Introduction to Continuous Compliance and Remediation

Nathen Harvey

Chef

@nathenharvey

Introductions

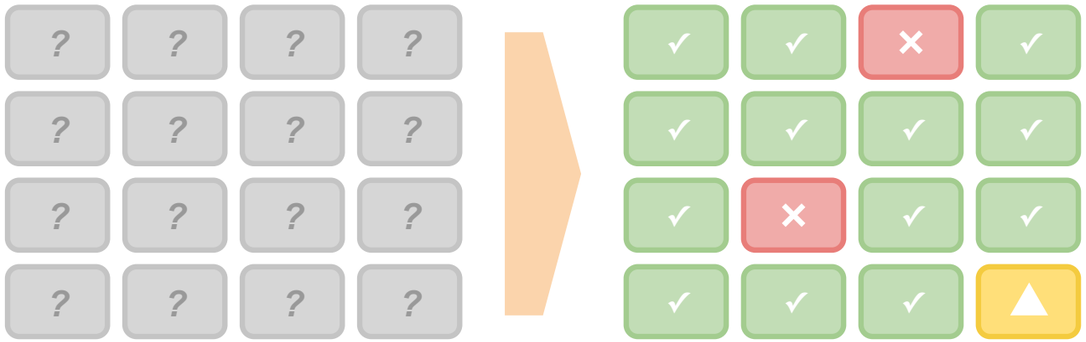
- Who are you?
- Why are you here?
- What was the last show you binge watched?

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

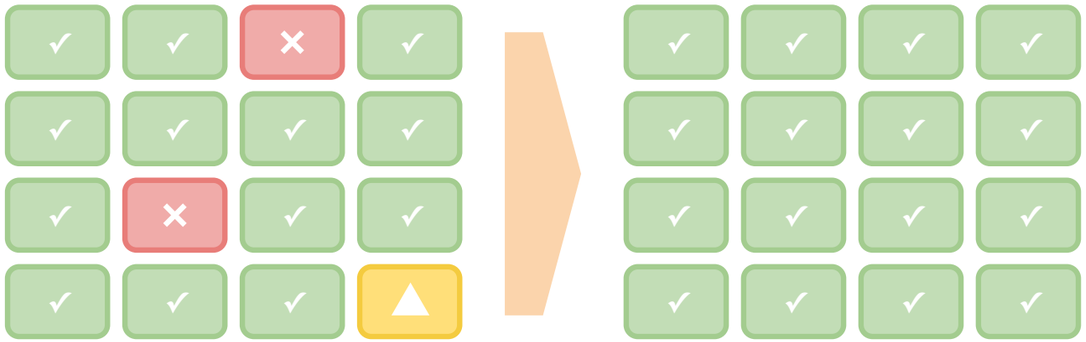
> Continuous visibility means that you enter into audits knowing the outcome.
>
> Jon Williams, NIU

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

Web & Media Giant

Can patch 250,000 nodes within 6 hours of a patch being made available

A tale of three personas...

… and a single language.

Security

Compliance

Language

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

Learning Environment

CHEF AUTOMATE

Nodes

Learning Environment

CHEF AUTOMATE

Node data

Nodes

Learning Environment

CHEF AUTOMATE

Node data

Laptop

Nodes

Learning Environment

CHEF AUTOMATE

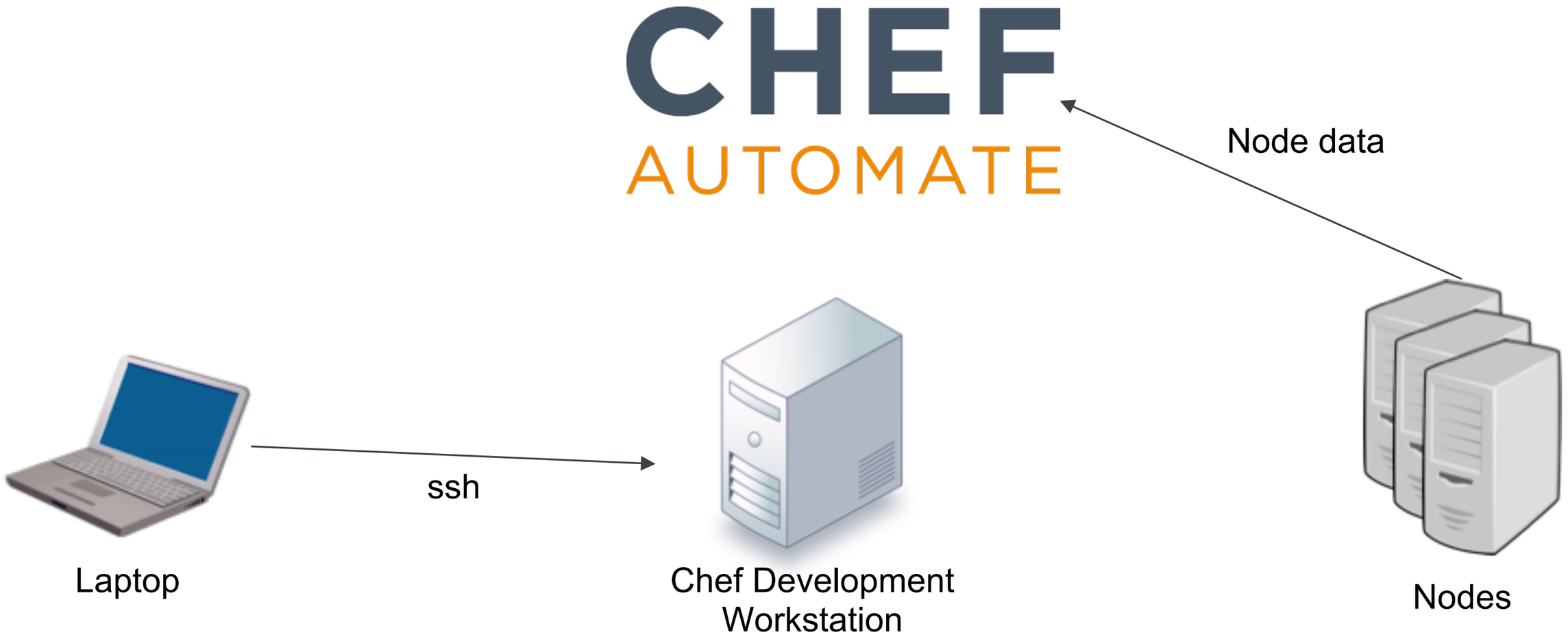
Node data

Laptop

Chef Development Workstation

Nodes

Learning Environment



Learning Environment

CHEF AUTOMATE

Node data

ssh

Laptop

Nodes

Chef Development Workstation

Access the Learning Environment

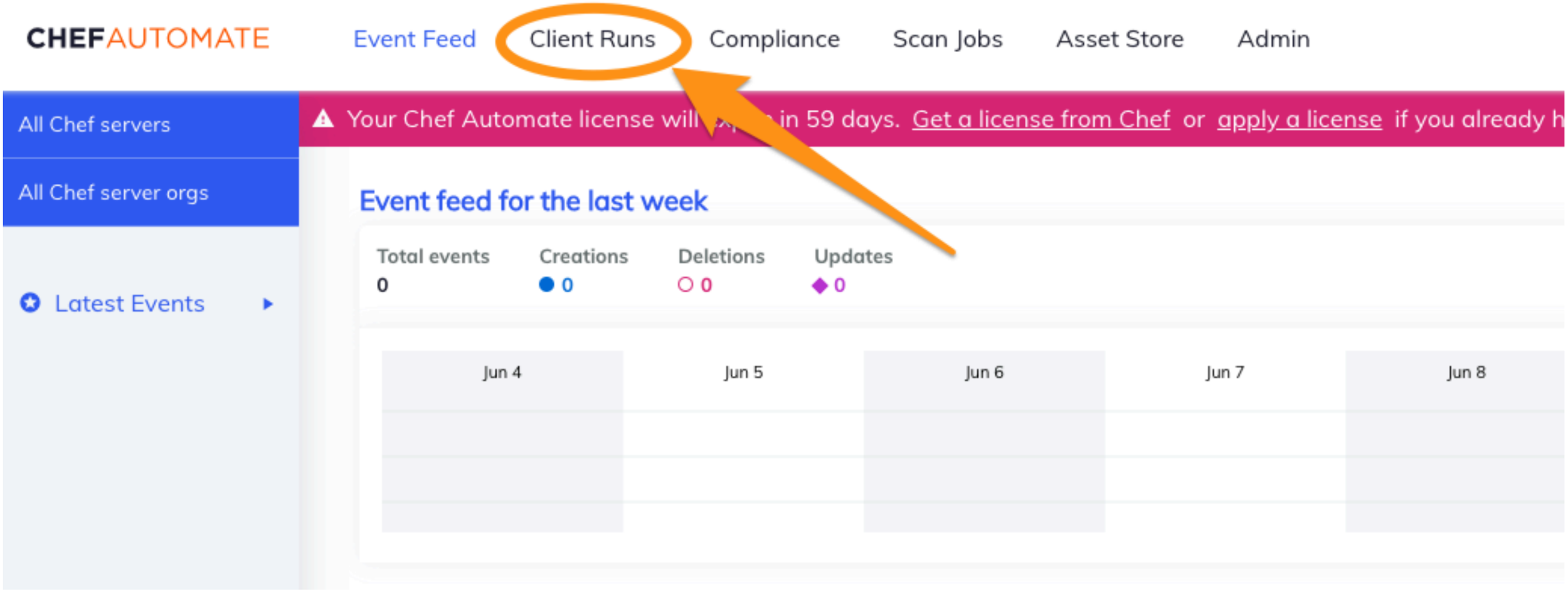
- Login to Chef Automate
- Find your workstation/node
- Find your workstation's IP address
- Login to your workstation

Let's log in to Chef Automate!

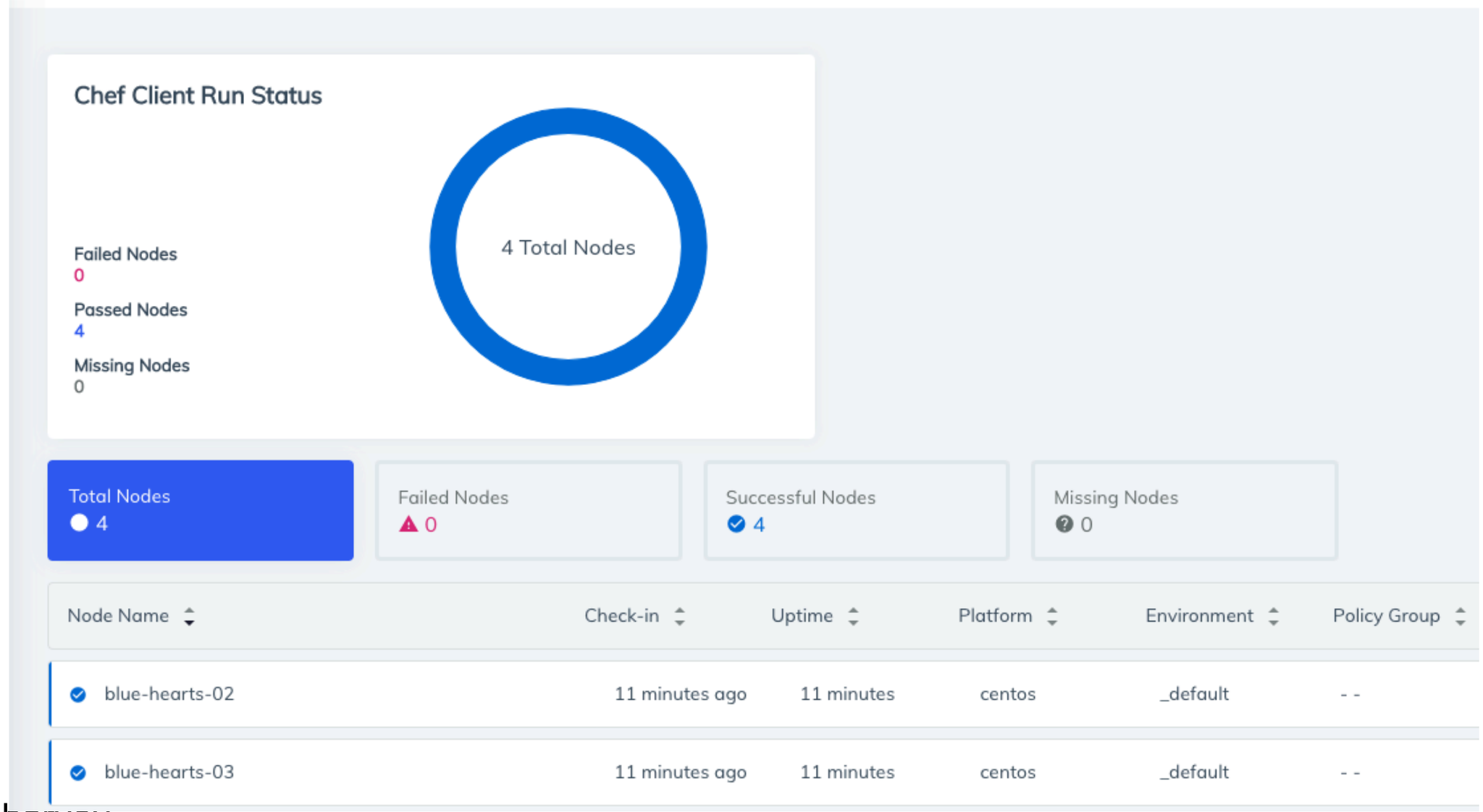
- **https://velocity-workshop.community.chefdemo.net**
- Uses a self-signed certificate in this lab

