



C  
h  
e  
f

# Agenda

# Topics

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

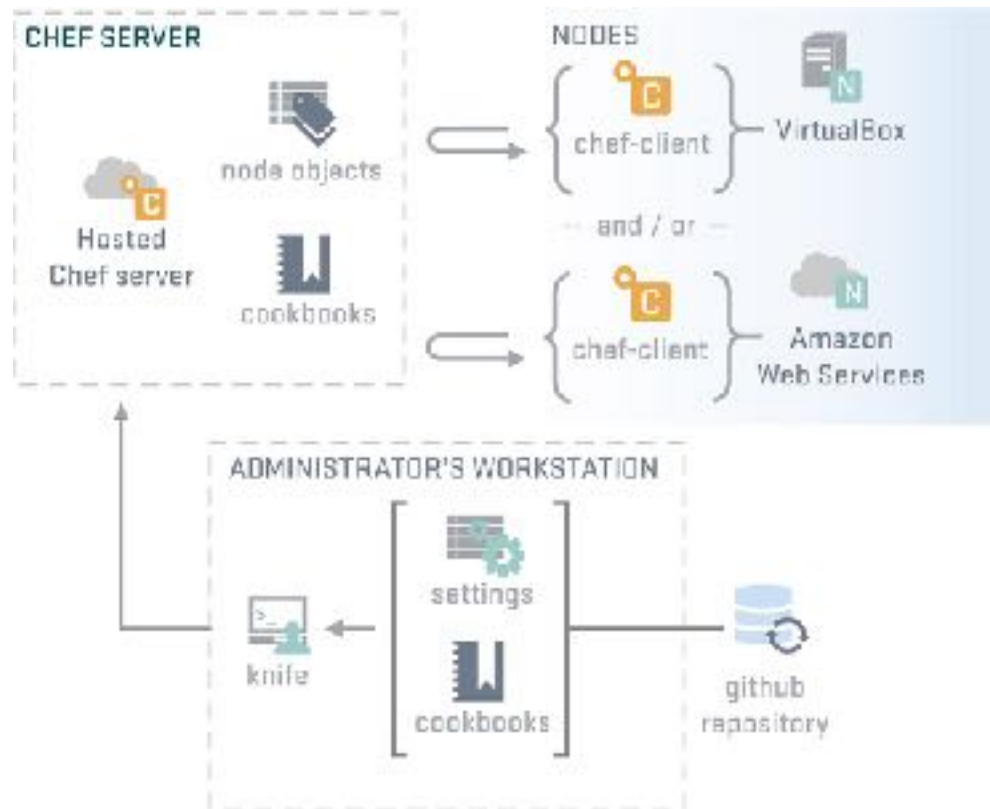
# Node Setup

Setup a Node to manage

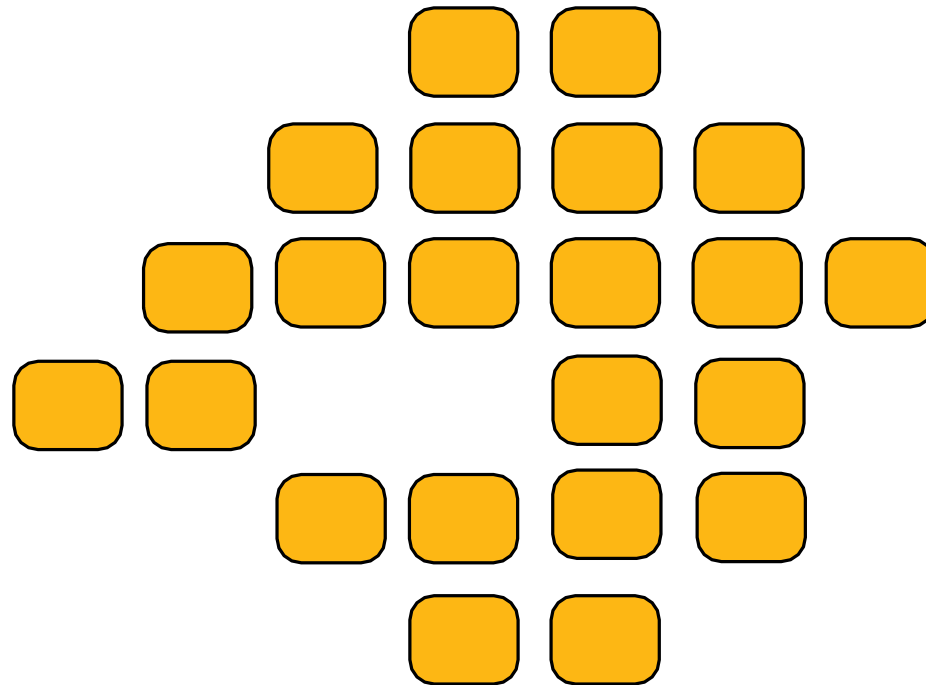
# Lesson Objectives

- 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



# Nodes

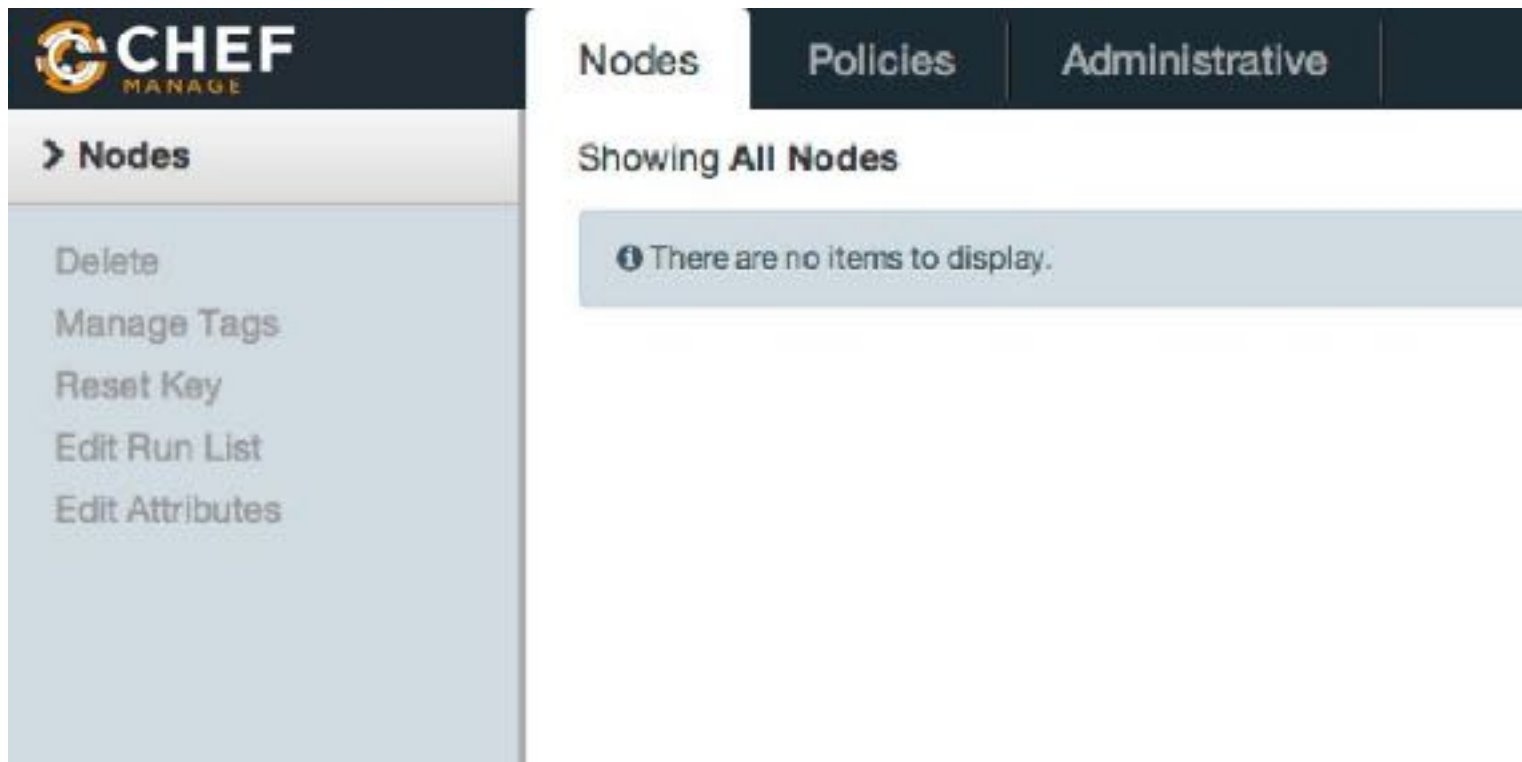


# Nodes

- 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



The screenshot displays the Chef Manage web interface. On the left is a sidebar with the 'CHEF MANAGE' logo and a 'Nodes' menu item. The main content area has tabs for 'Nodes', 'Policies', and 'Administrative', with 'Nodes' selected. Below the tabs, it says 'Showing All Nodes'. A light blue message box contains the text: 'There are no items to display.'

