

# Introduction to Continuous Compliance and Remediation

Nathen Harvey

Chef

@nathenharvey

O'REILLY®

Velocity

# Introductions

- Who are you?
- Why are you here?
- What was the last show you binge watched?

# Step one: Detect

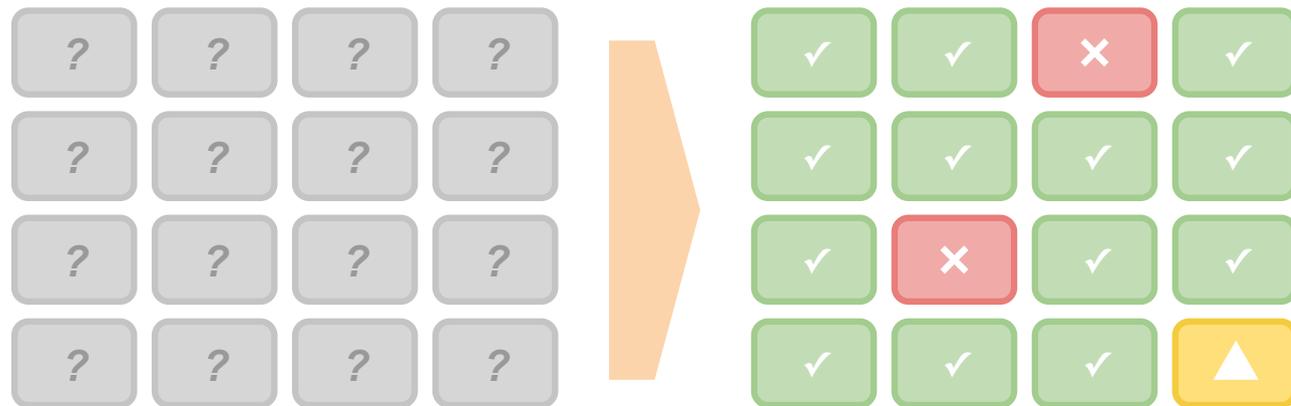
Gain visibility into current status to satisfy audits and drive decision-making

55%

of organizations do compliance assessments inconsistently or not at all.



Apply policies and gain a complete view across the fleet



- Accurately assess risk
- Prioritize remediation actions
- Maintain audit readiness
- Create and adjust policies



Continuous visibility means that you enter into audits knowing the outcome.

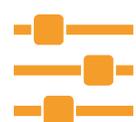
Jon Williams, NIU

# Step two: Correct

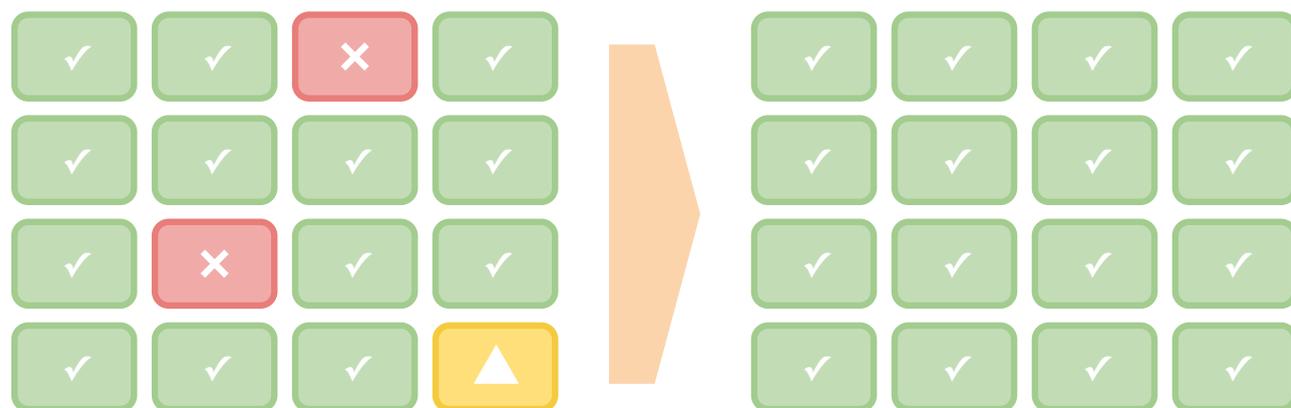
Remediate issues to improve performance and security

58%

of organizations need days or longer to remediate issues.



Develop, test, and deploy remediation to address issues across the fleet



- Prioritize actions based on impact
- Improve application performance
- Close security holes
- Prove policy compliance

# A tale of three personas...



```
exploit(basilic_diff_exe)
exploit(phptax_exec) >
exploit(phptax_exec) >
exploit(phptax_exec) >
exploit(phptax_exec) > exploit

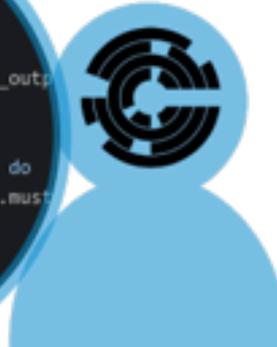
[*] Started reverse double handler
[*] 192.168.2.15580 - Sending request
[*] Accepted the first client connection
[*] Accepted the second client connection
[*] Command: echo 10XxRdYFI9zPXPrp\r\n
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] 10XxRdYFI9zPXPrp\r\n
[*] Reading...
```

Security



```
... they exist, from being u...
... the user partitions are not intended to support de...
... cannot attempt to create block or character special...
Note: The actions in the item refer to the /home partition, w...
partition that is defined in CentOS 6. If you have created othe...
recommended that the Remediation and Audit steps be appli...
audit:
... in the following commands to determine if the system is c...
# grep /etc/fstab | grep noexec
# grep /etc | grep noexec
... command emits no output then the system is r...
...
... file and add noexec *
```

Compliance



```
... side os resource
print os[:family]').must

'must provide file resource' do
load('print file("").type').must_outp
end

t 'must provide command resource' do
load('print command("").stdout').must
d

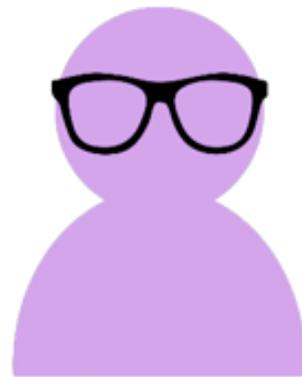
... reports empty describe calls'
describe').must_output
... keys, len...
```

DevOps

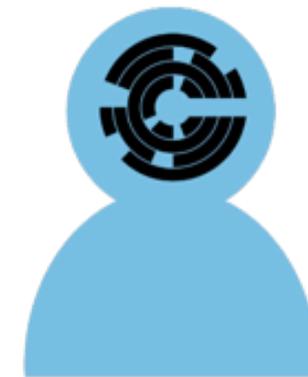
... and a single language.



Security



Compliance



DevOps

# From lemons...

```
$ grep "^key" /etc/tac_plus/tac_plus.conf | sed 's/key = //'
s00persecretkey
$
```

# ... create lemonade!

```
control 'sox-404.3.5' do
  title 'Network Device to Central Auth Encryption'
  impact 1.0
  desc "
    All communication between network devices and central auth
    must be encrypted. Our TACACS+ servers encrypt all the time
    and the presence of a pre-shared key proves it."

  describe ini('/etc/tac_plus/tac_plus.conf') do
    its('key') { should_not be_nil }
  end
end
```

# Map Documentation to Controls

**404.3.5:**  
*Communication between network devices and central authentication systems must be encrypted at all times.*

```
control 'sox-404.3.5' do
  title 'Network Device to Central Auth Encryption'
  impact 1.0
  desc "
    All communication between network devices and
    central auth must be encrypted. Our TACACS+ servers
    encrypt all the time and the presence of a
    pre-shared key proves it."

  describe ini('/etc/tac_plus/tac_plus.conf') do
    its('key') { should_not be_nil }
  end
end
```

# Share Context

**404.3.5:**  
*Communication between network devices and central authentication systems must be encrypted at all times.*

```
control 'sox-404.3.5' do
  title 'Network Device to Central Auth Encryption'
  impact 1.0
  desc "
    All communication between network devices and
    central auth must be encrypted. Our TACACS+ servers
    encrypt all the time and the presence of a
    pre-shared key proves it."

  describe ini('/etc/tac_plus/tac_plus.conf') do
    its('key') { should_not be_nil }
  end
end
```

# Automate Test Execution

**404.3.5:**  
*Communication between network devices and central authentication systems must be encrypted at all times.*

```
control 'sox-404.3.5' do
  title 'Network Device to Central Auth Encryption'
  impact 1.0
  desc "
    All communication between network devices and
    central auth must be encrypted. Our TACACS+ servers
    encrypt all the time and the presence of a
    pre-shared key proves it."

  describe ini('/etc/tac_plus/tac_plus.conf') do
    its('key') { should_not be_nil }
  end
end
```

# Today's Workshop

- Detect a compliance failure with InSpec and Chef Automate
- Create a Chef cookbook to remediate the failure
- Test the cookbook with Test Kitchen
- Remediate the failure with the new cookbook
- Validate our remediation in Chef Automate

# Learning Environment

**CHEF**  
AUTOMATE

# Learning Environment

**CHEF**  
AUTOMATE



Nodes

# Learning Environment

**CHEF**  
AUTOMATE

Node data



Nodes

# Learning Environment

**CHEF**  
AUTOMATE

Node data



Laptop



Nodes

# Learning Environment

# CHEF AUTOMATE

Node data



Laptop



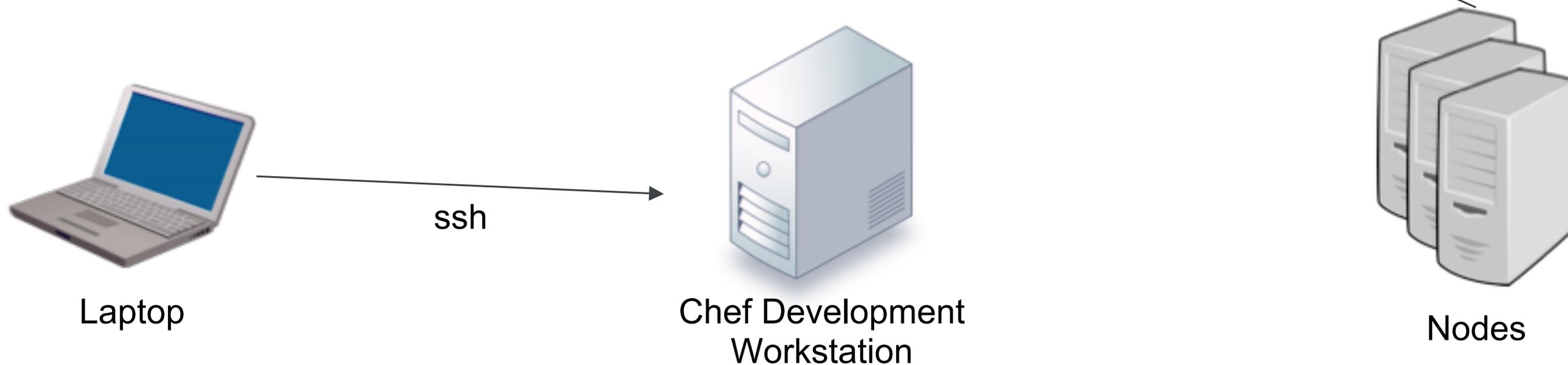
Chef Development  
Workstation



Nodes

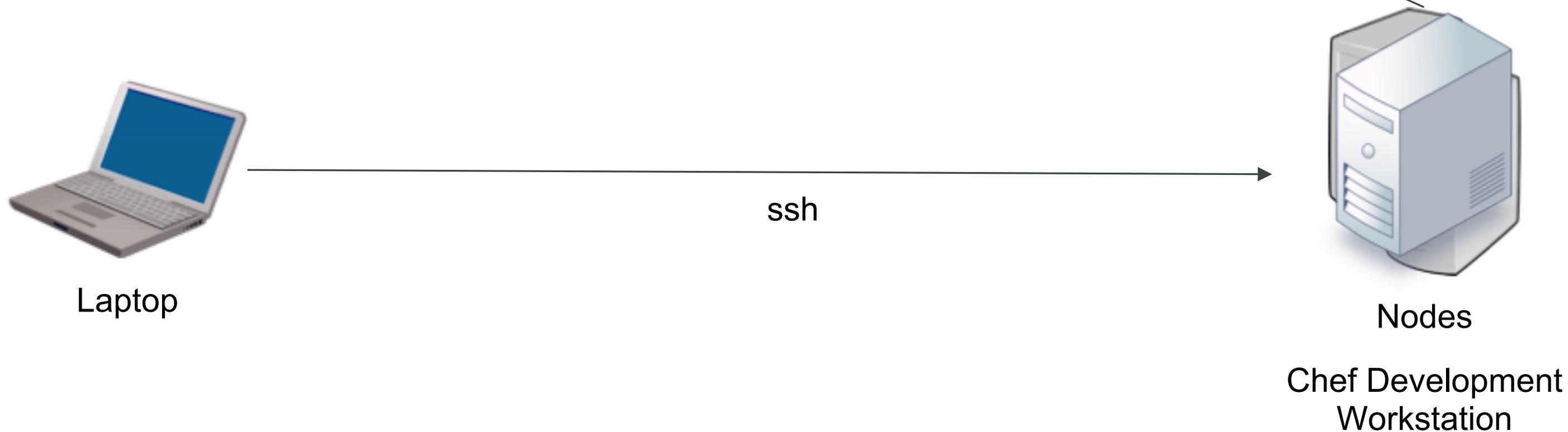
# Learning Environment

# CHEF AUTOMATE



# Learning Environment

# CHEF AUTOMATE



# Access the Learning Environment

- Login to Chef Automate
- Find your workstation/node
- Find your workstation's IP address
- Login to your workstation

# Let's log in to Chef Automate!

- <https://velocity-workshop.community.chefdemo.net>
- Uses a self-signed certificate in this lab
- Username: `admin`
- Password: `chef-automate`



# Browse to your node

CHEFAUTOMATE

Event Feed

Client Runs

Compliance

Scan Jobs

Asset Store

Admin

All Chef servers

All Chef server orgs

★ Latest Events ▶

⚠ Your Chef Automate license will expire in 59 days. [Get a license from Chef](#) or [apply a license](#) if you already have one.