- Username: `admin`
- Password: `chef-automate`

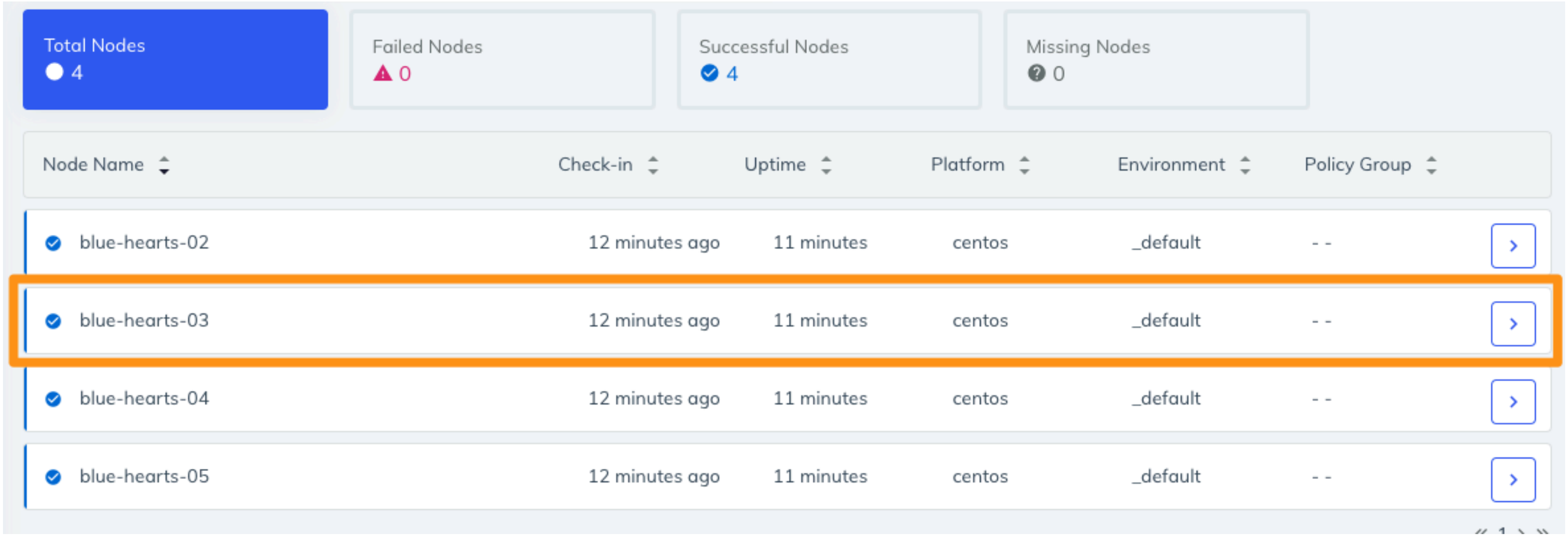
Browse to your node



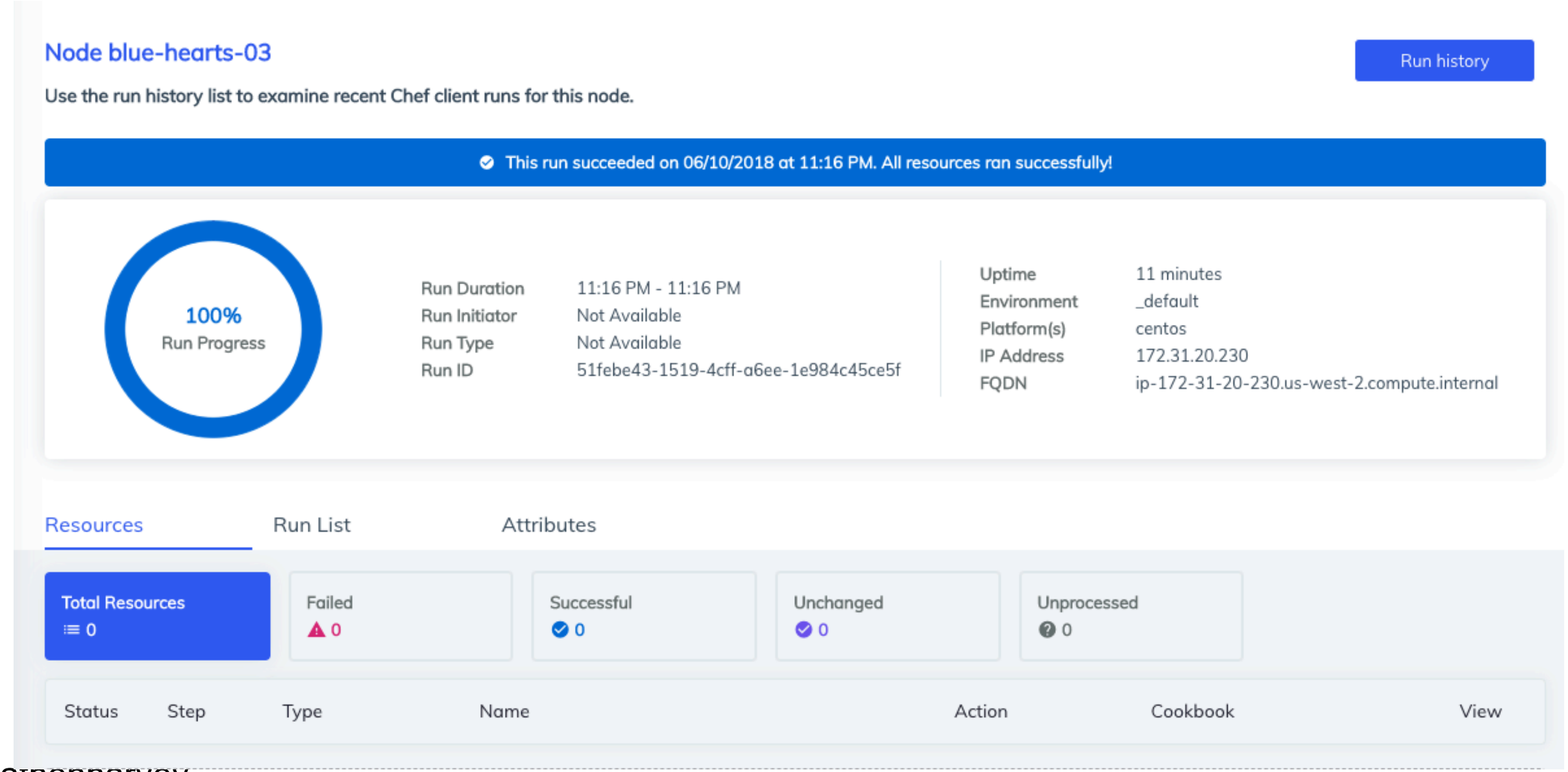
Browse to your node



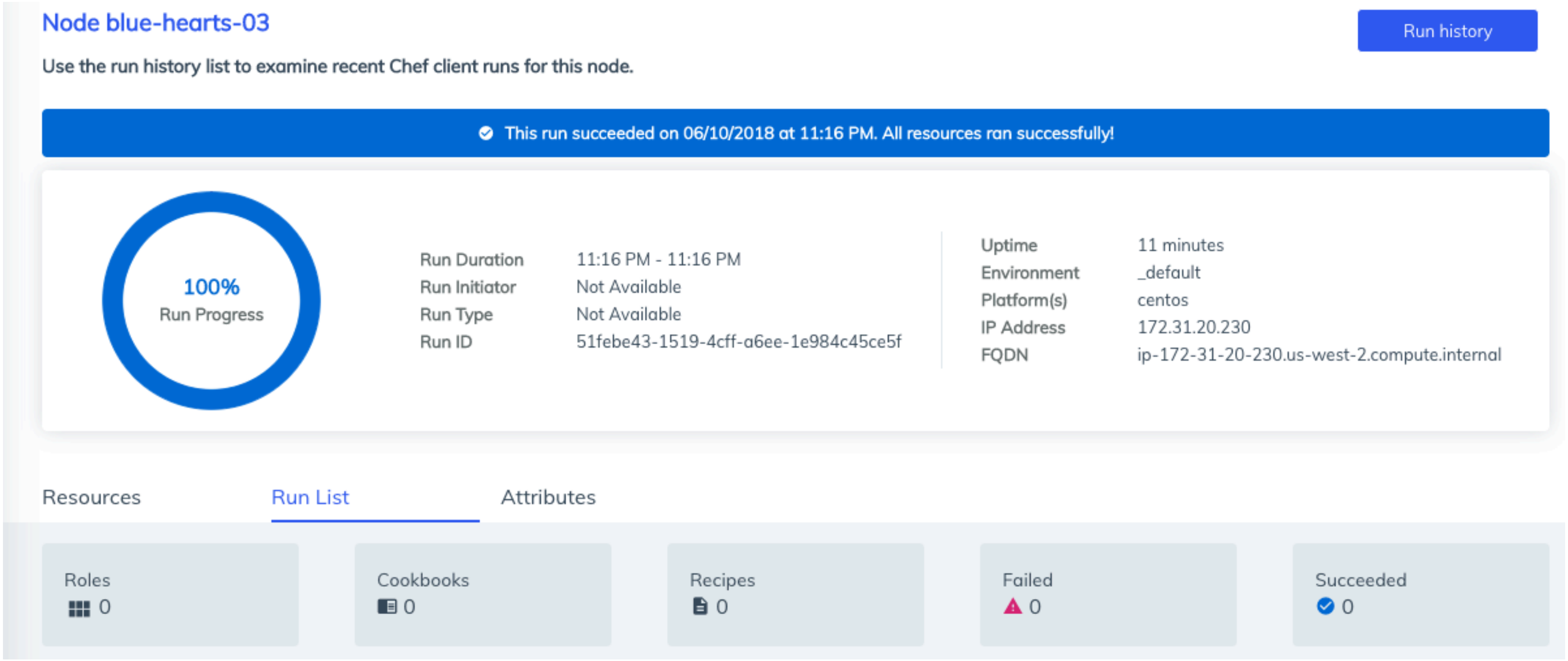
Browse to your node



View details of your node

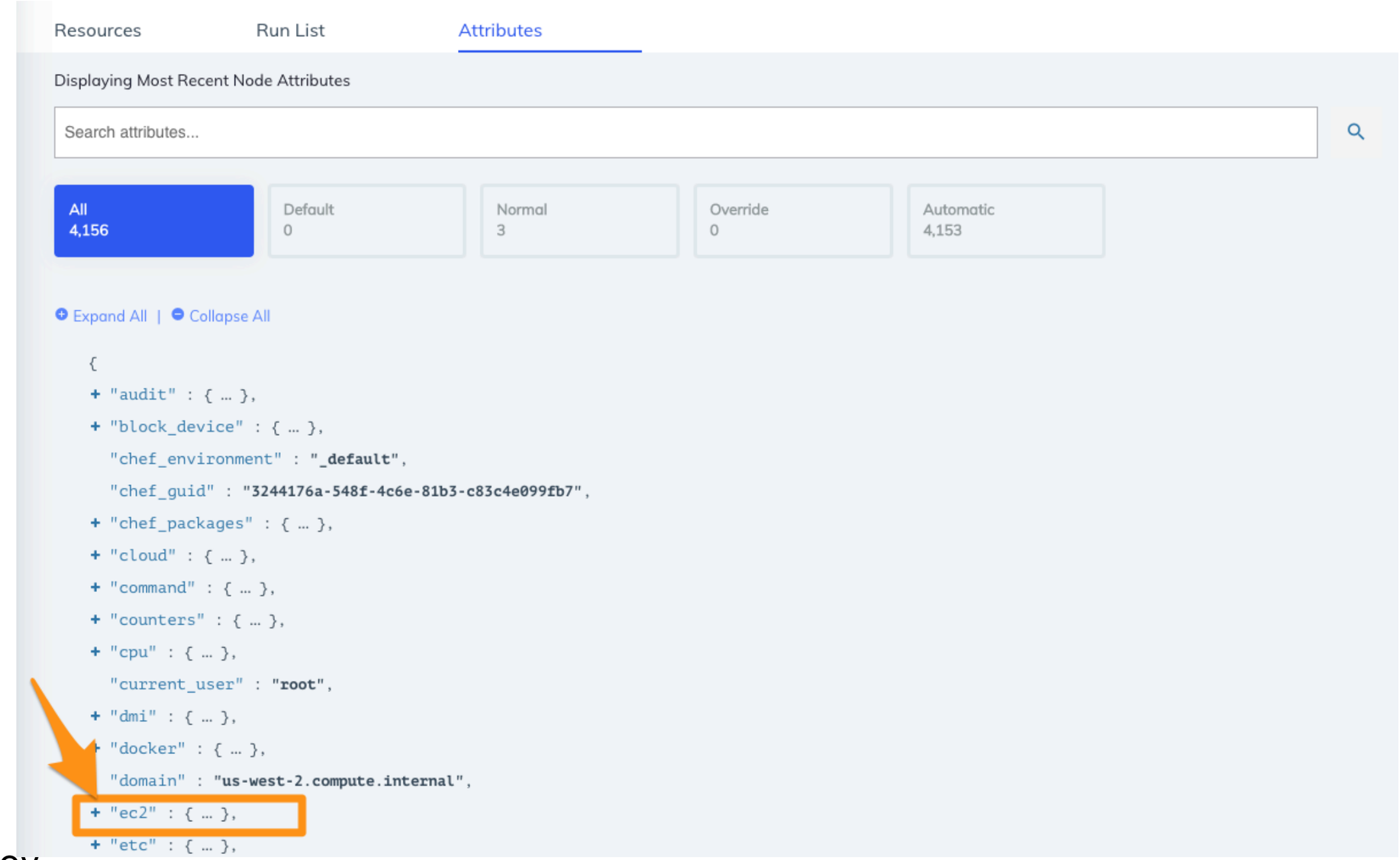


View details of your node



@nathenharvey

View details of your node



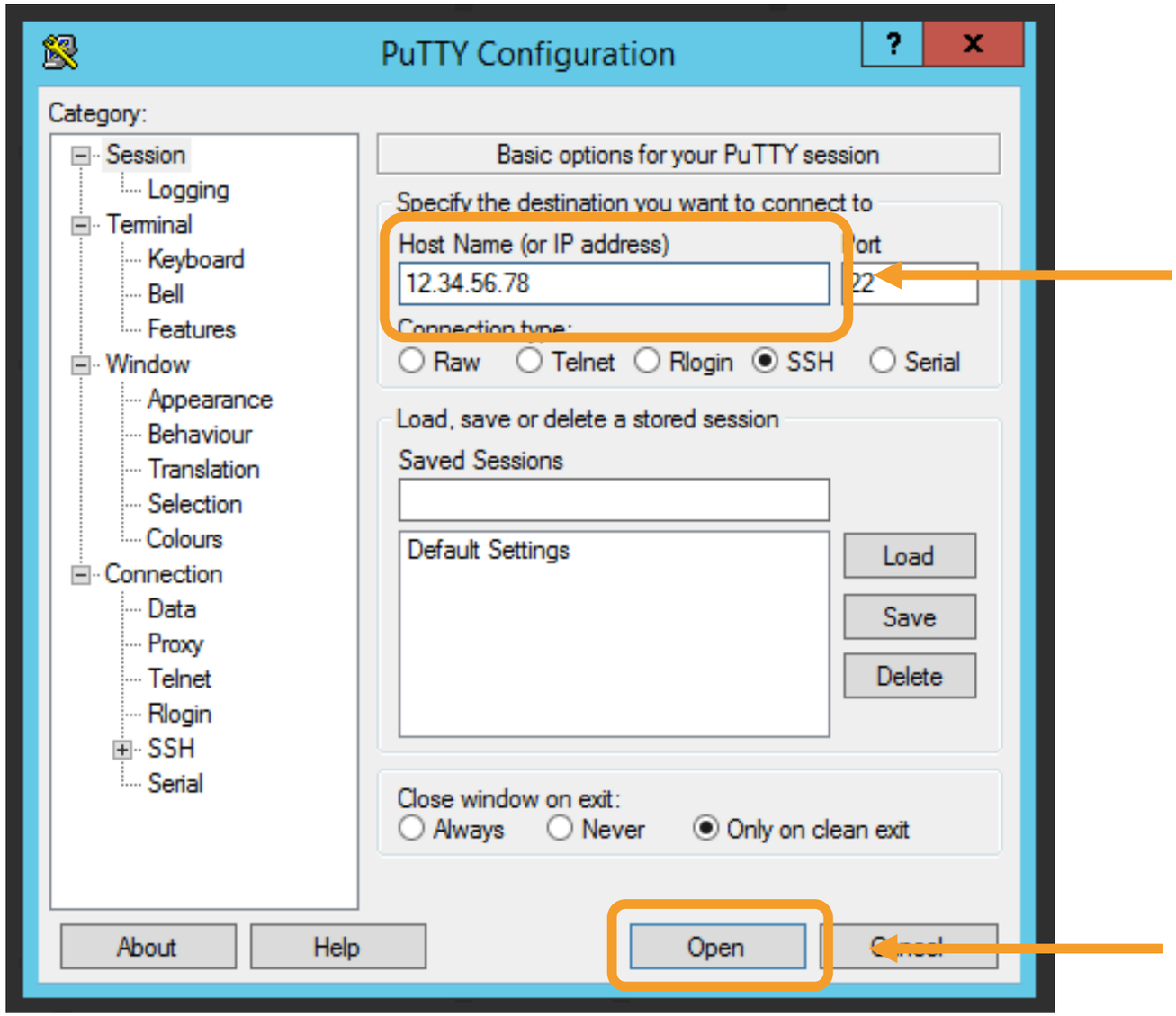
Find the IP of your node

```
+ "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
```

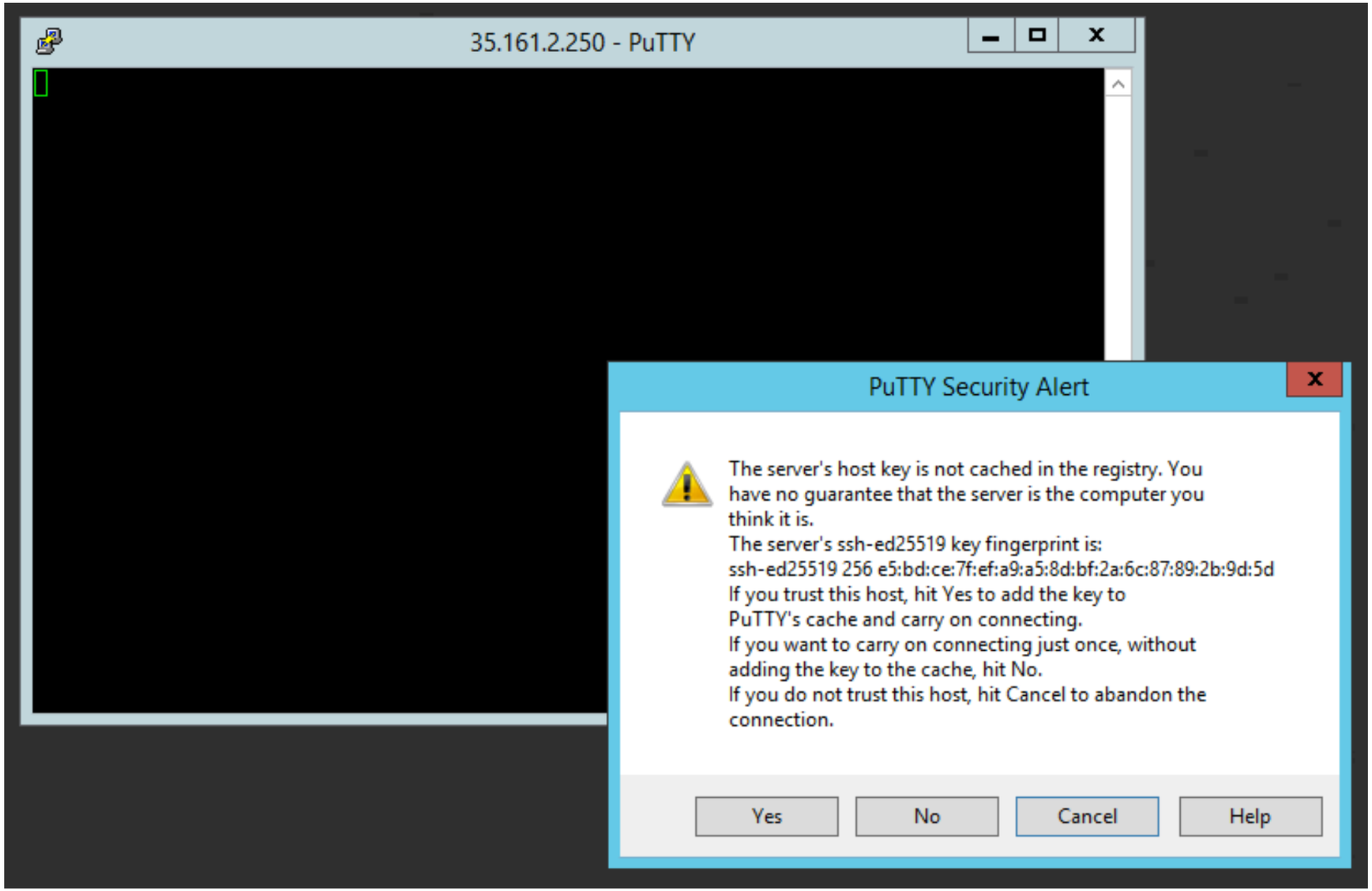
```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.
ECDSA key fingerprint is SHA256:zAtoeO29XbhRNvwg542cuh4qsKCEaX8hNIlEOCbgd3I.
Are you sure you want to continue connecting (yes/no)?
```

Log in to your remote workstation

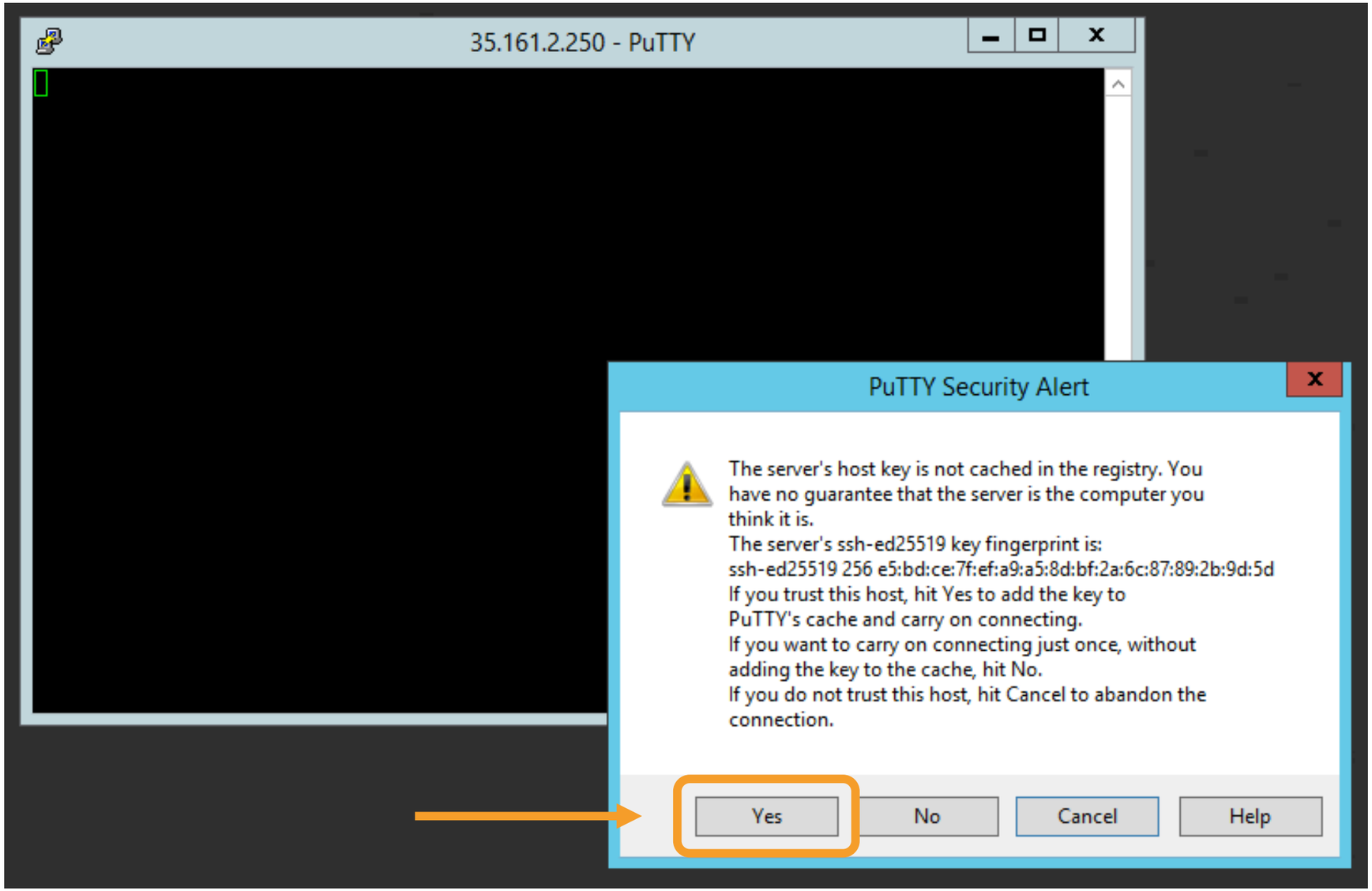