**CHEF MANAGE**

**> Nodes**

- Delete
- Manage Tags
- Reset Key
- Edit Run List
- Edit Attributes

**Nodes** Policies Administrative

Showing **All Nodes**

There are no items to display.

# Lab - Login

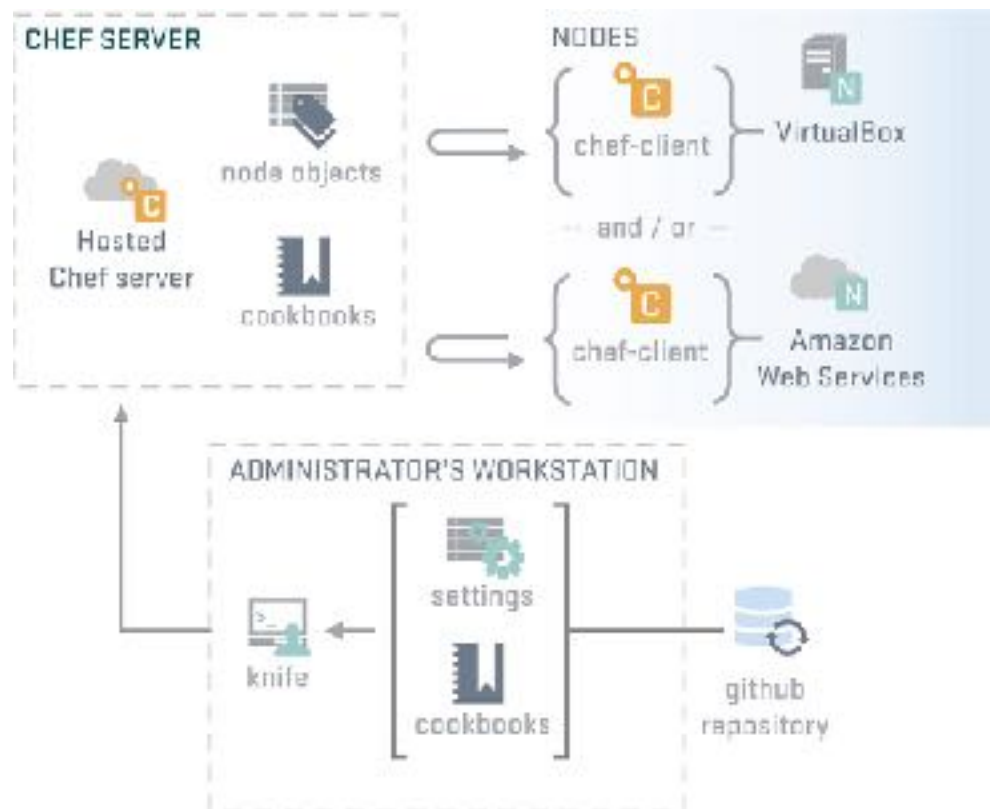
```
$ ssh root@<EXTERNAL_ADDRESS>
```

```
The authenticity of host
'uvolqrwls0jdgs3blvt.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
'uvolqrwls0jdgs3blvt.vm.cld.sr, 69.195.232.110' (RSA) to
the list of known hosts.
chef@uvolqrwls0jdgs3blvt.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 uv01qrwls0jdgs3blvt.vm.cld.sr
...
...
uv01qrwls0jdgs3blvt.vm.cld.sr Creating a new client identity for
module2 using the validator key.
uv01qrwls0jdgs3blvt.vm.cld.sr resolving cookbooks for run list: []
uv01qrwls0jdgs3blvt.vm.cld.sr Synchronizing Cookbooks:
uv01qrwls0jdgs3blvt.vm.cld.sr Compiling Cookbooks...
uv01qrwls0jdgs3blvt.vm.cld.sr [2014-01-28T11:03:14-05:00] WARN: Node
node2 has an empty run list.
uv01qrwls0jdgs3blvt.vm.cld.sr Converging 0 resources
uv01qrwls0jdgs3blvt.vm.cld.sr Chef Client finished, 0 resources updated
```

# knife bootstrap



Workstation



Nod  
e

# knife bootstrap

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



The diagram illustrates the components of a Chef environment. On the left is a large yellow rounded square labeled 'Chef Server'. To its right are two smaller yellow rounded squares. The top one is labeled 'Workstation' and the bottom one is labeled 'Node'. There are no connecting lines between these components.

**Chef  
Server**

Workstation

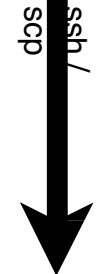
Node

# knife bootstrap

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



Workstation



Node

chef\_server\_url  
validation\_client\_name  
validation\_client\_key



# knife bootstrap

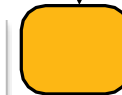
```
knife bootstrap HOSTNAME -x root -P PASSWORD -N module2
```



Workstation

scp  
ssh /

```
chef_server_url  
validation_client_name  
validation_client_key
```

