# The Joy of Cooking

## Whip up a Rails Environment with Chef

Nathen Harvey, CustomInk.com

@nathenharvey

https://github.com/nathenharvey/cooking-with-chef

CustomInk
Design T-shirts Online

## Agenda

- Infrastructure as Code
- Introduction to Chef
- Building a project in Chef
- Provision a server for your Rails App
- Additional resources

## Infrastructure as Code

- Enable the reconstruction of the business from nothing but
  - a source code repository
  - an application data backup
  - and bare metal resources

-Jesse Robins, Opscode

## Disposable Servers

CustomInk Design T-shirts Online

@nathenharvey

## Evolution of Server Provisioning

- Just build it
- Keep notes in server.txt
- Migrate notes to wiki
- Custom shell scripts (in git)
- Systems integration framework

## When should I use a systems integration framework?

**After** you outgrow Heroku

## Which framework?

CFEngine?

Puppet?

Chef?

## Wrong question!

YES - use a systems integration framework

YES - use one that works for your team

YES - this is a talk about **Chef**

(Why we chose Chef @CustomInk)

## Chef

- Declarative - What, not how
- Idempotent - Only take action if required
- Convergent - Takes care of itself

## Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec

## Resources

```
package "tmux" do
  action :install
end

directory "/u/apps/awesome" do
  owner "apache"
  group "apache"
  action :create
  recursive true
end
```

More resources...

## Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec
- Write **recipes**

## Recipes

```
include_recipe "app_user"

app_name = node["app_name"]
app_user = node["app_user"]
app_group = node["app_group"]

%w(releases shared).each do |dir|
  directory "/u/apps/#{app_name}/#{dir}" d
    mode "0755"
    owner app_user
    group app_group
    recursive true
  end
end
```

## Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec
- Write **recipes**
- Package recipes in **cookbooks**

## Cookbooks

```
|-- ldirectord
|   |-- README.md
|   |-- attributes
|   |   `-- default.rb
|   |-- metadata.rb
|   |-- recipes
|   |   `-- default.rb
|   `-- templates
|       `-- default
|           `-- site.cf.erb
```

CustomInk
Design T-shirts Online

@nathenharvey

# Cookbooks

```
|-- monit
|    |-- README.rdoc
|    |-- attributes
|    |    `-- default.rb
|    |-- files
|    |    `-- ubuntu
|    |         `-- monit.default
|    |-- metadata.rb
|    |-- recipes
|    |    `-- default.rb
|    `-- templates
|         `-- default
|              `-- monitrc.erb
|
```

# Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec
- Write **recipes**
- Package recipes in **cookbooks**
- Apply recipes to **nodes**

## Nodes

- Representation of a host
  - runs the Chef client
  - has attributes
  - has a list of recipes to be applied

## Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec
- Write **recipes**
- Package recipes in **cookbooks**
- Apply recipes to **nodes**
- Group things into **roles**

## Roles

- mechanism for easily composing sets of functionality
- have attributes and a list of recipes to be applied

# Roles

```
name "base"
description "Base of all nodes"
default_attributes(
  "newrelic" => {
    "license_key" => "cbb1f5..."
  }
)

run_list(
  "recipe[base_config]",
  "recipe[users]",
  "recipe[groups]",
  "recipe[sudo]"
)
```

