

1. Vagrant/VirtualBox Installation/Configuration
2. Create Local Projects (Chef Solo/Puppet Standalone)
3. Create User w/SSH. Install Language/Framework/DB
4. Create/Deploy Application (Push). Manage Settings.

Break For Lunch!

5. Convert to Client/Server Chef/Puppet
6. Use Community Resources (Cookbooks, PuppetForge)
7. Deploy Application (Pull). Manage Secrets.
8. Bonus Round: Move To The Cloud!

Part 2

Create Local Cookbook / Chef Solo

Installing the Chef Client

Select the kind of system you would like to install the Chef Client on. The versions listed have been tested and are supported.

Ubuntu

12.04

x86_64

Quick Install Instructions

Open a shell on the target system and run the following command to download client:

```
curl -L https://www.opscode.com/chef/install.sh | sudo bash
```

Downloads

You can install manually by downloading the package below after you have seen information about manual installation, [please read the documentation](#).

11.8.2-1

[chef_11.8.2-1.ubuntu.12.04_amd64.deb](#)

<http://www.getchef.com/chef/install/>

```
$ vagrant up workstation
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo -i
```

```
root@workstation:~# ls
```

```
chef_11.8.2-1.ubuntu.12.04_amd64.deb
```

```
chef-server_11.0.10-1.ubuntu.12.04_amd64.deb
```

```
root@workstation:~# dpkg -i chef_11.8.2-1.ubuntu.12.04_amd64.deb
```

```
(Reading database ... 39575 files and directories currently installed.)
```

```
Preparing to replace chef 11.4.0-1.ubuntu.11.04 (using chef_11.8.2-1.ubuntu.12.04_amd64.deb) ...
```

```
Unpacking replacement chef ...
```

```
Setting up chef (11.8.2-1.ubuntu.12.04) ...
```

```
Thank you for installing Chef!
```

```
root@workstation:~# chef-solo --version
```

```
Chef: 11.8.2
```

```
root@workstation:~# cd /vagrant
```

```
root@workstation:/vagrant# ls
```

Vagrantfile

```
root@workstation:/vagrant# knife cookbook create mycookbook -o ./cookbooks
```

```
root@workstation:/vagrant# knife cookbook create mycookbook -o ./cookbooks \  
-C "Christopher H. Laco" -m "claco@chrislaco.com" -I apachev2
```

```
WARNING: No knife configuration file found
```

```
** Creating cookbook mycookbook
```

```
** Creating README for cookbook: mycookbook
```

```
** Creating CHANGELOG for cookbook: mycookbook
```

```
** Creating metadata for cookbook: mycookbook
```

```
root@workstation:/vagrant# tree .
```

```
├── cookbooks
│   ├── mycookbook
│       ├── attributes
│       ├── CHANGELOG.md
│       ├── definitions
│       ├── files
│       │   ├── default
│       ├── libraries
│       ├── metadata.rb
│       ├── providers
│       ├── README.md
│       ├── recipes
│       │   ├── default.rb
│       ├── resources
│       └── templates
│           └── default
└── Vagrantfile
```

```
12 directories, 5 files
```

```
root@workstation:/vagrant# cat cookbooks/mycookbook/metadata.rb
```

```
name          'mycookbook'
maintainer     'Christopher H. Laco'
maintainer_email 'claco@chrislaco.com'
license        'Apache 2.0'
description    'Installs/Configures mycookbook'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'
```

```
root@workstation:/vagrant# cat cookbooks/mycookbook/recipes/default.rb
```

```
#
# Cookbook Name:: mycookbook
# Recipe:: default
#
# Copyright 2014, Christopher H. Laco
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
```

```
root@workstation:/vagrant# exit
```

```
root@workstation:/vagrant# vim Vagrantfile
```

```
~ #85
```

```
# config.vm.provision :chef_solo do |chef|
#   chef.cookbooks_path = "../my-recipes/cookbooks"
#   chef.roles_path = "../my-recipes/roles"
#   chef.data_bags_path = "../my-recipes/data_bags"
#   chef.add_recipe "mysql"
#   chef.add_role "web"
#
#   # You may also specify custom JSON attributes:
#   chef.json = { :mysql_password => "foo" }
# end
```

```
workstation.vm.provision :chef_solo do |chef|
  chef.cookbooks_path = "./cookbooks"
  chef.roles_path = "./roles"
  chef.data_bags_path = "./data_bags"
  chef.add_recipe "mycookbook::default"
end
```

```
root@workstation:/vagrant# exit
```



```
$ vagrant provision workstation
```

```
[workstation] The cookbook path '...workspace/roles' doesn't exist. Ignoring...
```

```
[workstation] The cookbook path '...workspace/data_bags' doesn't exist. Ignoring...
```

```
[workstation] Running provisioner: chef_solo...
```

Shared folders that Chef requires are missing on the virtual machine.

This is usually due to configuration changing after already booting the machine. The fix is to run a `vagrant reload` so that the proper shared folders will be prepared and mounted on the VM.

```
$ mkdir ./roles ./data_bags
```

```
$ vagrant reload workstation
```

```
[workstation] Attempting graceful shutdown of VM...
```

```
[workstation] Clearing any previously set forwarded ports...
```

```
[workstation] Creating shared folders metadata...
```

```
...
```

```
[workstation] -- /vagrant
```

```
[workstation] -- /tmp/vagrant-chef-1/chef-solo-2/roles
```

```
[workstation] -- /tmp/vagrant-chef-1/chef-solo-1/cookbooks
```

```
[workstation] -- /tmp/vagrant-chef-1/chef-solo-3/data_bags
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
[2014-01-03T19:25:27+00:00] INFO: Forking chef instance to converge...
```

```
[2014-01-03T19:25:27+00:00] INFO: *** Chef 11.8.2 ***
```

```
[2014-01-03T19:25:27+00:00] INFO: Chef-client pid: 1311
```

```
[2014-01-03T19:25:27+00:00] INFO: Setting the run_list to
```

```
["recipe[mycookbook::default]"] from JSON
```

```
[2014-01-03T19:25:27+00:00] INFO: Run List is [recipe[mycookbook::default]]
```

```
[2014-01-03T19:25:27+00:00] INFO: Run List expands to [mycookbook::default]
```

```
[2014-01-03T19:25:27+00:00] INFO: Starting Chef Run for workstation
```

```
[2014-01-03T19:25:27+00:00] INFO: Running start handlers
```

```
[2014-01-03T19:25:27+00:00] INFO: Start handlers complete.
```

```
[2014-01-03T19:25:27+00:00] INFO: Chef Run complete in 0.02684892 seconds
```

```
[2014-01-03T19:25:27+00:00] INFO: Running report handlers
```

```
[2014-01-03T19:25:27+00:00] INFO: Report handlers complete
```

```
[2014-01-03T19:25:27+00:00] INFO: Forking chef instance to converge...
```

