Collaboration and Version Control
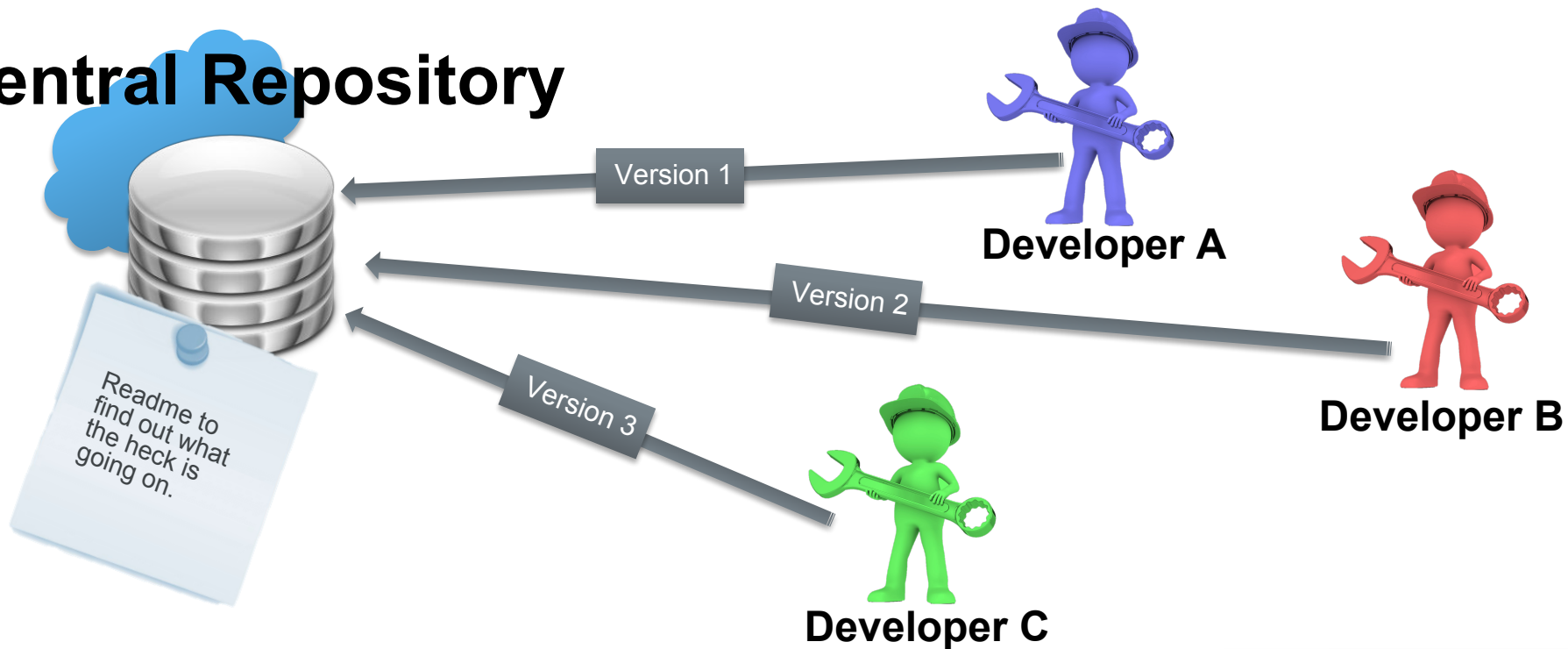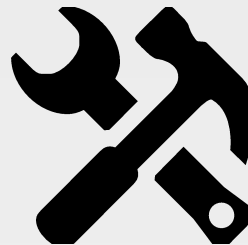
Git Version Control

git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows.

git is commonly used to track changes to your cookbooks.

Lab: Install git

❑ Add the additional policy to the "setup.rb":

The package named 'git' is installed.

❑ Then apply this recipe with chef-client.

Lab: Adding the git Package

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

package 'git' do                                    +
  action :install
end

file '/etc/motd' do
  content 'Property of ...'
end
```

Lab: Re-apply the Setup Recipe

```
$ sudo chef-client --local-mode setup.rb
```

```
Converging 4 resources
Recipe: @recipe_files::/home/chef/setup.rb
  * yum_package[tree] action install (up to date)
  * yum_package[git] action install
    - install version 1.7.1-3.el6_4.1 of package git
  * file[/etc/motd] action create (up to date)
```
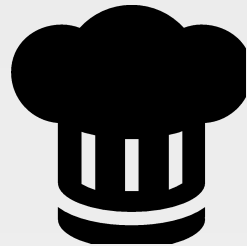
Lab: Install git

✓ Add the additional policy to the "setup.rb":

   The package named 'git' is installed.

✓ Then apply this recipe with chef-client.

Group Exercise: Version Control

This is a probably a good point to capture the initial state of our cookbook.

Objective:

- ✓ Use chef to generate a cookbook
- ✓ Move the setup recipe into the new cookbook
- ✓ Add the new cookbook to version control

CHEF

GL: Move into the Cookbook Directory

```
$ cd cookbooks/workstation
```

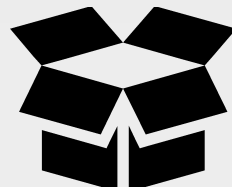# GL: Initialize the Directory as a git Repository

