

# Tuning Ohai



## Objectives

After completing this module, you should be able to:

- Describe how you configure the node to automatically load ohai plugins
- Describe how to enable ohai hints
- Describe how to remove plugins that you do not want executed
- Describe where you can find more information about better Ohai performance

After completing this module you will be able to

# EXERCISE



## Run Ohai Smoother

*There are a few more things to learn about Ohai that could help increase performance.*

### Objective:

- ☐ Configure the node to automatically load ohai plugins
- ☐ Enable ohai hints
- ☐ Remove plugins that you do not want executed
- ☐ Choose only the plugins you want executed

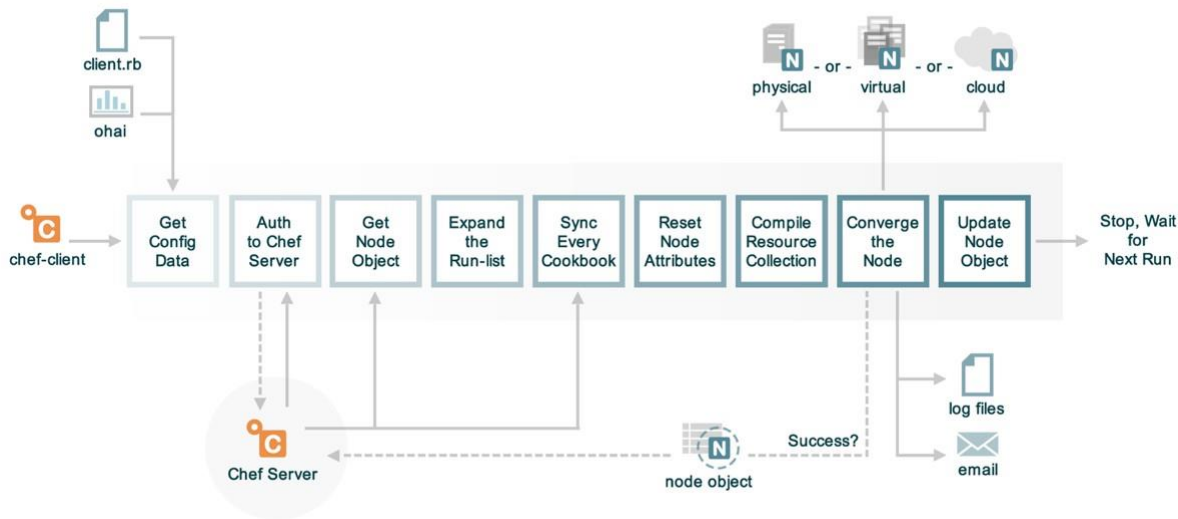
# CONCEPT



## Node Configuration File

All nodes have a configuration file that contain details about node, overrides, and other data.

# The Anatomy of a chef-client Run



# CONCEPT



## chef-client cookbook

The chef-client cookbook contains a number of useful recipes:

- **service (default)**  
Run chef-client as a service, converging at a periodic interval
- **delete\_validation**  
Remove the organization validation key [SECURITY ISSUE]
- **config**  
Define node configuration

<https://supermarket.chef.io/cookbooks/chef-client>

## Viewing the chef-client config Recipe



~/cookbooks/chef-client/recipes/config.rb

```
# ... OTHER RESOURCES ...
template "#{node["chef_client"]["conf_dir"]}/client.rb" do
  source 'client.rb.erb'
  owner d_owner
  group d_group
  mode 00644
  variables(
    :chef_config => node['chef_client']['config'],
    :chef_requires => chef_requires,
    :ohai_disabled_plugins => node['ohai']['disabled_plugins'],
    :start_handlers => node['chef_client']['config']['start_handlers'],
    :report_handlers => node['chef_client']['config']['report_handlers'],
    :exception_handlers => node['chef_client']['config']['exception_handlers']
  )
  notifies :create, 'ruby_block[reload_client_config]', :immediately
end
```



```
~/cookbooks/chef-client/templates/default/client.rb.erb
```

```
<% if node.attribute?("ohai") && node["ohai"].attribute?("plugin_path") -%>  
  
Ohai::Config[:plugin_path] << "<%= node["ohai"]["plugin_path"] %>"  
  
<% end -%>  
  
<% unless @ohai_disabled_plugins.empty? -%>  
Ohai::Config[:disabled_plugins] = [<%= @ohai_disabled_plugins.map { |k| k...  
  
<% end -%>
```



