The Chef Infra Server

A Hub for Configuration Data

1W

Objectives

After completing this module, you should be able to

- ➢ Connect your local workstation (laptop) to a Chef Infra Server
- ➢ Clone cookbooks from a GitHub repository
- ➢ Generate a Policyfile
- ➢ Generate a Policyfile.lock.json
- ➢ Push a Policy (and it’s required cookbooks) to a Chef Infra Server
- ➢ Bootstrap a node
- ➢ Manage a node via a Chef Infra Server

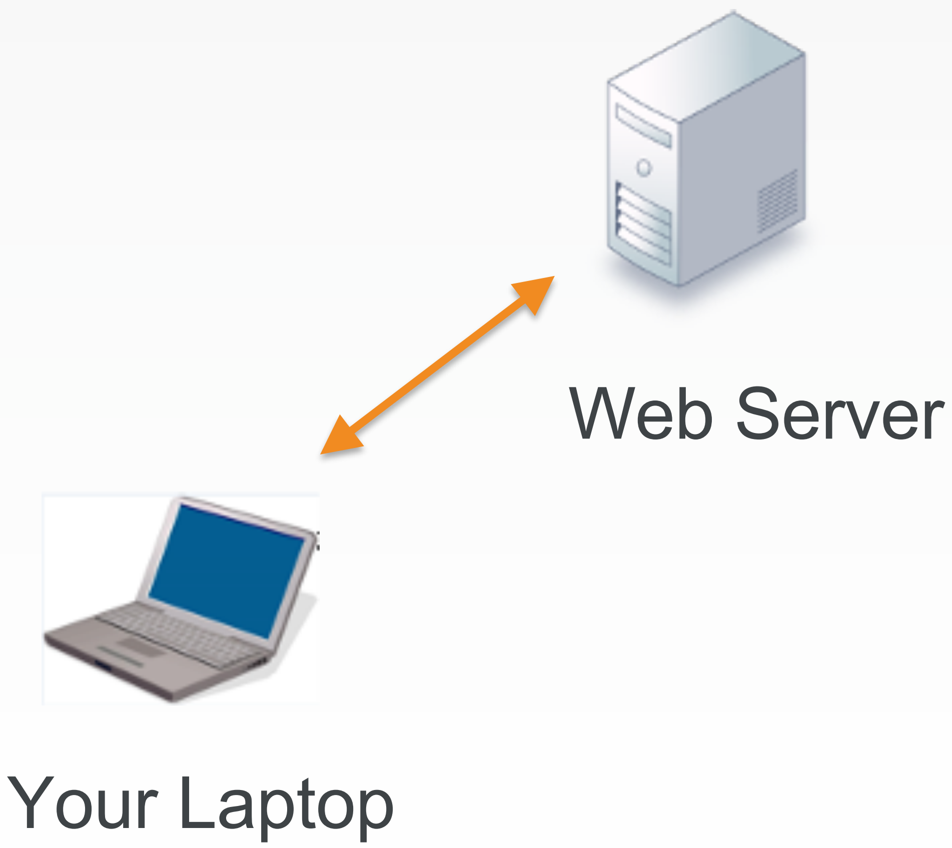
Managing an Additional System Without Chef Infra Server

To manage another system, you would need to:

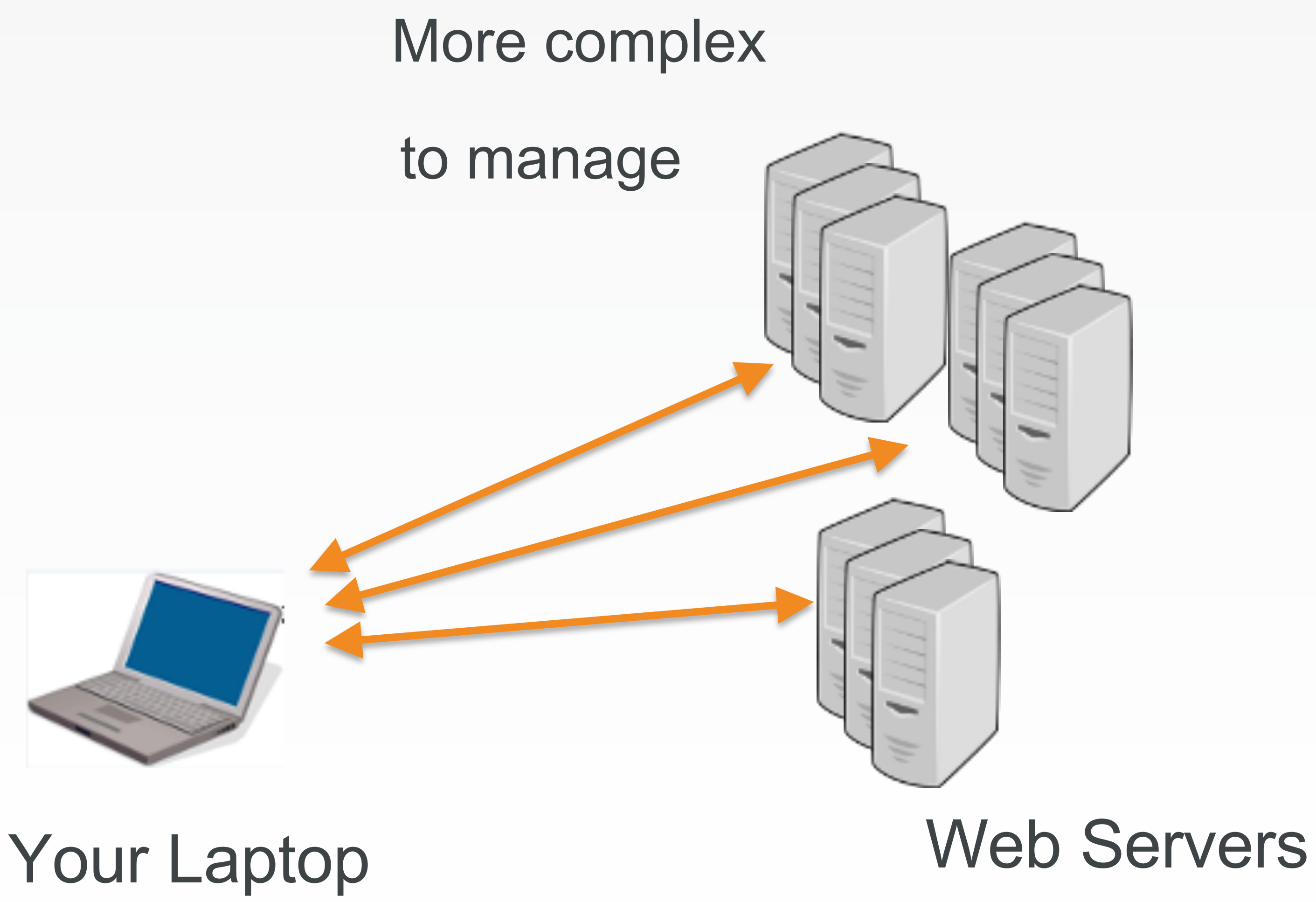
1. Provision a new node within your company or appropriate cloud provider with the appropriate access to login to administrate the system.
2. Install the Chef tools.
3. Transfer the apache or apache cookbook.
4. Run chef-client on the new node to apply the apache or apache cookbook's default recipe.