Congratulations on your first Chef convergence!

```
$ vim roles/workstation.rb
```

```
name "workstation"  
description "Workstation Configuration"  
run_list "recipe[mycookbook::default]"
```

```
$ vim Vagrantfile
```

```
# chef.add_recipe "mycookbook::default"  
chef.add_role "workstation"
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
[2014-01-03T19:25:27+00:00] INFO: Run List is [role[workstation]]
```

```
[2014-01-03T19:25:27+00:00] INFO: Run List expands to [mycookbook::default]
```

```
[2014-01-03T19:25:27+00:00] INFO: Starting Chef Run for workstation
```

```
[2014-01-03T19:25:27+00:00] INFO: Running start handlers
```

```
[2014-01-03T19:25:27+00:00] INFO: Start handlers complete.
```

```
...
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ cd /vagrant/roles
```

```
vagrant@workstation:/vagrant/roles$ knife role from file workstation.rb -z
```

```
WARNING: No knife configuration file found
```

```
Updated Role workstation!
```

```
vagrant@workstation:/vagrant/roles$ ls
```

```
workstation.json  workstation.rb
```

```
vagrant@workstation:/vagrant/roles$ cat workstation.json
```

```
{  
  "name": "workstation",  
  "description": "Workstation Configuration",  
  "run_list": [  
    "recipe[mycookbook::default]"  
  ]  
}
```

```
$ rm roles/workstation.rb
```

```
$ vagrant provision workstation
```

```
[2014-01-03T19:43:38+00:00] FATAL: Stacktrace dumped to /var/chef/cache/chef-stacktrace.out
```

```
[2014-01-03T19:43:38+00:00] ERROR: undefined method `run_list_for' for #<Hash:0x00000000c8a9d8>
```

```
=====
Error expanding the run_list:
=====
```

```
Unexpected Error:
```

```
-----
```

```
NoMethodError: undefined method `run_list_for' for #<Hash:0x00000000c8a9d8>
```

```
vagrant@workstation:/vagrant/roles$ vim workstation.json
```

```
{  
  "name": "workstation",  
  "description": "Workstation Configuration",  
  "json_class": "Chef::Role",  
  "chef_type": "role",  
  "run_list": [  
    "recipe[mycookbook::default]"  
  ]  
}
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
[2014-01-03T19:25:27+00:00] INFO: Starting Chef Run for workstation
```

```
[2014-01-03T19:25:27+00:00] INFO: Running start handlers
```

```
[2014-01-03T19:25:27+00:00] INFO: Start handlers complete.
```

Part 2 Complete!

Part 3

Create User/SSH - Install Ruby/Rails/MySQL


```
$ openssl passwd -1 password
```

```
$1$ezZaXpFc$cqr4qoT4sZDxXaCrkVfJ1.
```

```
$1$fr36jZxV$lzx6.Lzn/YtoQH1NUPdrH0
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo apt-get install whois -y
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following NEW packages will be installed:
```

```
  whois
```

```
...
```

```
vagrant@workstation:~$ mkpasswd -m sha-512
```

```
Password:
```

```
$6$ECIy/pVovVQ6qja$A0owsw1LKsPPUQe2CKi.5IfAWuLGvyj3IgiRJfFxG3/MrmYTtA0PRE0qaBVv/  
lBgmSfI27T.Tg2EzrC7A72qI/
```

```
$ cp ./cookbooks/mycookbook/recipes/default.rb \  
    ./cookbooks/mycookbook/recipes/users.rb
```

```
$ vim ./cookbooks/mycokbook/recipes/users.rb
```

```
user "build" do  
  home "/home/build"  
  shell "/bin/bash"  
  password "$1$ezZaXpFc$cqr4qoT4sZDxXaCrkVfJ1."  
  supports :manage_home => true  
end
```

```
$ vim ./roles/workstation.json
```

```
{  
  "name": "workstation",  
  "description": "Workstation Configuration",  
  "json_class": "Chef::Role",  
  "chef_type": "role",  
  "run_list": [  
    "recipe[mycookbook::users]"  
  ]  
}
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
[2014-01-03T23:55:41+00:00] INFO: Start handlers complete.
```

```
[2014-01-03T23:55:41+00:00] INFO: user[build] created
```

```
[2014-01-03T23:55:41+00:00] INFO: Chef Run complete in 0.071330842
```

```
$ ssh build@192.168.33.10
```

```
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-23-generic x86_64)
```

```
...
```

```
build@workstation:~$ exit
```

```
$ mkdir ./cookbooks/mycookbook/files/default/ssh
```

```
$ ssh-keygen -t rsa
```

Generating public/private rsa key pair.

Enter file in which to save the key (/Users/claco/.ssh/id_rsa): **cookbooks/
mycookbook/files/default/ssh/id_rsa**

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in **cookbooks/mycookbook/files/default/ssh/id_rsa**.

Your public key has been saved in **cookbooks/mycookbook/files/default/ssh/id_rsa.pub**.

The key fingerprint is:

17:9a:cd:df:ff:7a:b2:60:90:d6:57:26:0f:22:ae:45 claco@hank

The key's randomart image is:

```
+--[ RSA 2048 ]-----+
|                        |
|                        |
|          E . o o |
```

```
$ vim ./cookbooks/mycookbook/recipes/users.rb
```

```
user "build" do
  home "/home/build"
  shell "/bin/bash"
  password "$1$ezZaXpFc$cqr4qoT4sZDxXaCrkVfJ1."
  supports :manage_home => true
end
```

```
directory "/home/build/.ssh" do
  owner "build"
  group "build"
  mode "0700"
end
```

```
cookbook_file "/home/build/.ssh/authorized_keys" do
  source "ssh/id_rsa.pub"
  owner "build"
  group "build"
  mode "0600"
end
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
INFO: directory[/home/build/.ssh] created directory /home/build/.ssh
```

```
INFO: directory[/home/build/.ssh] owner changed to 1001
```

```
INFO: directory[/home/build/.ssh] group changed to 1002
```

```
INFO: directory[/home/build/.ssh] mode changed to 700
```

```
INFO: cookbook_file[/home/build/.ssh/authorized_keys] created file /home/build/.ssh/authorized_keys
```

```
INFO: cookbook_file[/home/build/.ssh/authorized_keys] updated file contents /home/build/.ssh/authorized_keys
```

```
INFO: cookbook_file[/home/build/.ssh/authorized_keys] owner changed to 1001
```

```
INFO: cookbook_file[/home/build/.ssh/authorized_keys] group changed to 1002
```

```
INFO: cookbook_file[/home/build/.ssh/authorized_keys] mode changed to 600
```

```
$ ssh build@192.168.33.10 -i ./cookbooks/mycookbook/files/default/ssh/id_rsa
```

```
Last login: Fri Jan 3 01:11:05 2014 from 192.168.33.1
```

```
...
```

```
build@workstation:~$ exit
```

```
$ mkdir ./cookbooks/mycookbook/templates/default/sudo
```

```
$ vim ./cookbooks/mycookbook/templates/default/sudo/user.erb
```

```
<%= @username %>    ALL=(ALL) ALL
```

```
$ vim ./cookbooks/mycookbook/recipes/users.rb
```

```
template "/etc/sudoers.d/build" do
  source "sudo/user.erb"
  owner "root"
  group "root"
  mode "0440"
  variables :username => "build"
end
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
INFO: template[/etc/sudoers.d/build] created file /etc/sudoers.d/build
```

```
INFO: template[/etc/sudoers.d/build] updated file contents /etc/sudoers.d/build
```

```
INFO: template[/etc/sudoers.d/build] owner changed to 0
```

```
INFO: template[/etc/sudoers.d/build] group changed to 0
```

```
INFO: template[/etc/sudoers.d/build] mode changed to 440
```

```
$ vagrant ssh workstation / build@workstation:~$ sudo -u build
```

```
$ ssh build@192.168.33.10 -i ./cookbooks/mycookbook/files/default/ssh/id_rsa
```

```
Last login: Fri Jan  3 01:11:05 2014 from 192.168.33.1
```

```
...
```

```
build@workstation:~$ sudo whoami
```

```
[sudo] password for build:
```

```
root
```



```
$ cp ./cookbooks/mycookbook/recipes/default.rb \  
    ./cookbooks/mycookbook/recipes/apt.rb
```

```
$ vim ./cookbooks/mycookbook/recipes/apt.rb
```

```
execute "apt-get update" do  
  command "apt-get update"  
end
```

```
$ vim ./roles/workstation.json
```

```
{  
  "name": "workstation",  
  "description": "Workstation Configuration",  
  "json_class": "Chef::Role",  
  "chef_type": "role",  
  "run_list": [  
    "recipe[mycookbook::users]",  
    "recipe[mycookbook::apt]"  
  ]  
}
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
INFO: Run List expands to [mycookbook::users, mycookbook::apt]
```

```
...
```

```
INFO: execute[apt-get update] ran successfully
```

```
...
```

```
INFO: Chef Run complete in 11.662078191 seconds
```

```
INFO: Running report handlers
```

```
INFO: Report handlers complete
```

```
INFO: Forking chef instance to converge...
```

```
$ cp ./cookbooks/mycookbook/recipes/default.rb \
    ./cookbooks/mycookbook/recipes/mysql.rb

$ vim ./cookbooks/mycookbook/recipes/mysql.rb

package "mysql-server"
package "mysql-client" do
  action :install
end

package "build-essential"
package "libmysqlclient-dev"

$ vim ./roles/workstation.json

{
  ...
  "run_list": [
    "recipe[mycookbook::users]",
    "recipe[mycookbook::apt]",
    "recipe[mycookbook::mysql]"
  ]
}
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
INFO: Run List expands to [mycookbook::users, mycookbook::apt, mycookbook::mysql]
```

```
...
```

```
INFO: execute[apt-get update] ran successfully
```

```
...
```

```
INFO: Chef Run complete in 11.662078191 seconds
```

```
INFO: Running report handlers
```

```
INFO: Report handlers complete
```

```
INFO: Forking chef instance to converge...
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
...
```

```
mysql> exit
```

```
$ cp ./cookbooks/mycookbook/recipes/default.rb \
    ./cookbooks/mycookbook/recipes/ruby.rb

$ vim ./cookbooks/mycookbook/recipes/ruby.rb

package "ruby1.9.3"

$ cp ./cookbooks/mycookbook/recipes/default.rb \
    ./cookbooks/mycookbook/recipes/rails.rb

$ vim ./cookbooks/mycookbook/recipes/rails.rb

include_recipe "mycookbook::ruby"
gem_package "rails" do
  options("--no-ri --no-rdoc")
end

$ vim ./roles/workstation.json

"run_list": [
  "recipe[mycookbook::users]",
  "recipe[mycookbook::apt]",
  "recipe[mycookbook::mysql]",
  "recipe[mycookbook::rails]"
```

```
$ vagrant provision workstation
```

```
[workstation] Running provisioner: chef_solo...
```

```
Generating chef JSON and uploading...
```

```
Running chef-solo...
```

```
...
```

```
INFO: Run List expands to [mycookbook::users, mycookbook::apt, mycookbook::mysql,  
mycookbook::rails]
```

```
...
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ which ruby
```

```
/usr/bin/ruby
```

```
vagrant@workstation:~$ ruby --version
```

```
ruby 1.9.3p0 (OMG!) (2011-10-30 revision 33570) [x86_64-linux]
```

```
vagrant@workstation:~$ which rails && rails --version
```

```
/usr/local/bin/rails
```

```
Rails 4.0.2
```

```
vagrant@workstation:~$ sudo -u build -i
```

```
build@workstation:~$ rails new myapp --database=mysql --skip-bundle
```

```
ruby 1.9.3p0 (OMG!) (2011-10-30 revision 33570) [x86_64-linux]
```

```
vagrant@workstation:~$ which rails && rails --version
```

```
create
```

```
create  README.rdoc
```

```
create  Rakefile
```

```
.....
```

```
create  vendor/assets/stylesheets/.keep
```

```
build@workstation:~$ cd myapp && cat config/database.yml
```

```
development:
```

```
  adapter: mysql2
```

```
  encoding: utf8
```

```
  database: myapp_development
```

```
  pool: 5
```

```
  username: root
```

```
  password:
```

```
  socket: /var/run/mysqld/mysqld.sock
```

```
build@workstation:~/myapp$ vim Gemfile
```

```
gem 'therubyracer', platforms: :ruby
```

```
gem 'unicorn'
```

```
gem 'capistrano', '~> 2.15.4'
```

```
gem 'capistrano-unicorn'
```

```
build@workstation:~/myapp$ bundle install
```

```
Fetching gem metadata from https://rubygems.org/.....
```

```
Fetching additional metadata from https://rubygems.org/..
```

```
Resolving dependencies...
```

```
Using rake (10.1.1)
```

```
Using i18n (0.6.9)
```

```
...
```

```
Your bundle is complete!
```

```
build@workstation:~/myapp$ rake -T
```

```
rake about          # List versions of all Rails frameworks and the environment
```

```
rake assets:clean[keep] # Remove old compiled assets
```

```
rake assets:clobber   # Remove compiled assets
```



```
build@workstation:~/myapp$ rake db:create db:migrate
```

```
build@workstation:~/myapp$ rails server
```

```
=> Booting WEBrick
```

```
=> Rails 4.0.2 application starting in development on http://0.0.0.0:3000
```

```
=> Run `rails server -h` for more startup options
```

```
=> Ctrl-C to shutdown server
```

```
[2014-01-04 02:04:35] INFO  WEBrick 1.3.1
```

```
[2014-01-04 02:04:35] INFO  ruby 1.9.3 (2011-10-30) [x86_64-linux]
```

```
[2014-01-04 02:04:35] INFO  WEBrick::HTTPServer#start: pid=14775 port=3000
```



Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

1. Use `rails generate` to create your models and controllers

To see all available options, run it without parameters.

2. Set up a default route and remove *public/index.html*

Routes are set up in *config/routes.rb*.

3. Create your database

Run `rake db:create` to create your database. If you're not using SQLite (the default), edit *config/database.yml* with your username and password.

Browse the documentation

[Rails Guides](#)

[Rails API](#)

[Ruby core](#)

[Ruby standard library](#)

Browser: `http://192.168.33.10:3000`

Part 3 Complete!

