



CHEF™

Getting Started with Compliance Automation



CHEFTM

Running Scans

InSpec: Turn security and compliance into code

- Translate compliance into Code
- Clearly express statements of policy
- Move risk to build/test from runtime
- Find issues early
- Write code quickly
- Run code anywhere
- Inspect machines, data and APIs

Part of a process of continuous compliance



A simple example of an InSpec CIS rule

```
control 'cis-1.4.1' do
  title '1.4.1 Enable SELinux in /etc/grub.conf'
  desc '
    Do not disable SELinux and enforcing in your GRUB configuration.
    These are important security features that prevent attackers from
    escalating their access to your systems. For reference see ...
  '
  impact 1.0
  expect(grub_conf.param 'selinux').to_not eq '0'
  expect(grub_conf.param 'enforcing').to_not eq '0'
end
```

Objectives

After completing this module, you should be able to:

- Add a node to test for compliance.
- Run a Compliance scan.
- Test for compliance with InSpec from the command line interface.

CONCEPT



Adding a Node to Scan

To add a node you'll need:

- The IP address or FQDN of the nodes to be tested.
- Access configuration (ssh or WinRM).
- The node's username and password OR
- The node's username plus security key pair.

EXERCISE



Group Lab: Adding a Node to Scan

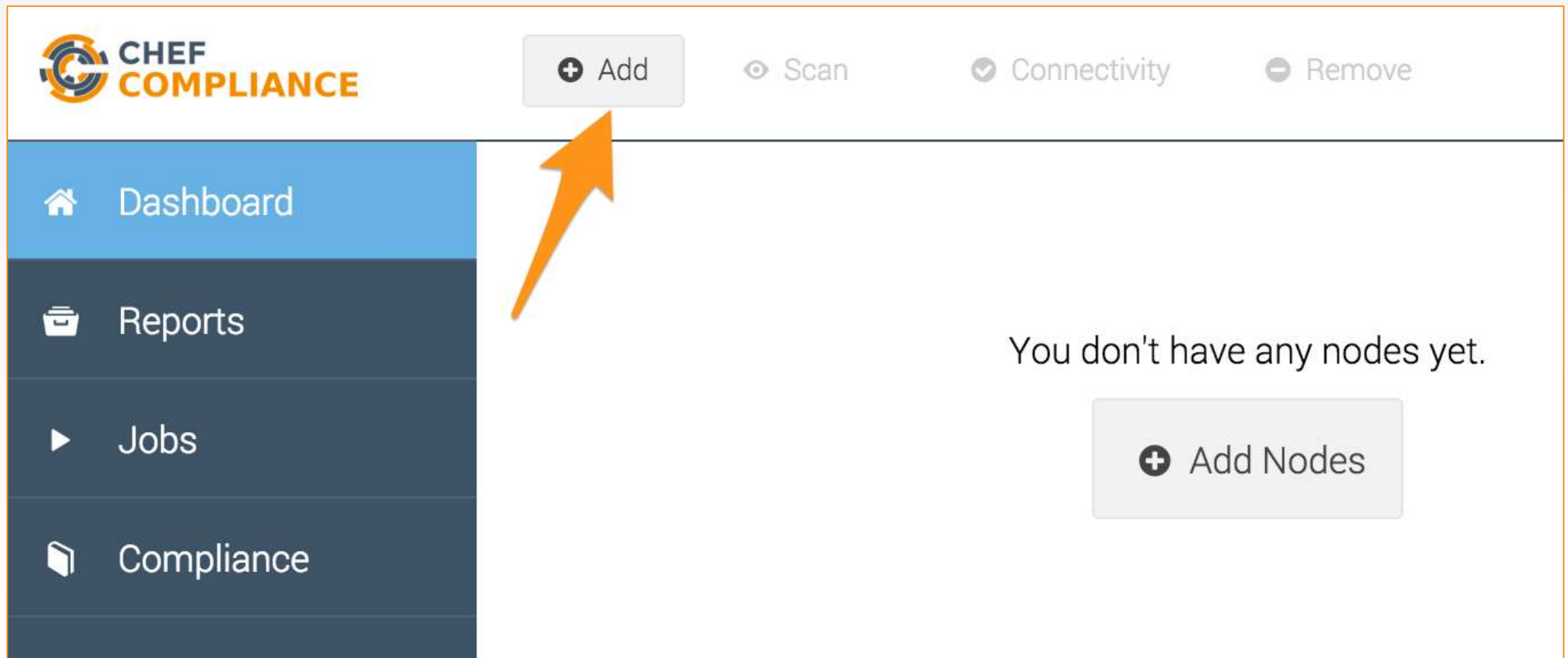
Objective:

- ☐ Add a Linux Node to Scan
- ☐ Test connectivity

Note: In the next module you will perform the same exercises as in this module but using a Windows node as your target node.

GL: Adding a Node to Scan

1. From your Chef Compliance Dashboard, click Add Node.



GL: Adding a Node

2. From the resulting page, enter the node's FQDN or IP address.
3. Leave environment blank. A 'default' environment will be used
4. Accept the default **SSH** Access configuration
5. Type **chef** in the **username** field.
6. Click the **password** link as shown in this illustration.

Dashboard / Add nodes

Enter nodes (IPs or hostnames):

ec2-52-91-159-53.compute-1.amazonaws.com x Add your nodes via IP or hostname

Add to environment:

default

Access configuration:

☒ SSH ☐ WinRM

Username

chef

Use Key Pair:

Select a login key

Add new key pairs or use login with password instead.

Sudo Configuration:

☐ Disable sudo

Optional sudo password *

GL: Adding a Node to Scan

7. Type the password (**chef**) in the password field.
8. Click the **Add 1 node** button as shown in this illustration.

Add to environment:

default

Access configuration:

SSH

WinRM

Username

chef

Password-based login is generally not recommended and should be limited to development and legacy systems. Make sure you have a sufficiently complex password configured.

....

Use login with **public key** instead.

