Chef Resources

Chef's Fundamental Building Blocks

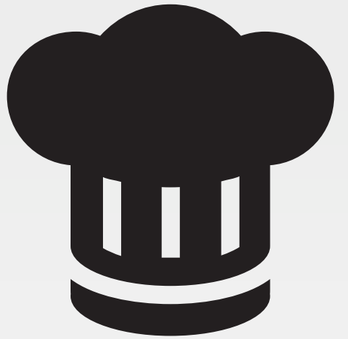
CHEF

Objectives

After completing this module, you should be able to:

- Define Chef Resources
- Use the chef-client command
- Create a basic Chef recipe file

CHEF



Group Lab: Hello, World?

I heard Chef is written in Ruby. If that's the case it's required that we write a quick "Hello, world!" application.

Objective:

- ❑ Create a recipe file writes out 'Hello, world!' to a text file
- ❑ Apply the recipe to the workstation

CHEF

Learning Chef

One of the best ways to learn a technology is to apply the technology in every situation that it can be applied.

A number of chef tools are installed on the system so lets put them to use.

Choose an Editor

You'll need to choose an editor to edit files:

Tips for using these editors can be found below in your participant guide.

atom

Visual Studio Code (← this one is awesome)

CHEF

DOCS

Resources

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.

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

CHEF

Example: Package

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

The package named 'httpd' is installed.

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

Example: Service

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

The service named 'ntp' is enabled (start on reboot) and started.

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

Example: File

```
file '/etc/motd' do
  content 'This computer is the property...'
end
```

The file name '/etc/motd' is created with content 'This computer is the property ...'

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

Example: File

```
file '/etc/php.ini.default' do
  action :delete
end
```

The file name '/etc/php.ini.default' is deleted.

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

Working within Home Directory

```
> cd ~
```

GL: Create and Open a Recipe File

```
> code hello.rb
```

GL: Create a Recipe File Named hello.rb

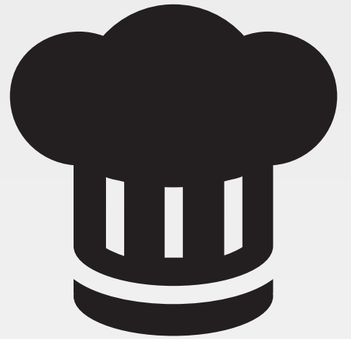
~\hello.rb

```
file 'C:\hello.txt' do
  content 'Hello, world!'
end
```

The file named ’c:\hello.txt' is created with the content 'Hello, world!'

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

CHEF



Group Lab: Hello, World?

I heard Chef is written in Ruby. If that's the case it's required that we write a quick "Hello, world!" application.

Objective:

- ✓ Create a recipe file writes out 'Hello, world!' to a text file
- ❑ Apply the recipe to the workstation

CHEF

CONCEPT

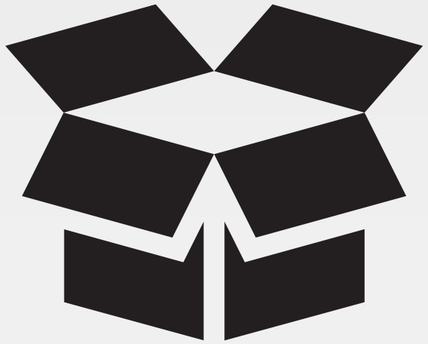
chef-client

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.

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

CONCEPT



chef-apply

chef-apply is an executable program that runs a single recipe from the command line

CHEF

GL: Apply a Recipe File

```
> chef-apply hello.rb -l info
```

```
[2017-08-11T04:00:38+00:00] INFO: Run List is []
[2017-08-11T04:00:38+00:00] INFO: Run List expands to []
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[C:\hello.txt] action create[2017-08-11T04:00:38+00:00] INFO: Processing
file[C:\hello.txt] action create ((chef-apply cookbook)::(chef-apply recipe) line
1)
[2017-08-11T04:00:38+00:00] INFO: file[C:\hello.txt] created file C:\hello.txt

    - create new file C:\hello.txt[2017-08-11T04:00:38+00:00] INFO:
file[C:\hello.txt] updated file contents C:\hello.txt


    - update content in file C:\hello.txt from none to 315f5b
    --- C:\hello.txt    2017-08-11 04:00:38.000000000 +0000
```