Part 4

Create/Deploy App (Push) / Manage Settings/Configs

```
$ cp ./cookbooks/mycookbook/recipes/default.rb \
    ./cookbooks/mycookbook/recipes/myapp.rb

$ cp ./roles/workstation.json roles/server.json

$ vim ./roles/server.json

{
  "name": "server",
  "description": "Server Configuration",
  "json_class": "Chef::Role",
  "chef_type": "role",
  "run_list": [
    "recipe[mycookbook::users]",
    "recipe[mycookbook::apt]",
    "recipe[mycookbook::mysql]",
    "recipe[mycookbook::ruby]",
    "recipe[mycookbook::myapp]"
  ]
}
```

```
$ vim Vagrantfile
```

```
$ cp ./roles/workstation.json roles/server.json
```

```
$ vim ./roles/server.json
```

```
config.vm.define "server" do |server|
```

```
  server.vm.hostname = "server"
```

```
  server.vm.network :private_network, ip: "192.168.33.20"
```

```
  server.vm.provision :chef_solo do |chef|
```

```
    chef.cookbooks_path = "./cookbooks"
```

```
    chef.roles_path = "./roles"
```

```
    chef.data_bags_path = "./data_bags"
```

```
    chef.add_role "server"
```

```
  end
```

```
end
```

```
$ vagrant up server (or provision)
```

```
Bringing machine 'server' up with 'virtualbox' provider...
```

```
...
```

```
[server] Running provisioner: chef_solo...
```

```
...
```

```
INFO: Setting the run_list to ["role[server]"] from JSON
```

```
INFO: Run List is [role[server]]
```

```
INFO: Run List expands to [mycookbook::users, mycookbook::apt, mycookbook::mysql,  
mycookbook::myapp]
```

```
INFO: Starting Chef Run for server
```

```
INFO: Running start handlers
```

```
INFO: Start handlers complete.
```

```
INFO: execute[apt-get update] ran successfully
```

```
INFO: Chef Run complete in 16.189321731 seconds
```

```
INFO: Running report handlers
```

```
INFO: Report handlers complete
```

```
INFO: Forking chef instance to converge...
```

```
$ vim ./cookbooks/mycookbook/recipes/myapp.rb
```

```
directory "/var/www/myapp" do
  owner "build"
  group "build"
  mode "0755"
  recursive true
end
```

```
$ vagrant provision server
```

```
...
```

```
INFO: directory[/var/www/myapp] created directory /var/www/myapp
```

```
INFO: directory[/var/www/myapp] owner changed to 1001
```

```
INFO: directory[/var/www/myapp] group changed to 1002
```

```
INFO: directory[/var/www/myapp] mode changed to 755
```

```
$ vagrant ssh server
```

```
vagrant@server:~$ ls -lta /var/www
```

```
drwxr-xr-x  2 build build 4096 Jan  4 03:21 myapp
```



```
$ vim ./cookbooks/mycookbook/recipes/myapp.rb
```

```
execute "add-mysql-user" do
  command "/usr/bin/mysql -u root -D mysql -r -B -N -e \"CREATE USER
    'myapp'@'localhost' IDENTIFIED BY 'password'\""

  action :run

  only_if { `/usr/bin/mysql -u root -D mysql -r -B -N -e "SELECT COUNT(*) FROM
    user where User='myapp' and Host='localhost'".to_i == 0 }

end
```

Copy from HandsOnDevOps/examples/myapp.rb

development:

adapter: mysql2

encoding: utf8

database: myapp_development

pool: 5

username: root

password:

socket: /var/run/mysqld/mysqld.sock

production:

adapter: mysql2

encoding: utf8

database: myapp

pool: 5

username: myapp

password: password

socket: /var/run/mysqld/mysqld.sock

```
$ vim ./cookbooks/mycookbook/templates/default/database.yml.erb
```

```
<%= @environment %>:
  adapter: <%= @adapter %>
  encoding: <%= @encoding %>
  database: <%= @database %>
  pool: <%= @pool %>
  username: <%= @username %>
  password: <%= @password %>
  socket: <%= @socket %>
```

```
$ vim ./cookbooks/mycookbook/recipes/myapp.rb
```

```
directory "/var/www/myapp/shared/config" do
  owner "build"
  group "build"
  mode "0755"
  recursive true
end
```

```
$ vim ./cookbooks/mycookbook/templates/default/database.yml.erb
```

```
template "/var/www/myapp/shared/config/database.yml" do
  source "database.yml.erb"
  owner "build"
  group "build"
  mode "0440"
  variables(
    :environment => "production",
    :adapter => "mysql2"
    :encoding => "utf8",
    :database => "myapp",
    :pool => 10,
    :username => "myapp",
    :password => "password",
    :socket => "/var/run/mysqld/mysqld.sock"
  )
end
```

Copy from HandsOnDevOps/examples/myapp.rb

```
$ vagrant provision server
...
[server] Running provisioner: chef_solo...
...
INFO: directory[/var/www/myapp/shared/config] created directory /var/www/myapp/
shared/config
INFO: directory[/var/www/myapp/shared/config] owner changed to 1001
INFO: directory[/var/www/myapp/shared/config] group changed to 1002
INFO: directory[/var/www/myapp/shared/config] mode changed to 755

INFO: template[/var/www/myapp/shared/config/database.yml] created file /var/www/
myapp/shared/config/database.yml
INFO: template[/var/www/myapp/shared/config/database.yml] updated file contents /
var/www/myapp/shared/config/database.yml
INFO: template[/var/www/myapp/shared/config/database.yml] owner changed to 1001
INFO: template[/var/www/myapp/shared/config/database.yml] group changed to 1002
INFO: template[/var/www/myapp/shared/config/database.yml] mode changed to 440
```

```
$ vagrant ssh server
```

```
vagrant@server:~$ sudo cat /var/www/myapp/shared/config/database.yml
```

```
production:
  adapter: mysql2
  encoding: utf8
  database: myapp
  pool: 10
  username: myapp
  password: password
  socket: /var/run/mysqld/mysqld.sock
```

```
$ vim cookbooks/mycookbook/templates/default/unicorn.rb.erb
```

```
listen <%= @listen %>
```

```
$ vim cookbooks/mycookbook/recipes/myapp.rb
```

```
template "/var/www/myapp/shared/config/unicorn.rb" do
  source "unicorn.rb.erb"
  owner "build"
  group "build"
  mode "0440"
  variables :listen => "8080"
end
```

```
$ vagrant provision server
```

```
...
```

```
[server] Running provisioner: chef_solo...
```

```
...
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] created file /var/www/myapp/  
shared/config/unicorn.rb
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] updated file contents /var/  
www/myapp/shared/config/unicorn.rb
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] owner changed to 1001
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] group changed to 1002
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] mode changed to 440
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo -u build -i cd myapp
```

```
build@workstation:~/myapp$ capify .
```

```
[add] writing './Capfile'
```

```
[add] writing './config/deploy.rb'
```

```
[done] capified!
```

```
build@workstation:~/myapp$ vim config/deploy.rb
```



```
require "bundler/capistrano"
require "capistrano-unicorn"

set :application, "myapp"
set :deploy_to,   "/var/www/#{application}"
set :user, "build"
set :use_sudo, false
set :repository, "."
set :scm, :none
set :deploy_via, :copy
set :unicorn_bin, :unicorn_rails
set :rails_env, :production
set :unicorn_rack_env, :production

role :web, "192.168.33.20"
role :app, "192.168.33.20"
role :db,  "192.168.33.20", :primary => true
```