Sudo Configuration:

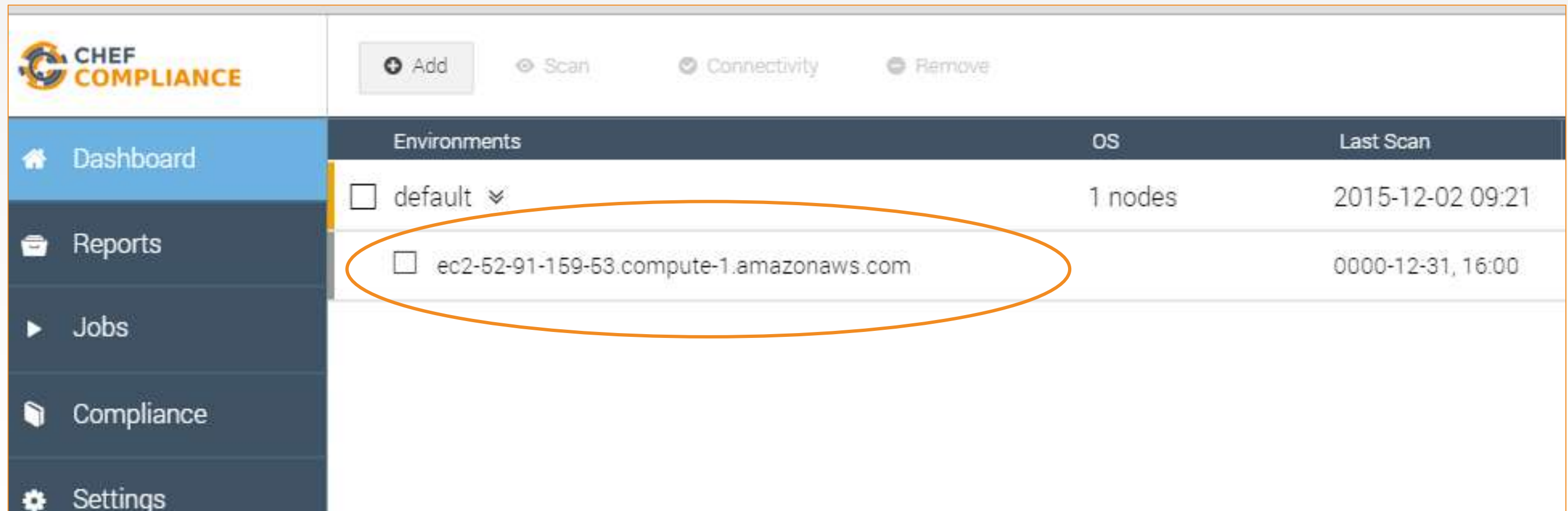
☐ Disable sudo

Optional sudo password *

Add 1 node

GL: Adding a Node to Scan

At this point your Compliance Dashboard should list the node you just added.

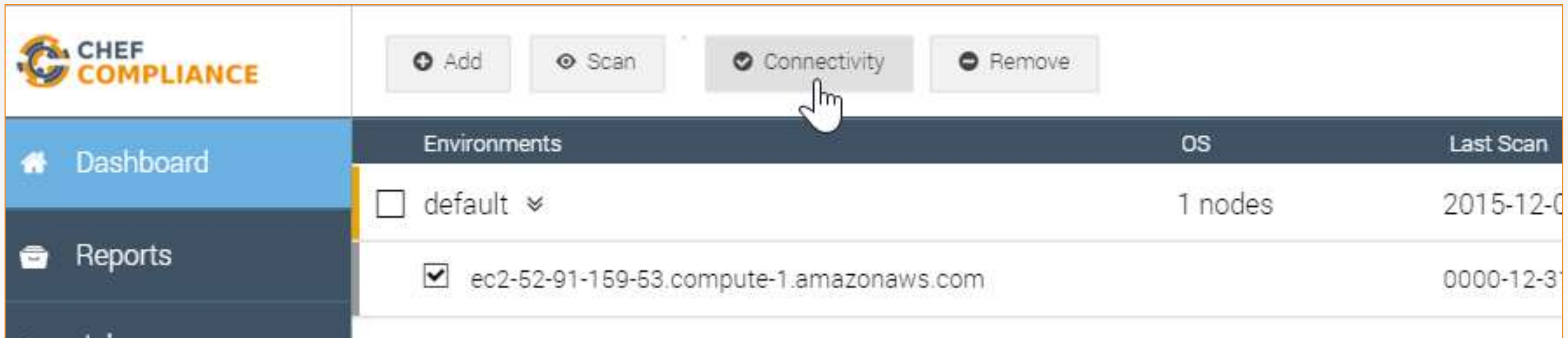


The screenshot shows the Chef Compliance dashboard. On the left is a sidebar with navigation links: Dashboard, Reports, Jobs, Compliance, and Settings. The main area displays a table with columns: Environments, OS, and Last Scan. The table lists two environments: 'default' and 'ec2-52-91-159-53.compute-1.amazonaws.com'. The second environment is circled in orange. Above the table are buttons for 'Add', 'Scan', 'Connectivity', and 'Remove'.

