Lab 1

Verify Local Environment

Allotted Time: 30 minutes

Overview

In this exercise each participant configures and tests a local development environment and registers for the Chef Community Slack.

Connect to Node (Everyone)

- ssh chef@NODE

If you don't have ssh available please downlowd an ssh client. For Windows, a good option is putty.

Introduction to Git

Customize your workstation (Everyone)

```
   git config --global user.name "YOUR NAME"
   git config --global user.email "YOUR EMAIL ADDRESS"
```

Example:

```
$ git config --global user.name "Jennifer Davis"
$ git config --global user.email "sparklydevops@gmail.com"
```

Verify .gitconfig creation (Everyone)

```
   cat ~/.gitconfig
```

Example:

```
   cat ~/.gitconfig
[user]
   name = Jennifer Davis
   email = sparklydevops@gmail.com
```

Set your preferred git editor (Everyone)

If you don't set your preferred editor, it will use the default text editor for the system.

- emacs
- nano
- vi/vim

```
git config --global core.editor EDITORNAME
```

Example:

```
$ git config --global core.editor nano
```

Add any additional desired git aliases (Optional)

Verify the configuration (Everyone)

```
git config --list
```

Example Output:

```
[sumac@ip-10-0-0-107 ~]$ git config --list
user.name=Jennifer Davis
user.email=sparklydevops@gmail.com
core.editor=vi
```

Create a project Directory (Everyone)

```
mkdir wd
cd wd
```

Create a GitHub identity (Everyone)

- If you don't already have a github account, create one.
- Browse to http://github.com. Supply a username, email address, and password.
- Free plan is fine. Other plans allow you to have private repositories.

(Optional) Setting up your GitHub keys (Everyone)

If you want to skip the added burden of entering your username and password each time at the prompt with git, you can follow the steps here to set up your ssh keys:

https://help.github.com/articles/generating-ssh-keys/

Example output of successful setup of keys:

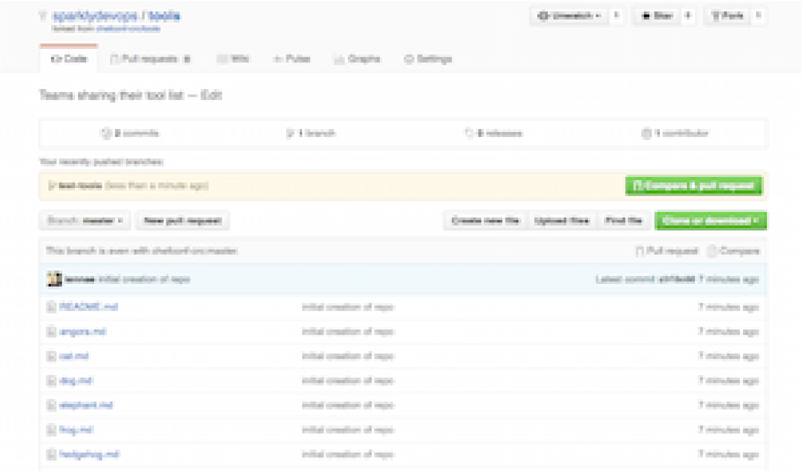
```
[sumac@ip-10-0-0-107 ~]$ ssh -T git@github.com
Hi sparklydevops! You've successfully authenticated, but GitHub does not provide shell access.
```

Share team tools (1 person from each team)

- Fork the tools repo https://github.com/chefconf-crc/tools
- Clone your fork to your local node.
- Create a branch teamname-tools.
- Update the teamname.md with your list of tools
- Create a pull request to have your team's tool list merged.

Example Output:

```
[sumac@ip-10-0-0-107 tools]$ git checkout -b test-tools
Switched to a new branch 'test-tools'
[sumac@ip-10-0-0-107 tools]$  vi test.md
[sumac@ip-10-0-0-107 tools]$ git add test.md
[sumac@ip-10-0-0-107 tools]$ git commit -m "Adding team test to tools"
[test-tools af2f678] Adding team test to tools
 1 file changed, 3 insertions(+)
 create mode 100644 test.md
[sumac@ip-10-0-0-107 tools]$ git push origin test-tools
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To git@github.com:sparklydevops/tools.git
 * [new branch]      test-tools -> test-tools
```



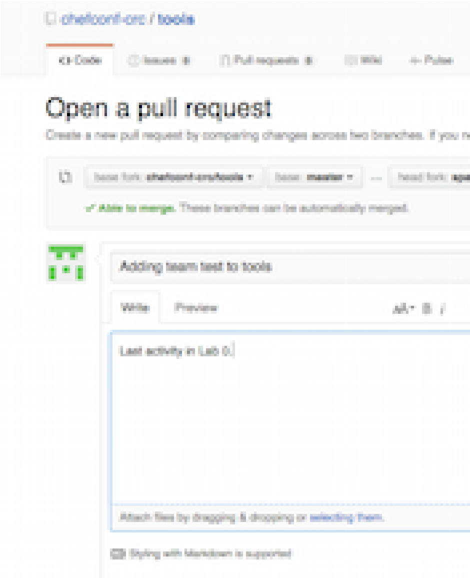
Teams sharing their tool list — Edit

test-tools (less than a minute ago)

This branch is even with chefconf-crc:master.

