



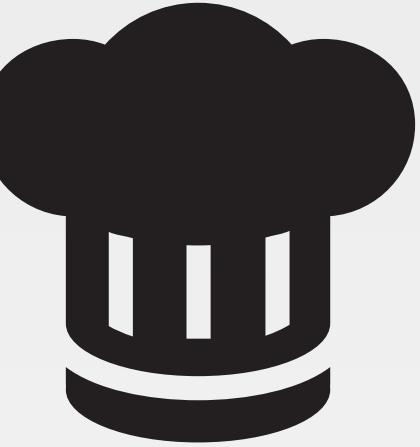
Chef Resources

Chef's Fundamental Building Blocks

Objectives

After completing this module, you should be able to:

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



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



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>

Example: powershell_script

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

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

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

Example: service

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

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

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

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'.

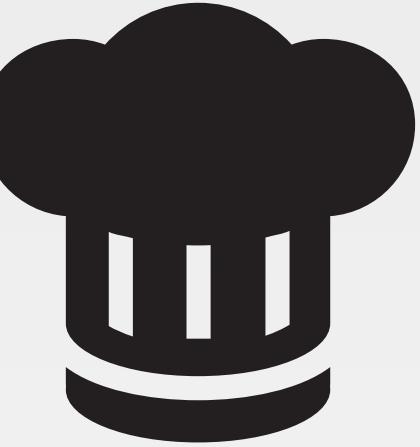
https://docs.chef.io/resource_file.html

Example: file

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

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

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



Group Lab: Hello, World?

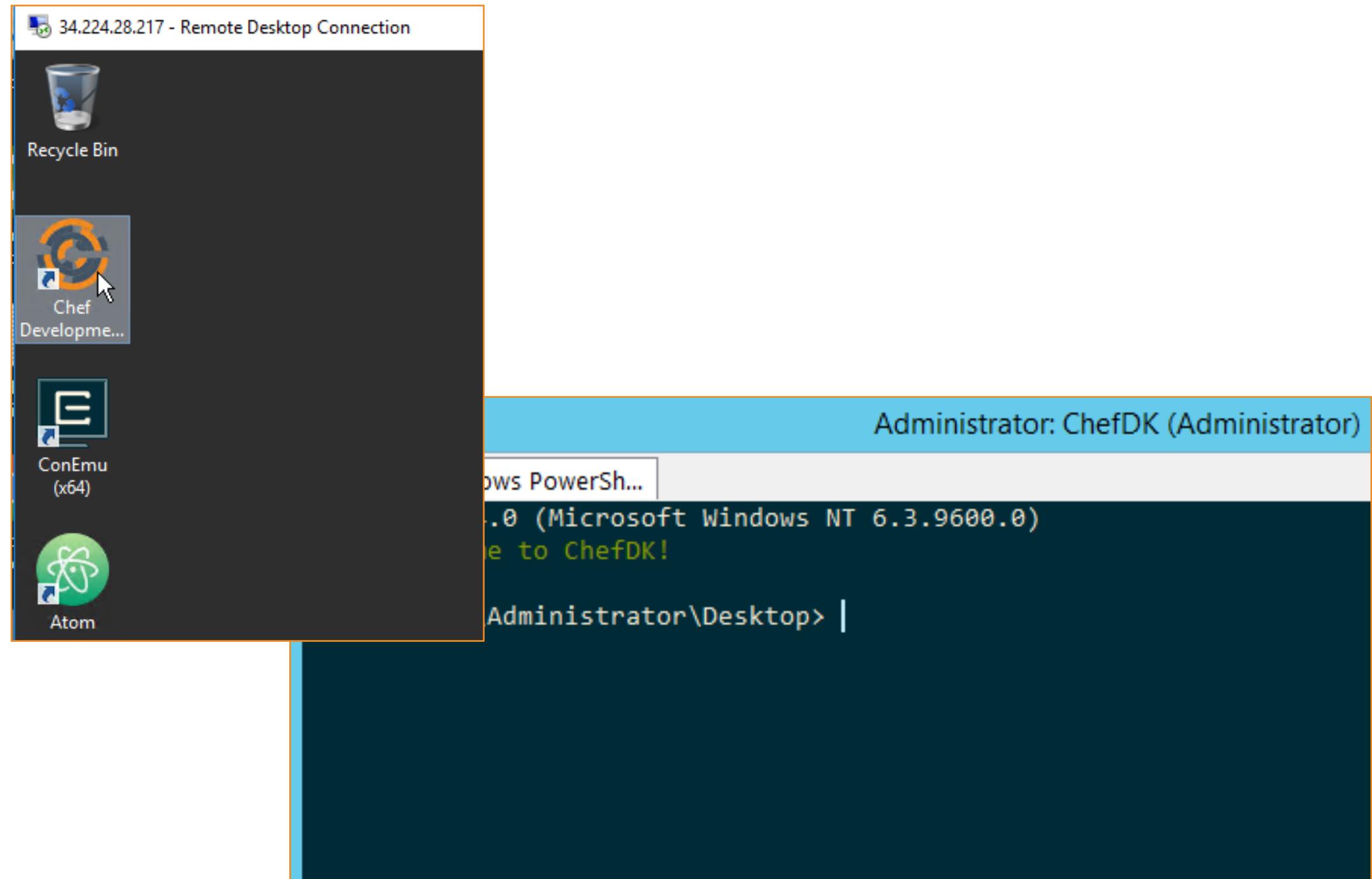
I heard Chef is written in Ruby. If that's the case it's required that we write a quick "Hello, world!" application.

