Chef Resources and Recipes

Chef's Fundamental Building Blocks

Objectives

After completing this module, you should be able to:

- Define Chef Resources
- Create a basic Chef recipe file
- Use Chef to set the policy on your workstation
- Use the chef-client command

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/resource_reference.html

CHEF

Resource Definition

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

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

Resource Definition

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

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

Resource Definition

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

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

CONCEPT

Resource Definition

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

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

Resource Definition

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

?

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

Example: powershell_script

```
powershell_script 'Install IIS' do
  code 'Add-WindowsFeature Web-Server'
  action :run
end
```

The powershell_script resource named 'Install IIS' is run with the code 'Add-WindowsFeature Web-Server'.

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

CHEF

Example: service

```
service 'w3svc' do
  action [ :enable, :start ]
end
```

The service named 'w3svc' is enabled (start on reboot) and started.

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

Example: file

```
file 'C:\inetpub\wwwroot\Default.htm' do
  content 'Hello, world!'
  rights :read, 'Everyone'
end
```

The file 'C:\inetpub\wwwroot\Default.htm' with the content 'Hello, world!' and grants 'read' rights for 'Everyone'.

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

Example: file

```
file 'C:\PHP\php.ini' do
  action :delete
end
```

The file name 'C:\PHP\php.ini' is deleted.

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

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.

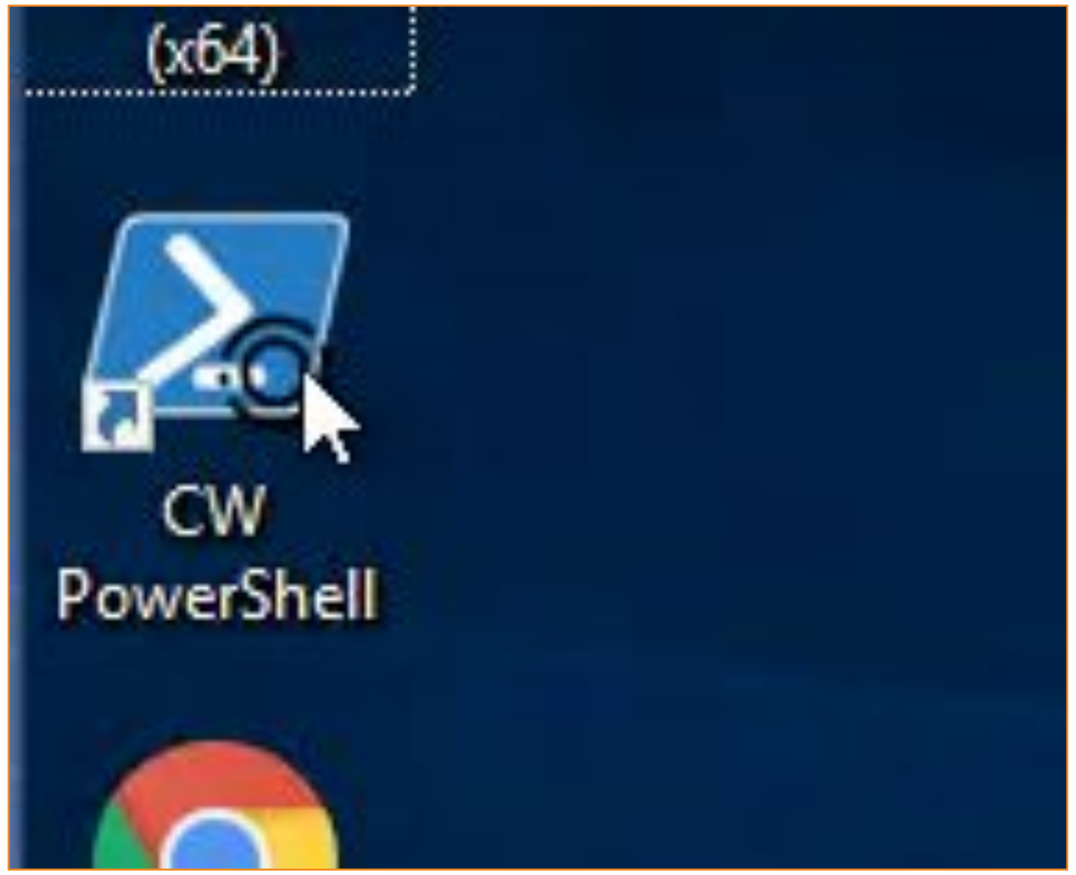
Objective:

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

GL: Start the Chef Workstation Powershell

Double-click the **Chef Workstation Powershell** shortcut on your virtual workstation's desktop.