Latest commit 7 minutes ago

README.md	initial creation of repo	7 minutes ago
angora.md	initial creation of repo	7 minutes ago
cat.md	initial creation of repo	7 minutes ago
dog.md	initial creation of repo	7 minutes ago
elephant.md	initial creation of repo	7 minutes ago
frog.md	initial creation of repo	7 minutes ago
hedgehog.md	initial creation of repo	7 minutes ago



chefconf-crc / tools
Code
Issues
Pull requests
Wiki
Pulse
Write
Preview
Attach files by dragging & dropping or selecting them.
Styling with Markdown is supported

Open a pull request

Create a new pull request by comparing changes across two branches. If you n

base fork: chefconf-crc/tools base: master ... head fork: spa

✓ Able to merge. These branches can be automatically merged.

Adding team test to tools

Last activity in Lab 0.

Join the Chef Community Slack

Follow the instructions on the Community Slack Page and then sign in to the Chef Community Slack. Join channel #chefconf-crc. You can either use Slack from a web browser or from the app.

Questions? You can ping the Chef Staff support on Slack at:

- sigje
-

Outcomes

- You will have logged into your node and configured git.
- You will have updated the tools that your team uses and created a pull request.
- You will have signed up for the Chef Community Slack.

Lab 2

Create Chef Cookbook using chef generate

Allotted Time: 20 minutes

In this activity, you will generate a sample cookbook using the chef generate command. This will be a quick review for many, the essentials to generate the cookbook, create a kitchen.docker.yml file and converge the node. This will verify that you know the essentials for creating and converging on your system and that everything is working correctly.

Create cookbook and an install_apache recipe.

```
 cd ~/wd
 chef generate cookbook apache
 cd apache
 chef generate recipe . install_apache
 vi recipes/install_apache.rb
```

Add the follow `resources` to the `install_apache` recipe.

```
package 'httpd'

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

Edit the default recipe:

```
   vi recipes/default.rb
```

Update the contents of the `default.rb` recipe with the following contents:

```
include_recipe 'apache::install_apache'
```

Create a `.kitchen.docker.yml` to match the following configuration.

```
---
driver:
  name: docker

provisioner:
  name: chef_zero

# Uncomment the following verifier to leverage Inspec instead of Busser (the
# default verifier)
# verifier:
#   name: inspec
#   format: doc
```

```
platforms:
  - name: centos-6.5
    driver_config:
      forward:
      - 80:80

suites:
  - name: default
    run_list:
      - recipe[apache::default]
    attributes:
```

Update which configuration kitchen uses by setting the *KITCHEN_LOCAL_YAML* file and then converge your node.

```
$ export KITCHEN_LOCAL_YAML=.kitchen.docker.yml
$ kitchen list
```

Verify that you see 1 instance with a driver Docker.

```
Instance        Driver  Provisioner  Verifier  Transport  Last Action
default-centos-65  Docker  ChefZero    Busser   Ssh     <Not Created>
```

Create, and converge the node and login.

```
$ kitchen converge
$ kitchen login
```

How do you know if your recipe worked? Kitchen converge finishes without errors, and you have a port up and running. You can check by browsing directly to the node because you have set up port forwarding!

Validate your node has apache up and running:

```
curl localhost
```

Outcomes

- Use chef generate command to create a cookbook.
- Understand basic kitchen configuration.
- Successfully running Apache on your node.

Lab 3

Translate a runbook for installing MongoDB into chef

Allotted Time: 20 minutes

In this activity your team will be creating a community cookbook from scratch.

1. Read the following instructions.
2. Identify who will do what on the team and pair as needed.
3. Create a teamname-mongodb cookbook
4. Commit teamname-mongodb to github repo; make sure all team members have appropriate access to github repo.
5. Upload teamname-mongodb cookbook to the class supermarket https://supermarket.reusablechef.com/dashboard.

MongoDB is an open-source, document-oriented database designed for ease of development and scaling. The MongoDB documentation site includes a installation guide on how to install MongoDB on Red Hat Enterprise Linux, CentOS Linux, Fedora Linux, or a related system.

- Chef Resource Documentation

Read the installation guide, identify the resources you need, and create a mongodb cookbook populated with an appropriate recipe.

As a team identify

- What users do you need? What groups?
- What packages do you need?
- What configurations do you need?
- Where is data stored?
- What directories do you need to create?
- What services do you need?

Hint:

Use the yum community cookbook

The yum community cookbook provides resource *yum_repository* that allows us to make individual yum repositories available for use.

In the installation guide, we see that we need to configure a yum repo in order to install mongodb packages using yum.

