

# Chef: Scripts to Recipes

Installing Tomcat 8 on CentOS 6.7

# Presenter

*Morgan Drake, Chef Solutions Architect*

Linux engineer since the lower Devonian

Recovering geologist



# You

*New to Chef*

Smart, intelligent, and funny

Understand what Chef does but not how it does it

Understand some Linux system administration and application deployment



# Schedule

---

- NOW** – Housekeeping
- @ 3 min** – Slides & Exercises
- @ 75 min** – Questions
- @ 80 min** – Resources and Wrap-up

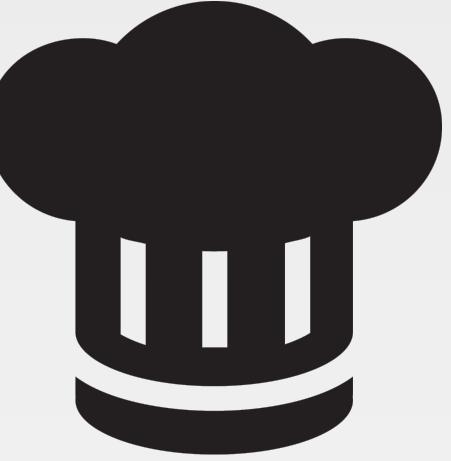
# Agenda

---

- ✓ Introductions
- Setup Workstation
- Convert Script to a Recipe
- Questions and Wrap-Up



# EXERCISE



## Pre-built Workstation

*We will provide for you a workstation with Chef development tools installed, including the ChefDK, Docker, and the kitchen-docker gem.*

### Objective:

- Login to the Remote Workstation

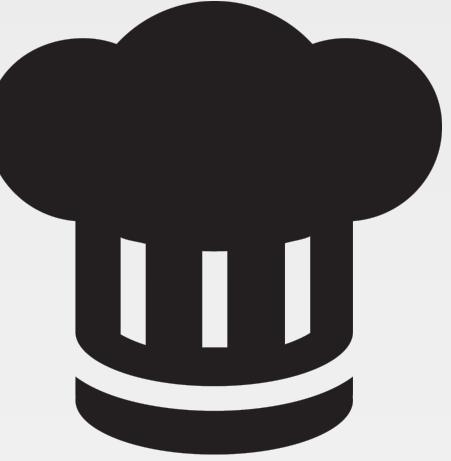
# Logging in to the Workstation



```
> ssh chef@IPADDRESS
```

```
The authenticity of host '54.209.164.144 (54.209.164.144)' can't  
be established. RSA key fingerprint is  
SHA256:tKoTsPbn6ER9BLThZqntXTxIYem3zV/iTQWvhLrBIBQ. Are you sure  
you want to continue connecting (yes/no)? yes  
chef@54.209.164.144's password: PASSWORD  
chef@ip-172-31-15-97 ~]$
```

# EXERCISE



## Pre-built Workstation

*We will provide for you a workstation with all the tools installed.*

### Objective:

- ✓ Login to the Remote Workstation

# Agenda

---

- ✓ Introductions
- ✓ Setup Workstation
- Convert Script to a Recipe
- Questions and Wrap-Up



# MOTIVATION

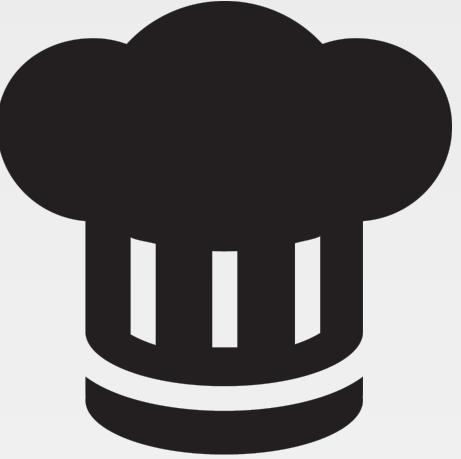


## Installing Tomcat

As an exercise you have been tasked with installing Tomcat 8 on a CentOS 6.7 node. We have provided you the machine and have already installed the tools.

Together we are going to walk through the process of writing Chef code and then applying it to the system to explore the various tools and important concepts within Chef.

# EXERCISE



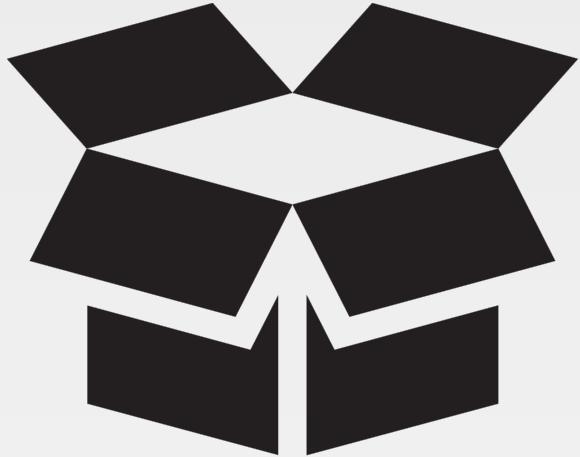
## Learning How to Chef

*We've got this script that works in a file or on our Wiki. Could I automate the creation with Chef?*

### Objective:

- Create a cookbook
- Write tests
- Write the first resource
- Apply the cookbook to the system

# CONCEPT



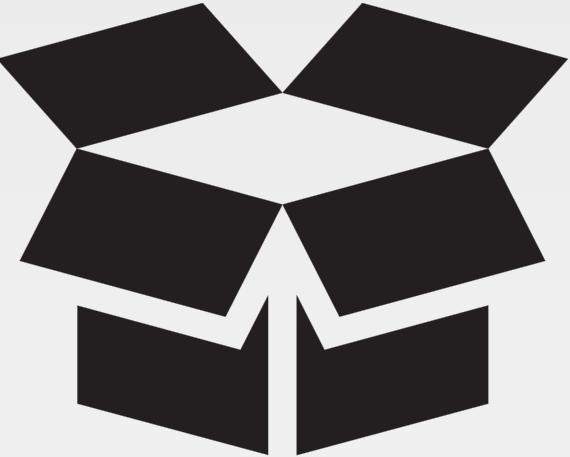
## Cookbook

A Chef cookbook is the fundamental unit of configuration and policy distribution.

Each cookbook defines a scenario, such as everything needed to install and configure MySQL, and then it contains all of the components that are required to support that scenario.