CustomInk Design T-shirts Online

@nathenharvey

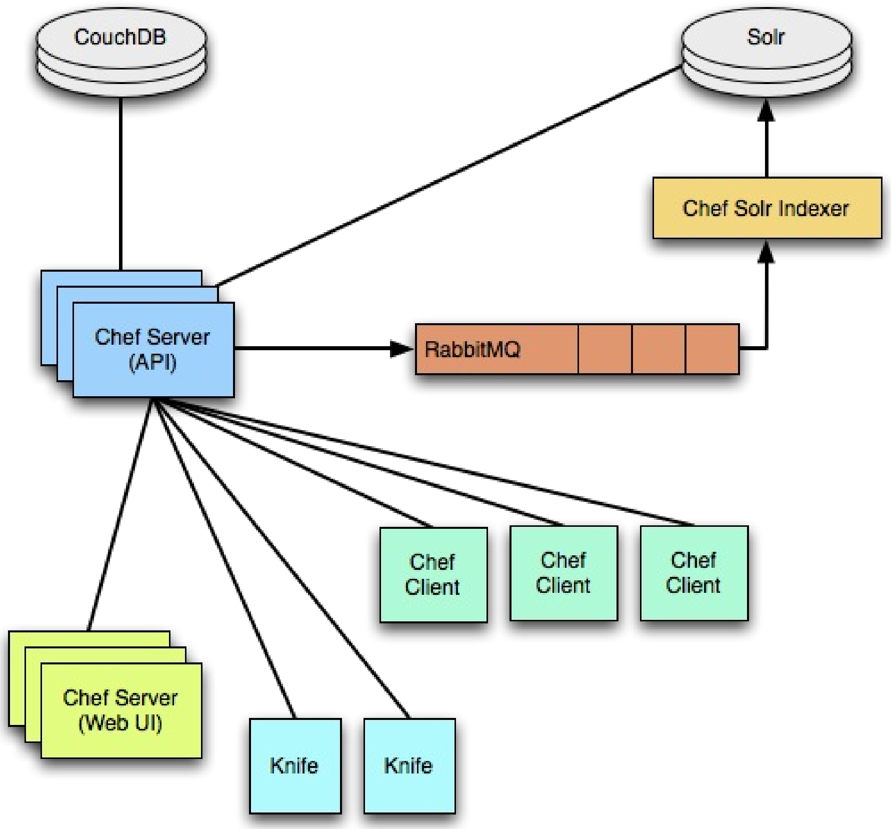
## Building a Chef Project

- First, come up with your specification
- Abstract the **resources** in your spec
- Write **recipes**
- Package recipes in **cookbooks**
- Apply recipes to **nodes**
- Group things into **roles**

CustomInk
Design T-shirts Online

@nathenharvey

# What is Chef?


CouchDB
Solr
Chef Solr Indexer
Chef Server (API)
RabbitMQ
Chef Client
Chef Client
Chef Client
Chef Server (Web UI)
Knife
Knife

## What is Chef?

- Server - API, search, Web UI
- Client - chef-client
- Command line tool - knife

# knife

```
$ knife help list
bootstrap
client
configure
cookbook
cookbook-site
data-bag
environment
exec
index
knife
node
role
search
shef
ssh
status
tag
```

## What is Chef?

- Server - API, search, Web UI
- Client - chef-client
- Command line tool - knife
- Inspection library - ohai

# ohai

Collects detailed, extensible information about a host.

```
{
  "uptime": "13 days 06 hours 16 minutes 0
  "platform": "ubuntu",
  "os_version": "2.6.32-38-generic",
  "cpu": {
    "total": 3,
    "real": 0,
    "2": {
      "cache_size": "4096 KB",
      "model": "2",
      "family": "6",
...
```

## What is Chef?

- Server - API, search, Web UI
- Client - chef-client
- Command line tool - knife
- Inspection library - ohai
- REPL - shef

## shef

The (poorly named) Chef REPL

```
$ shef -a
Ohai2u nharvey@nathenharvey.local!
```

## shef

```
chef > attributes
chef:attributes > set["shef_example"] = "He
 => "Hello, DCRUG!"
chef:attributes > quit
 => :attributes
```

CustomInk Design T-shirts Online

@nathenharvey

## shef

```
chef > recipe
chef:recipe > echo off
chef:recipe > file "/tmp/hello" do
chef:recipe >     content node.shef_exampl
chef:recipe ?>    mode "0777"
chef:recipe ?>    action :create
chef:recipe ?>  end
```

CustomInk
Design T-shirts Online

@nathenharvey

## shef

```
chef:recipe > run_chef
[Thu, 15 Mar 2012 12:11:02 -0400] DEBUG: P
[Thu, 15 Mar 2012 12:11:02 -0400] INFO: Pro
[Thu, 15 Mar 2012 12:11:02 -0400] INFO: fi
[Thu, 15 Mar 2012 12:11:02 -0400] INFO: fi
chef:recipe > exit
 => :recipe
chef > exit
```