```
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/
gpgcheck=1
```

```
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
```

translates to chef dsl as:

```
yum_repository 'mongodb-org-3.2' do
  description 'MongoDB Repository'
baseurl 'https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/'
  gpgcheck true
  gpgkey 'https://www.mongodb.org/static/pgp/server-3.2.asc'
  enabled true
  action :create
```

Make sure your hosted chef account has connected to the supermarket.

Browse to the class supermarket website at https://supermarket.reusablechef.com.

Create (or login to your existing) hosted chef account.

Creating a hosted chef account

If you have not created a hosted chef account, fill in the form with a real email address, as you will need to verify your account.

Check your email for the verification link.

Click on the verification link. This will generate your pemfile. Copy and save this pemfile to your node (and you may want to save this locally as well). The node will go away, this hosted chef account is yours free to do with as you wish.

Login to your existing hosted chef account.

Copy your user pemfile to your node. If you don't have this currently, you will need to regenerate (which will invalidate your old pemfile, and any use of the old pemfile will fail).

Private versus Public Supermarket

Note. When you use the Public Supermarket you should link your Github account to your Chef account so that your contributions on GitHub are signed with your Contributors License Agreement.

If you haven't signed the Contributors License Agreement, and your company doesn't have a company license agreement make sure that you Sign the individual contributor license agreement.. This will ensure that any contributions you make in this class and in the future can get included in community cookbooks.

Configure stove on your node (Everyone)

Stove

You want to set up authentication to supermarket using your pem, and set the endpoint to be the class supermarket. (If you don't set it up as the class supermarket, it will push to the public supermarket instead!) If you forget to configure extended-metadata then your cookbook uploads will not include the metadata information that comes from Chef 12+ (issues_url for example).

```
$ stove login --username YOURUSERNAME --key ~/.chef/HOSTEDCHEF.pem
```

Upload completed cookbook to the supermarket (1 person)

Change to the cookbook directory and then execute stove.

```
$ chef exec stove --endpoint https://supermarket.reusablechef.com/api/v1
```

Login to the class supermarket and add the rest of your team as moderators to the cookbook.

https://supermarket.reusablechef.com

Outcome

- mongodb running on the node within a docker container
- code checked into repository on github
- A successful push to teamname-mongodb to the class supermarket https://supermarket.reusablechef.com/dashboard.

Hint

- If you need further hints, you can check out this working example https://github.com/nathenharvey/install_mongo of minimal requirements to install mongodb.
- Be aware that mongodb attempts to change ulimits in the init file which won't work in a docker container. So a second converge will fail!

Lab 4

Translate our MongoDB cookbook from recipes into resources

Allotted Time: 30 minutes total

In this activity your team will be updating your teamname-mongodb community cookbook to use custom resources.

1. Identify who will do what on the team and pair as needed.
2. Create a custom resource for installing mongodb.
3. Update documentation for teamname-mongodb.
4. Commit teamname-mongodb to github repo.
5. Upload teamname-mongodb cookbook to the class supermarket https://supermarket.reusablechef.com/dashboard.

Custom Resources 12.5 Style

- simple extension of Chef
- implemented as part of a cookbook
- follows easy, repeatable syntax patterns
- effectively leverages resources that are built into Chef
- is reusable in the same way as resources that are built into Chef

Defined as ruby file (.rb extension) located in cookbook's resources directory.

```
property :name, RubyType, default: 'value'

load_current_value do
  # some Ruby
end

action :name do
 # a mix of built-in Chef resources and Ruby
end

action :name do
 # a mix of built-in Chef resources and Ruby
end
```

In the following example, this resource would create a Example:

```
resource_name :httpd

property :homepage, String, default: '<h1>Hello world!</h1>'
```

```
load_current_value do
  if ::File.exist?('/var/www/html/index.html')
    homepage IO.read('/var/www/html/index.html')
  end
end

action :create do
  package 'httpd'

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

  file '/var/www/html/index.html' do
    content homepage
  end
end
```

Example of using this resource

```
httpd 'build website' do
  homepage '<h1>Welcome to the Example Co. website!</h1>'
  action :create
end
```

Walk-through of creating custom resources.

Outcomes

- custom resource file in resources directory of cookbook
- simplified recipe in cookbook
- updated cookbook to the class supermarket
- cookbook committed to github

Lab 5

Maintainable Chef

Allotted Time: 30 minutes total

1. Identify who will do what on the team and pair as needed.
2. Add the ubuntu platform to your kitchen configuration, and update your cookbook to handle the new platform.
3. Clean any rubocop and foodcritic issues reported.
4. Add inspec integration tests to MongoDB cookbook.

Adding Ubuntu platform

Update the `.kitchen.docker.yml` in your cookbook.

To platforms section, add

- name: ubuntu-16.04

Check the installation guide for mongodb to see Ubuntu requirements.

Linting

Run *foodcritic*, the chef linting tool from within your team's mongodb cookbook.

