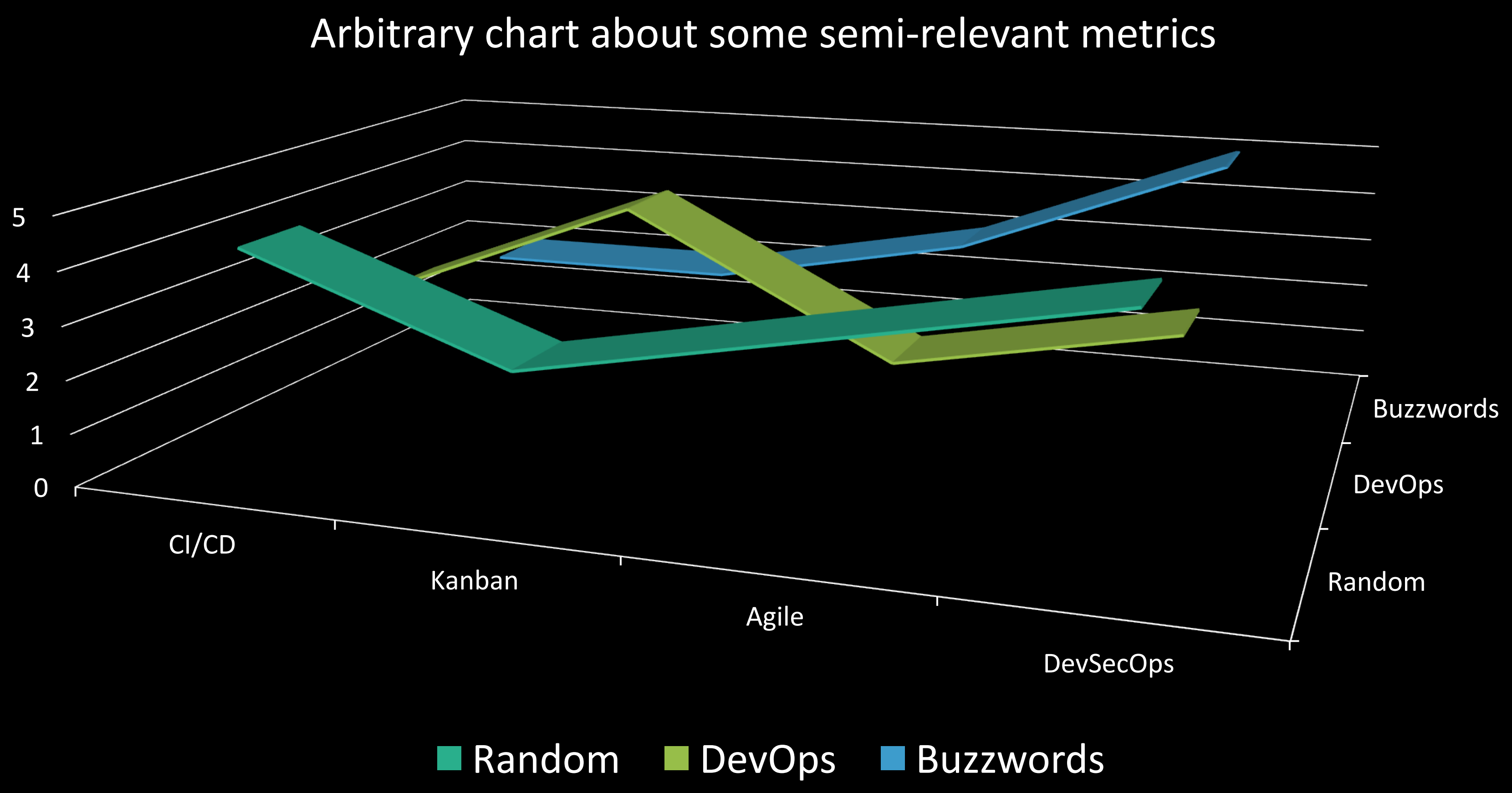
Automating Security & Compliance for Fun and Profit

Nicole Johnson

Manager, Solutions Architects @ Chef

Innovation | Security | Compliance



What is DevOps?

DevOps promotes a set of practices that emphasize collaboration and communication of both software developers and information technology professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing and releasing software can happen rapidly, frequently, and more reliably.

Translated:

Deliver High Quality, Working Software Faster

What is DevSecOps?

Through Security as Code, we have and will learn that there is simply a better way for security practitioners, like us, to operate and contribute value with less friction. We know we must adapt our ways quickly and foster innovation to ensure data security and privacy issues are not left behind because we were too slow to change.

Translated:

Deliver High Quality, Secure, Working Software Faster and More Safely

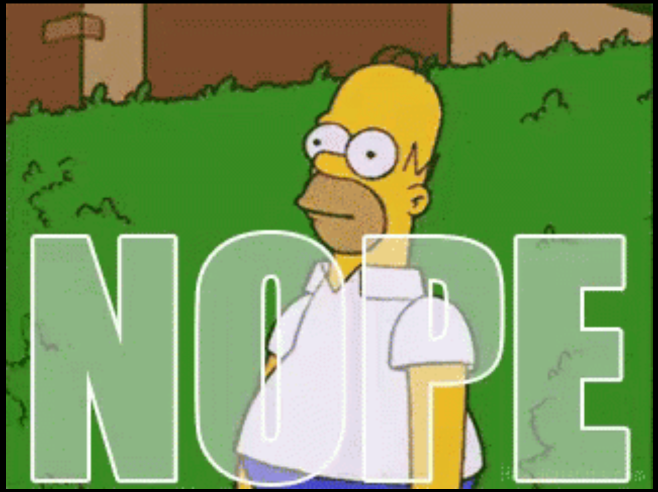
Modernize without introducing risk

- Understand impact and risk associated with being insecure
- Understand barriers to entry
- Consider security and compliance from the beginning
- Utilize effective and repeatable testing patterns

What do we mean when we say compliance?

- Compliance of POLICY
- POLICY =
  - Configuration Policy
  - Environment Policy
  - Organizational Security
  - Regulatory Compliance
  - Process and procedural Policy

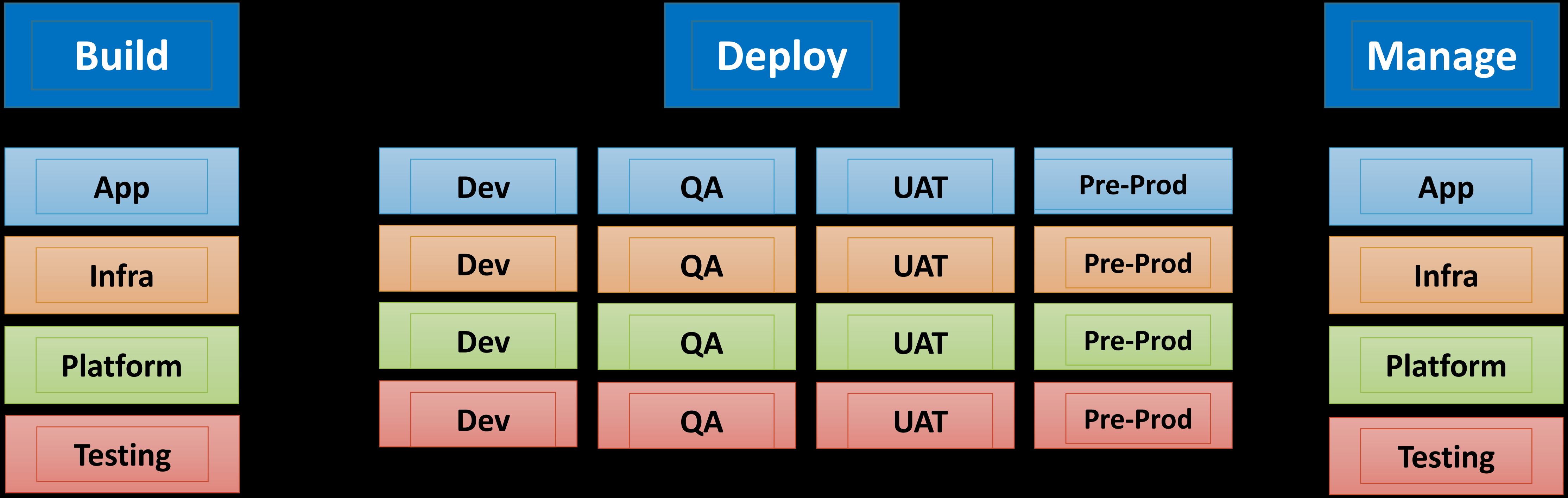
- But what about auditability?



Incorporate security and compliance from the beginning… what is the beginning?

- Manage systems and applications using infra code
- Test-driven Development
- Cross-functional standards and patterns
- Standardization of toolsets

Continuous Testing? Yes Please!