## Event feed for the last week

Total events  
0

Creations  
● 0

Deletions  
○ 0

Updates  
◆ 0

Jun 4

Jun 5

Jun 6

Jun 7

Jun 8

# Browse to your node

### Chef Client Run Status

4 Total Nodes

Failed Nodes  
0

Passed Nodes  
4

Missing Nodes  
0

Total Nodes  
● 4

Failed Nodes  
▲ 0

Successful Nodes  
● 4

Missing Nodes  
? 0

Node Name	Check-in	Uptime	Platform	Environment	Policy Group
✓ blue-hearts-02	11 minutes ago	11 minutes	centos	_default	--
✓ blue-hearts-03	11 minutes ago	11 minutes	centos	_default	--

# Browse to your node

Node Name	Check-in	Uptime	Platform	Environment	Policy Group	
<input checked="" type="checkbox"/> blue-hearts-02	12 minutes ago	11 minutes	centos	_default	--	<a href="#">&gt;</a>
<input checked="" type="checkbox"/> blue-hearts-03	12 minutes ago	11 minutes	centos	_default	--	<a href="#">&gt;</a>
<input checked="" type="checkbox"/> blue-hearts-04	12 minutes ago	11 minutes	centos	_default	--	<a href="#">&gt;</a>
<input checked="" type="checkbox"/> blue-hearts-05	12 minutes ago	11 minutes	centos	_default	--	<a href="#">&gt;</a>

# View details of your node

Node blue-hearts-03

Run history

Use the run history list to examine recent Chef client runs for this node.

✔ This run succeeded on 06/10/2018 at 11:16 PM. All resources ran successfully!



Run Duration	11:16 PM - 11:16 PM
Run Initiator	Not Available
Run Type	Not Available
Run ID	51febe43-1519-4cff-a6ee-1e984c45ce5f

Uptime	11 minutes
Environment	_default
Platform(s)	centos
IP Address	172.31.20.230
FQDN	ip-172-31-20-230.us-west-2.compute.internal

Resources

Run List

Attributes

Total Resources  
☰ 0

Failed  
▲ 0

Successful  
✔ 0

Unchanged  
✔ 0

Unprocessed  
? 0

Status	Step	Type	Name	Action	Cookbook	View
--------	------	------	------	--------	----------	------

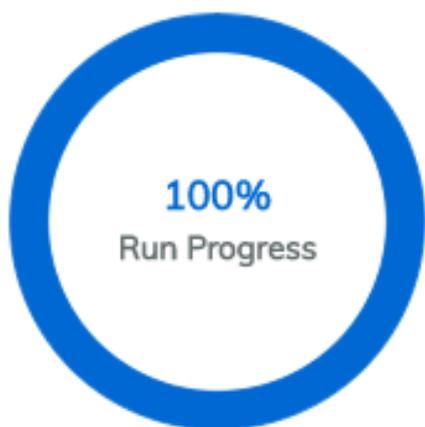
# View details of your node

Node blue-hearts-03

Run history

Use the run history list to examine recent Chef client runs for this node.

✔ This run succeeded on 06/10/2018 at 11:16 PM. All resources ran successfully!



Run Duration	11:16 PM - 11:16 PM
Run Initiator	Not Available
Run Type	Not Available
Run ID	51febe43-1519-4cff-a6ee-1e984c45ce5f

Uptime	11 minutes
Environment	_default
Platform(s)	centos
IP Address	172.31.20.230
FQDN	ip-172-31-20-230.us-west-2.compute.internal

Resources

Run List

Attributes

Roles  
0

Cookbooks  
0

Recipes  
0

Failed  
0

Succeeded  
0

# View details of your node

Resources    Run List    Attributes

Displaying Most Recent Node Attributes

All  
4,156

Default  
0

Normal  
3

Override  
0

Automatic  
4,153

[+ Expand All](#) | [- Collapse All](#)

```
{  
  + "audit" : { ... },  
  + "block_device" : { ... },  
  "chef_environment" : "_default",  
  "chef_guid" : "3244176a-548f-4c6e-81b3-c83c4e099fb7",  
  + "chef_packages" : { ... },  
  + "cloud" : { ... },  
  + "command" : { ... },  
  + "counters" : { ... },  
  + "cpu" : { ... },  
  "current_user" : "root",  
  + "dmi" : { ... },  
  + "docker" : { ... },  
  "domain" : "us-west-2.compute.internal",  
  + "ec2" : { ... },  
  + "etc" : { ... },  
}
```



+ "ec2" : { ... },

# Find the IP of your node

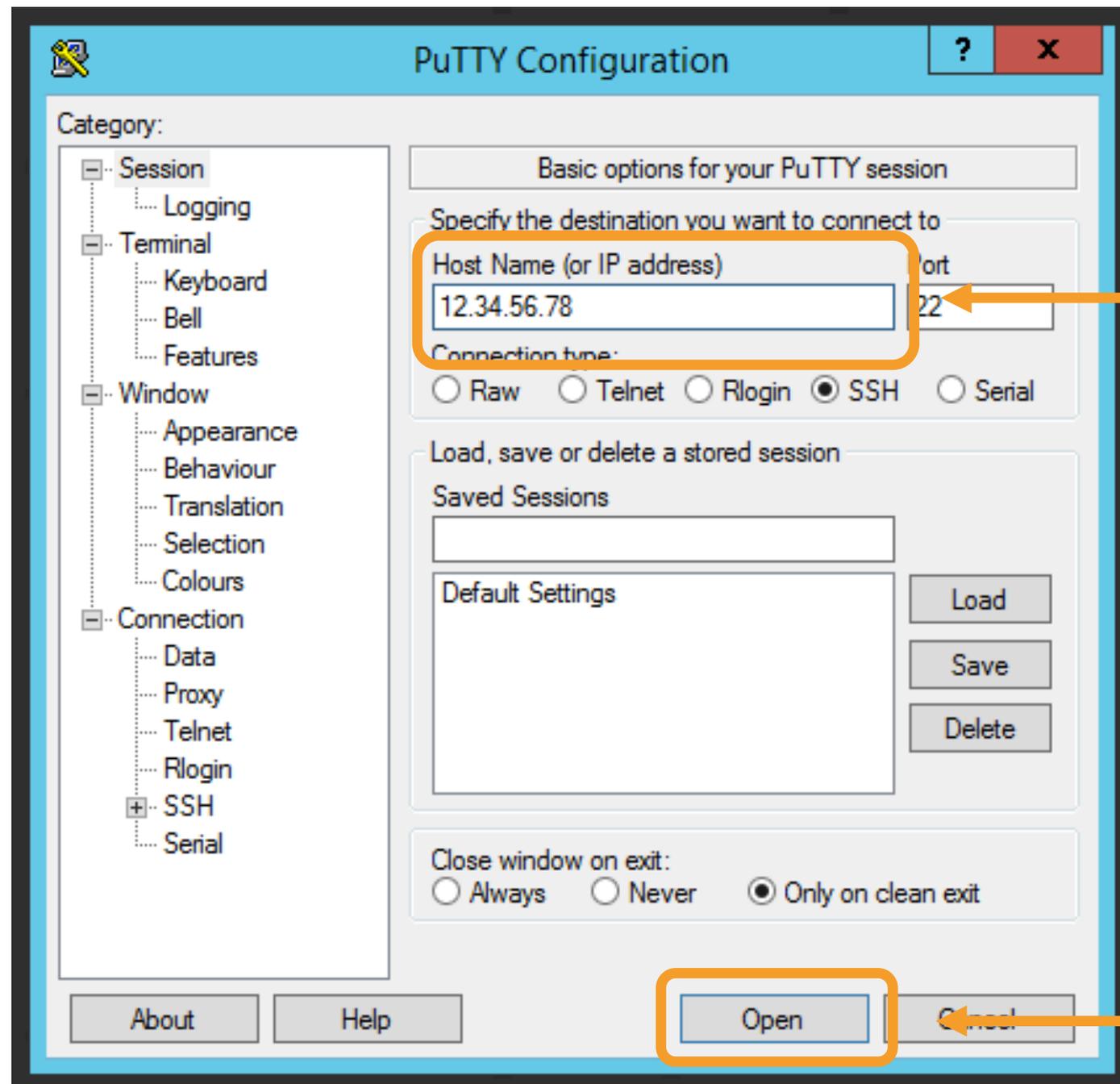
```
+ "network_interfaces_macs" : { ... },  
  "placement_availability_zone" : "us-west-2b",  
  "product_codes" : "aw0evgkw8e5c1q413zgy5pjce",  
  "profile" : "default-hvm",  
  "public_hostname" : "ec2-54-149-195-207.us-west-2.compute.amazonaws.com",  
  "public_ipv4" : "54.149.195.207",
```

# Log in to your remote workstation



```
$ ssh chef@12.34.56.78
```

# Using PuTTY on Windows



# Log in to your remote workstation



```
$ ssh chef@12.34.56.78
```

```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.  
ECDSA key fingerprint is SHA256:zAtoe029XbhRNvvg542cuh4qsKCEaX8hNI1E0Cbgd3I.  
Are you sure you want to continue connecting (yes/no)?
```

