Chef Fundamentals

training@getchef.com

Nathen Harvey

- Community Director
- Co-host of the Food Fight Show Podcast

foodfightshow.org
FOOD FIGHT

- @nathenharvey

Webinar Objectives and Style

Multi-week Webinar Series

- After completing of this webinar series you will be able to
  - Automate common infrastructure tasks with Chef
  - Describe Chef’s architecture
  - Describe Chef’s various tools
  - Apply Chef’s primitives to solve your problems

How to learn Chef

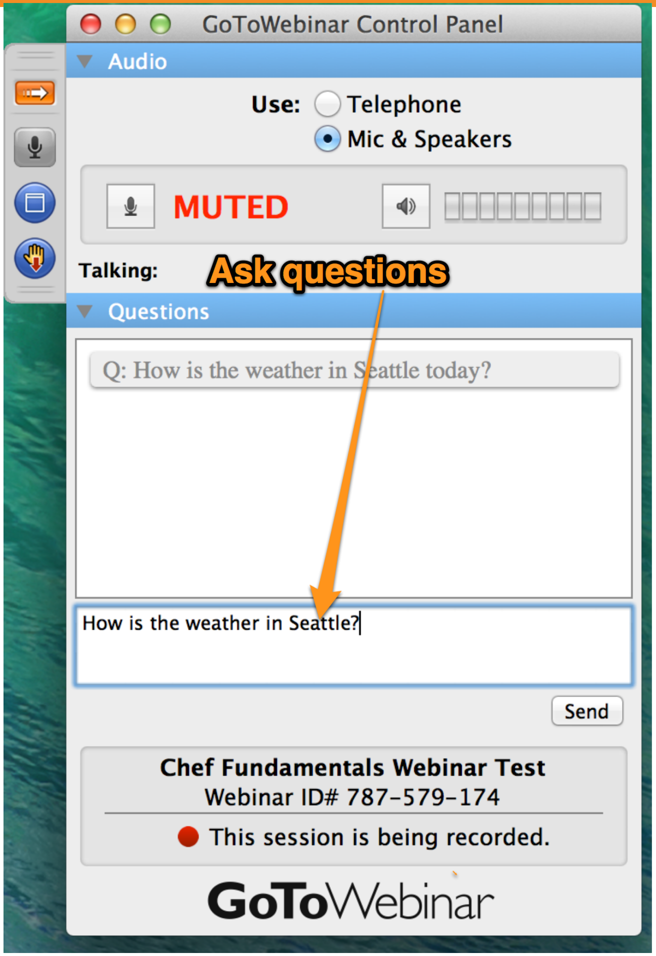
- You bring the domain expertise about your business and infrastructure
- Chef provides a framework for automating your infrastructure
- Our job is to work together to teach you how to model and automate your infrastructure with Chef

Chef is a Language

- Learning Chef is like learning the basics of a language
- 80% fluency will be reached very quickly
- The remaining 20% just takes practice
- The best way to **learn** Chef is to ***use*** Chef

Questions & Answers

- Ask questions in the chat window when they come to you
    - We'll answer as many questions as we can at the end of the session



Q: How is the weather in Seattle today?

How is the weather in Seattle?

Chef Fundamentals Webinar Test
Webinar ID# 787-579-174
This session is being recorded.

Slides and Video

- This webinar is being recorded. The video will be made available shortly after the session has ended.
- The slides used throughout this webinar will be made available at the end of each webinar.
- Watch http://learnchef.com for updates.

Agenda

Topics

- ~~Overview of Chef~~
- ~~Workstation Setup~~
- Node Setup - **Today**
- Chef Resources and Recipes - **Today**
- Working with the Node object
- Common configuration with Data Bags
- Roles and Environments
- Community Cookbooks and Further Resources

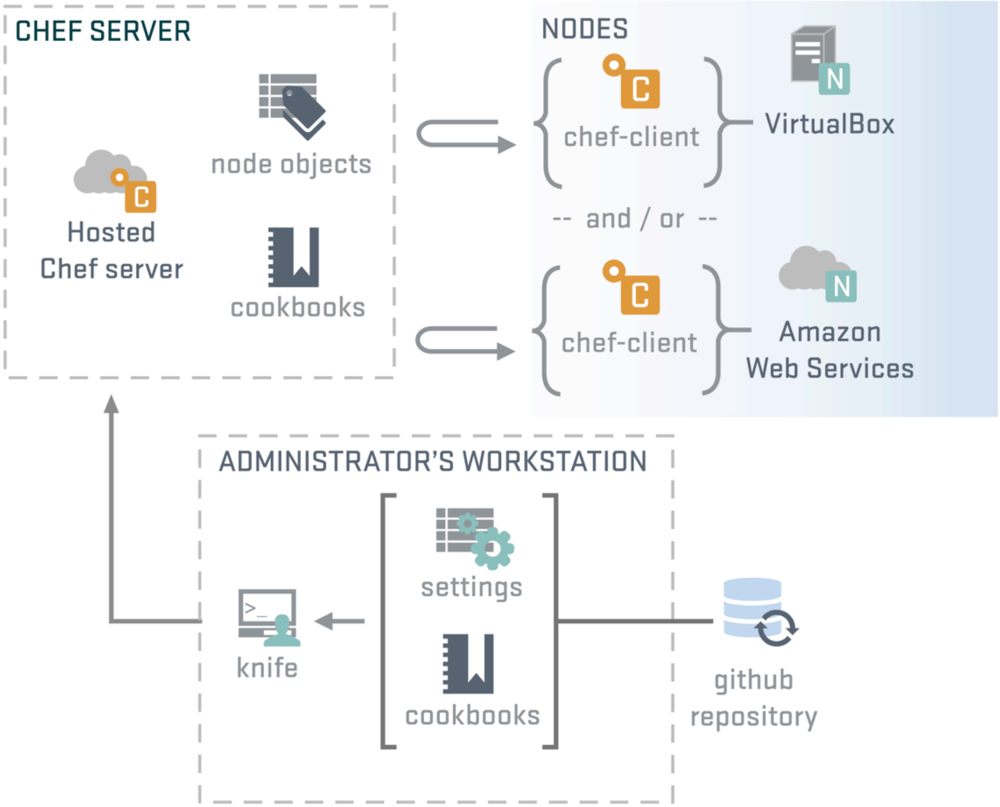
Node Setup

Setup a Node to manage

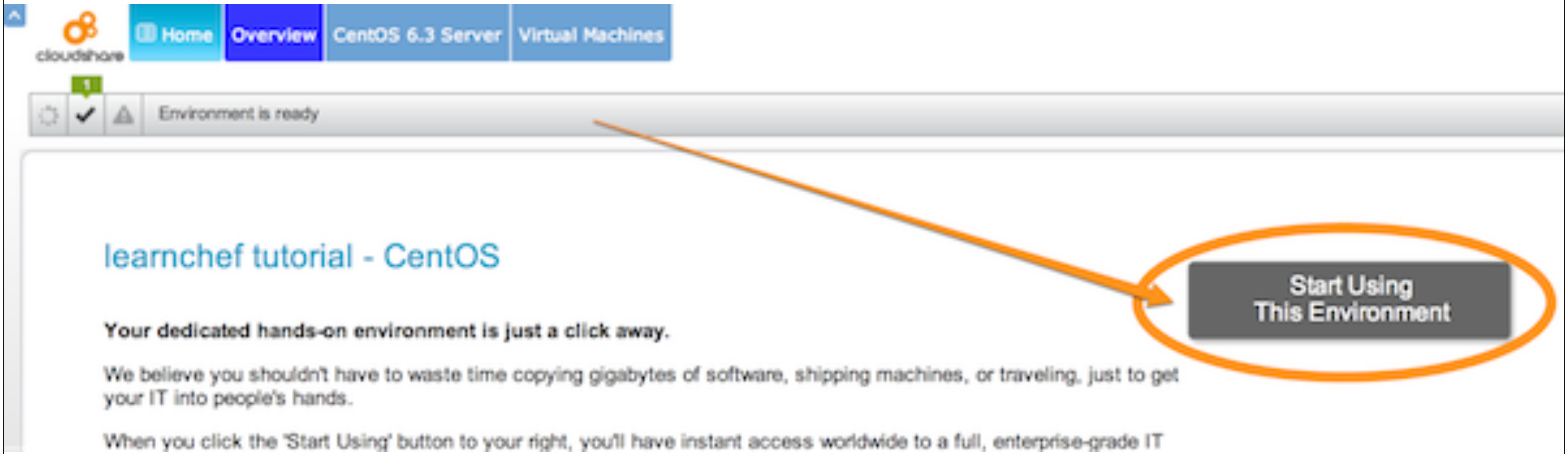
Lesson Objectives

- After completing the lesson, you will be able to
  - Login to the node in your Chef Training Lab
  - Install Chef nodes using "knife bootstrap"
  - Explain how knife bootstrap configures a node to use the Organization created in the previous section
  - Explain the basic configuration needed to run chef-client

Chef Infrastructure



Launch Chef Training Lab



Environment is ready

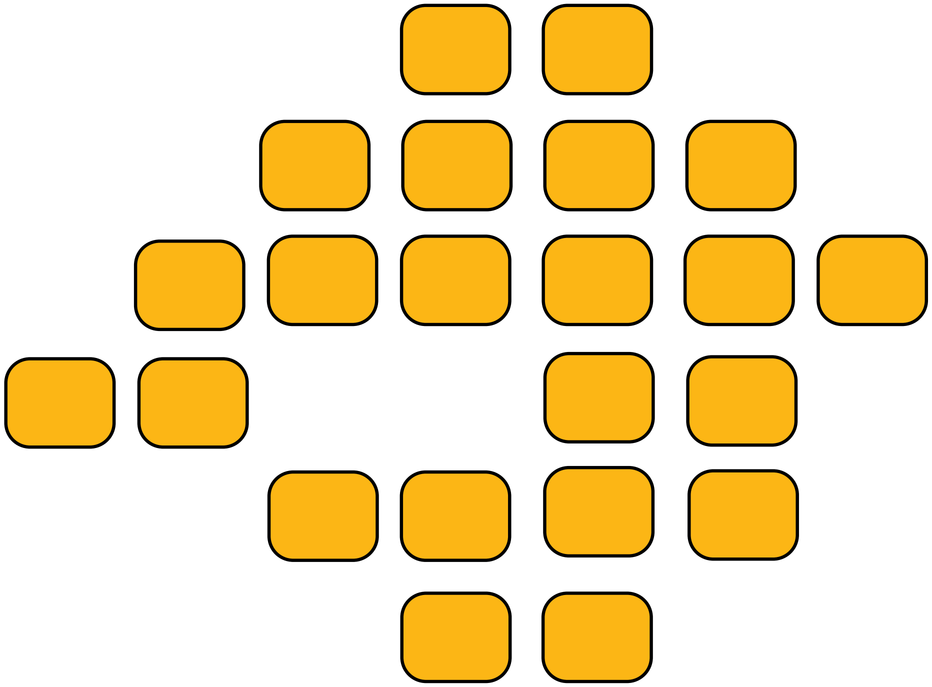
learnchef tutorial - CentOS

Your dedicated hands-on environment is just a click away.

We believe you shouldn't have to waste time copying gigabytes of software, shipping machines, or traveling, just to get your IT into people's hands.

When you click the 'Start Using' button to your right, you'll have instant access worldwide to a full, enterprise-grade IT

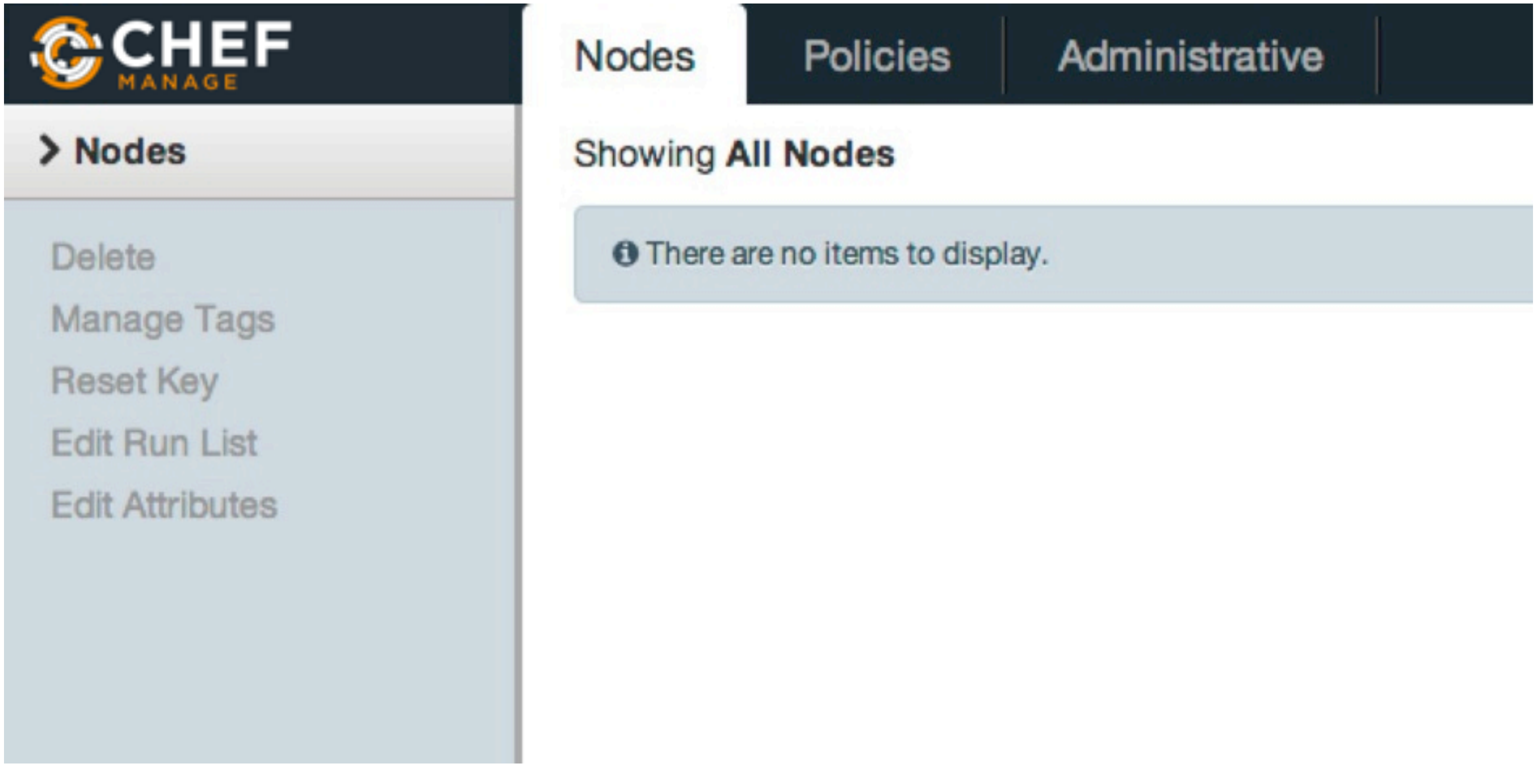
Nodes



Nodes

- Nodes represent the servers in your infrastructure these may be
  - Physical or virtual servers
  - Hardware that you own
  - Compute instances in a public or private cloud