Environments	OS	Last Scan
<input type="checkbox"/> default	1 nodes	2015-12-02 09:21
<input type="checkbox"/> ec2-52-91-159-53.compute-1.amazonaws.com		0000-12-31, 16:00

GL: Testing Connectivity to Your Node

1. Click the **check box** next to your node and then click the **Connectivity** button.

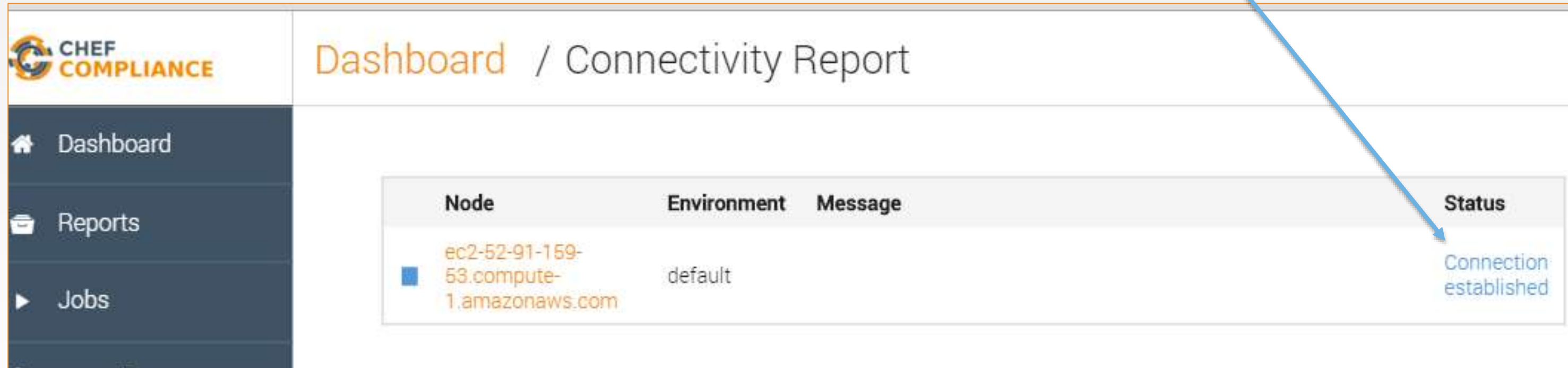


The screenshot shows the Chef Compliance web interface. On the left is a sidebar with the 'CHEF COMPLIANCE' logo and navigation links for 'Dashboard' and 'Reports'. The main content area has a top bar with buttons: '+ Add', 'Scan', 'Connectivity' (which is highlighted with a mouse cursor), and '- Remove'. Below this is a table with columns 'Environments', 'OS', and 'Last Scan'. The table contains two rows: one for the 'default' environment with 1 node and a scan date of 2015-12-0, and another for a specific node 'ec2-52-91-159-53.compute-1.amazonaws.com' with a scan date of 0000-12-3.


Environments	OS	Last Scan
<input type="checkbox"/> default ▾	1 nodes	2015-12-0
<input checked="" type="checkbox"/> ec2-52-91-159-53.compute-1.amazonaws.com		0000-12-3

GL: Testing Connectivity to Your Node

The Status column of you node should now indicate **Connection established**.



The screenshot shows the Chef Compliance dashboard with a sidebar on the left containing 'Dashboard', 'Reports', and 'Jobs'. The main content area is titled 'Dashboard / Connectivity Report'. It features a table with the following data:

Node	Environment	Message	Status
 ec2-52-91-159-53.compute-1.amazonaws.com	default		Connection established

A blue arrow points from the text 'Connection established' in the original image to the 'Status' column header in the table.

Adding Nodes in Bulk

You could add additional nodes by simply repeating the previous steps.

You could also add a number of nodes at once by separating each hostname or IP address with a comma or a space, as shown in this illustration.

Chef Compliance also supports bulk loading of nodes via API.

Enter nodes (IPs or hostnames):

ec2-52-23-162-169.compute-1.amazonaws.com x ec2-52-91-130-201.compute-1.amazonaws.com x

Add your nodes via IP or hostname

Add to environment:

default

Access configuration:

 SSH  WinRM

Username

chef

Password-based login is generally not recommended and should be limited to development and legacy systems. Make sure you have a sufficiently complex password configured.

....

Use login with **public key** instead.

Sudo Configuration:

☐ Disable sudo

Optional sudo password *

Add 2 nodes

Adding Nodes in Bulk via API

After class you can go to the following link.

The resulting kitchen_sink.rb will step you through how to upload nodes in bulk.