Objective:

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

GL: Launch the ChefDK PowerShell

Double-click the ChefDK icon on your virtual workstation.



GL: Navigate to the Home Directory



```
> cd C:\Users\chef\cookbooks>
```

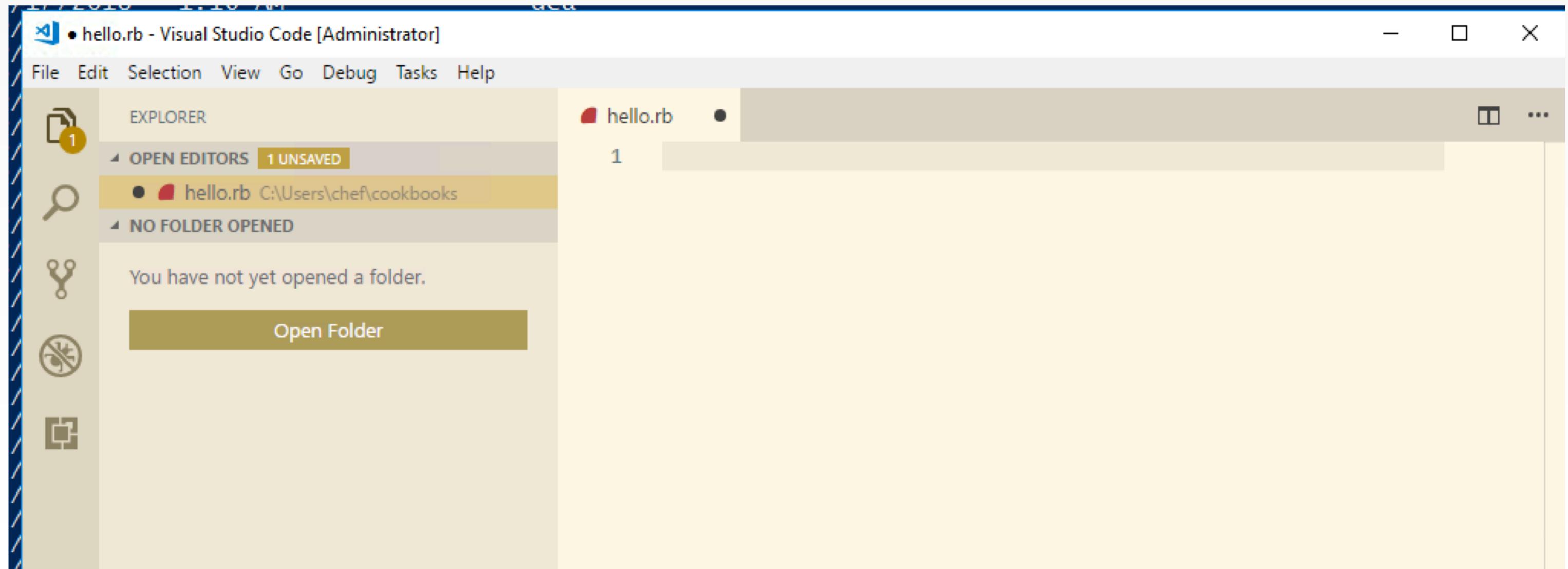
```
PS C:\Users\chef\cookbooks>
```

LOCAL

GL: Create and Open a Recipe File



```
C:\Users\chef\cookbooks> code hello.rb
```