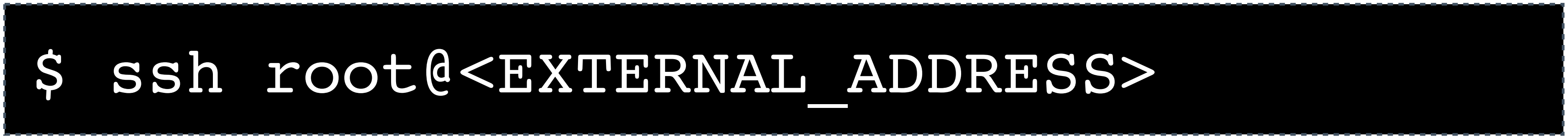
We Have No Nodes Yet



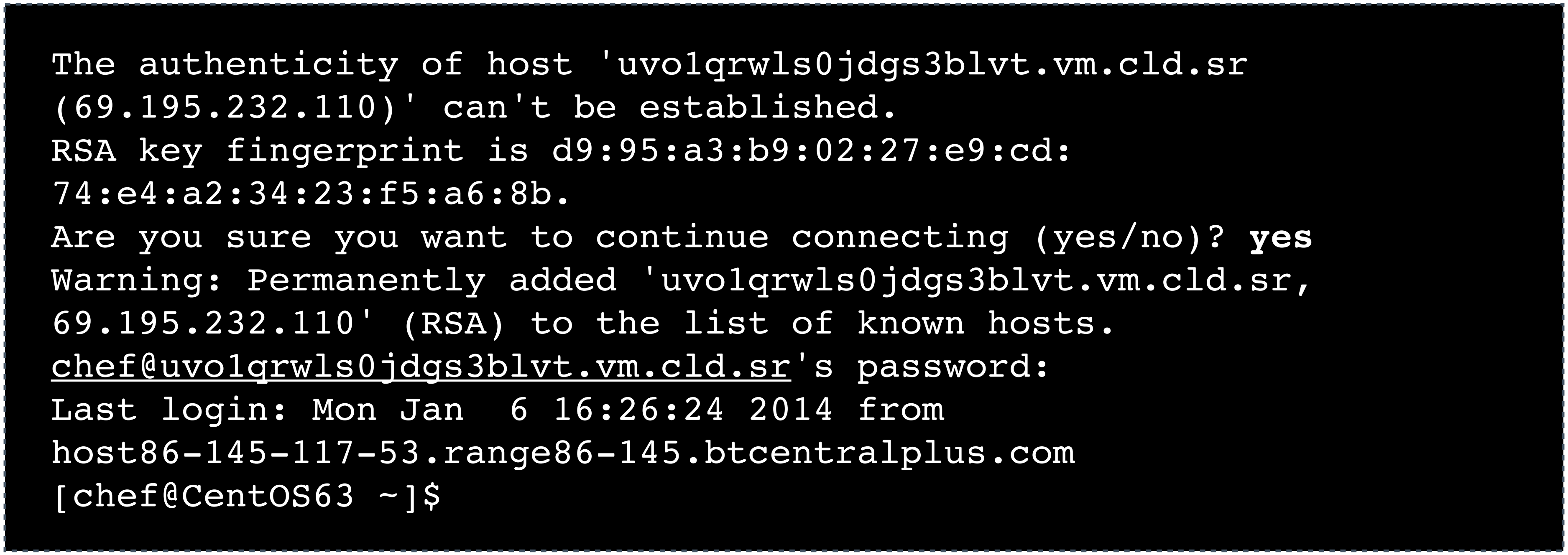
Showing **All Nodes**

There are no items to display.

Lab - Login



```
$ ssh root@<EXTERNAL_ADDRESS>
```

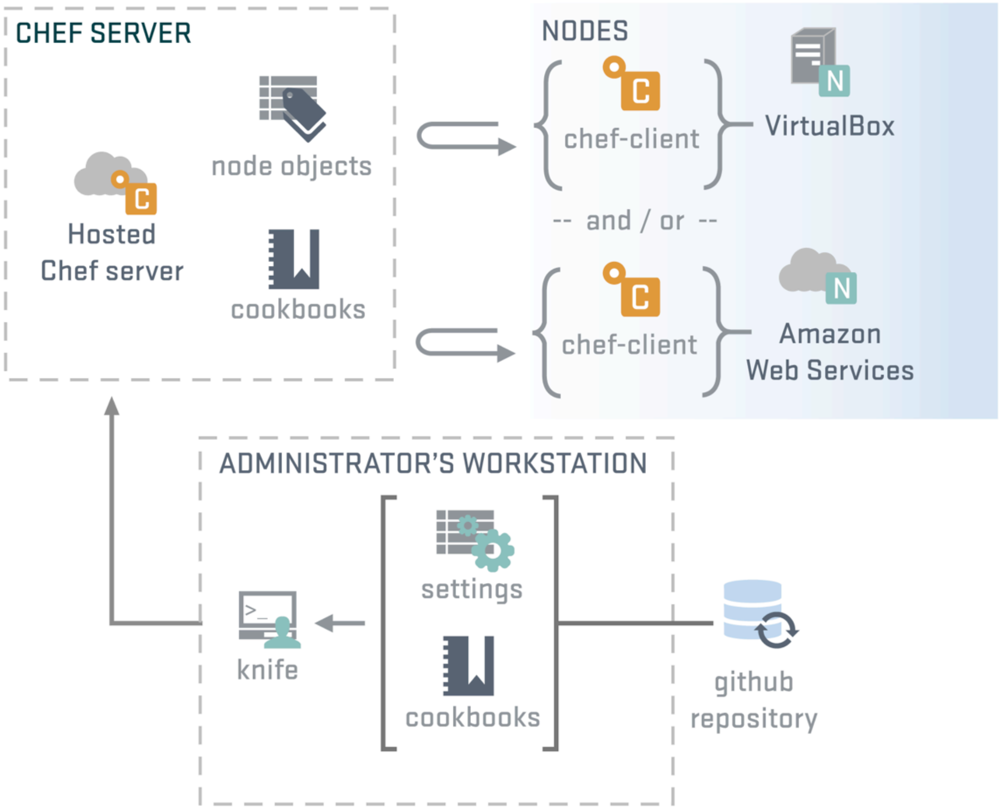


```
The authenticity of host 'uvo1qrwls0jdgs3blvt.vm.cld.sr
(69.195.232.110)' can't be established.
RSA key fingerprint is d9:95:a3:b9:02:27:e9:cd:
74:e4:a2:34:23:f5:a6:8b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'uvo1qrwls0jdgs3blvt.vm.cld.sr,
69.195.232.110' (RSA) to the list of known hosts.
chef@uvo1qrwls0jdgs3blvt.vm.cld.sr's password:
Last login: Mon Jan  6 16:26:24 2014 from
host86-145-117-53.range86-145.btcentralplus.com
[chef@CentOS63 ~]$
```

Checkpoint

- At this point you should have
  - One virtual machine (VM) or server that you'll use for the lab exercises
  - The IP address or public hostname
  - An application for establishing an ssh connection
  - 'sudo' or 'root' permissions on the VM

Chef Infrastructure

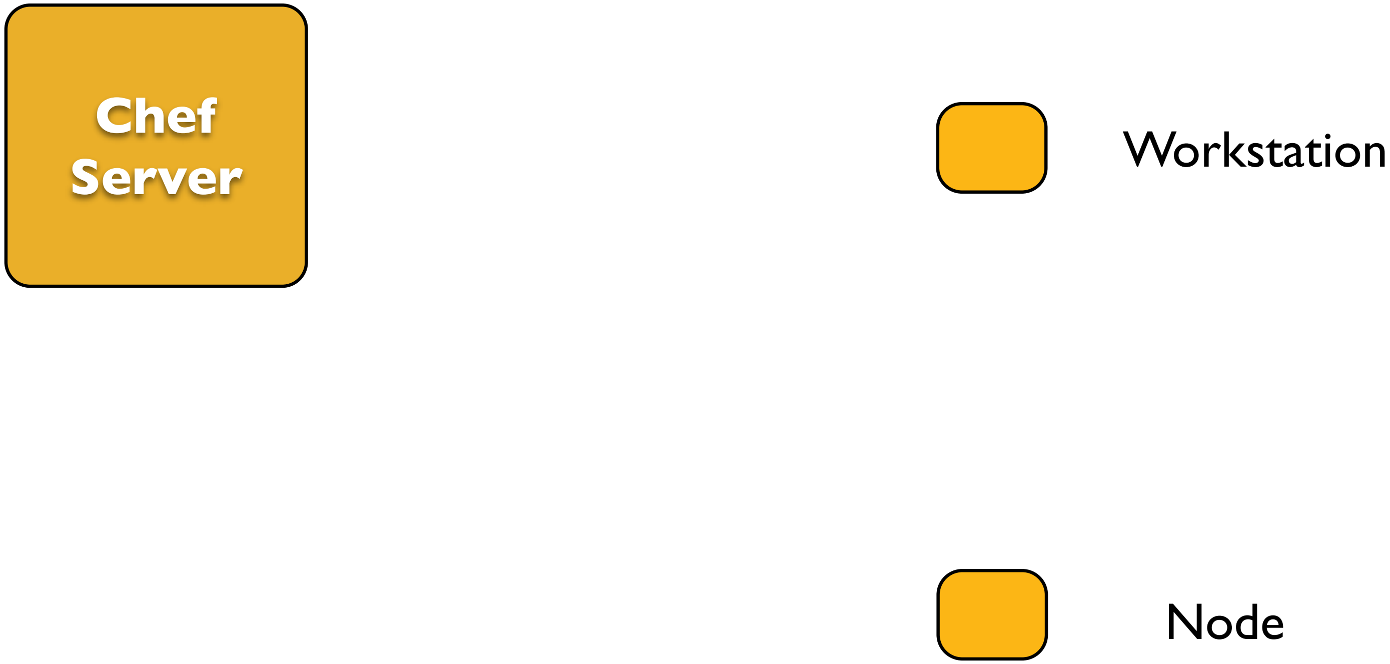


"Bootstrap" the Target Instance

```
$ knife bootstrap <EXTERNAL_ADDRESS> -x chef -P chef -N "module2"
```

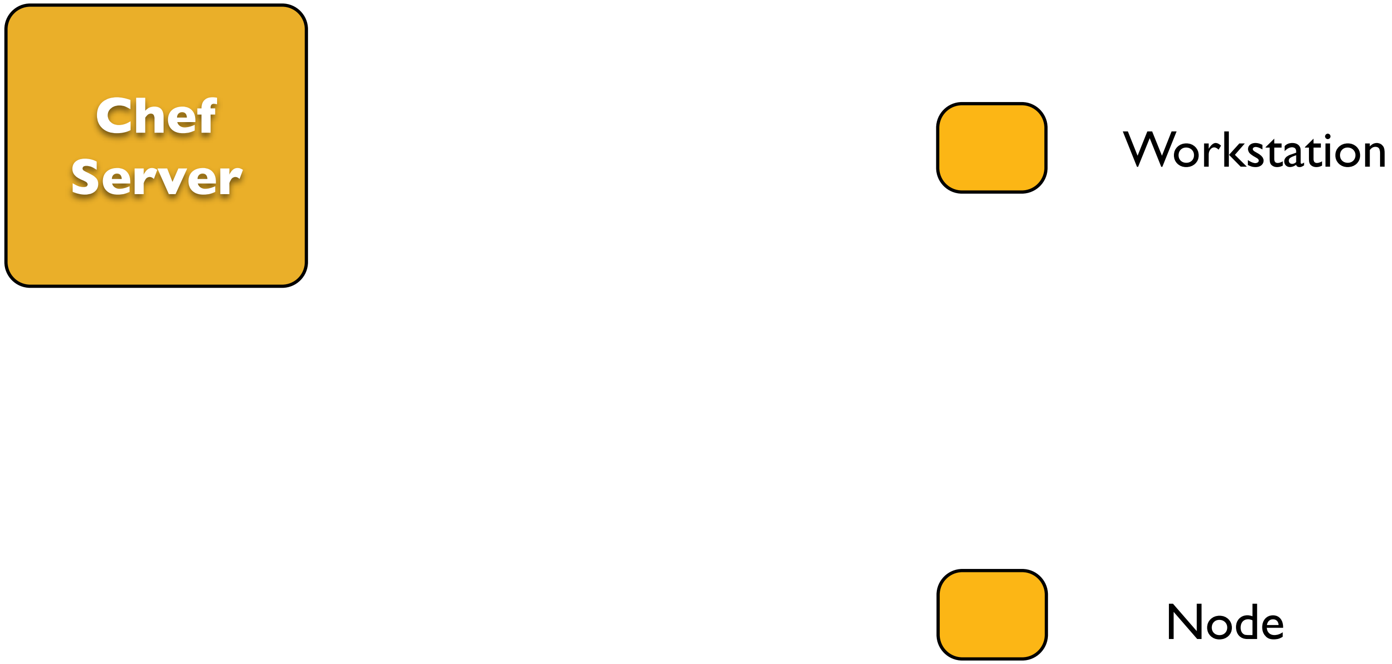
```
Bootstrapping Chef on uvo1qrwls0jdgs3blvt.vm.cld.sr
...
...
uvo1qrwls0jdgs3blvt.vm.cld.sr Creating a new client identity for
module2 using the validator key.
uvo1qrwls0jdgs3blvt.vm.cld.sr resolving cookbooks for run list: []
uvo1qrwls0jdgs3blvt.vm.cld.sr Synchronizing Cookbooks:
uvo1qrwls0jdgs3blvt.vm.cld.sr Compiling Cookbooks...
uvo1qrwls0jdgs3blvt.vm.cld.sr [2014-01-28T11:03:14-05:00] WARN: Node
node2 has an empty run list.
uvo1qrwls0jdgs3blvt.vm.cld.sr Converging 0 resources
uvo1qrwls0jdgs3blvt.vm.cld.sr Chef Client finished, 0 resources updated
```