CustomInk Design T-shirts Online

@nathenharvey

## shef

```
$ cat /tmp/hello
Hello, DCRUG!
```

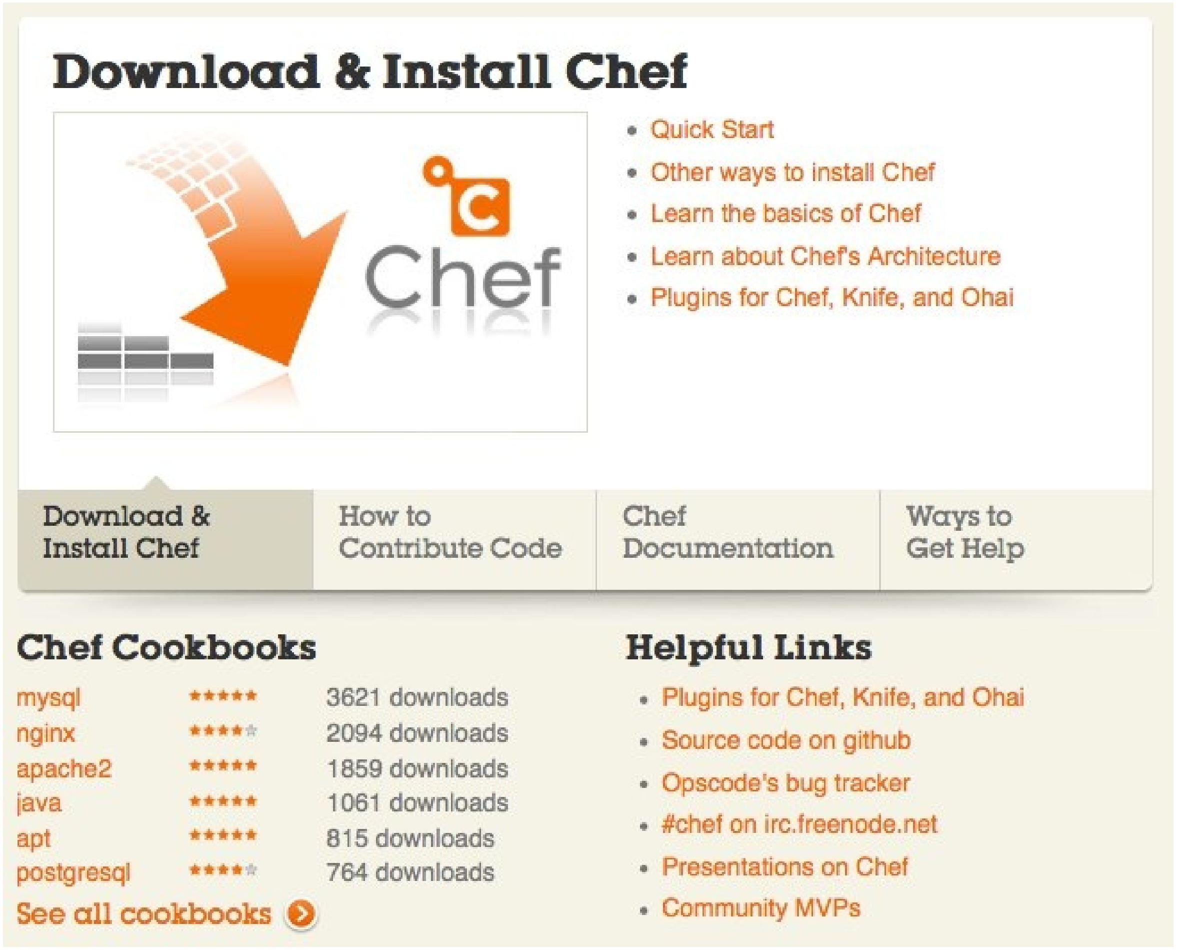
## What is Chef?

- Server - API, search, Web UI
- Client - chef-client
- Command line tool - knife
- Inspection library - ohai
- REPL - shef
- Community

CustomInk Design T-shirts Online

@nathenharvey

# community.opscode.com



## Download & Install Chef

- Quick Start
- Other ways to install Chef
- Learn the basics of Chef
- Learn about Chef's Architecture
- Plugins for Chef, Knife, and Ohai

## Chef Cookbooks

| | | |
|---|---|---|
| mysql | ★★★★★ | 3621 downloads |
| nginx | ★★★★☆ | 2094 downloads |
| apache2 | ★★★★★ | 1859 downloads |
| java | ★★★★★ | 1061 downloads |
| apt | ★★★★★ | 815 downloads |
| postgresql | ★★★★☆ | 764 downloads |