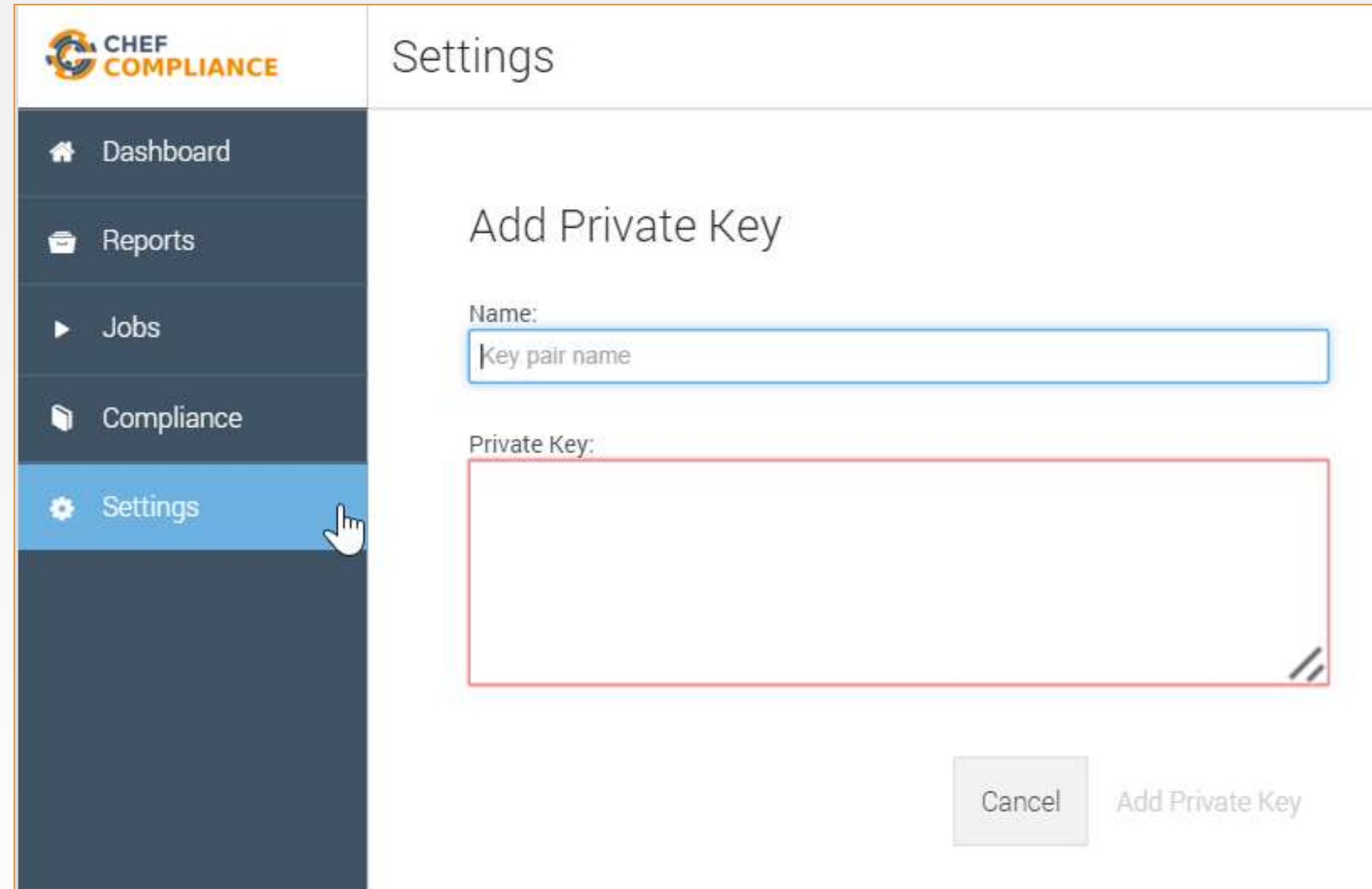
```
1  ### Script to export Chef Server nodes and import them to Chef Compliance
2  ### Change the 'api_url', 'api_user' and 'api_pass' variables below
3  ### Go to your chef-repo and check Chef Server access first
4  # cd chef-repo; knife environment list
5  ### Save this Ruby script as kitchen_sink.rb and run it like this:
6  # cat kitchen_sink.rb | knife exec
7  ### Chef Compliance API docs: https://docs.chef.io/api_compliance.html
8
9  require 'json'
10 require 'uri'
11 require 'net/http'
12 require 'openssl'
13
14 # This extracts data from the Chef Server. Auth done by `knife exec`
15 # Change loginKey and any other details that will be posted to the Chef Compliance API:
16 nodes_array = []
17 nodes.find('*:*') { |n|
18   nodes_array << { id: n.name,
19                     name: n.name,
```

<https://gist.github.com/alexp0p/01b0bba8d259adeeee320>

Private Keys

In the workplace, using a security key would be a more secure method for connecting to nodes than using the password method.

By clicking **Settings > Add Private Key** you will see where to paste a private key.



The screenshot shows the Chef Compliance web interface. On the left is a dark blue sidebar with navigation links: Dashboard, Reports, Jobs, Compliance, and Settings (highlighted with a gear icon and a mouse cursor). The main content area is titled 'Settings' and contains a section 'Add Private Key'. This section has two input fields: 'Name:' with a text box containing 'Key pair name', and 'Private Key:' with a larger text area. At the bottom right of the form are two buttons: 'Cancel' and 'Add Private Key'.

CONCEPT



Running Compliance Scans

You can run Compliance scans on demand or schedule them to run at a later time.

Chef Compliance maintains profiles as a collection of individual controls that comprise a complete audit.

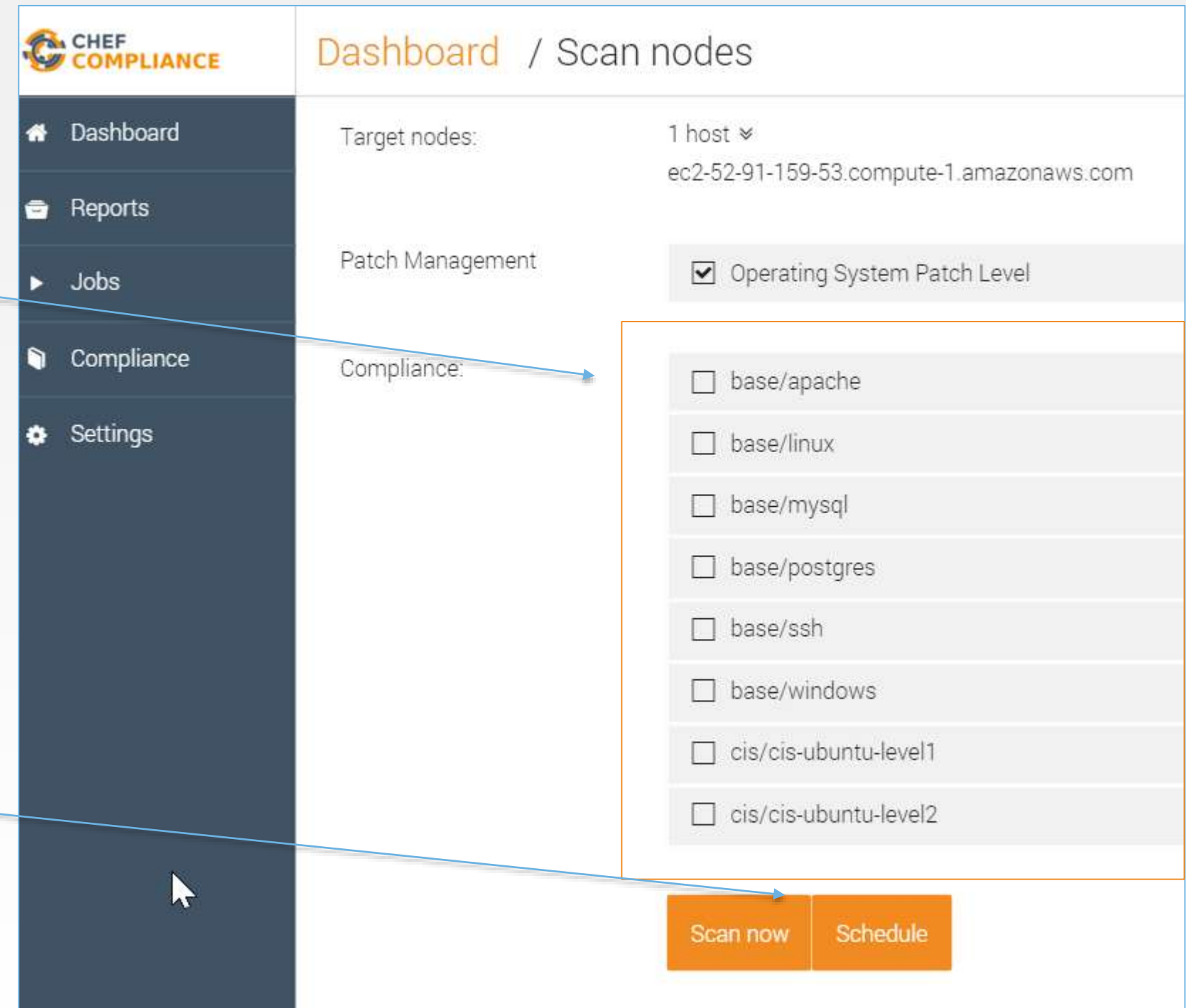
As mentioned previously, Chef Compliance comes with a few reference profiles of various compliance policies that you can leverage or use as examples to create your own.