Build

App

Infra

Platform

Testing

Deploy

Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod

Manage

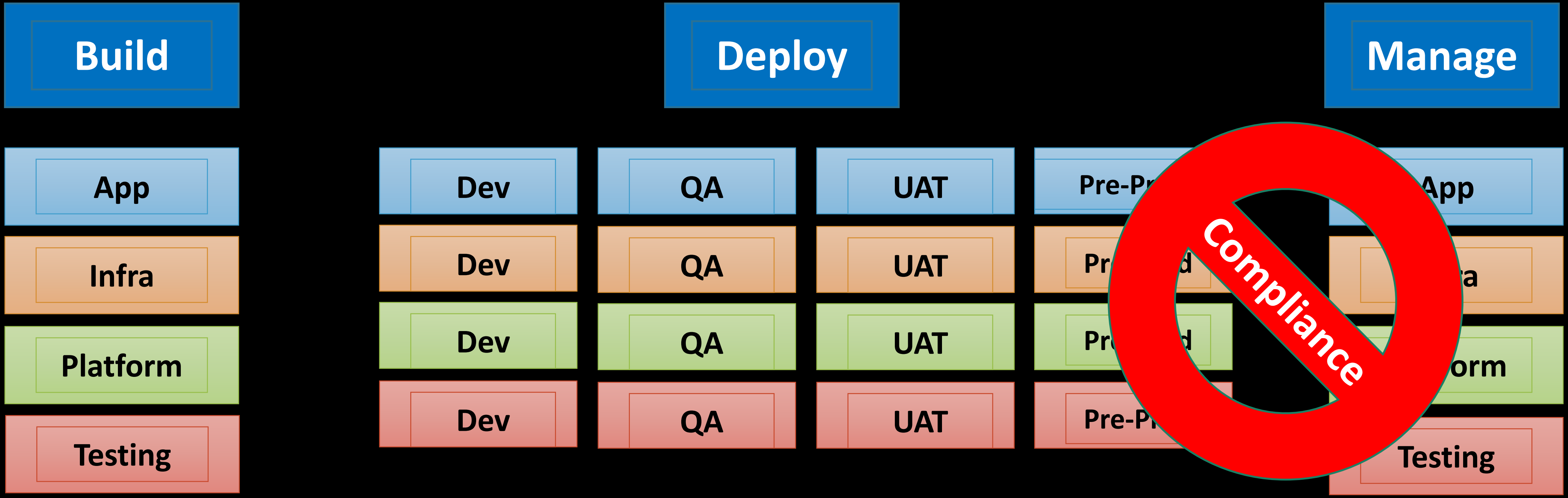
App

Infra

Platform

Testing

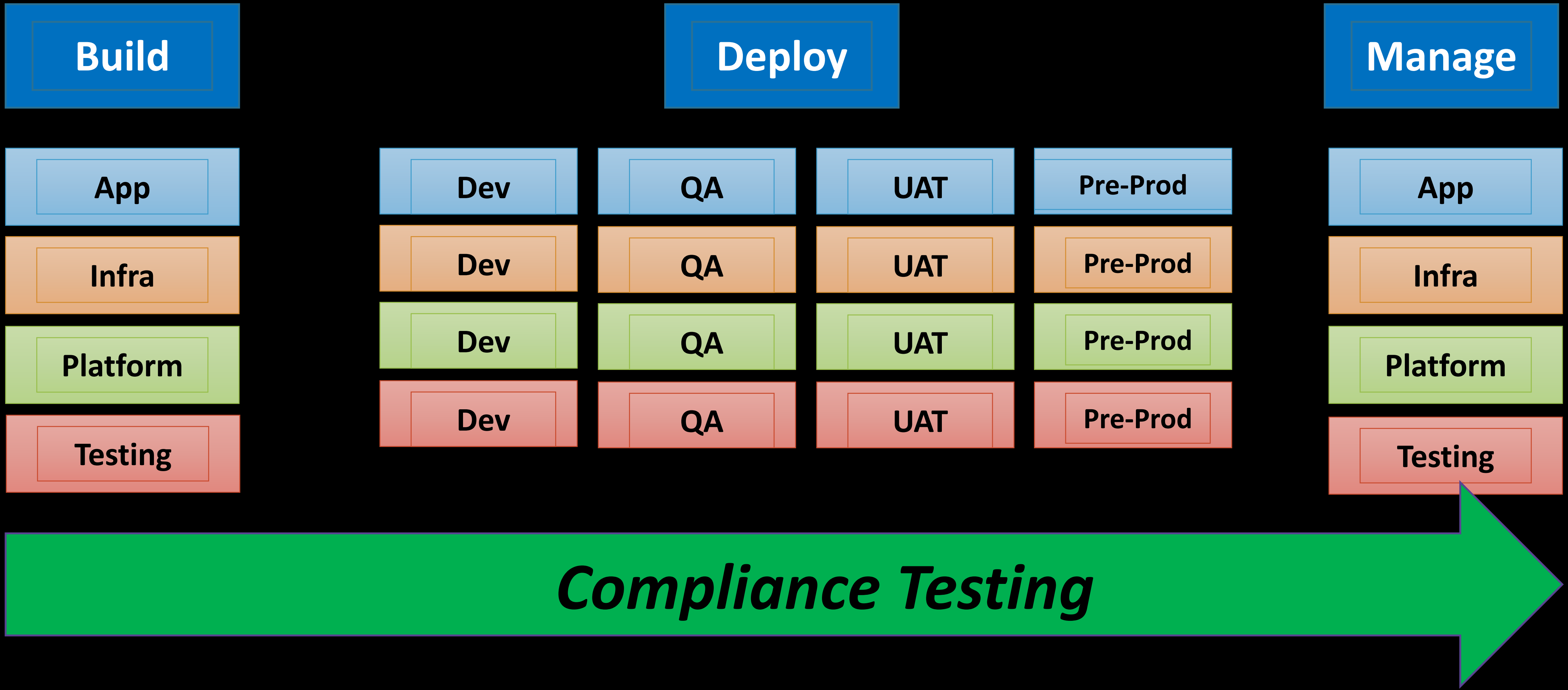
Oops – Someone forgot to call compliance



Build	Deploy				Manage
App	Dev	QA	UAT	Pre-Pr	App
Infra	Dev	QA	UAT	Pr d	a
Platform	Dev	QA	UAT	Pr d	orm
Testing	Dev	QA	UAT	Pre-P	Testing

So what do we do about it??

Compliance Testing Through Each Step



Build	Deploy				Manage
App	Dev	QA	UAT	Pre-Prod	App
Infra	Dev	QA	UAT	Pre-Prod	Infra
Platform	Dev	QA	UAT	Pre-Prod	Platform
Testing	Dev	QA	UAT	Pre-Prod	Testing

Compliance Testing

Infrastructure as Code + Compliance as Code = ❤

- Test-Driven Development + Compliance =
- Continuous Delivery + Compliance =
- Development + Operations + Compliance =

Hug Ops!

INSPEC

Inspec is compliance as code - a human-readable language for automating the continuous testing and compliance auditing of your entire infrastructure

github.com/chef/inspec

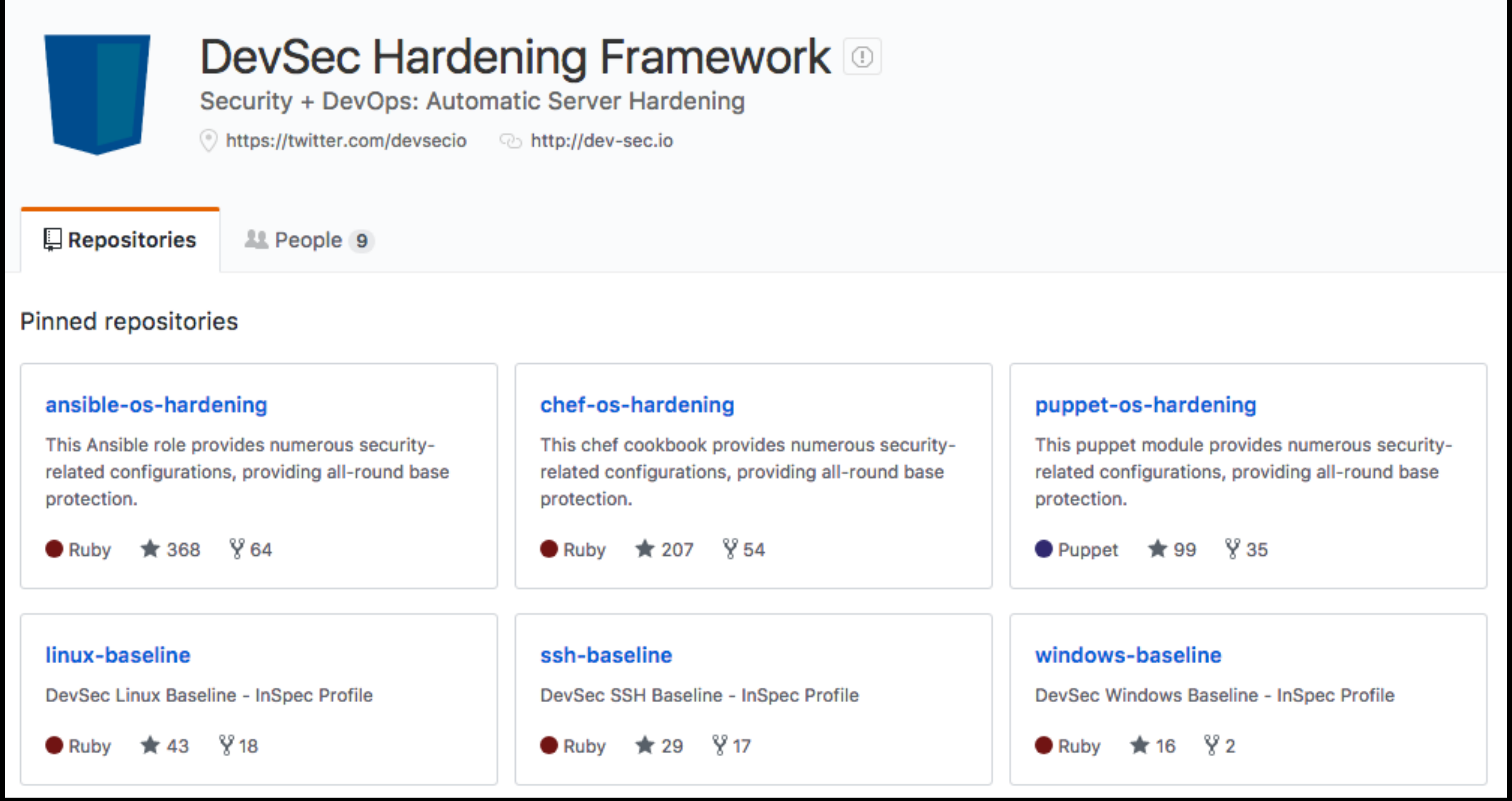
Why do you care?

- Infrastructure as code -> Compliance as code
- Declarative
- Auditability

More details

- Open-source run-time framework and rule language to specify compliance, security and policy for testing any node in your environment.
- Includes a collection of resources to write audit rules quickly and easily
- Examine nodes
- Run tests locally or remotely
- Detected security, compliance and policy issues are flagged in logs

Baseline Security Requirements



DevSec Hardening Framework

Security + DevOps: Automatic Server Hardening

https://twitter.com/devsecio http://dev-sec.io

Pinned repositories

ansible-os-hardening
This Ansible role provides numerous security-related configurations, providing all-round base protection.
● Ruby ★ 368 64

chef-os-hardening
This chef cookbook provides numerous security-related configurations, providing all-round base protection.
● Ruby ★ 207 54