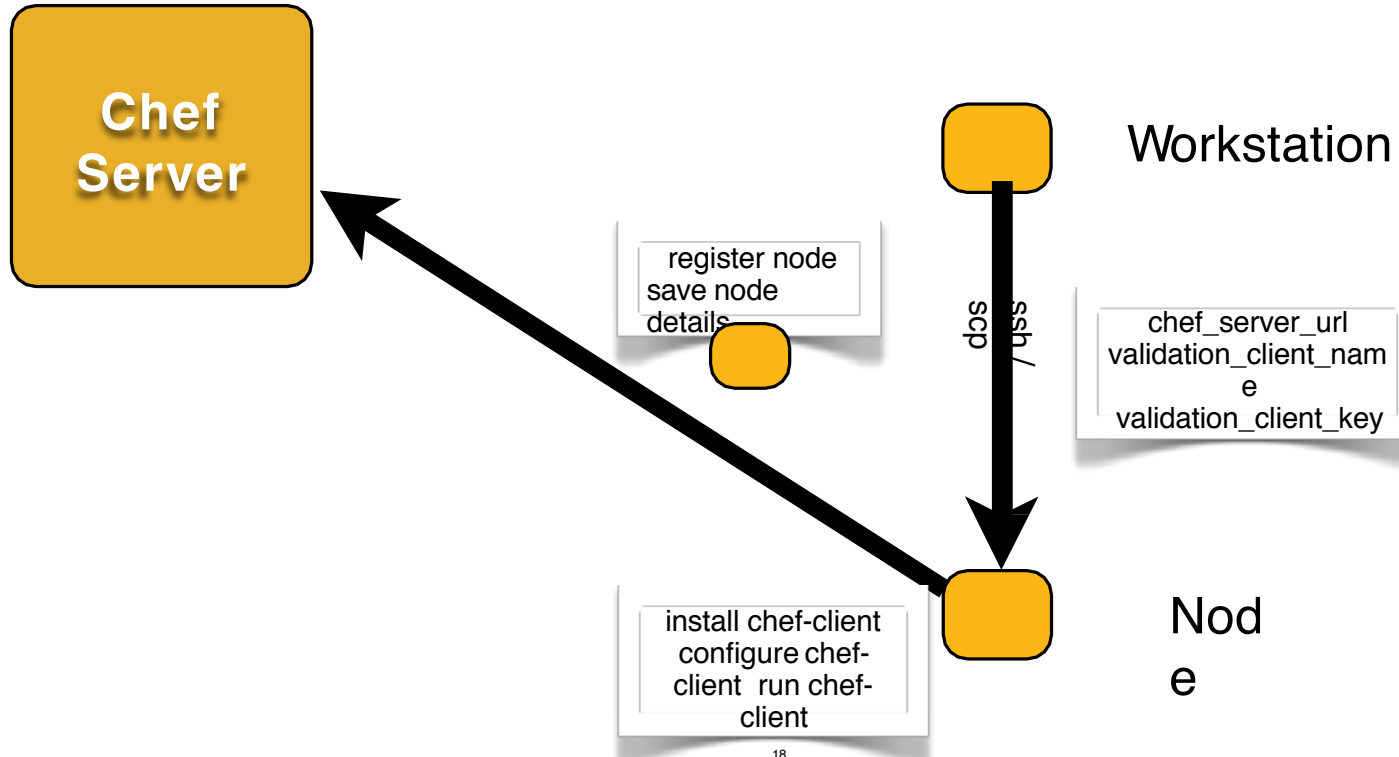


Node

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

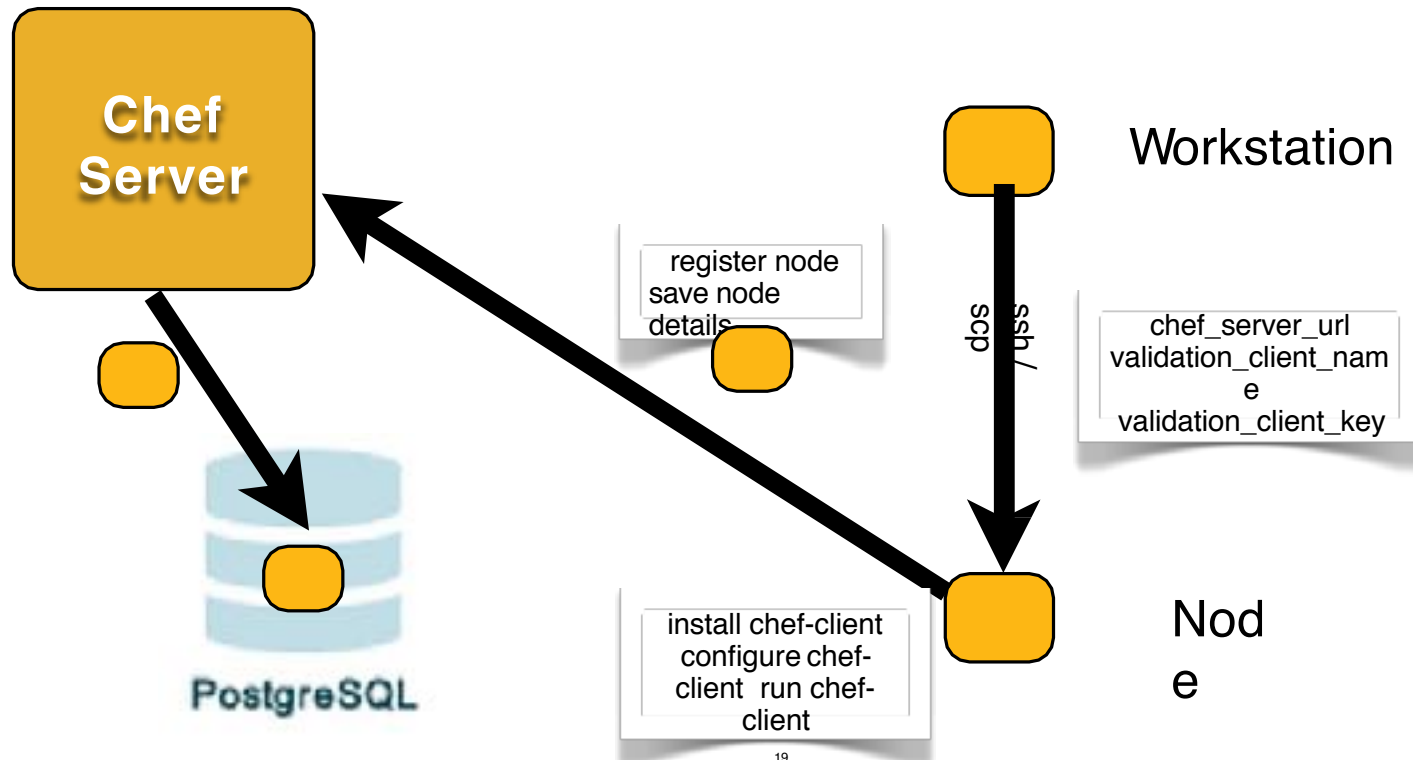
# knife bootstrap

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



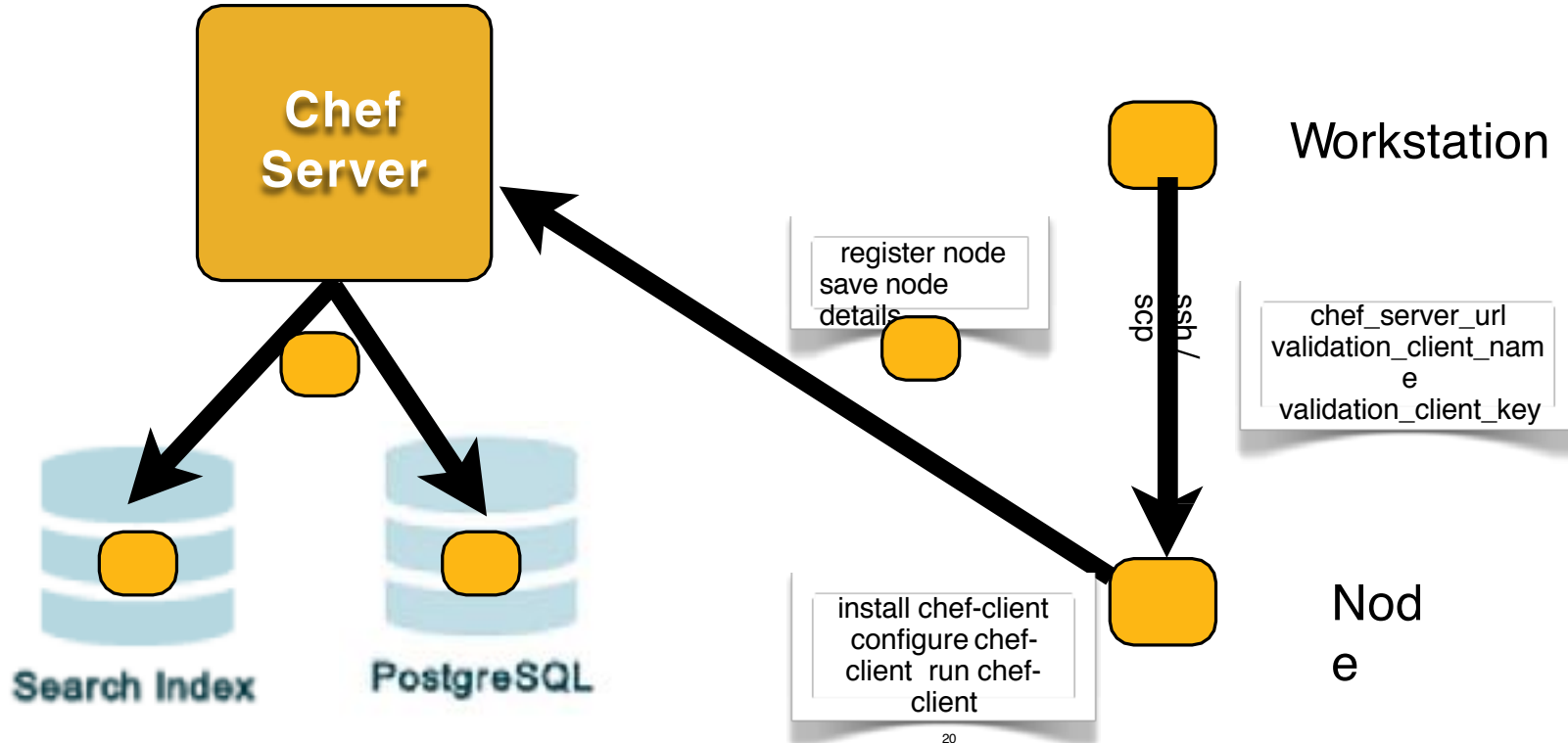
# knife bootstrap

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



# knife bootstrap

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



## 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

The screenshot shows the Chef Manage web interface. The top navigation bar includes 'Nodes', 'Reports', 'Policy', and 'Administration'. The left sidebar shows a list of actions for nodes: 'Delete', 'Manage Tags', 'Reset Key', 'Edit Run List', and 'Edit Attributes'. The main content area is titled 'Showing All Nodes' and displays a table of nodes. The first node, 'module2', is highlighted in orange. Below the table, the 'Nodes: module2' section is visible, with the 'Details' tab selected and highlighted in orange. The 'Details' tab shows two key metrics: 'Last Check In' and 'Uptime'.