[docs.chef.io/cookbooks.html](https://docs.chef.io/cookbooks.html)

# CONCEPT



## chef

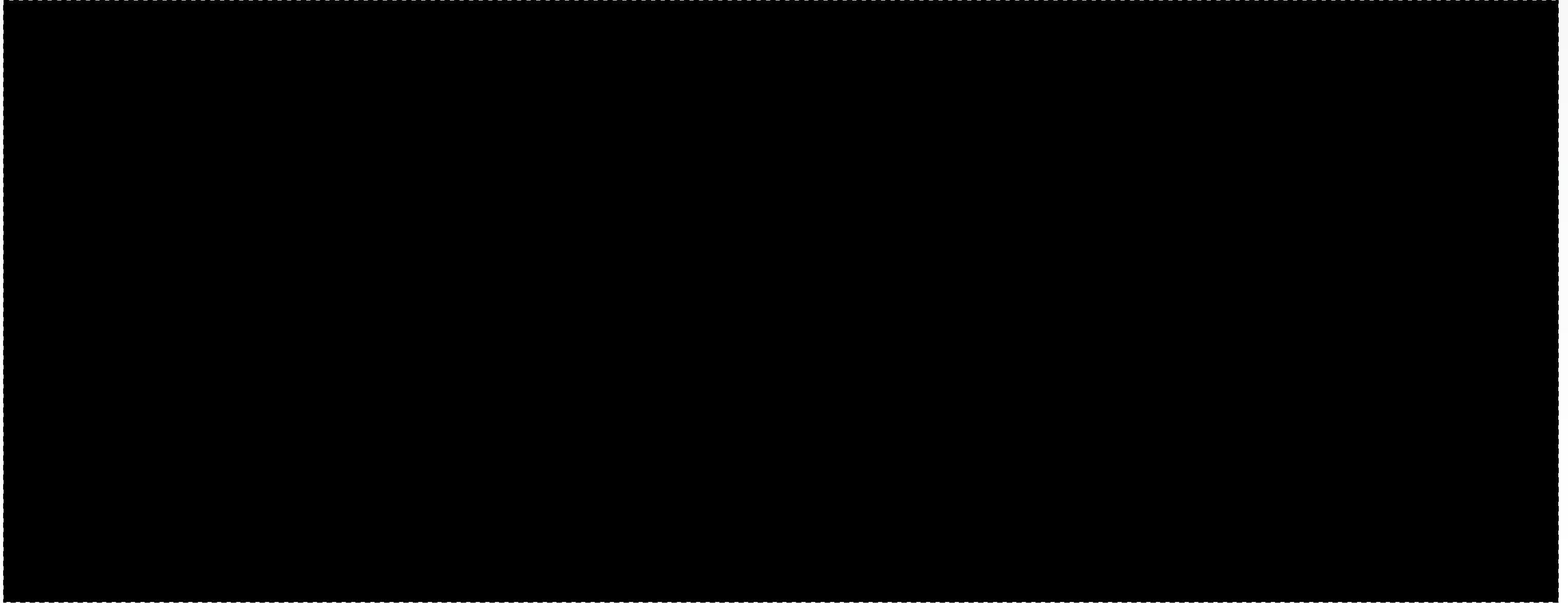
An executable program that allows you generate cookbooks and cookbook components.

[docs.chef.io/ctl\\_chef.html](https://docs.chef.io/ctl_chef.html)

# Creating a Cookbooks Directory



```
> mkdir cookbooks
```



# Creating a tomcat cookbook in cookbooks



```
> chef generate cookbook cookbooks/tomcat
```

```
Compiling Cookbooks...
```

```
Recipe: code_generator::cookbook
```

```
* directory[/home/chef/cookbooks/tomcat] action create
  - create new directory /home/chef/cookbooks/tomcat
* template[/home/chef/cookbooks/tomcat/metadata.rb] action
create_if_missing
  - create new file /home/chef/cookbooks/tomcat/metadata.rb
  - update content in file
/home/chef/cookbooks/tomcat/metadata.rb from none to d6d456
  (diff output suppressed by config)
* template[/home/chef/cookbooks/tomcat/README.md] action from
```

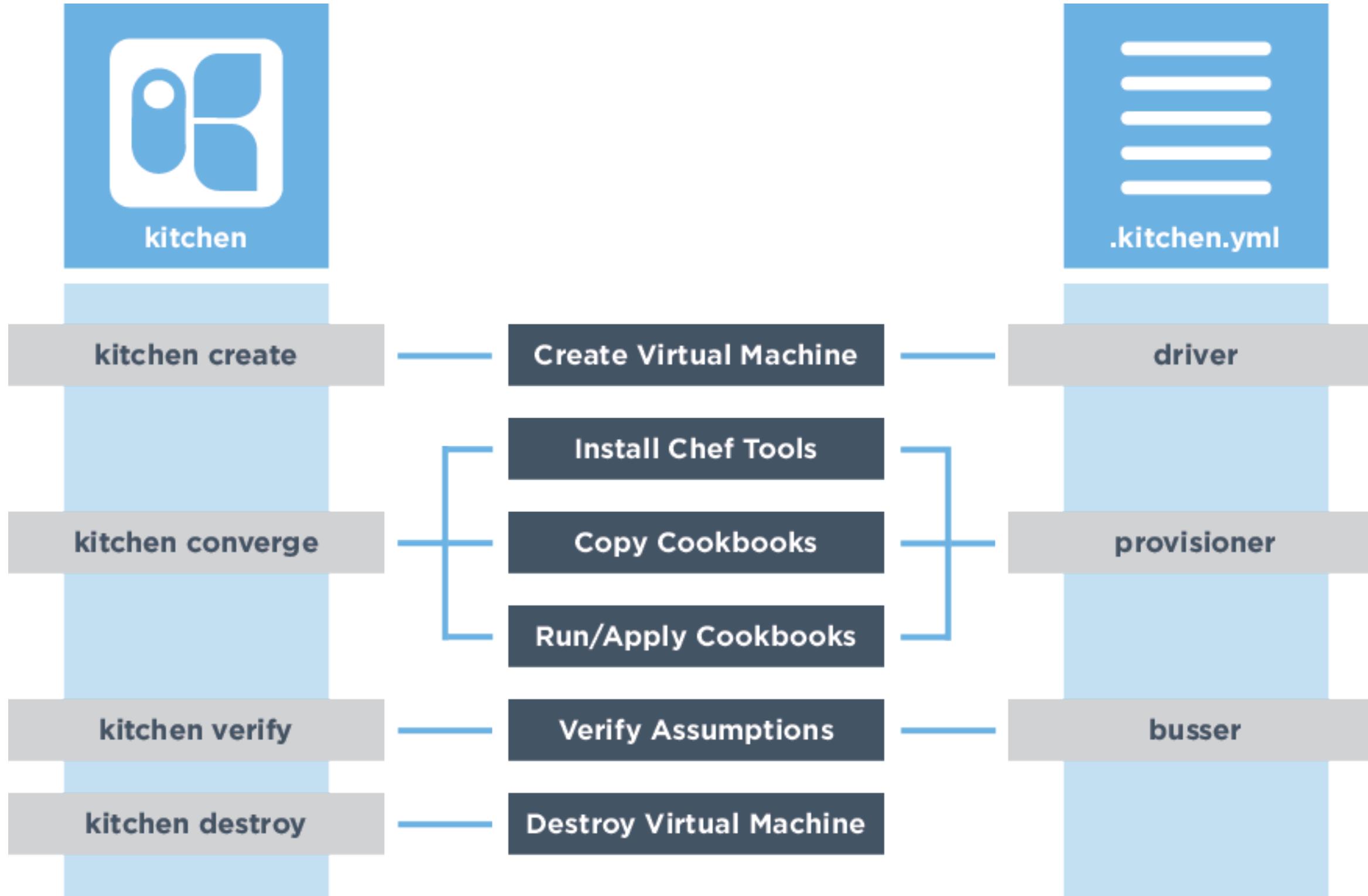
# Configuring Test Kitchen



```
> ls -alh cookbooks/tomcat
```

```
-rw-rw-r-- 1 chef chef    77 Sep 13 19:31 Berksfile
-rw-rw-r-- 1 chef chef 1.2K Sep 13 19:31 chefignore
drwxrwxr-x 3 chef chef 4.0K Sep 13 19:31 .delivery
drwxrwxr-x 7 chef chef 4.0K Sep 13 19:31 .git
-rw-rw-r-- 1 chef chef   185 Sep 13 19:31 .gitignore
-rw-rw-r-- 1 chef chef  483 Sep 13 19:31 .kitchen.yml
-rw-rw-r-- 1 chef chef  752 Sep 13 19:31 metadata.rb
-rw-rw-r-- 1 chef chef    54 Sep 13 19:31 README.md
drwxrwxr-x 2 chef chef 4.0K Sep 13 19:31 recipes
drwxrwxr-x 3 chef chef 4.0K Sep 13 19:31 spec
drwxrwxr-x 3 chef chef 4.0K Sep 13 19:31 test
```