| LOCAL

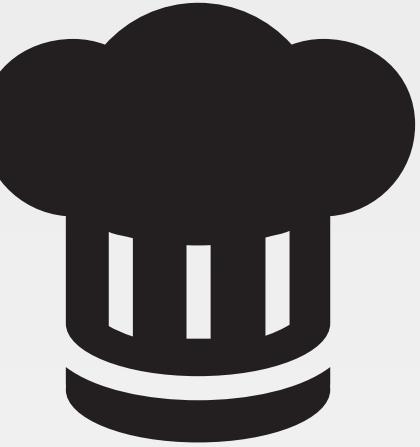
GL: Create a Recipe File Named hello.rb

~\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>



Group Lab: Hello, World?

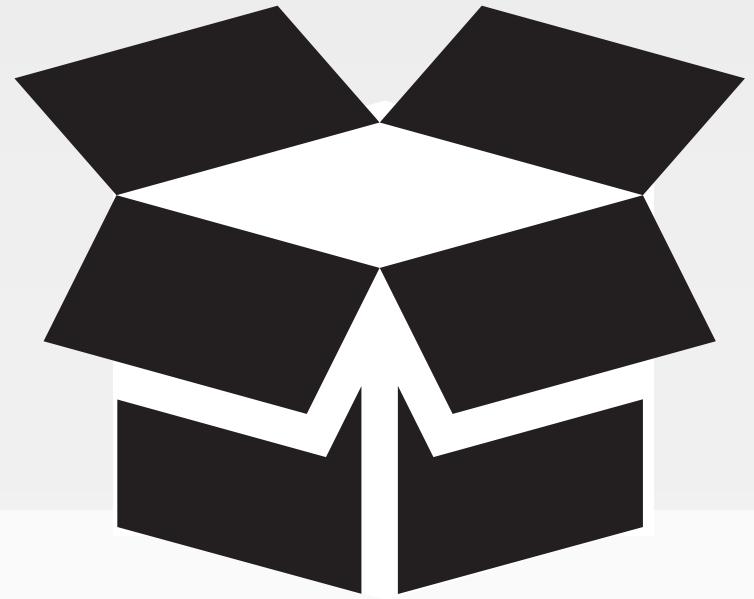
I heard Chef is written in Ruby. If that's the case it's required that we write a quick "Hello, world!" application.

Objective:

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

CONCEPT

chef-client



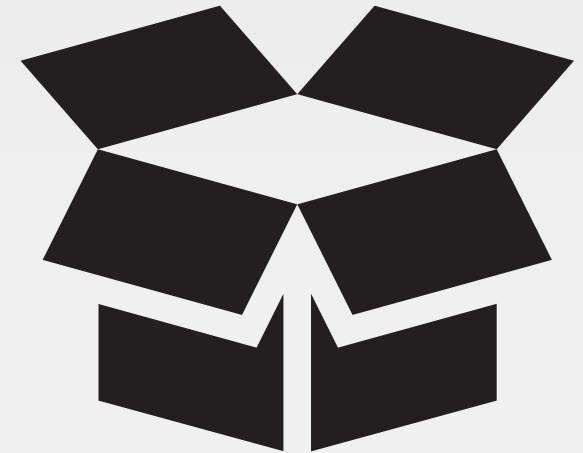
chef-client is an agent that runs locally on every node that is under management by Chef.

When 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

--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.

GL: Apply a Recipe File



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

```
...
[2018-09-08T16:48:26+00:00] WARN: No cookbooks directory found at or above current directory.
Assuming C:/Users/Administrator.

Starting Chef Client, version 13.3.42

...
Converging 1 resources

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 2018-09-08 16:49:39.000000000 +0000
      +++ C:\chef-hello20180908-3804-txt4m7.txt 2018-09-08 16:49:39.000000000 +0000
      @@ -1 +1,2 @@
      +Hello, world! ...

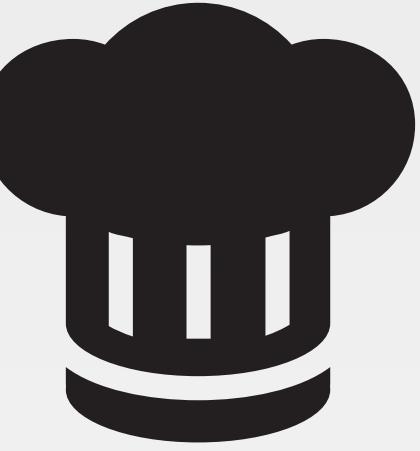

```

GL: What Does hello.txt Say?