Node Name	Platform	FQDN	IP Address
module2	centos	chefes01.example.com	10.1.60.201.90

Nodes: module2

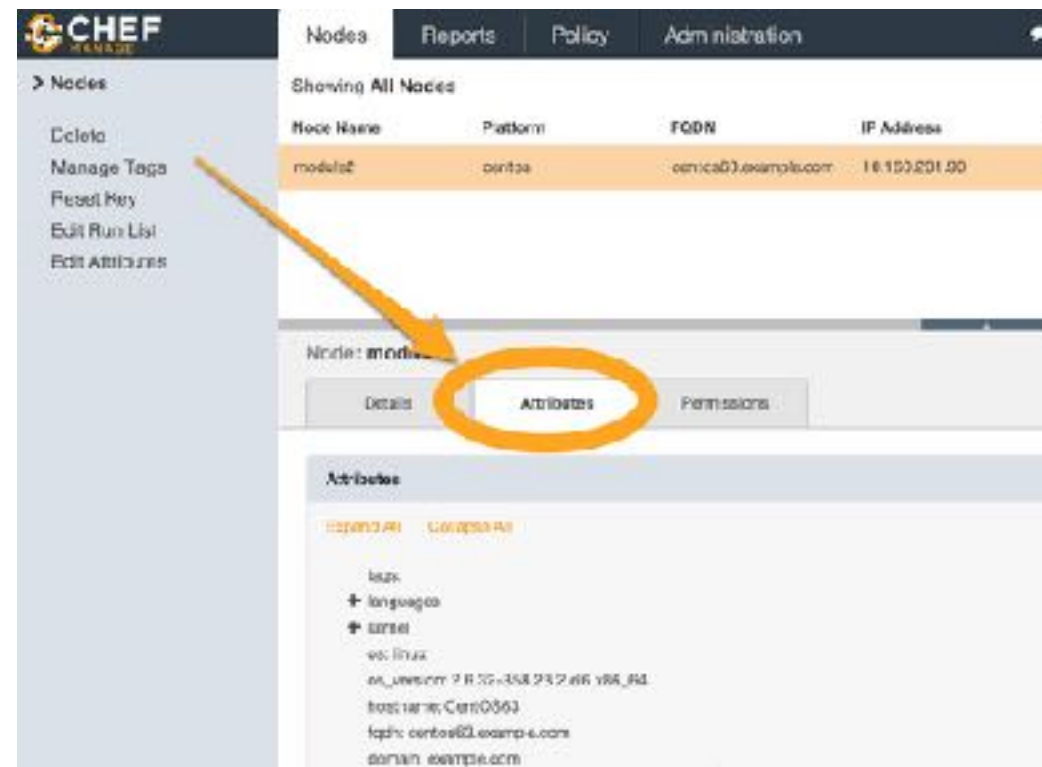
Details Attributes Permissions

Last Check In: **A Few Seconds Ago**  
2014-09-23 10:51:53 UTC

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

# View Node on Chef Server

- Click the 'Attributes' tab



The screenshot shows the Chef Manage web interface. On the left sidebar, under the 'Nodes' section, there are links for 'Create', 'Manage Tags', 'Reset Key', 'Edit Run List', and 'Edit Attributes'. An orange arrow points from the 'Edit Attributes' link to the 'Attributes' tab in the main content area. The main content area has a top navigation bar with 'Nodes', 'Reports', 'Policy', and 'Administration'. Below this, it says 'Showing All Nodes' and displays a table with columns 'Node Name', 'Platform', 'FQDN', and 'IP Address'. The table has one row with the values 'node123', 'centos', 'centos01.example.com', and '10.100.201.90'. Below the table, there are three tabs: 'Details', 'Attributes' (which is selected and circled in orange), and 'Permissions'. The 'Attributes' tab content shows a list of attributes: 'hostname', 'fqdn', 'lang', 'language', 'os', 'os\_linux', 'os\_version', 'hostname', 'fqdn', and 'domain'. The 'os' attribute is expanded, showing its details.

Node Name	Platform	FQDN	IP Address
node123	centos	centos01.example.com	10.100.201.90

Node: node123

Details Attributes Permissions

Attributes

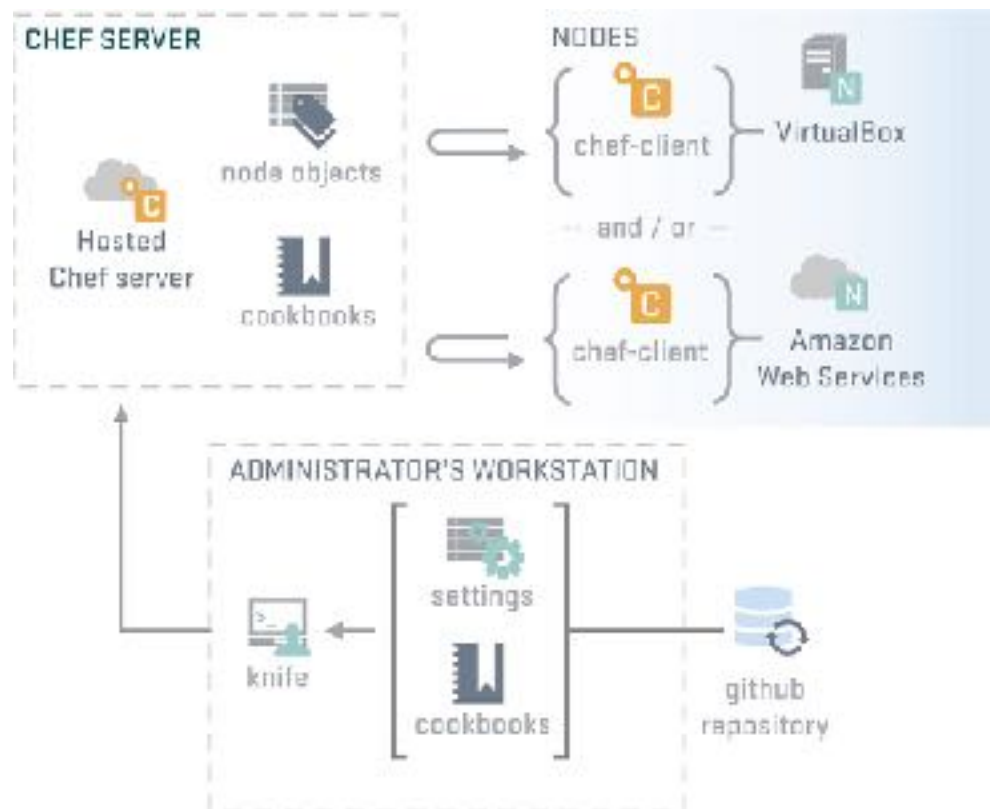
- hostname
- fqdn
- lang
- language
- os
- os\_linux
- os\_version: 2.6.32-358.el6.x86\_64
- hostname: Centos01
- fqdn: centos01.example.com
- domain: example.com

