command = %Q{
12    GRANT ALL ON *.*
13      TO '#{node[:mysql][:server_replication][:database_user]}'@'%'
14      IDENTIFIED BY
15      '#{node[:mysql][:server_replication][:database_password]}';
16    }
17     m.query(command)
18     m.query('FLUSH PRIVILEGES')
19   end
20   # only execute if mysql is running
21   only_if "pgrep 'mysqld$'"
22 end
```

node.json

```
{   "run_list": [ "recipe[mysql::server]",
                  "recipe[mysql::client]",
                  "recipe[mychefrepo]" ],
    "mysql": {  "server_root_password": "",
                "server_repl_password": "",
                "server_debian_password": "",
                "bind_address": "0.0.0.0",
                "allow_remote_root": true,
                "server_replication": {
                      "database": "moteel_production",
                      "database_user": "repl",
                      "database_password": "repl"}
}}
```

Run the script

Run the script

$ sudo chef-solo -c run_recipies.rb -j node.json

Commonly used resources

```
%w{rover fido bubbers}.each do |pet_name|
  execute "feed_pet_#{pet_name}" do
    command "echo 'Feeding: #{pet_name}'; touch '/tmp/#{pet_name}'"
    not_if { ::File.exists?("/tmp/#{pet_name}")}
  end
end

gem_package "bundler" do
  options(:prerelease => true, :format_executable => false)
end

package "tar" do
  version "1.16.1-1"
  action :install
end
```

Commonly used resources

```
%w{rover fido bubbers}.each do |pet_name|
  execute "feed_pet_#{pet_name}" do
    command "echo 'Feeding: #{pet_name}'; touch '/tmp/#{pet_name}'"
    not_if { ::File.exists?("/tmp/#{pet_name}")}
  end
end

gem_package "bundler" do
  options(:prerelease => true, :format_executable => false)
end

package "tar" do
  version "1.16.1-1"
  action :install
end
```

Commonly used resources

```
directory "/tmp/folder" do
  owner "root"
  group "root"
  mode 0755
  action :create
end

cookbook_file "/usr/local/bin/apache2_module_conf_generate.pl" do
  source "apache2_module_conf_generate.pl"
  mode 0755
  owner "root"
end

template "/tmp/somefile" do
  mode 00644
  source "somefile.erb"
  not_if { node[:some_value] }
end
```

Deploying to multiple servers

Problems

- Different servers might need different **node.json** file to configure them
- You have to ssh into different servers and and set up the prerequisites for chef

Deploying to multiple servers

Solution

1. Use chef server
   - Need to buy chef server service

   OR

   - Maintain a chef server of your own

OR

2. Use scripts like capistrano and some ruby to deploy from development machine

Using Capistrano

```
set :user, '...'
set :password, '.......'
set :main_server, '***.**.***.**'

server main_server, :app, :primary => true

namespace :setup do
  task :env, :depends => ["setup:install_prerequisite"], :roles => :app do
    setup.install_prerequisite
    cd_run "ruby change_local.rb #{user} #{main_server}"
    cd_run "#{try_sudo} ./setup_base.sh"
    cd_run "#{try_sudo} chef-solo -c run_recipies.rb -j local.json"
  end
end
```

Using Capistrano

```
namespace :setup do
  task :install_prerequisite, :roles => :app do
    run "mkdir -p /home/#{user}/.ssh"
    upload "deploy_keys/id_rsa", "/home/#{user}/.ssh/id_rsa"
    upload "deploy_keys/id_rsa.pub", "/home/#{user}/.ssh/id_rsa.pub"

    run "#{try_sudo} apt-get update"
    run "#{try_sudo} apt-get -y install aptitude git unzip"
    begin
      run "cd /home/#{user} && git clone git@git.tasawr.com:ashraf/moteel-setup.git"
    rescue
      cd_run "git pull origin master"
    end
    run "#{try_sudo} aptitude install -y ruby1.9.1 ruby1.9.1-dev make"
    run "#{try_sudo} gem install --no-rdoc --no-ri chef"
  end
end

def cd_run(cmd)
  run "cd /home/#{user}/moteel-setup && #{cmd}"
end
```

Using Capistrano

```
ruby_block "share the torrent file" do
  block do
    f = File.open(node['bittorrent']['torrent'],'rb')
    #read the .torrent file and base64 encode it
    enc = Base64.encode64(f.read)
    data = {
        'id'=>bittorrent_item_id(node['bittorrent']['file']),
        'seed'=>node.ipaddress,
        'torrent'=>enc
    }
    item = Chef::DataBagItem.new
    item.data_bag('bittorrent')
    item.raw_data = data
    item.save
  end
  action :nothing
  subscribes :create, resources(:bittorrent_torrent => node['bittorrent']['torrent'])
end
```

Deploying to multiple servers

1. Configure capistrano with the server **information**
2. Install prerequisites with capistrano
3. Run a ruby or any program to update the **node.json** file from the capistrano **information**
4. Go get yourself a cup of coffee

Wait for the script to complete

References

- Quick start
- Chef DSL
- Chef resources
- Some cookbooks

Author

A.K.M. Ashrafuzzaman

Lead Software Engineer,
Tasawr Interactive.

www.ashrafuzzaman.com
ashraf@tasawr.com