```
$ git init
```

```
Reinitialized existing Git repository in /home/chef/cookbooks/workstation/.git/
```

GL: Use 'git add' to Stage Files to be Committed

```
$ git add .
```

CONCEPT

Staging Area

The staging area has a file, generally contained in your Git directory, that stores information about what will go into your next commit.

It's sometimes referred to as the "index", but it's also common to refer to it as the staging area.

http://git-scm.com/book/en/v2/Getting-Started-Git-Basics

GL: Use 'git status' to View the Staged Files

```
$ git status

On branch master


Initial commit


Changes to be committed:
  (use "git rm --cached <file>..." to unstage)


    new file:   .gitignore
    new file:   .kitchen.yml
    new file:   Berksfile
    new file:   README.md
    new file:   chefignore
    new file:   metadata.rb
```

GL: Use 'git commit' to Save the Staged Changes

```
$ git commit -m "Initial commit"
master (root-commit) 9998472] Initial workstation cookbook
 Committer: ChefDK User <chef@ip-172-31-59-191.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com


After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author
```
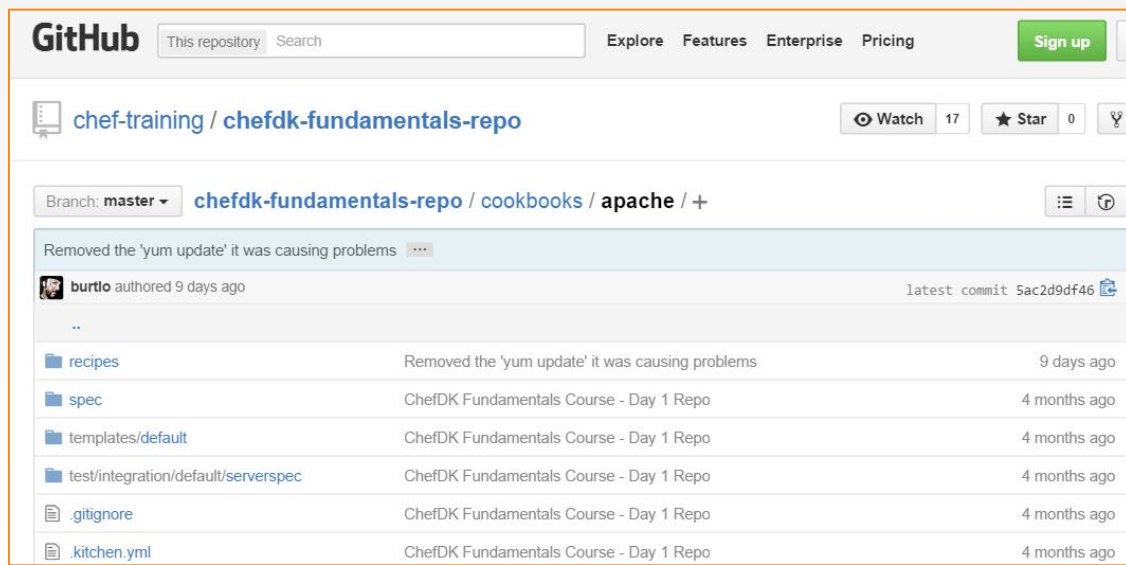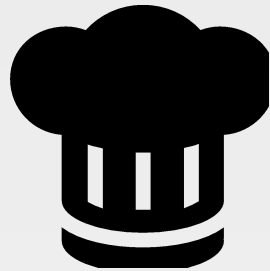
Git Version Control

If you use git versioning you should ultimately push the local git repository to a shared remote git repository.

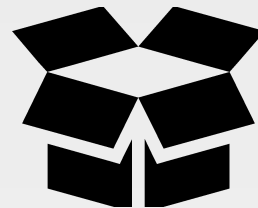In this way others could collaborate with you from a centralized location.

Changes Mean a New Version

Let's bump the version number and check in the code to source control.

Objective:

- ❑ Update the version of the "workstation" cookbook
- ❑ Commit the changes to the "workstation" cookbook to version control
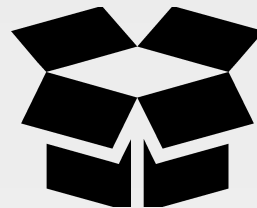
CHEF™

Cookbook Versions

A cookbook version represents a set of functionality that is different from the cookbook on which it is based.

https://docs.chef.io/cookbook_versions.html

CHEF

Semantic Versions

Given a version number **MAJOR**.**MINOR**.**PATCH**, increment the:

- **MAJOR** version when you make incompatible API changes
- **MINOR** version when you add functionality in a backwards-compatible manner
- **PATCH** version when you make backwards-compatible bug fixes

http://semver.org

Major, Minor, or Patch?

What kind of changes did you make to the cookbook?

GL: Update the Cookbook Version

~/cookbooks/workstation/metadata.rb

```
name               'workstation'
maintainer         'The Authors'
maintainer_email   'you@example.com'
license            'all_rights'
description        'Installs/Configures workstation'
long_description   'Installs/Configures workstation'
version            '0.2.0'
```

COMMIT

GL: Commit Your Work

```
$ cd ~/cookbooks/workstation
$ git add .
$ git status
$ git commit -m "Release version 0.2.0"
```

CHEF