It may take a few seconds for the prompt to display.



GL: Ensure You Are in Your Home Dir

```
> pwd
```

```
C:\Users\Administrator\Desktop>
```

Important: A lot of the work you'll do today will be done within your home directory (on Windows it's **C:\Users\Administrator)** or in a subdirectory to your home directory. You should always ensure you are in the proper directory as indicated in these course materials.

Generally, you can type **cd ~** to return to your home directory if you had navigated away from your home directory.

GL: Move to Your Home Directory

```
> cd ~
```

```
C:\Users\Administrator
```

GL: List the Contents of Your Home Directory

```
> dir
```

```
 Directory: C:\Users\Administrator

Mode                LastWriteTime     Length Name

----                -------------     ------ ----
d----          8/3/2018  10:33 PM             .atom
d----         7/11/2020   4:00 PM             .chef
d----         6/26/2020   7:21 PM             .chef-workstation
d-r--        11/13/2017   8:03 PM             Contacts
d-r--         6/28/2020   4:38 PM             Desktop
d-r--        11/13/2017   8:26 PM             Documents
d-r--         7/11/2020   2:31 PM             Downloads
d-r--        11/13/2017   8:03 PM             Favorites
d-r--          8/3/2018  10:21 PM             Links
...
```

GL: Create and Open a Recipe File

```
> atom hello.rb
```

It could take some time for Atom to launch the first time you launch it.

GL: Create a Recipe File Named hello.rb

```
~\hello.rb
```

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

Type these contents into your hello.rb file in Atom.