See all cookbooks

## Helpful Links

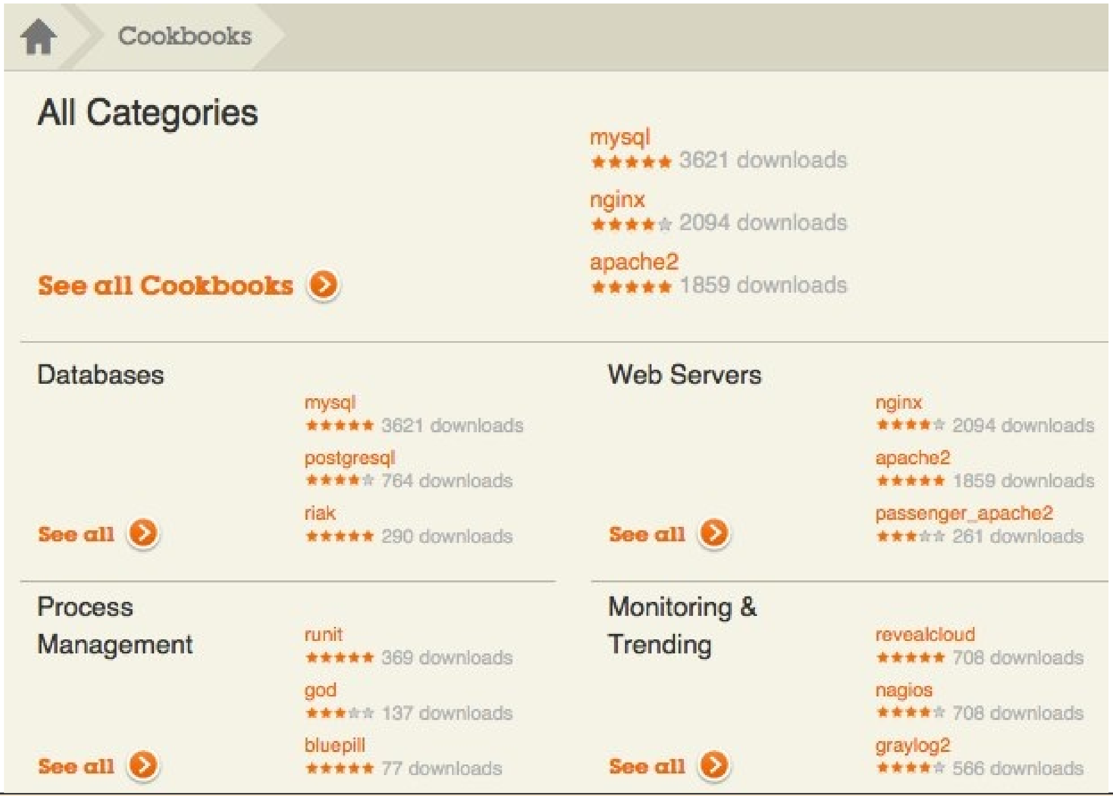
- Plugins for Chef, Knife, and Ohai
- Source code on github
- Opscode's bug tracker
- #chef on irc.freenode.net
- Presentations on Chef
- Community MVPs