# Test Kitchen Commands and Configuration



# Change into cookbooks/tomcat directory



```
> cd cookbooks/tomcat
```

# Edit .kitchen.yml

~/cookbooks/tomcat/.kitchen.yml

```
---
```

```
driver:
```

```
  name: docker
```

```
provisioner:
```

```
  name: chef_zero
```

```
verifier:
```

```
  name: inspec
```

```
platforms:
```

```
  - name: centos-6.7
```

```
  . . .
```

# Verify Testkitchen

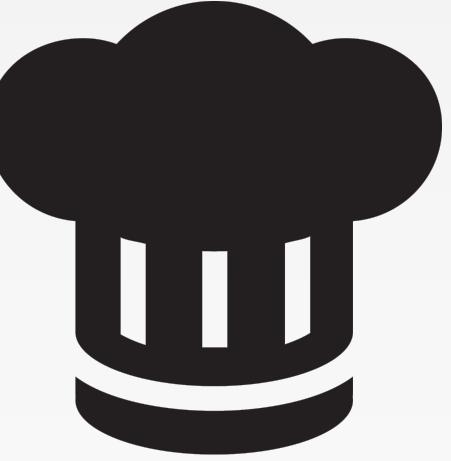


```
> cd cookbooks/tomcat
```

```
> kitchen list
```

Instance	Driver	Provisioner	Verifier	Transport	Last Action
default-centos-67	Docker	ChefZero	Inspec	Ssh	<Not Created>

# EXERCISE



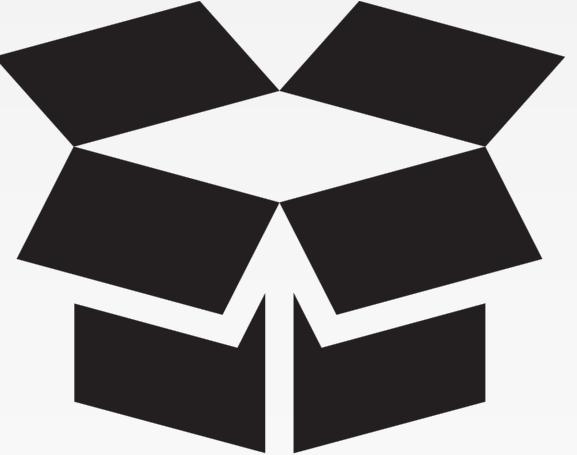
## Test Driven Development

*Before we write any Chef code, let's write a test to define success*

### Objective:

- Write a test for Tomcat
- Show the test fail with TestKitchen

# CONCEPT



## Test Driven Development

1. Define a test set for the unit first
2. Then implement the unit
3. Finally verify that the implementation of the unit makes the tests succeed.
4. Refactor

# Verifying that Tomcat is Running

```
$ curl http://localhost:8080
```

When the system is configured Tomcat should be running and available on the machine.

# Let's write a test



```
> $EDITOR ~/cookbooks/tomcat/test/smoke/default/default_test.rb
```

# Writing a test for tomcat

```
~/cookbooks/tomcat/test/smoke/default/default_test.rb
```

```
unless os.windows?

describe user('root'), :skip do
  :skip    it { should exist }
  end
end

describe port(8080) do
  it { should be_listening }
end
```

# Converge TestKitchen



```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)----->
Verifying <default-centos-67>...
Use `~/home/chef/cookbooks/tomcat/test/recipes/default` for testing

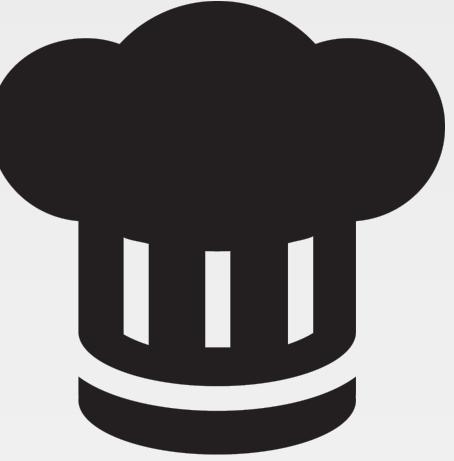
Target: ssh://kitchen@localhost:32770

User root
  ○

Port 8080
  ○ should be listening
    expected `Port 8080.listening?` to return true, got false

Test Summary: 0 successful, 1 failures, 1 skipped
```

# EXERCISE



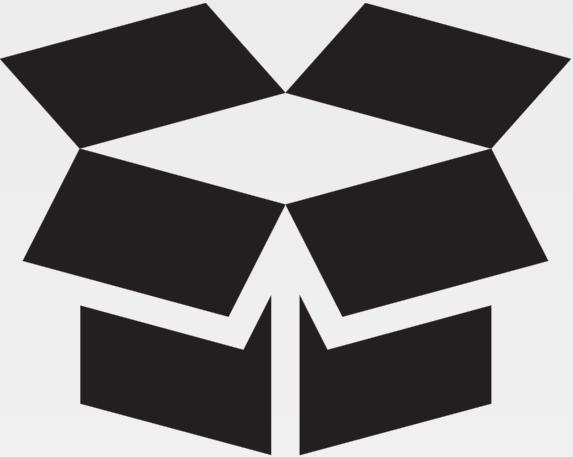
## Learning How to Chef

*How do I convert a command within the script into a resource within Chef?  
What is a resource?*