Copy from HandsOnDevOps/examples/deploy.rb

```
after "deploy:update", "db:copy_config"  
after "deploy:update", "unicorn:copy_config"
```

```
namespace :db do  
  desc 'Copies shared database.yml'  
  task :copy_config do  
    run "cd #{current_path} && ln -nfs #{shared_path}/config/database.yml  
#{release_path}/config/database.yml"  
  end  
end
```

```
namespace :unicorn do  
  desc 'Copies shared unicorn.rb'  
  task :copy_config do  
    run "cd #{current_path} && ln -nfs #{shared_path}/config/unicorn.rb  
#{release_path}/config/unicorn.rb"  
  end  
end
```

Copy from HandsOnDevOps/examples/deploy.rb

```
build@workstation:~/myapp$ cap -vT
```

```
cap bundle:install          # Install the current Bundler environment.
cap db:copy_config          # Copies shared database.yml
cap unicorn:copy_unicorn_config # Copies shared unicorn.rb
cap invoke                  # Invoke a single command on the remote servers.
...
```

```
build@workstation:~/myapp$ cap invoke COMMAND=hostname
```

```
* 2014-01-04 04:28:51 executing `invoke`
```

```
* executing "hostname"
```

```
  servers: ["192.168.33.20"]
```

```
Password:
```

```
  [192.168.33.20] executing command
```

```
** [out :: 192.168.33.20] server
```

```
  command finished in 20ms
```

```
build@workstation:~/myapp$ cap deploy:setup
```

```
* 2014-01-04 04:31:52 executing `deploy:setup`
```

```
* executing "mkdir -p /var/www/myapp /var/www/myapp/releases /var/www/myapp/shared /var/www/myapp/shared/system /var/www/myapp/shared/log /var/www/myapp/shared/pids"
```

```
servers: ["192.168.33.20"]
```

```
Password:
```

```
[192.168.33.20] executing command
```

```
command finished in 16ms
```

```
* executing "chmod g+w /var/www/myapp /var/www/myapp/releases /var/www/myapp/shared /var/www/myapp/shared/system /var/www/myapp/shared/log /var/www/myapp/shared/pids"
```

```
servers: ["192.168.33.20"]
```

```
[192.168.33.20] executing command
```

```
command finished in 5ms
```

```
build@workstation:~/myapp$ cap deploy:cold
```

```
* 2014-01-04 16:47:29 executing `deploy:cold`
```

```
* 2014-01-04 16:47:29 executing `deploy:update`
```

```
* getting (via checkout) revision to /tmp/20140104164729
```

```
executing locally: cp -R . /tmp/20140104164729
```

```
command finished in 4ms
```

```
servers: ["192.168.33.20"]
```

```
Password:
```

```
[192.168.33.20] executing command
```

```
command finished in 16ms
```

```
** sftp upload /tmp/20140104164729.tar.gz -> /tmp/20140104164729.tar.gz
```

```
servers: ["192.168.33.20"]
```

```
[192.168.33.20] executing command
```

```
command finished in 5ms
```

```
...#$^%$#.....
```

```
exec rake RAILS_ENV=production db:migrate"
```

```
* 2014-01-04 17:25:53 executing `deploy:start`
```

```
$ vagrant ssh server
```

```
vagrant@server:~$ sudo ls -lta /var/www/myapp/current/config
```

```
lrwxrwxrwx  1 build build   39 Jan  4 17:25 unicorn.rb -> /var/www/myapp/shared/  
config/unicorn.rb
```

```
lrwxrwxrwx  1 build build   41 Jan  4 17:25 database.yml -> /var/www/myapp/shared/  
config/database.yml
```

```
build@workstation:~/myapp$ cap unicorn:start
```

```
* 2014-01-04 17:29:49 executing `unicorn:start`
```

```
Password:
```

```
[192.168.33.20] executing command
```

```
** [out :: 192.168.33.20] Starting Unicorn...
```

```
command finished in 437ms
```

```
vagrant@server:~$ sudo ps aux | grep unicorn
```

```
build      13357  0.0  1.7  70828 17712 ?           Sl   17:29   0:00 unicorn_rails  
master -c /var/www/myapp/current/config/unicorn.rb -E production -D
```

```
build      13360  2.0  5.5 165648 56644 ?           Sl   17:29   0:02 unicorn_rails  
worker[0] -c /var/www/myapp/current/config/unicorn.rb -E production -D
```

The page you were looking for doesn't exist.

You may have mistyped the address or the page may
have moved.

If you are the application owner check the logs for more information.

Browser: http://192.168.33.20:8000

Part 4 Complete!

LUNCH!!!!!!!!!!!!!!

Part 5

Convert to Chef Client/Server

Installing the Chef Server

Select the kind of system you would like to install the Chef Server on. The versions listed have been tested and are supported.

Ubuntu

12.10

x86_64

Downloads

You can manually install after you select a Chef version, fill out the form below package. For more information about manual installation, [please read the doc](#)

11.0.10

[chef-server_11.0.10-1.ubuntu.12.04_amd64.deb](#)

Instructions

Below are a few easy steps to get your server up and running.