```
$  ssh chef@12.34.56.78
```

```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.
ECDSA key fingerprint is SHA256:zAtoeO29XbhRNvwg542cuh4qsKCEaX8hNIlEOCbgd3I.
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
```

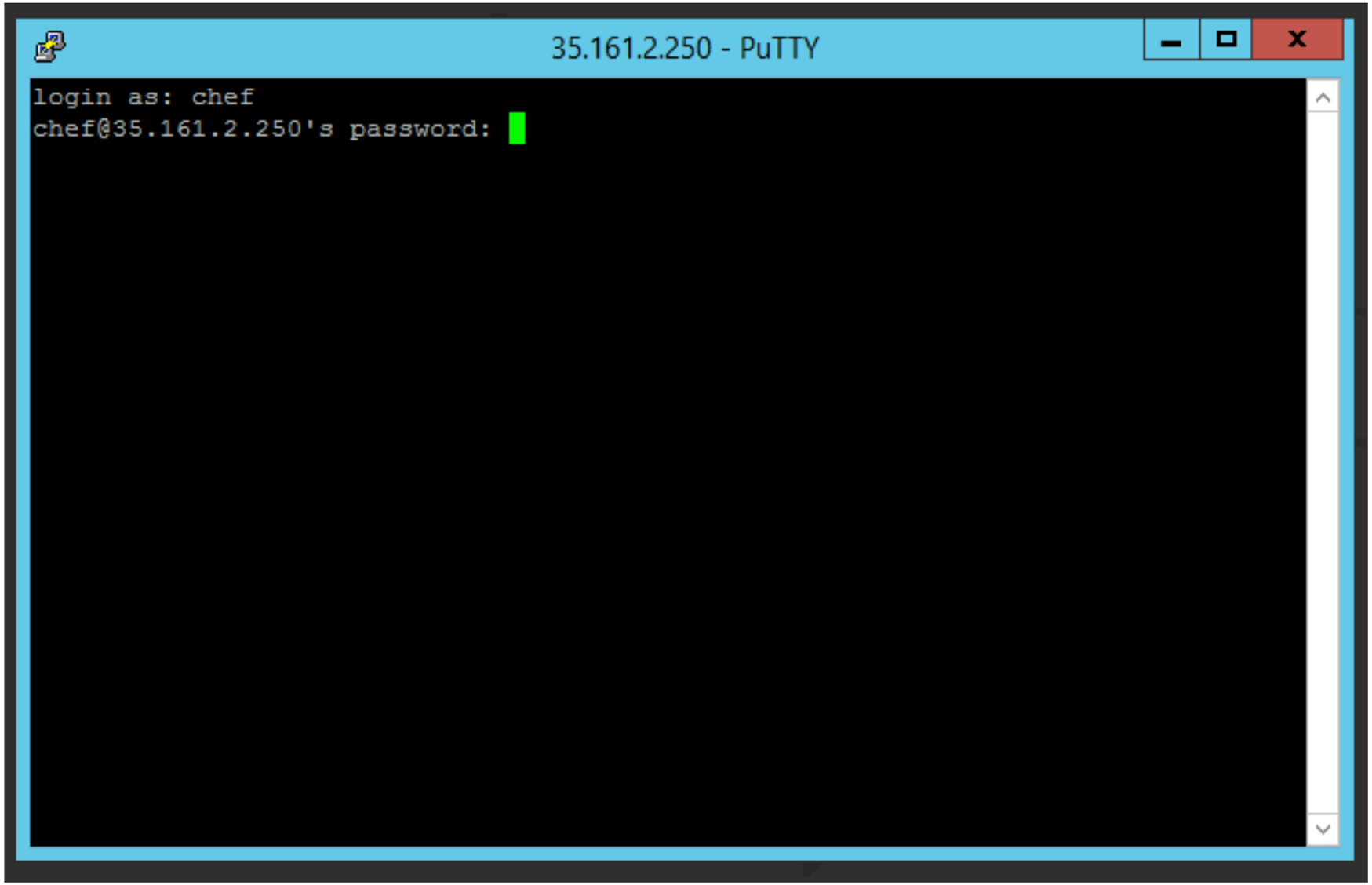
```
The authenticity of host 12.34.56.78 (12.34.56.78)' can't be established.
ECDSA key fingerprint is SHA256:zAtoeO29XbhRNvwg542cuh4qsKCEaX8hNIlEOCbgd3I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '12.34.56.78' (ECDSA) to the list of known hosts.
chef@12.34.56.78's 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:zAtoeO29XbhRNvwg542cuh4qsKCEaX8hNIlEOCbgd3I.
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

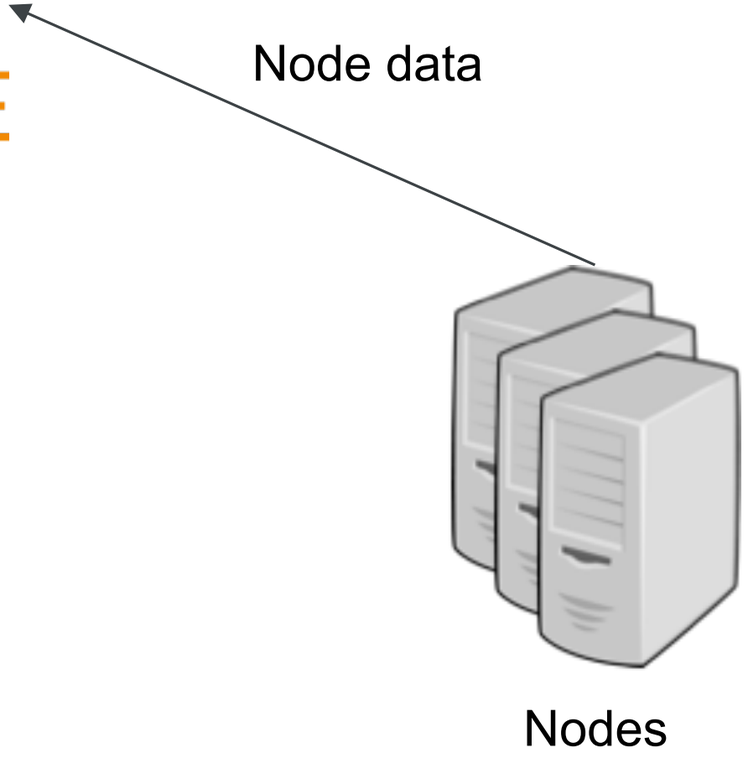
InSpec

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



Go home