### Objective:

- Create a cookbook
- Write tests
- Write the first resource
- Apply the cookbook to the system

# CONCEPT



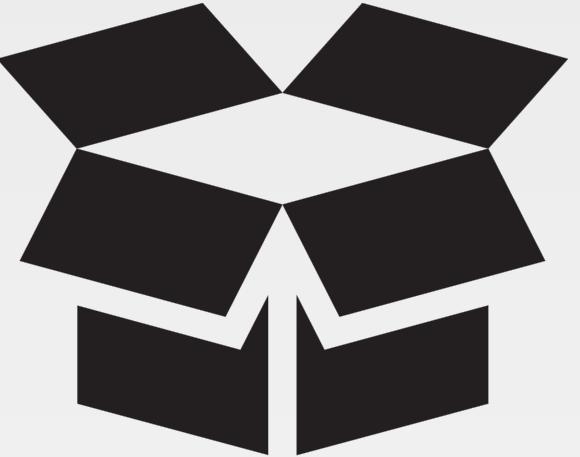
## Resource

A resource is a statement of configuration policy.

It describes the desired state of an element of your infrastructure and the steps needed to bring that item to the desired state.

[docs.chef.io/resources.html](https://docs.chef.io/resources.html)

# CONCEPT



## Resource: Package

Use the **package** resource to manage packages.

This resource is the base resource for several other resources used for package management on specific platforms. While it is possible to use each of these specific resources, it is recommended to use the **package** resource as often as possible. (e.g. `apt_package`, `yum_package`, `windows_package`)

[docs.chef.io/resource\\_package.html](https://docs.chef.io/resource_package.html)

# The Package Resource

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

The package named 'httpd' is installed.

# Script

```
$ sudo yum install java-1.7.0-openjdk-devel
```

# Resource

```
package 'java-1.7.0-openjdk-devel' do
  action :install
end
```

# Opening the Default Recipe



```
> $EDITOR ~/cookbooks/tomcat/recipes/default.rb
```

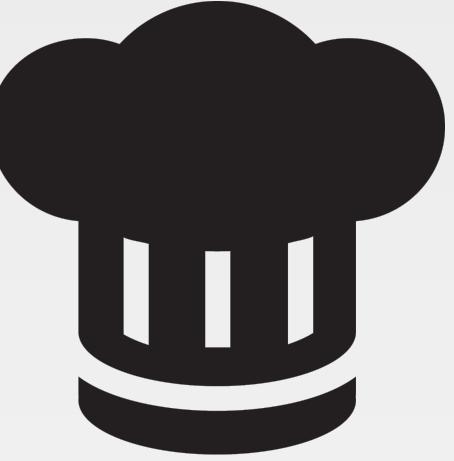
# Adding the Package Resource to the Recipe



~/cookbooks/tomcat/recipes/default.rb

```
#  
# Cookbook Name:: tomcat  
# Recipe:: default  
  
#  
# Copyright (c) 2016 The Authors, All Rights Reserved.  
package 'java-1.7.0-openjdk-devel' do  
  action :install  
end
```

# EXERCISE



## Learning How to Chef

*Now that I have written a resource into a recipe in a cookbook - how do I apply that change to the my system?*

### Objective:

- ✓ Create a cookbook
- ✓ Write tests
- ✓ Write the first resource
- Apply the cookbook to the system

# Converge TestKitchen



```
> kitchen converge
```

(lots of output!)

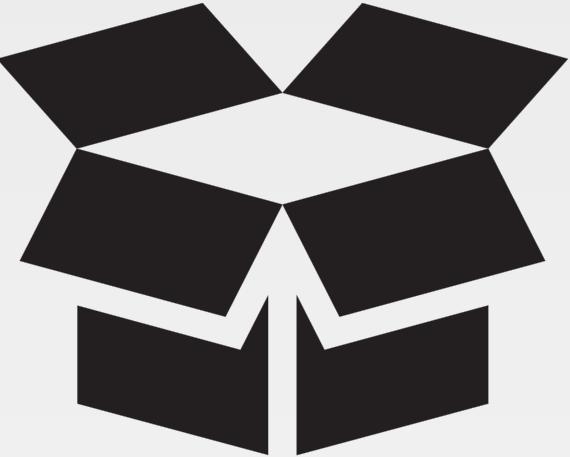
# Verify TestKitchen status



```
> kitchen list
```

Instance	Driver	Provisioner	Verifier	Transport	Last Action
default-centos-67	Docker	ChefZero	Inspec	Ssh	Converged

# CONCEPT



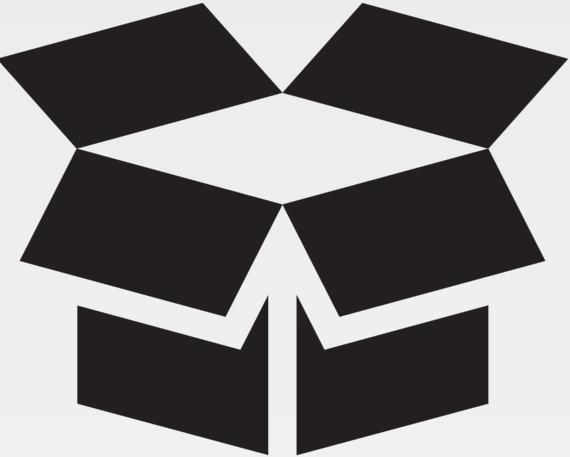
## chef-client

A chef-client is an agent that runs locally on every node that is under management by Chef.

When a chef-client is run, it will perform all of the steps that are required to bring the node into the expected state.

[docs.chef.io/chef\\_client.html](https://docs.chef.io/chef_client.html)

# CONCEPT



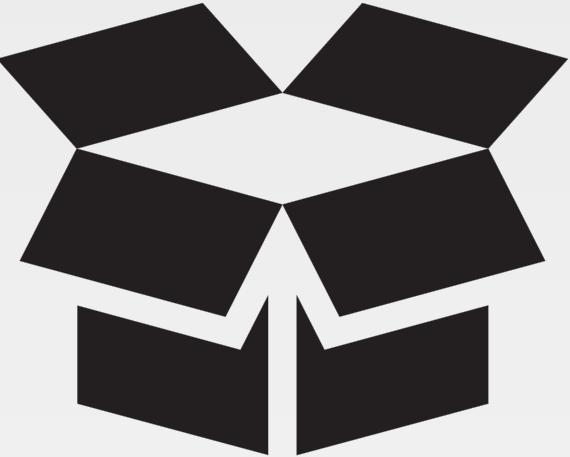
## **chef-client --local-mode (or -z)**

chef-client's default mode attempts to contact a Chef Server and ask it for the recipes to run for the given node.