Compliance Profiles Used in Scans

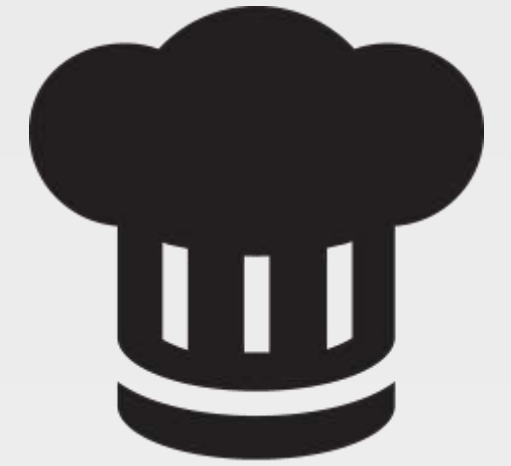
This image shows the default Compliance Profiles as accessed from the Scan Nodes page.

You should be thoughtful with which profiles choose.

Notice how you can also choose to run a scan on demand or schedule a scan.



EXERCISE



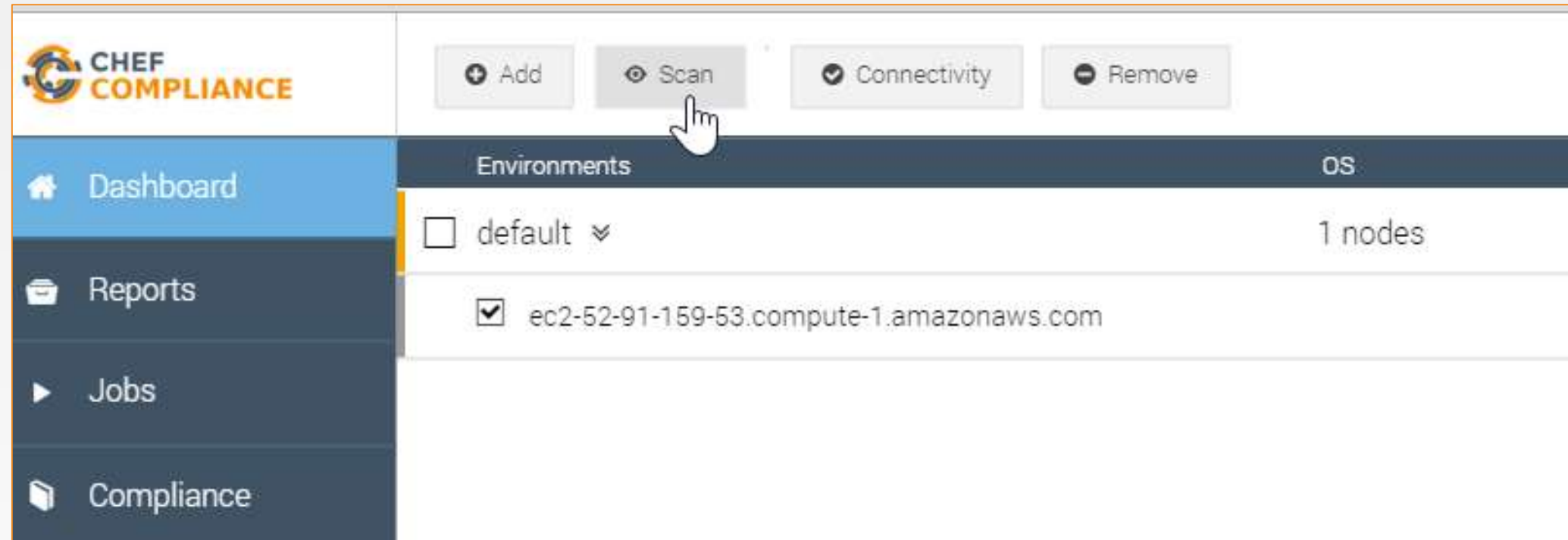
Group Lab: Running a Scan

Objective:

- ☐ Run a Compliance scan.
- ☐ View the output of a scan.

GL: Running a Scan

1. Click the **check box** next to your node and then click the **Scan** button.



GL: Running a Scan

2. From the resulting page, check the **base/ssh** profile and uncheck any other check boxes.
3. Click the **Scan now** button.