Managing Additional Systems

Now



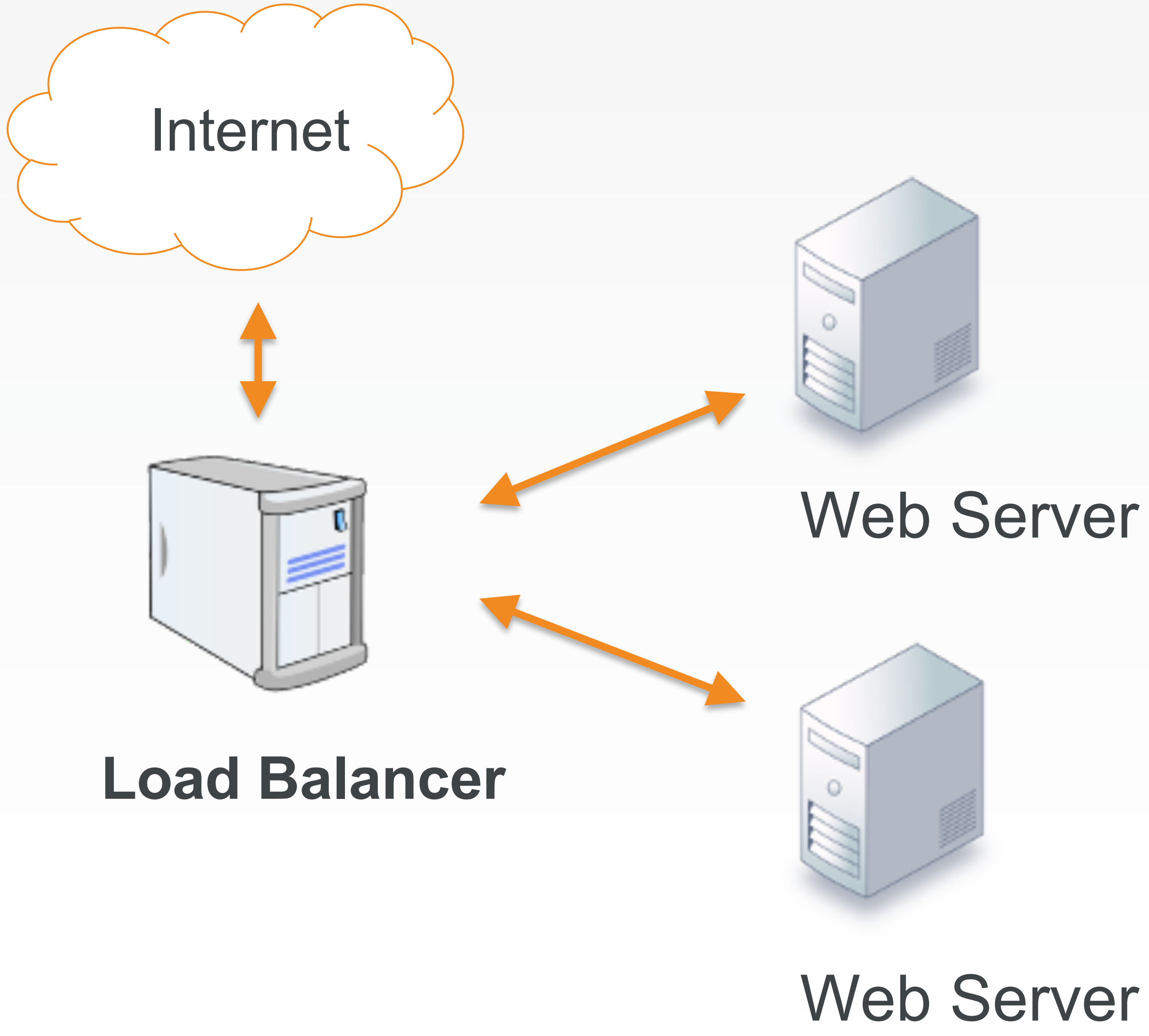
Future


More complex

to manage

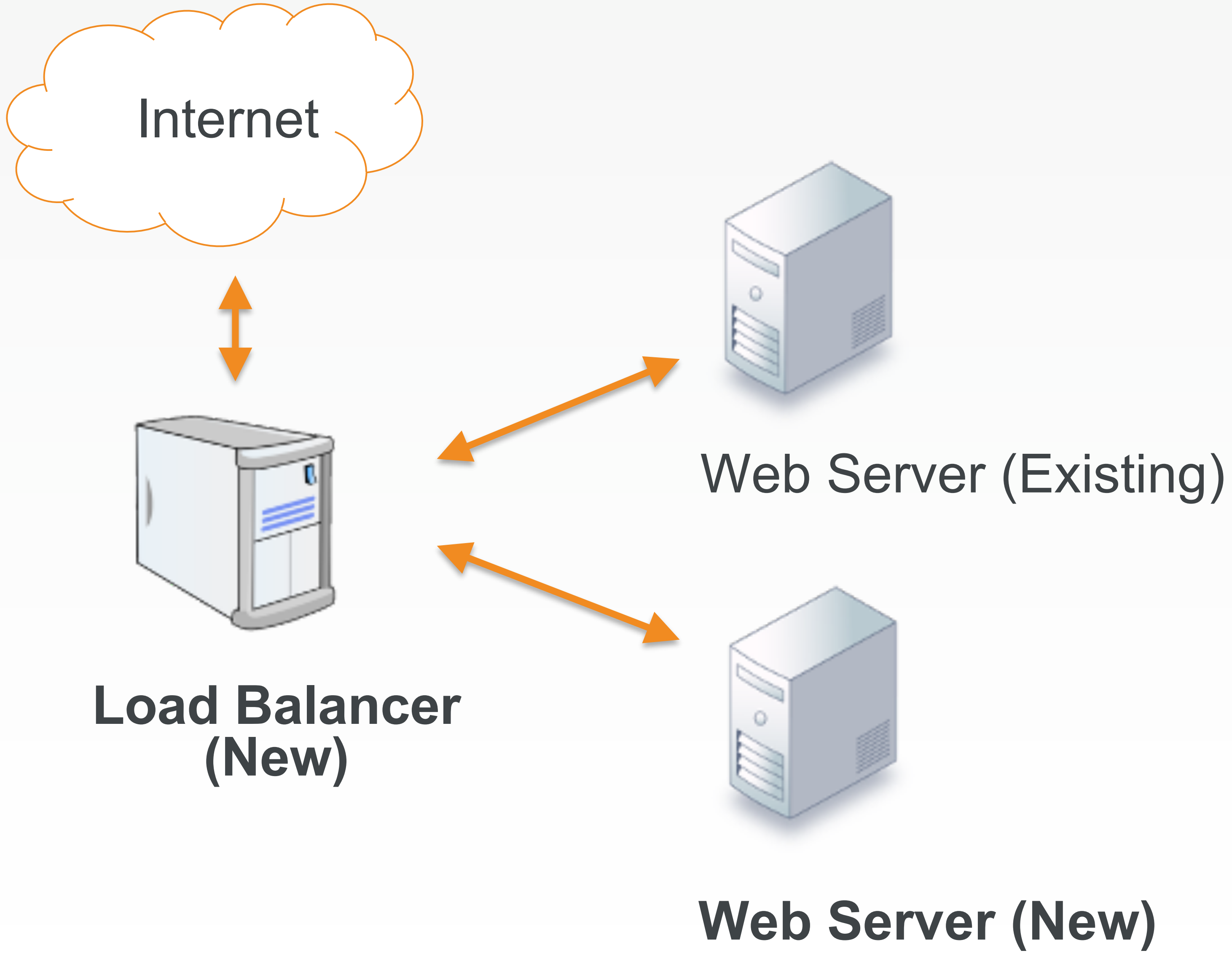
Managing User Traffic

A load balancer can forward incoming user web requests to other nodes.



Managing User Traffic

Today you will set up a new load balancer that will direct web requests to similarly-configured nodes.



Steps to Setup Load Balancer and Web Servers

Web Server

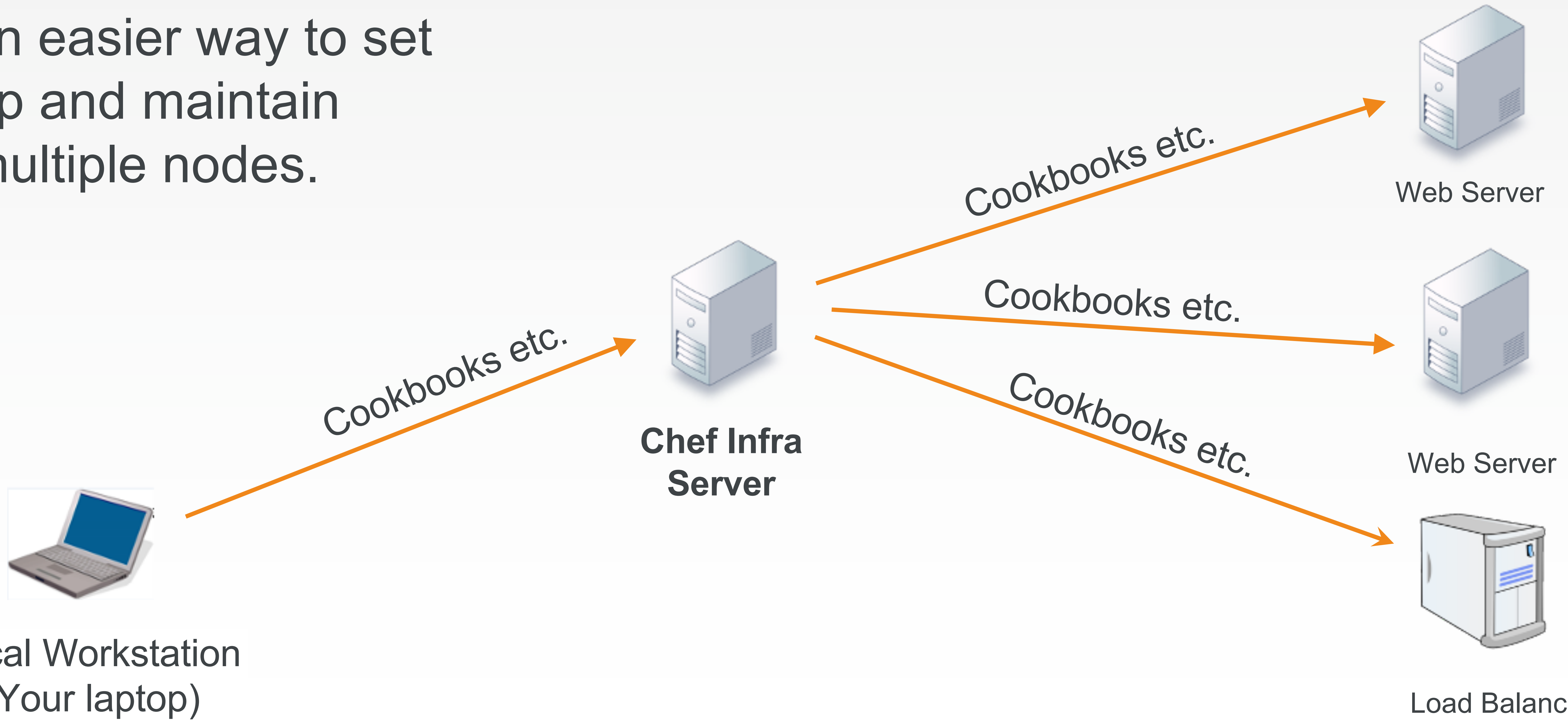
1. Provision the instance
2. Install Chef
3. Copy the web server cookbook
4. Apply the cookbook

Load Balancer

1. Create the haproxy (load balancer) cookbook
2. Provision the instance
3. Install Chef
4. Copy the haproxy cookbook
5. Apply the cookbook

The Chef Infra Server

An easier way to set up and maintain multiple nodes.



GL: Managing Nodes with Hosted Chef

It will be easier to distribute the cookbooks we write to multiple nodes if we store them in a central repository.

Objective:

- ❑ Download and copy the required cookbooks to your local machine
- ❑ Upload the 'apache' cookbook to the Chef Infra Server via a Policyfile.lock.json

GL: Code Repository

This GitHub repository contain copies of the work that you have done up to this point for the ‘apache’ and ‘apache’ cookbooks:

https://github.com/shekhar2010us/chef-essentials-repo-15

GL: Managing Nodes with Hosted Chef