CustomInk Design T-shirts Online

@nathenharvey

## Community Site

- Publish and share cookbooks



Cookbooks

All Categories

mysql
★★★★★ 3621 downloads

nginx
★★★★☆ 2094 downloads

apache2
★★★★★ 1859 downloads

See all Cookbooks

Databases

mysql
★★★★★ 3621 downloads

postgresql
★★★★☆ 764 downloads

riak
★★★★★ 290 downloads

See all

Web Servers

nginx
★★★★☆ 2094 downloads

apache2
★★★★★ 1859 downloads

passenger_apache2
★★★☆☆ 261 downloads

See all

Process Management

runit
★★★★★ 369 downloads

god
★★★☆☆ 137 downloads

bluepill
★★★★★ 77 downloads

See all

Monitoring & Trending

revealcloud
★★★★★ 708 downloads

nagios
★★★★☆ 708 downloads

graylog2
★★★★☆ 566 downloads

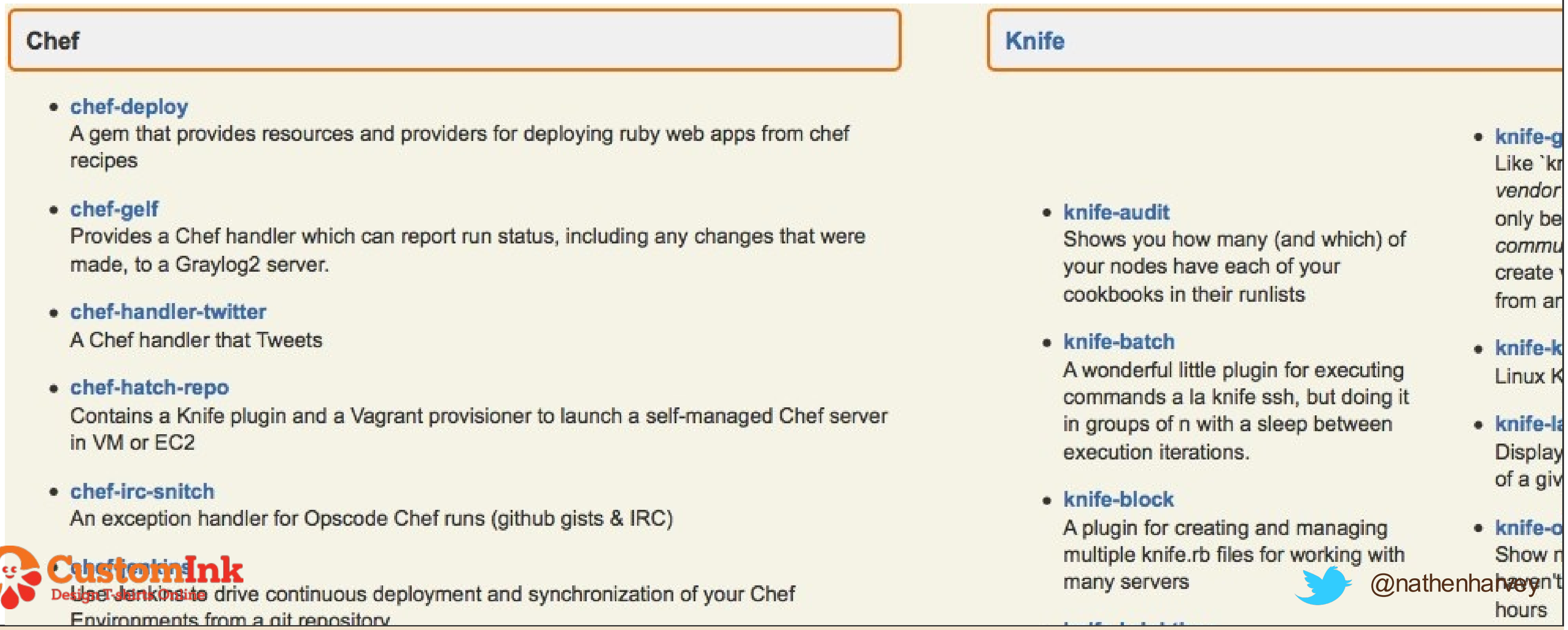
See all

CustomInk Design T-shirts Online

@nathenharvey

# Community Site

- Publish and share plugins for Chef, Knife, and Ohai



**Chef**

- **chef-deploy**
  A gem that provides resources and providers for deploying ruby web apps from chef recipes
- **chef-gelf**
  Provides a Chef handler which can report run status, including any changes that were made, to a Graylog2 server.
- **chef-handler-twitter**
  A Chef handler that Tweets
- **chef-hatch-repo**
  Contains a Knife plugin and a Vagrant provisioner to launch a self-managed Chef server in VM or EC2
- **chef-irc-snitch**
  An exception handler for Opscode Chef runs (github gists & IRC)
- **chef-jenkins**
  Use Jenkins to drive continuous deployment and synchronization of your Chef Environments from a git repository

**Knife**

- **knife-audit**
  Shows you how many (and which) of your nodes have each of your cookbooks in their runlists
- **knife-batch**
  A wonderful little plugin for executing commands a la knife ssh, but doing it in groups of n with a sleep between execution iterations.
- **knife-block**
  A plugin for creating and managing multiple knife.rb files for working with many servers

- **knife-g**
  Like `kr
  *vendor*
  only be
  *commu*
  create
  from ar
- **knife-k**
  Linux K
- **knife-la**
  Display
  of a giv
- **knife-o**
  Show n
  haven't
  hours

## Chef Deployment Options

- chef-solo
- Chef Server
- Hosted Chef
- Private Chef

## chef-solo

- What you don't get:
  - central server
  - authentication
  - authorization
  - search indexes
  - persistent attributes