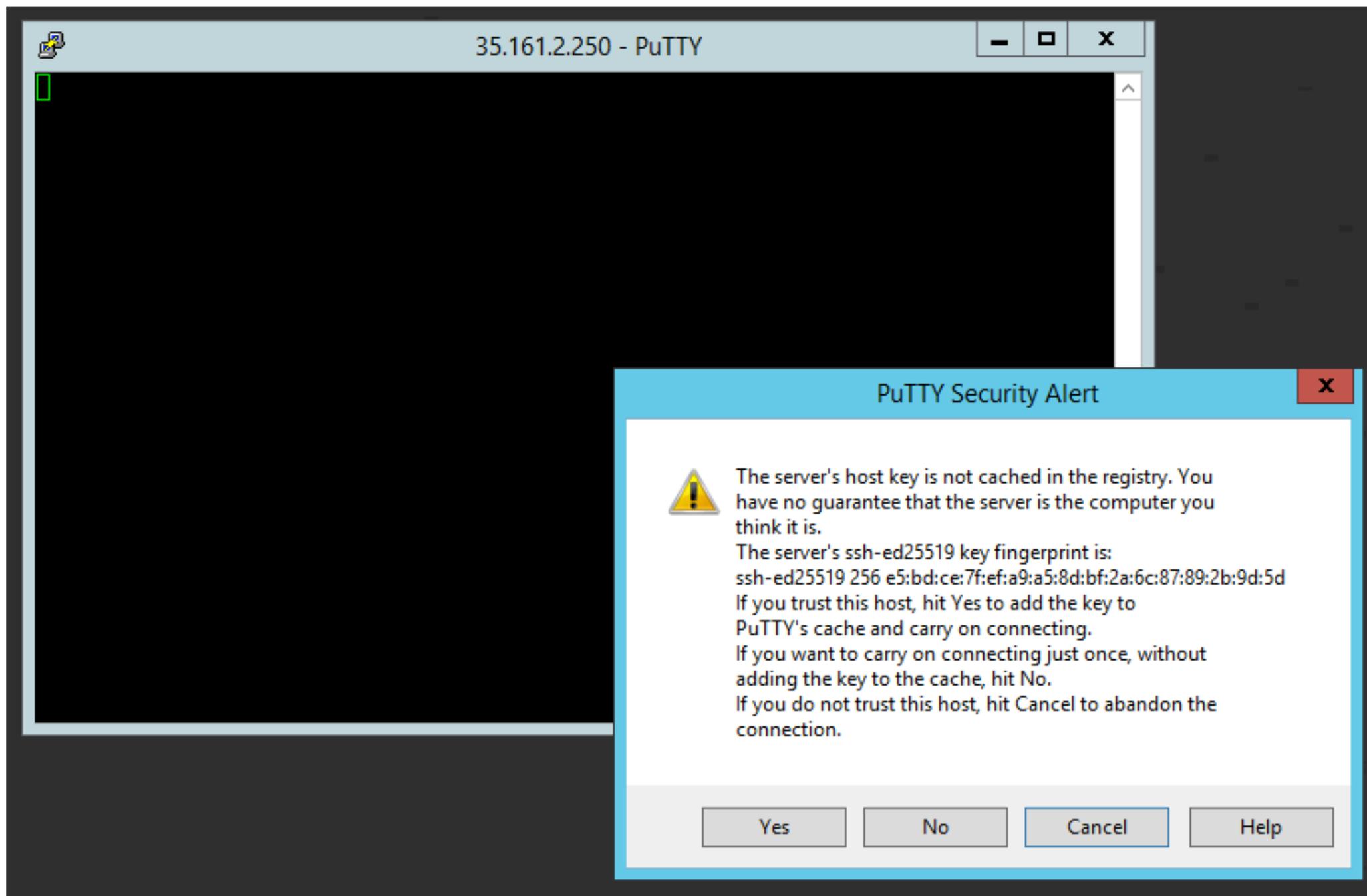
# Log in to your remote workstation



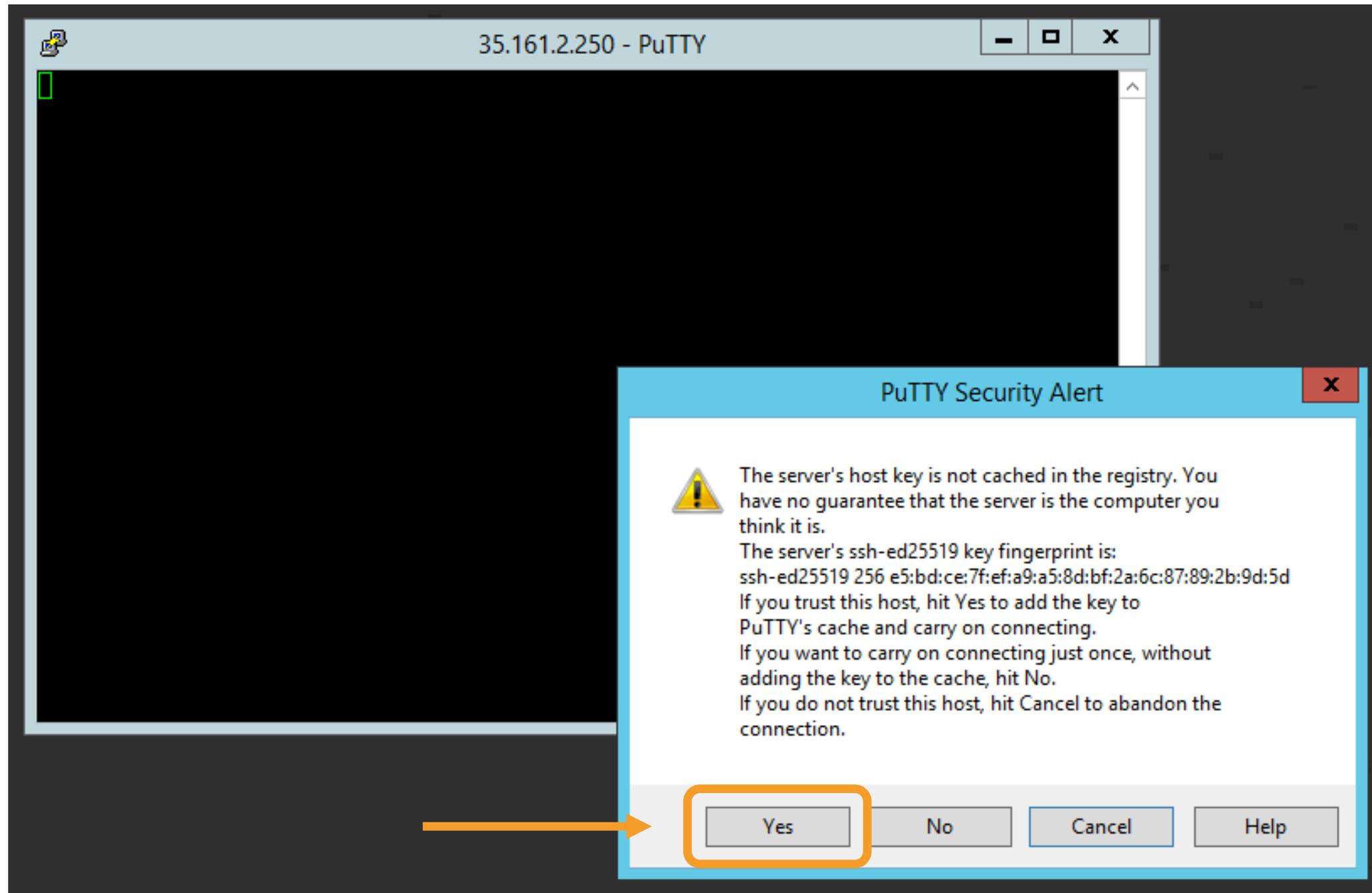
```
$ ssh chef@12.34.56.78
```

```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.  
ECDSA key fingerprint is SHA256:zAtoe029XbhRNvvg542cuh4qsKCEaX8hNI1E0Cbgd3I.  
Are you sure you want to continue connecting (yes/no)? yes
```

# Using PuTTY on Windows



# Using PuTTY on Windows



# Log in to your remote workstation



```
$ ssh chef@12.34.56.78
```

```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.  
ECDSA key fingerprint is SHA256:zAtoe029XbhRNvvg542cuh4qsKCEaX8hNI1E0Cbgd3I.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '12.34.56.78' (ECDSA) to the list of known hosts.  
chef@12.34.56.78's password:
```

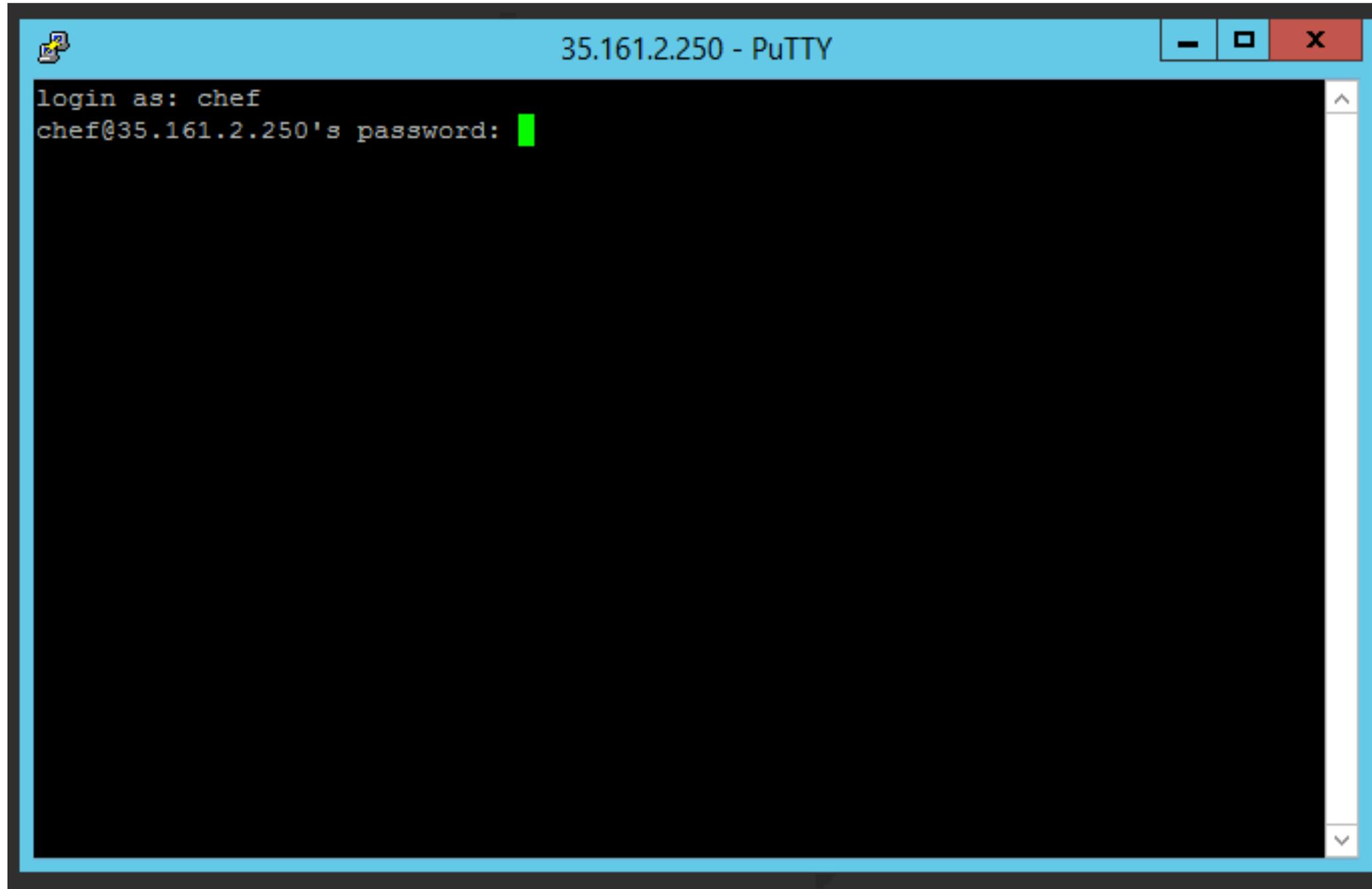
# Log in to your remote workstation



```
$ ssh chef@12.34.56.78
```

```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.  
ECDSA key fingerprint is SHA256:zAtoe029XbhRNvvg542cuh4qsKCEaX8hNI1E0Cbgd3I.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '12.34.56.78' (ECDSA) to the list of known hosts.  
chef@12.34.56.78's password: velocity
```

# Using PuTTY on Windows



# Create a file with your name



```
$ touch firstname-lastname
```

# Create a file with your name



```
$ touch will-robinson
```

# List your home directory



```
$ ls -t
```

```
will-robinson      cookbooks      Berksfile      profiles  
nodes              Berksfile.lock config.json
```

# Verify the installation



```
$ which inspec
```

```
/opt/chefdk/bin/inspec
```

# Verify the installation



```
$ inspec version
```

```
2.1.72
```

```
Your version of InSpec is out of date! The latest version is 2.2.10.
```

# Verify the installation



```
$ which chef
```

```
/opt/chefdk/bin/chef
```

# Verify the installation



```
$ chef --version
```

```
Chef Development Kit Version: 3.0.36
```

```
chef-client version: 14.1.12
```

```
delivery version: master (7206afaf4cf29a17d2144bb39c55b7212cfafcc7)
```

```
berks version: 7.0.2
```

```
kitchen version: 1.21.2
```

```
inspec version: 2.1.72
```

# Chef DK - The Chef Development Kit

## Foodcritic

### Test Your "Chef Style"

- Validate your Chef code against Chef best practices
- Extend with rules to enforce organizational Chef development best practices
- Enforce compliance & security practices

## CookStyle

### Validate your Ruby

- Validate your Chef code against Ruby best practices
- Identify potential Ruby errors (unclosed strings, etc.)
- Identify style/convention that helps write better code (single quotes vs. double quotes)

## ChefSpec

### Simulate Chef

- Validate your Chef code will run
- Testing for more Chef advanced used cases
- Useful for regression testing

## Test Kitchen

### Let's do this (almost) for real

- Validate your Chef code against Chef best practices
- Extend with rules to enforce organizational Chef development best practices
- Enforce compliance & security practices

## InSpec

### Verify automation results & ensure compliance

- Assert the intention of your Chef code
- Verify on live systems that your Chef code produced the correct result
- Confirm your Chef code did not produce compliance drift or failures

# Running Chef on the Node

**CHEF**  
AUTOMATE

Node data



Nodes

# Go home



```
$ cd ~
```

# Run chef



```
$ run_chef
```

```
Starting Chef Client, version 14.1.12
```

```
resolving cookbooks for run list: []
```

```
Synchronizing Cookbooks:
```

```
Installing Cookbook Gems:
```

```
Compiling Cookbooks...
```

```
[2018-06-11T03:36:50+00:00] WARN: Node blue-hearts-03 has an empty run list.
```

```
Converging 0 resources
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Client finished, 0/0 resources updated in 01 seconds
```

# Check the converge status in Automate

Run history

# Check the converge status in Automate

Run history



# Check the converge status in Automate

Run history

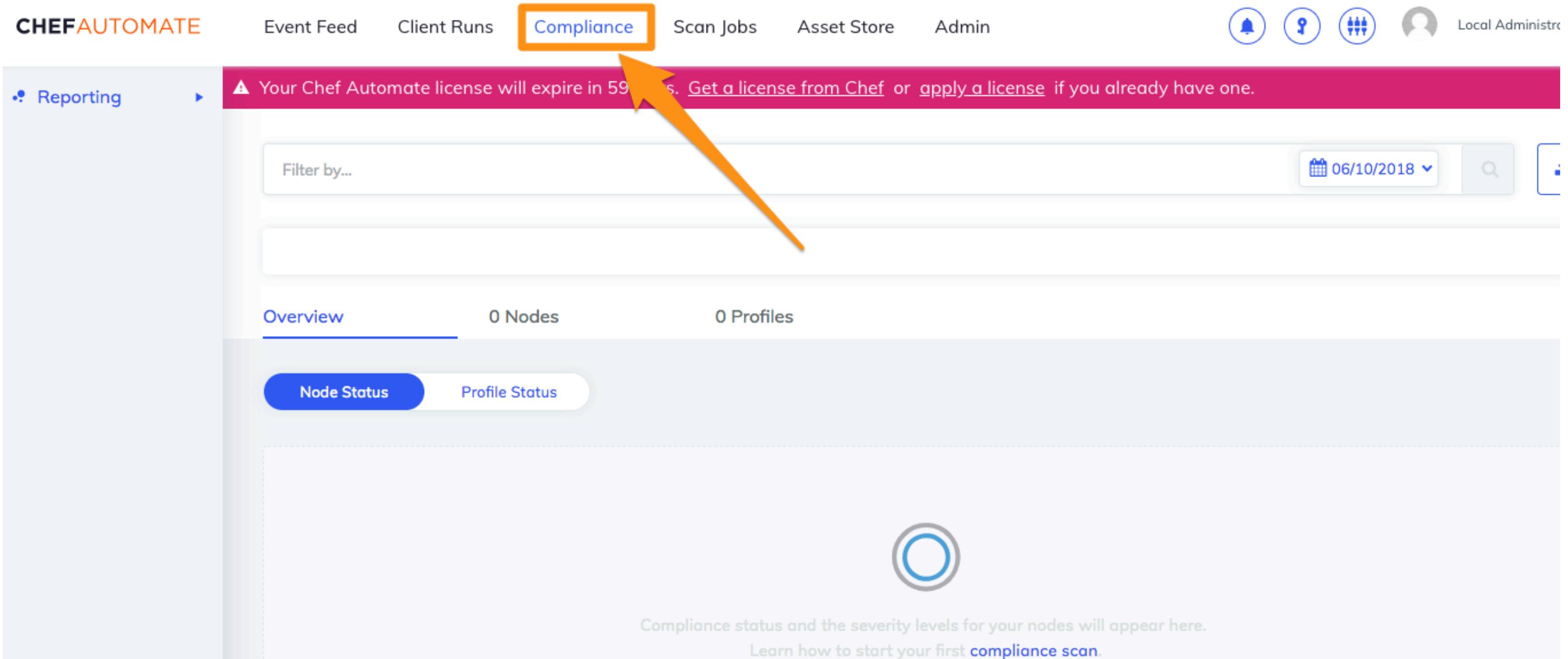


### Run History ×

2 ▲ 0 ● 2 Last 24 hours ▾

✓	06/10/18	23:36:50
	a few seconds	23:36:50
< ✓	06/10/18	23:16:44
	a few seconds	23:16:44

# Compliance data in Automate



The screenshot shows the Chef Automate interface. At the top, the navigation bar includes 'CHEFAUTOMATE', 'Event Feed', 'Client Runs', 'Compliance' (highlighted with an orange box and an arrow), 'Scan Jobs', 'Asset Store', and 'Admin'. On the right, there are icons for notifications, a key, a group of people, and a user profile labeled 'Local Administrator'. A pink banner at the top of the main content area contains a warning: 'Your Chef Automate license will expire in 59 days. [Get a license from Chef](#) or [apply a license](#) if you already have one.' Below the banner is a search bar with 'Filter by...' and a date selector set to '06/10/2018'. The main content area shows 'Overview' with '0 Nodes' and '0 Profiles'. There are two tabs: 'Node Status' (active) and 'Profile Status'. The main content area is currently empty, displaying a large circular loading icon and the text: 'Compliance status and the severity levels for your nodes will appear here. [Learn how to start your first compliance scan.](#)'

# Compliance data in Automate

Overview

0 Nodes

0 Profiles

Node Status

Profile Status



Compliance status and the severity levels for your nodes will appear here.  
Learn how to start your first [compliance scan](#).