It will be easier to distribute the cookbooks we write to multiple nodes if we store them in a central repository.

Objective:

✓ Download and copy the required cookbooks to your local machine

❑ Upload the ‘apache’ cookbook to the Chef Infra Server via a Policyfile.lock.json

GL: Navigate to the cookbooks Directory

```
$ cd ~/chef-repo/cookbooks
$ ls
```

```
myiis    chefignore    apache    starter
```

GL: Remove the starter Example Cookbook

```
$ rm -rf starter
```

Note: Powershell users can use **'Remove-Item starter'**

GL: Navigate to the 'apache' Directory

```
$ cd ~/chef-repo/cookbooks/apache
$ ls -l (or dir if using Powershell)
```

```
total 56
-rwxr-xr-x@ 1 sdelfante  staff   148 Jul 10 07:33 CHANGELOG.md
-rwxr-xr-x@ 1 sdelfante  staff    70 Jul 10 07:33 LICENSE
-rwxr-xr-x@ 1 sdelfante  staff   504 Jul 10 07:33 Policyfile.rb
-rwxr-xr-x@ 1 sdelfante  staff    53 Jul 10 07:33 README.md
-rwxr-xr-x@ 1 sdelfante  staff  1176 Jul 10 07:33 chefignore
-rwxr-xr-x@ 1 sdelfante  staff   741 Jul 10 07:33 kitchen.yml
-rwxr-xr-x@ 1 sdelfante  staff   717 Jul 10 07:33 metadata.rb
drwxr-xr-x@ 4 sdelfante  staff   128 Jul 10 07:33 recipes
drwxr-xr-x@ 4 sdelfante  staff   128 Jul 10 07:33 spec
drwxr-xr-x@ 3 sdelfante  staff    96 Jul 10 07:33 templates
drwxr-xr-x@ 3 sdelfante  staff    96 Jul 10 07:33 test
```

GL: Managing Nodes with Hosted Chef

It will be easier to distribute the cookbooks we write to multiple nodes if we store them in a central repository.