## chef-solo

Execute cookbooks that are stored on disk or available at at URL

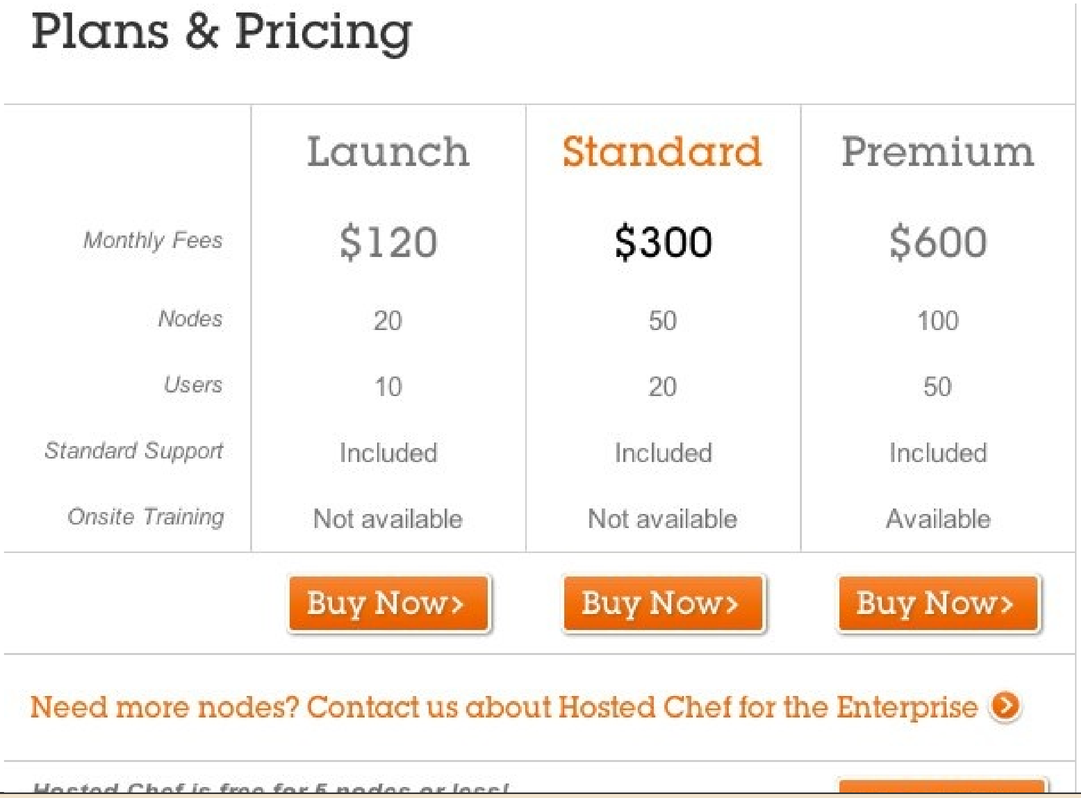
```
chef-solo -c ~/solo.rb  \
  -j ~/node.json \
  -r http://foo.com/chef-solo.tar.gz
```

# Chef Server

- Open source
- Run it yourself, wherever you like
- Complicated to set-up and manage

# Hosted Chef

- Best way to get started



Plans & Pricing

| | Launch | Standard | Premium |
|---|---|---|---|
| *Monthly Fees* | $120 | $300 | $600 |
| *Nodes* | 20 | 50 | 100 |
| *Users* | 10 | 20 | 50 |
| *Standard Support* | Included | Included | Included |
| *Onsite Training* | Not available | Not available | Available |
| | Buy Now› | Buy Now› | Buy Now› |

Need more nodes? Contact us about Hosted Chef for the Enterprise

Hosted Chef is free for 5 nodes or less!

CustomInk Design T-shirts Online

@nathenharvey

## Private Chef

- Commercial offering
- Managed by Opscode
- Pricing
  - installation fee
  - service contract

## Get started with Hosted Chef

1. Create a Hosted Chef account
2. Install and Update dependencies - ruby, ruby gems, ruby-dev and git-core
3. Install Chef and create directories needed
4. Connect to Hosted Chef

## chef-repo directory

```
chef-repo
|-- .chef
|   |-- knife.rb
|   |-- dcrug-validator.pem
|   `-- dcrug.pem
|-- README.md
|-- Rakefile
|-- certificates
|-- config
|   `-- rake.rb
|-- cookbooks
|-- data_bags
|-- environments
`-- roles
```

## Verify setup

```
$ knife client list
dcrug-validator
```

CustomInk Design T-shirts Online

@nathenharvey

# Provision a server

- knife ec2 server create
- knife rackspace server create
- Vagrant

# Vagrantfile

```
VagranConfit::lonfig.run do |config|
  config.vm.box_url = "https://s3.amazonaw
  config.vm.box = "ubuntu64-10.0.4-ruby-1.

  config.vm.forward_port 80, 8080

  config.vm.provision :chef_client do |che
    chef.chef_server_url = "https://api.op
    chef.validation_key_path = "chef-repo/
    chef.validation_client_name = "dcrug-v
    chef.node_name = "dcrug.local"
  end
end
```

CustomInk
Design T-shirts Online

@nathenharvey

## Launch Vagrant & Check In

```
$ vagrant up
READY

$ knife node list
dcrug.local
```

## Initial set-up steps

- Register with hosted chef
- Create a chef-repo
- Install chef
- Configure knife.rb
- Configure Vagrant file
- Register Vagrant instance with Chef hosted