Dashboard / Scan nodes

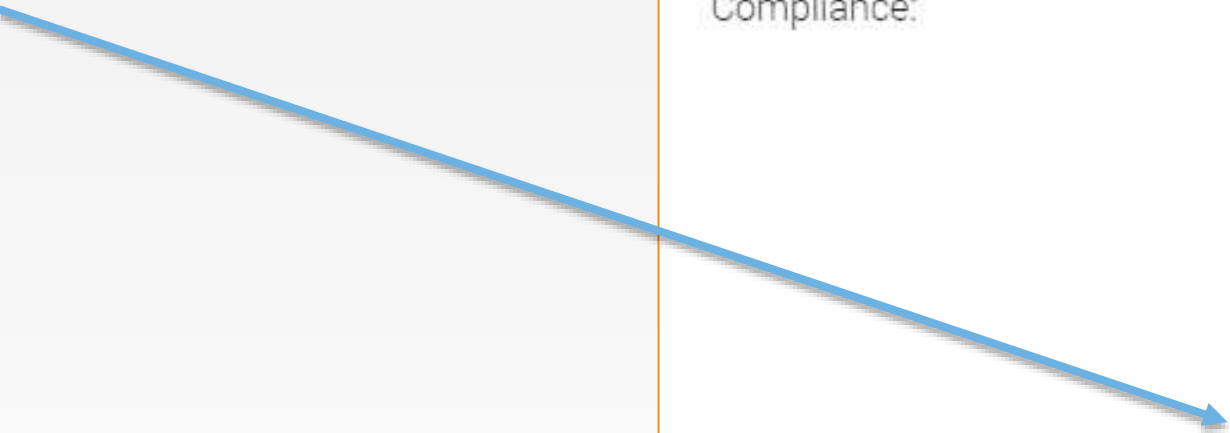
Target nodes: 1 host ▾
ec2-52-91-159-53.compute-1.amazonaws.com

Patch Management ☐ Operating System Patch Level

Compliance:

- ☐ base/apache
- ☐ base/linux
- ☐ base/mysql
- ☐ base/postgres
- ☒ base/ssh
- ☐ base/windows
- ☐ cis/cis-ubuntu-level1
- ☐ cis/cis-ubuntu-level2

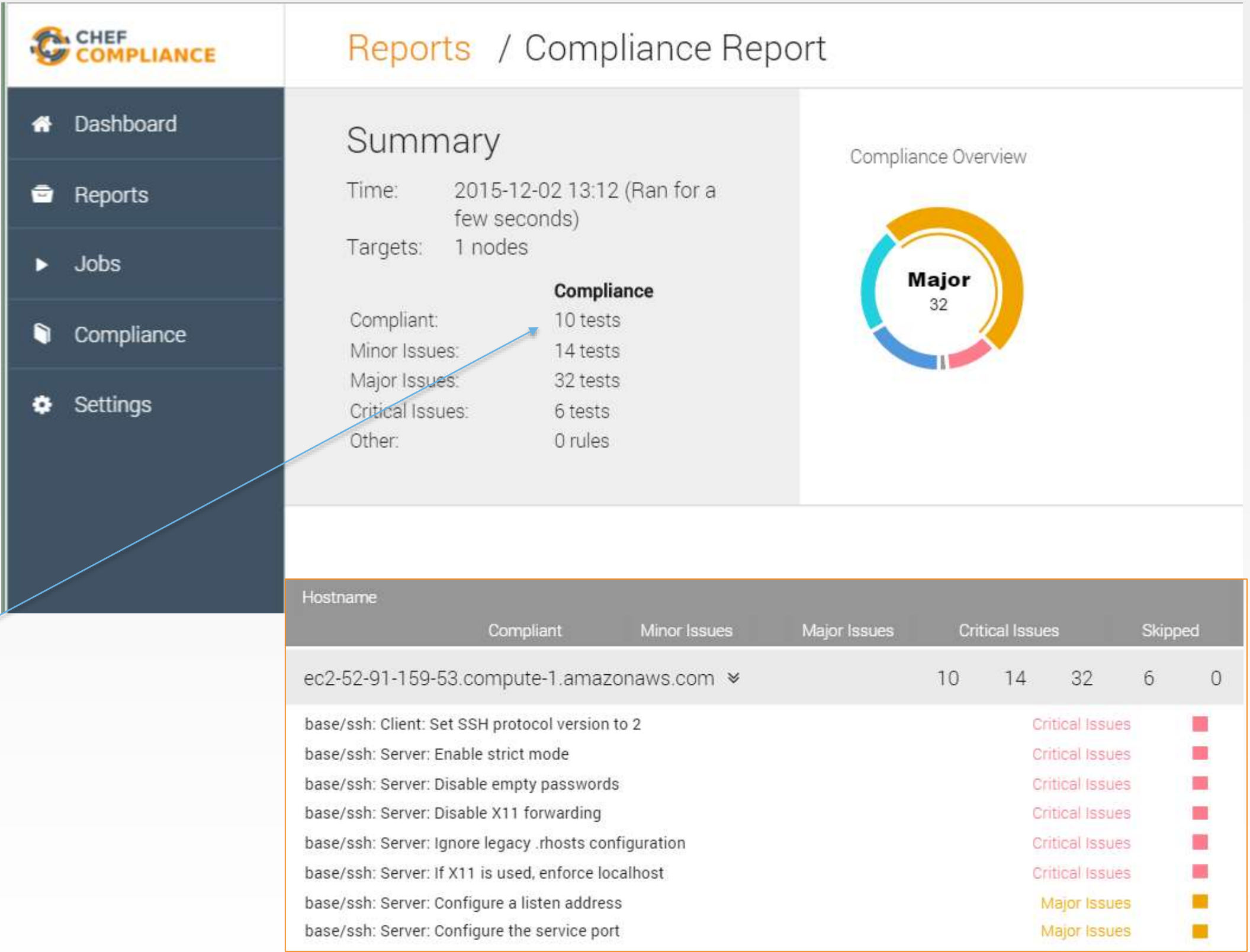
Scan now **Schedule**



Scan Results

A Compliance Report should now display and your scan results should be similar to that shown here.