```
$ cd ~
```

Run chef

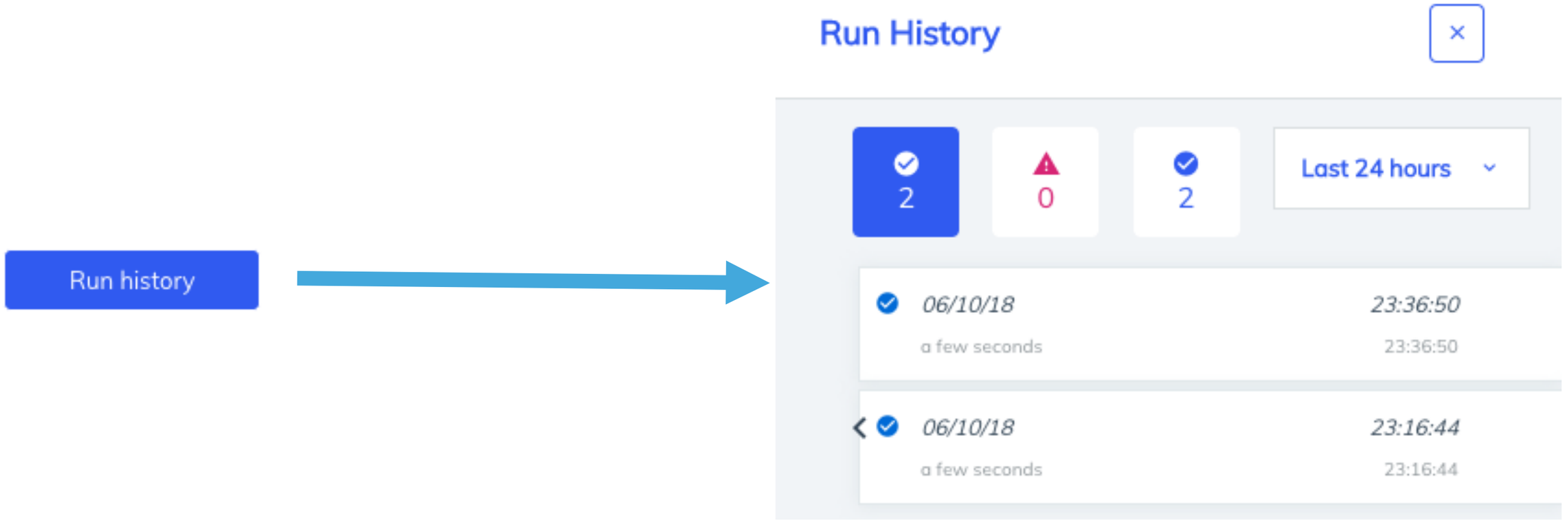
```
$ run_chef
```

```
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
```

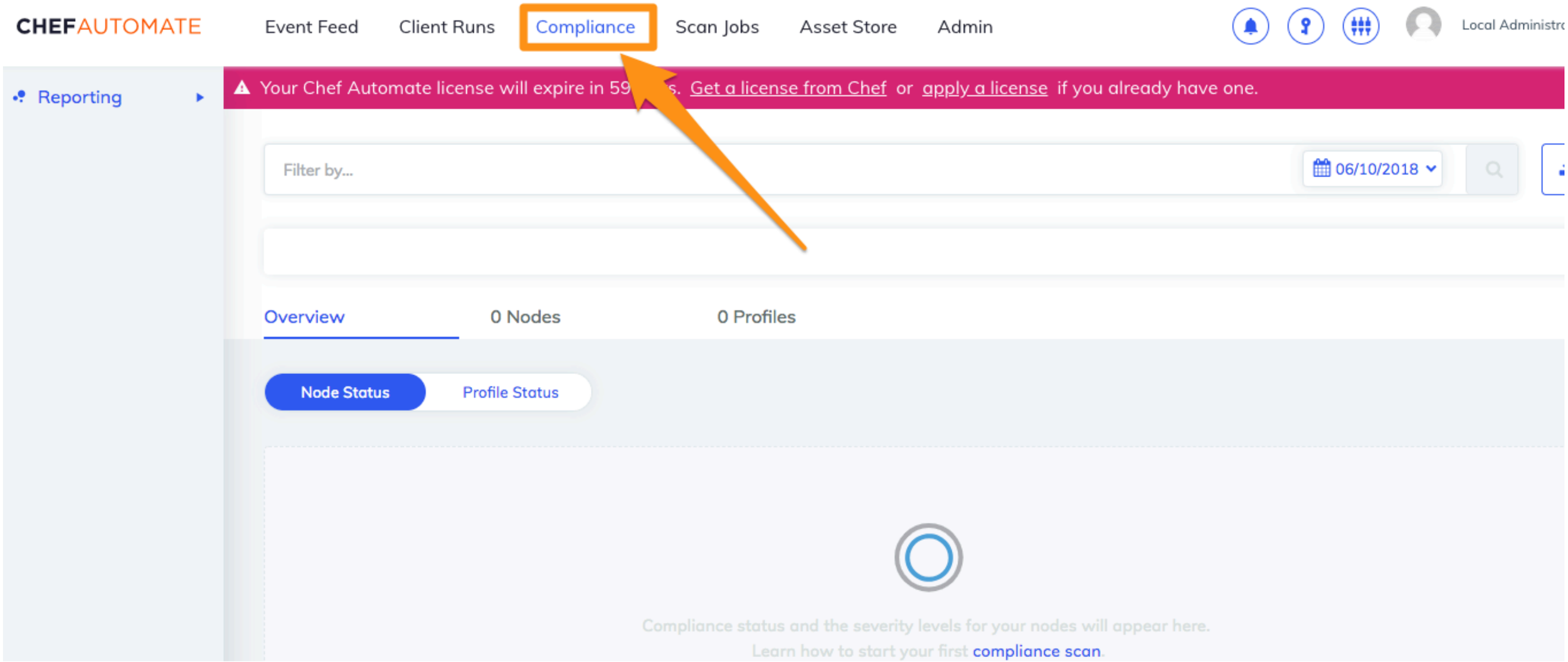
Check the converge status in Automate

Check the converge status in Automate

Check the converge status in Automate

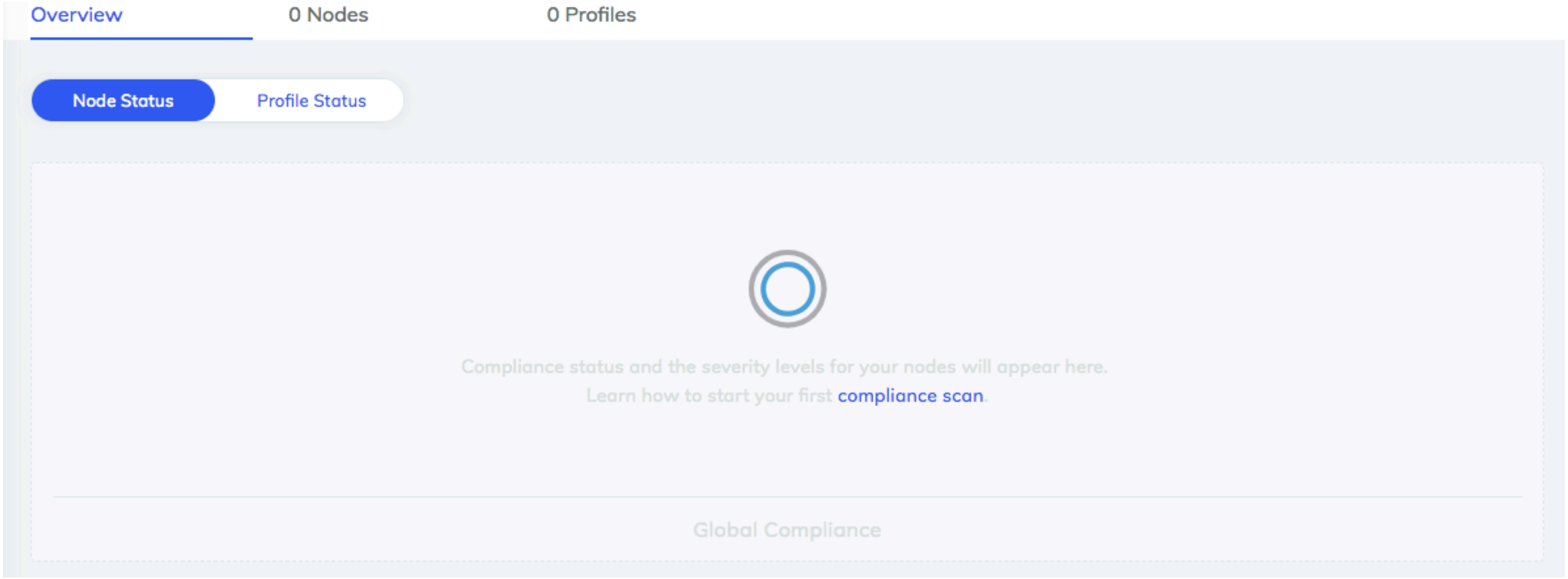


Compliance data in Automate



@nathenharvey

Compliance data in Automate



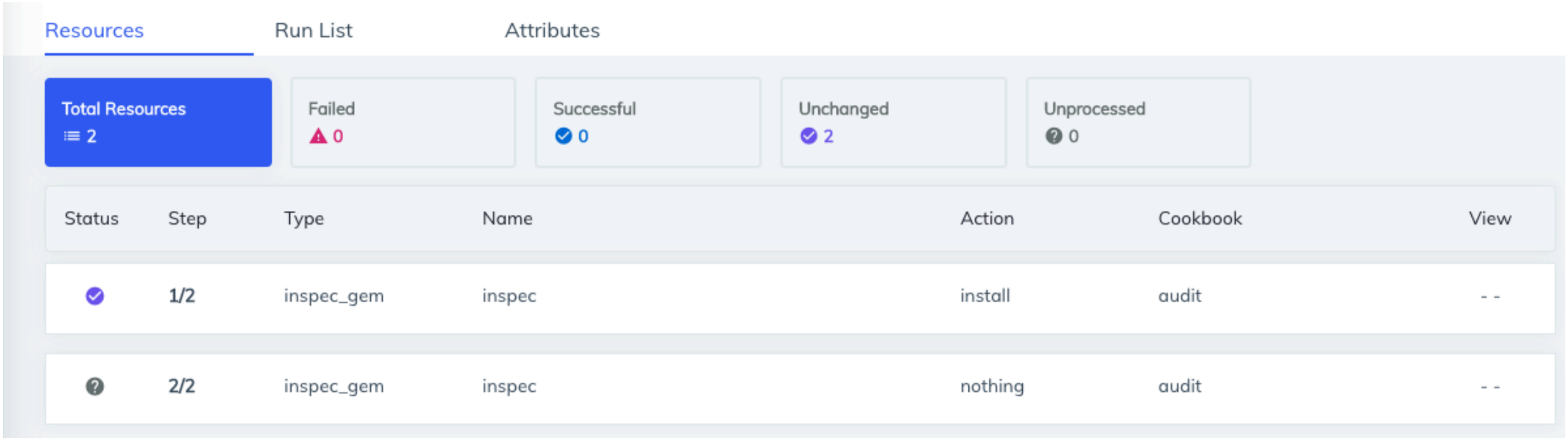
@nathenharvey

Run Chef with the audit cookbook

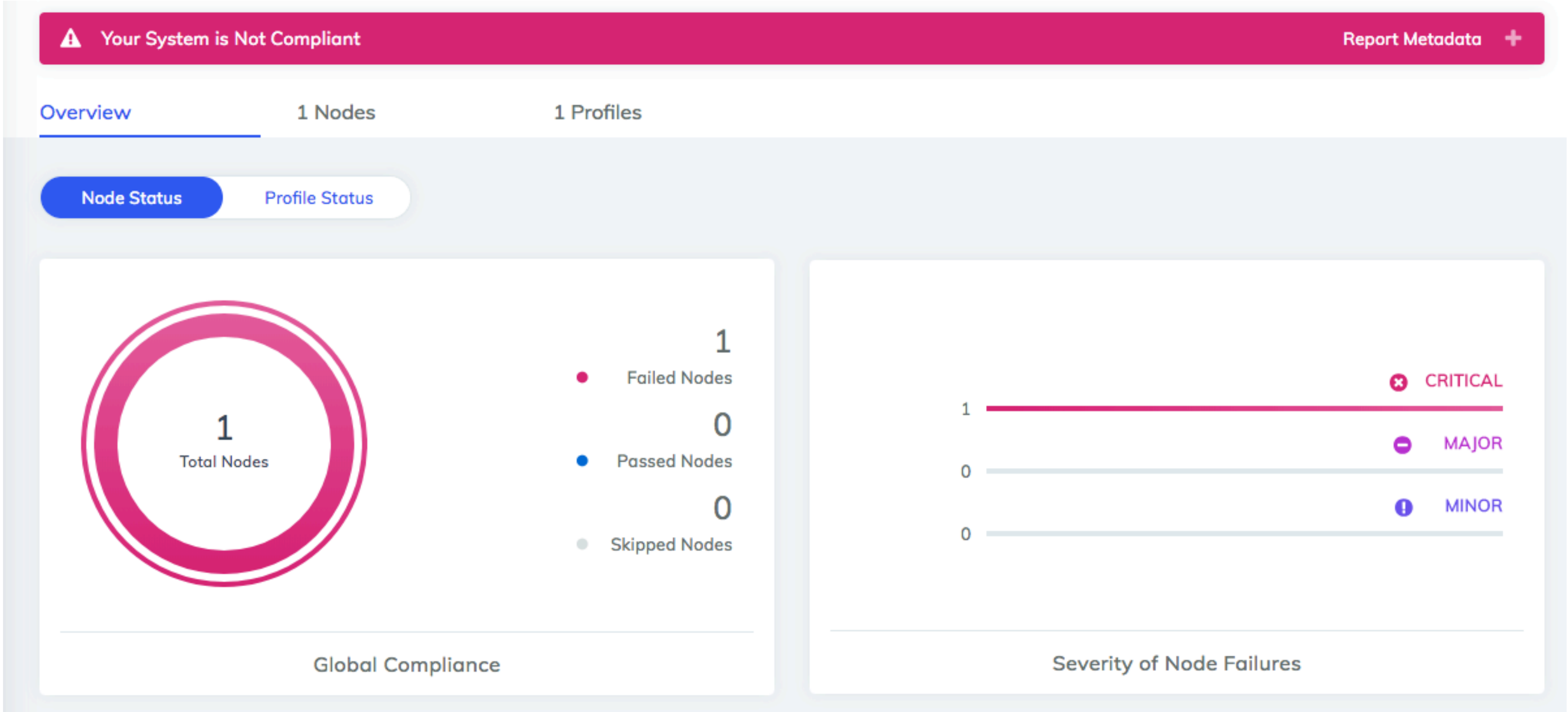
```
$ run_chef "recipe[audit::default]"
```