We are overriding that behavior to have it work in a local mode.

[docs.chef.io/chef\\_client.html](https://docs.chef.io/chef_client.html)

# CONCEPT



**chef-client --runlist "recipe[COOKBOOK::RECIPE]"**

In local mode, we need to provide a list of recipes to apply to the system. This is called a **run list**. A run list is an ordered collection of recipes to execute.

Each recipe in the run list must be addressed with the format **recipe[COOKBOOK NAME::RECIPE NAME]**.

[docs.chef.io/chef\\_client.html](https://docs.chef.io/chef_client.html)

# Applying tomcat Cookbook's default Recipe



```
> sudo chef-client --local-mode --runlist "recipe[tomcat::default]"
```

```
- tomcat (0.1.0)
```

```
Compiling Cookbooks...
```

```
Converging 1 resources
```

```
Recipe: tomcat::default
```

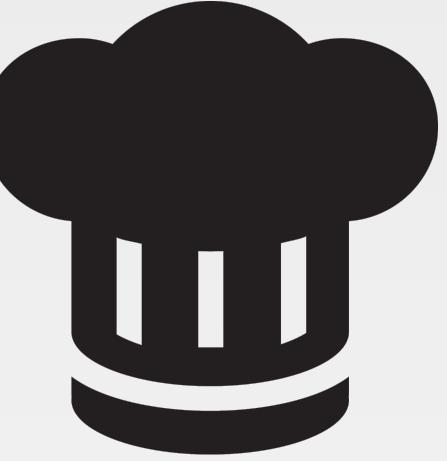
```
* yum_package[java-1.7.0-openjdk-devel] action install
  - install version 1.7.0.101-2.6.6.1.el6_7 of package java-1.7.0-openjdk-devel
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Client finished, 1/1 resources updated in 48 seconds
```

# EXERCISE



## Learning How to Chef

*Now I see the workflow and understand Chef a little better. What's next?*

### Objective:

- ✓ Create a cookbook
- ✓ Write tests
- ✓ Write the first resource
- ✓ Apply the cookbook to the system

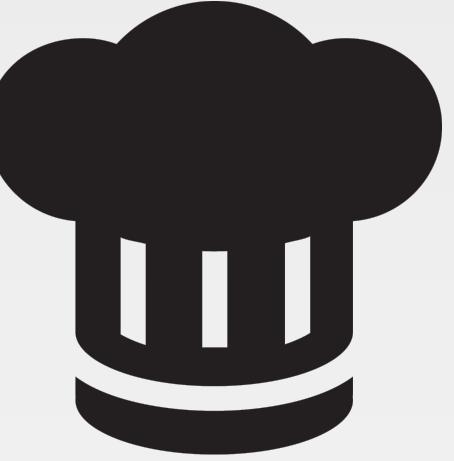
# DISCUSSION



## Q&A

What questions can we answer for you?

# EXERCISE



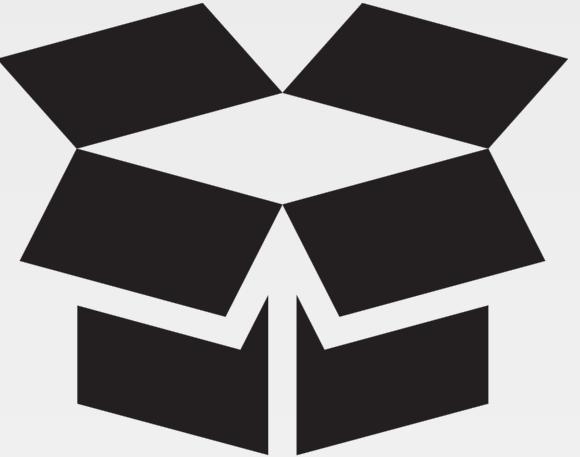
## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

- Create required users
- Download and extract Tomcat
- Set the correct directory permissions
- Setup the Tomcat service
- Verify the site is running

# CONCEPT



## Resource: group

Use the **group** resource to manage a local group.

[docs.chef.io/resource\\_group.html](https://docs.chef.io/resource_group.html)

# The group Resource

```
group 'beverages' do
  action :create
end
```

The group named 'beverages' is created with members 'latte' and 'mocha'.

# Script

```
$ sudo groupadd chef
```

# Resource

```
group 'chef' do
  action :create
end
```

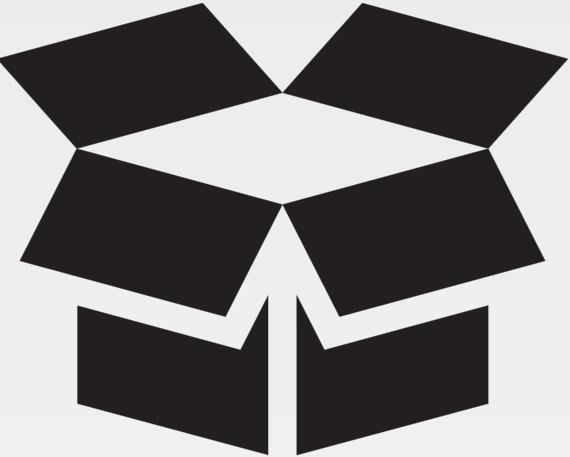
# Adding the group Resource

~/cookbooks/tomcat/recipes/default.rb

```
package 'java-1.7.0-openjdk-devel' do
  action :install
end

group 'tomcat' do
  action :create
end
```

# CONCEPT



## Resource: user

Use the **user** resource to add users, update existing users, remove users, and to lock/unlock user passwords.

[docs.chef.io/resource\\_user.html](https://docs.chef.io/resource_user.html)

# The user Resource

```
user 'latte' do
  action :create
  group 'beverages'
  shell '/bin/sh'
  home '/home/latte'
end
```

The user named 'latte' is created in the group 'beverages' with the shell '/bin/sh' and the home '/home/latte'.