```
foodcritic .
```

Run *rubocop*, the ruby linting tool from within your team's mongodb cookbook.

```
rubocop .
```

Clean up any errors that you discover with your mongodb cookbook.

Integration Tests

Add inspec tests to your team's mongodb cookbook.

https://github.com/chef/inspec

Outcomes

- No foodcritic or rubocop errors.
- Integration test added to cookbook.
- cookbook committed to github

- A successful update to your teams assigned cookbook to the class supermarket https://supermarket.reusablechef.com/dashboard.

Lab 6

Extending Existing Cookbooks

Allotted Time: 1 hour total

In this activity, your team will be assigned a specific community cookbook and you will extend this cookbook.

1. Examine the cookbook.
2. Read open issues.
3. Read open pull requests.
4. Identify specific tasks for your team to complete.
5. Execute on your plan.
6. Measure success. How successful were you on completing what you planned?

Within your team as you examine the assigned cookbook, assess the following:

- Does it make sense for this cookbook to be multi-platform?
- Is the cookbook attribute or resource driven?
- How tightly bound is the cookbook to any dependencies?
- Are there hidden assumptions based on the current implementation of the cookbook?
- Are there items that impact the quality of the cookbook reported on the supermarket? Quality metrics on the supermarket currently only measure the output of a foodcritic run, and whether the cookbook has more than 1 collaborator. https://github.com/chef-cookbooks/cookbook-quality-metrics
- Is any major functionality missing?
- Is there testing?
- Is there a fixture cookbook for testing?
- Are the foodcritic and rubocop rules applicable?
- What is the overall test coverage for this cookbook?
- Is there missing documentation?
- Is the scope well defined?
- Are dependencies accurate and up to date?
- Are all resources, recipes, and attributes described?
- Is there example usage?
- Is the platform coverage accurate based on the current cookbook?
- Is the project under active development?
- Are there a large number of current pull requests? Are any of them useful to implement?
- Is there repetitive code that can be converted into a reusable pattern?
- Does this project affect/influence any other active projects in this workshop?

- Is the investment (time) required to correct existing issues/extend the cookbook greater than a doing a full-rewrite?
- Does it make sense to incrementally improve or completely rewrite the cookbook?

Update the file COOKBOOK.md in the assessments repo (git@github.com:chefconf-crc/assessments.git)

The plan should include specific information about what will be done and who will be doing it.

The plan could include * new or updated tests * start of a new cookbook * additional functionality * new or updated documentation * refactored code from attribute driven to resource driven, or removal of repetitious code

Update the file COOKBOOK.md in the plans repo (git@github.com:chefconf-crc/plans.git).

Outcomes

- Updated COOKBOOK.md in the assessments repo in chefconf-crc. git@github.com:chefconf-crc/assessments.git
- Updated COOKBOOK.md in the plans repo in chefconf-crc organization. git@github.com:chefconf-crc/plans.git
- A successful update to your teams assigned cookbook to the class supermarket https://supermarket.reusablechef.com/dashboard.

TA's are ready and willing to help you with your questions!

Lab 7

Moderation and Contribution

Allotted Time: 1 hour total

In this activity, your team will be assigned a new community cookbook to review what another team did. You will decide on what your team can contribute, and issue a pull request to have your contributions added.

In addition you will review another team's requests to you, and accept the pull requests or request additional information as needed.

1. Examine the cookbook.
2. Review the previous team's assessment, plan and contributions.
3. Read open issues.
4. Read open pull requests.
5. Identify specific tasks for your team to complete. Update the COOKBOOK.md plan adding your teams plan at the bottom.
6. Execute on your plan.
7. Measure success. How successful were you on completing what you planned?

Outcomes

- An updated COOKBOOK.md in the assessments repo in chefconf-crc. git@github.com:chefconf-crc/assessments.git
- An updated COOKBOOK.md in the plans repo in chefconf-crc organization. git@github.com:chefconf-crc/plans.git
- A successful update to your teams assigned cookbook to the class supermarket https://supermarket.reusablechef.com/dashboard.

TA's are ready and willing to help you with your questions!