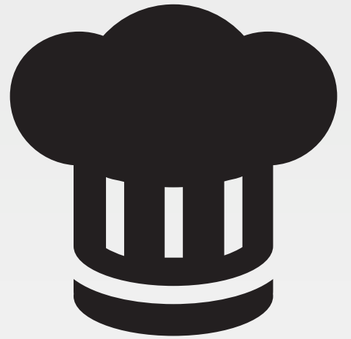

CHEF

GL: What Does hello.txt Say?

```
> gc C:\hello.txt
```

```
Hello, world!
```


CHEF



Group Lab: Hello, World?

I heard Chef is written in Ruby. If that's the case it's required that we write a quick "Hello, world!" application.

Objective:

- ✓ Create a recipe file writes out 'Hello, world!' to a text file
- ✓ Apply the recipe to the workstation

Discussion

1. What would happen if you applied the recipe again?

2. What would happen if the package were to become uninstalled?

Lab: Test and Repair

What would happen if the file contents were modified?

- ❑ Modify the contents of 'C:\hello.txt' manually with your text editor
- ❑ Run the chef-client command again

Modify the Contents of the File

```
C:\hello.txt
```

```
Goodbye, world!
```


CHEF

Re-apply the Policy Defined in the Recipe

```
> chef-apply hello.rb -l info
```

```
[2017-08-11T04:02:45+00:00] INFO: file[C:\hello.txt] updated file contents
C:\hello.txt

    - update content in file C:\hello.txt from f701ac to 315f5b
    --- C:\hello.txt    2017-08-11 04:02:33.000000000 +0000
    +++ C:\chef-hello20170811-2148-13txos5.txt  2017-08-11 04:02:45.000000000
+0000
    @@ -1,2 +1,2 @@
    -Adios world!
    +Hello, world!
```


CHEF

Lab: Test and Repair

What would happen if the file contents were modified?

- ✓ Modify the contents of 'C:\hello.txt' manually with your text editor
- ✓ Run the chef-client command again

CONCEPT

Test and Repair

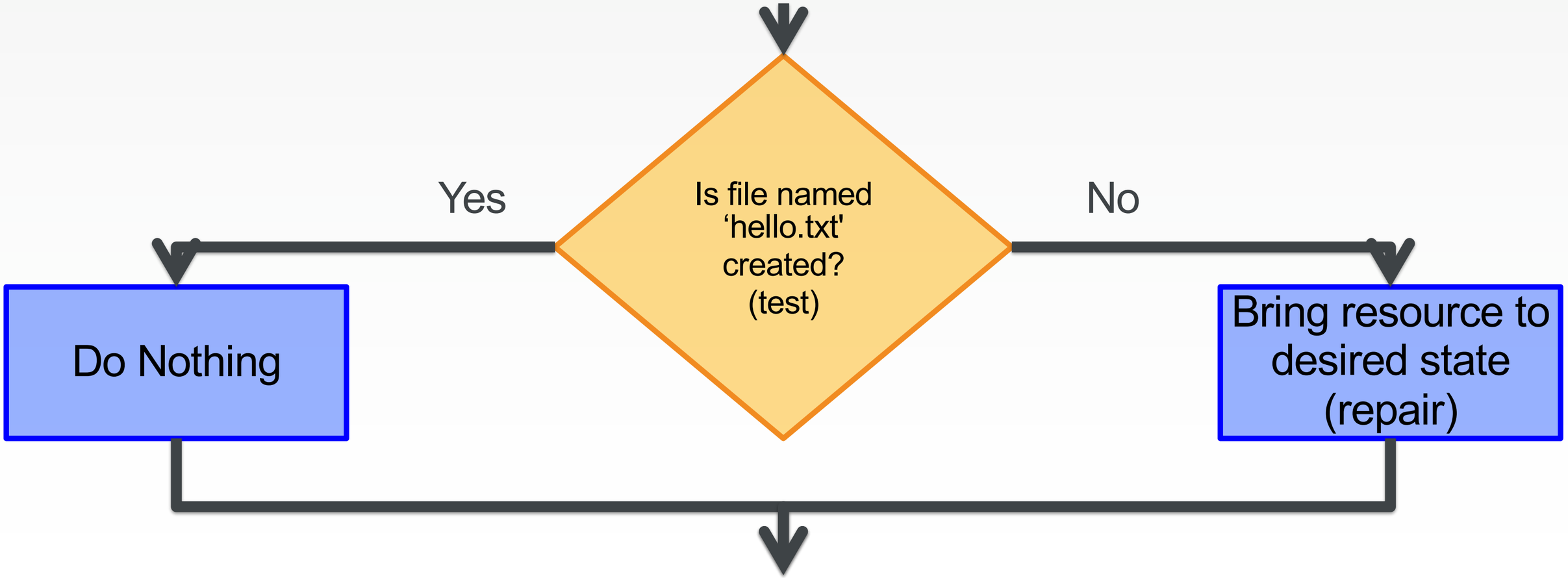
`chef-client` takes action only when it needs to. Think of it as test and repair.