# Script

```
$ sudo useradd -g chef chef
```

# Resource

```
user 'chef' do
  group 'chef'
end
```

# Adding the user Resource

~/cookbooks/tomcat/recipes/default.rb

```
group 'tomcat' do
  action :create
end
```

```
user 'tomcat' do
  group 'tomcat'
  action :create
end
```

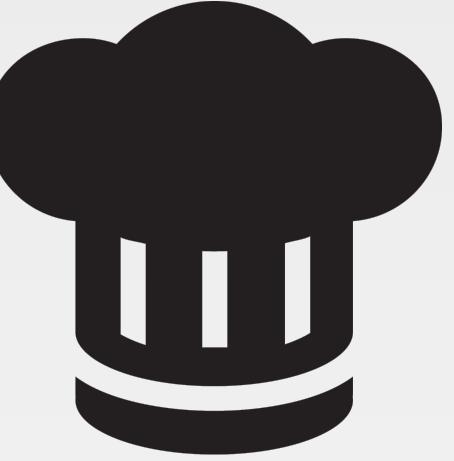
# Converge TestKitchen



```
> kitchen converge
```

(lots of output!)

# EXERCISE



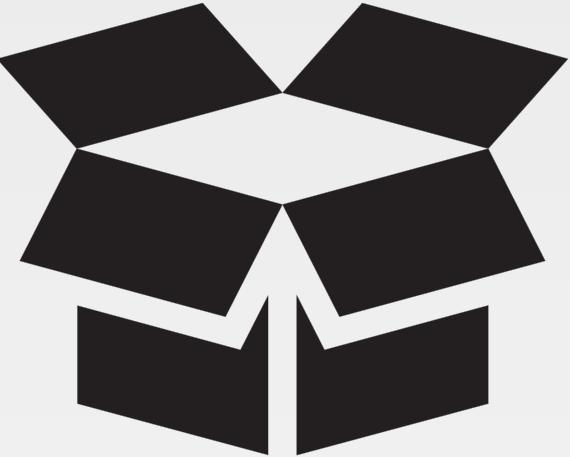
## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

- Create require users
- Download and extract Tomcat
- Set the correct directory permissions
- Setup the Tomcat service
- Verify the site is running

# CONCEPT



## Resource: `remote_file`

Use the `remote_file` resource to transfer a file from a remote location using file specificity. This resource is similar to the `file` resource.

[docs.chef.io/resource\\_remote\\_file.html](https://docs.chef.io/resource_remote_file.html)

# The remote\_file Resource

```
remote_file '/tmp/app-archive.tar.gz' do
  source 'http://project.com/application/latest/app-archive.tar.gz'
  action :create
end
```

The remote\_file named '/tmp/app-archive.archive.tar.gz' is created from source 'http://project.com/application/latest/app-archive.tar.gz'.

# Script

```
$ cd /tmp  
$ wget  
http://mirror.sdunix.com/apache/tomcat/  
tomcat-8/v8.0.33/bin/apache-tomcat-  
8.0.33.tar.gz
```

# Resource

```
remote_file '/tmp/apache-tomcat-  
8.0.33.tar.gz' do  
  source 'http://mirror.sdunix.com...'  
  action :create  
end
```

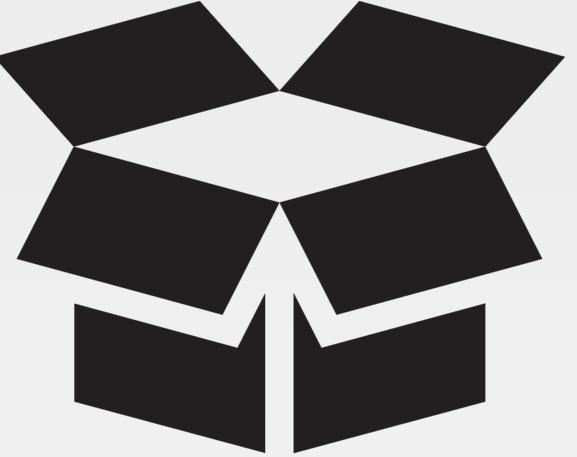
# Adding the remote\_file Resource

~/cookbooks/tomcat/recipes/default.rb

```
user 'tomcat' do
  group 'tomcat'
end

remote_file '/tmp/tomcat.tar.gz' do
  source 'http://chef.run/2wWavdE'
  action :create
end
```

# CONCEPT



## Resource: directory

Use the directory resource to manage a directory, which is a hierarchy of folders that comprises all of the information stored on a computer. The root directory is the top-level, under which the rest of the directory is organized.

The directory resource uses the name property to specify the path to a location in a directory. Typically, permission to access that location in the directory is required.

[docs.chef.io/resource\\_directory.html](https://docs.chef.io/resource_directory.html)

# The directory Resource

```
directory '/opt/application/conf' do
  action :create
  recursive true
end
```

The directory named '/opt/application/conf' is created recursively.

# Script

```
$ sudo mkdir /opt/tomcat
```

# Resource

```
directory '/opt/tomcat' do
  action :create
  recursive true
end
```

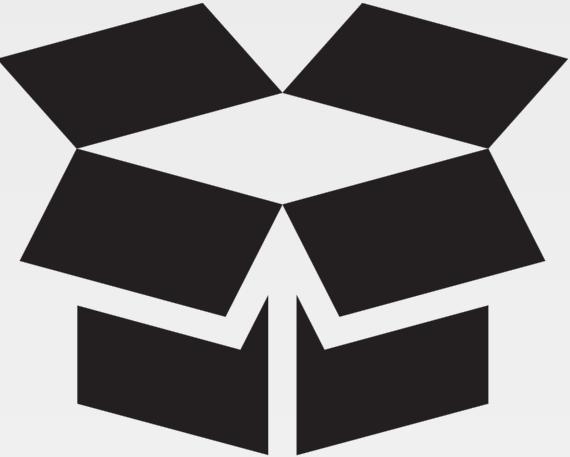
# Adding the directory Resource

~/cookbooks/tomcat/recipes/default.rb

```
remote_file '/tmp/tomcat.tar.gz' do
  source 'http://chef.run/2wWavdE'
  action :create
end
```

```
directory '/opt/tomcat' do
  action :create
  recursive true
end
```

# CONCEPT



## Resource: execute

Use the execute resource to execute a single command. Commands that are executed with this resource are (by their nature) not idempotent, as they are typically unique to the environment in which they are run. Use `not_if` and `only_if` to guard this resource for idempotence.

[docs.chef.io/resource\\_execute.html](https://docs.chef.io/resource_execute.html)

# The execute Resource

```
execute 'extract_archive' do
  command 'tar xvf archive.tar.gz'
  cwd '/tmp'
end
```

The execute resource named 'extract\_archive' is run with the command 'tar xvf archive.tar.gz' in the cwd '/tmp'.