```
Starting Chef Client, version 14.1.12
...
  - Chef::Handler::AuditReport
Running handlers complete
Chef Client finished, 0/2 resources updated in 03 seconds
```

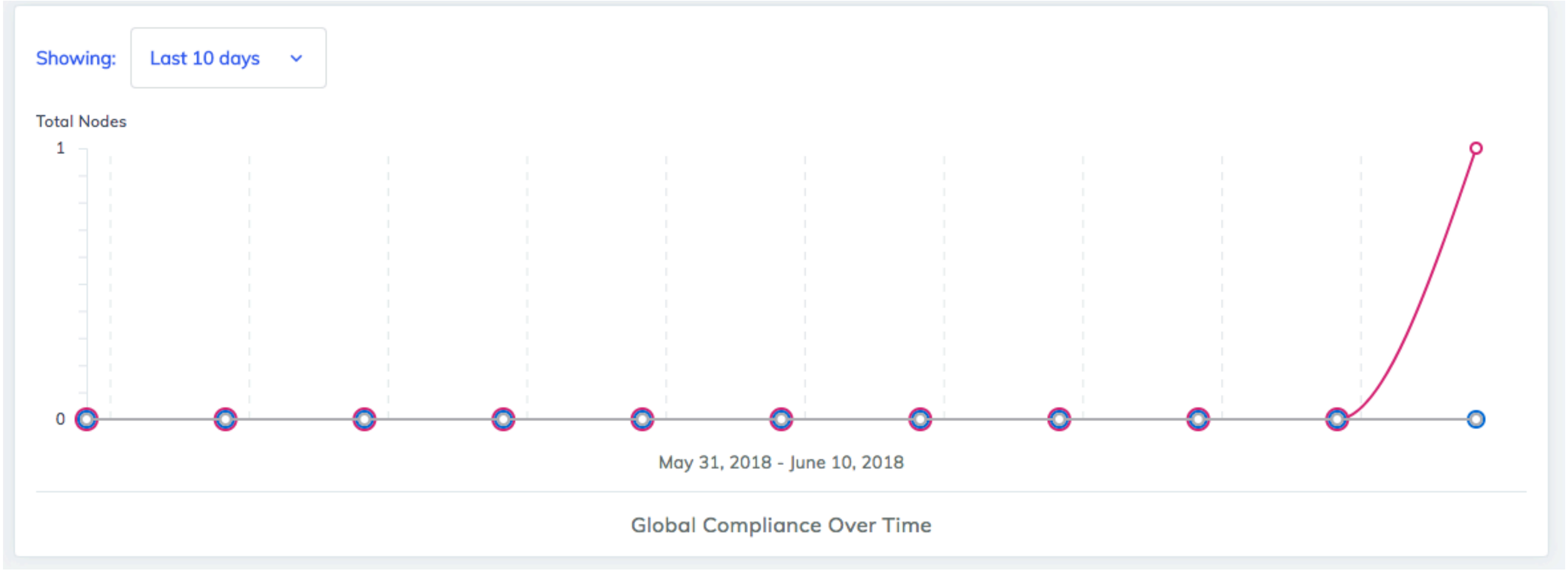
Check the converge status in Automate



Check the compliance status in Automate

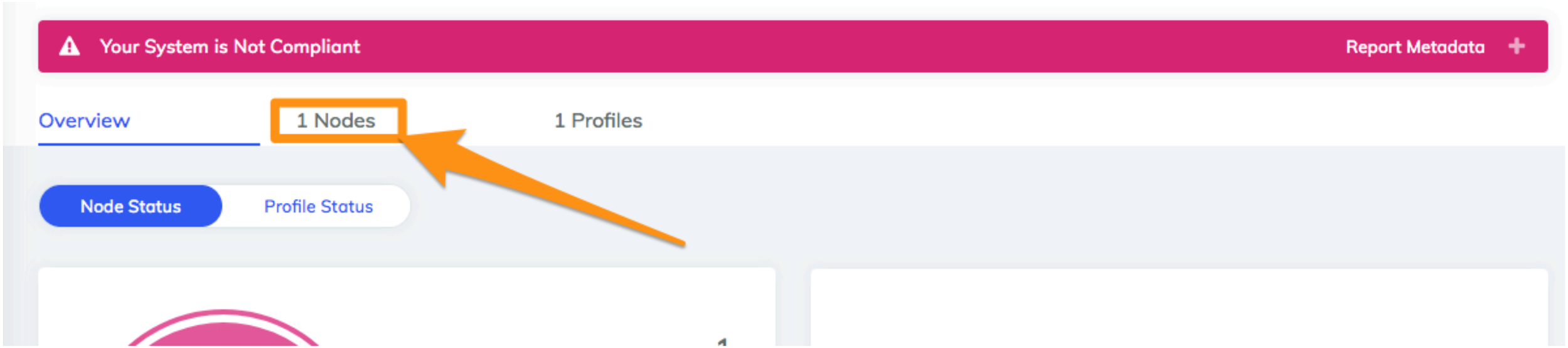


Check the compliance status in Automate

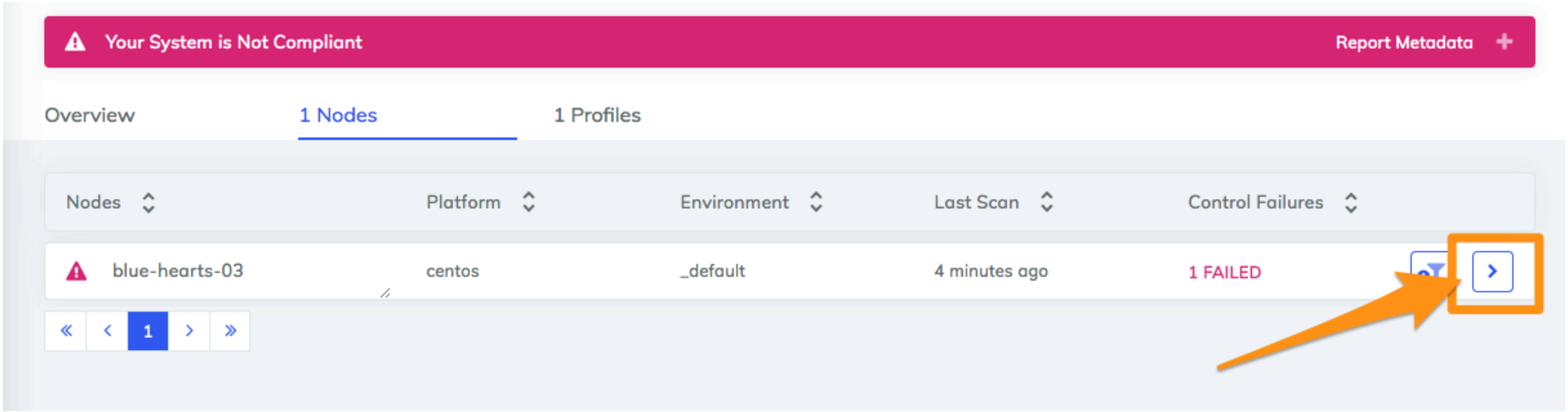


@nathenharvey

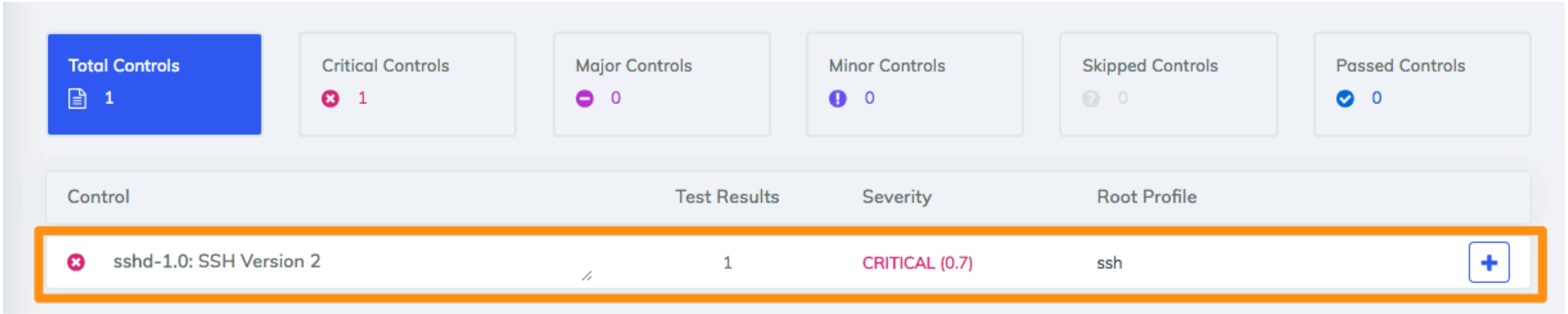
Check the compliance status in Automate



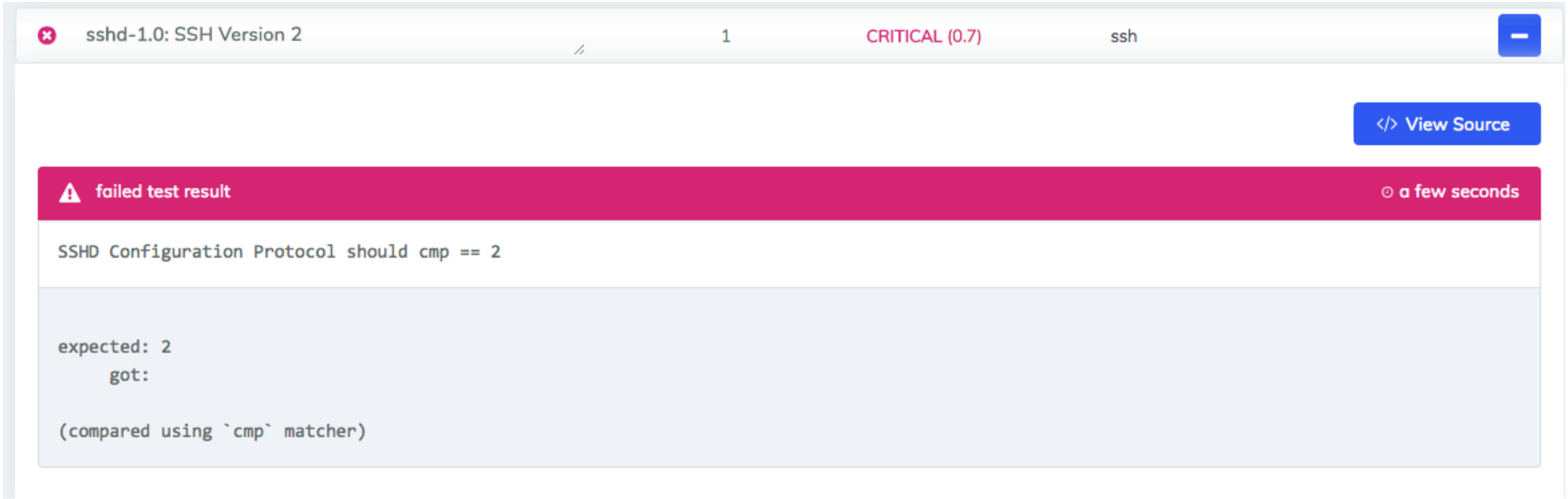
Check the compliance status in Automate



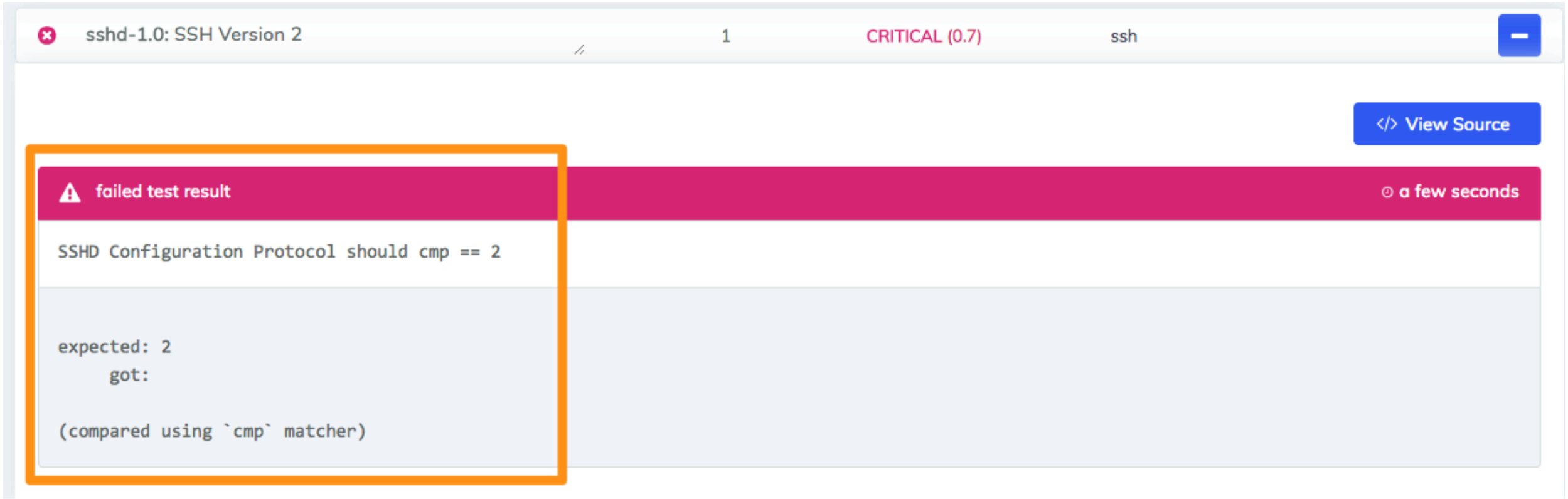
Check the compliance status in Automate



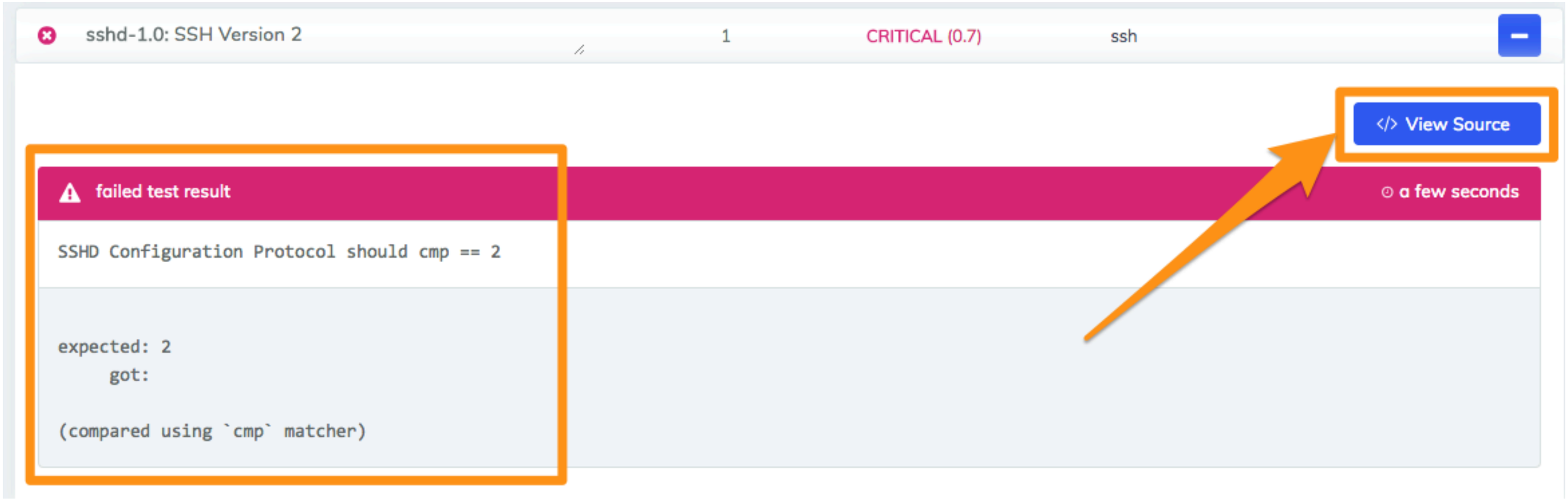
Check the compliance status in Automate



Check the compliance status in Automate



Check the compliance status in Automate



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)
```