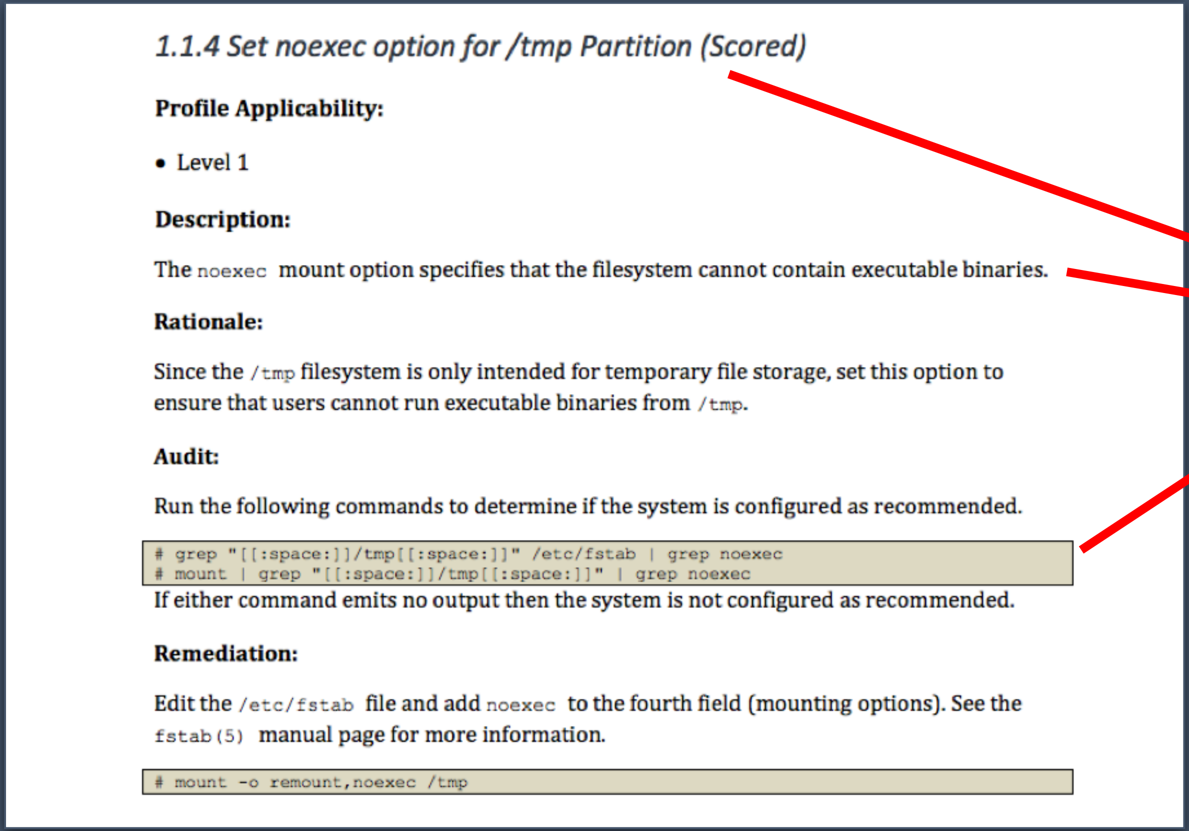
puppet-os-hardening
This puppet module provides numerous security-related configurations, providing all-round base protection.
● Puppet ★ 99 35

linux-baseline
DevSec Linux Baseline - InSpec Profile
● Ruby ★ 43 18

ssh-baseline
DevSec SSH Baseline - InSpec Profile
● Ruby ★ 29 17

windows-baseline
DevSec Windows Baseline - InSpec Profile
● Ruby ★ 16 2

- Linux and Windows Baseline
- SSH Baseline Config
- SSL Baseline
- Docker Security
- Etc.



1.1.4 Set noexec option for /tmp Partition (Scored)

Profile Applicability:

- Level 1

Description:

The `noexec` mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Since the `/tmp` filesystem is only intended for temporary file storage, set this option to ensure that users cannot run executable binaries from `/tmp`.

Audit:

Run the following commands to determine if the system is configured as recommended.

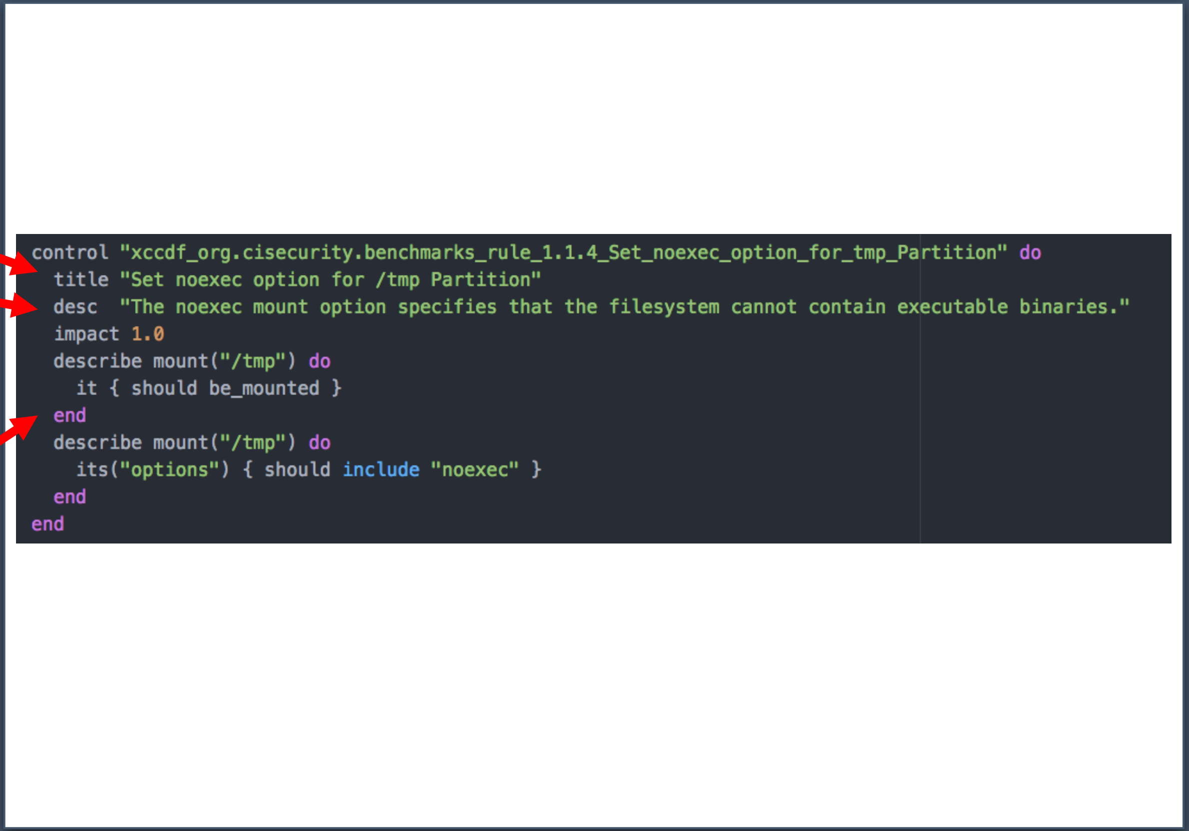
```
# grep "[[:space:]]/tmp[[:space:]]" /etc/fstab | grep noexec
# mount | grep "[[:space:]]/tmp[[:space:]]" | grep noexec
```

If either command emits no output then the system is not configured as recommended.

Remediation:

Edit the `/etc/fstab` file and add `noexec` to the fourth field (mounting options). See the `fstab(5)` manual page for more information.

```
# mount -o remount,noexec /tmp
```

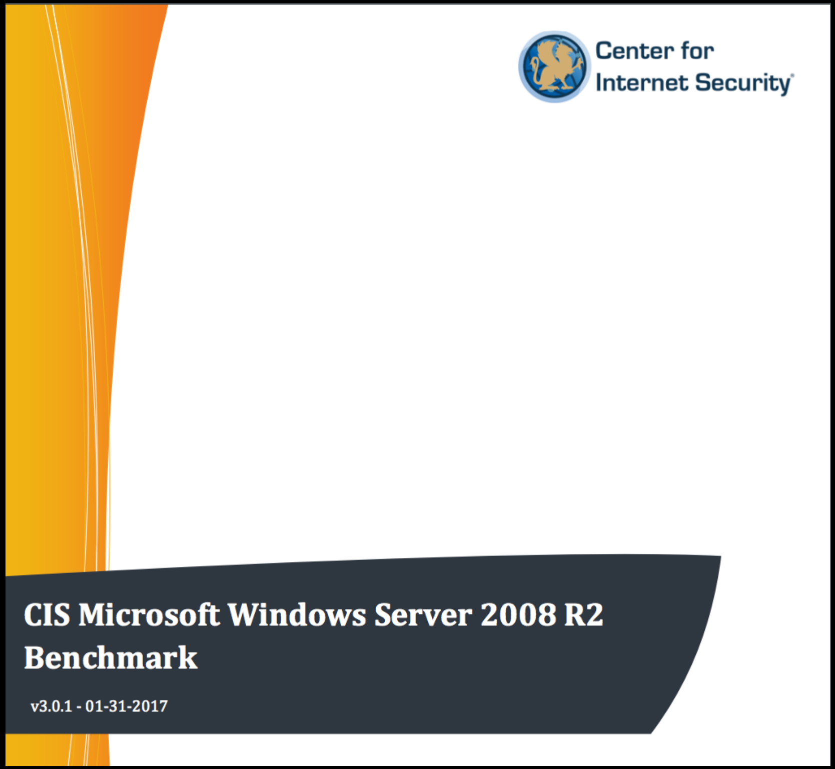


```
control "xccdf_org.cisecurity.benchmarks_rule_1.1.4_Set_noexec_option_for_tmp_Partition" do
  title "Set noexec option for /tmp Partition"
  desc  "The noexec mount option specifies that the filesystem cannot contain executable binaries."
  impact 1.0
  describe mount("/tmp") do
    it { should be_mounted }
  end
  describe mount("/tmp") do
    its("options") { should include "noexec" }
  end
end
```