All resources have default actions. When an action is not provided the default action is assumed. **For the execute resource the default action is run.**

# Script

```
$ sudo tar xvf apache-tomcat-8*tar.gz -C /opt/tomcat --strip-components=1
```

# Resource

```
execute 'extract_tomcat' do
  command 'tar xvf apache-tomcat-8...'
  cwd '/tmp'
end
```

# Adding the execute Resource

```
~/cookbooks/tomcat/recipes/default.rb
```

```
directory '/opt/tomcat' do
  action :create
  recursive true
end
```

```
execute 'extract_tomcat' do
  command 'tar xvf tomcat.tar.gz -C /opt/tomcat --strip-
components=1'
  cwd '/tmp'
end
```

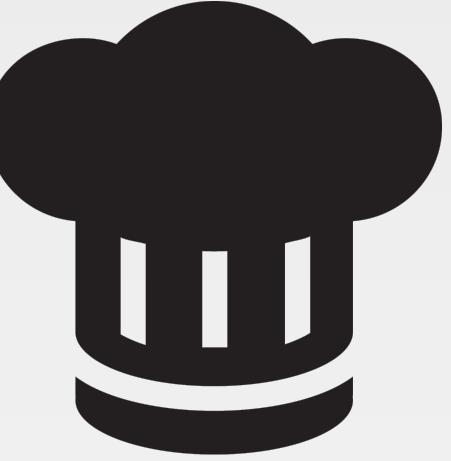
# Converge TestKitchen



```
> kitchen converge
```

(lots of output!)

# EXERCISE



## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

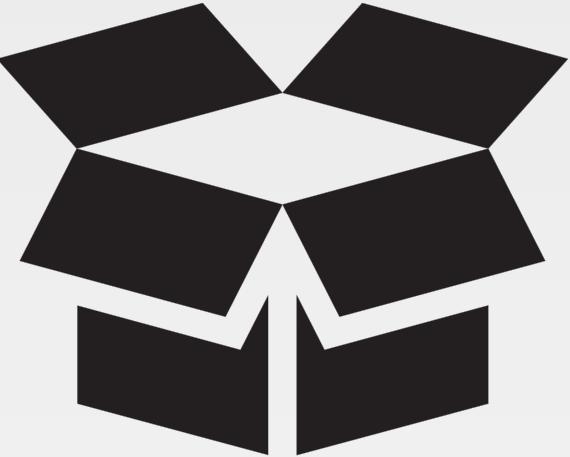
- ✓ Create require users
- ✓ Download and extract Tomcat
- Set the correct directory permissions
- Setup the Tomcat service
- Verify the site is running

# Set the Correct Directory Permissions

```
$ cd /opt/tomcat  
$ sudo chgrp -R tomcat conf  
$ sudo chmod g+rwx conf  
$ sudo chmod g+r conf/*  
$ sudo chown -R tomcat webapps/ work/ temp/ logs/ conf/
```

The Tomcat directory needs to be setup with the correct permissions and owners for the application to work correctly.

# CONCEPT



## Resource: directory

A directory may have an owner, group, mode (rights on Windows) specified as attributes.

The default action for the directory resource is create.

[docs.chef.io/resource\\_directory.html](https://docs.chef.io/resource_directory.html)

# Script

```
$ cd /opt/tomcat  
$ sudo chgrp -R tomcat conf  
$ sudo chmod g+rwx conf
```

# Resource

```
execute 'chgrp -R tomcat  
/opt/tomcat/conf'  
  
directory '/opt/tomcat/conf' do  
  group 'tomcat'  
  mode '0474'  
end
```

# Adding the execute Resources

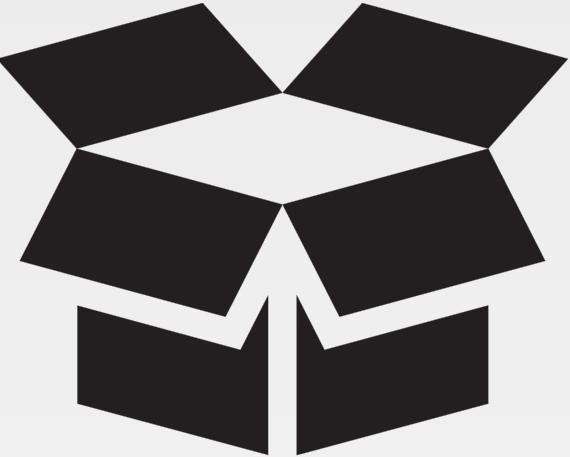
~/cookbooks/tomcat/recipes/default.rb

```
execute 'extract_tomcat' do
  command 'tar xvf tomcat.tar.gz -C /opt/tomcat --strip-components=1'
  cwd '/tmp'
end
```

```
execute 'chgrp -R tomcat /opt/tomcat/conf'

directory '/opt/tomcat/conf' do
  group 'tomcat'
  mode '0474'
  action :create
end
```

# CONCEPT



## Resource: execute

The name of the execute resource can also be the command. This is often times the choice with shorter or clearer commands.

[docs.chef.io/resource\\_execute.html](https://docs.chef.io/resource_execute.html)

# Script

```
$ cd /opt/tomcat  
$ sudo chmod g+r conf/*  
$ sudo chown -R tomcat webapps/ work/  
temp/ logs/
```

# Resource

```
execute 'chmod g+r conf/*' do  
  cwd '/opt/tomcat'  
end  
  
execute 'chown -R chef webapps/ work/  
temp/ logs/ conf/' do  
  cwd '/opt/tomcat'  
end
```

*NOTE: Even the current working directory property (cwd) is not needed if the command is provided the entire path to the location of target.*

```
execute 'chmod g+r /opt/tomcat/conf/*'
```

# Adding the execute Resources

~/cookbooks/tomcat/recipes/default.rb

```
directory '/opt/tomcat/conf' do
  group 'tomcat'
  mode '0474'
  action :create
end

execute 'chmod g+r conf/*' do
  cwd '/opt/tomcat'
end

execute 'chown -R tomcat webapps/ work/ temp/ logs/ conf/' do
  cwd '/opt/tomcat'
end
```

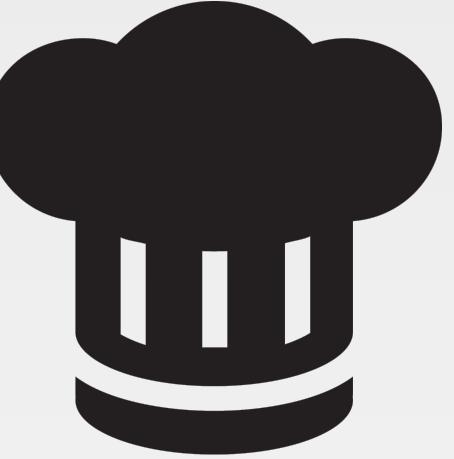
# Converge TestKitchen



```
> kitchen converge
```

(lots of output!)