## Provision for Rails

- Apache
- Passenger
- MySQL
- Rails application

## Add passenger_apache2 cookbook

```
$ knife cookbook site install passenger_apache2
```

## Add mysql cookbook

```
$ knife cookbook site install mysql
```

CustomInk
Design T-shirts Online

@nathenharvey

## Create a Cookbook

```
$ knife cookbook create dcrug

** Creating cookbook dcrug
** Creating README for cookbook: dcrug
** Creating CHANGELOG for cookbook: dcrug
** Creating metadata for cookbook: dcrug
```

## Write our recipes

- default.rb
- web.rb
- db.rb

## Web Recipe

Set-up some directories

```
%w(releases shared shared/system shared/pi
  directory "#{deploy_to}/#{app_name}/#{di
    action :create
    owner app_user
    group app_group
    mode "0664"
    recursive true
  end
end
```

CustomInk
Design T-shirts Online

@nathenharvey

# Web Recipe

Configure Apache / Passenger

```
web_app app_name do
  docroot "#{deploy_to}/current/public"
  server_name "#{app_name}.#{node["domain"]
  server_aliases [ app_name, "localhost", 
  rails_env "production"
end
```

## Database Recipe

Create the database

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

mysql_database app_name do
  connection mysql_connection_info
  action :create
end
```

CustomInk Design T-shirts Online

@nathenharvey

## Database Recipe

Create the database user

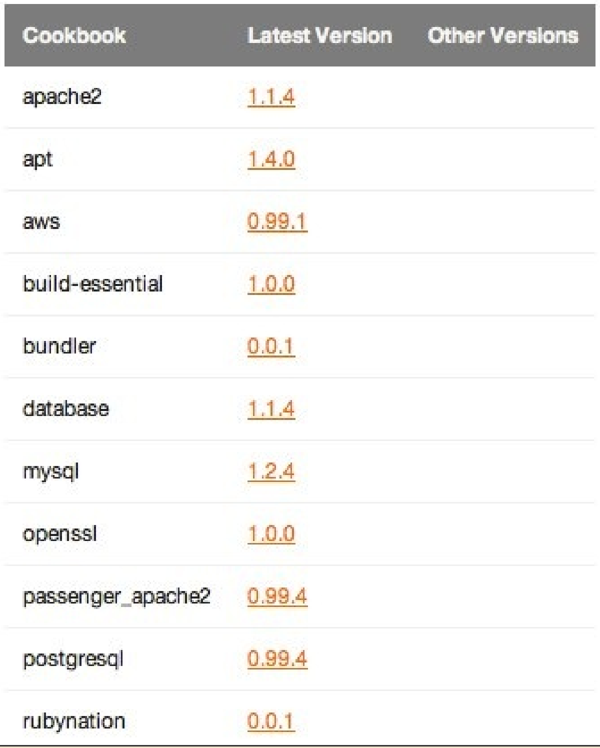
```
mysql_database_user node["database"]["user
  connection mysql_connection_info
  password  node["database"]["pw"]
  database_name node["database"]["name"]
  host "%"
  action :grant
end
```

CustomInk
Design T-shirts Online

@nathenharvey

## Upload cookbooks to Chef server

```
$ knife cookbook upload -a
```



| Cookbook | Latest Version | Other Versions |
|---|---|---|
| apache2 | 1.1.4 | |
| apt | 1.4.0 | |
| aws | 0.99.1 | |
| build-essential | 1.0.0 | |
| bundler | 0.0.1 | |
| database | 1.1.4 | |
| mysql | 1.2.4 | |
| openssl | 1.0.0 | |
| passenger_apache2 | 0.99.4 | |
| postgresql | 0.99.4 | |
| rubynation | 0.0.1 | |

## Create some roles

- Group recipes together using roles
- Apply roles to nodes
- Our roles:
  - base_ubuntu
  - dcrug_web
  - dcrug_db

## base_ubuntu Role

```
name "base_ubuntu"
description "all Ubuntu servers"
run_list(
  "recipe[apt]",
  "recipe[build-essential]"
)
```

## dcrug_web Role

```
name "dcrug_web"
description "DCRUG Webserver nodes"
run_list(
  "recipe[dcrug::web]"
)
```

## dcrug_db Role