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

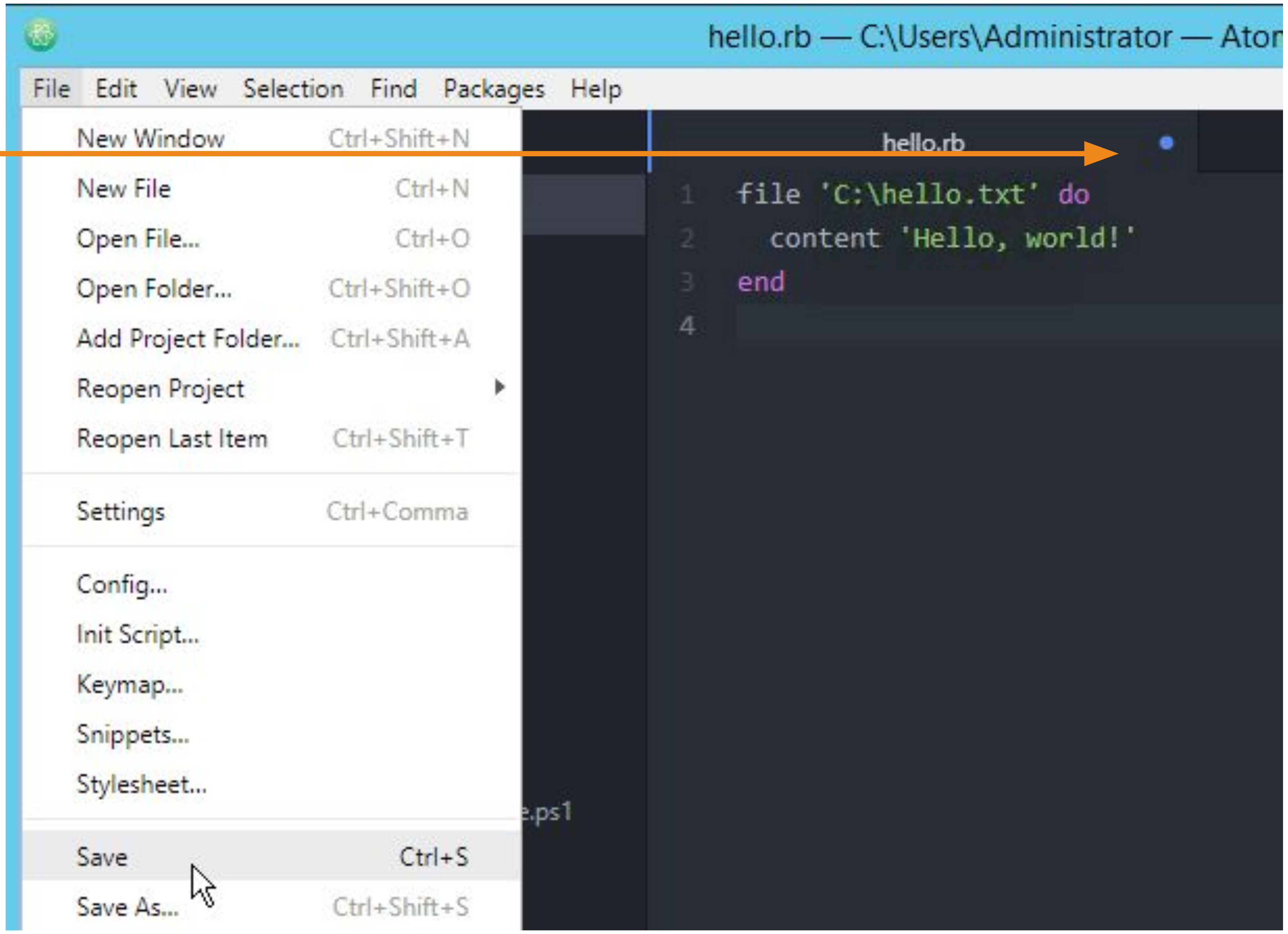
https://docs.chef.io/recipes/

CHEF

GL: Create a Recipe File Named hello.rb

In Atom, the blue dot here indicates your changes have not been saved yet.

Save your changes using the File menu or **Ctrl > s**.



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

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

CONCEPT

chef-client

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

When the chef-client application 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

CHEF

CONCEPT

--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 local mode.

CHEF

GL: Apply a Recipe File

```
> chef-client --chef-license accept --local-mode hello.rb
```

```
[2020-07-12T16:53:48+00:00] WARN: No cookbooks directory found at or above current directory.
Assuming C:/Users/Admin istrator.
✔ 2 product licenses accepted.
+---------------------------------------------+
...
Recipe: @recipe_files::C:/Users/Administrator/hello.rb
  * file[C:\hello.txt] action create
    - create new file C:\hello.txt
    - update content in file C:\hello.txt from none to 315f5b
    --- C:\hello.txt    2020-07-12 16:53:57.798746200 +0000
    +++ C:\chef-hello20200712-3056-1rev5o6.txt  2020-07-12 16:53:57.798746200 +0000
    @@ -1 +1,2 @@
    +Hello, world!
...
```

GL: What Does hello.txt Say?

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

```
Hello, world!
```

The **gc** command stands for Get Content in Windows. It's similar to the **cat** command in Linux systems.

Discussion

What would happen if you ran the command again?

What would happen if the file were removed?

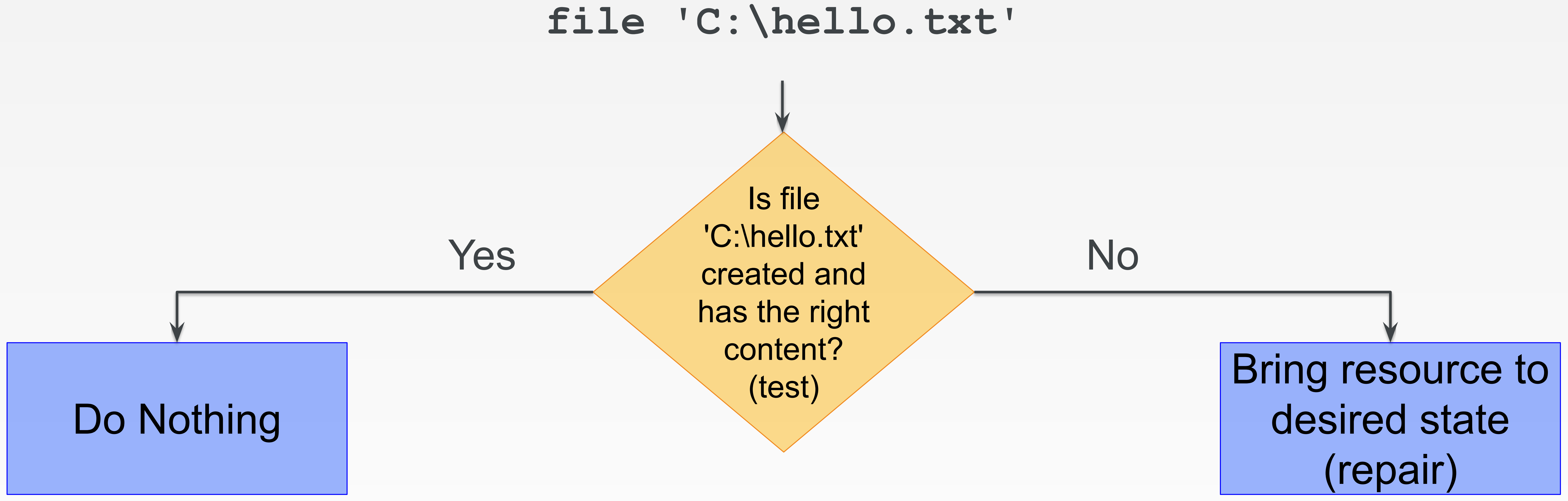
CONCEPT

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.