Global Compliance

# Run Chef with the audit cookbook



```
$ run_chef "recipe[audit::default]"
```

```
Starting Chef Client, version 14.1.12
```

```
...
```

```
- Chef::Handler::AuditReport
```

```
Running handlers complete
```

```
Chef Client finished, 0/2 resources updated in 03 seconds
```

# Check the converge status in Automate

Resources    Run List    Attributes

**Total Resources**  
☰ 2

Failed  0

Successful  0

Unchanged  2

Unprocessed  0

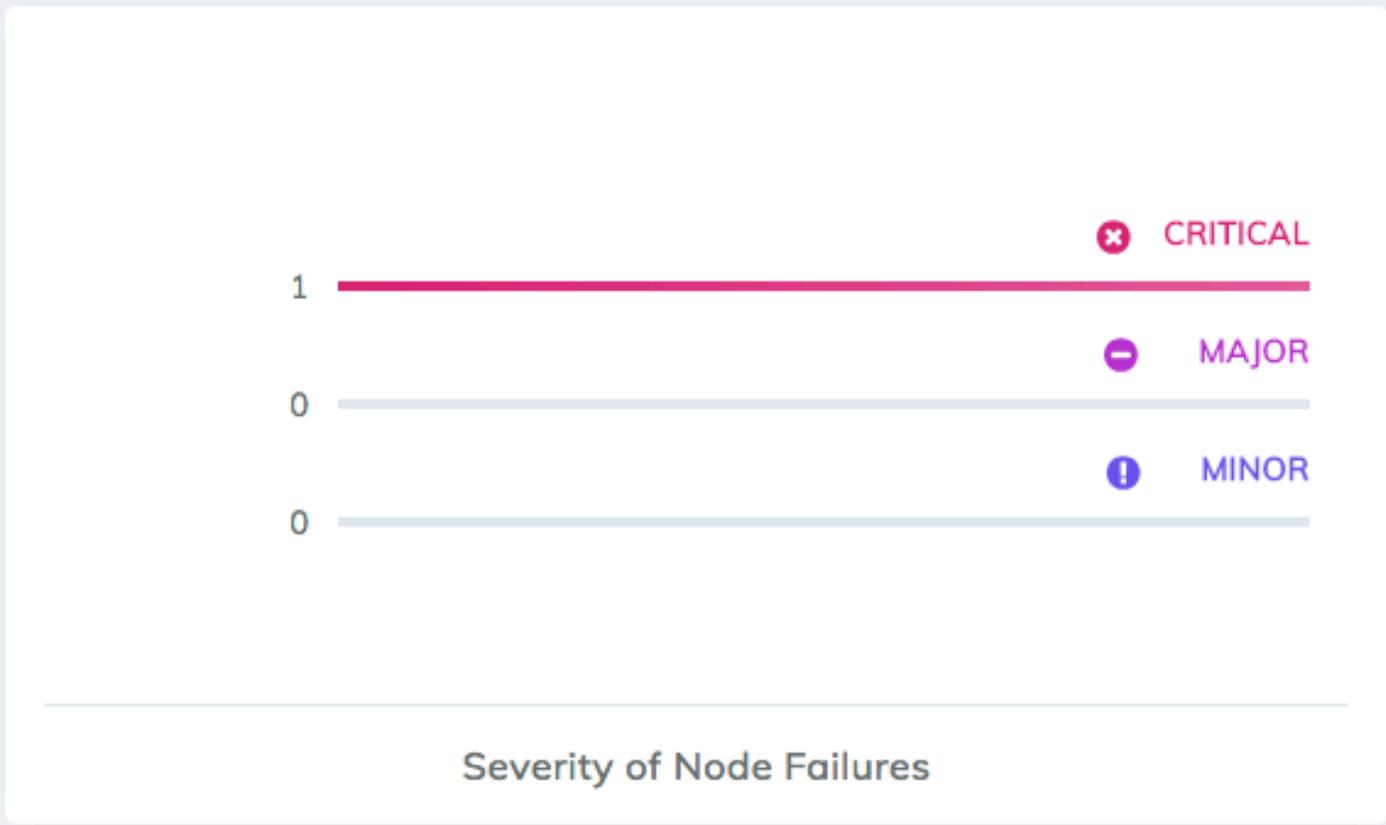
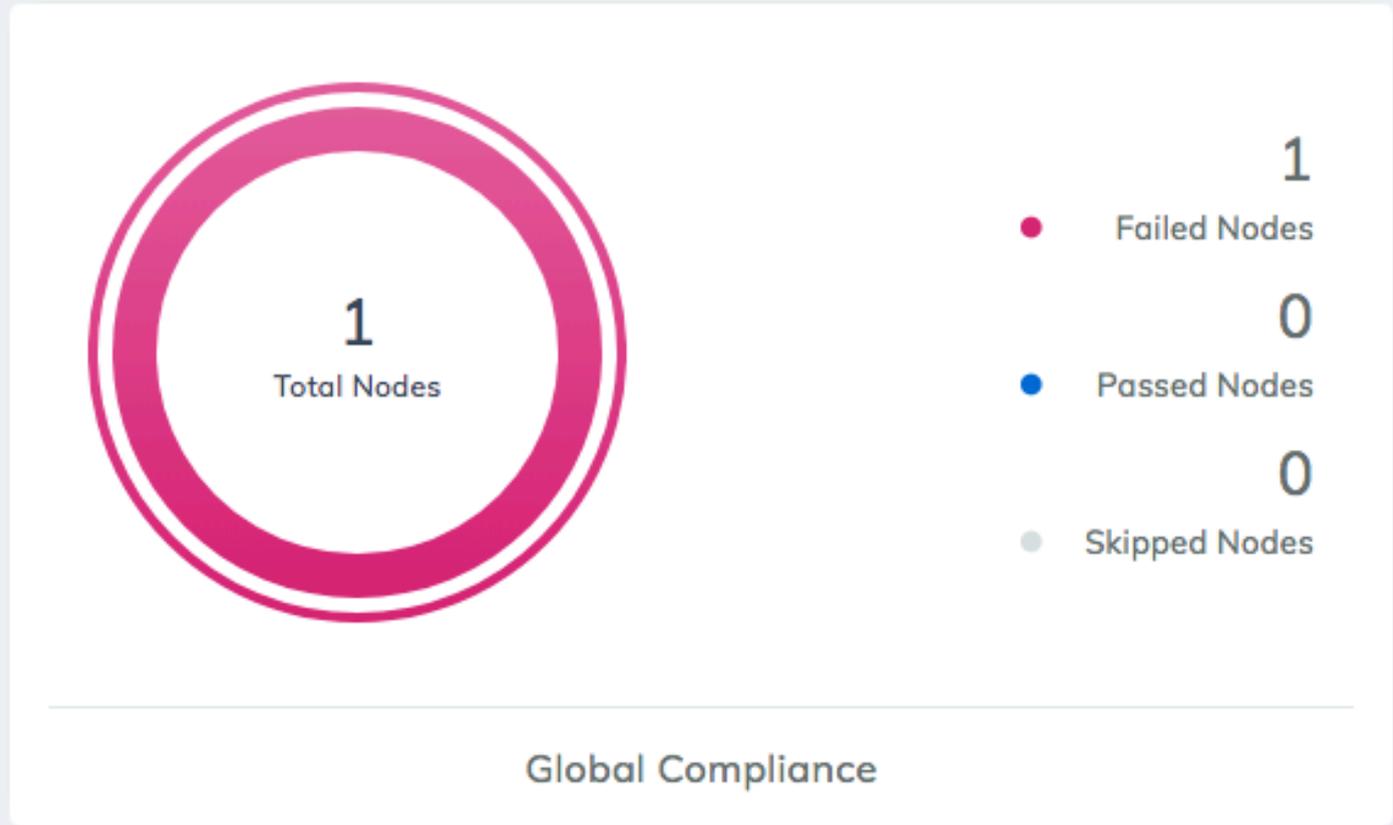
Status	Step	Type	Name	Action	Cookbook	View
	1/2	inspec_gem	inspec	install	audit	--
	2/2	inspec_gem	inspec	nothing	audit	--

# Check the compliance status in Automate

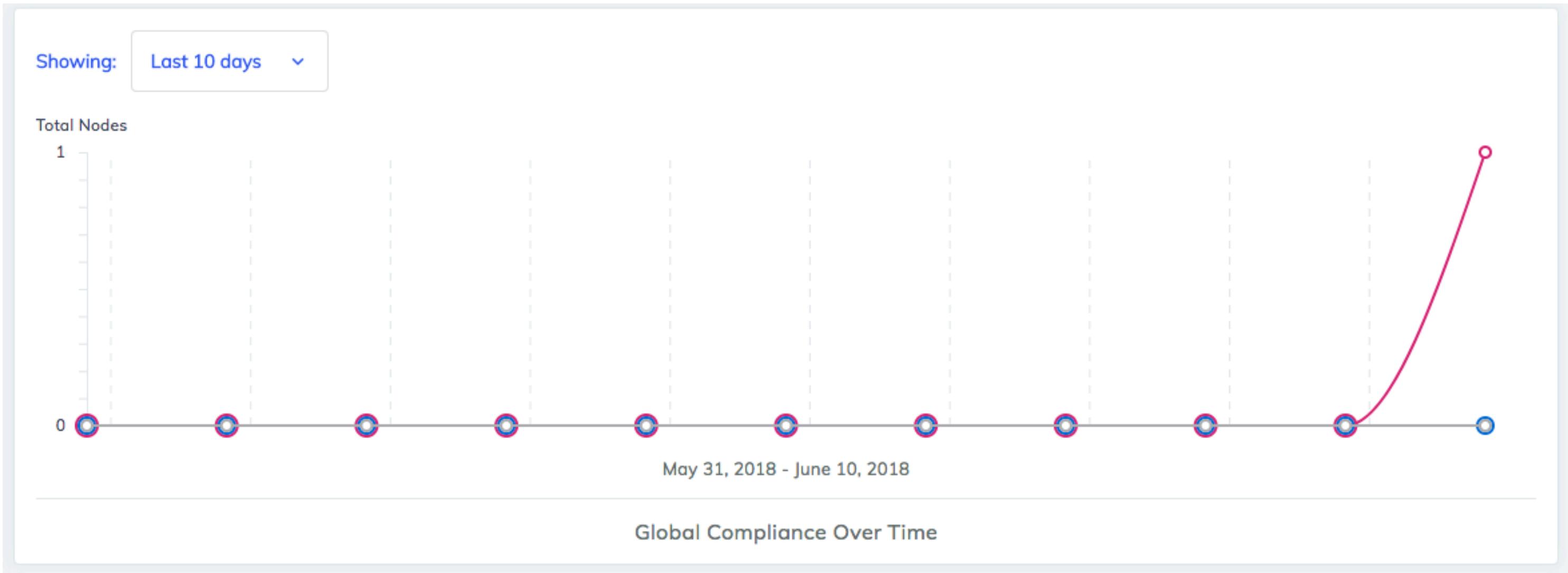
**⚠ Your System is Not Compliant** Report Metadata +

Overview 1 Nodes 1 Profiles

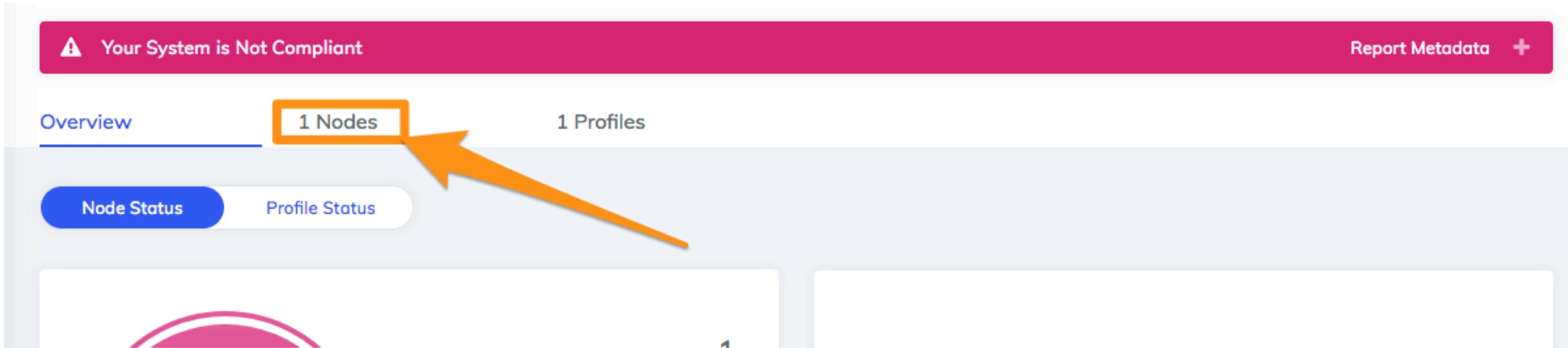
**Node Status** Profile Status



# Check the compliance status in Automate



# Check the compliance status in Automate



The screenshot shows the Chef Automate compliance status interface. At the top, a pink banner displays a warning icon and the text "Your System is Not Compliant", with a "Report Metadata +" link on the right. Below this, the "Overview" tab is active, showing "1 Nodes" and "1 Profiles". A blue button labeled "Node Status" is selected, and a white button labeled "Profile Status" is also visible. An orange arrow points from the "1 Nodes" text to the "Node Status" button. The main content area is currently empty, showing a partial pink arc at the bottom left.