# 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 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
```

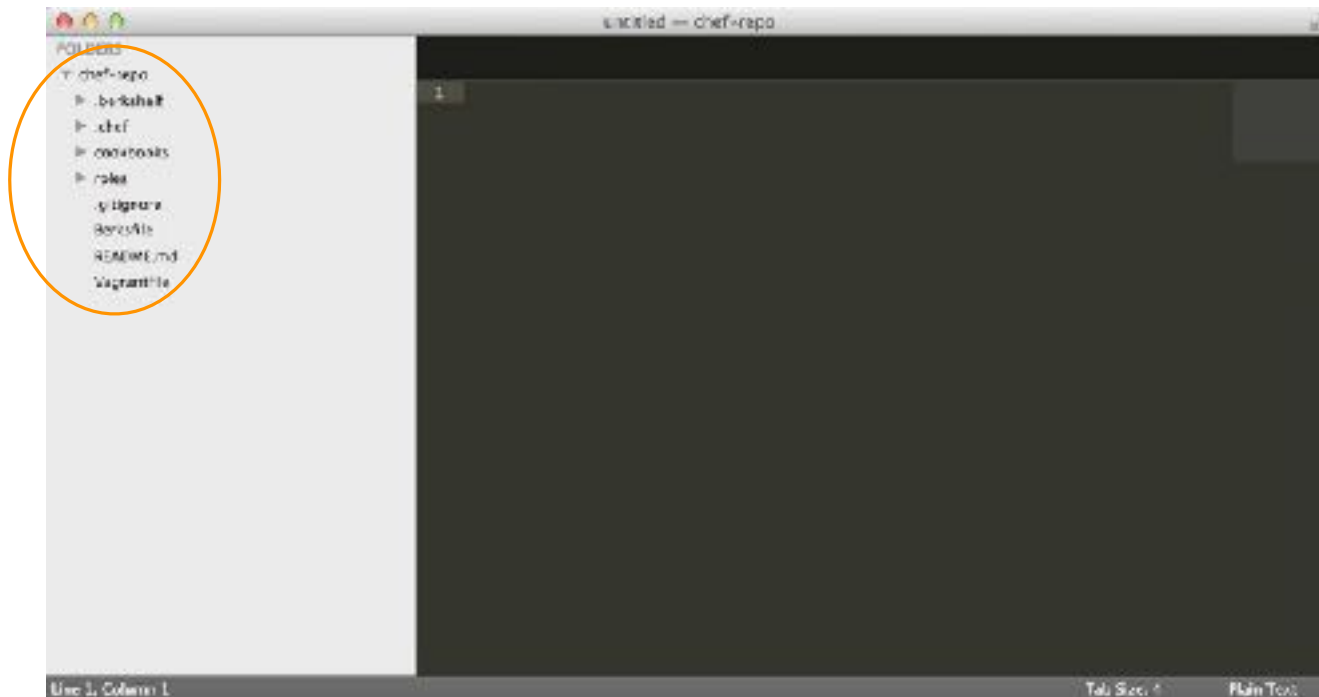
```
total 24
drwxr-xr-x 13 opcode opcode 442 Jan 24 21:25 .
drwxr-xr-x  5 opcode opcode 170 Jan 24 21:25 ..
-rw-r--r--  1 opcode opcode 412 Jan 24 21:25 CHANGELOG.md
-rw-r--r--  1 opcode opcode 1447 Jan 24 21:25 README.md
drwxr-xr-x  2 opcode opcode  68 Jan 24 21:25 attributes
drwxr-xr-x  2 opcode opcode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

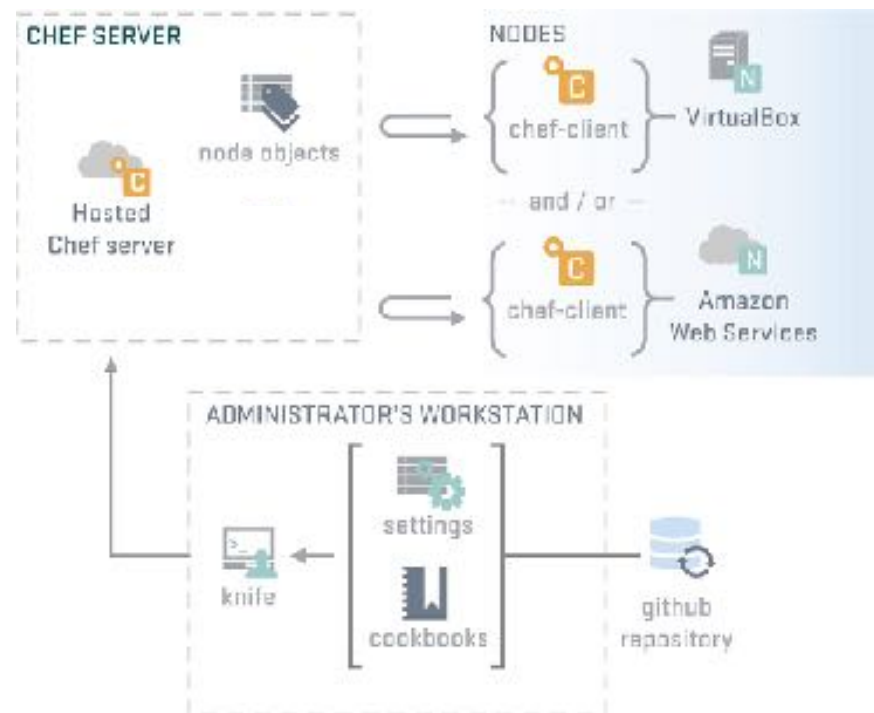
# Exercise: Upload the cookbook

```
$ knife cookbook upload apache
```

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

# Upload a cookbook

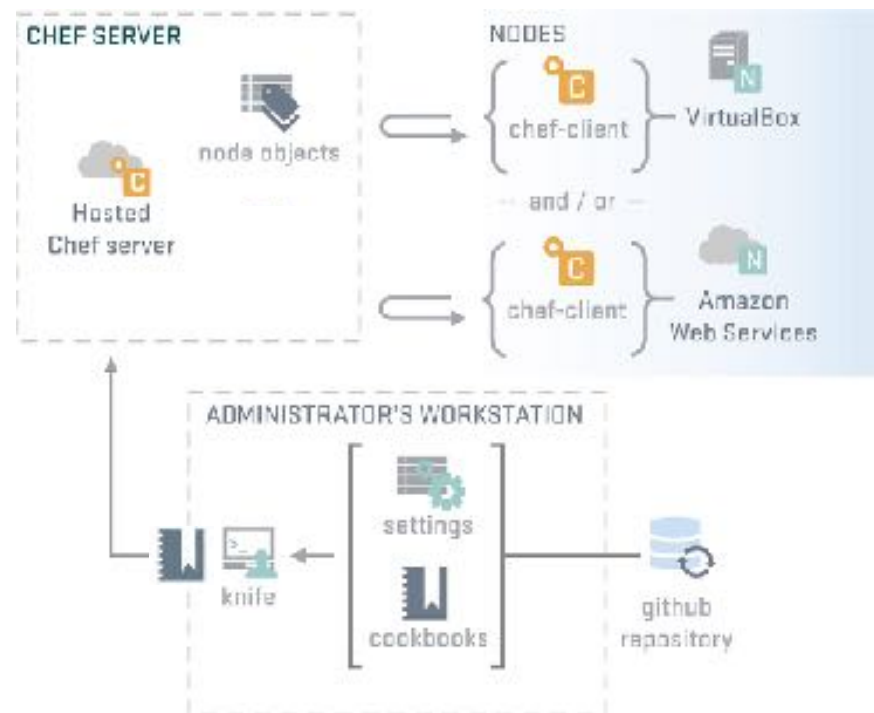
```
knife cookbook upload apache
```





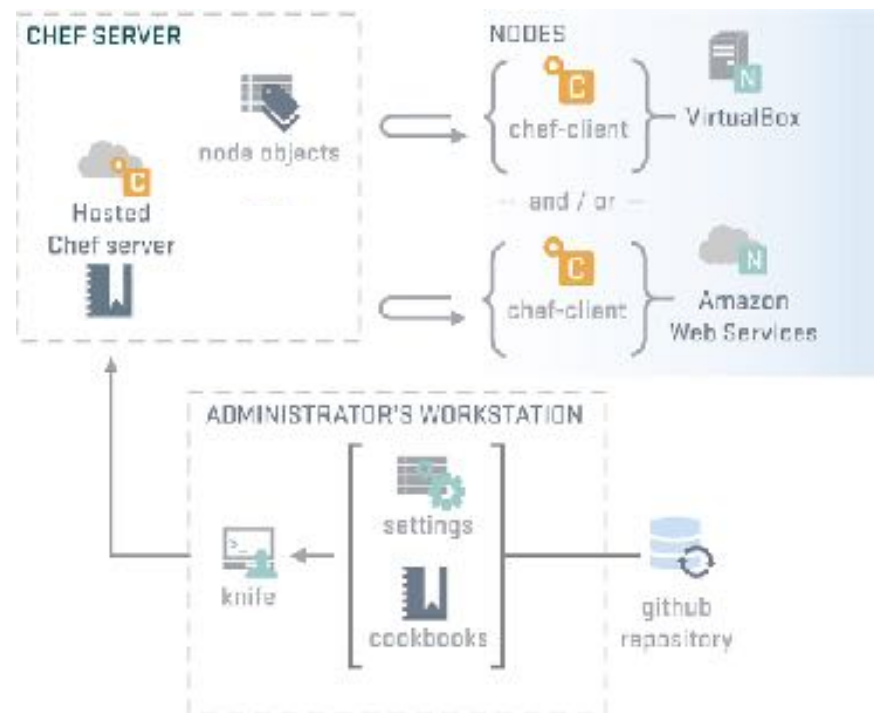
# Upload a cookbook

```
knife cookbook upload apache
```