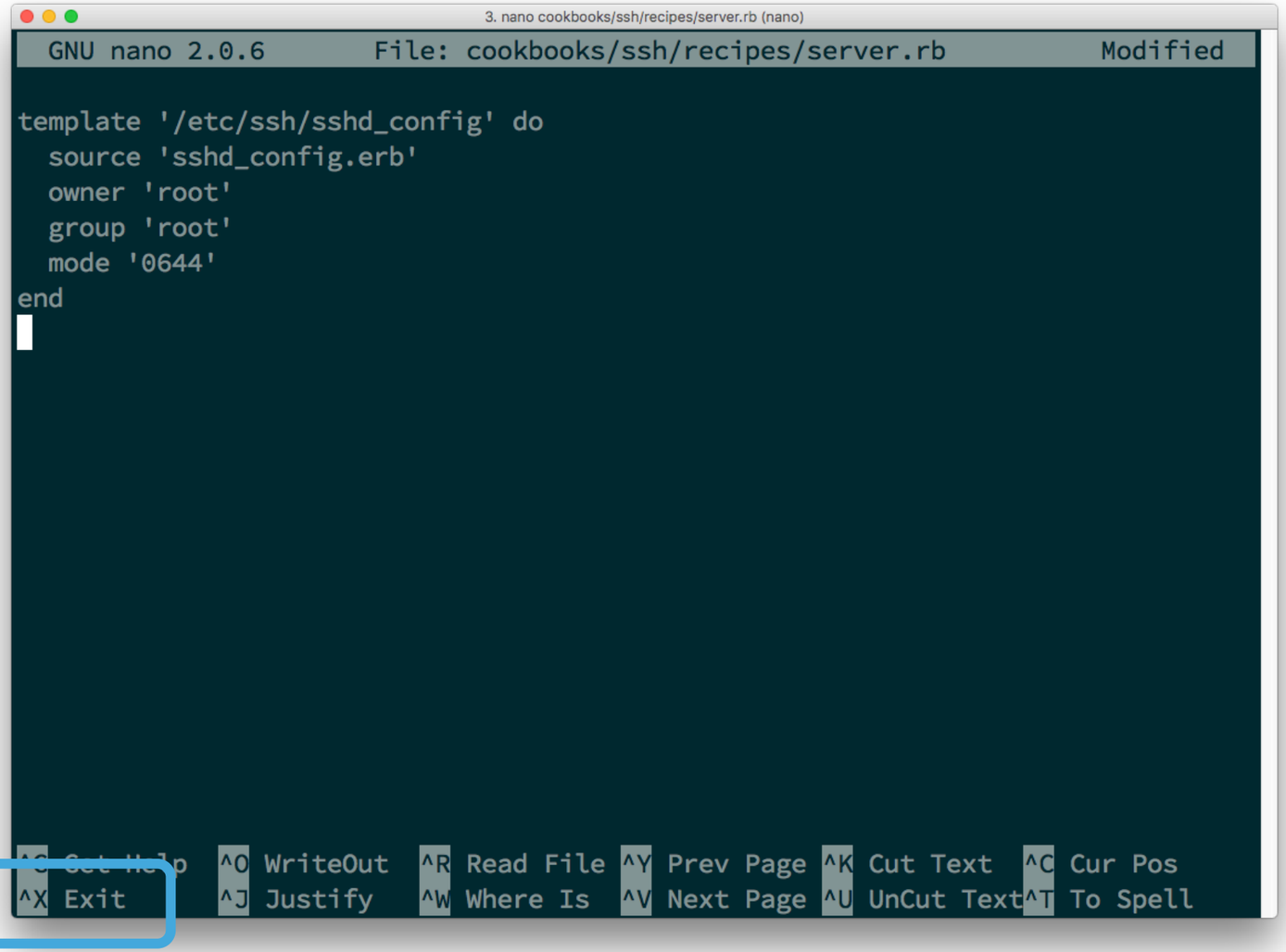
Server Recipe

~/cookbooks/ssh/recipes/server.rb

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

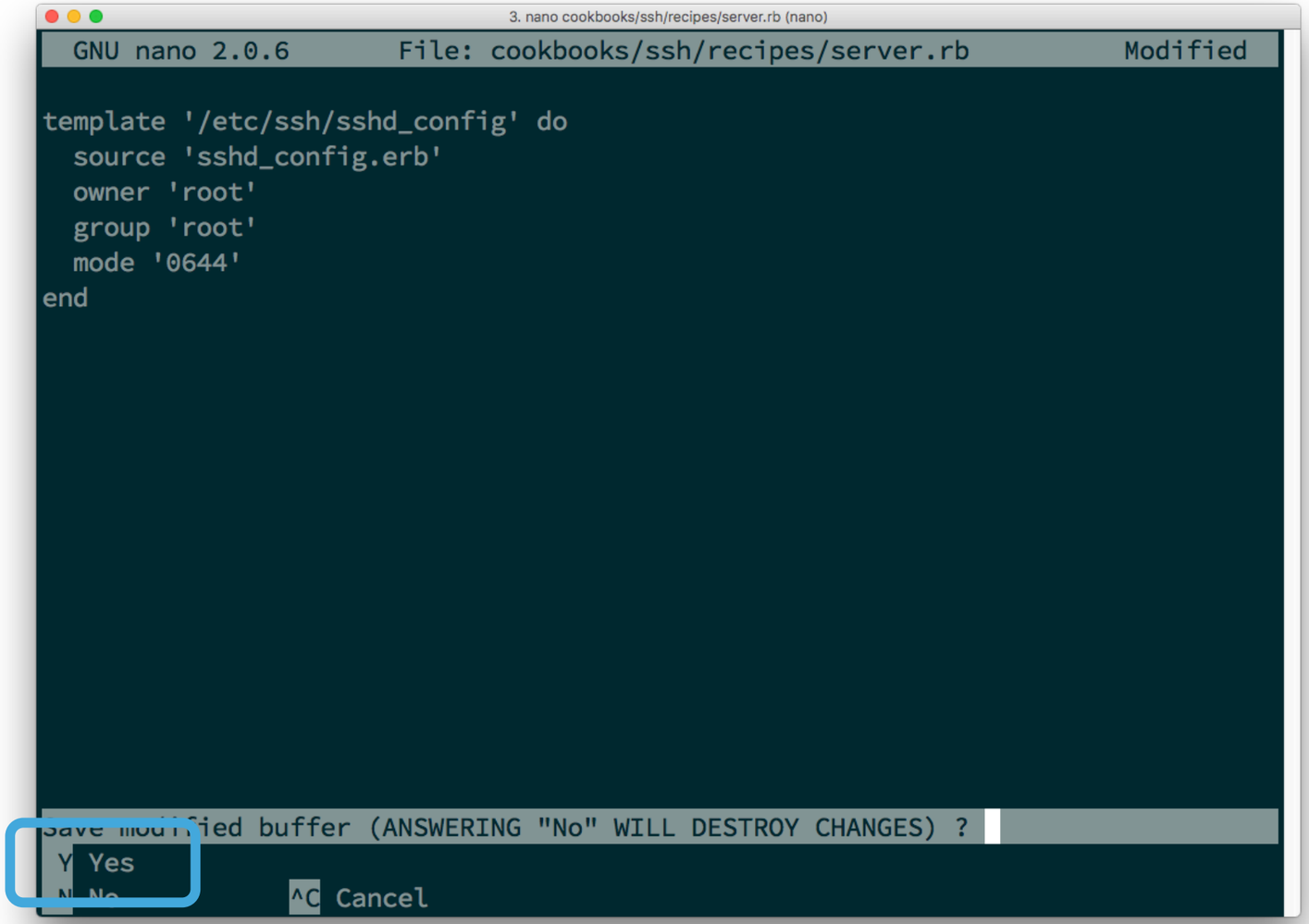
Never used a command-line text editor before? Type:
`nano cookbooks/ssh/recipes/server.rb`

Using nano



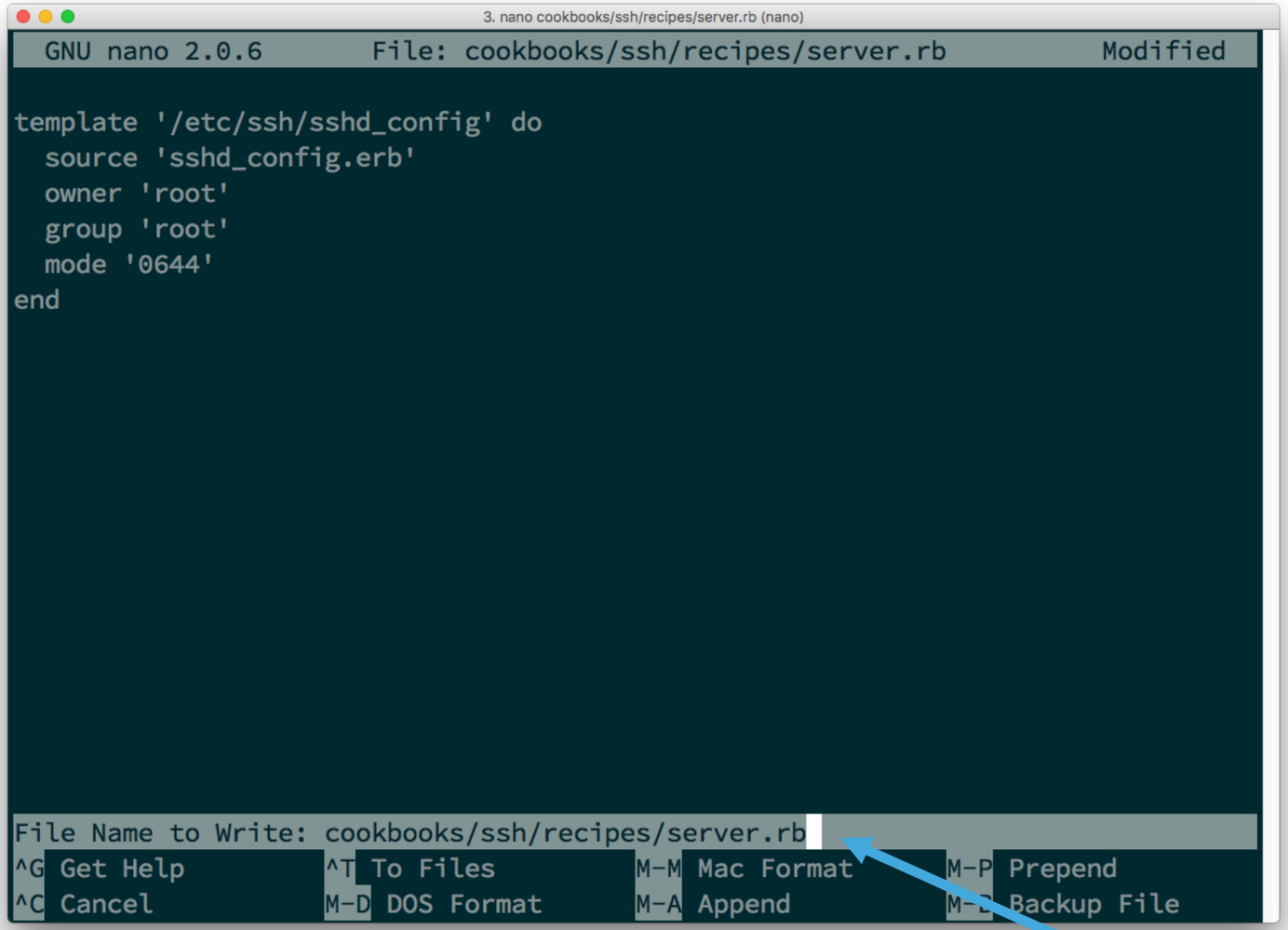
Press Control-x

Using nano



Press Y

Using nano



Press `Enter` to confirm filename

Remember...