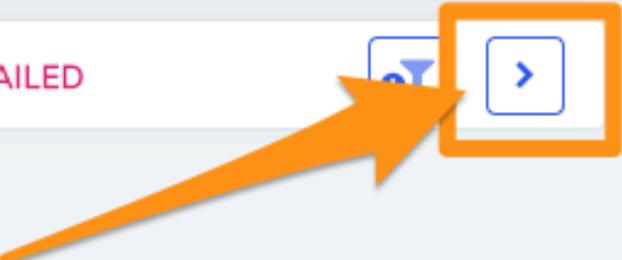
# Check the compliance status in Automate

**⚠ Your System is Not Compliant** Report Metadata +

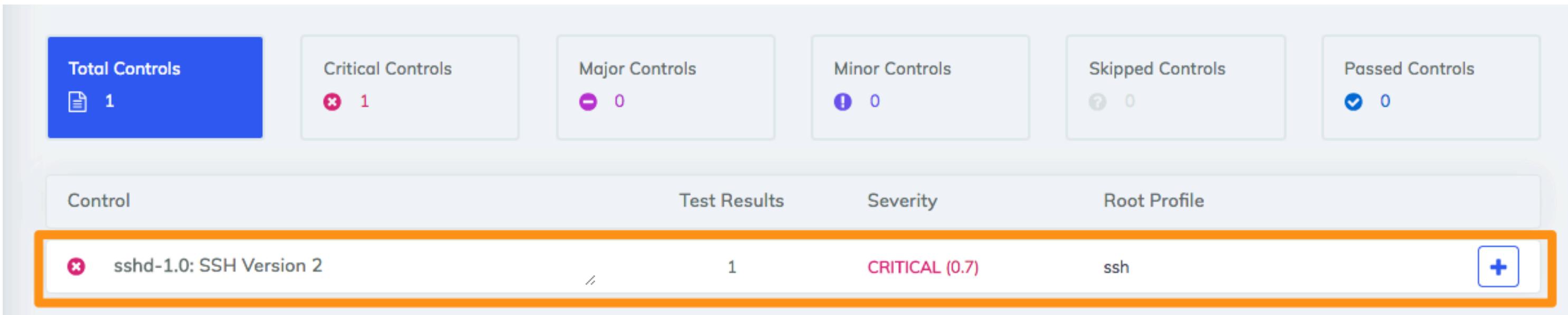
Overview 1 Nodes 1 Profiles

Nodes	Platform	Environment	Last Scan	Control Failures
<b>⚠</b> blue-hearts-03	centos	_default	4 minutes ago	1 FAILED

« < 1 > »



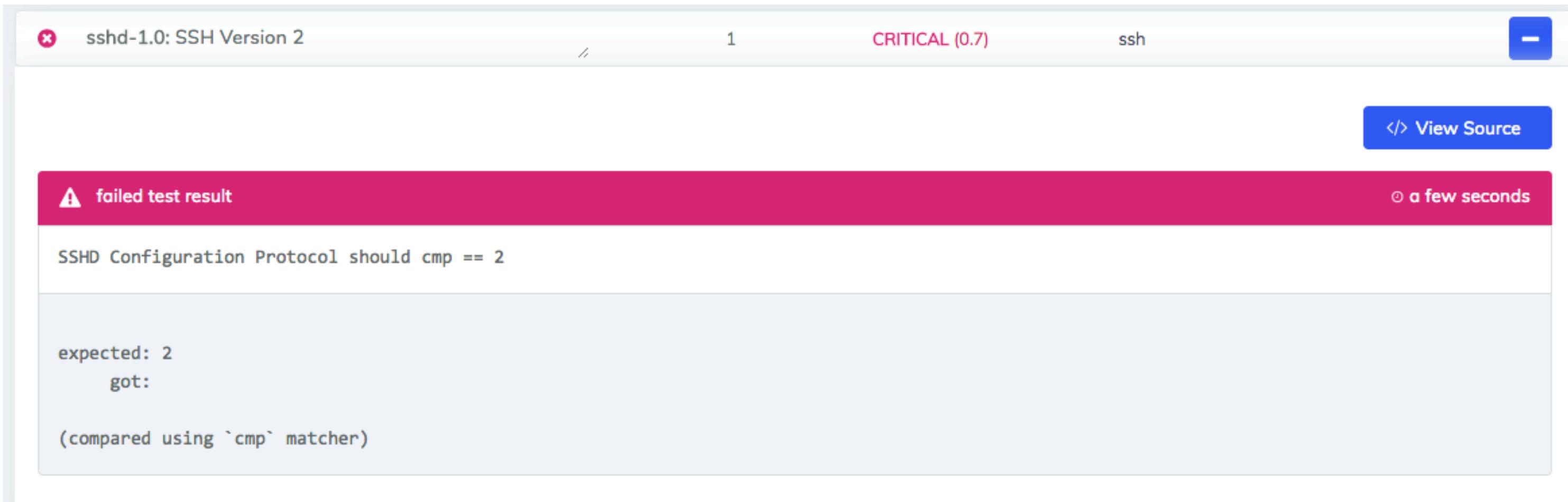
# Check the compliance status in Automate



The screenshot displays the Chef Automate compliance dashboard. At the top, there are six summary cards: Total Controls (1), Critical Controls (1), Major Controls (0), Minor Controls (0), Skipped Controls (0), and Passed Controls (0). Below these is a table with the following columns: Control, Test Results, Severity, and Root Profile. A single row is highlighted with an orange border, showing a critical control failure.

Control	Test Results	Severity	Root Profile
✖ sshd-1.0: SSH Version 2	1	CRITICAL (0.7)	ssh

# Check the compliance status in Automate



The screenshot shows a compliance check result in the Automate interface. The header bar indicates the check is for 'sshd-1.0: SSH Version 2', has a severity of 'CRITICAL (0.7)', and is associated with the 'ssh' profile. A 'View Source' button is visible in the top right. A red banner indicates a 'failed test result' with a refresh icon and the text 'a few seconds'. The test description is 'SSHD Configuration Protocol should cmp == 2'. The failure details show 'expected: 2' and 'got:' followed by '(compared using `cmp` matcher)'.

sshd-1.0: SSH Version 2 // 1 CRITICAL (0.7) ssh

[View Source](#)

**failed test result** a few seconds

SSHD Configuration Protocol should cmp == 2

expected: 2  
got:  
(compared using `cmp` matcher)

# Check the compliance status in Automate



The screenshot displays a browser window with a tab titled "sshd-1.0: SSH Version 2". The page content includes a header with "1" and "CRITICAL (0.7)", and a "View Source" button. A prominent pink error banner reads "failed test result" with a refresh icon and "a few seconds" timer. Below the banner, the test failure details are shown in a light blue box:

```
SSHD Configuration Protocol should cmp == 2

expected: 2
got:

(compared using `cmp` matcher)
```

# Check the compliance status in Automate



The screenshot displays a Chef Automate interface for a compliance check. The top header shows the resource name 'sshd-1.0: SSH Version 2', a count of '1', a severity level of 'CRITICAL (0.7)', and the resource type 'ssh'. A blue 'View Source' button is located in the top right corner. A prominent orange box highlights a 'failed test result' section. This section contains the following text: 'SSH Configuration Protocol should cmp == 2', 'expected: 2', 'got:', and '(compared using `cmp` matcher)'. An orange arrow points from the 'failed test result' box to the 'View Source' button. A refresh icon and the text 'a few seconds' are visible in the top right of the failed test result box.

# Check the compliance status in Automate



sshd-1.0: SSH Version 2

Only SSH version 2 should be enabled

```
control 'sshd-1.0' do
  impact 0.7
  title 'SSH Version 2'
  desc 'Only SSH version 2 should be enabled'
  describe sshd_config do
    its('Protocol') { should cmp 2 }
  end
end
```

# Review the Setup

tying it all together...

a.k.a. "How the heck did that happen?"

# Go home again



```
$ cd ~
```

# List contents



```
$ ls
```

```
will-robinson      cookbooks          Berksfile          profiles
nodes              Berksfile.lock    config.json
```

# List cookbooks



```
$ ls cookbooks
```

```
audit
```

# Audit Cookbook

- Installs InSpec (if necessary - included in Chef 13 by default)
- Run InSpec profiles
- Report results to Chef Automate

# Attributes for the Audit cookbook



```
$ cat config.json
```

```
{
  "audit": {
    "collector": "chef-automate",
    "profiles": [
      {
        "name": "ssh",
        "path": "/home/chef/profiles/ssh"
      }
    ]
  }
}
```

# Our ssh InSpec profile



```
$ tree profiles/ssh
```

```
profiles/ssh/  
├── controls  
│   └── ssh.rb  
└── inspec.yml
```

```
1 directory, 2 files
```

# Our ssh InSpec profile



```
$ cat profiles/ssh/controls/ssh.rb
```

```
control 'sshd-1.0' do
  impact 0.7
  title 'SSH Version 2'
  desc 'Only SSH version 2 should be enabled'
  describe sshd_config do
    its('Protocol') { should cmp 2 }
  end
end
```

# Run locally with InSpec



```
$ inspec exec profiles/ssh
```

```
Profile: SSH Configuration (ssh)
```

```
Version: 0.1.0
```

```
Target: local://
```

- × sshd-1.0: SSH Version 2
  - × SSHD Configuration Protocol should cmp == 2

```
expected: 2
```

```
got:
```

```
(compared using `cmp` matcher)
```

```
Profile Summary: 0 successful controls, 1 control failure, 0 controls skipped
```

```
Test Summary: 0 successful, 1 failure, 0 skipped
```

# Next Steps

- Automate the remediation of the failing control
- Test the remediation before deploying
- Deploy the remediation, and use the audit cookbook to report back to Automate
- View the compliant node in Automate

# Create an SSH Chef Cookbook

- A recipe to deploy a proper `sshd_config` configuration file
- A local test environment configured to test our changes

# Move to the cookbooks directory



```
$ cd ~/cookbooks
```

# Generate a new ssh cookbook



```
$ chef generate cookbook ssh
```

```
Generating cookbook ssh
```

- Ensuring correct cookbook file content
- Committing cookbook files to git
- Ensuring delivery configuration
- Ensuring correct delivery build cookbook content
- Adding delivery configuration to feature branch
- Adding build cookbook to feature branch
- Merging delivery content feature branch to master

```
Your cookbook is ready. Type `cd ssh` to enter it.
```

```
There are several commands you can run to get started locally developing and testing your cookbook.  
Type `delivery local --help` to see a full list.
```

```
Why not start by writing a test? Tests for the default recipe are stored at:
```

```
test/integration/default/default_test.rb
```

```
If you'd prefer to dive right in, the default recipe can be found at:
```

```
recipes/default.rb
```

# Add a server recipe to the ssh cookbook



```
$ chef generate recipe ssh server
```

```
Recipe: code_generator::recipe
* directory[./ssh/spec/unit/recipes] action create (up to date)
* cookbook_file[./ssh/spec/spec_helper.rb] action create_if_missing (up to date)
* template[./ssh/spec/unit/recipes/server_spec.rb] action create_if_missing
  - create new file ./ssh/spec/unit/recipes/server_spec.rb
  - update content in file ./ssh/spec/unit/recipes/server_spec.rb from none to 7c8724
  (diff output suppressed by config)
* directory[./ssh/test/integration/default] action create (up to date)
* template[./ssh/test/integration/default/server_test.rb] action create_if_missing
  - create new file ./ssh/test/integration/default/server_test.rb
  - update content in file ./ssh/test/integration/default/server_test.rb from none to f2f1c1
  (diff output suppressed by config)
* template[./ssh/recipes/server.rb] action create
  - create new file ./ssh/recipes/server.rb
  - update content in file ./ssh/recipes/server.rb from none to f29497
  (diff output suppressed by config)
```

# Add a template to the cookbook



```
$ chef generate template ssh sshd_config -s /etc/ssh/sshd_config
```

```
Recipe: code_generator::template
```

```
* directory[./ssh/templates/default] action create
```

```
- create new directory ./ssh/templates/default
```

```
* file[./ssh/templates/sshd_config.erb] action create
```

```
- create new file ./ssh/templates/sshd_config.erb
```

```
- update content in file ./ssh/templates/sshd_config.erb from none to a16b11
```

```
(diff output suppressed by config)
```

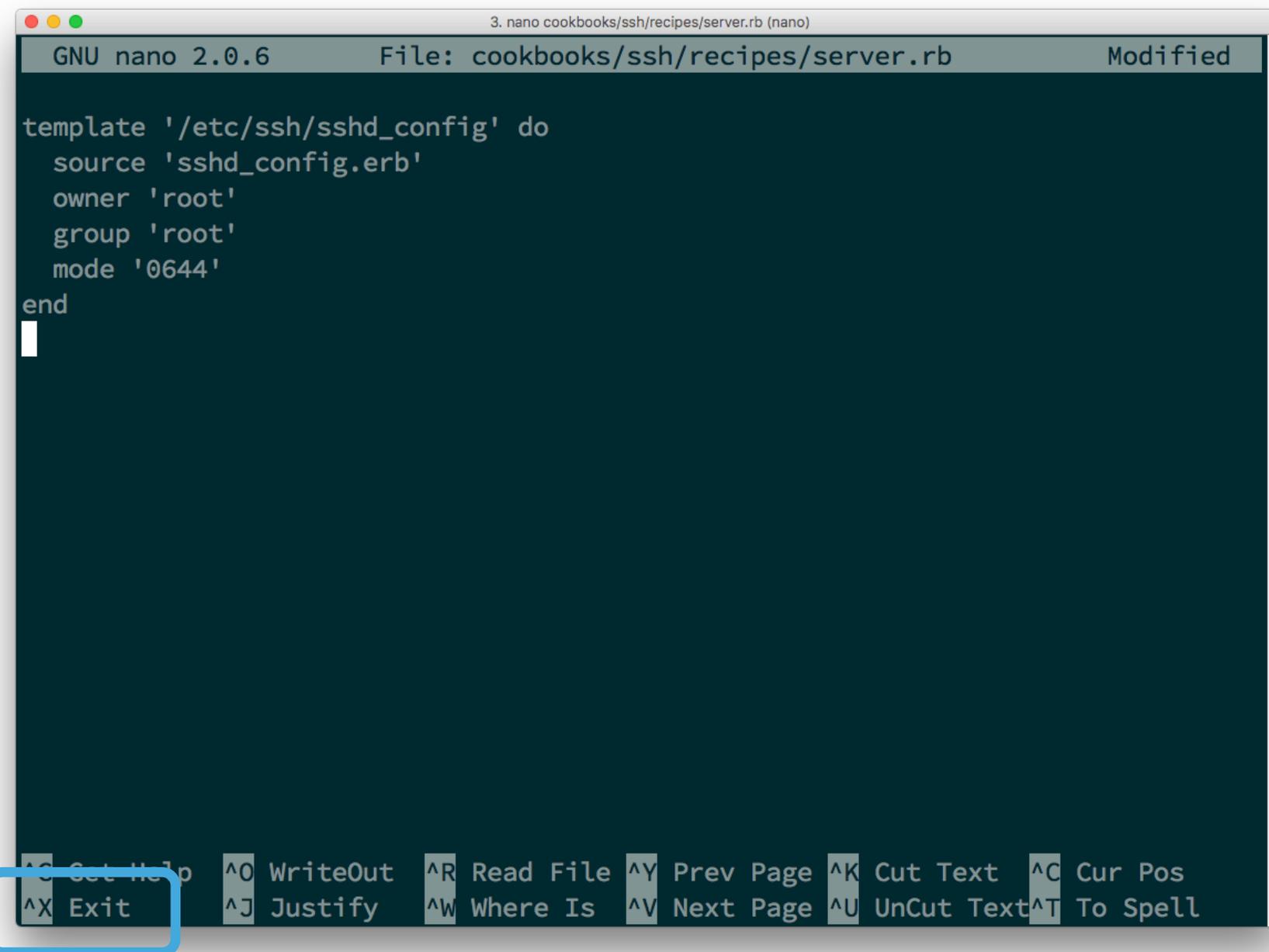
# Server Recipe

```
~/cookbooks/ssh/recipes/server.rb
```

```
template '/etc/ssh/sshd_config' do
  source 'sshd_config.erb'
  owner 'root'
  group 'root'
  mode '0644'
end
```