Objective:

❑ Download and copy the required cookbooks to your local machine

❑ Upload the 'apache' cookbook to the Chef Infra Server via a Policyfile.lock.json

Policyfile.rb and the Policyfile.lock.json

Now that we have our cookbooks in our chef-repo, we can create our Policyfile.rb and then generate our Policyfile.lock.json as we discussed in the Policyfiles module.

GL: Creating the Policyfile.rb and the Policyfile.lock.json

Objective:

- ❑ Create the Policyfile.rb
- ❑ Edit the Policyfile.rb
- ❑ Generate the Policyfile.lock.json

GL: Generate the Policyfile

```
> cd ~/chef-repo
> chef generate policyfile workstation
```

```
Recipe: code_generator::policyfile
  * template[/Users/sdelfante/chef-repo/workstation.rb] action create (up to
date)
```

A Note About Policyfile Naming

Give Policyfile a file name that will help you identify its purpose when viewed in your local chef-repo. For example, **chef generate policyfile *workstation*** will create a Policyfile named **workstation.rb**.

A Note About Policyfile Location

In this course we will be creating only three Policyfiles so we are creating them at the ~/chef-repo location.

However, in practice you may have a greater number of Policyfiles so you should store your Policyfiles in a in a directory called **policyfiles**.

For example at: **~/chef-repo/policyfiles/**

GL: Verify that the Policyfile Exists

```
> ls (or dir for Windows)
```

```
workstation.rb README.md cookbooks roles
```

GL: Edit the New Policyfile

~/chef-repo/workstation.rb

```
# Policyfile.rb - Describe how you want Chef Infra Client to build your system.
#...skipping for brevity…
# https://docs.chef.io/policyfile.html
# A name that describes what the system you're building with Chef does.
name 'workstation'

# Where to find external cookbooks:
default_source :supermarket

# run_list: chef-client will run these recipes in the order specified.
run_list 'workstation::default'

# Specify a custom source for a single cookbook:
cookbook 'workstation', path: 'cookbooks/workstation'
```

Replace the contents of the Policyfile.rb below the `# https://docs.chef.io/policyfile.html` line with the code in green.

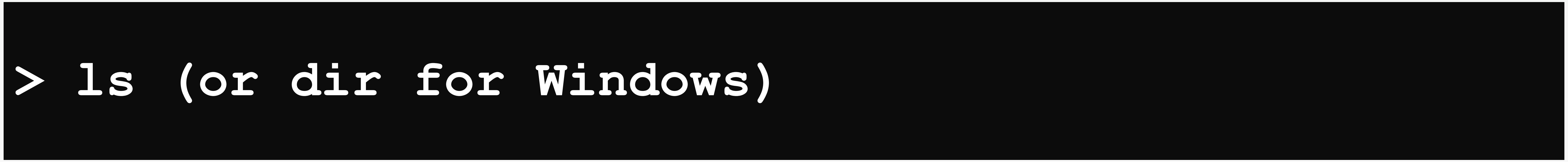
GL: Generate the Policyfile.lock.json

```
~/chef-repo> chef install workstation.rb
```

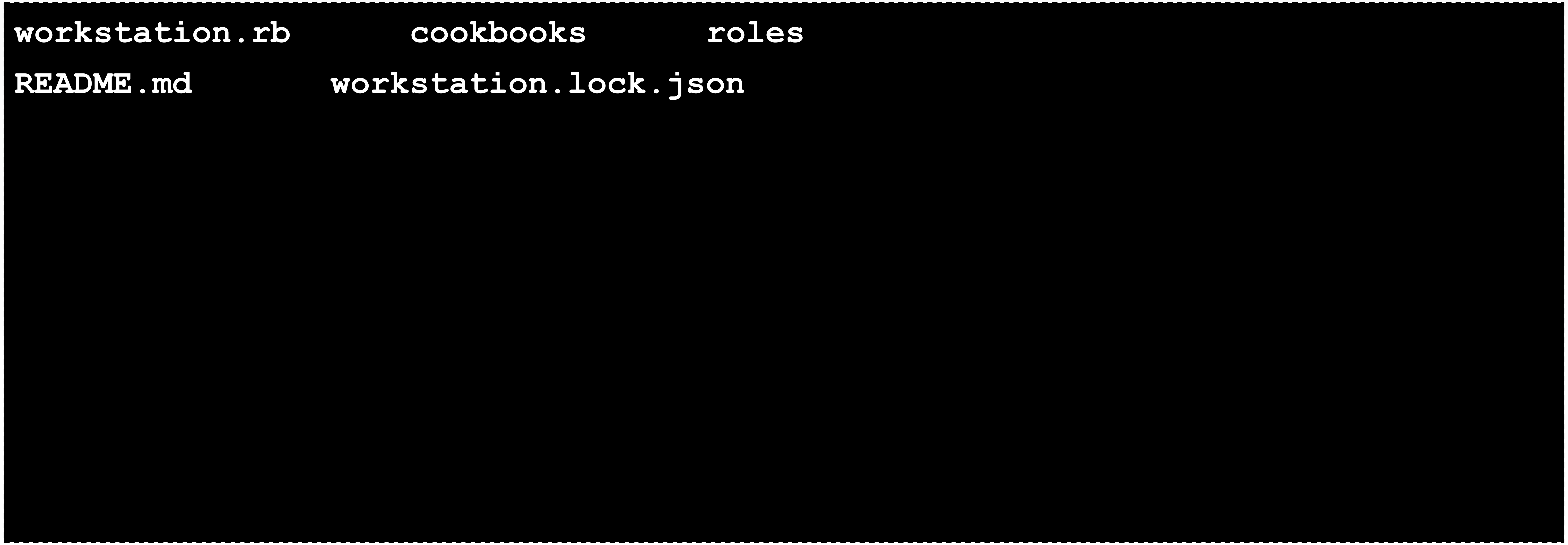
```
Building policy workstation
Expanded run list: recipe[workstation::default]
Caching Cookbooks...
Installing workstation >= 0.0.0 from path

Lockfile written to /Users/centos/chef-repo/workstation.lock.json
Policy revision id:
49eef2f1f18967a5a3be025346c58f129b9eea5e2f3c914516ccc7a67deb2b80
```

GL: Verify that the Policyfile.lock.json Exists



```
> ls (or dir for Windows)
```



```
workstation.rb          cookbooks          roles
README.md          workstation.lock.json
```

GL: View the New workstation.lock.json

`~/chef-repo/workstation.lock.json`

```
{
  "revision_id": "49eef2f1f18967a5a3be025346c58f129b9eea5e2f3c914516ccc7a67deb2b80",
  "name": "workstation",
  "run_list": [
    "recipe[workstation::default]"
  ],
  "included_policy_locks": [

  ],
  "cookbook_locks": {
    "workstation": {
      "version": "0.1.0",
      "identifier": "cd0db3edaf69e3616516c189c75c0d94f4a87b84",

      ...
```

These values were derived from the **`workstation.rb`** that you created a few moments ago.