knife bootstrap



knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



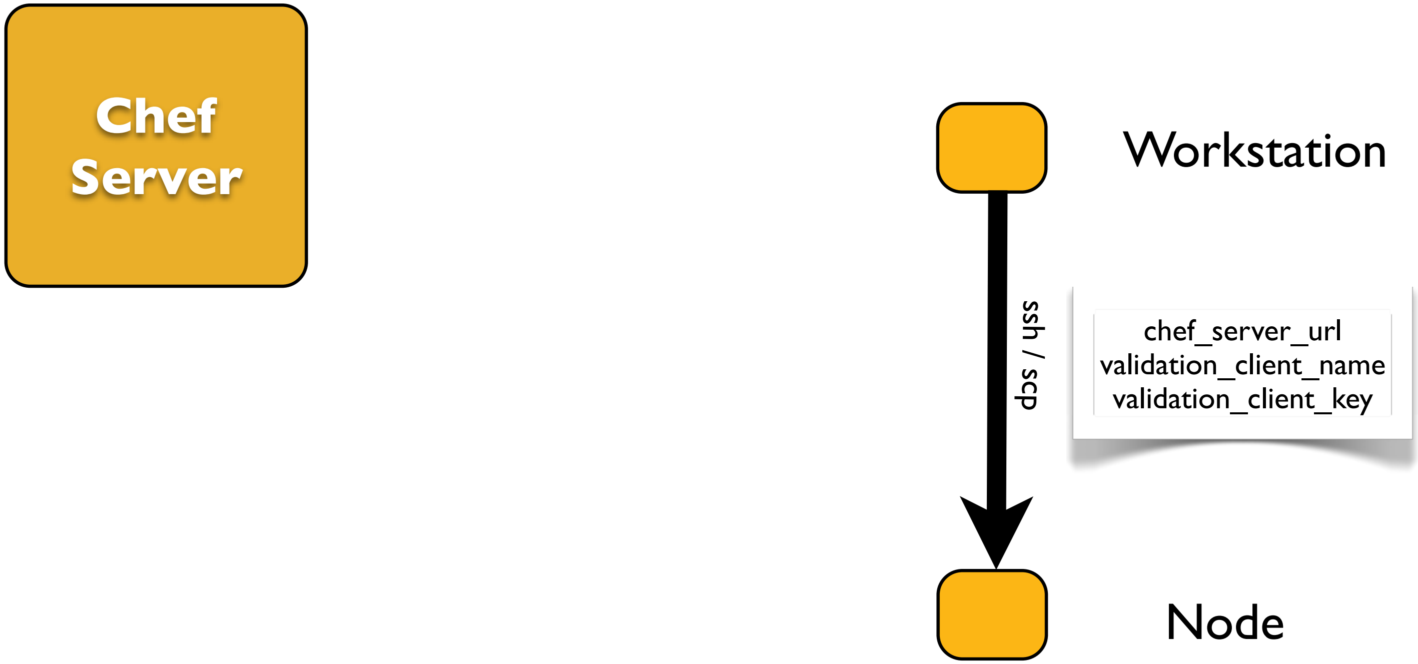
Chef Server

Workstation

Node

knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



Chef Server

Workstation

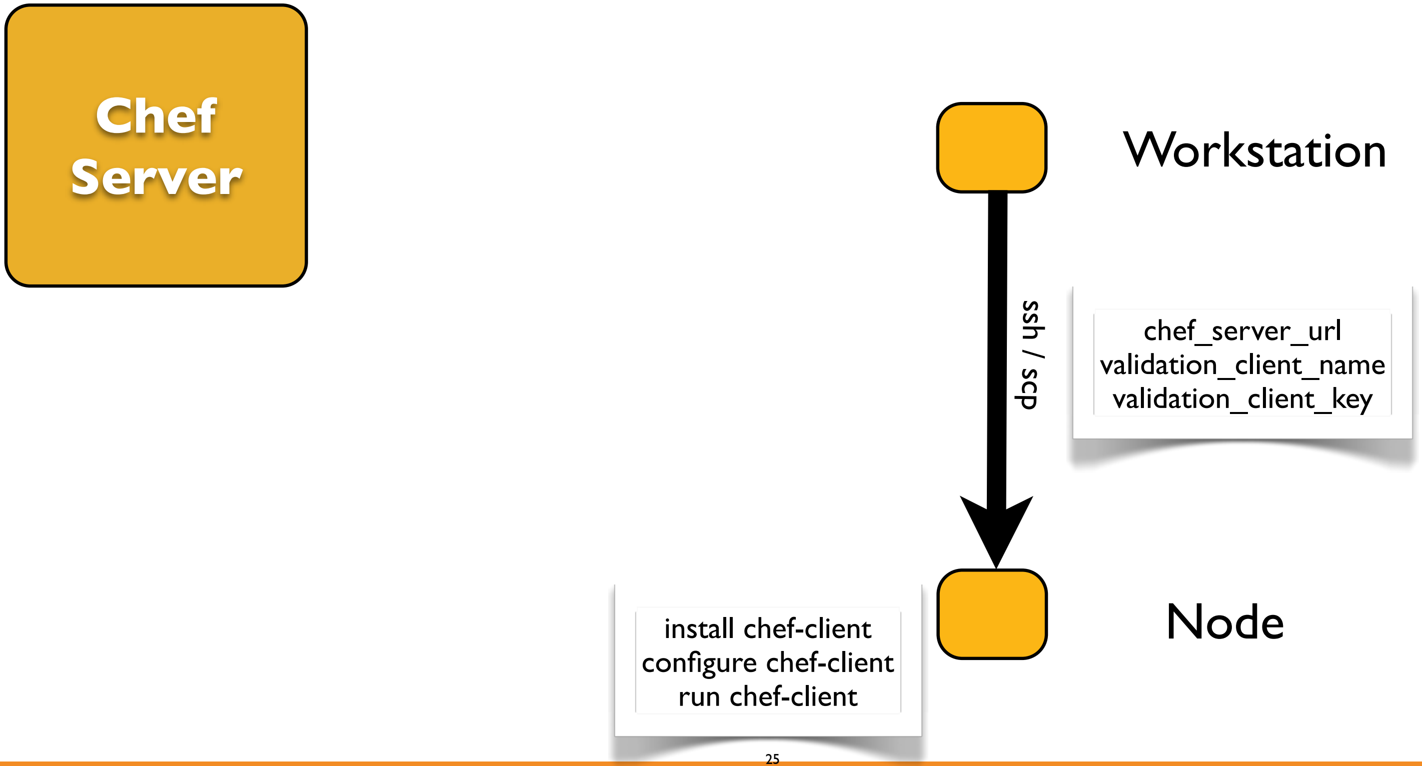
ssh / scp

chef_server_url
validation_client_name
validation_client_key

Node

knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



Chef Server

Workstation

ssh / scp

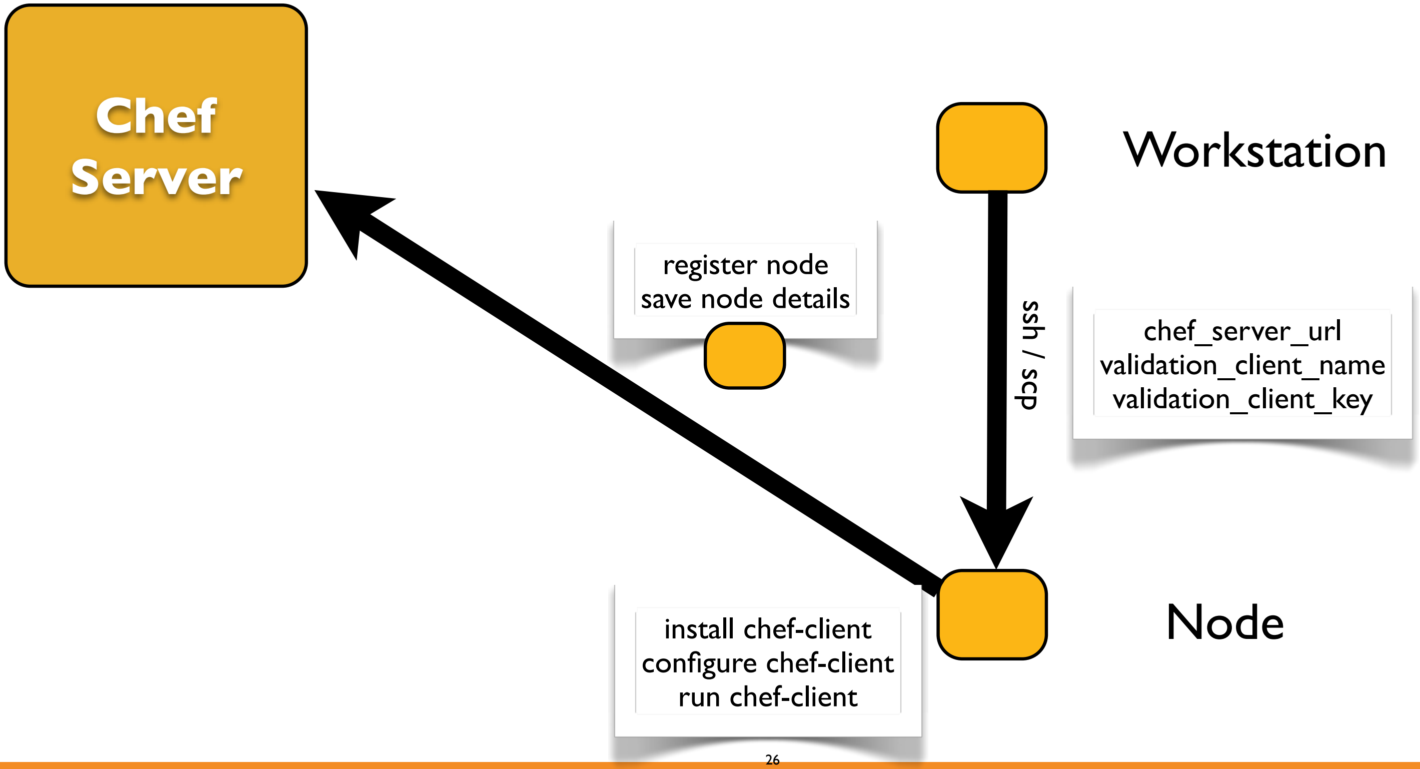
chef_server_url
validation_client_name
validation_client_key