Never used a command-line text editor before? Type:  
nano cookbooks/ssh/recipes/server.rb

# Using nano



```
3. nano cookbooks/ssh/recipes/server.rb (nano)
GNU nano 2.0.6      File: cookbooks/ssh/recipes/server.rb      Modified

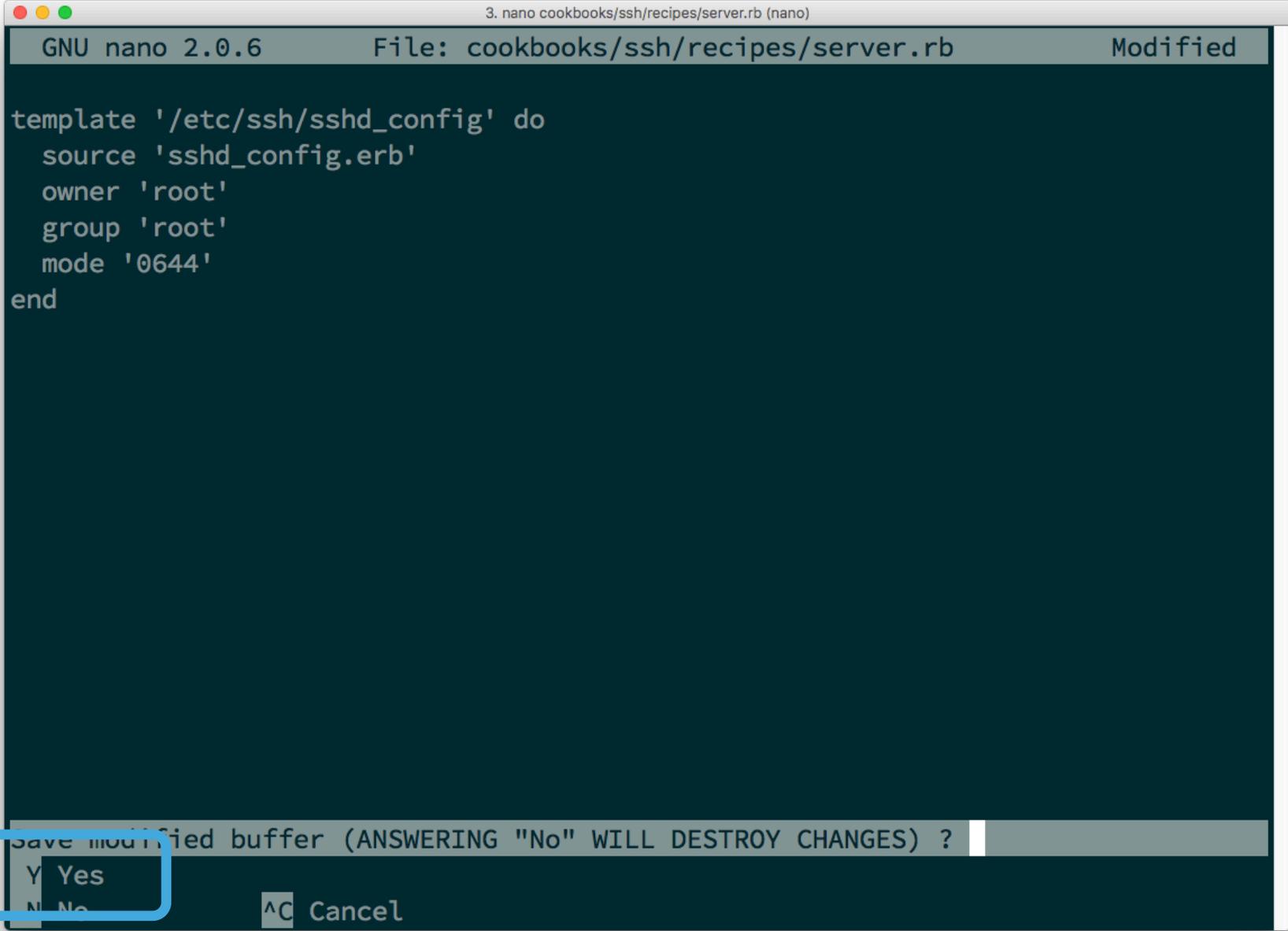
template '/etc/ssh/sshd_config' do
  source 'sshd_config.erb'
  owner 'root'
  group 'root'
  mode '0644'
end

```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Press Control-x

# Using nano



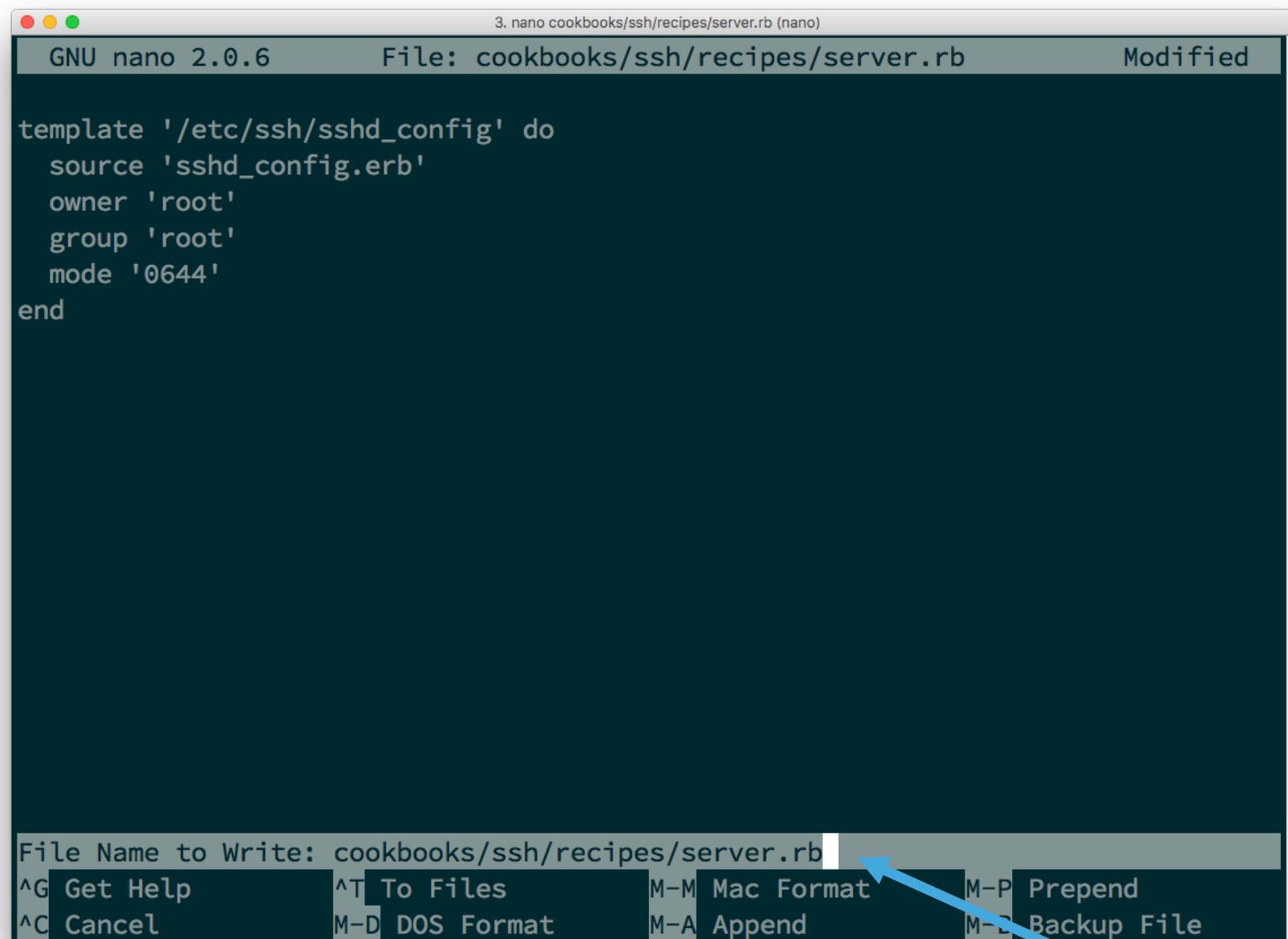
```
3. nano cookbooks/ssh/recipes/server.rb (nano)
GNU nano 2.0.6      File: cookbooks/ssh/recipes/server.rb      Modified

template '/etc/ssh/sshd_config' do
  source 'sshd_config.erb'
  owner 'root'
  group 'root'
  mode '0644'
end

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No
^C Cancel
```

Press Y

# Using nano



```
3. nano cookbooks/ssh/recipes/server.rb (nano)
GNU nano 2.0.6      File: cookbooks/ssh/recipes/server.rb      Modified

template '/etc/ssh/sshd_config' do
  source 'sshd_config.erb'
  owner 'root'
  group 'root'
  mode '0644'
end

File Name to Write: cookbooks/ssh/recipes/server.rb
^G Get Help      ^T To Files      M-M Mac Format    M-P Prepend
^C Cancel        M-D DOS Format    M-A Append        M-B Backup File
```

Press Enter to confirm filename

# Remember...

Infrastructure policies need testing!

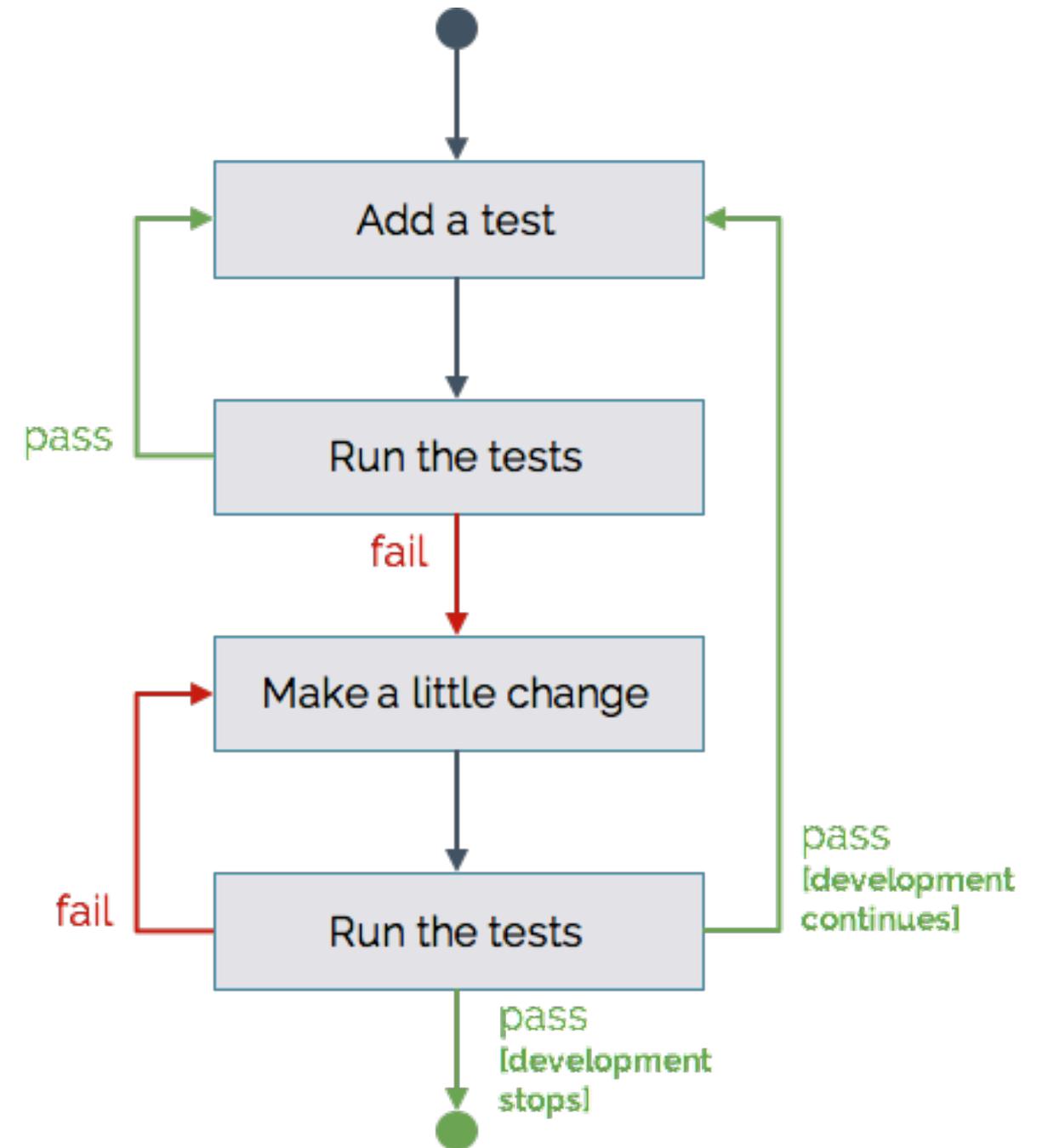
- Linting
- Static analysis
- Unit testing
- Integration Testing
- Compliance Testing