GL: View the New workstation.lock.json

~/chef-repo/workstation.lock.json

```
…
      "dotted_decimal_identifier": "57717436663687651.27414221151651676.14933411003268",
      "source": "cookbooks/workstation",
      "cache_key": null,
      "scm_info": null,
      "source_options": {
        "path": "cookbooks/workstation"
      }
    }
  },
  "default_attributes": {


  …
```

These values were derived from the **workstation**.rb that you created a few moments ago.

GL: View the New workstation.lock.json

~/chef-repo/workstation.lock.json

```
…
},
  "override_attributes": {

  },
  "solution_dependencies": {
    "Policyfile": [
      [
        "workstation",
        ">= 0.0.0"
      ]
    ],
    "dependencies": {
      "workstation (0.1.0)": [
      ]
    }
```

These values were derived from the **workstation**.rb that you created a few moments ago.

GL: Creating the workstation.rb and the workstation.lock.json

Objective:

- ✓ Create the workstation.rb
- ✓ Edit the workstation.rb
- ✓ Generate the workstation.lock.json

GL: Pushing the workstation.lock.json to Chef Infra Server

Objective:

- □ Use `chef push` to upload a workstation.lock.json
- □ Use `chef show-policy` to verify it’s on Chef Infra Server

GL: Push the workstation.lock.json to Chef Infra Server

```
~/chef-repo> chef push prod workstation.lock.json
```

```
Uploading policy workstation (49eef2f1f1) to policy group prod
Uploaded workstation 0.1.0 (cd0db3ed)knife client edit CLIENT (options)
```

GL: Verify the workstation Policy is on Chef Infra Server

```
~/chef-repo> chef show-policy
```

```
workstation
=====

* prod:   49eef2f1f1
```

Here we can see that the **workstation** Policy has been uploaded to Chef Infra Server and is in the **prod** policy group.

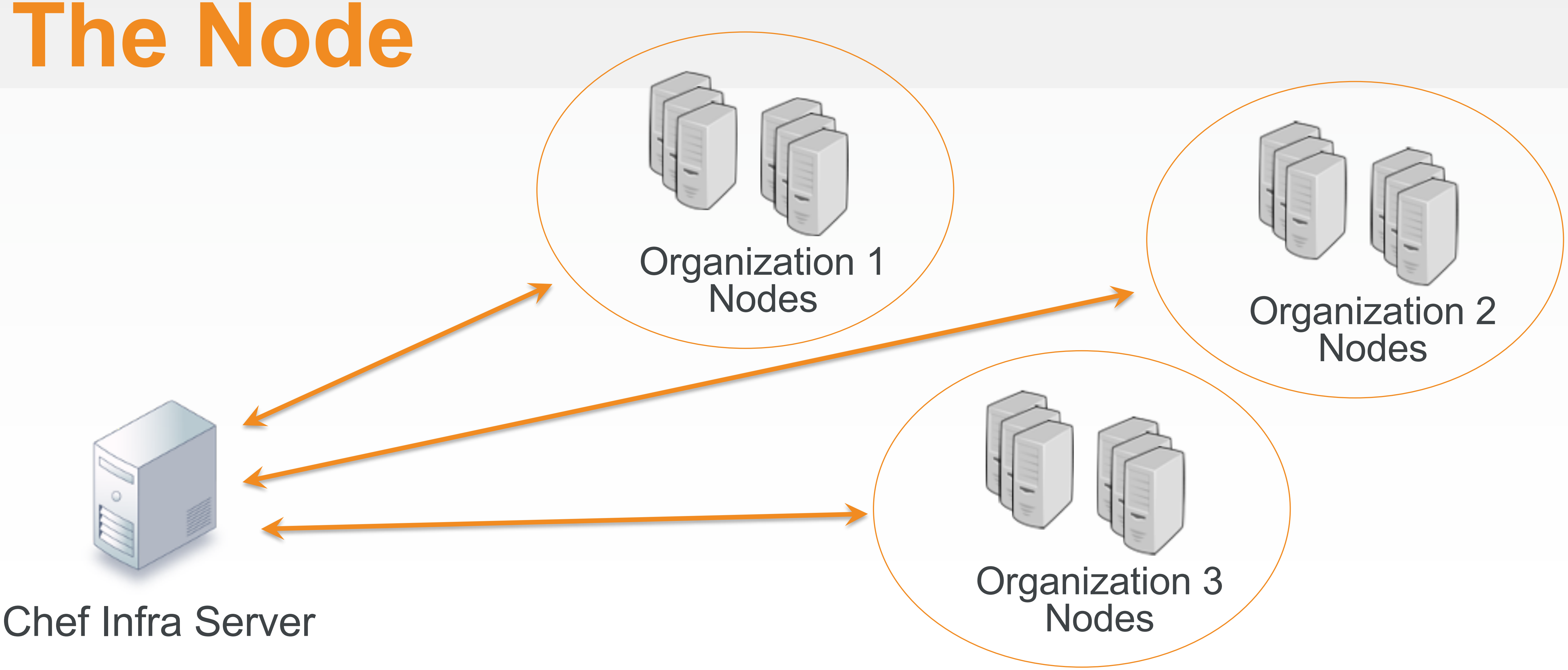
Also notice the policy name that was derived from the contents of the **`workstation.lock.json`.**

GL: Verify the workstation Policy is on Chef Infra Server

```
~/chef-repo> chef show-policy workstation prod
```

```
--- Content of workstation.lock.json ---
```

The Node



Organization 1 Nodes

Organization 2 Nodes

Organization 3 Nodes

Chef Infra Server

Upload apache cookbook to server

GL: Bootstrap Your Node

In this lab you will use a new instance and bootstrap it as a managed node.

You'll need the FQDN or Public IP of that instance to perform this lab.

Bootstrapping a Node

Often, the node you are bootstrapping may not have Chef installed. It may also not have details of where the Chef Infra Server is located or the credentials to securely talk to that Server.

To add those credentials we can **bootstrap** that node to install all those components.

https://learn.chef.io/modules/beyond-the-basics#/

GL: Change to the chef-repo

```
$ cd ~/chef-repo
```

knife

knife is a command-line tool that provides an interface between a local chef-repo and the Chef Infra Server.