Node

install chef-client
configure chef-client
run chef-client

knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



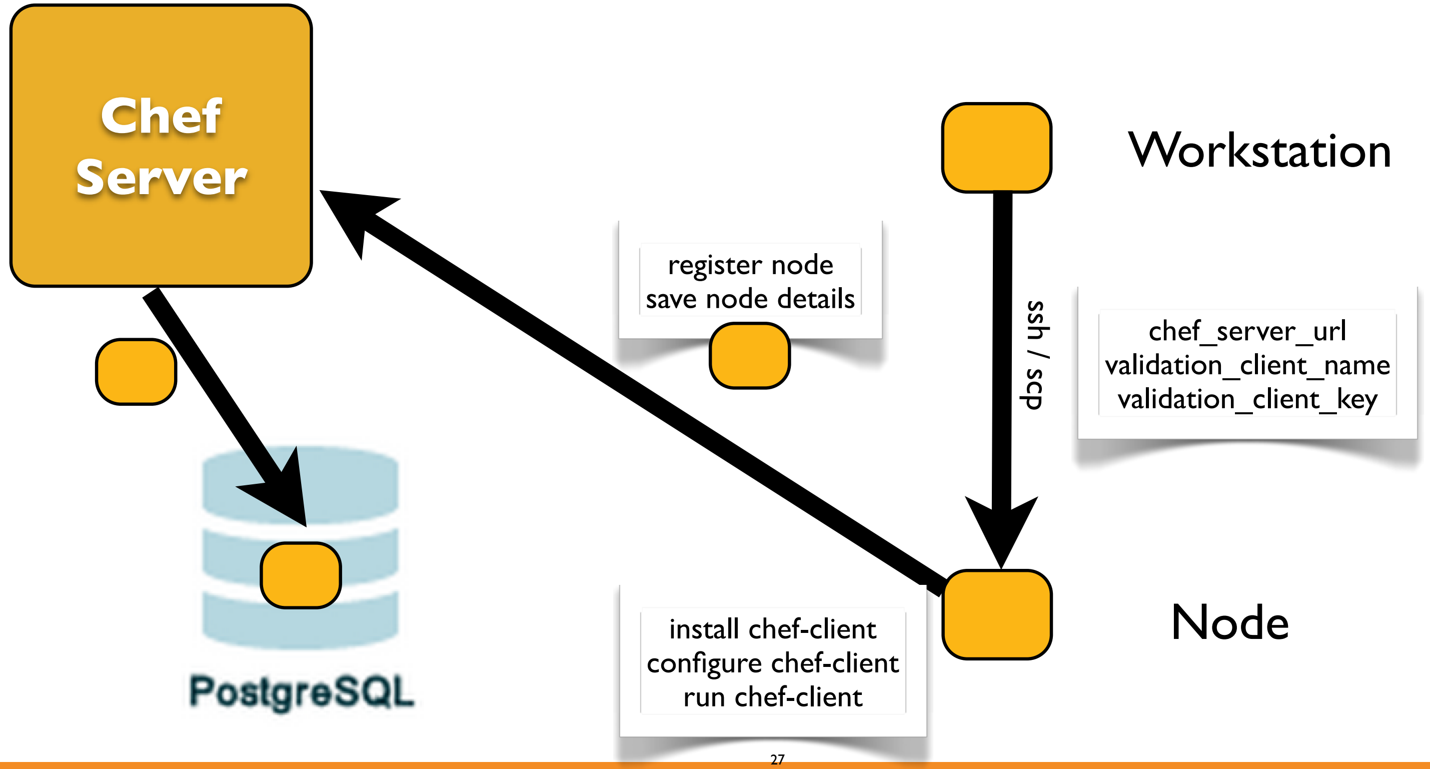
register node
save node details

chef_server_url
validation_client_name
validation_client_key

install chef-client
configure chef-client
run chef-client

knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



Chef Server

PostgreSQL

register node
save node details

Workstation

ssh / scp

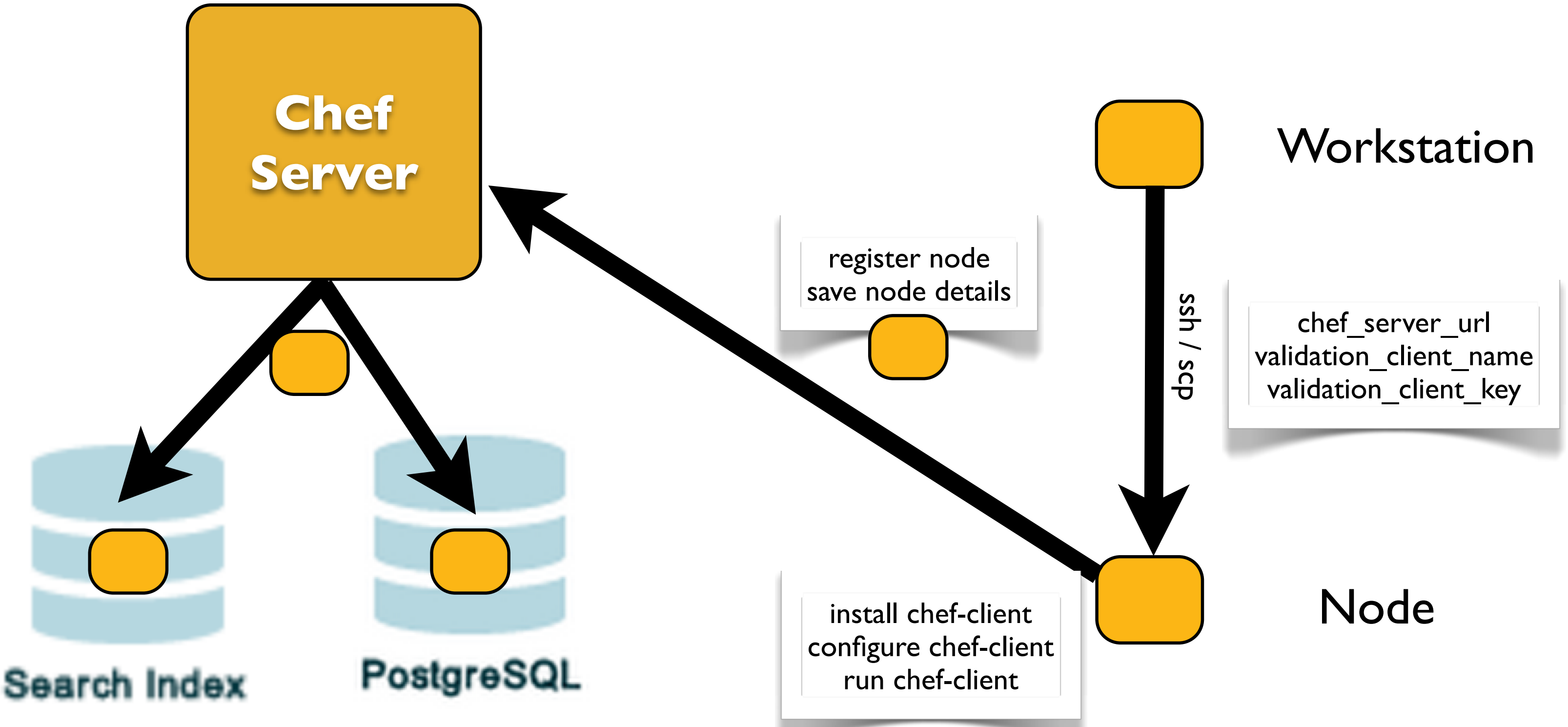
chef_server_url
validation_client_name
validation_client_key

Node

install chef-client
configure chef-client
run chef-client

knife bootstrap

```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



register node
save node details

chef_server_url
validation_client_name
validation_client_key

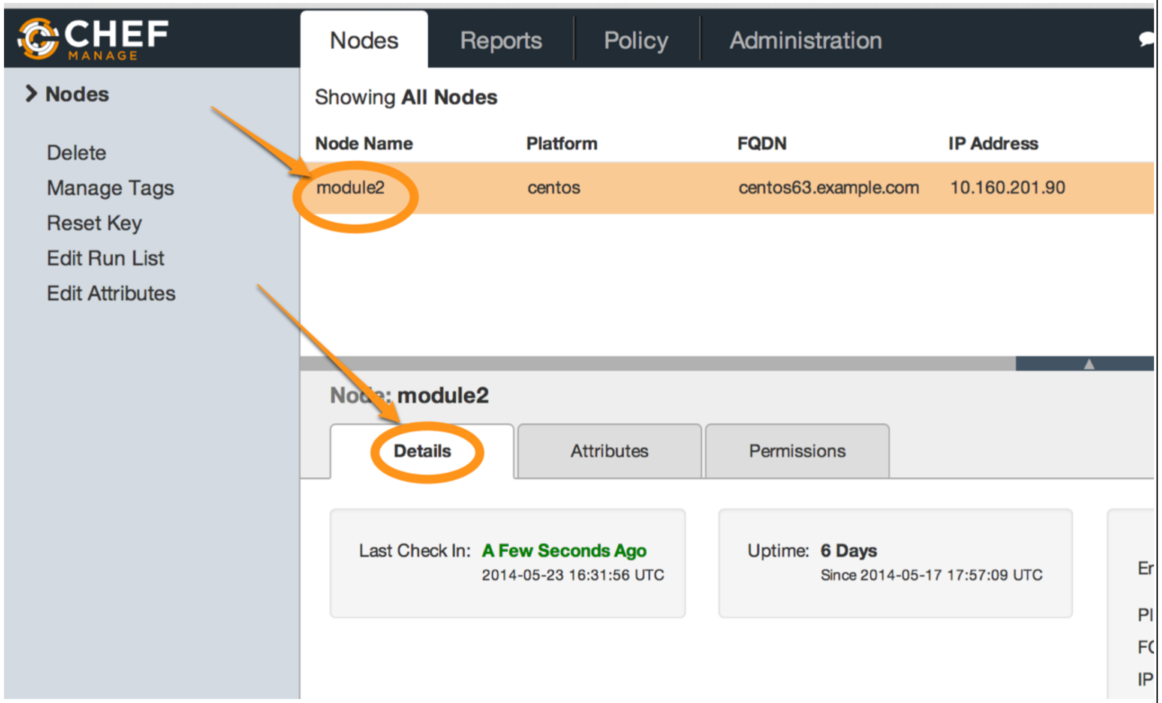
install chef-client
configure chef-client
run chef-client

Verify Your Target Instance’s Chef-Client is Configured Properly

```
$ ssh root@<EXTERNAL_ADDRESS>

root@CentOS63:~$ ls /etc/chef
client.pem  client.rb  first-boot.json validation.pem

root@CentOS63:~$ which chef-client
/usr/bin/chef-client
```

View Node on Chef Server

- Click the 'Details' tab



Showing **All Nodes**

Node Name	Platform	FQDN	IP Address
module2	centos	centos63.example.com	10.160.201.90

Node: module2

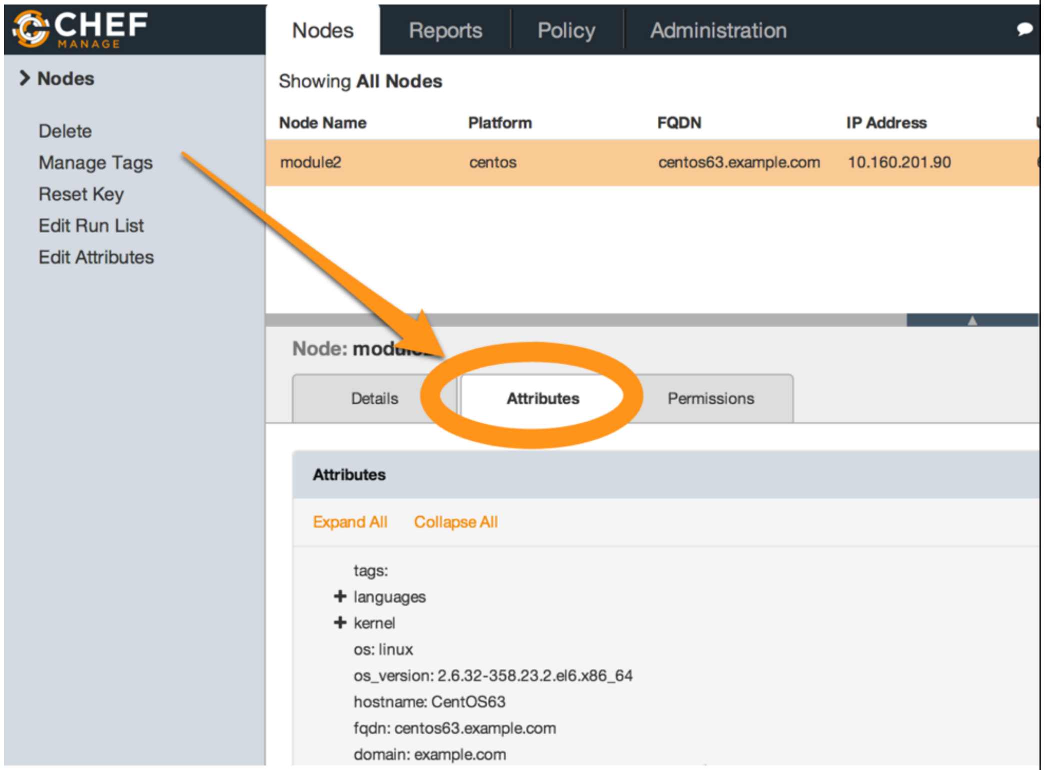
Last Check In: **A Few Seconds Ago** 2014-05-23 16:31:56 UTC

Uptime: **6 Days** Since 2014-05-17 17:57:09 UTC

Er
Pl
F(
IP

View Node on Chef Server

- Click the 'Attributes' tab



Showing **All Nodes**

Node Name	Platform	FQDN	IP Address
module2	centos	centos63.example.com	10.160.201.90

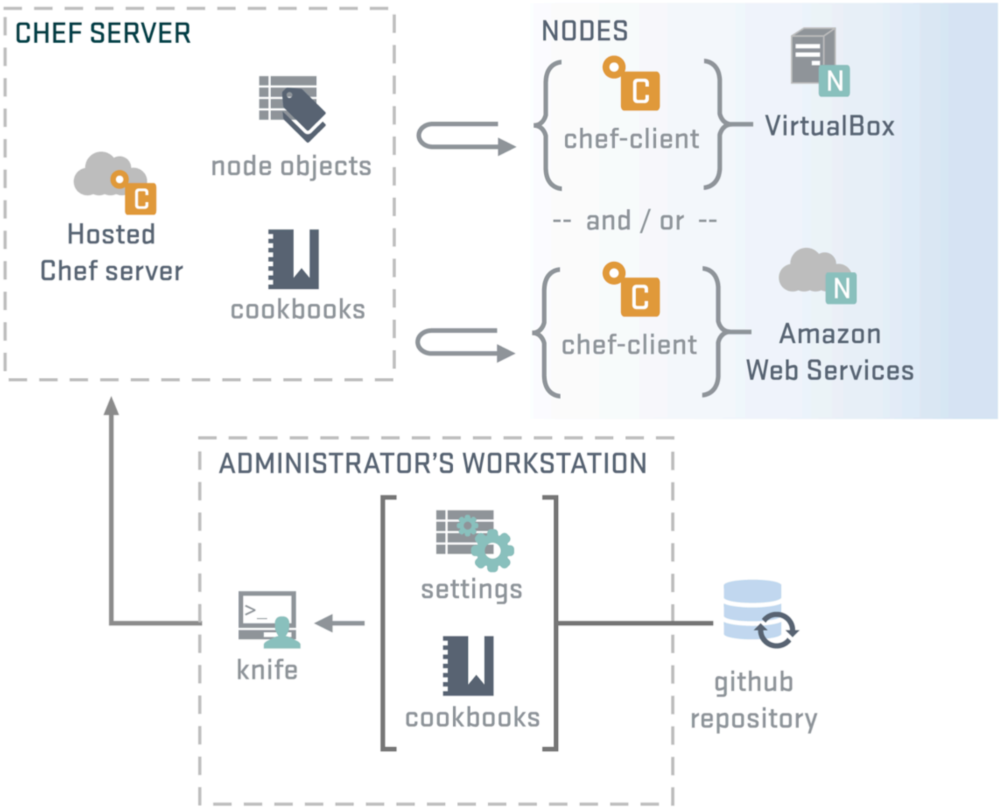
Attributes