Test and Repair


file 'C:\hello.txt'
Is file 'C:\hello.txt' created and has the right content? (test)
Yes
No
Do Nothing
Bring resource to desired state (repair)

Lab: Goodbye Recipe

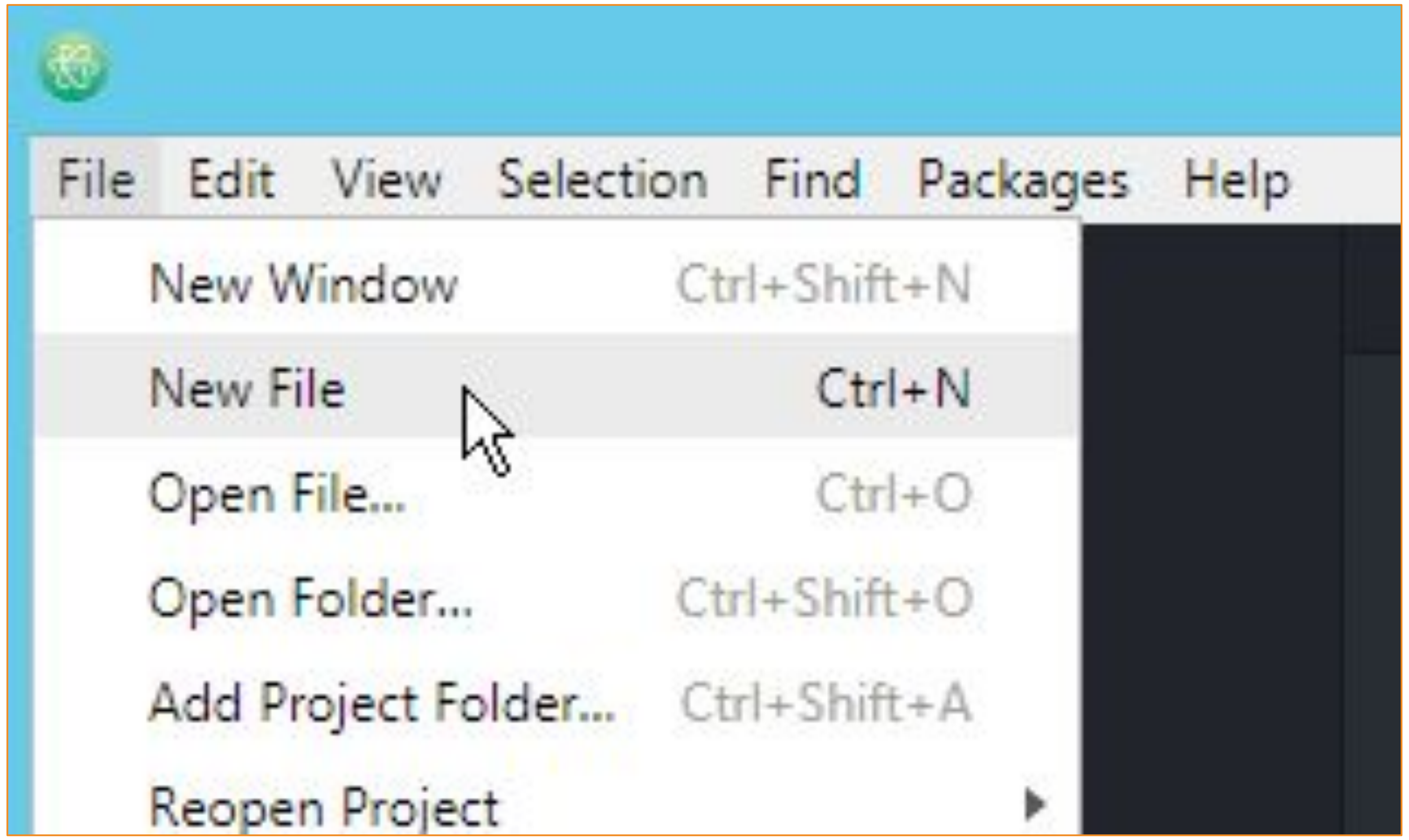
We know how to create a file with Chef, what about removing it?