GL: Run 'knife node –help'

```
$ knife node --help
```

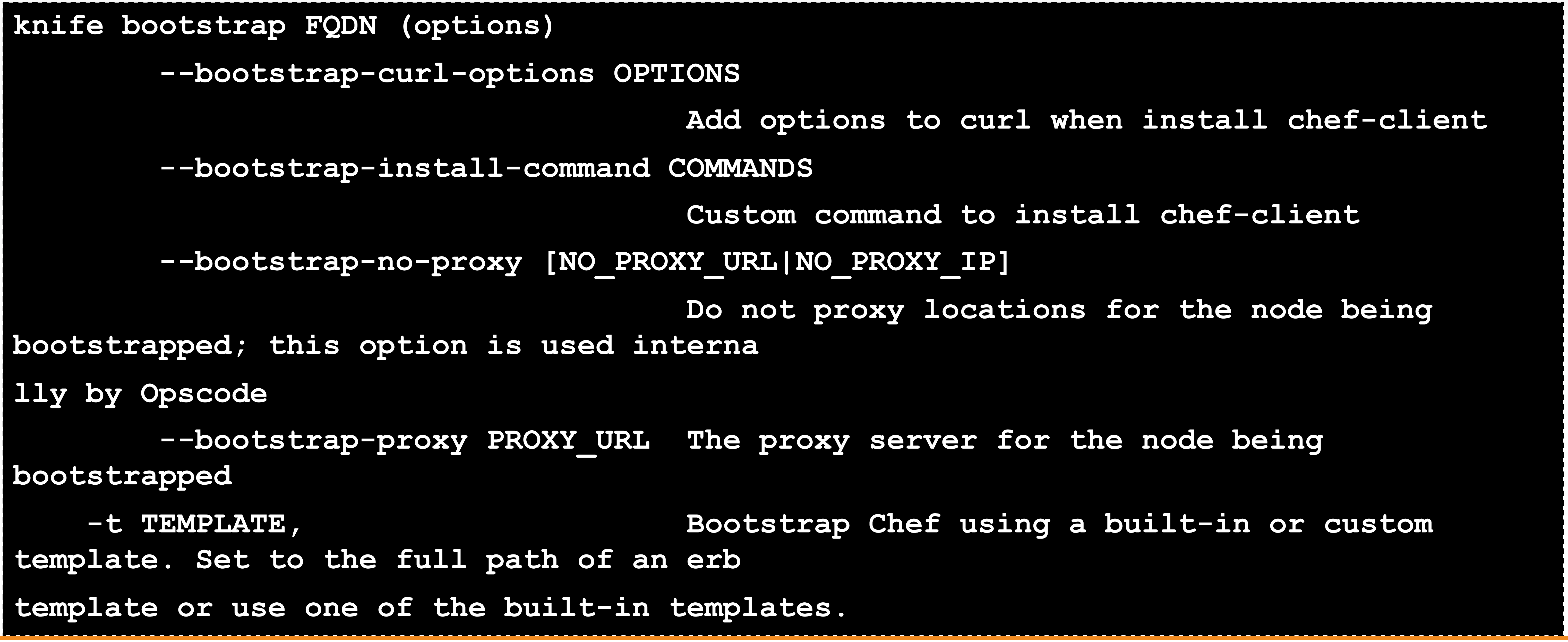
```
** NODE COMMANDS **
knife node bulk delete REGEX (options)
knife node create NODE (options)
knife node delete NODE (options)
knife node edit NODE (options)
knife node environment set NODE ENVIRONMENT
knife node from file FILE (options)
knife node list (options)
knife node run_list add [NODE] [ENTRY[,ENTRY]] (options)
knife node run_list remove [NODE] [ENTRY[,ENTRY]] (options)
knife node run_list set NODE ENTRIES (options)
knife node show NODE (options)
```

GL: Run 'knife node list'

```
$ knife node list
```

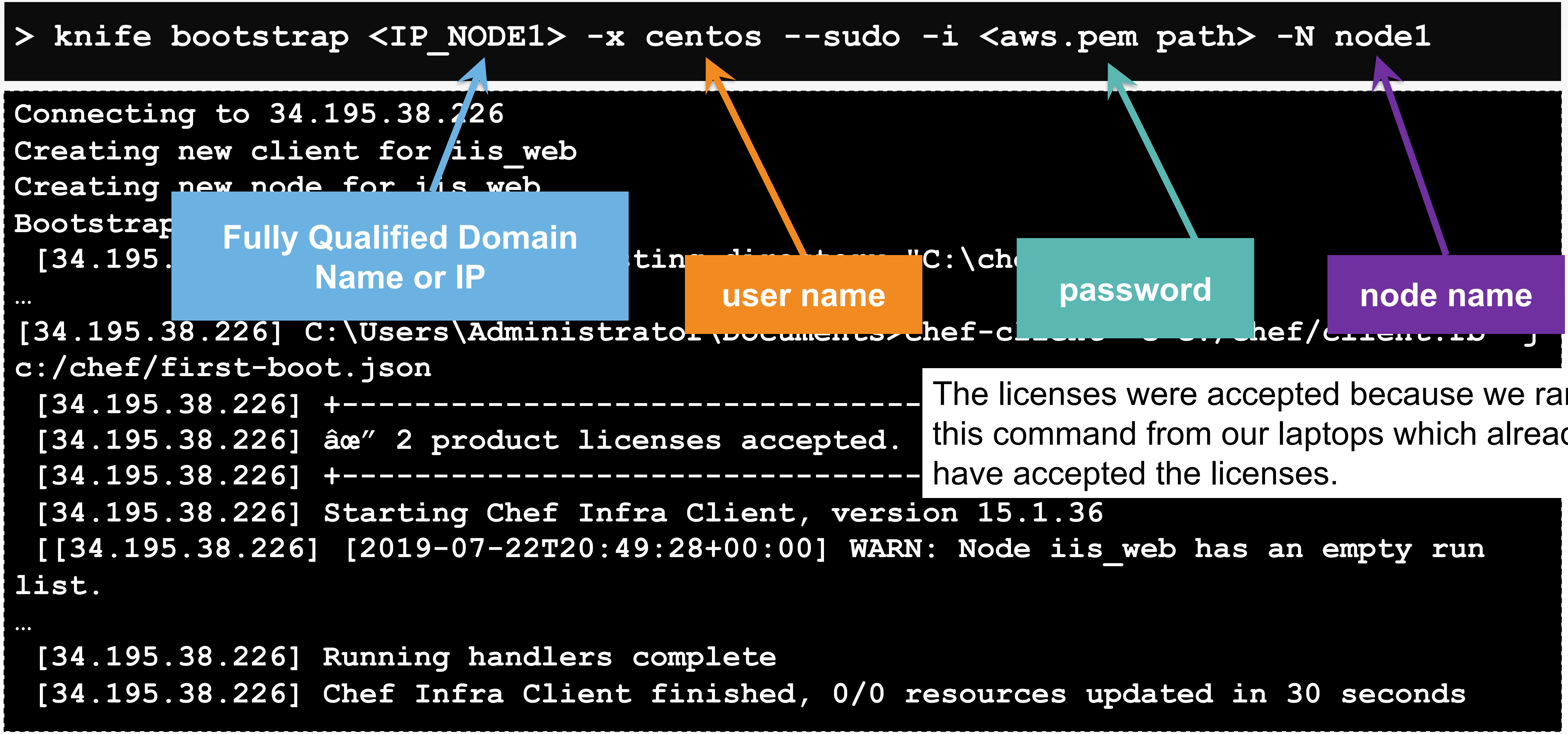
GL: Run 'knife bootstrap –help'

```
$ knife bootstrap --help
```



```
knife bootstrap FQDN (options)
        --bootstrap-curl-options OPTIONS
                                     Add options to curl when install chef-client
        --bootstrap-install-command COMMANDS
                                     Custom command to install chef-client
        --bootstrap-no-proxy [NO_PROXY_URL|NO_PROXY_IP]
                                     Do not proxy locations for the node being
bootstrapped; this option is used interna
lly by Opscode
        --bootstrap-proxy PROXY_URL  The proxy server for the node being
bootstrapped
    -t TEMPLATE,                     Bootstrap Chef using a built-in or custom
template. Set to the full path of an erb
template or use one of the built-in templates.
```