```
  tags:
+ languages
+ kernel
  os: linux
  os_version: 2.6.32-358.23.2.el6.x86_64
  hostname: CentOS63
  fqdn: centos63.example.com
  domain: example.com
```

Node

- The node is registered with Chef Server
- The Chef Server displays information about the node
- This information comes from Ohai - we'll see Ohai later.....

Checkpoint


CHEF SERVER
Hosted Chef server
node objects
cookbooks
NODES
chef-client
VirtualBox
-- and / or --
chef-client
Amazon Web Services
ADMINISTRATOR'S WORKSTATION
knife
settings
cookbooks
github repository

Chef Resources and Recipes

Writing an Apache cookbook

Lesson Objectives

- After completing the lesson, you will be able to
  - Describe in detail what a **cookbook** is
  - Create a new cookbook
  - Explain what a **recipe** is
  - Describe how to use the **package**, **service**, and **template** resources
  - **Upload a cookbook** to the Chef Server
  - Explain what a **run list** is, and how to set it for a node via knife
  - Explain the output of a chef-client run

What is a cookbook?

- A cookbook is like a “package” for Chef recipes.
  - It contains all the recipes, files, templates, libraries, etc. required to configure a portion of your infrastructure
- Typically they map 1:1 to a piece of software or functionality.

The Problem and the Success Criteria

- **The Problem:** We need a web server configured to serve up our home page.
- **Success Criteria:** We can see the homepage in a web browser.

Desired state: our policy

- Apache web server should be installed
- Apache should be running and configured to start when the machine boots
- Our home page should be displayed

- Please note in this course we're teaching Chef primitives, not web server management
- This is probably not the Apache HTTP server configuration you would use in production

Exercise: Create a new Cookbook

```
$ knife cookbook create apache
```

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

Exercise: Explore the cookbook

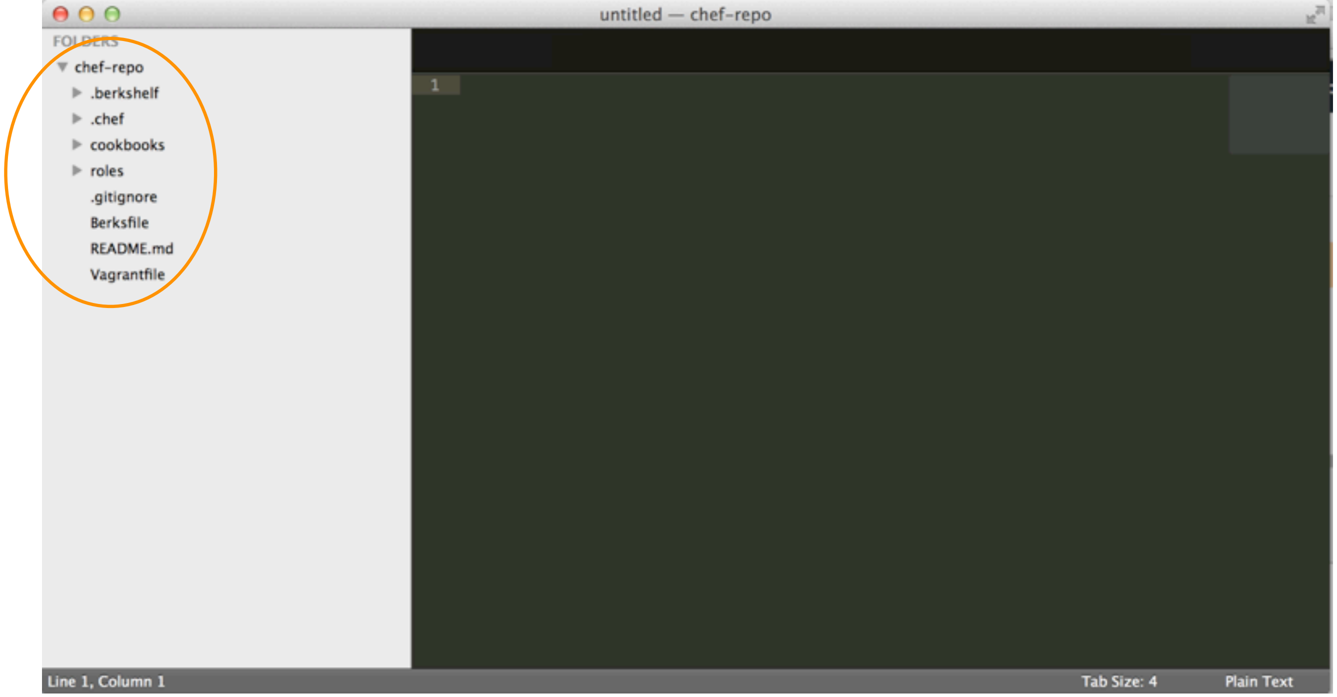
```
$ ls -la cookbooks/apache
```

```
total 24
drwxr-xr-x  13 opscode  opscode   442 Jan 24 21:25 .
drwxr-xr-x   5 opscode  opscode   170 Jan 24 21:25 ..
-rw-r--r--   1 opscode  opscode   412 Jan 24 21:25 CHANGELOG.md
-rw-r--r--   1 opscode  opscode  1447 Jan 24 21:25 README.md
drwxr-xr-x   2 opscode  opscode    68 Jan 24 21:25 attributes
drwxr-xr-x   2 opscode  opscode    68 Jan 24 21:25 definitions
drwxr-xr-x   3 opscode  opscode   102 Jan 24 21:25 files
drwxr-xr-x   2 opscode  opscode    68 Jan 24 21:25 libraries
-rw-r--r--   1 opscode  opscode   276 Jan 24 21:25 metadata.rb
drwxr-xr-x   2 opscode  opscode    68 Jan 24 21:25 providers
drwxr-xr-x   3 opscode  opscode   102 Jan 24 21:25 recipes
drwxr-xr-x   2 opscode  opscode    68 Jan 24 21:25 resources
drwxr-xr-x   3 opscode  opscode   102 Jan 24 21:25 templates
```

Exercise: Open a project drawer if you're using Sublime Text

- If you're using Sublime, then File>Open the chef-repo directory you created earlier

Access the cookbook files from the left menu



Exercise: Edit the default recipe

OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#
```

Exercise: Add a package resource to install Apache to the default recipe

OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#

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

SAVE FILE!

So the resource we just wrote...

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

So the resource we just wrote...

- Is a **package** resource

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

So the resource we just wrote...

- Is a package resource
- Whose **name** is *httpd*

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

So the resource we just wrote...

- Is a package resource
- Whose name is *httpd*
- With an install **action**

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

Notice we didn’t say how to install the package

- Resources are **declarative** - that means we say *what* we want to have happen, rather than *how*

- Resources take action through **Providers** - providers perform the how
- Chef uses the **platform** the node is running to determine the correct **provider** for a resource

Package Resource

```
package "git"
```

```
yum install git

apt-get install git

pacman sync git

pkg_add -r git
```

Providers are determined by node's platform

Exercise: Add a service resource to ensure the service is started and enabled at boot

OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
...
# All rights reserved - Do Not Redistribute
#

package "httpd" do
  action :install
end

service "httpd" do
  action [ :enable, :start ]
end
```

SAVE FILE!

So the resource we just wrote...

```
service "httpd" do
  action [ :enable, :start ]
end
```

So the resource we just wrote...

- Is a **service** resource

```
service "httpd" do
  action [ :enable, :start ]
end
```

So the resource we just wrote...

- Is a service resource
- Whose **name** is *httpd*

```
service "httpd" do
  action [ :enable, :start ]
end
```

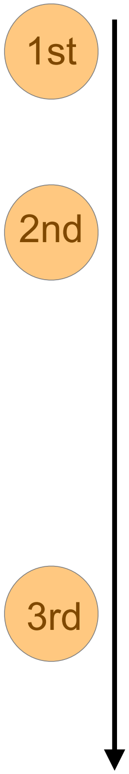
So the resource we just wrote...

- Is a service resource
- Whose **name** is *httpd*
- With two **actions**:
  - enable
  - start

```
service "httpd" do
  action [ :enable, :start ]
end
```

Order Matters

- Resources are executed in order



1st

2nd

3rd

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => :true
  action [:enable, :start]
end
```

Exercise: Add a cookbook_file resource to copy the home page in place

OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
...

service "httpd" do
  action [ :enable, :start ]
end

template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

SAVE FILE!

So the resource we just wrote...

```
template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

So the resource we just wrote...

- Is a **template** resource

```
template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

So the resource we just wrote...

- Is a template resource
- Whose **name** is:
  /var/www/html/index.html

```
template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

So the resource we just wrote...

- Is a template resource
- Whose **name** is: */var/www/html/index.html*
- With two **parameters**:
  - **source** of *index.html.erb*
  - **mode** of *“0644”*

```
template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

Full contents of the apache recipe

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#

package "httpd" do
  action :install
end

service "httpd" do
  action [ :enable, :start ]
end

template "/var/www/html/index.html" do
  source "index.html.erb"
  mode "0644"
end
```

Exercise: Add index.html to your cookbook's files/default directory

OPEN IN EDITOR: cookbooks/apache/templates/default/index.html.erb

```
<h1>Hello, world!</h1>
```

SAVE FILE!

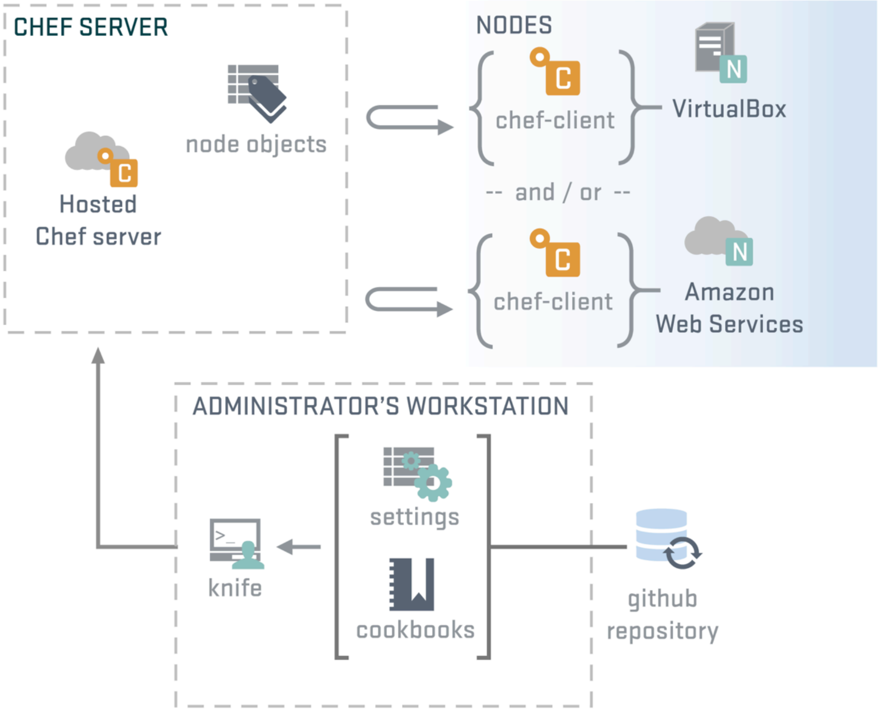
Exercise: Upload the cookbook

```
$ knife cookbook upload apache
```

```
Uploading apache                 [0.1.0]
Uploaded 1 cookbook.
```

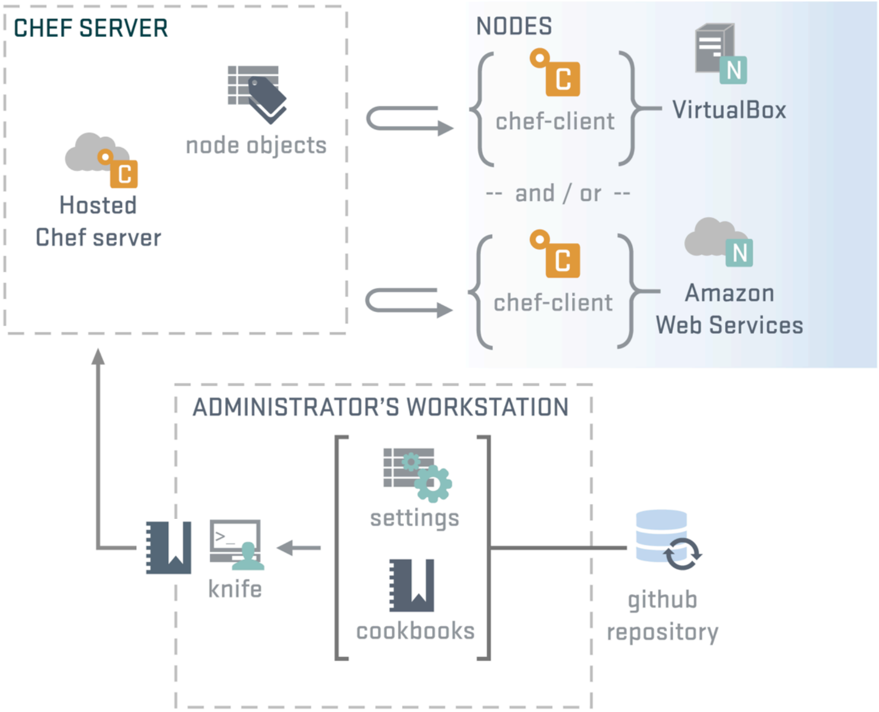
Upload a cookbook