1. Install the file using the correct method for your system (i.e. "dpkg -i chef
2. sudo chef-server-ctl reconfigure

<http://www.getchef.com/chef/install/>

```
$ vagrant up workstation
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo -i
```

```
root@workstation:~# ls
```

```
chef_11.8.2-1.ubuntu.12.04_amd64.deb
```

```
chef-server_11.0.10-1.ubuntu.12.04_amd64.deb
```

```
root@workstation:~# dpkg -i chef-server_11.0.10-1.ubuntu.12.04_amd64.deb
```

```
(Reading database ... 62333 files and directories currently installed.)
```

```
Unpacking chef-server (from chef-server_11.0.10-1.ubuntu.12.04_amd64.deb) ...
```

```
Unpacking replacement chef ...
```

```
Setting up chef-server (11.0.10-1.ubuntu.12.04) ...
```

```
Thank you for installing Chef Server!
```

```
The next step in the install process is to run:
```

```
sudo chef-server-ctl reconfigure
```

```
root@workstation:~# chef-server-ctl reconfigure
```

```
...
```

```
Recipe: chef-server::erchef
```

```
  * service[erchef] action restart
    - restart service service[erchef]
```

```
Chef Client finished, 270 resources updated
```

```
chef-server Reconfigured!
```

```
root@workstation:~# cp /etc/chef-server/chef-validator.pem /vagrant
```

```
root@workstation:~# cp /etc/chef-server/admin.pem /vagrant
```

```
root@workstation:~# knife configure
```

```
WARNING: No knife configuration file found
```

```
Where should I put the config file? [/root/.chef/knife.rb]
```

```
Please enter the chef server URL: [https://workstation:443]
```

```
Please enter an existing username or clientname for the API: [vagrant] admin
```

```
Please enter the validation clientname: [chef-validator]
```

```
Please enter the location of the validation key: [/etc/chef-server/chef-validator.pem]
```

```
Please enter the path to a chef repository (or leave blank): /vagrant
```

```
*****
```

```
You must place your client key in:
```

```
  /root/.chef/admin.pem      (cp /etc/chef-server/admin.pem /root/.chef/admin.pem)
```

```
Before running commands with Knife!
```

```
*****
```

```
You must place your validation key in:
```

```
  /etc/chef-server/chef-validator.pem
```

```
Before generating instance data with Knife!
```

```
root@workstation:~# cp /etc/chef-server/admin.pem .chef
```

```
root@workstation:~# knife client list
```

```
chef-validator
```

```
chef-webui
```

```
root@workstation:~# knife role from file /vagrant/roles/*
```

```
Updated Role server!
```

```
Updated Role workstation!
```

```
root@workstation:~# knife role show server
```

```
root@workstation:~# knife cookbook upload -o /vagrant/cookbooks mycookbook
```

```
Uploading mycookbook [0.1.0]
```

```
Uploaded 1 cookbook.
```

```
root@workstation:~# knife cookbook show mycookbook
```

```
root@workstation:~# exit
```

```
$ vim Vagrantfile
```

```
server.vm.provision :chef_solo do |chef|  
  chef.cookbooks_path = "./cookbooks"  
  chef.roles_path = "./roles"  
  chef.data_bags_path = "./data_bags"  
  chef.add_recipe "mycookbook::default"  
end
```

```
config.vm.define "server" do |server|  
  server.vm.hostname = "server"  
  server.vm.network :private_network, ip: "192.168.33.20"
```

```
server.vm.provision :chef_client do |chef|  
  chef.chef_server_url = "https://192.168.33.10"  
  chef.validation_key_path = "./chef-validator.pem"
```

```
  chef.add_role "server"
```

```
end
```

```
end
```

```
$ vagrant provision server
```

```
...
```

```
[server] Running provisioner: chef_client...
```

```
...
```

```
INFO: Client key /etc/chef/client.pem is not present - registering
```

```
INFO: template[/var/www/myapp/shared/config/unicorn.rb] updated file contents /var/  
www/myapp/shared/config/unicorn.rb
```

```
...
```

```
INFO: Loading cookbooks [mycookbook]
```

```
=====
```

```
Error Syncing Cookbooks:
```

```
=====
```

```
Unexpected Error:
```

```
-----
```

```
Chef::Exceptions::RedirectLimitExceeded:
```

```
Chef::Exceptions::RedirectLimitExceeded
```

```
...
```

```
FATAL: Stacktrace dumped to /var/chef/cache/chef-stacktrace.out
```

```
Please fix your recipes so that they properly complete.
```


CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

CHEF 11 REQUIRES WORKING DNS RESOLUTION

```
$ vagrant ssh server
```

```
vagrant@server:~$ sudo -i
```

```
root@server:~# echo "192.168.33.10 workstation" >> /etc/hosts
```

```
root@server:~# cat /etc/hosts
```

```
127.0.0.1          server server
```

```
127.0.1.1          server server
```

```
# The following lines are desirable for IPv6 capable hosts
```

```
::1          ip6-localhost ip6-loopback
```

```
fe00::0 ip6-localnet
```

```
ff00::0 ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
```

```
ff02::2 ip6-allrouters
```

```
192.168.33.10 workstation
```

```
root@server:~# exit
```

```
$ vagrant provision server
```

```
INFO: Chef Run complete in 4.818434674 seconds
```

Part 5 Complete!