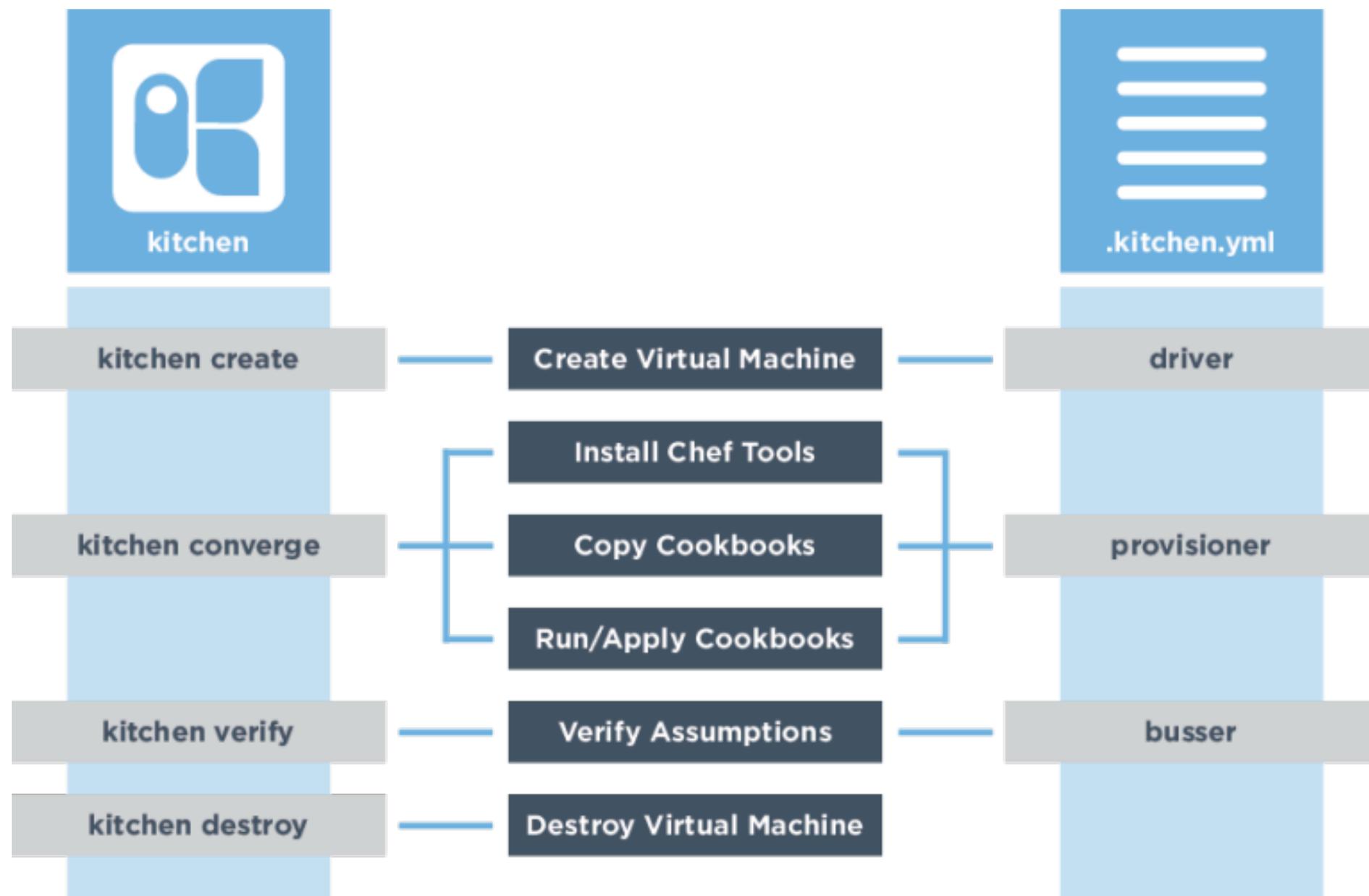
**"Infrastructure  
as Code"**  
should be tested  
like ANY other  
codebase.

# Test-driven Development

- Write a test, watch it fail
- Write some code
- Write and run more tests
- Code review
- Delivery pipeline to production
- Lowered chance of production failure



# Testing the change



Microsoft Azure



# Test Kitchen Configuration (1 of 3)



```
~/cookbooks/ssh/.kitchen.yml
```

```
---
```

```
driver:
```

```
- name: vagrant
```

```
+ name: docker
```

```
...
```

# Test Kitchen Configuration (2 of 3)

```
~/cookbooks/ssh/.kitchen.yml
```

```
...
```

```
platforms:
```

```
- - name: ubuntu-16.04
```

```
- - name: centos-7.2
```

```
+ - name: centos-7.3
```

```
...
```

# Test Kitchen Configuration (3 of 3)

```
~/cookbooks/ssh/.kitchen.yml
```

```
suites:
```

```
- - name: default
```

```
+ - name: server
```

```
  run_list:
```

```
- - recipe[ssh::default]
```

```
+ - recipe[ssh::server]
```

```
  verifier:
```

```
    inspec_tests:
```

```
- - test/smoke/default
```

```
+ - /home/chef/profiles/ssh
```

```
  attributes:
```

# Move to the ssh cookbook directory



```
$ cd ~/cookbooks/ssh
```

# List the kitchens



```
$ kitchen list
```

Instance	Driver	Provisioner	Verifier	Transport	Last Action	Last Error
server-centos-73	Docker	ChefZero	Inspec	Ssh	<Not Created>	<None>

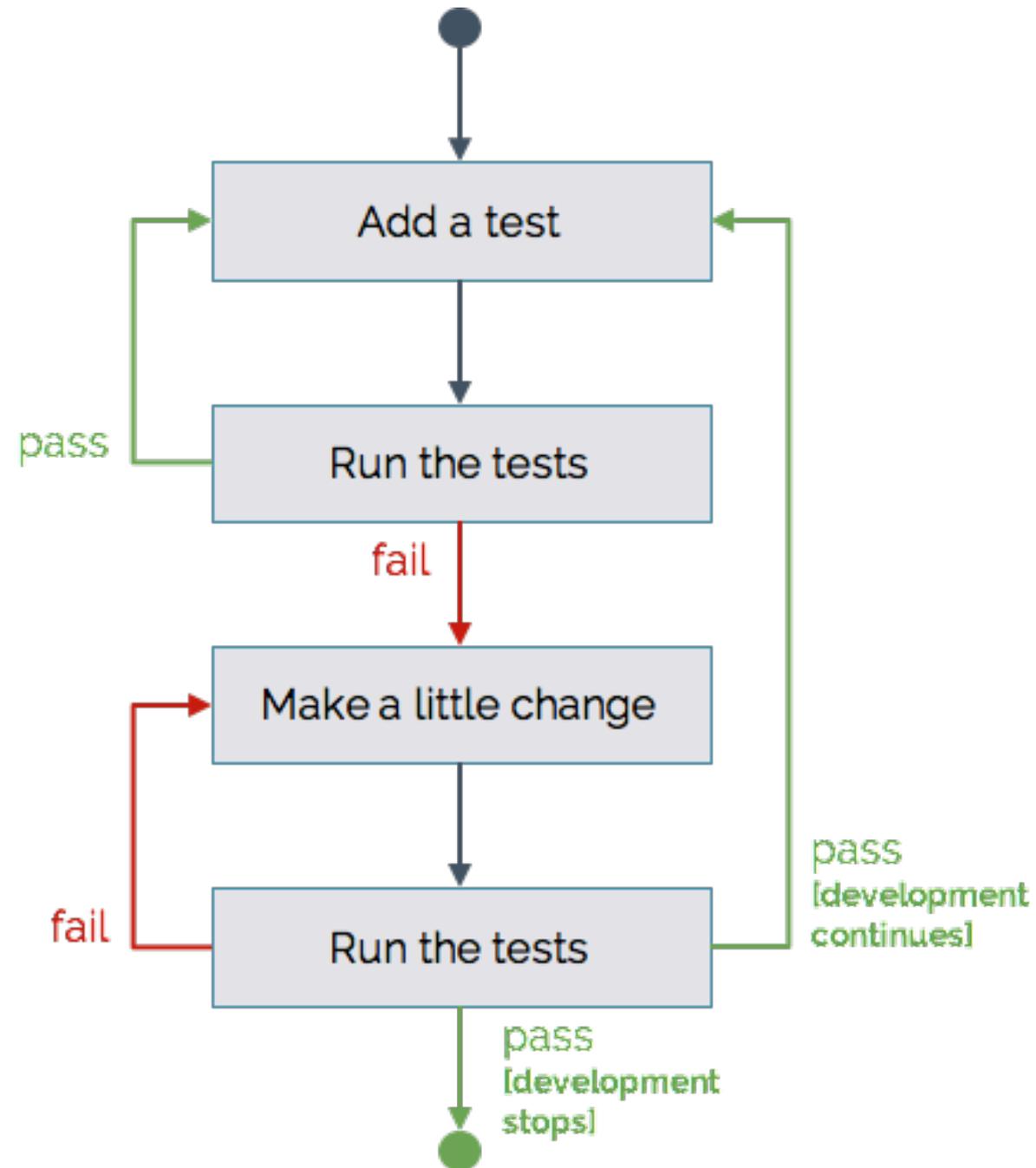
# Converge



```
$ kitchen converge
```

```
-----> Starting Kitchen (v1.21.2)
-----> Creating <server-centos-73>...
      setsebool: SELinux is disabled.
      Sending build context to Docker daemon 209.9 kB
      Step 1/16 : FROM centos:centos7
...
      Running handlers:
      Running handlers complete
      Chef Client finished, 1/1 resources updated in 01 seconds
      Downloading files from <server-centos-73>
      Finished converging <server-centos-73> (0m21.21s).
-----> Kitchen is finished. (1m3.05s)
```

# Test-Driven Development



# Verify the Kitchen



```
$ kitchen verify
```

```
...
-----> Verifying <server-centos-73>...
  Loaded ssh
[2018-06-11T04:06:59+00:00] ERROR: Cannot find a UUID for your node.

Profile: SSH Configuration (ssh)
Version: 0.1.0
Target:  ssh://kitchen@localhost:32768

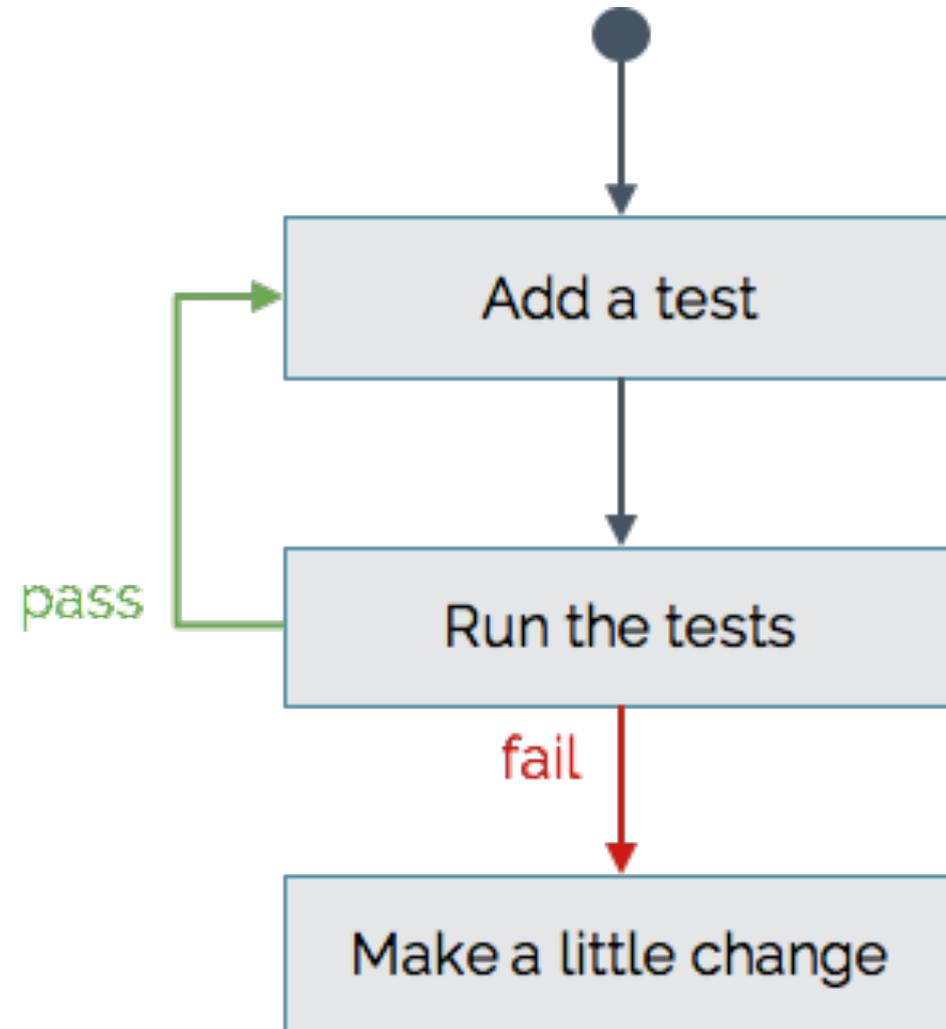
  x  sshd-1.0: SSH Version 2
    x  SSHD Configuration Protocol should cmp == 2

    expected: 2
      got:

    (compared using `cmp` matcher)

Profile Summary: 0 successful controls, 1 control failure, 0 controls skipped
Test Summary: 0 successful, 1 failure, 0 skipped
```