```
knife cookbook upload apache
```



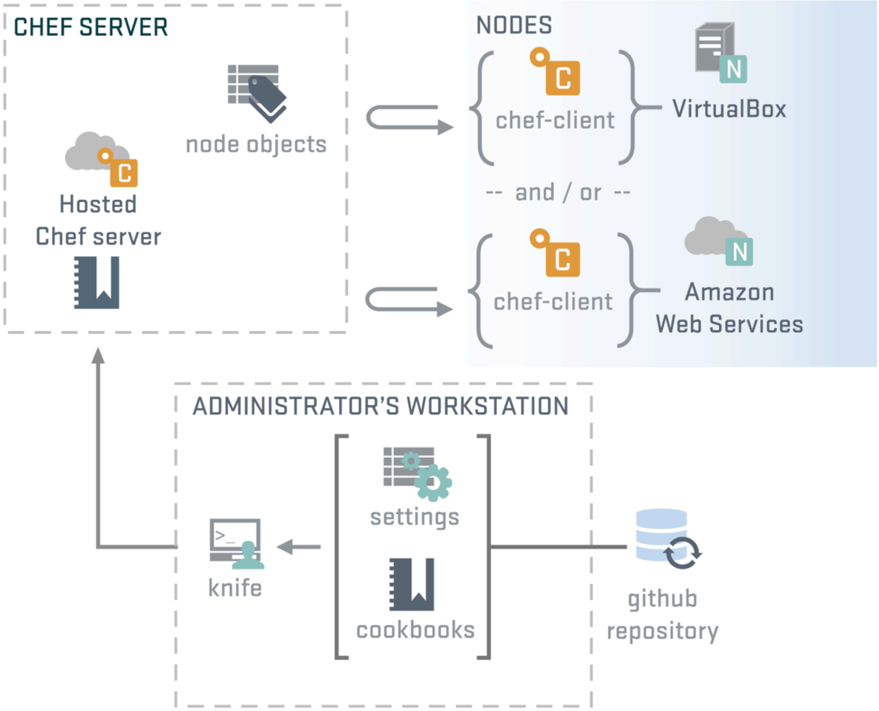
Upload a cookbook

```
knife cookbook upload apache
```



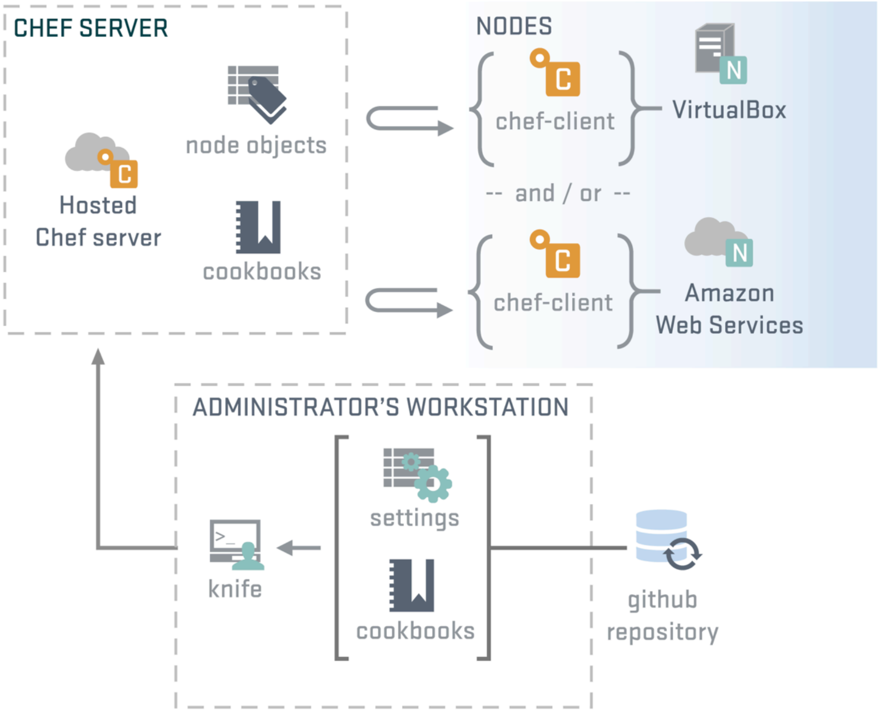
Upload a cookbook

```
knife cookbook upload apache
```



Upload a cookbook

```
knife cookbook upload apache
```



The Run List

- The Run List is the ordered set of recipes and roles that the Chef Client will execute on a node
  - Recipes are specified by “**recipe[*name*]**”

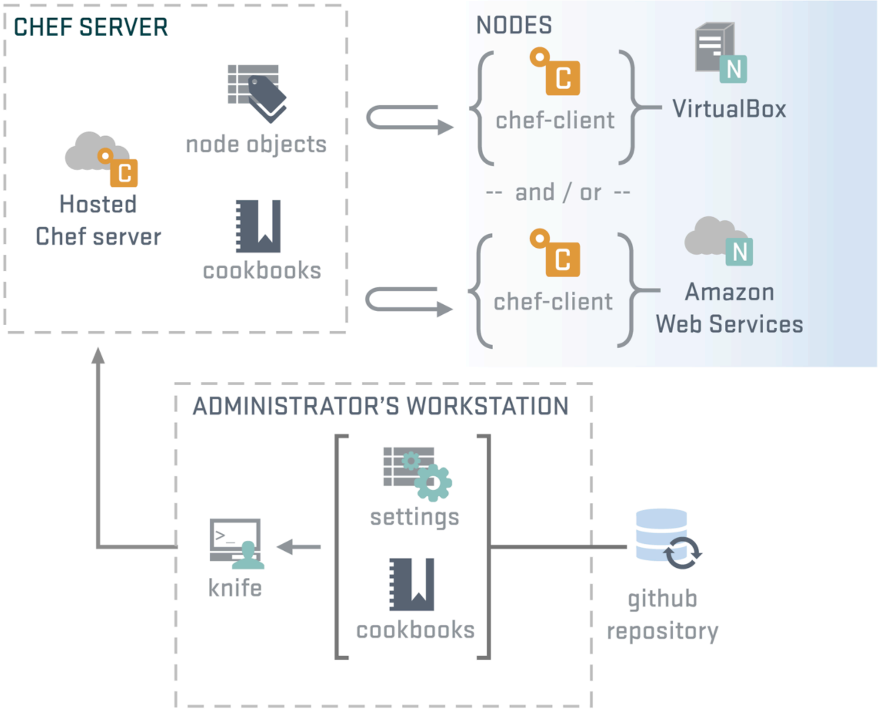
Exercise: Add apache recipe to test node's run list

```
$ knife node run_list add module2 "recipe[apache]"
```

```
module2:
  run_list: recipe[apache]
```

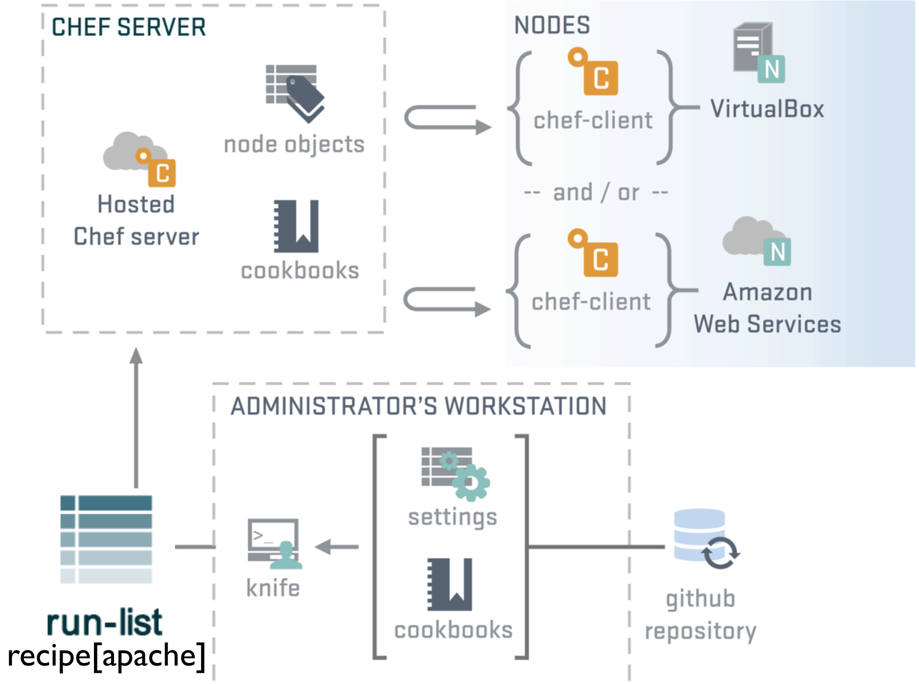
Upload a cookbook

```
knife node run_list add module2 “recipe[apache]”
```



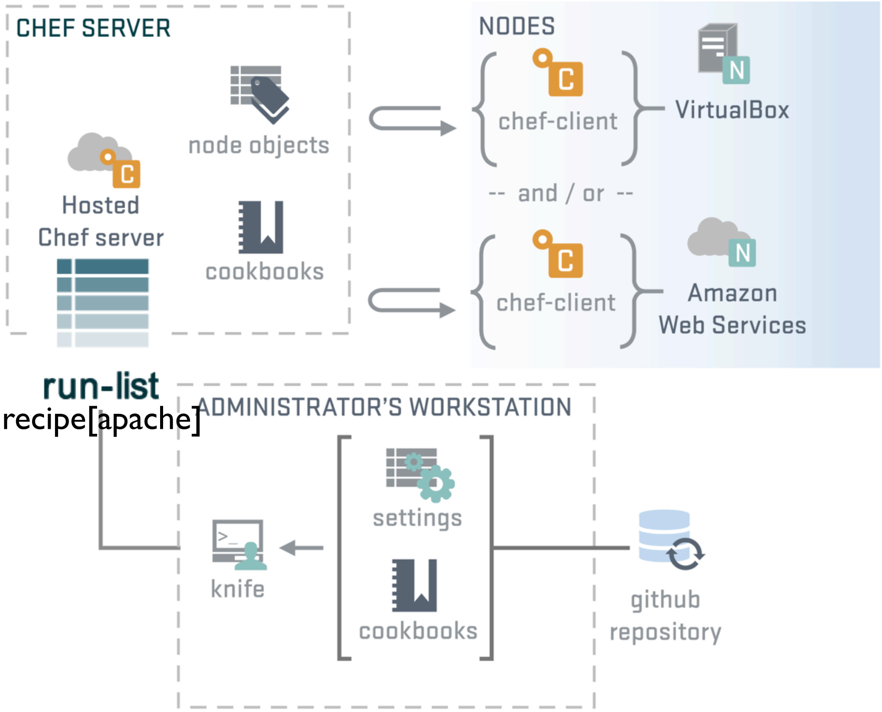
Upload a cookbook

```
knife node run_list add module2 “recipe[apache]”
```



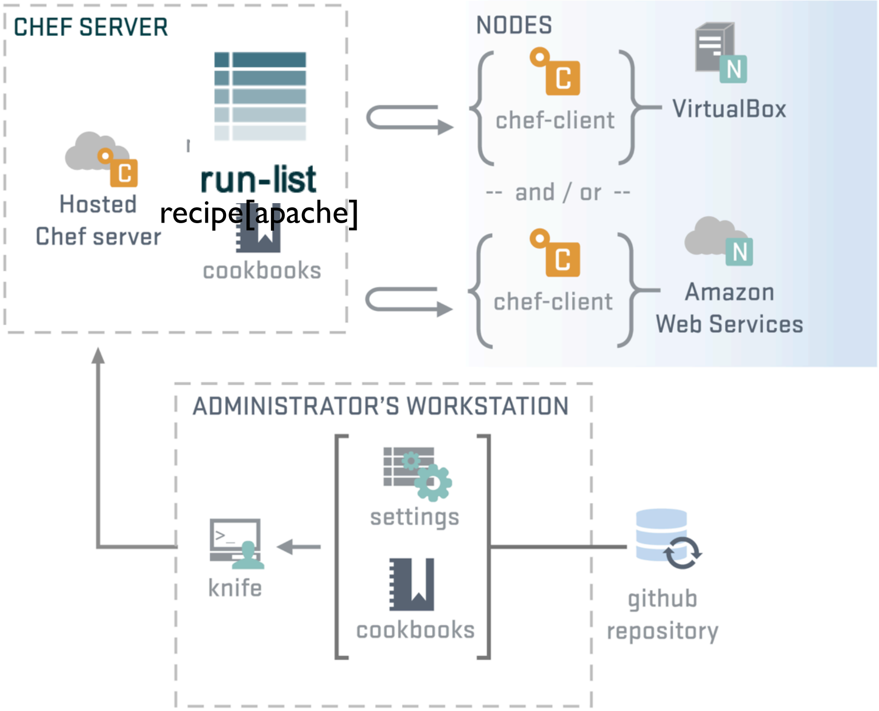
Upload a cookbook

```
knife node run_list add module2 “recipe[apache]”
```



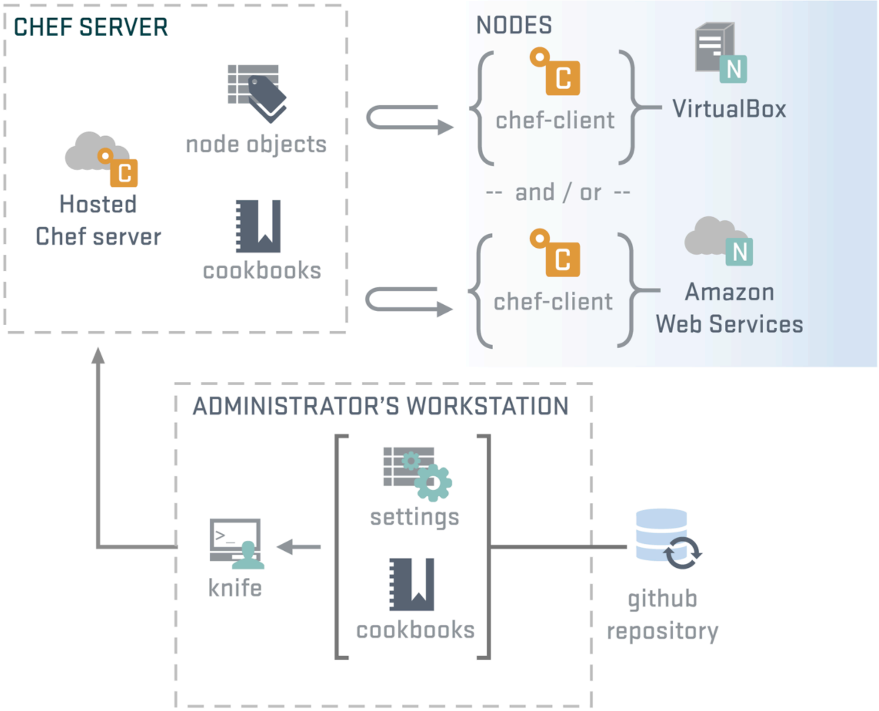
Upload a cookbook