GL: Bootstrap Your Windows Node



```
> knife bootstrap <IP_NODE1> -x centos --sudo -i <aws.pem path> -N node1
```

```
Connecting to 34.195.38.226
Creating new client for iis_web
Creating new node for iis_web
Bootstrap
 [34.195.                          ting directory "C:\che
…
[34.195.38.226] C:\Users\Administrator\Documents>chef-client -c c:/chef/client.rb -j
c:/chef/first-boot.json
 [34.195.38.226] +-----------------------------------
 [34.195.38.226] âœ” 2 product licenses accepted.
 [34.195.38.226] +-----------------------------------
 [34.195.38.226] Starting Chef Infra Client, version 15.1.36
 [[34.195.38.226] [2019-07-22T20:49:28+00:00] WARN: Node iis_web has an empty run
list.
…
 [34.195.38.226] Running handlers complete
 [34.195.38.226] Chef Infra Client finished, 0/0 resources updated in 30 seconds
```

The licenses were accepted because we ran this command from our laptops which already have accepted the licenses.

GL: Run 'knife node list' Again

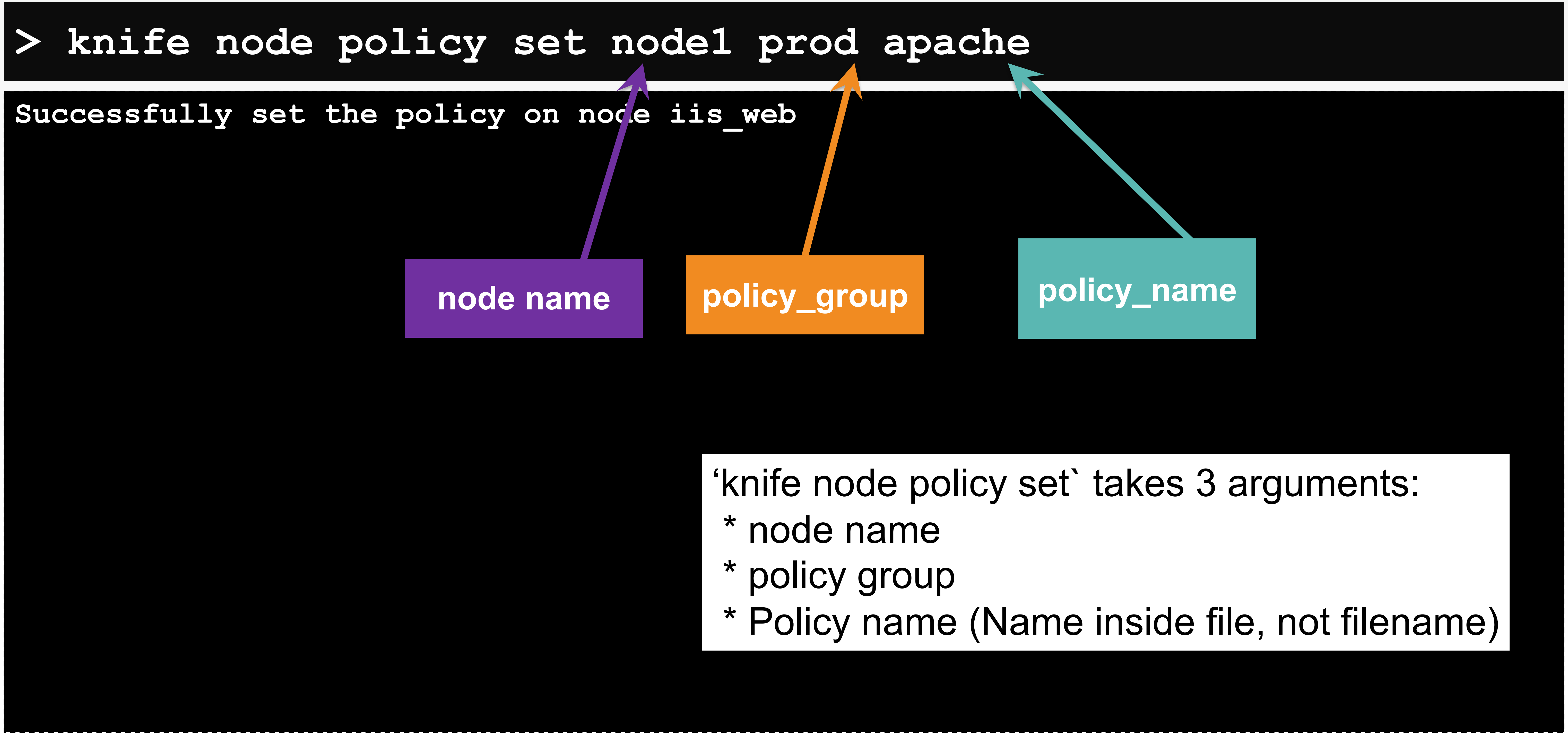
```
$ knife node list
```

```
node1
```

Now our Windows node is bootstrapped and instantiated in Chef Infra Server but it has no Policyfile yet. We'll add that in a moment.

Note: We could've combined `knife` commands to set the Policy while we bootstrapped the node, but we'll run these commands one at a time for clarity sake.