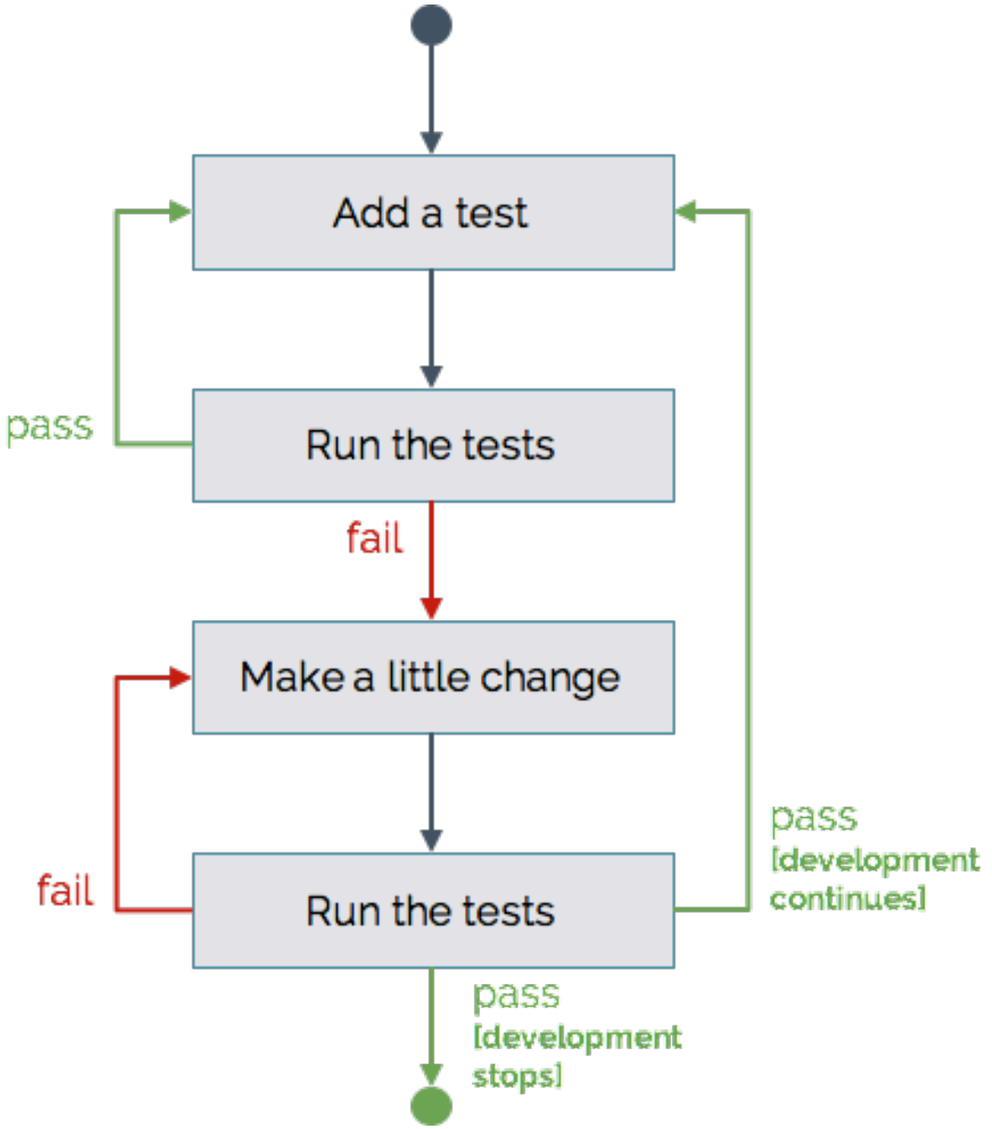
Infrastructure policies need testing!

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

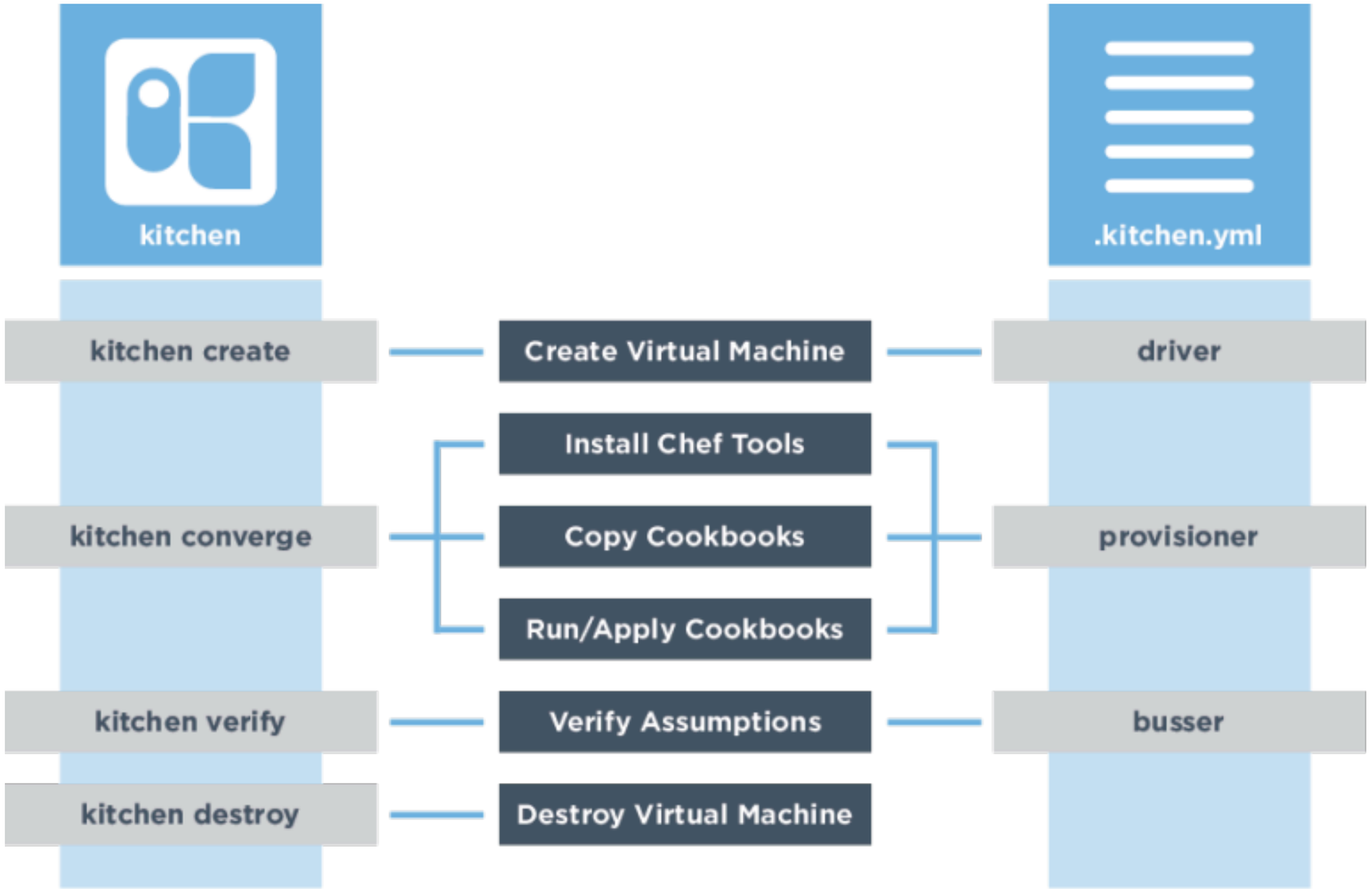
"Infrastructure as Code" should be tested like ANY other codebase.

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



amazon web services

Microsoft Azure

vmware

openstack™

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
```

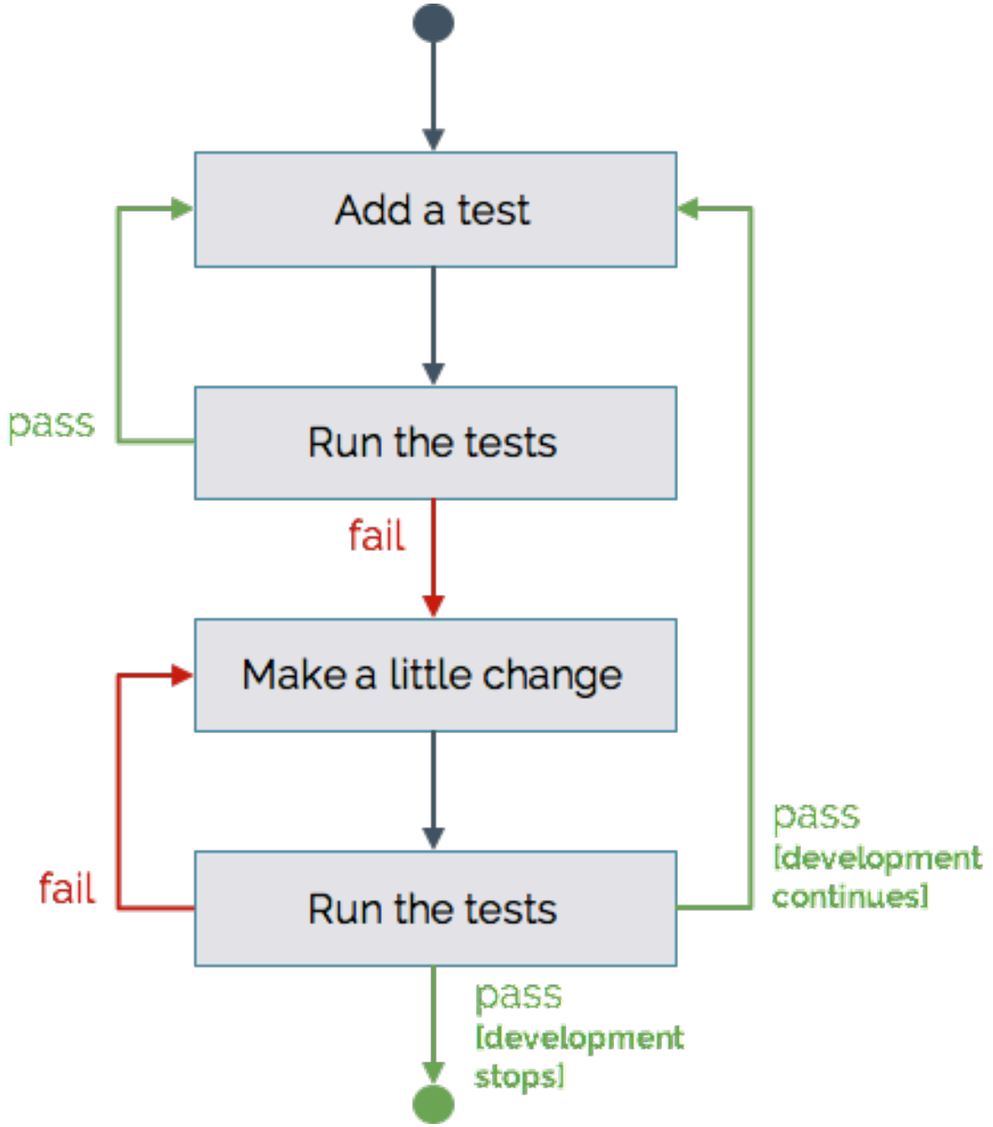
```
Instance          Driver  Provisioner  Verifier  Transport  Last Action    Last Error
server-centos-73  Docker  ChefZero     Inspec    Ssh        <Not Created>  <None>
```

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

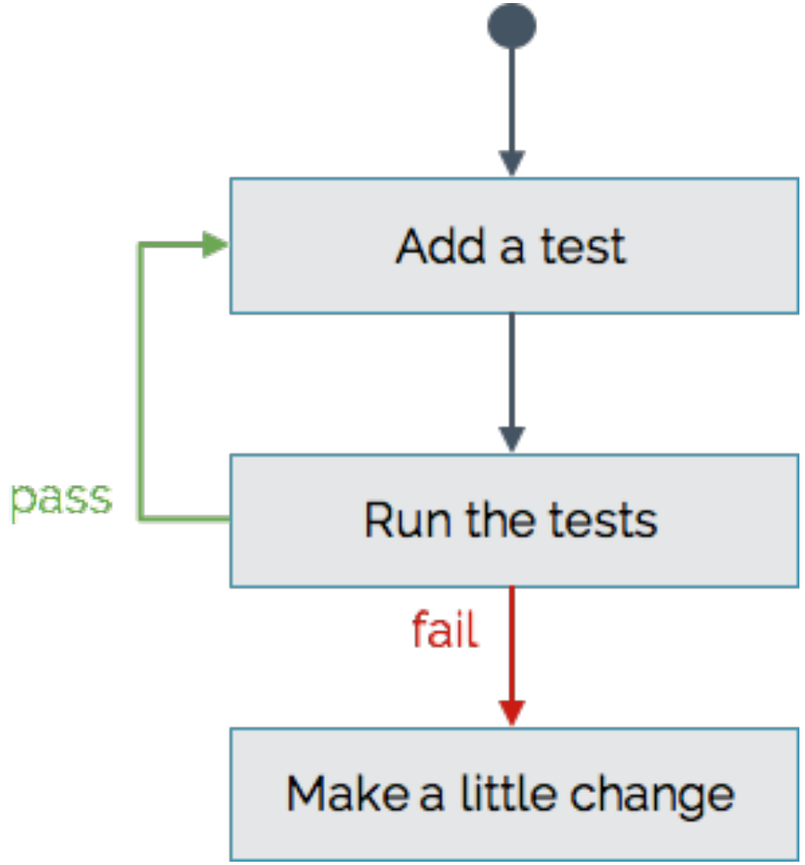


Verify the Kitchen