```
knife node run_list add module2 “recipe[apache]”
```



Upload a cookbook

```
knife node run_list add module2 “recipe[apache]”
```



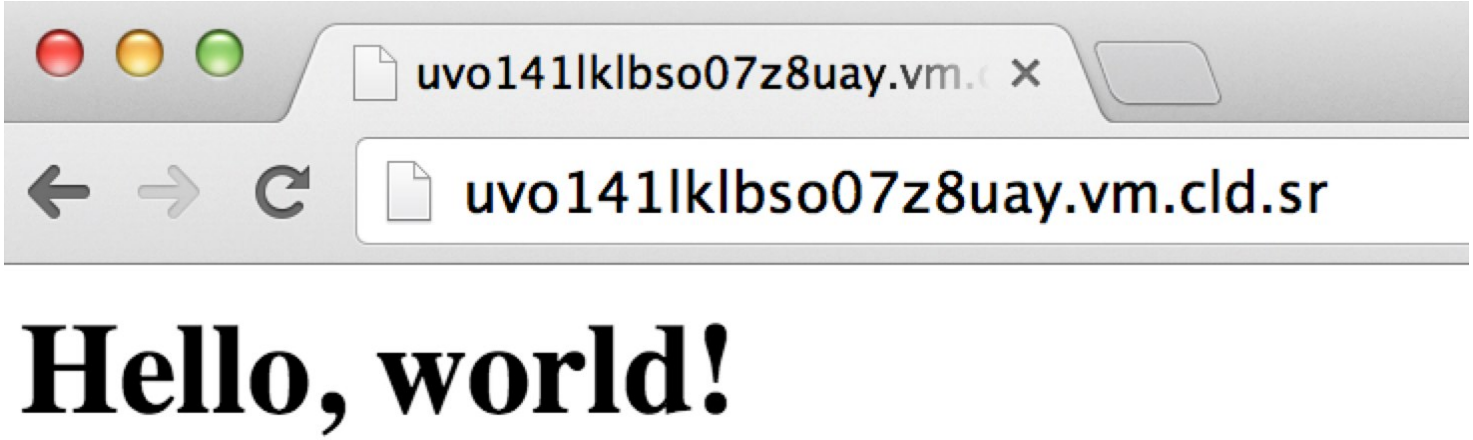
Exercise: Run Chef Client

```
root@CentOS63:~$ sudo chef-client
```

```
Starting Chef Client, version 11.10.4
resolving cookbooks for run list: ["apache"]
Synchronizing Cookbooks:
  - apache
Compiling Cookbooks...
Converging 3 resources
Recipe: apache::default
  * package[httpd] action install

    - install version 2.2.15-29.el6.centos of package httpd

  * service[httpd] action enable
    - enable service service[httpd]

  * service[httpd] action start
    - start service service[httpd]

  * template[/var/www/html/index.html] action create
    - create new file /var/www/html/index.html
    - update content in file /var/www/html/index.html from none to 17d291
```

Exercise: Verify that the home page works

- Open a web browser
- Type in the the URL for your test node

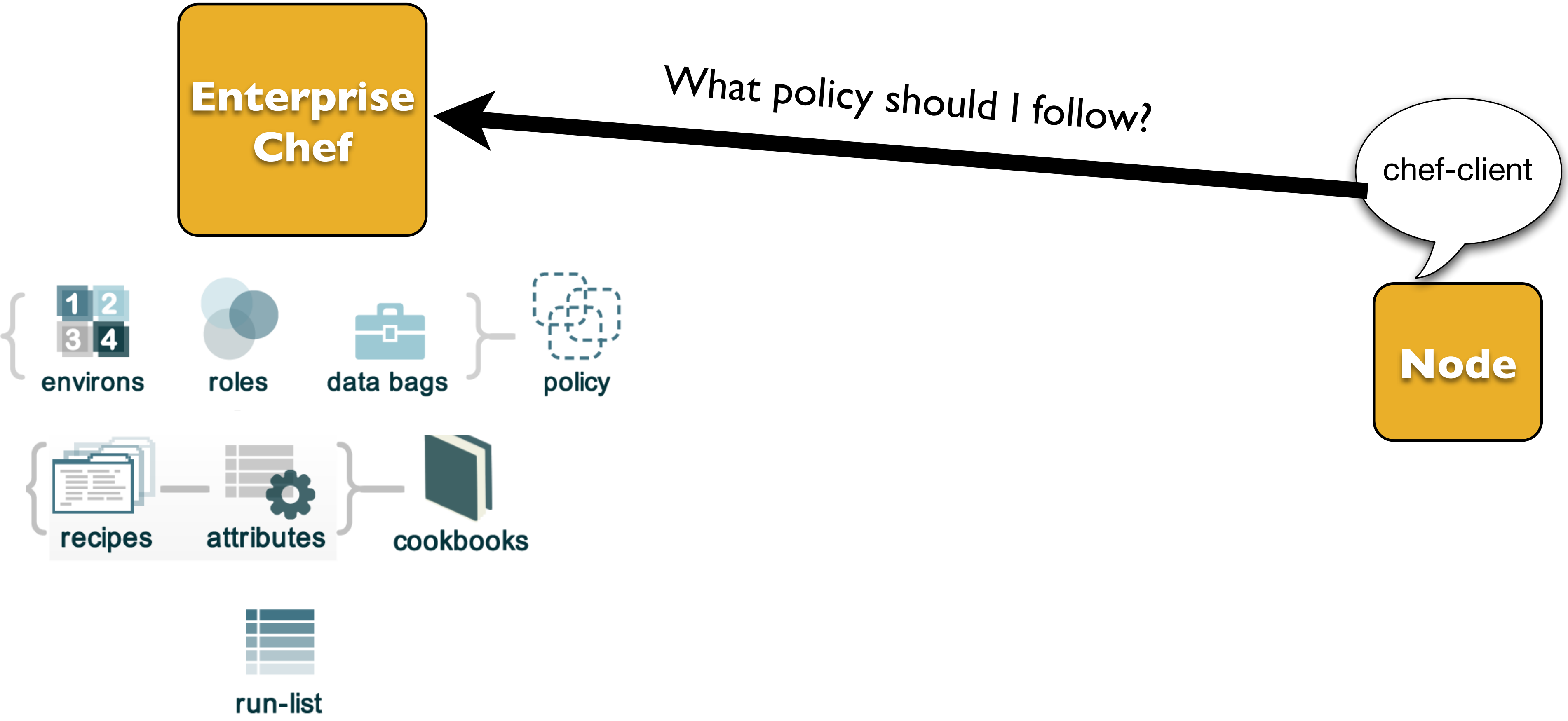


Hello, world!

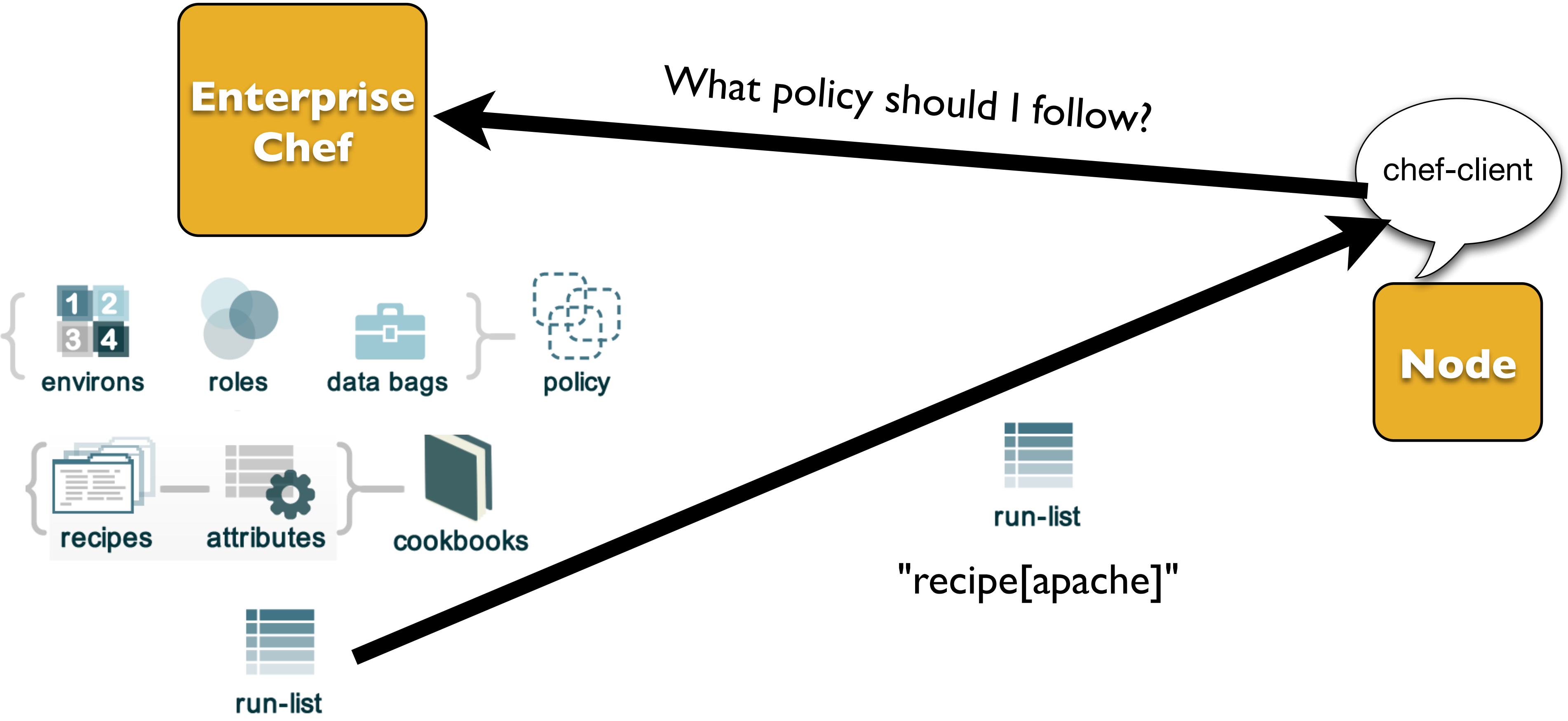
Congratulate yourself!

- You have just written your first Chef cookbook!
- (clap!)

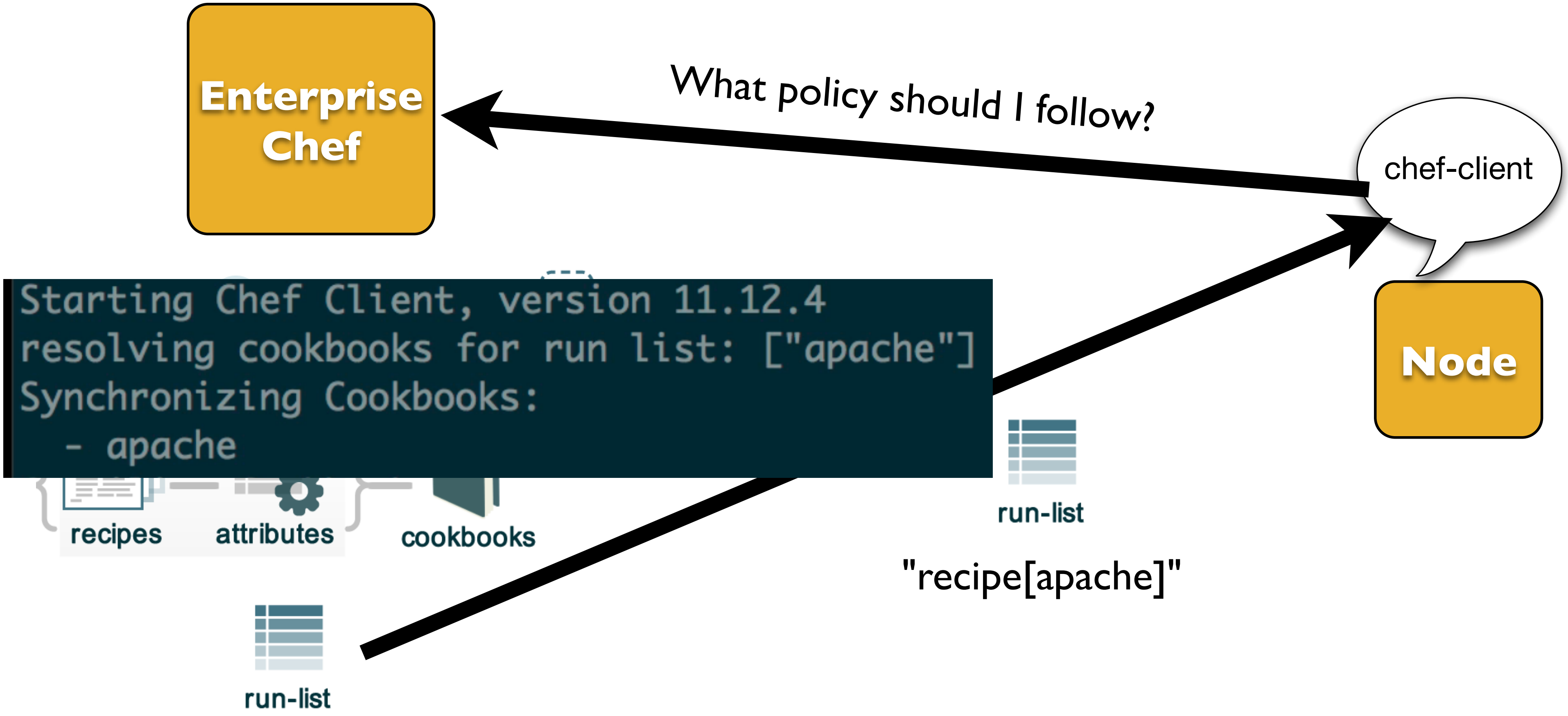
Run List


Enterprise Chef
What policy should I follow?
chef-client
Node
environs
roles
data bags
policy
recipes
attributes
cookbooks
run-list

Run List

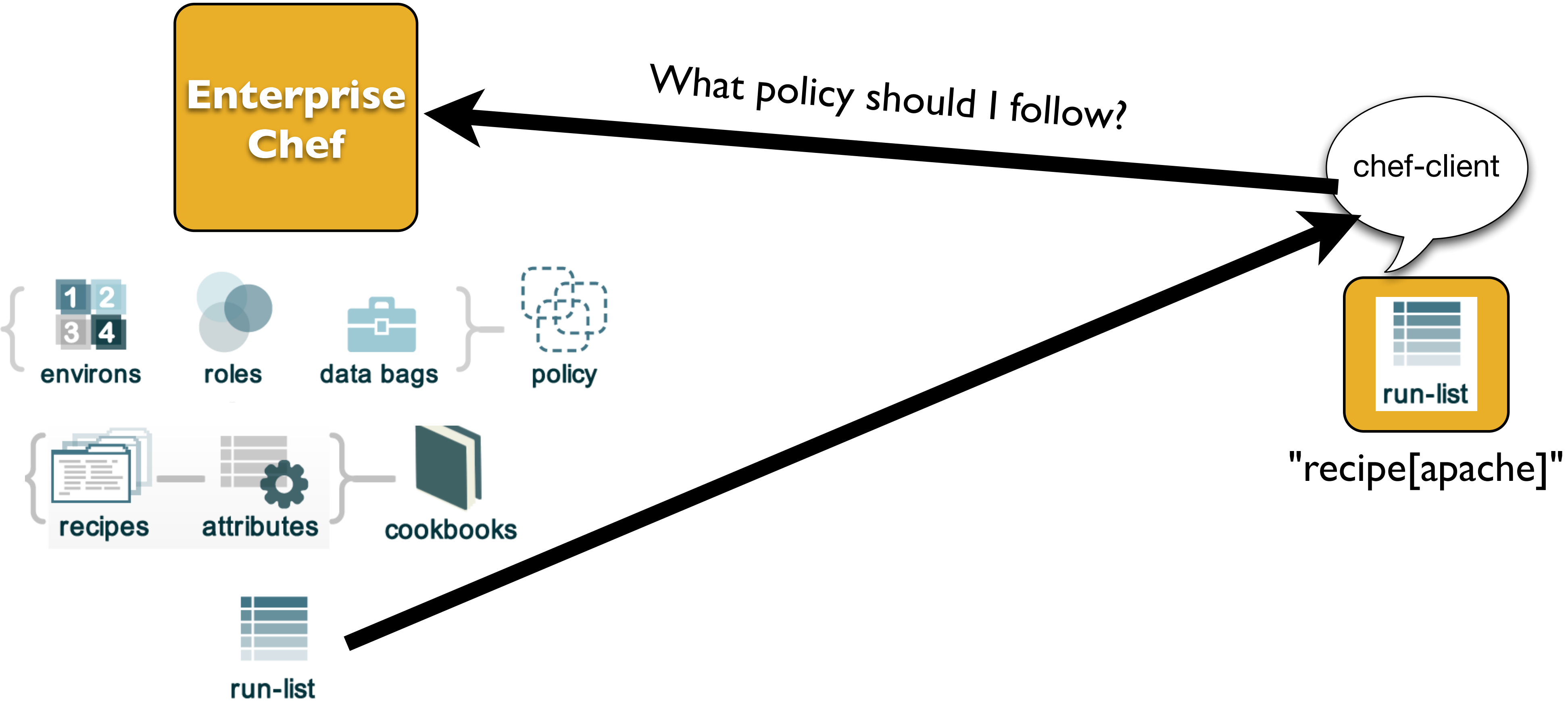

Enterprise Chef
What policy should I follow?
chef-client
Node
environs
roles
data bags
policy
recipes
attributes
cookbooks
run-list
run-list
"recipe[apache]"

Run List

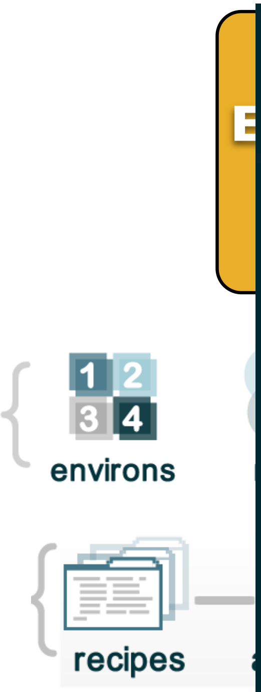


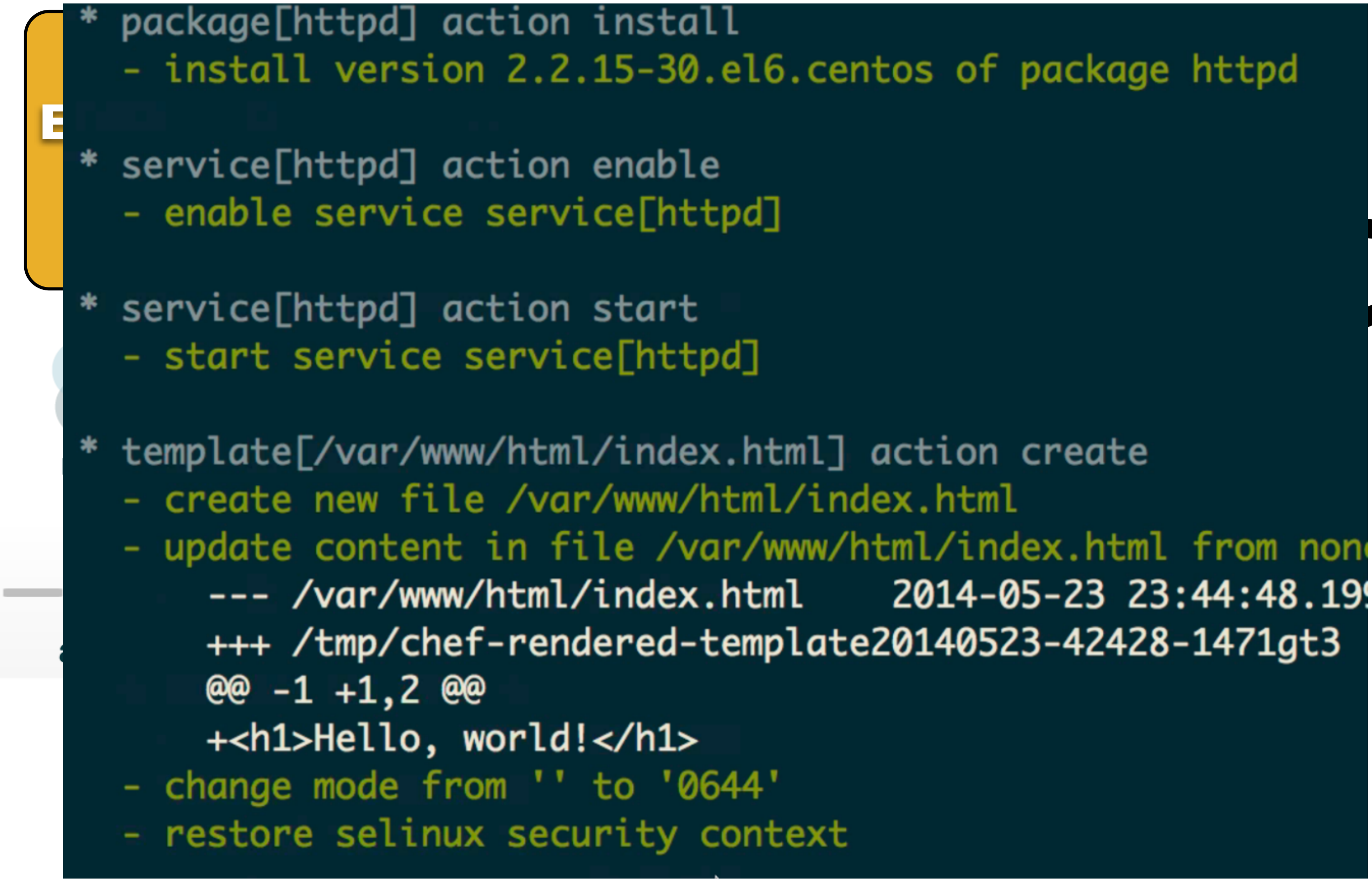
```
Starting Chef Client, version 11.12.4
resolving cookbooks for run list: ["apache"]
Synchronizing Cookbooks:
  - apache
```

Run List

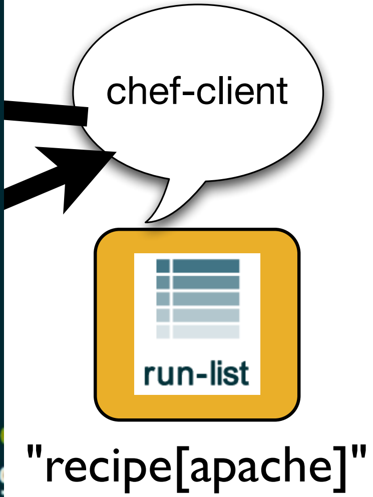

Enterprise Chef
What policy should I follow?
chef-client
environs
roles
data bags
policy
recipes
attributes
cookbooks
run-list
run-list
"recipe[apache]"

Run List