Compliance Benchmarks

CENTER FOR INTERNET SECURITY

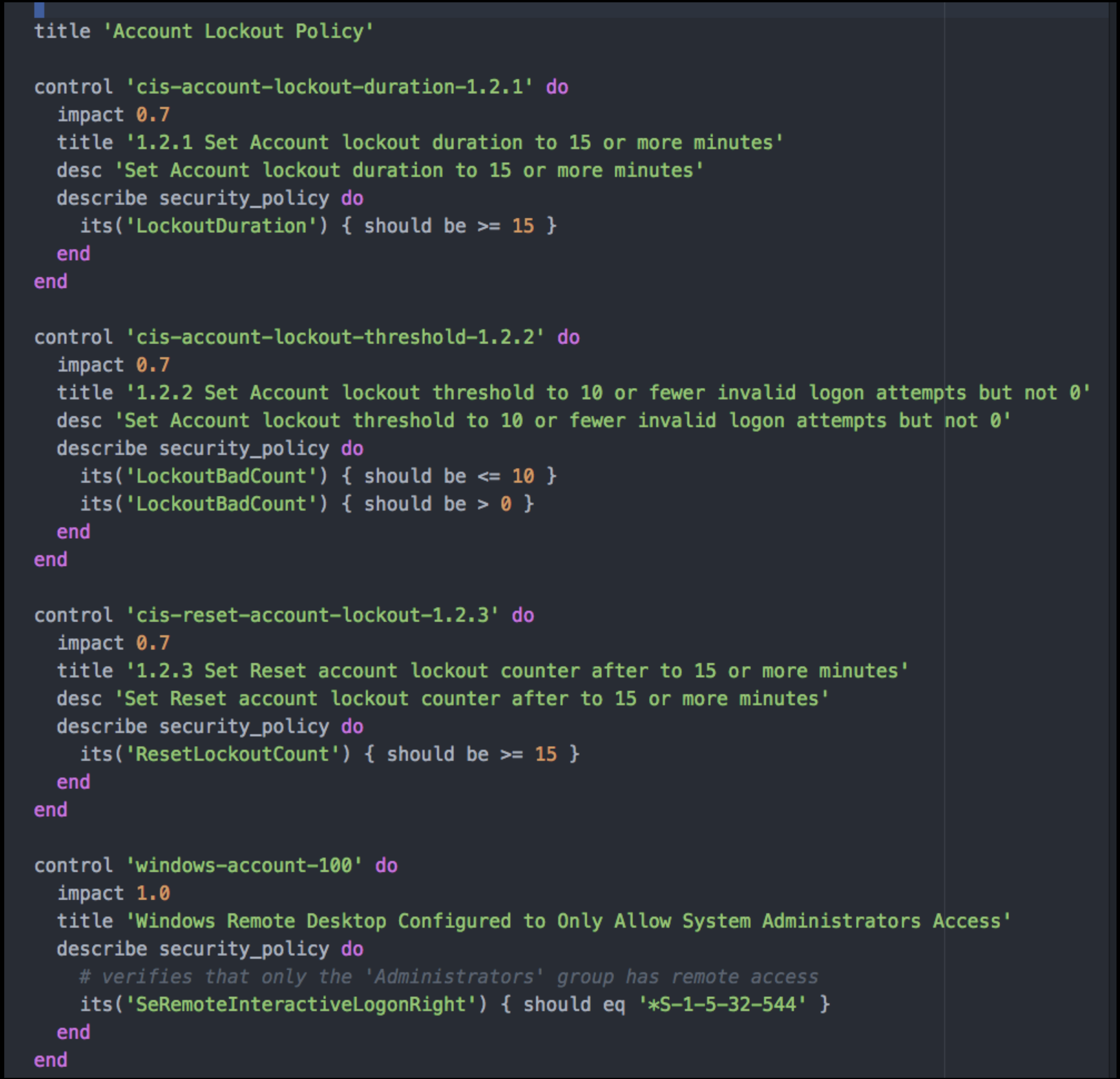
Windows



Center for Internet Security

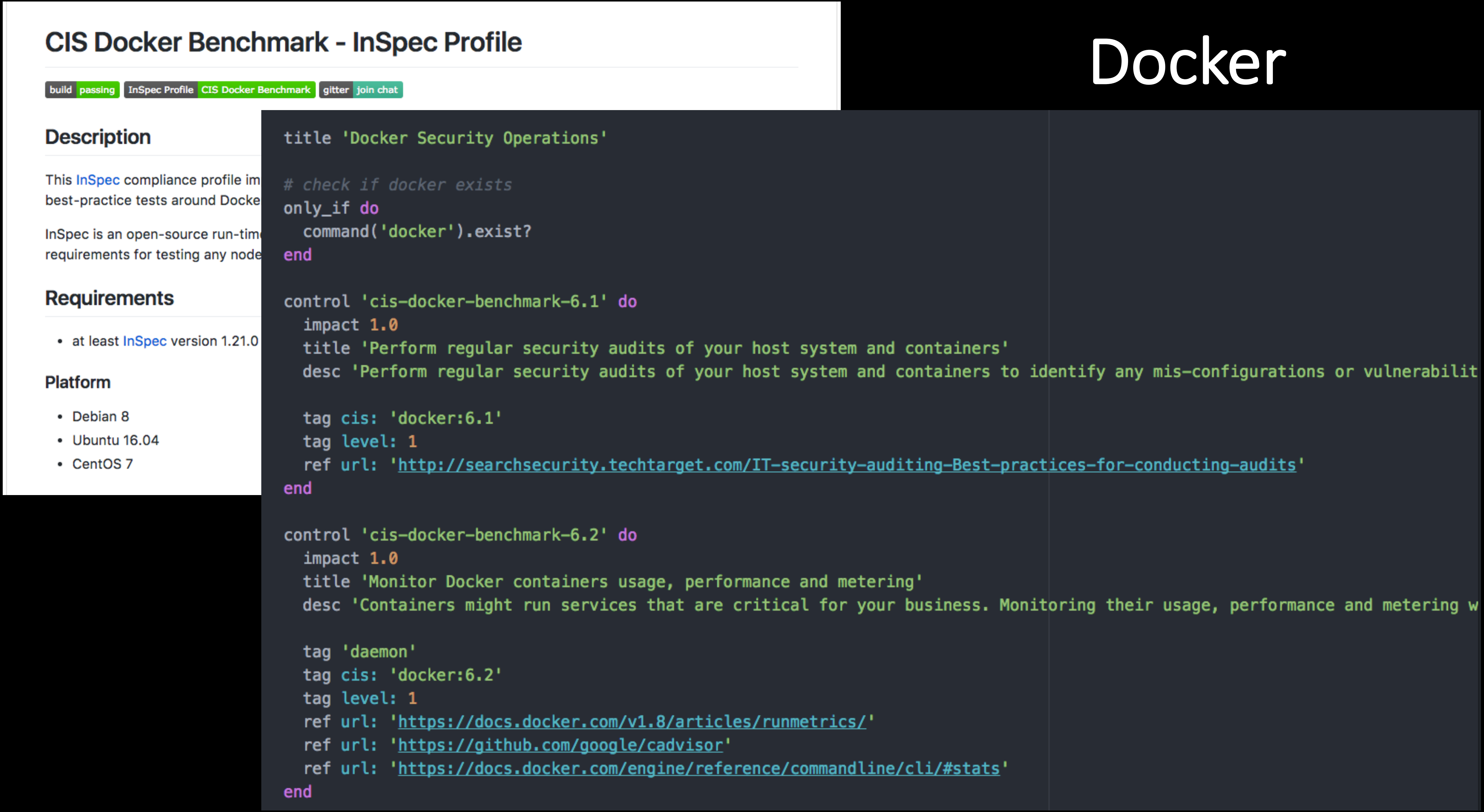
CIS Microsoft Windows Server 2008 R2 Benchmark

v3.0.1 - 01-31-2017



```
title 'Account Lockout Policy'

control 'cis-account-lockout-duration-1.2.1' do
  impact 0.7
  title '1.2.1 Set Account lockout duration to 15 or more minutes'
  desc 'Set Account lockout duration to 15 or more minutes'
  describe security_policy do
    its('LockoutDuration') { should be >= 15 }
  end
end

control 'cis-account-lockout-threshold-1.2.2' do
  impact 0.7
  title '1.2.2 Set Account lockout threshold to 10 or fewer invalid logon attempts but not 0'
  desc 'Set Account lockout threshold to 10 or fewer invalid logon attempts but not 0'
  describe security_policy do
    its('LockoutBadCount') { should be <= 10 }
    its('LockoutBadCount') { should be > 0 }
  end
end

control 'cis-reset-account-lockout-1.2.3' do
  impact 0.7
  title '1.2.3 Set Reset account lockout counter after to 15 or more minutes'
  desc 'Set Reset account lockout counter after to 15 or more minutes'
  describe security_policy do
    its('ResetLockoutCount') { should be >= 15 }
  end
end

control 'windows-account-100' do
  impact 1.0
  title 'Windows Remote Desktop Configured to Only Allow System Administrators Access'
  describe security_policy do
    # verifies that only the 'Administrators' group has remote access
    its('SeRemoteInteractiveLogonRight') { should eq '*S-1-5-32-544' }
  end
end
```

Docker



CIS Docker Benchmark - InSpec Profile

Description

This InSpec compliance profile im
best-practice tests around Docke

InSpec is an open-source run-tim
requirements for testing any node

Requirements

- at least InSpec version 1.21.0

Platform

- Debian 8
- Ubuntu 16.04
- CentOS 7

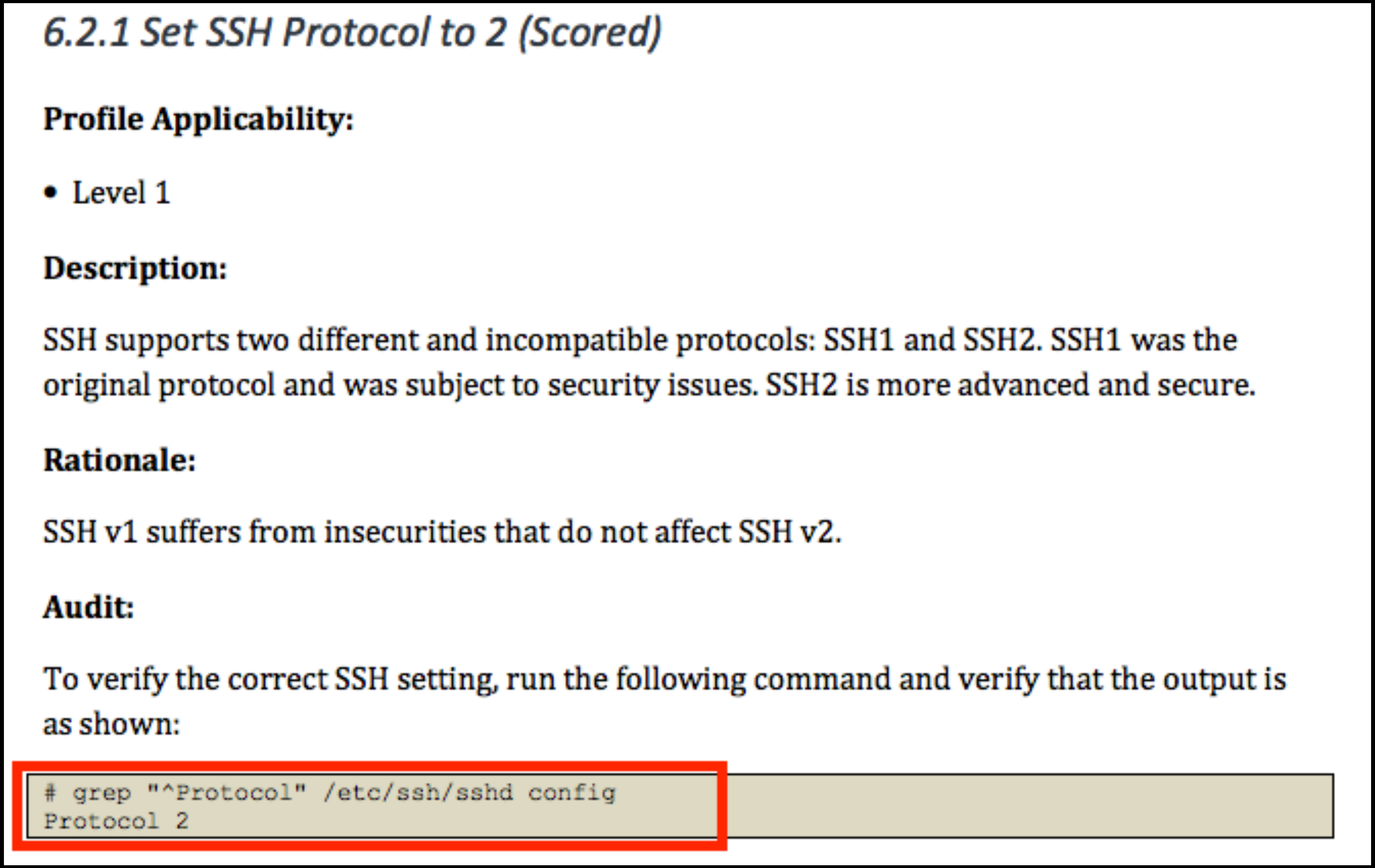
```ruby
title 'Docker Security Operations'

# check if docker exists
only_if do
  command('docker').exist?
end

control 'cis-docker-benchmark-6.1' do
  impact 1.0
  title 'Perform regular security audits of your host system and containers'
  desc 'Perform regular security audits of your host system and containers to identify any mis-configurations or vulnerabilit

  tag cis: 'docker:6.1'
  tag level: 1
  ref url: 'http://searchsecurity.techtarget.com/IT-security-auditing-Best-practices-for-conducting-audits'
end

control 'cis-docker-benchmark-6.2' do
  impact 1.0
  title 'Monitor Docker containers usage, performance and metering'
  desc 'Containers might run services that are critical for your business. Monitoring their usage, performance and metering w

  tag 'daemon'
  tag cis: 'docker:6.2'
  tag level: 1
  ref url: 'https://docs.docker.com/v1.8/articles/runmetrics/'
  ref url: 'https://github.com/google/cadvisor'
  ref url: 'https://docs.docker.com/engine/reference/commandline/cli/#stats'
end
```