GL: Apply the Policyfile to Your Windows Node



```
> knife node policy set node1 prod apache
```

```
Successfully set the policy on node iis_web
```

‘knife node policy set` takes 3 arguments:

 * node name
 * policy group
 * Policy name (Name inside file, not filename)

GL: View More Information About Your Node

```
$ knife node show node1
```

```
Node Name:    node1
Policy Name:   apache
Policy Group: prod
FQDN:         ip-172-…
IP:           34.195.38.226
Run List:
Recipes:
Platform:     centos 7.8.2003
Tags:
```

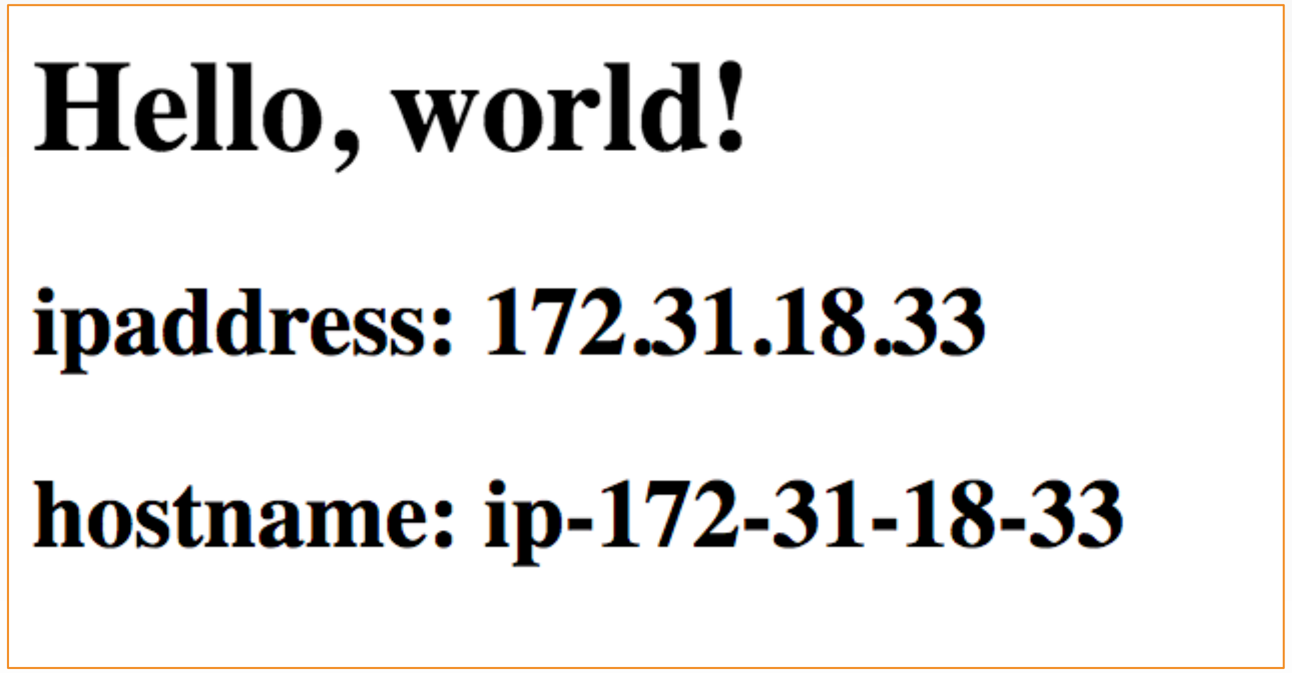
GL: Converge your Windows Node via winrm

```
$ knife ssh <node1_ip> -m -x centos -i <aws.pem path> "sudo chef-client"
```

```
34.195.38.226 Starting Chef Infra Client, version 15.1.36
34.195.38.226
34.195.38.226 Using policy 'apache' at revision
'49eef2f1f18967a5a3be025346c58f129b9eea5e2f3c914516ccc7a67deb2b80’
34.195.38.226 resolving cookbooks for run list: ["apache::default@0.2.1 (cd0db3e)"]
…
34.195.38.226  (up to date)
34.195.38.226 Running handlers:
34.195.38.226 Running handlers complete
34.195.38.226 Chef Infra Client finished, 2/4 resources updated in 46 seconds
34.195.38.226 [2019-07-22T21:53:44+00:00] ERROR: Failed to post reporting data to
server (HTTP 400), saving to c:/chef/cache/failed-reporting-data.json
```

The “Error” can be ignored. Our version of Hosted Chef still has Reporting installed, which has been deprecated in the latest versions of Chef Server.

GL: Verify that the New Node Serves the Page



Hello, world!

ipaddress: 172.31.18.33

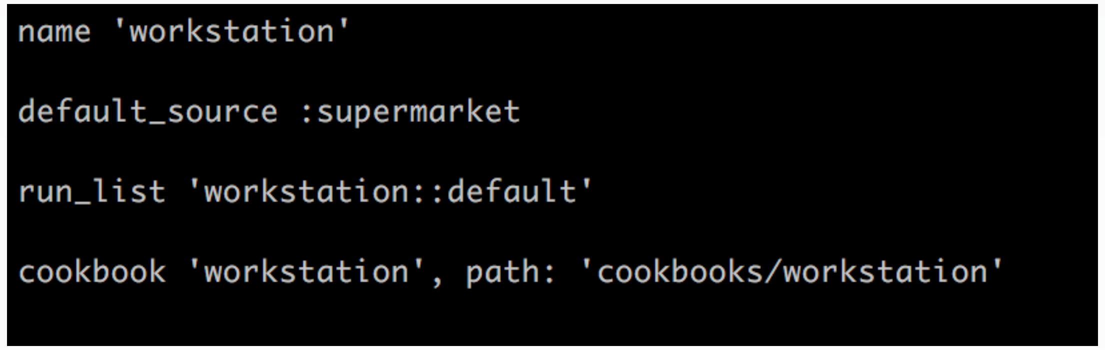
hostname: ip-172-31-18-33

Lab 11.1 <15 minutes>

Create workstation policyfile, install and upload to the chef infra server

Work directory is chef-repo

- Create policyfile for workstation - "chef generate policyfile workstation"
- Change the content of the policyfile
- Install the workstation policyfile - "chef install workstation.rb"
- check and verify workstation.lock.json
- check existing policies in the server - you should find nothing - "chef show-policy"
- upload workstation policy to chef server with prod policy_group - "chef push prod workstation.lock.json"
- check policies in the server again - you should find workstation



```
name 'workstation'

default_source :supermarket

run_list 'workstation::default'

cookbook 'workstation', path: 'cookbooks/workstation'
```

Lab 11.2 <30 minutes>

Create apache policyfile, install and upload to the chef infra server, bootstrap a node, apply apache policy

Work directory is chef-repo

- Create policyfile for apache - "chef generate policyfile apache"
- Change the content of the policyfile – same as workstation
- Install the apache policyfile - "chef install apache.rb"
- check and verify apache.lock.json
- upload apache policy to chef server with prod policy_group - "chef push prod apache.lock.json"
- check policies in the server - you should find workstation, apache - "chef show-policy"

Bootstrap Node

- Check nodes in the server – "knife node list"
- Bootstrap first node – "knife bootstrap <IP_first> -x centos --sudo -i aws.pem -N node1"
- Check nodes in server again
- Check node details – "knife node show node1"
- Set node1 policy to apache – "knife node policy set node1 prod apache"
- Check node details again
- Apply runlist – knife ssh <ip_first> -m -x centos -i <aws.pem path> "sudo chef-client"
- Test – in browser of through terminal ; curl <IP_first>

Review Questions

1. What is the benefit of storing cookbooks in a central repository?

2. What is the primary tool for communicating with the Chef Infra Server?

3. How did you add a node to your organization?

Q&A

What questions can you help you answer?

- Chef Infra Server
- Managed Chef
- Berkshelf
- Bootstrapping Nodes

CHEF™