```
* package[httpd] action install
  - install version 2.2.15-30.el6.centos of package httpd

* service[httpd] action enable
  - enable service service[httpd]

* service[httpd] action start
  - start service service[httpd]

* template[/var/www/html/index.html] action create
  - create new file /var/www/html/index.html
  - update content in file /var/www/html/index.html from none
    --- /var/www/html/index.html     2014-05-23 23:44:48.19
    +++ /tmp/chef-rendered-template20140523-42428-1471gt3
    @@ -1 +1,2 @@
    +<h1>Hello, world!</h1>
  - change mode from '' to '0644'
  - restore selinux security context
```



Chef Fundamentals Webinar Series

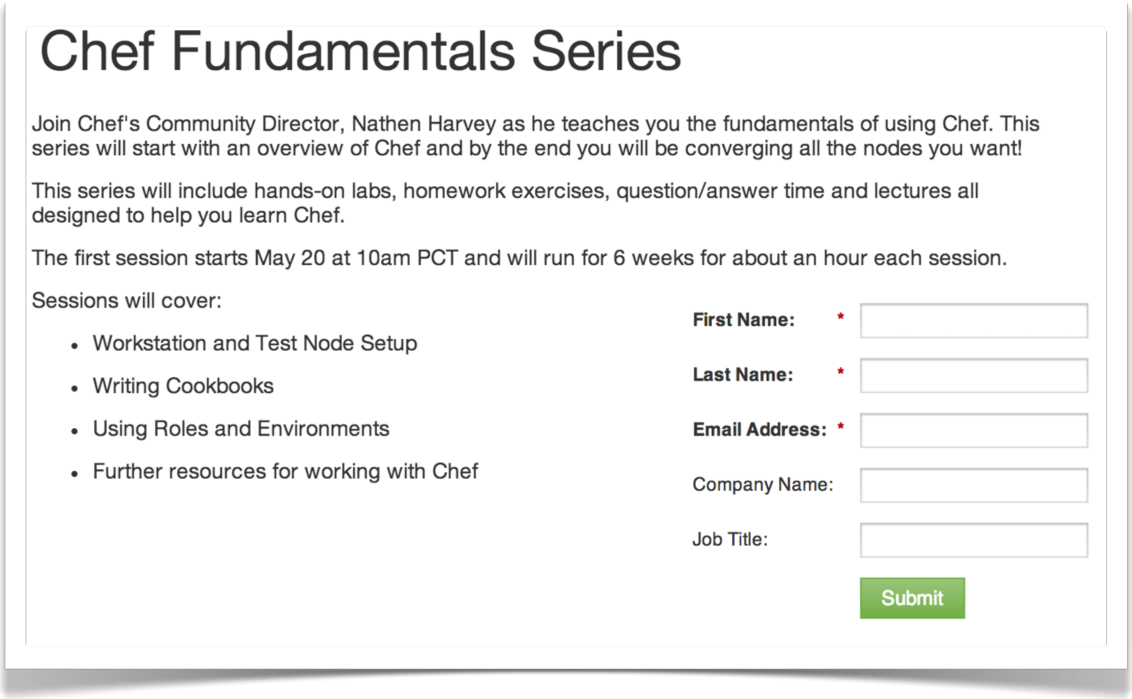
Six Week Series

- Module 1 - Overview of Chef
- Today - Node Setup, Chef Resources & Recipes
- June 3 - Working with the Node object
- June 10 - Common configuration data with Databags
- June 17 - Using Roles and Environments
- June 24 - Community Cookbooks and Further Resources

- * Topics subject to change, schedule unlikely to change

Sign-up for Webinar

- http://pages.getchef.com/cheffundamentalsseries.html



Chef Fundamentals Series

Join Chef's Community Director, Nathen Harvey as he teaches you the fundamentals of using Chef. This series will start with an overview of Chef and by the end you will be converging all the nodes you want!

This series will include hands-on labs, homework exercises, question/answer time and lectures all designed to help you learn Chef.

The first session starts May 20 at 10am PCT and will run for 6 weeks for about an hour each session.

Sessions will cover:

- Workstation and Test Node Setup
- Writing Cookbooks
- Using Roles and Environments
- Further resources for working with Chef

First Name: *

Last Name: *

Email Address: *

Company Name:

Job Title:

Additional Resources

- Chef Fundamentals Webinar Series
- https://www.youtube.com/watch?v=S5lHUpzoCYo&list=PL11cZfNdwNyPnZA9D1MbVqldGuOWqbumZ

- Discussion group for webinar participants
- https://groups.google.com/d/forum/learnchef-fundamentals-webinar

Additional Resources

- Learn Chef
- http://learnchef.com

- Documentation
- http://docs.opscode.com