Test Any Target

```
$ inspec exec test.rb

$ inspec exec test.rb -i ~/.aws/nathen.pem -t ssh://ec2-user@54.152.7.203

$ inspec exec test.rb -t winrm://Admin@192.168.1.2 --password super

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

Automated Security and Compliance Tests

- How do we make this into code??
- How do we make this auditable??



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

Description:

SSH supports two different and incompatible protocols: SSH1 and SSH2. SSH1 was the original protocol and was subject to security issues. SSH2 is more advanced and secure.

Rationale:

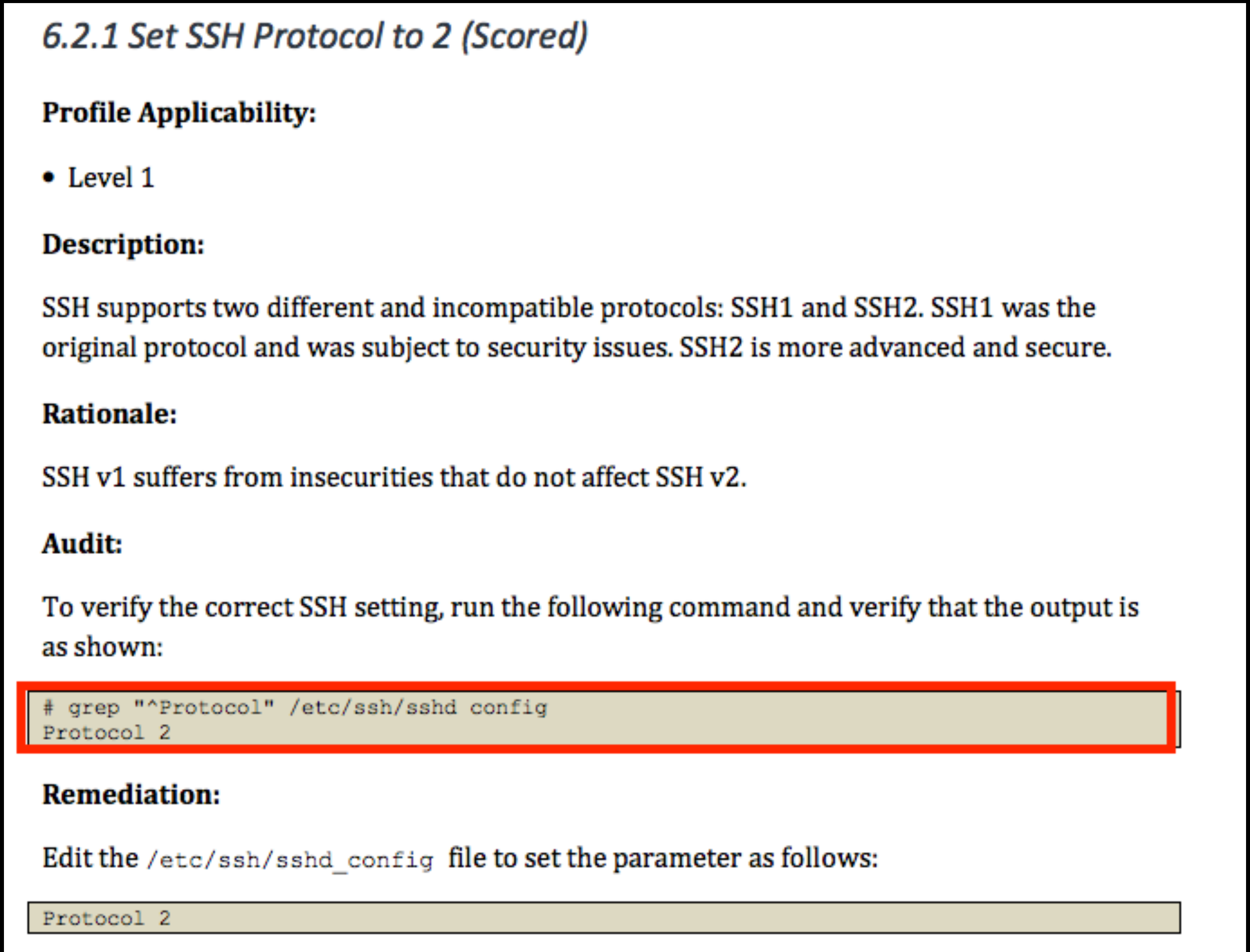
SSH v1 suffers from insecurities that do not affect SSH v2.

Audit:

To verify the correct SSH setting, run the following command and verify that the output is as shown:

```
# grep "^Protocol" /etc/ssh/sshd config
Protocol 2
```

Let’s try it!



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

Description:

SSH supports two different and incompatible protocols: SSH1 and SSH2. SSH1 was the original protocol and was subject to security issues. SSH2 is more advanced and secure.

Rationale:

SSH v1 suffers from insecurities that do not affect SSH v2.

Audit:

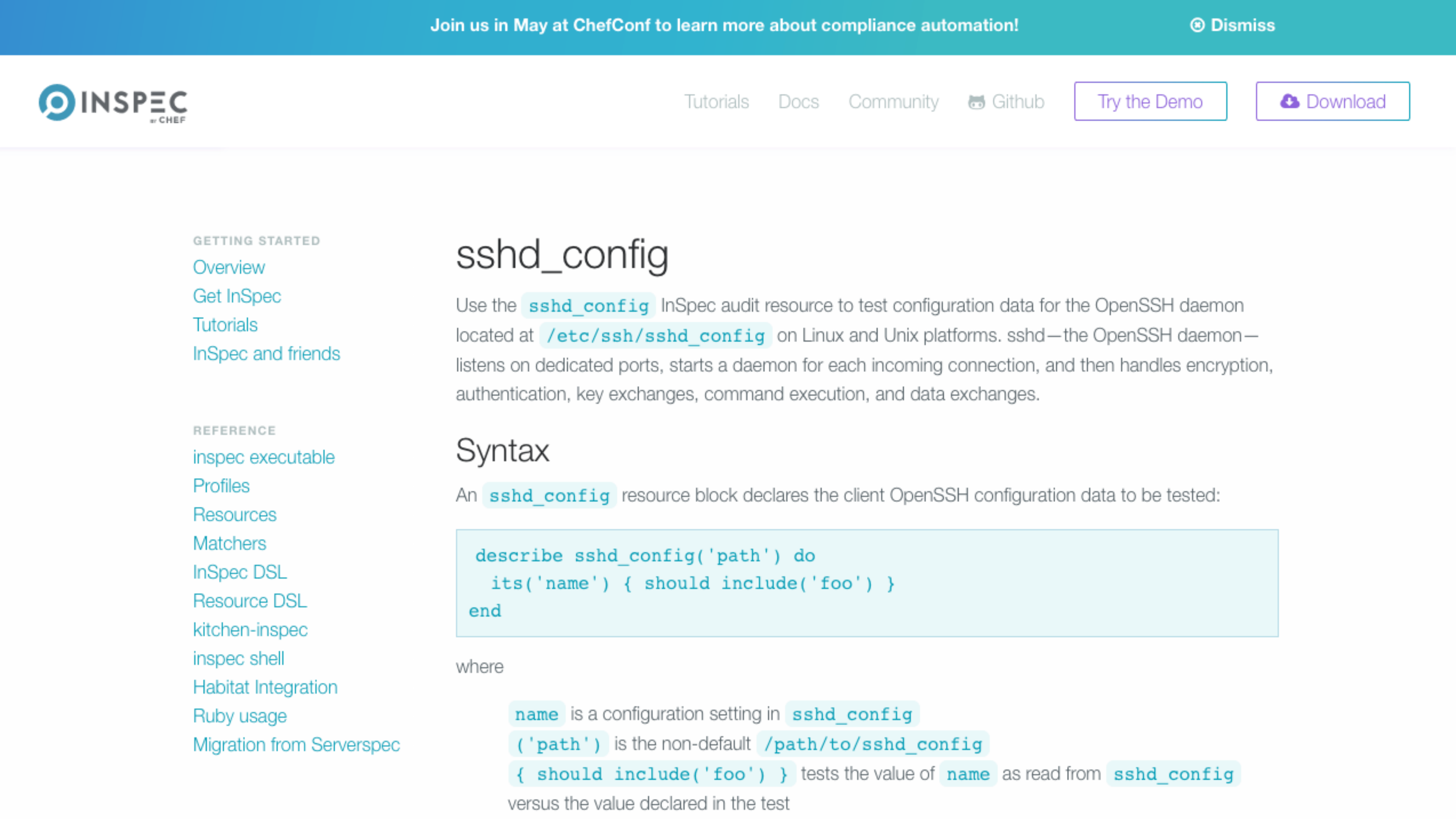
To verify the correct SSH setting, run the following command and verify that the output is as shown:

```
# grep "^Protocol" /etc/ssh/sshd config
Protocol 2
```

Remediation:

Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
Protocol 2
```



GETTING STARTED
Overview
Get InSpec
Tutorials
InSpec and friends

REFERENCE
inspec executable
Profiles
Resources
Matchers
InSpec DSL
Resource DSL
kitchen-inspec
inspec shell
Habitat Integration
Ruby usage
Migration from Serverspec

sshd_config

Use the `sshd_config` InSpec audit resource to test configuration data for the OpenSSH daemon located at `/etc/ssh/sshd_config` on Linux and Unix platforms. sshd—the OpenSSH daemon—listens on dedicated ports, starts a daemon for each incoming connection, and then handles encryption, authentication, key exchanges, command execution, and data exchanges.