```
name "dcrug_db"
description "DCRUG Database nodes"
run_list(
  "recipe[dcrug::db]"
)
```

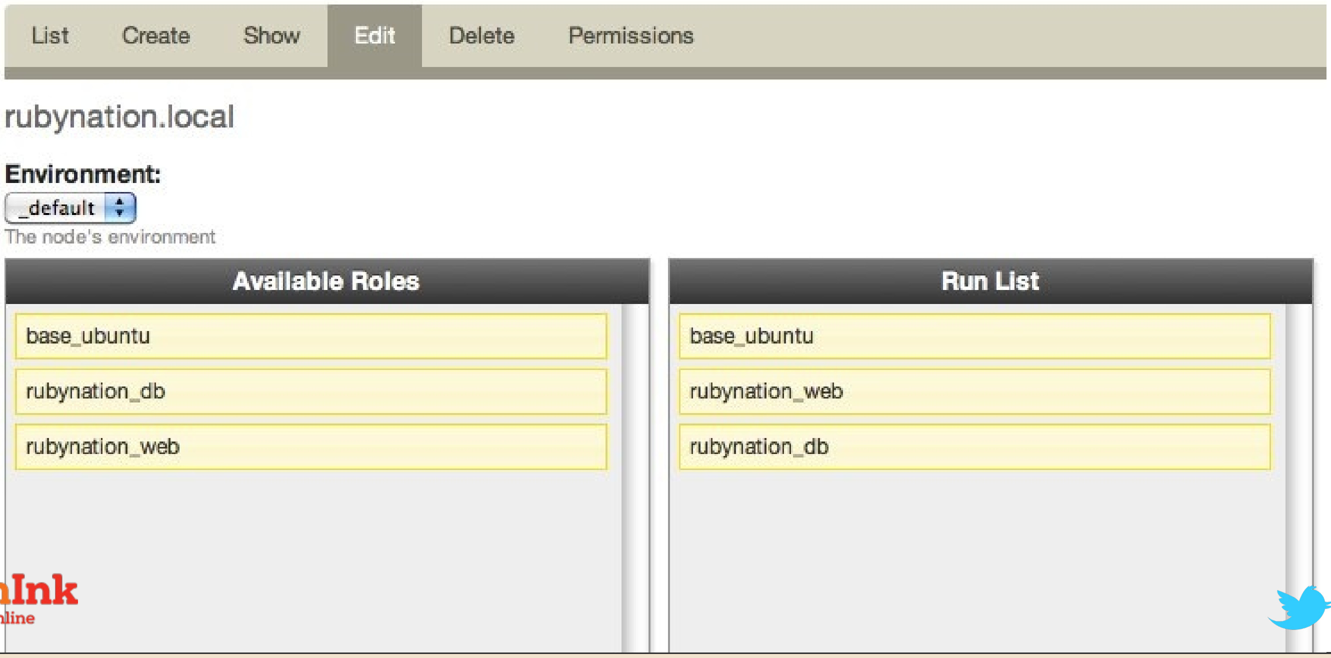
CustomInk
Design T-shirts Online

## Upload the roles to the server

```
$ knife role from file roles/base_ubuntu.rb
$ knife role from file roles/dcrug_web.rb
$ knife role from file roles/dcrug_db.rb
```

# Assign the roles to our nodes

```
$ knife node run_list add dcrug.local "role[base_ubuntu]"
$ knife node run_list add dcrug.local "role[dcrug_web]"
$ knife node run_list add dcrug.local "role[dcrug_db]"
```



rubynation.local

**Environment:**
_default
The node's environment

| Available Roles | Run List |
|---|---|
| base_ubuntu | base_ubuntu |
| rubynation_db | rubynation_web |
| rubynation_web | rubynation_db |

## Run chef-client

- Automatically
- `knife ssh`
- `vagrant provision`

## Review

- Server provisioned and communicating with the Chef API
- Apache and Passenger installed with a default configuration
- MySQL installed and running

## Deploying with Capistrano

Without Chef:

```
role :web, "web01","web02","web03"
```

## Deploying with Capistrano

With Chef search

```
webservers = []
web_query = Chef::Search::Query.new
web_query.search(:node,
            'role:dcrug_web') do |h|
  websevers << h["fqdn"]
end

role :web, *webservers
```

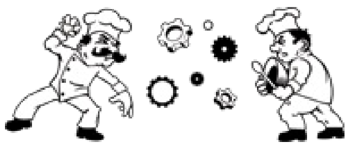
# But wait, there's more!

- Encrypted databags
- Environments
- Lightweight Resources and Providers (LWRP)
- Exception and report handlers

## Want more?

- http://community.opscode.com
- http://wiki.opscode.com
- Opscode Training Materials
- #chef on irc.freenode.net

## Want even more?

- http://foodfightshow.org 
  - Episode 5: Getting Started with Chef
- Visual Introduction to Chef

## Shameless Plugs

- DevOpsDC
- Washington DC MongoDB Users Group
- CustomInk Friday Tech Lunch

## Find Me

- @nathenharvey
- http://nathenharvey.com
- ~~nharvey@customink.com~~
- nathen.harvey@gmail.com
- Cooking with Chef, this presentation