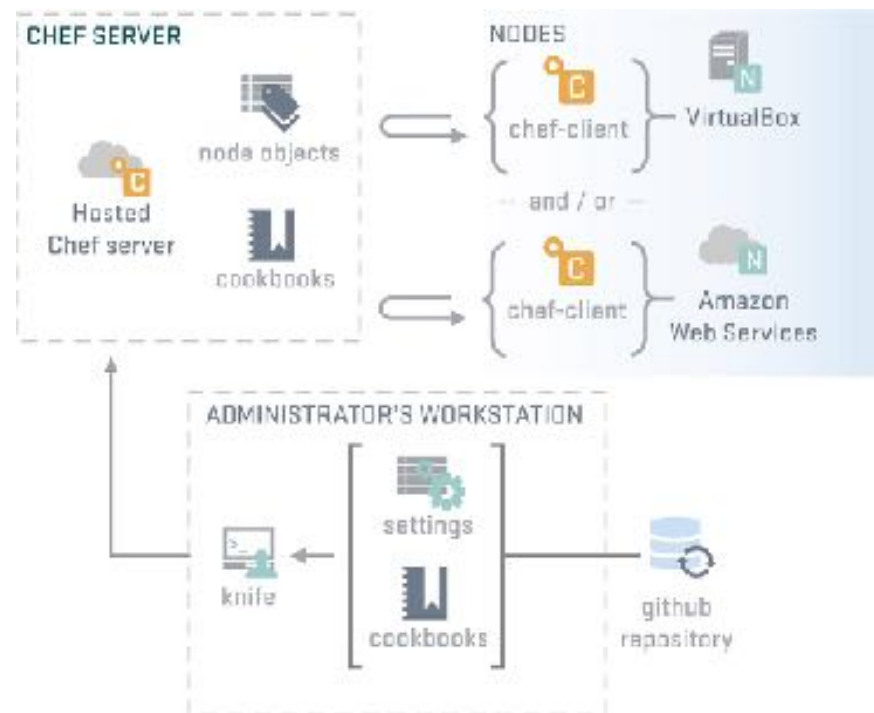
# Upload a cookbook

```
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*]**”

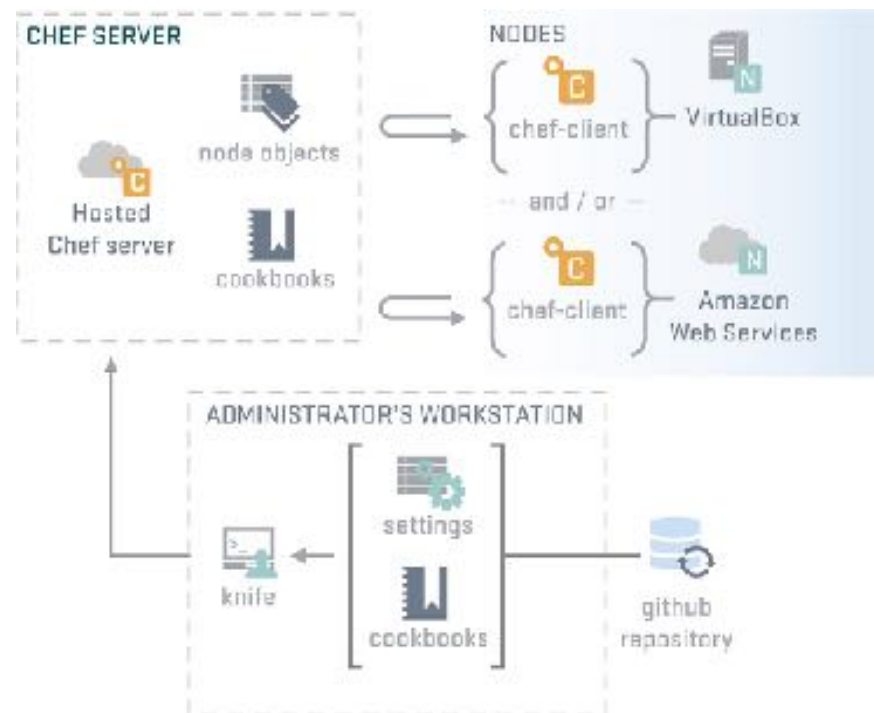
## Exercise: Add apache recipe to test node's run list

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

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

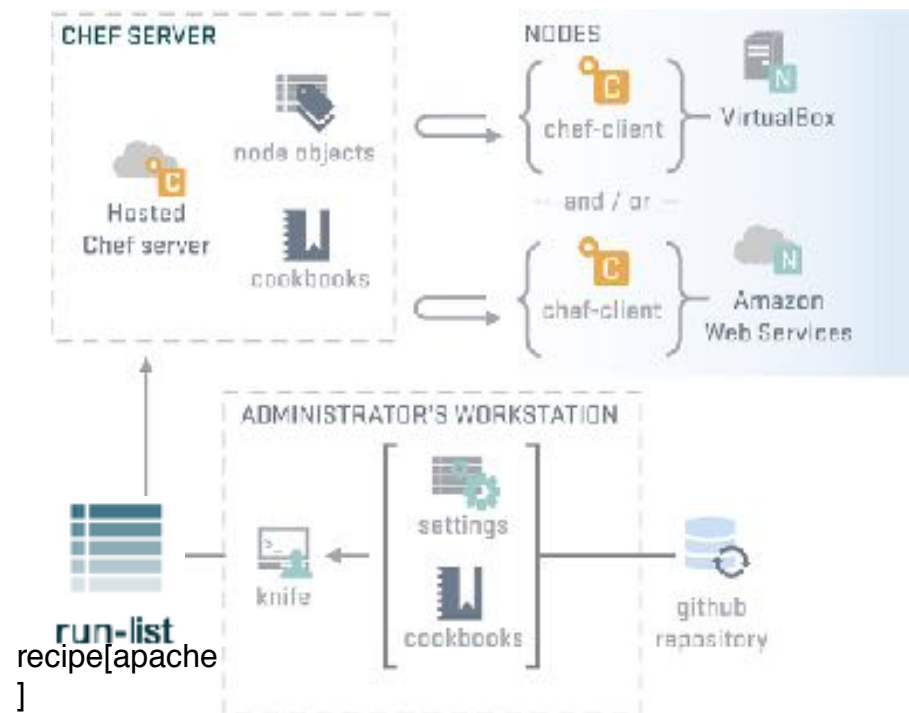
# Upload a cookbook

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



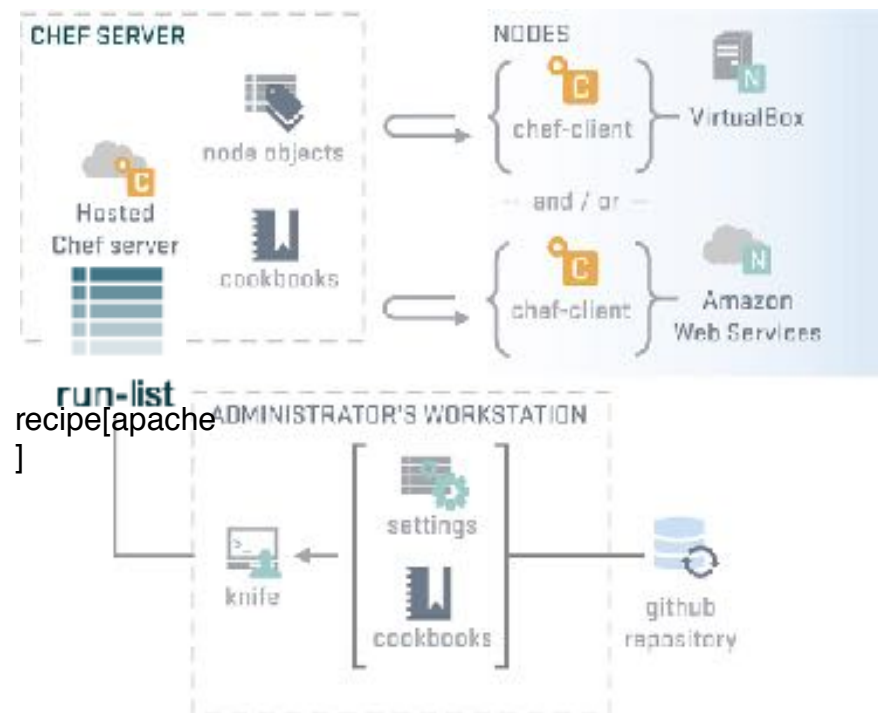