Part 6

Use Community Cookbooks

```
$ vagrant up workstation
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo -i
```

```
root@workstation:~# gem install berkshelf
```

```
Fetching: activesupport-3.2.16.gem (100%)
```

```
Fetching: addressable-2.3.5.gem (100%)
```

```
Fetching: buff-ruby_engine-0.1.0.gem (100%)
```

```
Fetching: buff-shell_out-0.1.1.gem (100%)
```

```
...
```

```
root@workstation:~# cd /vagrant/cookbooks/mycookbook && berks init .
```

```
  create  Berksfile
```

```
  create  Thorfile
```

```
  create  cheffignore
```

```
  create  .gitignore
```

```
    run  git init from "."
```

```
  create  Gemfile
```

```
  create  Vagrantfile
```

```
Successfully initialized
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks config
```

```
Enter value for chef.chef_server_url (default: 'http://localhost:4000'):  
https://workstation:443
```

```
Enter value for chef.node_name (default: 'workstation'): admin
```

```
Enter value for chef.client_key (default: '/etc/chef/client.pem'):  
/vagrant/admin.pem
```

```
Enter value for chef.validation_client_name (default: 'chef-validator'):
```

```
Enter value for chef.validation_key_path (default: '/etc/chef/validation.pem'):  
/vagrant/chef-validator.pem
```

```
Enter value for vagrant.vm.box (default: 'Berkshelf-CentOS-6.3-x86_64-minimal'):  
HandsOnDevOps
```

```
Enter value for vagrant.vm.box_url (default: 'https://dl.dropbox.com/u/31081437/  
Berkshelf-CentOS-6.3-x86_64-minimal.box'):  
../../hands-on-dev-ops.box
```

```
Config written to: '/root/.berkshelf/config.json'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# cat Berksfile
```

```
site :opscode
```

```
metadata
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload
```

```
Using mycookbook (0.1.0) from metadata
```

```
Uploading mycookbook (0.1.0) to: 'https://workstation:443/'
```

```
Ridley::Errors::ConnectionFailed: SSL_connect returned=1 errno=0 state=SSLv3 read  
server certificate B: certificate verify failed
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim ~/.berkshelf/config.json
```

```
"ssl":{"verify":true}}    ->    "ssl":{"verify":false}}
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload
```

```
Using mycookbook (0.1.0) from metadata
```

```
Uploading mycookbook (0.1.0) to: 'https://workstation:443/'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim Berksfile
```

```
site :opscode
```

```
metadata
```

```
cookbook "apt"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
name          'mycookbook'
```

```
...
```

```
version       '0.1.0'
```

```
depends       'apt'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/apt.rb
```

```
execute "apt-get update" do...end
```

```
include_recipe "apt::default"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload --force
```

```
Using mycookbook (0.1.0) from metadata
```

```
Installing apt (2.3.4) from site: 'http://cookbooks.opscode.com/api/v1/cookbooks'
```

```
Uploading mycookbook (0.1.0) to: 'https://workstation:443/'
```

```
Uploading apt (2.3.4) to: 'https://workstation:443/'
```



```
$ vagrant provision server
...
[server] Running provisioner: chef_client...
...
INFO: Loading cookbooks [apt, mycookbook]
INFO: Storing updated cookbooks/apt/resources/preference.rb in the cache.
...
INFO: execute[apt-get-update] ran successfully
INFO: package[update-notifier-common] sending run action to execute[apt-get-update]
(immediate)
...
INFO: directory[/var/cache/local] created directory /var/cache/local
...
INFO: directory[/var/cache/local/preseeding] created directory /var/cache/local/
preseeding
...
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim Berksfile
```

```
site :opscode
```

```
metadata
```

```
cookbook "apt"
```

```
cookbook "mysql"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
name          'mycookbook'
```

```
...
```

```
depends        'apt'
```

```
depends        'mysql'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/mysql.rb
```

```
node.set['mysql']['server_root_password'] = ""
```

```
include_recipe "mysql::client"
```

```
include_recipe "mysql::server"
```

```
package "build-essential"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload --force
```

```
...
```

```
Uploading apt (2.3.4) to: 'https://workstation:443/'
```

```
Uploading mysql (4.0.14) to: 'https://workstation:443/'
```

```
Uploading openssl (1.1.0) to: 'https://workstation:443/'
```

```
Uploading build-essential (1.4.2) to: 'https://workstation:443/'
```

```
$ vagrant provision server
```

```
...
```

```
[server] Running provisioner: chef_client...
```

```
INFO: Loading cookbooks [apt, build-essential, mycookbook, mysql, openssl]
```

```
...
```

```
INFO: template[/etc/init/mysql.conf] updated file contents /etc/init/mysql.conf
```

```
...
```

```
INFO: service[mysql] reloaded
```

```
INFO: Chef Run complete in 1.460246852 seconds
```

```
INFO: Running report handlers
```

```
INFO: Report handlers complete
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim Berksfile
```

```
site :opscode
```

```
metadata
```

```
cookbook "apt"
```

```
cookbook "build-essential"
```

```
cookbook "mysql"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
name          'mycookbook'
```

```
...
```

```
depends        'apt'
```

```
depends        'build-essential'
```

```
depends        'mysql'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/mysql.rb
```

```
include_recipe "mysql::client"
```

```
include_recipe "mysql::server"
```

```
package "build-essential"
```

```
include_recipe "build-essential::default"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload --force
```

```
$ vagrant provision server
```

```
...
```

```
[server] Running provisioner: chef_client...
```

```
...
```

```
INFO: Loading cookbooks [apt, mycookbook]
```

```
INFO: Storing updated cookbooks/apt/resources/preference.rb in the cache.
```

```
...
```

```
INFO: execute[apt-get-update] ran successfully
```

```
INFO: package[update-notifier-common] sending run action to execute[apt-get-update]  
(immediate)
```

```
...
```

```
INFO: directory[/var/cache/local] created directory /var/cache/local
```

```
...
```

```
INFO: directory[/var/cache/local/preseeding] created directory /var/cache/local/  
preseeding
```

```
...
```

```
root@workstation:# cd /vagrant
```

```
root@workstation:/vagrant# knife data bag create users -z
```

```
WARNING: No knife configuration file found
```

```
Created data_bag[users]
```

```
root@workstation:# cat ./cookbooks/mycookbook/files/default/ssh/id_rsa.pub
```

```
root@workstation:/vagrant# vim ./data_bags/users/build.json
```

```
{  
  "id": "build",  
  "groups": ["sysadmin"],  
  "password": "$6$ECIy/pVovVQ6qja$A0owsw1LKsPPUQe2CKi....",  
  "ssh_keys": [  
    "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCbK2w... claco@hank"  
  ],  
  "shell": "/bin/bash"  
}
```

```
root@workstation:/vagrant# knife data bag from file users data_bags/users
```

```
Updated data_bag_item[users::build]
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim Berksfile
```

```
...
```

```
cookbook "mysql"
```

```
cookbook "users"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
...
```

```
depends          'mysql'
```

```
depends          'users'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/users.rb
```

```
include_recipe "users::sysadmins"
```

```
file "/etc/sudoers.d/build" do
```

```
  action :delete
```

```
end
```

```
template "/etc/sudoers.d/sysadmin" do
```

```
  source "sudo/user.erb"
```

```
  owner "root"
```

```
  group "root"
```

```
  mode "0440"
```

```
  variables :username => "%sysadmin"
```

```
end
```

```
root@workstation:/vagrant# cd cookbooks/mycookbook && berks upload --force
```

```
...
```

```
Uploading users (1.6.0) to: 'https://workstation:443/'
```

```
$ vagrant provision server
```

```
[server] Running provisioner: chef_client...
```

```
...
```

```
Loading cookbooks [apt, build-essential, mycookbook, mysql, openssl, users]
```

```
INFO: Storing updated cookbooks/mycookbook/recipes/users.rb in the cache.
```

```
...
```

```
INFO: template[/etc/sudoers.d/sysadmin] created file /etc/sudoers.d/sysadmin
```

```
...
```

```
$ vagrant ssh server
```

```
vagrant@server:~$ sudo cat /etc/sudoers.d/sysadmin
```

```
%sysadmin    ALL=(ALL) ALL
```

```
vagrant@server:~$ sudo tail /etc/group
```

```
sysadmin:x:2300:build
```



```
root@workstation:/vagrant/cookbooks/mycookbook# vim Berksfile
```

```
site :opscode
```

```
...
```

```
cookbook "build-essential"
```

```
cookbook "database"
```

```
cookbook "mysql"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
name                'mycookbook'
```

```
...
```

```
depends              'build-essential'
```

```
depends              'database'
```

```
depends              'mysql'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/myapp.rb
```

```
include_recipe "database::mysql"  
... remove "execute mysql" lines...
```

```
mysql_connection_info = {  
  :username => "root",  
  :password => node["mysql"]["server_root_password"],  
  :socket    => node["mysql"]["socket"]  
}
```

```
mysql_database "myapp" do  
  connection mysql_connection_info  
  action :create  
end
```

```
mysql_database_user "myapp" do  
  connection      mysql_connection_info  
  password        "password"  
  database_name   "myapp"  
  host            "localhost"  
  privileges      [:all]  
  action          [:create, :grant]  
end
```

```
root@workstation:/vagrant# cd cookbooks/mycookbook && berks upload --force
```

```
...
```

```
Uploading database (1.6.0) to: 'https://workstation:443/'
```

```
Uploading mysql (4.0.14) to: 'https://workstation:443/'
```

```
Uploading users (1.6.0) to: 'https://workstation:443/'
```

```
Uploading postgresql (3.3.4) to: 'https://workstation:443/'
```

```
Uploading openssl (1.1.0) to: 'https://workstation:443/'
```

```
Uploading aws (1.0.0) to: 'https://workstation:443/'
```

```
Uploading xfs (1.1.0) to: 'https://workstation:443/'
```

```
$ vagrant provision server
```

```
root@workstation:~# sudo -u build -i cd myapp
```

```
build@workstation:~/myapp$ cap deploy:migrate
```

```
* 2014-01-05 03:27:24 executing `deploy:migrate`
```

```
* executing "ls -x /var/www/myapp/releases"
```

```
  servers: ["192.168.33.20"]
```

```
Password:
```

```
* executing "cd /var/www/myapp/releases/20140104172547 && bundle exec rake  
RAILS_ENV=production db:migrate"
```

Part 6 Complete!