# Test-Driven Development



# Edit the SSH Configuration Template

```
~/cookbooks/ssh/templates/sshd_config.erb
```

```
#ListenAddress 0.0.0.0
```

```
#ListenAddress ::
```

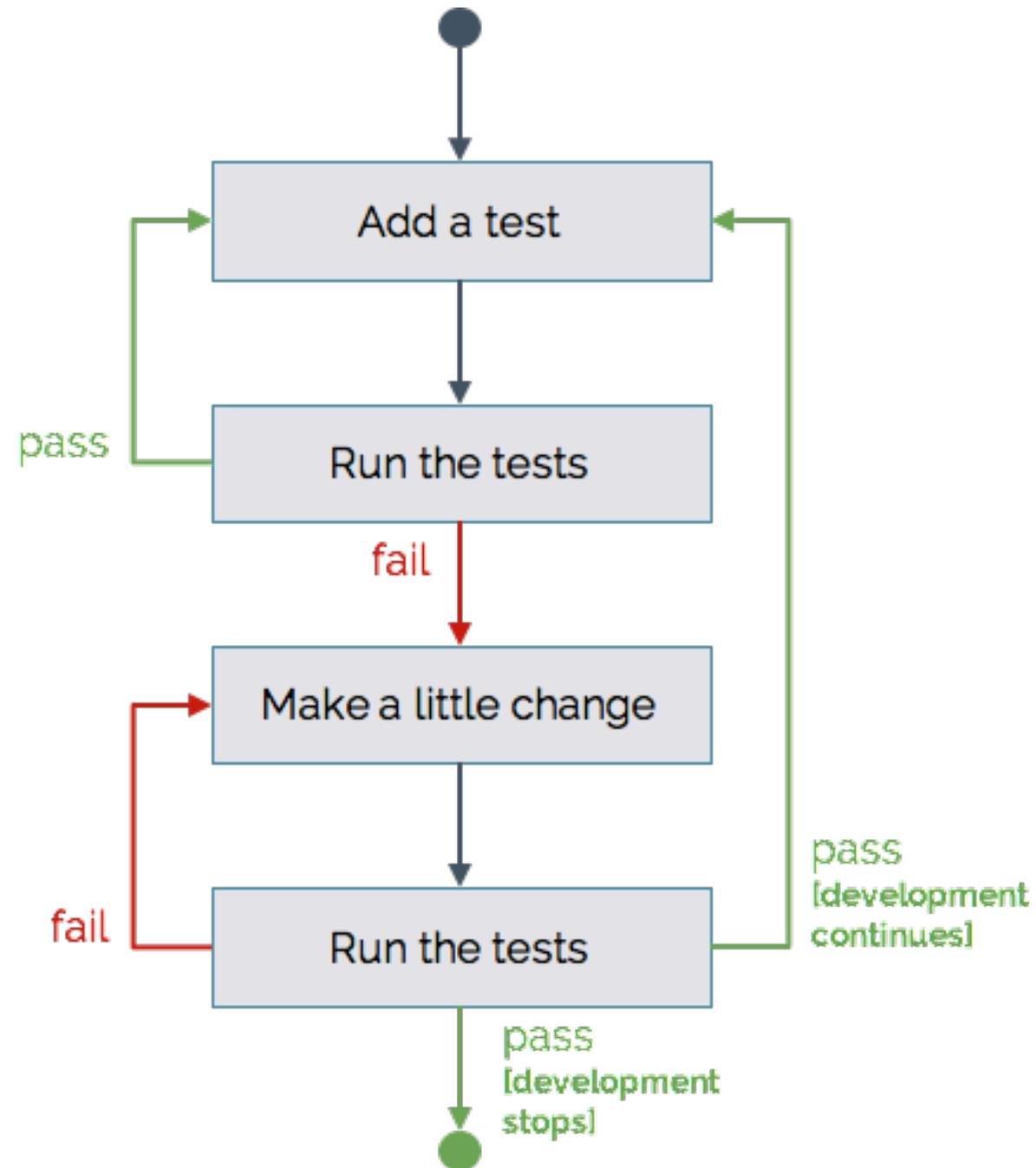
```
# The default requires explicit activation of protocol 1
```

```
- #Protocol 2
```

```
+ Protocol 2
```

```
# HostKey for protocol version 1
```

# Test-Driven Development



# Converge (apply our new cookbook change)



```
$ kitchen converge
```

```
-----> Starting Kitchen (v1.21.2)
-----> Converging <server-centos-73>...
...
      # The default requires explicit activation of protocol 1
      -#Protocol 2
      +Protocol 2
...
Running handlers:
Running handlers complete
Chef Client finished, 1/1 resources updated in 00 seconds
Downloading files from <server-centos-73>
Finished converging <server-centos-73> (0m4.74s).
-----> Kitchen is finished. (0m6.71s)
```

# Verify the Kitchen



```
$ kitchen verify
```

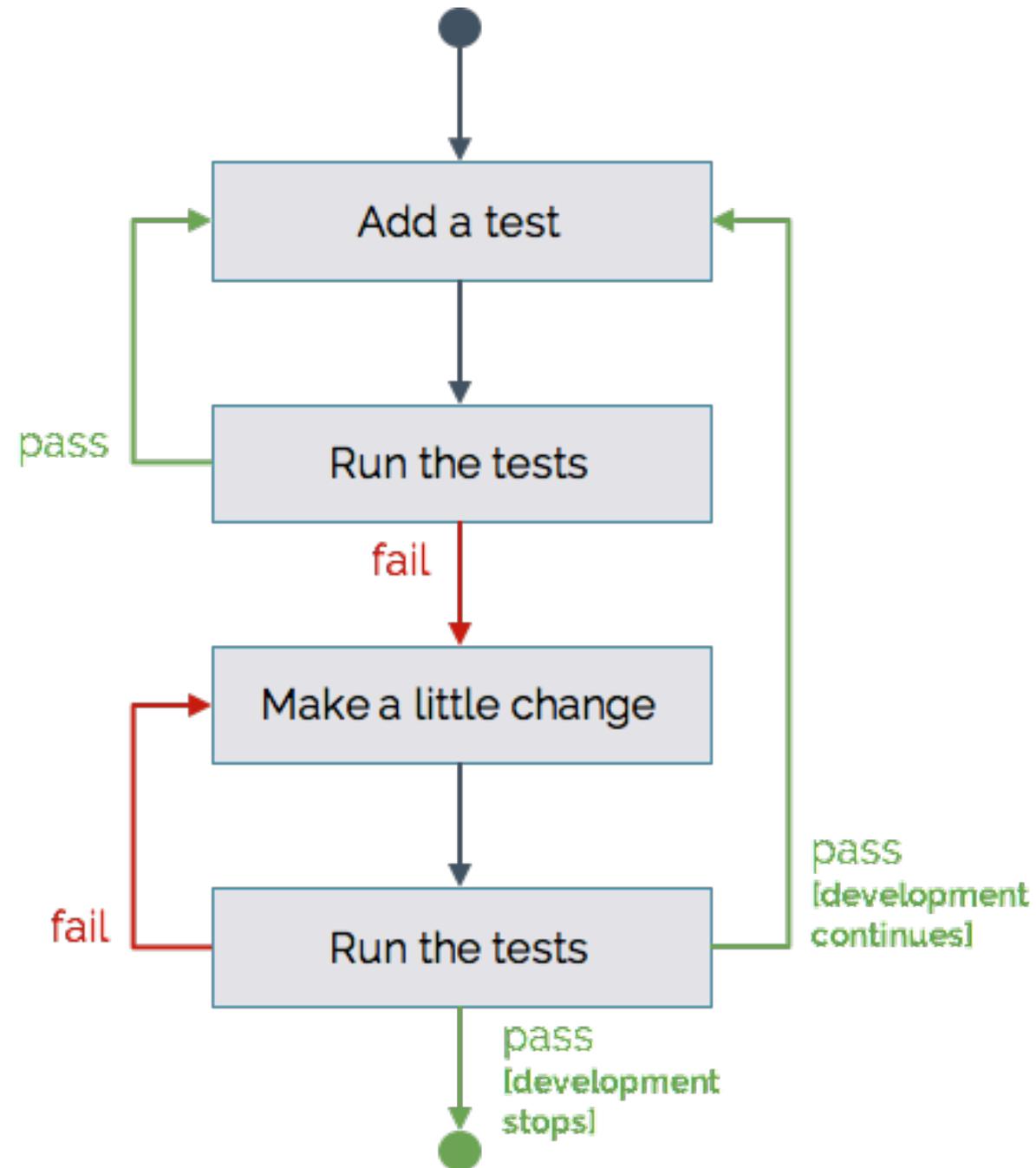
```
-----> Starting Kitchen (v1.21.2)
-----> Setting up <server-centos-73>...
         Finished setting up <server-centos-73> (0m0.00s).
-----> Verifying <server-centos-73>...
         Loaded ssh
[2018-06-11T04:09:36+00:00] ERROR: Cannot find a UUID for your node.
```

```
Profile: SSH Configuration (ssh)
Version: 0.1.0
Target:  ssh://kitchen@localhost:32768
```

- ✓ sshd-1.0: SSH Version 2
- ✓ SSHD Configuration Protocol should cmp == 2

```
Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
Test Summary: 1 successful, 0 failures, 0 skipped
         Finished verifying <server-centos-73> (0m1.20s).
-----> Kitchen is finished. (0m3.17s)
```

# Test-Driven Development



# End-to-End Kitchen Test (1 of 2)



```
$ kitchen test
```

```
-----> Starting Kitchen (v1.21.2)
-----> Cleaning up any prior instances of <server-centos-73>
-----> Destroying <server-centos-73>...
...
-----> Testing <server-centos-73>
-----> Creating <server-centos-73>...
...
-----> Converging <server-centos-73>...
...
-----> Installing Chef Omnibus (install only if missing)
...
```

# End-to-End Kitchen Test (2 of 2)



```
$ kitchen test
```

```
-----> Setting up <server-centos-73>...
```

```
...
```

```
-----> Verifying <server-centos-73>...
```

```
...
```

```
Target:  ssh://kitchen@localhost:32769
```

```
✓ sshd-1.0: SSH Version 2
```

```
✓ SSHD Configuration Protocol should cmp == 2
```

```
Profile Summary: 1 successful control, 0 control failures, 0 controls skipped
```

```
Test Summary: 1 successful, 0 failures, 0 skipped
```

```
...
```

```
-----> Destroying <server-centos-73>...
```

```
...
```

```
-----> Kitchen is finished. (0m23.89s)
```

# What's next?

- Test-driven development cycle is complete
- Deploy the change (with confidence!)

# Remediate with Chef



```
$ run_chef "recipe[ssh::server],recipe[audit::default]"
```

```
Starting Chef Client, version 14.1.12
```

```
...
```

```
Synchronizing Cookbooks:
```

```
- audit (7.0.0)
```

```
- ssh (0.1.0)
```

```
...
```

```
-#Protocol 2
```

```
+Protocol 2
```

```
...
```

```
Chef Client finished, 1/3 resources updated in 04 seconds
```

# Verify Converge Status in Automate

Resources

Run List

Attributes

Total Resources

≡ 3

Failed



0

Successful



1

Unchanged



2

Unprocessed



0

Status	Step	Type	Name	Action	Cookbook	View
✓	1/3	inspec_gem	inspec	install	audit	--
✓	2/3	template	/etc/ssh/sshd_config	create	ssh	--
?	3/3	inspec_gem	inspec	nothing	audit	--

# Verify Compliance Status in Automate

✔ Your System is Compliant

Report Metadata +

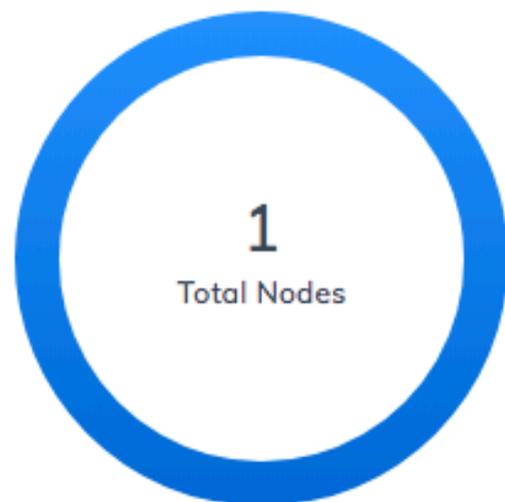
Overview

1 Nodes

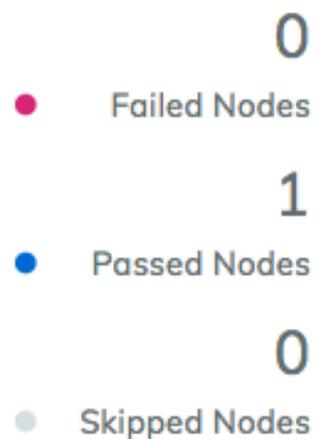
1 Profiles

Node Status

Profile Status



Global Compliance



Severity of Node Failures

# Verify Compliance Status in Automate

Summary of Compliance Status:

- Total Controls: 1
- Critical Controls: 0
- Major Controls: 0
- Minor Controls: 0
- Skipped Controls: 0
- Passed Controls: 1

Control	Test Results	Severity	Root Profile
✓ sshd-1.0: SSH Version 2	// 1	CRITICAL (0.7)	ssh

An orange arrow points to a blue plus sign icon in the bottom right corner of the table row, indicating a link to view details for the control.

# Verify Compliance Status in Automate

Control	Test Results	Severity	Root Profile
<input checked="" type="checkbox"/> sshd-1.0: SSH Version 2	1	CRITICAL (0.7)	ssh

[View Source](#)

passed test result a few seconds

```
SSHD Configuration Protocol should cmp == 2
```

# Ready for more?



- **Learn Chef Rally**  
[learn.chef.io](https://learn.chef.io)
- **Classroom-style Training**

@nathenharvey

# Get started with **CHEF**AUTOMATE

- <https://learn.chef.io/modules/chef-automate-pilot/>  
*Set up your own demo environment*
- <https://automate.chef.io/>  
*Install on-prem, generate a trial license*
- AWS OpsWorks for Chef Automate  
*Managed service*
- AWS and Azure Marketplace

# Join us on Slack!

- <http://community-slack.chef.io>
- #general (for Chef stuff)
- #inspec





**CHEF**™