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

```
Hello, world!
```



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



What would happen if you ran the command again?

What would happen if the file were removed?



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!

Re-apply the Policy Defined in the Recipe



```
C:\Users\Administrator> chef-apply hello.rb -l info
```

```
Starting Chef Client, version 13.3.42
```

```
...
```

```
Converging 1 resources
```

```
Recipe: @recipe_files::C:/Users/Administrator/hello.rb
```

```
* file[C:\hello.txt] action create
```

```
- update content in file C:\hello.txt from 2ffd26 to 315f5b
```

```
--- C:\hello.txt      2018-09-08 19:29:21.000000000 +0000
```

```
+++ C:\chef-hello20180908-3068-129bjez.txt 2018-09-08 19:30:18.000000000
```

```
+0000
```

```
@@ -1,2 +1,2 @@
```

```
-Goodbye, world!
```

```
+Hello, world!
```



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

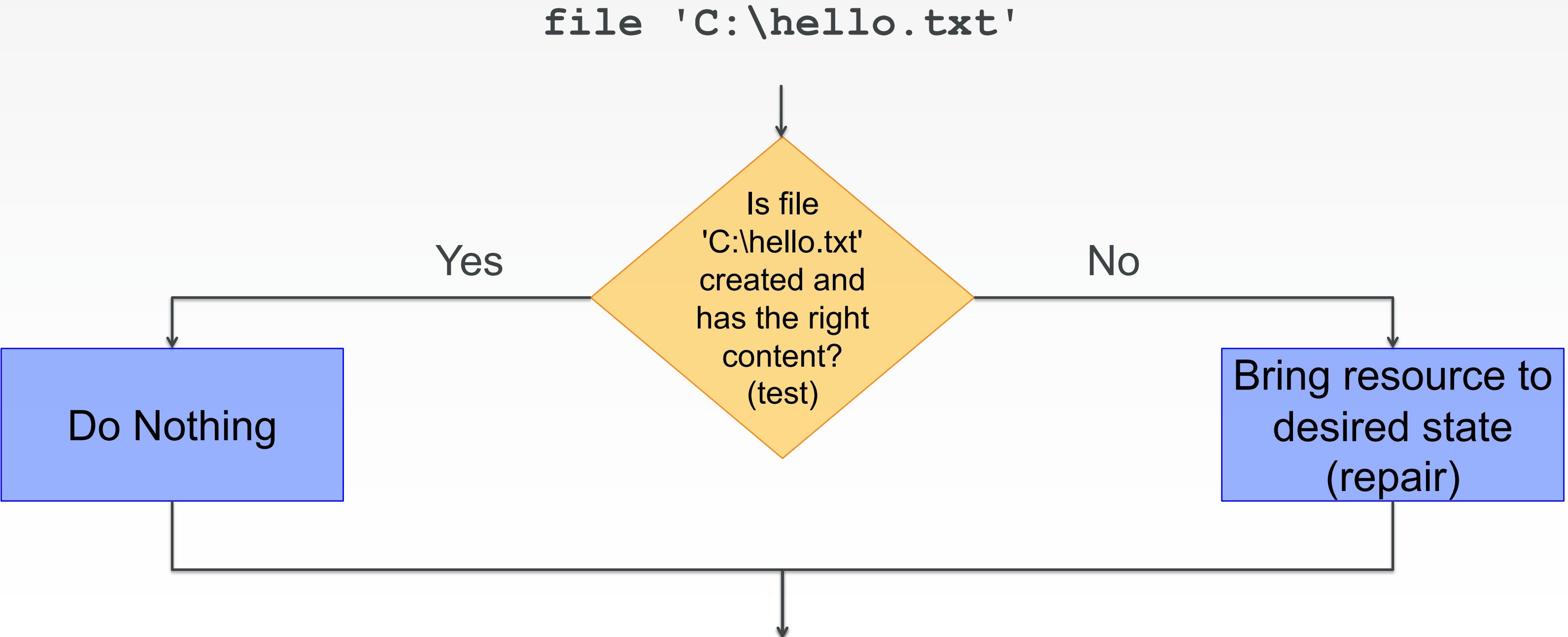
Test and Repair



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

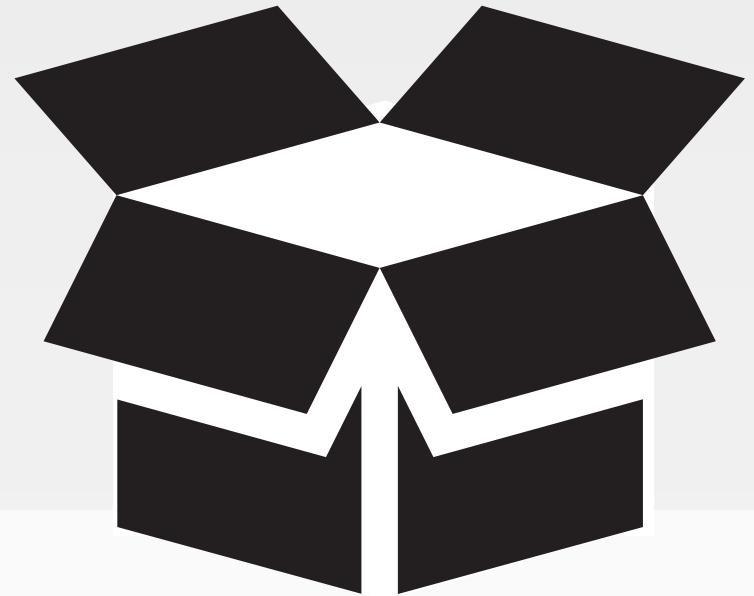
Chef looks at the current state of each resource
and takes action only when that resource is out of
policy.

Test and Repair



CONCEPT

Resource Definition

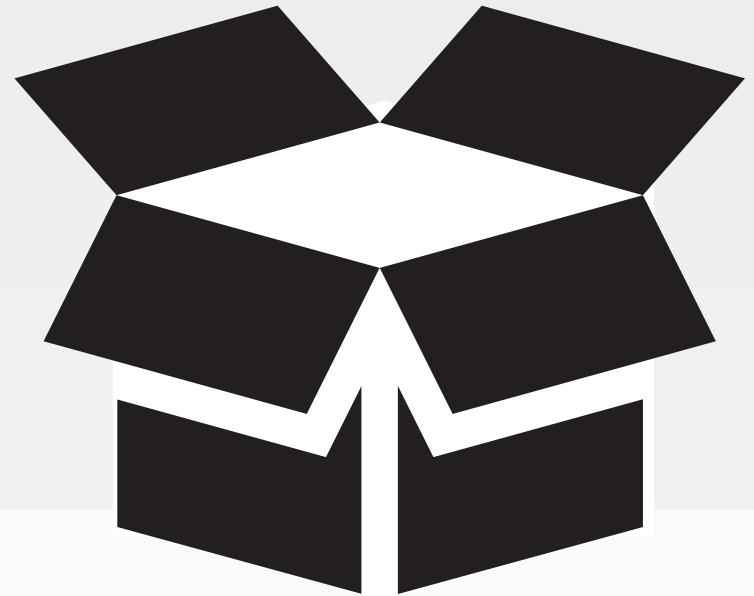


```
file 'C:\\hello.txt' do
  content 'Hello, world!'
end
```

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

CONCEPT

Resource Definition

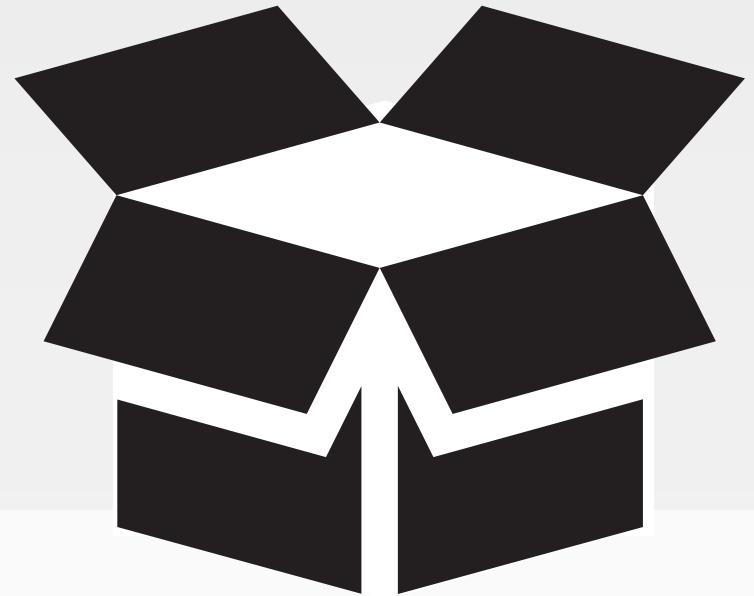


```
file 'C:\\hello.txt' do
  content 'Hello, world!'
end
```

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

CONCEPT

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**

CONCEPT

Resource Definition



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

?