Chef looks at the current state of each resource and takes action only when that resource is out of policy.

CHEF

Test and Repair

```
file 'C:\hello.txt'
```



Test and Repair

What would happen if the file permissions (mode), owner, or group changed?

Have we defined a policy for these properties?

CONCEPT

Resource Definition

```
file '/hello.txt' do
  content 'Hello, world!'
end
```

The **TYPE** named **NAME** should be **ACTION'd** with **PROPERTIES**

CHEF

CONCEPT

Resource Definition

```
file '/hello.txt' do
  content 'Hello, world!'
end
```

The TYPE named NAME should be **ACTION'd** with PROPERTIES

CHEF

CONCEPT

Resource Definition

```
file '/hello.txt' do
  content 'Hello, world!'
end
```

The **TYPE** named **NAME** should be **ACTION'd** with **PROPERTIES**

CHEF

CONCEPT

Resource Definition

```
file '/hello.txt' do
  content 'Hello, world!'
end
```

The TYPE named NAME should be **ACTION'd** with PROPERTIES

CHEF

CONCEPT

Resource Definition

```
file '/hello.txt' do
  content 'Hello, world!'
end
```

?

The **TYPE** named **NAME** should be **ACTION'd** with **PROPERTIES**

CHEF

Lab: The `file` Resource

- ❑ **Read** https://docs.chef.io/resources.html
- ❑ **Discover the file resource's:**
  - default action
  - **rights** attribute
- ❑ **Update the `file` policy in "hello.rb" to:**

  The file named 'C:\hello.txt' is created with 'read' rights for 'Everyone'.

Lab: The Updated file Resource

~\hello.rb

```
file 'C:\hello.txt' do
  content 'Hello, world!'
  rights :read, 'Everyone'
  action :create
end
```

This sets the file's rights to allow 'Everyone' access.

The default action is to create (not necessary to define it).

CHEF

Lab: The `file` Resource

- ✓ **Read** https://docs.chef.io/resources.html
- ✓ **Discover the file resource's:**
  - default action
  - **rights** attribute
- ✓ **Update the `file` policy in "hello.rb" to:**

  The file named 'C:\hello.txt' is created with 'read' rights for 'Everyone'.

Questions

What questions can we answer for you?

CHEF

Discussion

What is a resource?

What are some other possible examples of resources?

How did the example resources we wrote describe the desired state of an element of our infrastructure?

What does it mean for a resource to be a statement of configuration policy?

CHEF

Q&A

What questions can we answer for you?

- chef-client
- chef-apply
- Resources
- Resource - default actions and default properties
- Test and Repair

CHEF™