Notice how in the upper Summary section in this example, 10 tests were compliant and 6 tests show critical issues with ssh.



Scan Results

The bottom half of the Compliance Report shown here has a table of details of test results.

These are sorted by severity.

If you click an issue as shown here, a bit more information about the issue displays.

Hostname	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
ec2-52-91-159-53.compute-1.amazonaws.com	10	14	32	6	0
base/ssh: Client: Set SSH protocol version to 2					
SSH Configuration Protocol should eq "2"			Critical Issues	10.0	
base/ssh: Server: Enable strict mode			Critical Issues		
base/ssh: Server: Disable empty passwords			Critical Issues		
base/ssh: Server: Disable X11 forwarding			Critical Issues		
base/ssh: Server: Ignore legacy .rhosts configuration			Critical Issues		
base/ssh: Server: If X11 is used, enforce localhost			Critical Issues		
base/ssh: Server: Configure a listen address			Major Issues		
base/ssh: Server: Configure the service port			Major Issues		
base/ssh: /etc/ssh should have limited access to 0755			Major Issues		

GL: Profile

To view the InSpec code that comprises this profile, do the following:

1. Click the **Compliance** button.
2. Click the relevant profile (**Basic SSH**).
3. Scroll down and click the **`Set SSH protocol version to 2`** profile.

The screenshot illustrates the steps to view the InSpec code for a specific compliance profile in the Chef Compliance interface. It shows the 'Reports / Compliance Report' page with a sidebar menu where 'Compliance' is selected. A secondary menu shows the 'Compliance profiles' list with 'Basic SSH' selected. The main content area displays the details for the 'Basic SSH' profile, including a list of client commands and the corresponding InSpec code.

CHEF COMPLIANCE Reports / Compliance Report

Summary

Time: 2
Targets: 1

Compliant:
Minor Issues:
Major Issues:
Critical Issues:
Other:

Compliance profiles

Name
Basic Apache 2
Basic Linux
Basic MySQL
Basic PostgreSQL
Basic SSH

Client: Do not permit local commands

Client: Configure expected port

Client: Set SSH protocol version to 2

Set the SSH protocol version to 2. Don't use legacy insecure SSHv1 connections anymore.