# PROBLEM



## The Work

- Add the chef-client cookbook to your cookbook collection
- Provide attributes in a wrapper cookbook, role, or environment for:

```
node['ohai']['plugin_path']
```

- Add chef-client cookbook to every node's run list

# EXERCISE



## Run Ohai Smoother

*There are a few more things to learn about Ohai that could help increase performance.*

### Objective:

- ✓ Configure the node to automatically load ohai plugins
- ☐ Enable ohai hints
- ☐ Remove plugins that you do not want executed
- ☐ Choose only the plugins you want executed

# CONCEPT



## Not all Node Plugins are Executed

Some of the plugins will not automatically run against your node. This is particularly true of the cloud provider plugins. As the node does not often know the environment in which is being run.

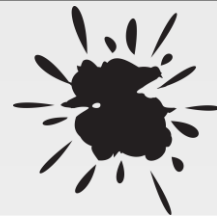
# CONCEPT



## Ohai Hints

For those plugins that are not executed you can leave them a hint file that will ensure they are executed.

# PROBLEM



## The Work

- Add the ohai cookbook to your cookbook collection
- Define a recipe that uses the ohai cookbook's ohai\_hint resource
- Add this recipe to every node's run list

# EXERCISE



## Run Ohai Smoother

*There are a few more things to learn about Ohai that could help increase performance.*

### Objective:

- ✓ Configure the node to automatically load ohai plugins
- ✓ Enable ohai hints
- ☐ Remove plugins that you do not want executed
- ☐ Choose only the plugins you want executed



~/cookbooks/chef-client/templates/default/client.rb.erb

```
<% if node.attribute?("ohai") && node["ohai"].attribute?("plugin_path") -%>

Ohai::Config[:plugin_path] << "<%= node["ohai"]["plugin_path"] %>"
<% end -%>

<% unless @ohai_disabled_plugins.empty? -%>
Ohai::Config[:disabled_plugins] = [<%= @ohai_disabled_plugins.map { |k| k...

<% end -%>
```

## Viewing the chef-client config Recipe

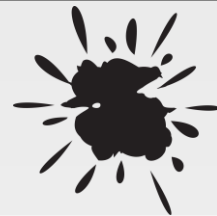


~/cookbooks/chef-client/recipes/config.rb

```
# ... OTHER RESOURCES ...
template "#{node["chef_client"]["conf_dir"]}/client.rb" do
  source 'client.rb.erb'
  owner d_owner
  group d_group
  mode 00644
  variables(
    :chef_config => node['chef_client']['config'],
    :chef_requires => chef_requires,
    :ohai_disabled_plugins => node['ohai']['disabled_plugins'],
    :start_handlers => node['chef_client']['config']['start_handlers'],
    :report_handlers => node['chef_client']['config']['report_handlers'],
    :exception_handlers => node['chef_client']['config']['exception_handlers']
  )
  notifies :create, 'ruby_block[reload_client_config]', :immediately
end
```



# PROBLEM



## The Work

- Add the chef-client cookbook to your cookbook collection
- Select attributes you want to remove
- Find the name of the plugin that provides those attributes
- Provide attributes in a wrapper cookbook, role, or environment for:

```
node['ohai']['disabled_plugins']
```

- Add chef-client cookbook to every node's run list

# EXERCISE



## Run Ohai Smoother

*There are a few more things to learn about Ohai that could help increase performance.*

### Objective:

- ✓ Configure the node to automatically load ohai plugins
- ✓ Enable ohai hints
- ✓ Remove plugins that you do not want executed
- ☐ Choose only the plugins you want executed

# CONCEPT



## Whitelist Node Attributes

When it seems like you are disabling far too many plugins and you want to consider attacking it from the other side:

<http://ckbk.it/whitelist-node-attrs>

# EXERCISE



## Run Ohai Smoother

*There are a few more things to learn about Ohai that could help increase performance.*

### Objective:

- ✓ Configure the node to automatically load ohai plugins
- ✓ Enable ohai hints
- ✓ Remove plugins that you do not want executed
- ✓ Choose only the plugins you want executed

# DISCUSSION



## Q&A

What questions can we answer for you?