Objective:

- ❑ Create a recipe named ‘goodbye.rb’ that removes 'C:\hello.txt'
- ❑ Apply the recipe to the workstation

Lab: Editor Tip

In the next step you will create another file under the Administrator directory. You can do that in Atom by clicking **File > New File** and then naming the file via **Save As**.



Lab: Adding a file Resource to Delete a File

```
~\goodbye.rb
```

```
file 'C:\hello.txt' do
  action :delete
end
```

Lab: Goodbye Recipe

We know how to create a file with Chef, what about removing it?

Objective:

- ✔ Create a recipe that removes 'C:\hello.txt'
- ❑ Apply the recipe to the workstation

Lab: Apply a Recipe File

```
> chef-client --local-mode goodbye.rb
```

```
resolving cookbooks for run list: []
...
[2020-06-27T16:17:42+00:00] WARN: Node WIN-NJ5OO7IAJNR.ec2.internal has an
empty run list.
Converging 1 resources

Recipe: @recipe_files::C:/Users/Administrator/goodbye.rb

  * file[C:\hello.txt] action delete

    - delete file C:\hello.txt
```

Lab: Test that the File was Deleted

```
> Test-Path C:\hello.txt
```

```
False
```

GL: Disable Limited User Account

Managing files is nice but what about Registry keys?

Objective:

- ❑ Create a recipe that disables Limited User Account
- ❑ Apply that recipe

GL: Disable the Limited User Account

~\disable-uac.rb

```
system_policies = 'HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System'

registry_key system_policies do
  values [{
    :name => 'EnableLUA',
    :type => :dword,
    :data => 0
  }]
end
```

GL: Disable the Limited User Account

~\disable-uac.rb

```
system_policies = 'HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System'

registry_key system_policies do
  values [{
    :name => 'EnableLUA',
    :type => :dword,
    :data => 0
  }]
end
```

GL: Disable Limited User Account

Managing files is nice but what about Registry keys?

Objective:

- ✔ Create a recipe that disables Limited User Account
- ❑ Apply that recipe

GL: Apply a Recipe File

```
> chef-client --local-mode disable-uac.rb
```

```
Starting Chef Infra Client, version 17.3.48
...

Converging 1 resources

Recipe: @recipe_files::C:/Users/Administrator/disable-uac.rb

  * registry_key[HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System]
action create

    - set value {:name=>"EnableLUA", :type=>:dword, :data=>0}


Running handlers:
```

Lab: Disable Consent Prompt

- ❑ Update the recipe file named "disable-uac.rb" to also define:

  The registry_key named 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System' has the values:

  [{ :name => 'ConsentPromptBehaviorAdmin', :type => :dword, :data => 0 }]

- ❑ Use **chef-client** to apply the recipe file named "disable-uac.rb"

Lab: Disable the Limited User Account

~\disable-uac.rb

```
system_policies = 'HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System'

registry_key system_policies do
  # ... ENABLE LUA VALUES (NOT SHOWN HERE TO CONSERVE SPACE)
end

registry_key system_policies do
  values [{
    :name => 'ConsentPromptBehaviorAdmin',
    :type => :dword,
    :data => 0
  }]
end
```

Lab: Disable Consent Prompt

- ✔ Update the recipe file named "disable-uac.rb" to also define:

  The registry_key named 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System' has the values:

  [{ :name => 'ConsentPromptBehaviorAdmin', :type => :dword, :data => 0 }]

- ❑ Use **chef-client** to apply the recipe file named "disable-uac.rb"

CHEF

Lab: Apply a Recipe File

```
> chef-client --local-mode disable-uac.rb
```

```
Starting Chef Infra Client, version 17.3.48
...
[2020-06-27T16:25:53+00:00] WARN: Node WIN-NJ5OO7IAJNR.ec2.internal has an empty run list.
Converging 2 resources

Recipe: @recipe_files::C:/Users/Administrator/disable-uac.rb

  * registry_key[HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System] action
create (up to date)

  * registry_key[HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System] action
create

    - set value {:name=>"ConsentPromptBehaviorAdmin", :type=>:dword, :data=>0}
```

Review Questions

What is a resource?

What is a Chef recipe?

What is chef-client and what does it do?

Q&A

What questions can we answer for you?

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

CHEF™