Syntax

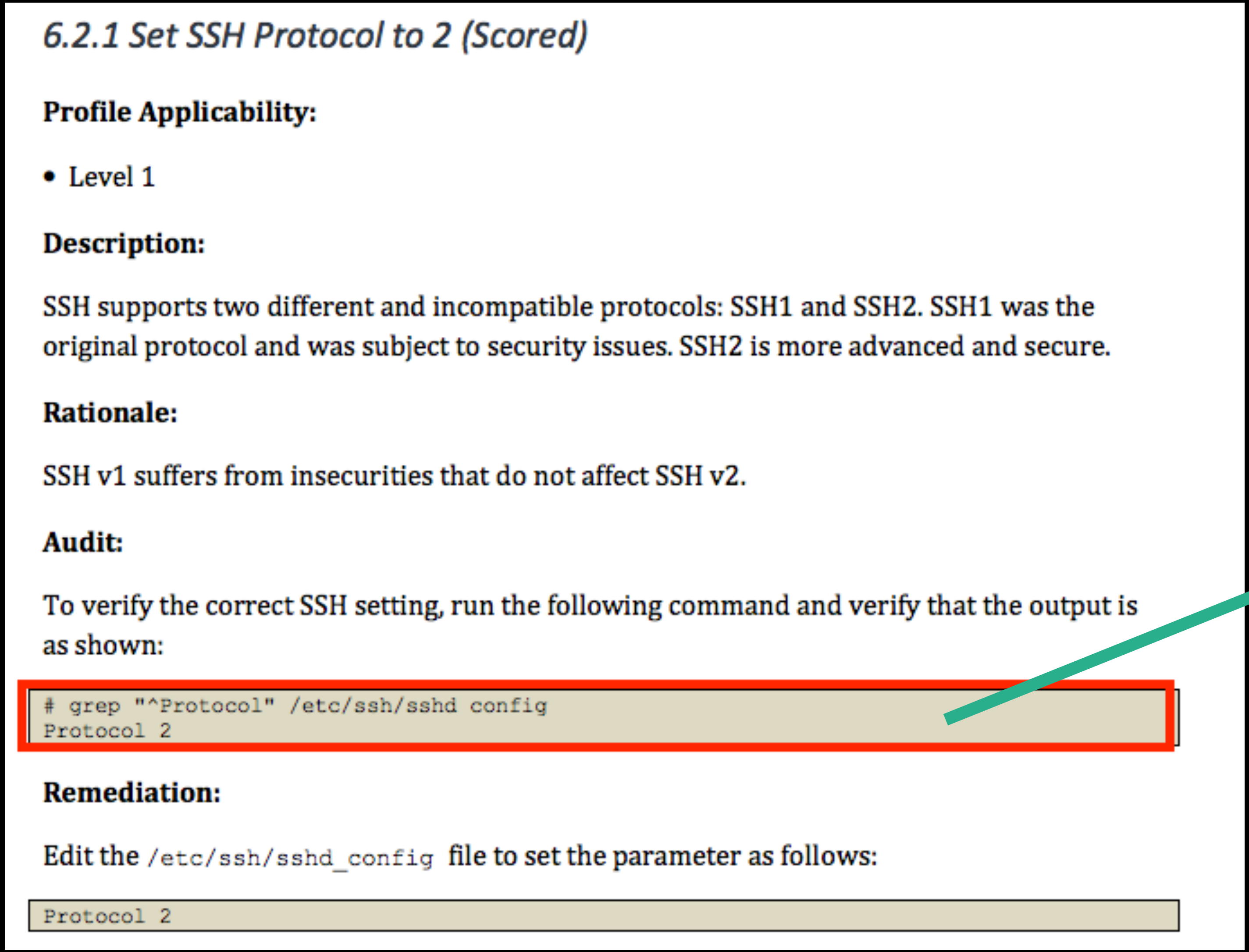
An `sshd_config` resource block declares the client OpenSSH configuration data to be tested:

```
describe sshd_config('path') do
  its('name') { should include('foo') }
end
```

where

- `name` is a configuration setting in `sshd_config`
- `('path')` is the non-default `/path/to/sshd_config`
- `{ should include('foo') }` tests the value of `name` as read from `sshd_config` versus the value declared in the test

Let’s try it!



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

Description:

SSH supports two different and incompatible protocols: SSH1 and SSH2. SSH1 was the original protocol and was subject to security issues. SSH2 is more advanced and secure.

Rationale:

SSH v1 suffers from insecurities that do not affect SSH v2.

Audit:

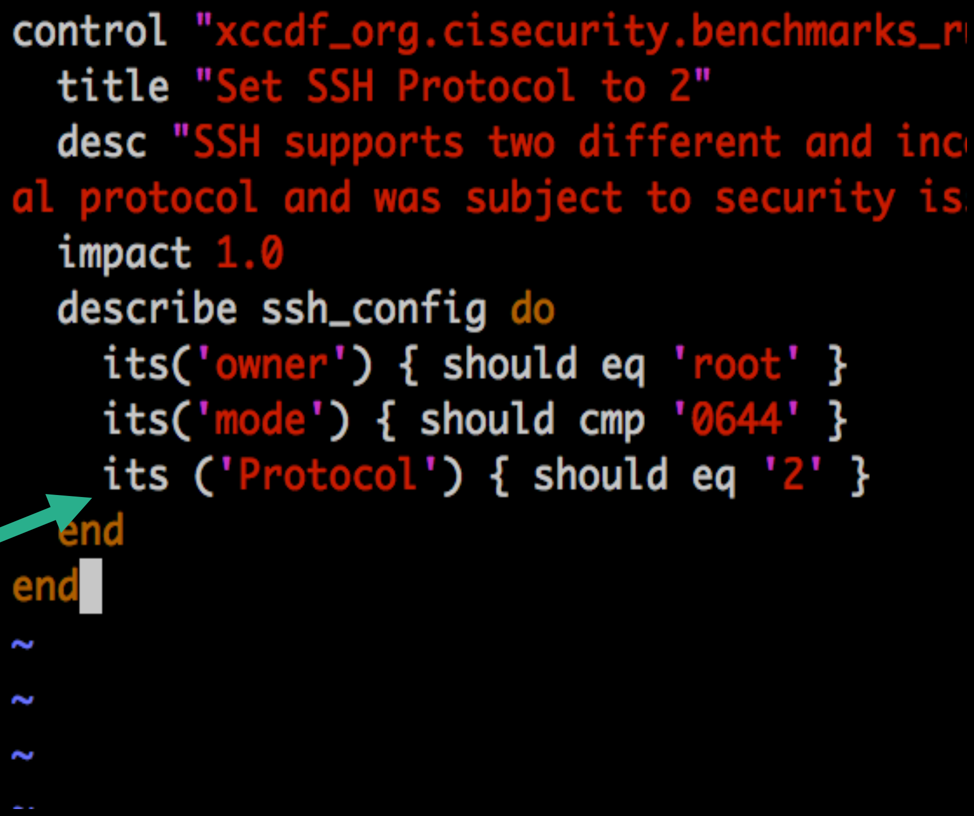
To verify the correct SSH setting, run the following command and verify that the output is as shown:

```
# grep "^Protocol" /etc/ssh/sshd config
Protocol 2
```

Remediation:

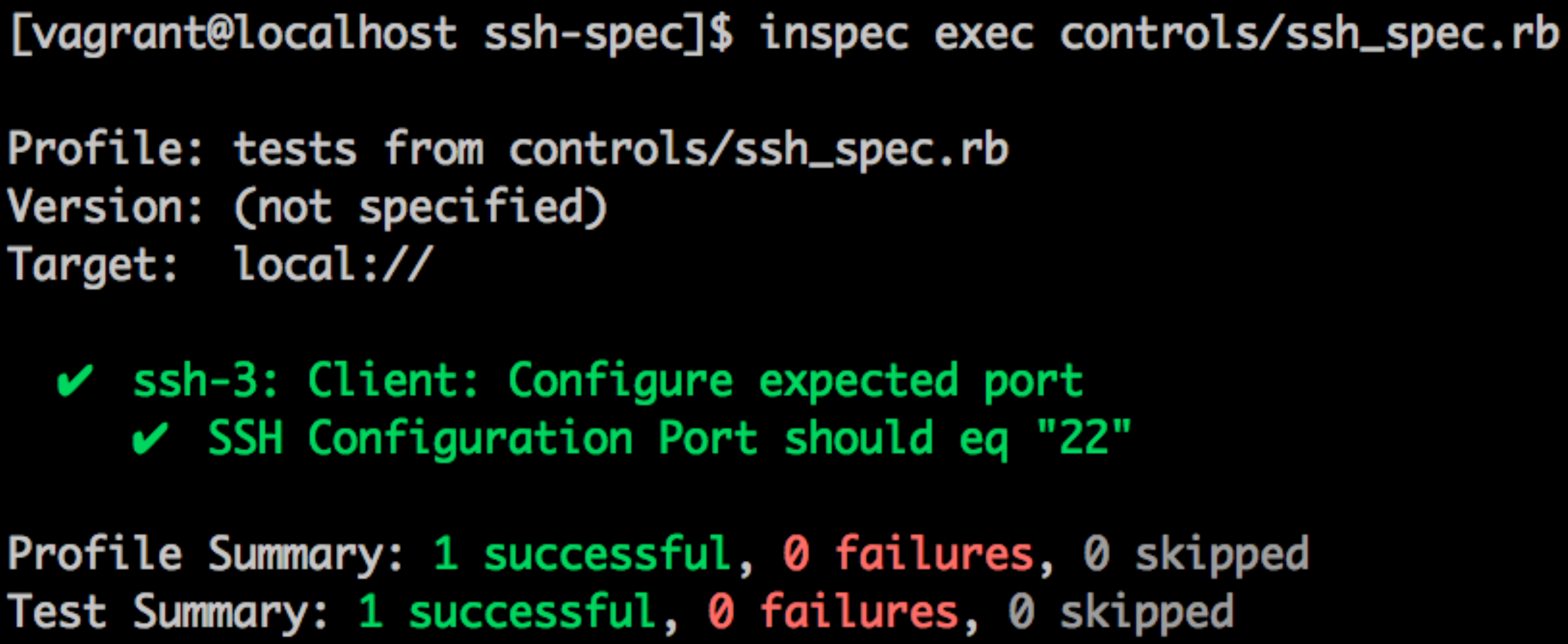
Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
Protocol 2
```