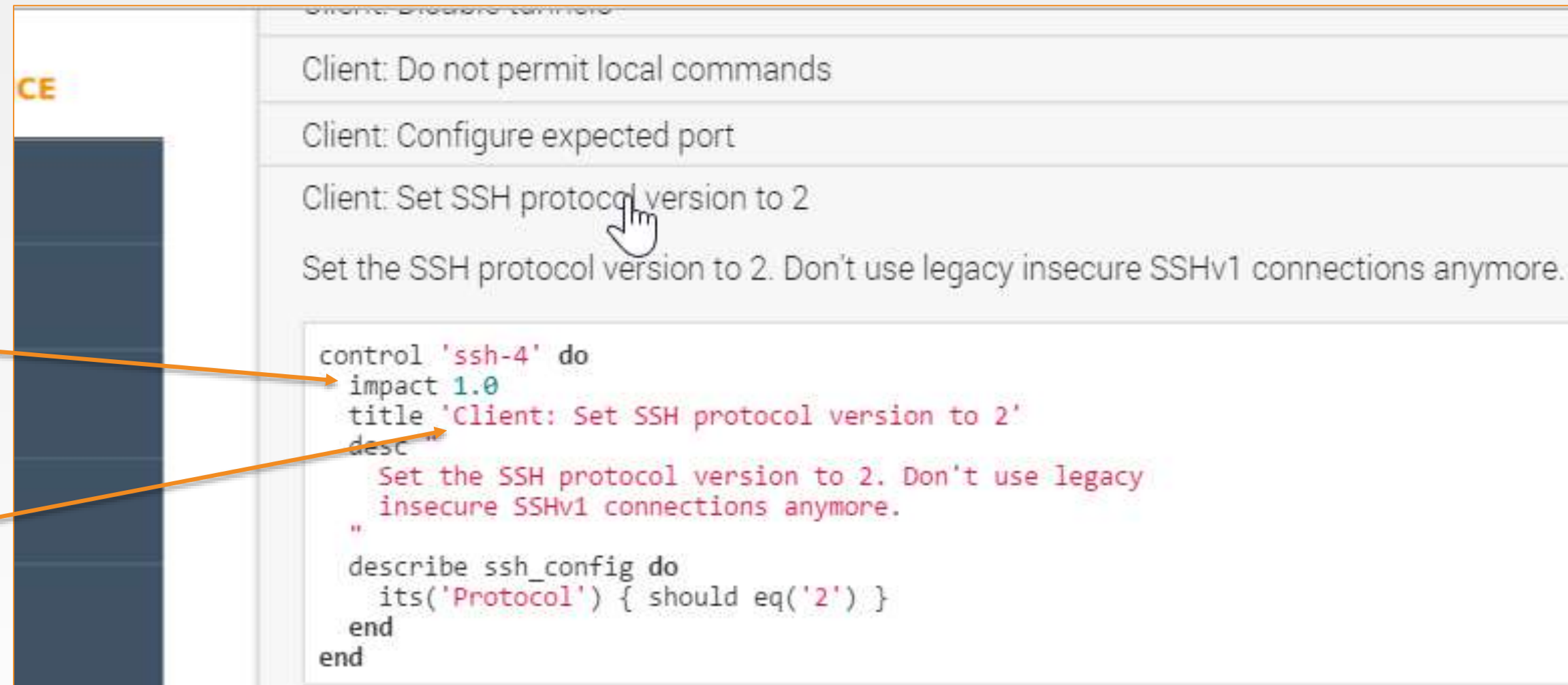
```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

Discussion: InSpec Profile Code

Let's discuss what this profile is doing.

The `impact` of 1.0 indicates this is a Critical issue.

The `title` is what populates the Compliance Report issue title.

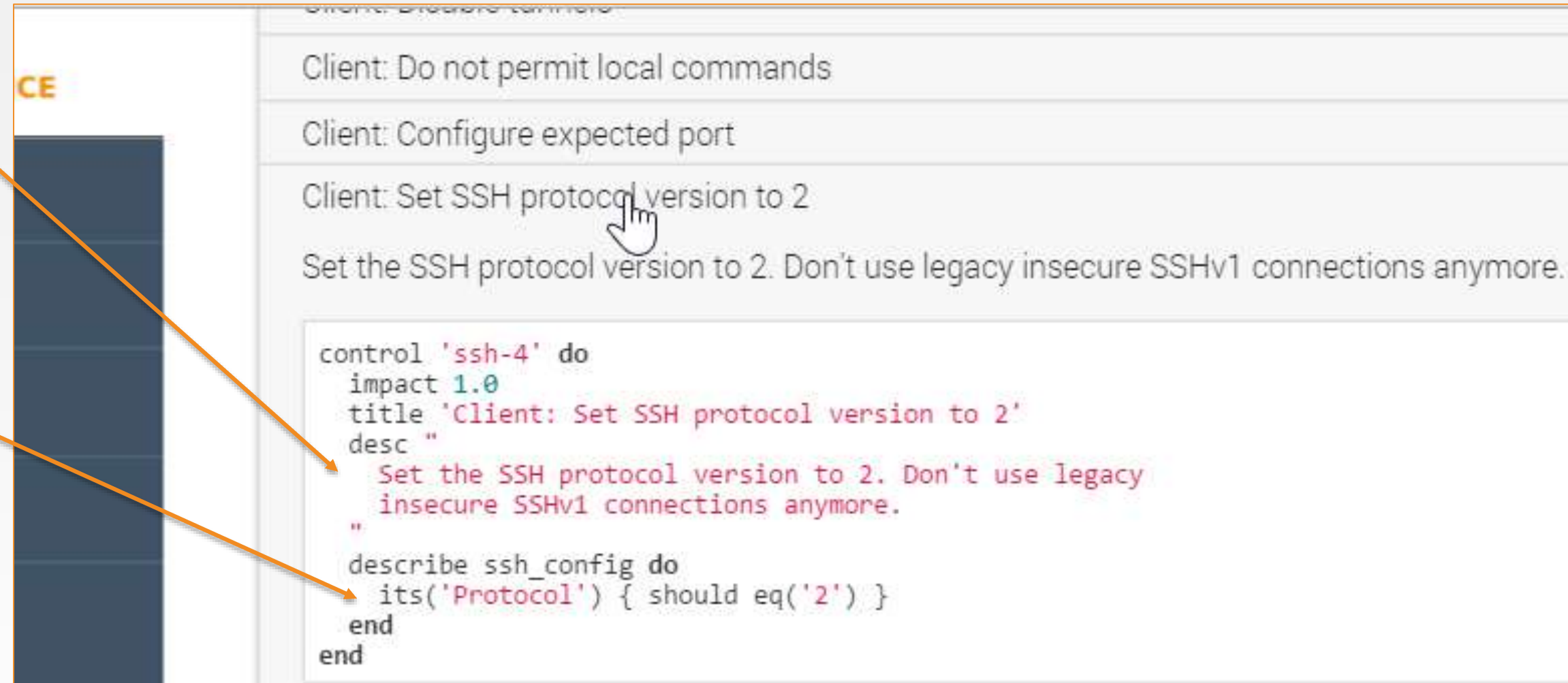
A screenshot of an InSpec profile code editor. On the left, a dark blue vertical bar represents the compliance report, with a yellow 'CE' (Critical Error) indicator at the top. Two orange arrows point from text on the left to the code: one points to the 'impact 1.0' line, and the other points to the 'title' line. The code itself is in a light gray box with a white border. It contains a control named 'ssh-4' with an impact of 1.0, a title, a description, and a describe block for 'ssh_config' that checks if the protocol is '2'.

```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy insecure SSHv1 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```


Discussion: InSpec Profile Code

The **desc** is typically human-readable description sourced from the CIS or source doc.

The **`describe`** section is the actual test that is executed.

A screenshot of an InSpec profile code editor. On the left, a dark blue sidebar contains a list of profile sections, with 'CE' highlighted at the top. Two orange arrows originate from the text on the left: one points to the 'desc' line in the code, and the other points to the 'describe' block. The main editor area shows the following code:

```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy insecure SSHv1 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

At the top of the editor, there is a list of client commands: 'Client: Do not permit local commands', 'Client: Configure expected port', and 'Client: Set SSH protocol version to 2'. A mouse cursor is hovering over the third command. Below this list, a summary text reads: 'Set the SSH protocol version to 2. Don't use legacy insecure SSHv1 connections anymore.'

Compliance Profile Severity Mapping

The table below shows the current mapping of Compliance Profile **impact** numbering to severity.

Set the SSH protocol version to 2. Don't use legacy insecure S

```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

Impact Numbering	Severity Designation
0.7 - 1.0	Critical Issues
0.4 - <0.7	Major Issues
0 - <0.4	Minor Issues

Critical Issues

Critical Issues

Critical Issues

Major Issues

Major Issues

Major Issues

Minor Issues

Minor Issues

<https://nvd.nist.gov/cvss.cfm>

Run InSpec from the Command Line

```
control 'sshd-11' do
  impact 1.0
  title 'Server: Set protocol version to SSHv2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore.
  "
  describe sshd_config do
    its('Protocol') { should eq('2') }
  end
end
```

Run InSpec from the Command Line

Test Locally:

```
$ inspec exec test.rb
```

Run InSpec from the Command Line

Remote via SSH:

```
$ inspec exec test.rb -t ssh://54.163.150.246 --user=chef --  
password=chef.io
```

Run InSpec from the Command Line

Docker Container

```
$ inspec exec test.rb -t docker://3dda08e75838
```



CHEF™

Running Scans