# Upload a cookbook

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



# Upload a cookbook

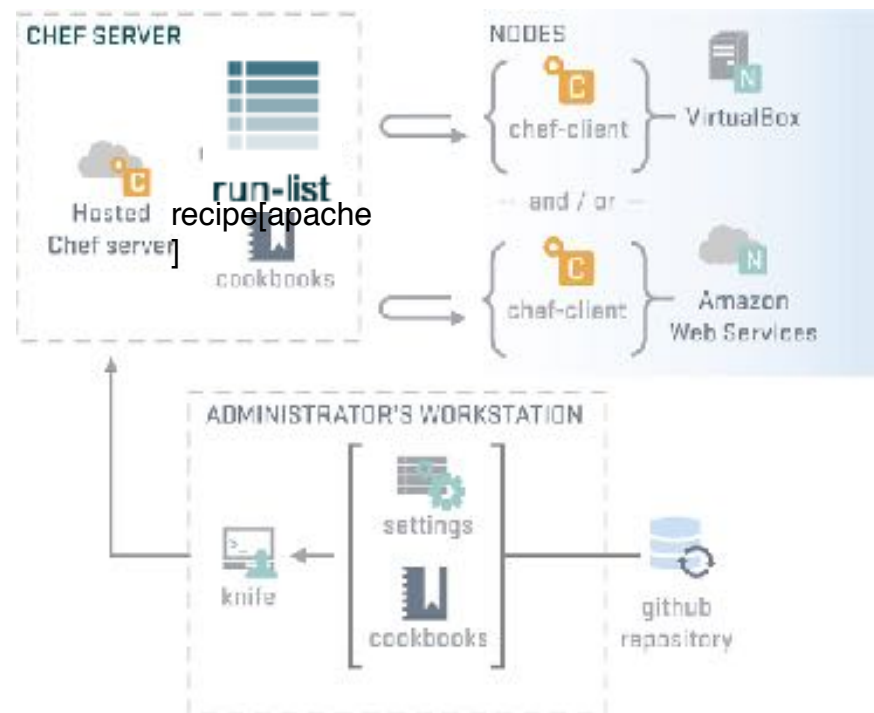
```
knife node run_list add module2 "recipe[apache]"
```





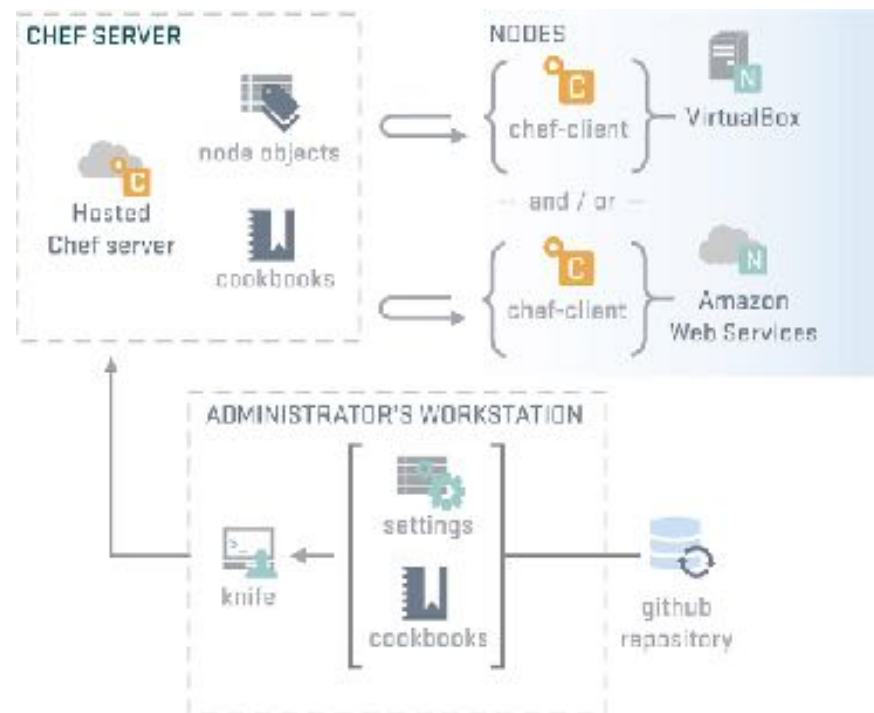
# Upload a cookbook

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



# Upload a cookbook

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



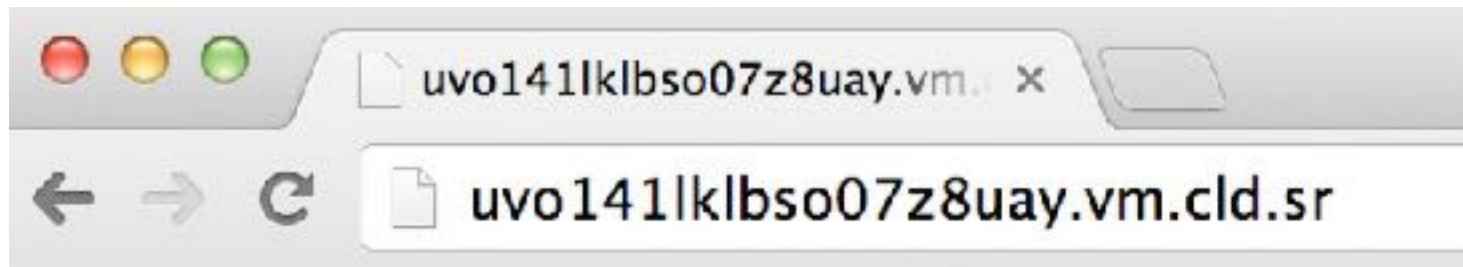
# Exercise: Run Chef Client