```
control "xccdf_org.cisecurity.benchmarks_r
  title "Set SSH Protocol to 2"
  desc "SSH supports two different and inc
al protocol and was subject to security is
  impact 1.0
  describe ssh_config do
    its('owner') { should eq 'root' }
    its('mode') { should cmp '0644' }
    its ('Protocol') { should eq '2' }
  end
end
~
~
~
```

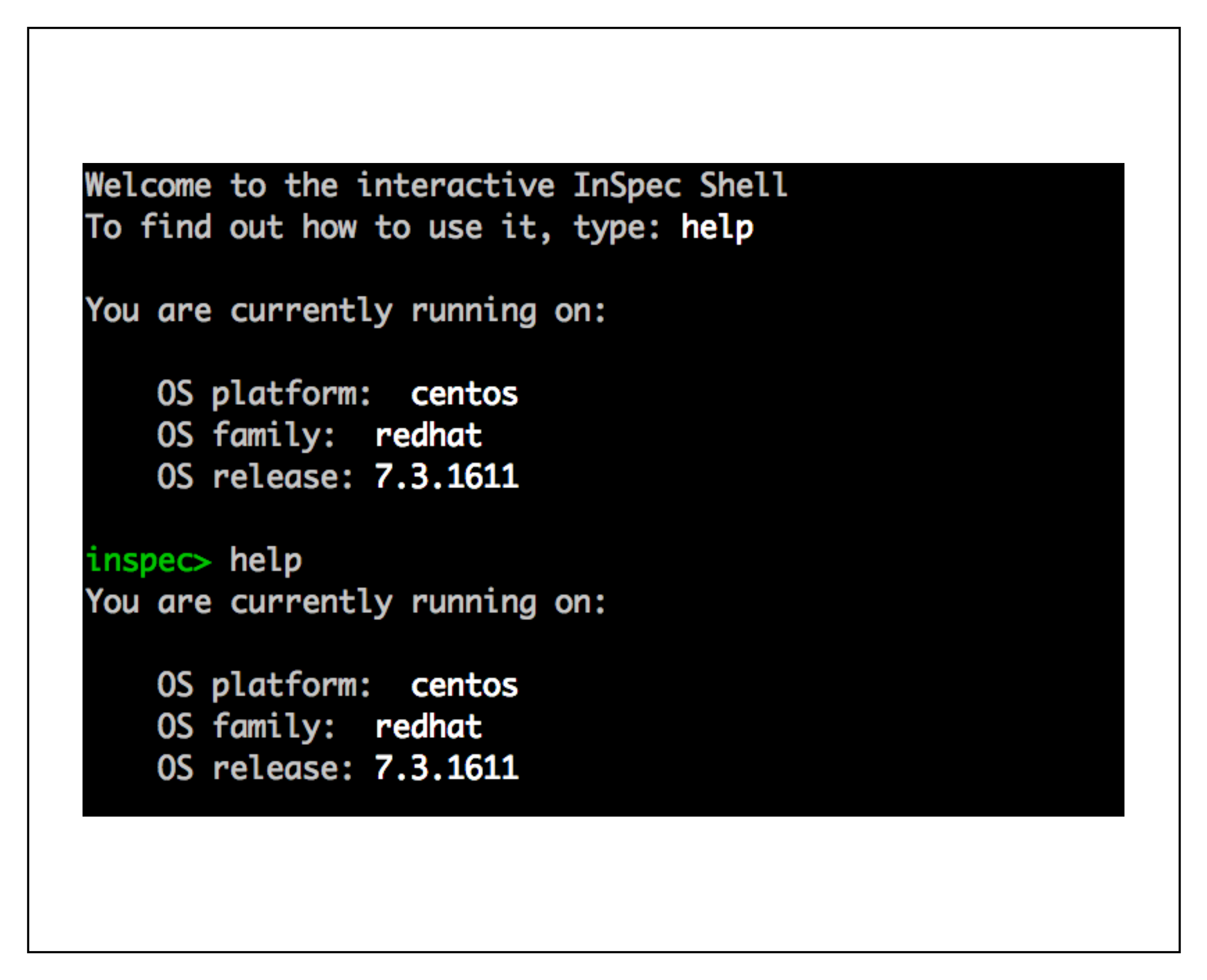
Inspec exec



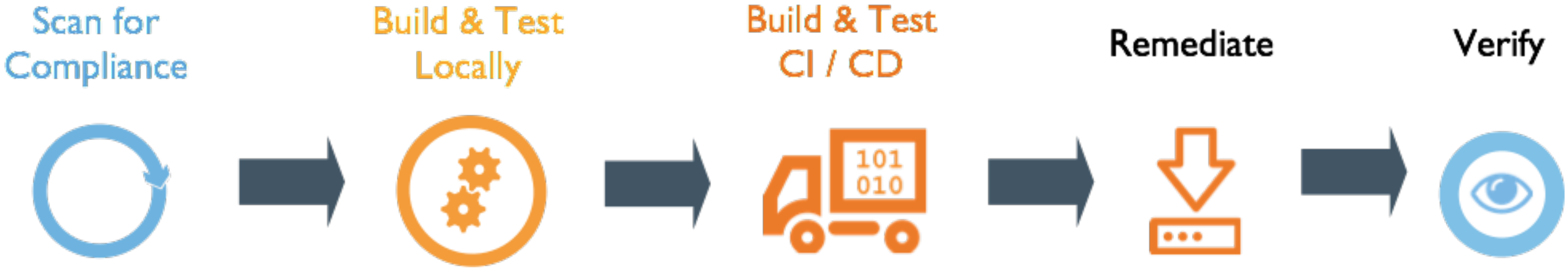
```
[vagrant@localhost ssh-spec]$ inspec exec controls/ssh_spec.rb

Profile: tests from controls/ssh_spec.rb
Version: (not specified)
Target:  local://

  ✔  ssh-3: Client: Configure expected port
     ✔  SSH Configuration Port should eq "22"

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

Inspec Shell

- Interactive shell – pry based REPL
- Used to quickly run Inspec controls and tests



```
Welcome to the interactive InSpec Shell
To find out how to use it, type: help

You are currently running on:

    OS platform:  centos
    OS family:  redhat
    OS release: 7.3.1611

inspec> help
You are currently running on:

    OS platform:  centos
    OS family:  redhat
    OS release: 7.3.1611
```