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



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).



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?



Lab: Goodbye Recipe

- Create a recipe file named "goodbye.rb" that defines the policy:
The file named 'C:\hello.txt' is deleted.
- Use chef-client to apply the recipe file named "goodbye.rb"

Lab: The Updated file Resource

□ ~\goodbye.rb

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

Lab: Apply a Recipe File



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

```
C:/Users/Administrator.  
Starting Chef Client, version 13.3.42  
...  
Compiling Cookbooks...  
[2018-09-08T19:50:35+00:00] WARN: Node WIN-VBVUIACREJ.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  
Running handlers:  
Running handlers complete  
Chef Client finished, 1/1 resources updated in 04 seconds
```

Lab: Test that the File was Deleted



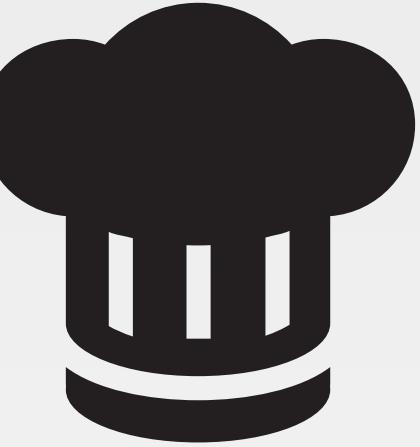
```
> Test-Path C:\hello.txt
```

```
False
```



Lab: Goodbye Recipe

- ✓ Create a recipe file named "goodbye.rb" that defines the policy:
The file named 'C:\hello.txt' is deleted.
- ✓ Use chef-client to apply the recipe file named "goodbye.rb"



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'

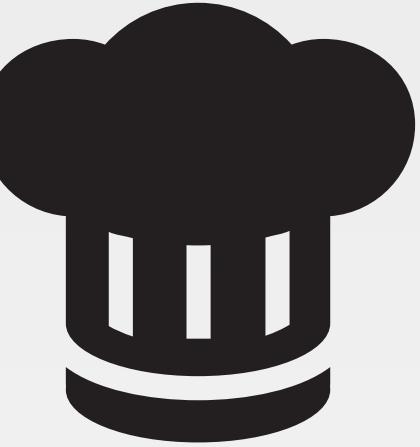
registry_key system_policies do
  values [ {
    :name => 'EnableLUA',
    :type => :dword,
    :data => 0
  }]
end
```

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

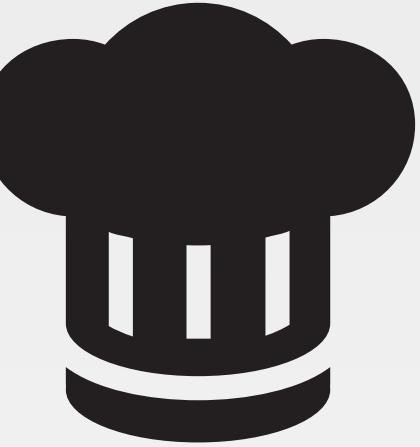
GL: Apply a Recipe File



```
> chef-apply disable-uac.rb -l info
```

```
Starting Chef Client, version 13.3.42
resolving cookbooks for run list: []
...
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 complete
Chef Client finished, 1/1 resources updated in 04 seconds
[2018-09-08T19:54:40+00:00] WARN: No config file found or specified on command
line, using command line
```

options



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



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"

GL: 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 ON THIS SLIDE TO CONSERVE SPACE)
end

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

Lab: Apply a Recipe File



```
> chef-apply disable-uac.rb -l info
```

```
Starting Chef Client, version 13.3.42
resolving cookbooks for run list: []
Synchronizing Cookbooks:
Installing Cookbook Gems:
Compiling Cookbooks...
[2018-09-08T20:41:05+00:00] WARN: Node WIN-VBVUIACREJ.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 }

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"

Let's Talk About Resources



Capture your answers because we're going to talk about them as a group.

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?

Q&A



What questions can we answer for you?

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



CHEF™