```
$ 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

  ×  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
```

Test-Driven Development

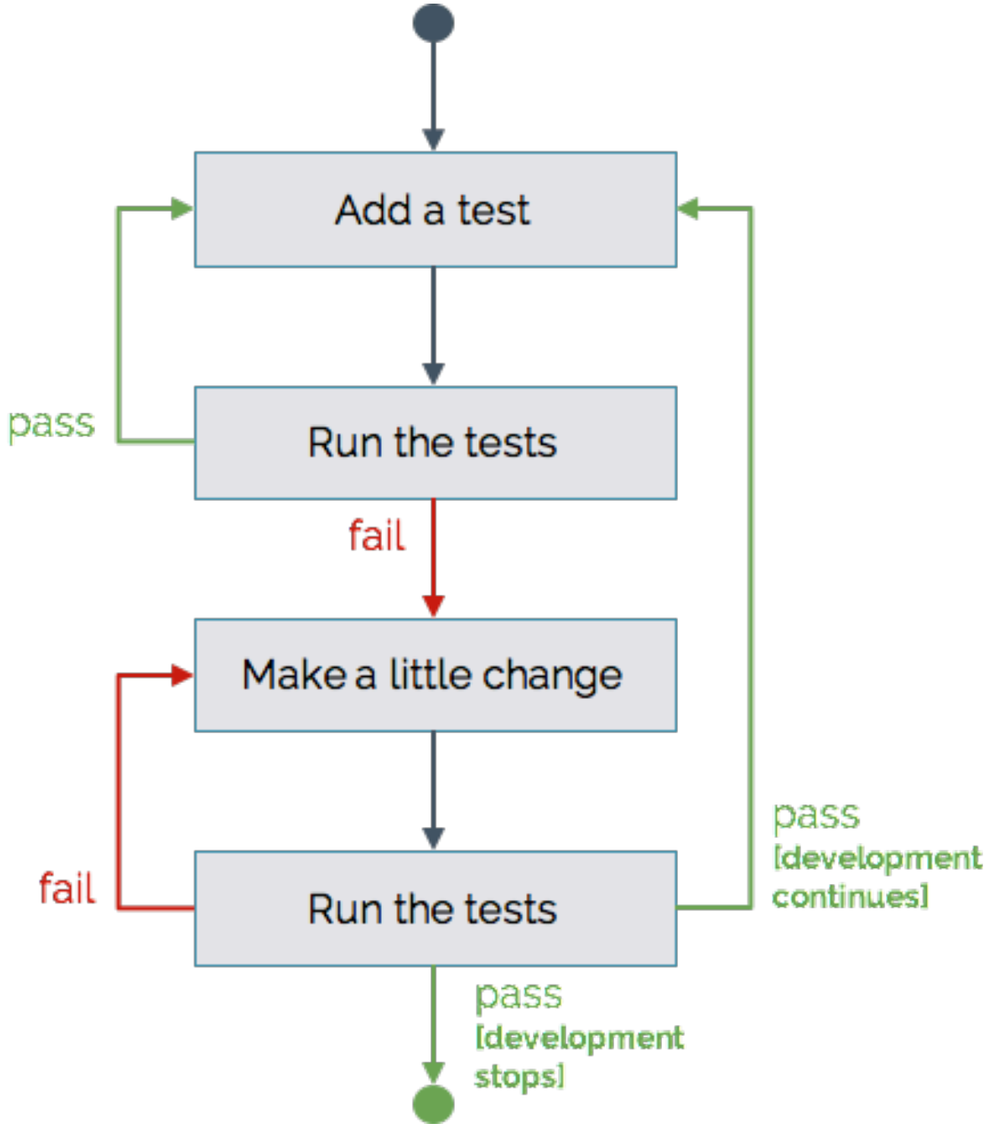


Edit the SSH Configuration Template

~/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
```

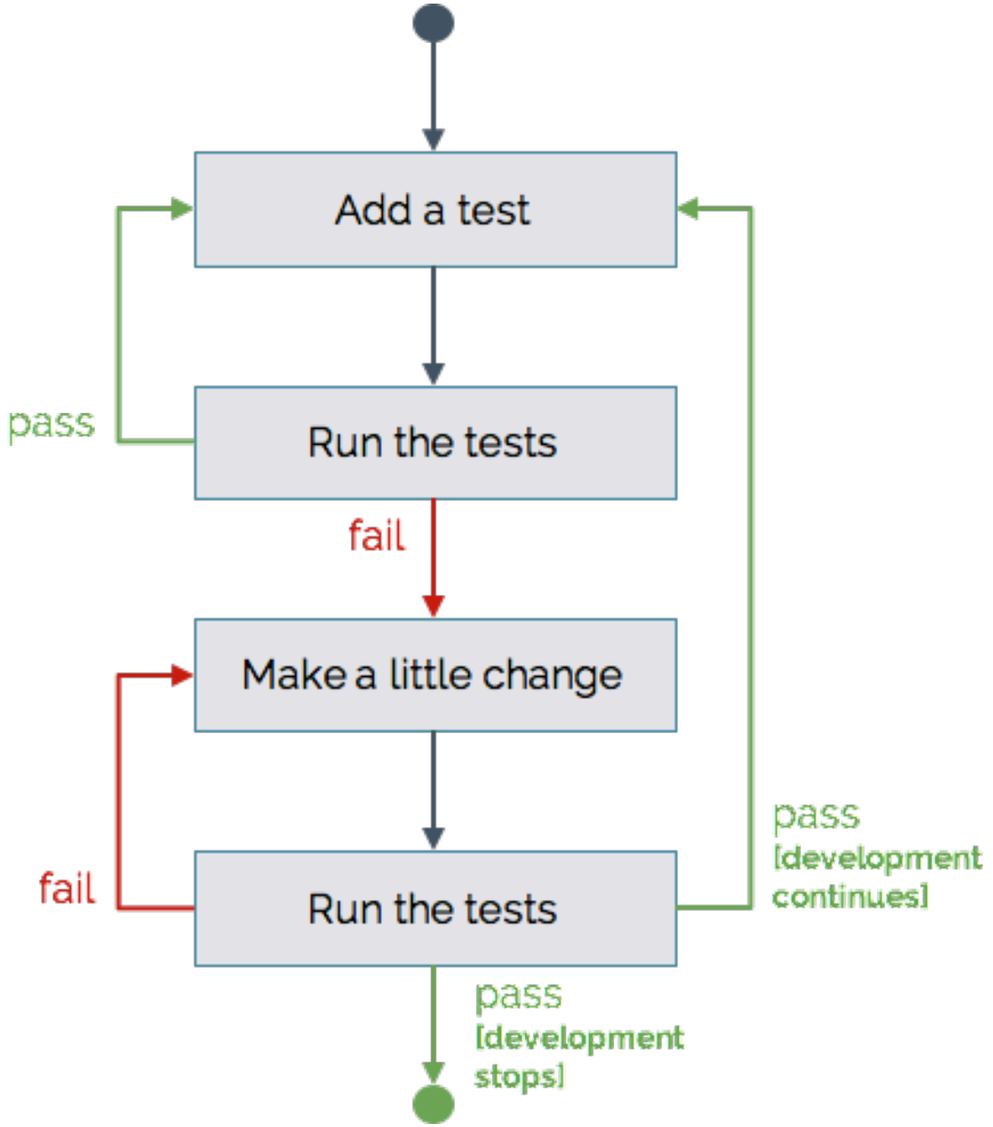
```
-----> Starting Kitchen (v1.21.2)
-----> Converging <server-centos-73>...
...
           # The default requires explicit activation of protocol 1
          -#Protocol 2
          +Protocol 2
...
       Running handlers:
       Running handlers complete
       Chef Client finished, 1/1 resources updated 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
```

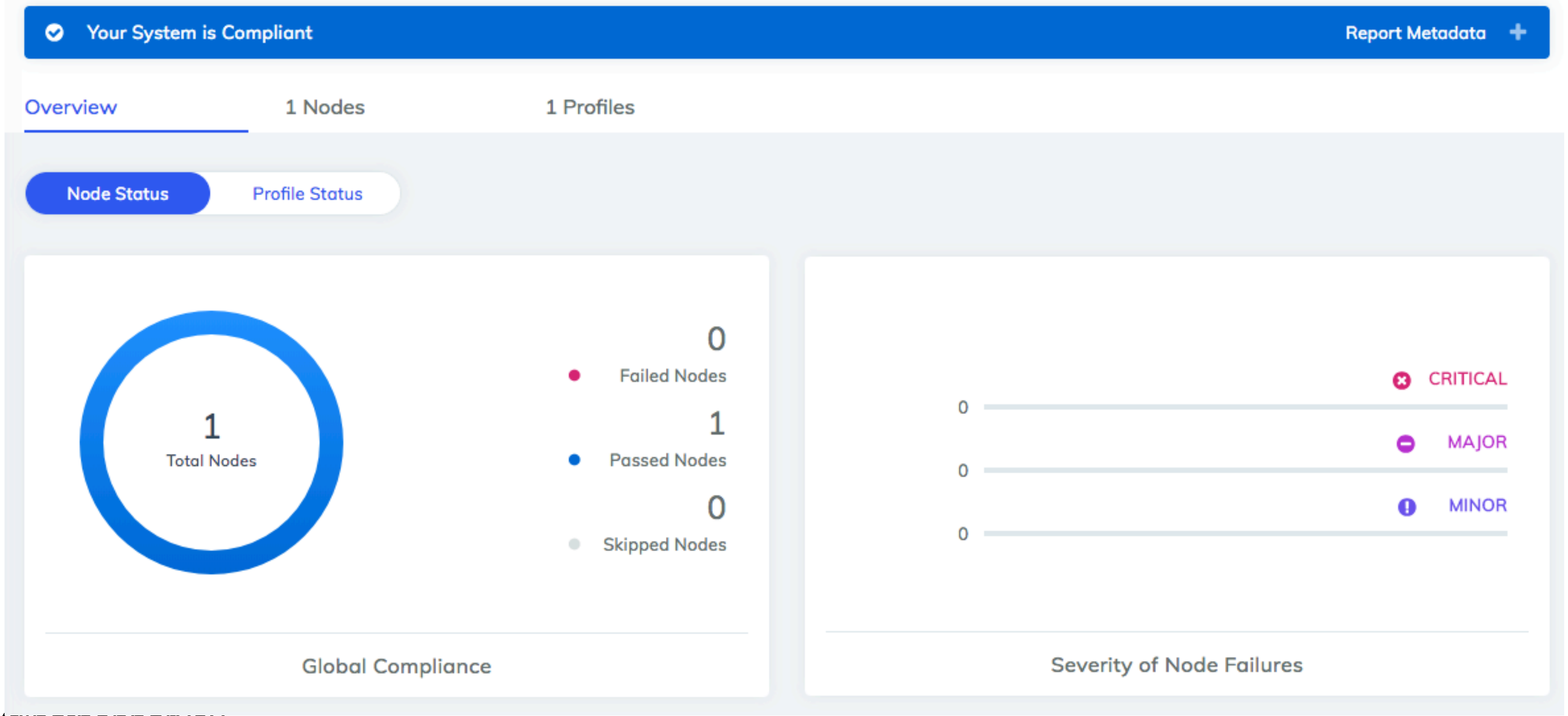
Verify Converge Status in Automate

Resources | Run List | Attributes

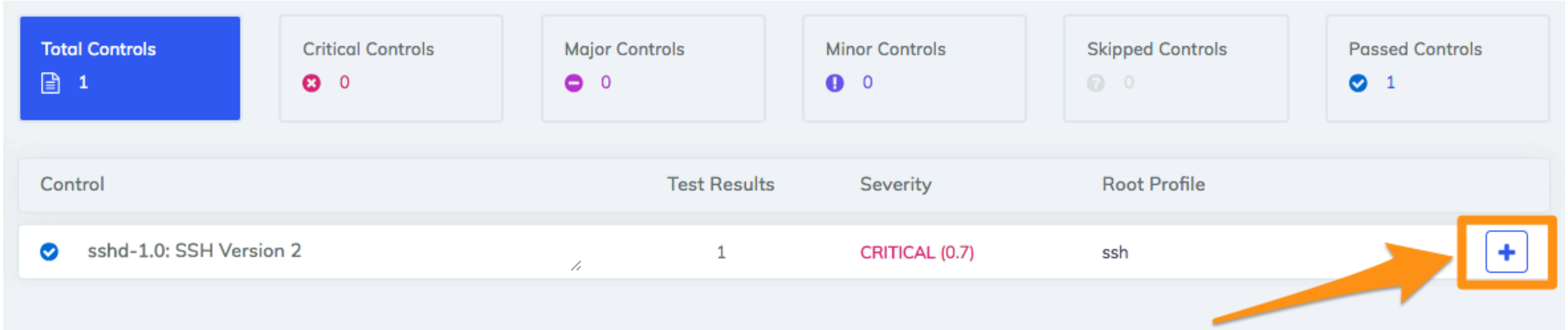
Total Resources	Failed	Successful	Unchanged	Unprocessed
3	0	1	2	0

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



Verify Compliance Status in Automate



Verify Compliance Status in Automate

Ready for more?

- **Learn Chef Rally**
  learn.chef.io
- **Classroom-style Training**

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™