```
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
```

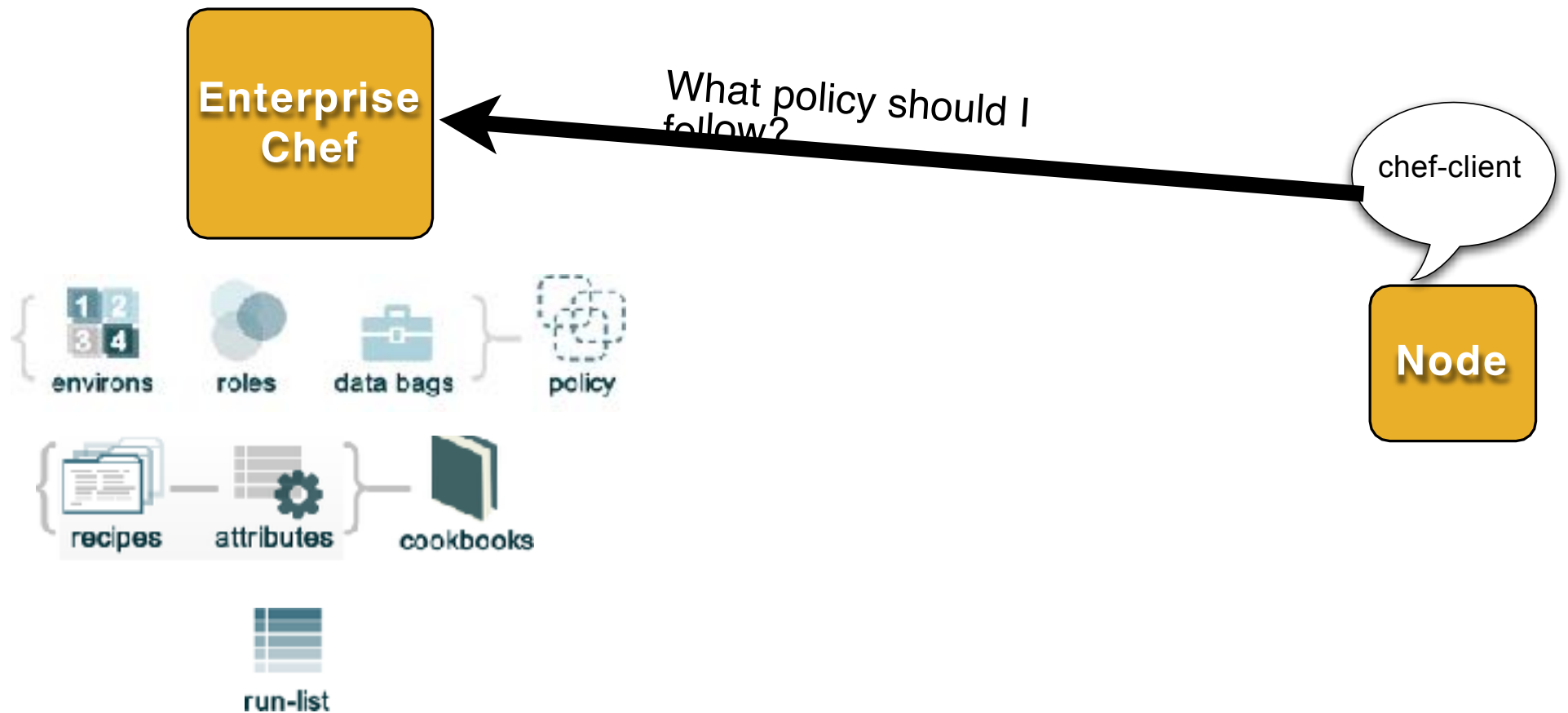
## Exercise: Verify that the home page works

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

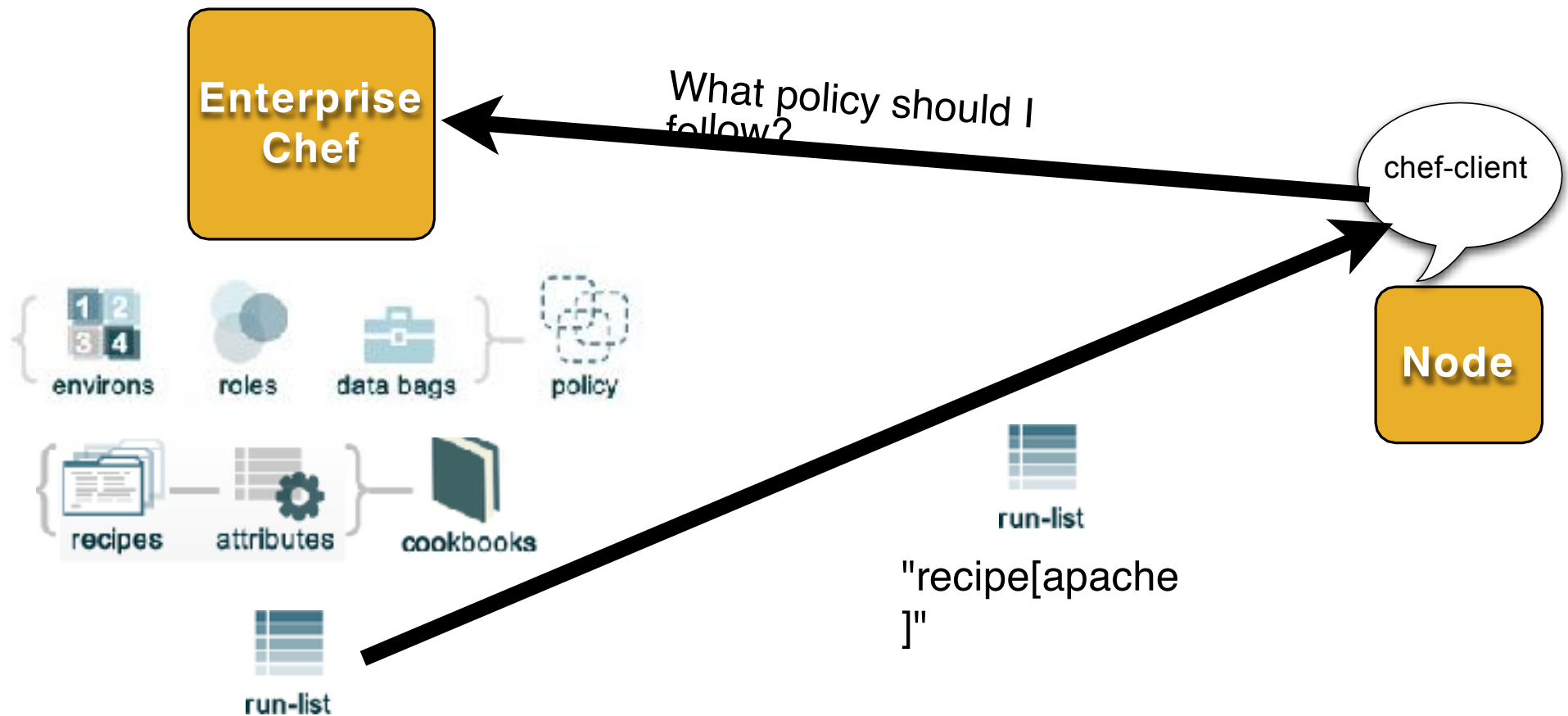


**Hello, world!**

# Run List



# Run List



# Run List

Enterprise Chef

What policy should I follow?

chef-client

Node

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



run-list

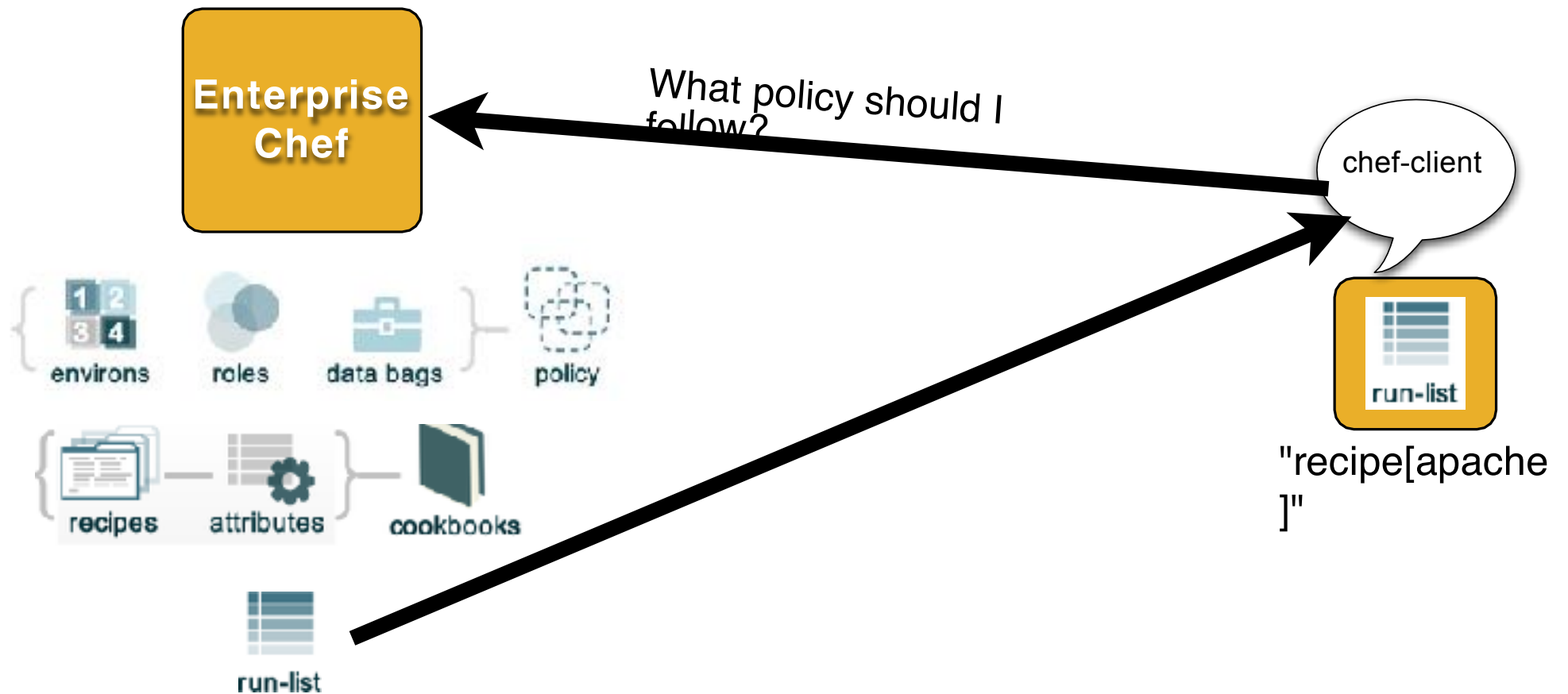
"recipe[apache  
]"

recipes attributes cookbooks



run-list

# Run List





# 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 non
    --- /var/www/html/index.html    2014-05-23 23:44:48.19
    +++ /tmp/chef-rendered-template20140523-42428-1471gt3
    @@ -1,2 @@
    +<h1>Hello, world!</h1>
  - change mode from '' to '0644'
  - restore selinux security context
```

chef-client

run-list

"recipe[apache  
]"

# 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>