Scan for Compliance

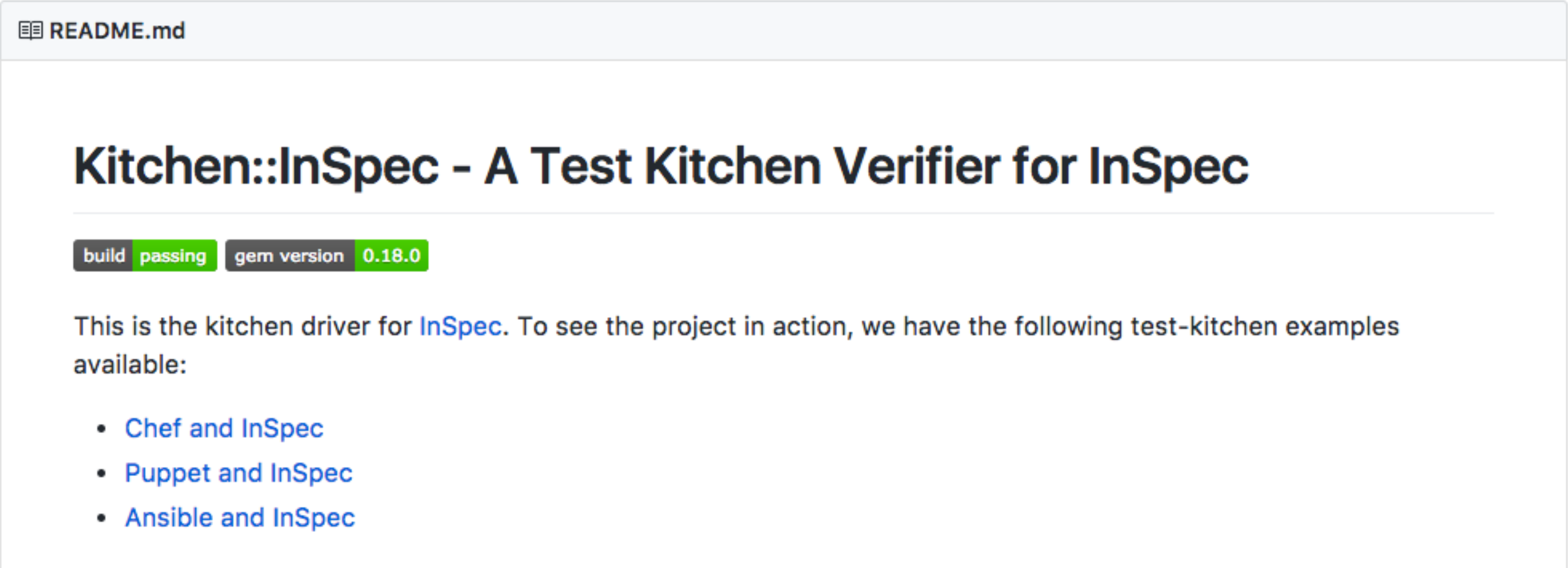
Build & Test Locally

Build & Test CI / CD

Remediate

Verify

Infrastructure as Code <-> Compliance as Code



README.md

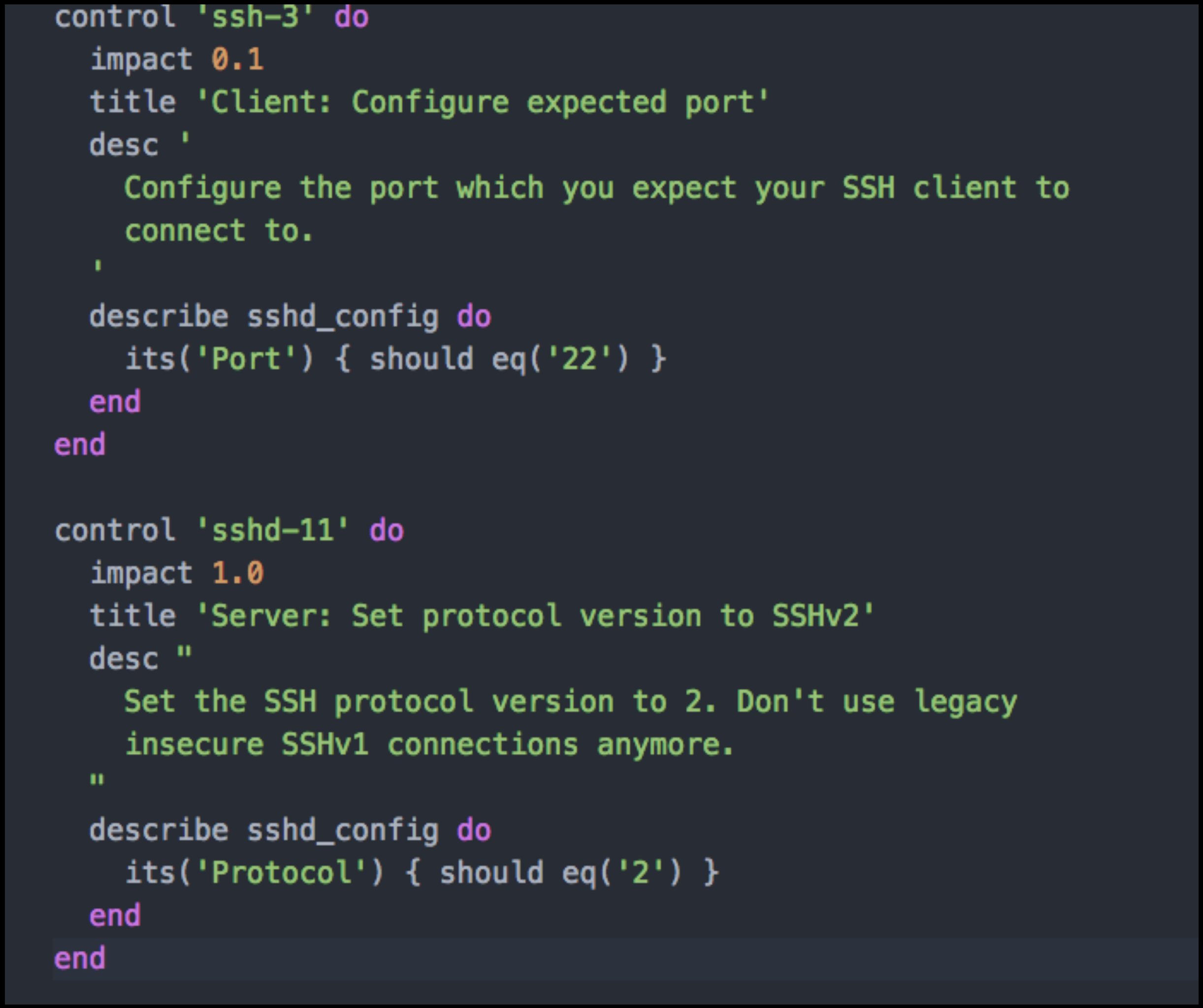
Kitchen::InSpec - A Test Kitchen Verifier for InSpec

This is the kitchen driver for InSpec. To see the project in action, we have the following test-kitchen examples available:

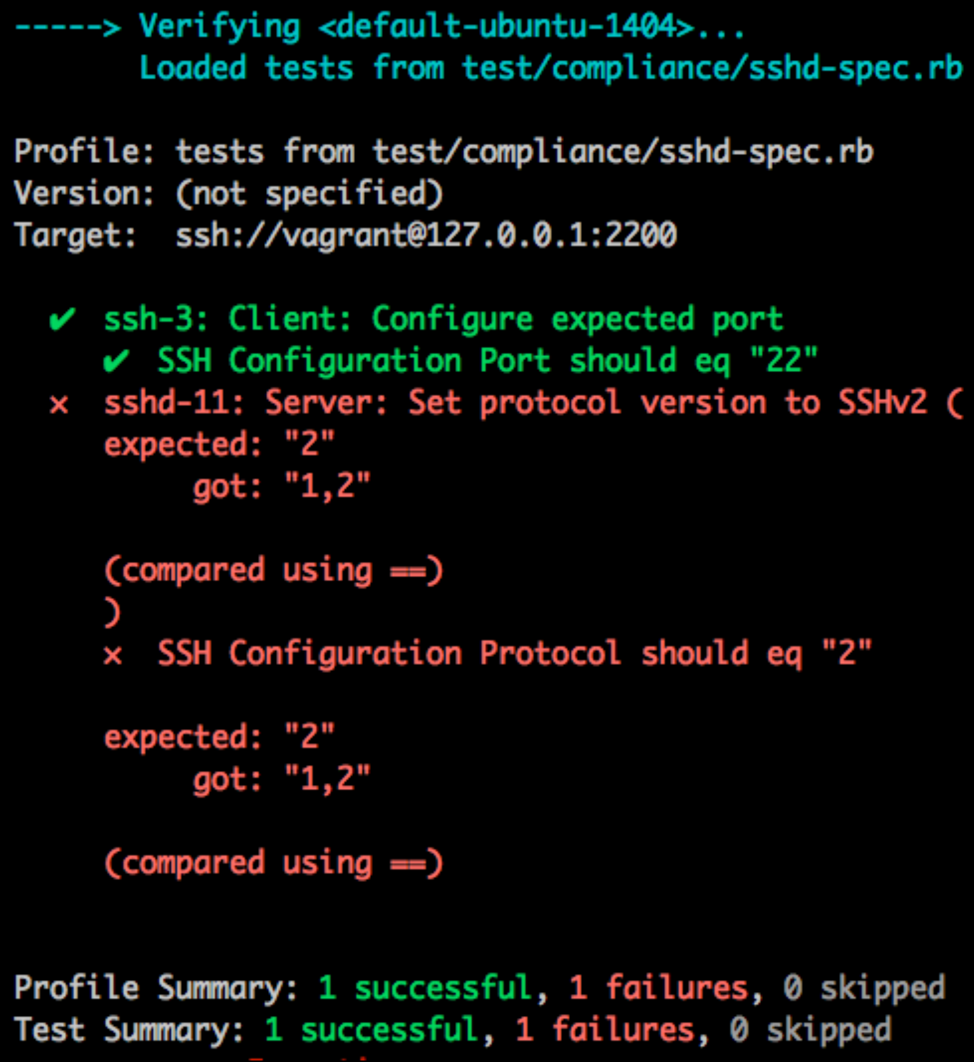
- Chef and InSpec
- Puppet and InSpec
- Ansible and InSpec

Test-Driven Development

Test-Driven Development – Test Locally

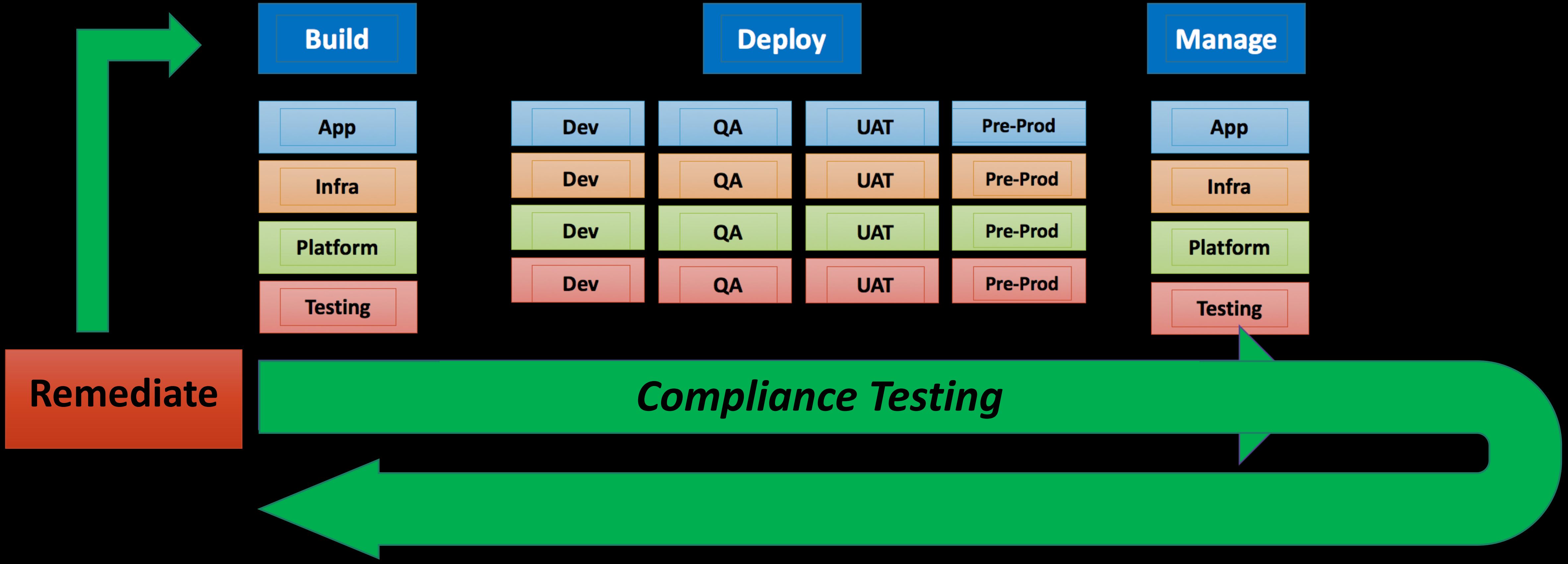


```
control 'ssh-3' do
  impact 0.1
  title 'Client: Configure expected port'
  desc '
    Configure the port which you expect your SSH client to
    connect to.
  '
  describe sshd_config do
    its('Port') { should eq('22') }
  end
end

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
```



```
-----> Verifying <default-ubuntu-1404>...
       Loaded tests from test/compliance/sshd-spec.rb

Profile: tests from test/compliance/sshd-spec.rb
Version: (not specified)
Target:  ssh://vagrant@127.0.0.1:2200

  ✔  ssh-3: Client: Configure expected port
     ✔  SSH Configuration Port should eq "22"
  ×  sshd-11: Server: Set protocol version to SSHv2 (
     expected: "2"
          got: "1,2"

     (compared using ==)
     )
     ×  SSH Configuration Protocol should eq "2"

     expected: "2"
          got: "1,2"

     (compared using ==)


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

The Path to Continuous Compliance



Build

- App
- Infra
- Platform
- Testing

Deploy

Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod
Dev	QA	UAT	Pre-Prod

Manage

- App
- Infra
- Platform
- Testing

Remediate

Compliance Testing

Questions?

Thank You!