# EXERCISE



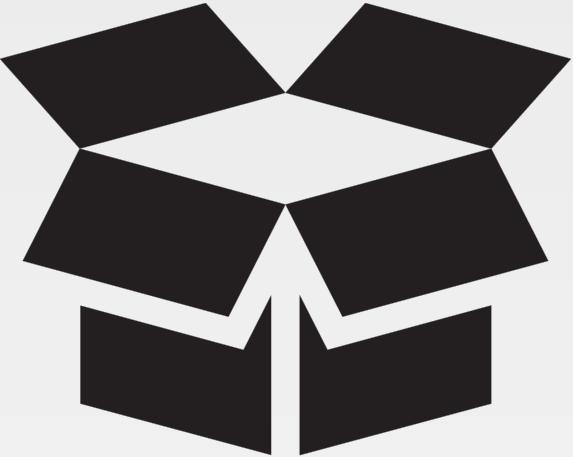
## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

- ✓ Create require users
- ✓ Download and extract Tomcat
- ✓ Set the correct directory permissions
- Setup the Tomcat service
- Verify the site is running

# CONCEPT



## Resource: template

A cookbook template is an Embedded Ruby (ERB) template that is used to dynamically generate static text files. Templates may contain Ruby expressions and statements, and are a great way to manage configuration files. Use the template resource to add cookbook templates to recipes; place the corresponding Embedded Ruby (ERB) template file in a cookbook's /templates directory.

[docs.chef.io/resource\\_template.html](https://docs.chef.io/resource_template.html)

# Script

```
$ sudo vi  
/etc/systemd/system/tomcat.service
```

# Resource

```
template  
'/etc/systemd/system/tomcat.service' do  
  source 'tomcat.service.erb'  
end
```

# Creating a Tomcat Service Template File



```
> chef generate template tomcat
```

Compiling Cookbooks...

```
Recipe: code_generator::template
  * directory[cookbooks/tomcat/templates/default] action create
    - create new directory cookbooks/tomcat/templates/default
  * template[cookbooks/tomcat/templates/default/tomcat8.erb]
action create
  - create new file
cookbooks/tomcat/templates/default/tomcat8.erb
  - update content in file
cookbooks/tomcat/templates/default/tomcat8.erb from none to e3b0c4
  (diff output suppressed by config)
```

# Editing the Tomcat Service Template File



```
> $EDITOR ~/cookbooks/tomcat/templates/tomcat.erb
```

# Adding the Contents to the Template File



~/cookbooks/tomcat/templates/tomcat.erb

<http://chef.run/2xgqiq4>

```
#!/bin/bash
# description: Tomcat Start Stop Restart
# processname: tomcat
# chkconfig: 234 20 80
JAVA_HOME=/usr/java/jdk1.7.0_60
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH
CATALINA_HOME=/usr/share/apache-tomcat-8.0.8

case $1 in
start)
sh $CATALINA_HOME/bin/startup.sh
```

## Vim Users:

Be sure to use `:set paste`  
before pasting text in  
your template.

# Adding the template Resource

~/cookbooks/tomcat/recipes/default.rb

```
execute 'chown -R tomcat webapps/ work/ temp/ logs/ conf/' do
  cwd '/opt/tomcat'
end
```

```
template '/etc/init.d/tomcat' do
  source 'tomcat.erb'
  mode '0755'
  action :create
end
```

# Script

```
$ sudo service tomcat8 start
```

# Resource

```
service 'tomcat8' do
  action [ :start ]
end
```

# Adding the service Resource

~/cookbooks/tomcat/recipes/default.rb

```
template '/etc/init.d/tomcat8' do
  source 'tomcat8.erb'
  mode '0755'
  action :create
end
```

```
service 'tomcat' do
  action :start
end
```

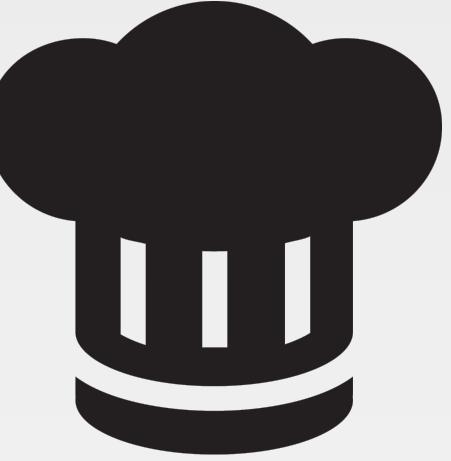
# Converge TestKitchen



```
> kitchen converge
```

(lots of output!)

# EXERCISE



## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

- ✓ Create require users
- ✓ Download and extract Tomcat
- ✓ Set the correct directory permissions
- ✓ Setup the Tomcat service
- Verify the site is running

# Converge TestKitchen



```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)----->
Verifying <default-centos-67>...
Use `/home/chef/cookbooks/tomcat/test/recipes/default` for testing

Target: ssh://kitchen@localhost:32770

  ○ User root should exist; User root This is an example test, r... (1
skipped)      This is an example test, replace with your own test.

  ✓ Port 8080 should be listening

Summary: 2 successful, 0 failures, 1 skipped
Finished verifying <default-centos-67> (0m0.75s).
-----> Kitchen is finished. (0m2.54s)
```

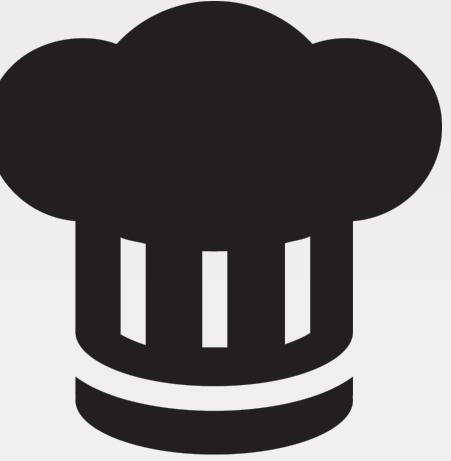
# Verifying that Tomcat is Running on port 8080



```
> curl http://localhost:8080
```

“kitchen login” will allow you to log in to your Test Kitchen instance.

# EXERCISE



## Build the Recipe

*Converting the first piece of the script showed the initial workflow. Now it's time to continue with the rest of the script.*

### Objective:

- ✓ Create require users
- ✓ Download and extract Tomcat
- ✓ Set the correct directory permissions
- ✓ Setup the Tomcat service
- ✓ Verify the site is running

# Agenda

---

- ✓ Introductions
- ✓ Setup Workstation
- ✓ Convert Script to a Recipe
- ❑ Questions and Wrap-Up



# DISCUSSION



## Q&A

What questions can we answer for you?

# MOTIVATION



## Learn Chef

This resource contains tutorials, videos, articles, documentation, and access to training that will increase your skills and knowledge of Chef.

[learn.chef.io](https://learn.chef.io)

[learn.chef.io/skills/joy-of-automating-episode-1](https://learn.chef.io/skills/joy-of-automating-episode-1)

# Agenda

---

- ✓ Introductions
- ✓ Setup Workstation
- ✓ Convert Script to a Recipe
- ✓ Questions and Wrap-Up





CHEF™