Part 7

Deploy Application (Pull) / Manage Secrets

```
root@workstation:/vagrant/cookbooks/mycookbook# Berksfile
```

```
site :opscode
```

```
...
```

```
cookbook "build-essential"
```

```
cookbook "database"
```

```
cookbook "git"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim metadata.rb
```

```
name                'mycookbook'
```

```
...
```

```
depends              'build-essential'
```

```
depends              'database'
```

```
depends              'git'
```

```
root@workstation:/vagrant/cookbooks/mycookbook# vim recipes/rails.rb \  
  recipes/myapp.rb
```

```
include_recipe "git::default"
```

```
root@workstation:/vagrant/cookbooks/mycookbook# berks upload --force
```

```
...
```

```
Uploading users (1.6.0) to: 'https://workstation:443/'
```

```
$ vagrant provision workstation && vagrant provision server
```

```
[server] Running provisioner: chef_client...
```

```
Loading cookbooks [apt, build-essential, git, mycookbook, mysql, openssl, users]
```

```
INFO: Storing updated cookbooks/mycookbook/recipes/users.rb in the cache.
```

```
$ vagrant ssh workstation
```

```
vagrant@workstation:~$ sudo -u build -i cd myapp
```

```
build@workstation:~/myapp$ git init .
```

```
Initialized empty Git repository in /home/build/myapp/.git/
```

```
build@workstation:~/myapp$ git add -A .
```

```
build@workstation:~/myapp$ git commit -m "Initial Commit"
```

```
[master (root-commit) 8b81977] Initial Commit
```

```
39 files changed, 918 insertions(+)
```

```
build@workstation:~/myapp$ sudo cp -Rv . /vagrant/myapp.git
```

```
`.' -> `/vagrant/myapp.git'
```

```
`./app' -> `/vagrant/myapp.git/app'
```

```
`./app/models' -> `/vagrant/myapp.git/app/models'
```

```
`./app/models/concerns' -> `/vagrant/myapp.git/app/models/concerns'
```

```
.....
```

```
$ vim ./cookbooks/mycookbook/recipes/myapp.rb
```

```
directory "/var/www/myapp/shared/config" do ... end
```

```
%w{config log bundle pids system cached-copy sockets}.each do |dir|
```

```
  directory "/var/www/myapp2/shared/#{dir}" do
```

```
    owner "build"
```

```
    group "build"
```

```
    mode "0755"
```

```
    recursive true
```

```
  end
```

```
end
```



```
template "/var/www/myapp2/shared/config/database.yml" do
...

template "/var/www/myapp2/shared/config/unicorn.rb" do
...
  variables :listen => "9000"
```

```

deploy_branch "/var/www/myapp2" do
  user "build"
  group "build"
  repo "file:///vagrant/myapp.git"
  shallow_clone true
  environment "RAILS_ENV" => "production"
  branch "master"
  action :force_deploy
  migrate true
  migration_command "bundle exec rake db:migrate > log/migrate.log 2>&1"
  #restart_command "/etc/init.d/unicorn-api upgrade"
  create_dirs_before_symlink %w{tmp public config deploy}
  symlink_before_migrate "config/database.yml" => "config/database.yml",
                        "config/unicorn.rb" => "config/unicorn.rb", "log" => "log"
  symlinks "system" => "public/system", "sockets" => "tmp/sockets", "pids"=> "tmp/
pids"
  before_migrate do
    execute "bundle install" do
      cwd release_path
      command "bundle install --path /var/www/myapp2/shared/bundle --deployment --
without development test"
    end
  end
end
end
end

```

Copy from HandsOnDevOps/examples/myapp2.rb

```
$ vagrant provision server
```

```
$ vagrant ssh server
```

```
vagrant@server:~$ cd /var/www/myapp2/current
```

```
vagrant@server:/var/www/myapp2/current$ ls -lta
```

```
lrwxrwxrwx  1 build build   26 Jan  5 18:05 log -> /var/www/myapp2/shared/log
```

```
vagrant@server:/var/www/myapp2/current$ ls -lta config
```

```
lrwxrwxrwx  1 build build   42 Jan  5 18:05 database.yml -> /var/www/myapp2/shared/  
config/database.yml
```

```
lrwxrwxrwx  1 build build   40 Jan  5 18:05 unicorn.rb -> /var/www/myapp2/shared/  
config/unicorn.rb
```

```
vagrant@server:~$ ps -aux | grep unicorn
```

```
build      4079  0.0  1.7  68208 17992 ?           Sl   18:42   0:00 unicorn_rails  
master -c /var/www/myapp2/current/config/unicorn.rb -E production -D
```

```
build      4082  4.4  6.0 242448 61160 ?           Sl   18:42   0:02 unicorn_rails  
worker[0] -c /var/www/myapp2/current/config/unicorn.rb -E production -D
```

The page you were looking for doesn't exist.

You may have mistyped the address or the page may have moved.

If you are the application owner check the logs for more information.

Browser: **http://192.168.33.20:9000**

```
$ mkdir .chef

$ openssl rand -base64 512 | tr -d '\r\n' > .chef/encrypted_data_bag_secret

$ echo ".chef" >> .gitignore

$ vim Vagrantfile

chef.encrypted_data_bag_secret_key_path = ".chef/encrypted_data_bag_secret"

$ vagrant provision workstation && vagrant provision server

[workstation] Running provisioner: chef_client...
Creating folder to hold client key...
Uploading chef client validation key...
Uploading chef encrypted data bag secret key...

root@workstation:~ EDITOR=vim knife data bag create passwords mysql \
    --secret-file /vagrant/.chef/encrypted_data_bag_secret -z

{
  "id": "mysql",
  "myapp": "password"
}
```

```
root@workstation:/vagrant# cat data_bags/passwords/mysql.json
```

```
{
  "id": "mysql",
  "myapp": {
    "encrypted_data": "ifq17q7z82jnFRr/Y2Zd+Sgvme7ga9k/xi8n0jRQM8s\n",
    "iv": "Xt6SuDTsUC5y59zN0qpHHQ==\n",
    "version": 1,
    "cipher": "aes-256-cbc"
  }
}
```

```
root@workstation:/vagrant# knife data bag create passwords
```

```
Created data_bag[passwords]
```

```
root@workstation:/vagrant# knife data bag from file passwords data_bags/passwords/
mysql.json
```

```
Updated data_bag_item[passwords::mysql]
```

```
$ vim ./cookbooks/mycookbook/recipes/myapp.rb
```

```
mysql_secret = Chef::EncryptedDataBagItem.load_secret("/vagrant/.chef/  
encrypted_data_bag_secret")
```

```
mysql_creds = Chef::EncryptedDataBagItem.load("passwords", "mysql", mysql_secret)
```

```
myapp_password = mysql_creds["myapp"]
```

```
mysql_database_user "myapp" do
```

```
  connection      mysql_connection_info
```

```
  password        myapp_password
```

```
  variables(  
    :environment => "production",  
    ...  
    :username => "myapp",  
    :password => myapp_password
```

```
root@workstation:~# knife cookbook upload -o /vagrant/cookbooks mycookbook --force
```

```
$ vagrant provision server
```

Part 7 Complete!

Part 8

TO THE CLOUD!

<http://developer.rackspace.com/devtrial/>