INSPEC

InSpec turns compliance and security requirements into code

The InSpec language lets you specify compliance and security requirements as code. Test large-scale environments while still moving at velocity. Integrate compliance and security requirements into your automated deployment pipeline. When compliance is code, you can identify issues during development and not after the fact.

- **Translate compliance into code.** InSpec is an open-source testing framework with a human-readable language for specifying compliance, security and policy requirements. When compliance is expressed as code, you can integrate it into your deployment pipeline and automatically test for adherence to security policies.
- **Clearly express statements of policy.** When compliance is code, rules are unambiguous and can be understood by everyone on the team. Developers know what standards they're expected to meet and auditors know exactly what is being tested. Replace spreadsheets filled with abstract descriptions with tangible tests that have a clear intent.
- **Find issues early.** Automated compliance tests can start at the beginning of the development cycle. You'll detect any issues well before your code goes into production, when problems are expensive and time consuming to fix.
- **Write code quickly.** The InSpec language includes a collection of resources that help you write audit controls quickly and easily. You can also create custom resources and overlays for your own particular situations. Attributes let you include site-specific configuration details such as credentials.
- **Run code anywhere.** InSpec code runs on multiple platforms, including Linux, Windows and others. It has a flexible execution model: InSpec can be invoked by the Chef client or, in agentless mode, by using SSH, WinRM, or Docker access.
- **Inspect machines, data, and APIs.** Knowing that your physical servers are in compliance is, of course, essential. But your applications rely on more than just server configurations. With InSpec, you can assess data in a database or inspect the configuration of virtual resources by using their API. For example, you can check security group settings on your IaaS infrastructure.

Here is an InSpec rule that ensures that insecure services and protocols, such as telnet, are not used.

```
describe package('telnetd') do
 it { should_not be_installed }
end

describe inetd_conf do
 its('telnet') { should eq nil }
end
```

The Payment Card Industry Data Security Standard (PCI DSS) requires that cardholder data that is sent across open, public networks be encrypted. Here is an InSpec rule that ensures that the web server is only listening on well-secured ports.

```
describe port(80) do
 it { should_not be_listening }
end

describe ssl(port: 443) do
  it { should be_enabled }
end
```

Compliance at velocity requires that members of different teams, such as development, operations, compliance and security, all have access to compliance rules. You can add metadata to InSpec rules to ensure that everyone can understand the requirements. For example, here is an InSpec rule to specify that only SSH version 2 is acceptable.

```
control 'sshd-11' do
 impact 1.0
 title 'Server: Set protocol version to SSH v2'
 desc 'Disallow insecure SSHv1 connections'
 describe sshd_conf do
   its('Protocol') { should eq('2') }
 end
end
```

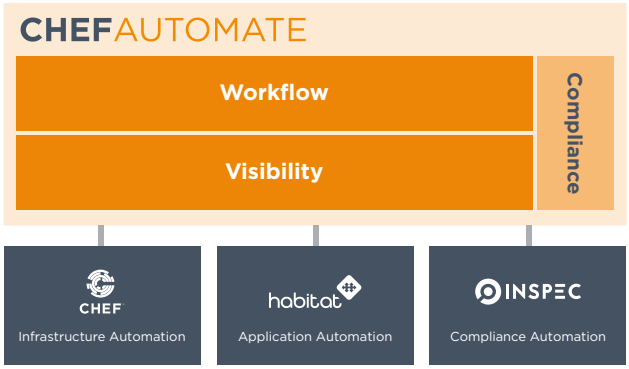
CHEF

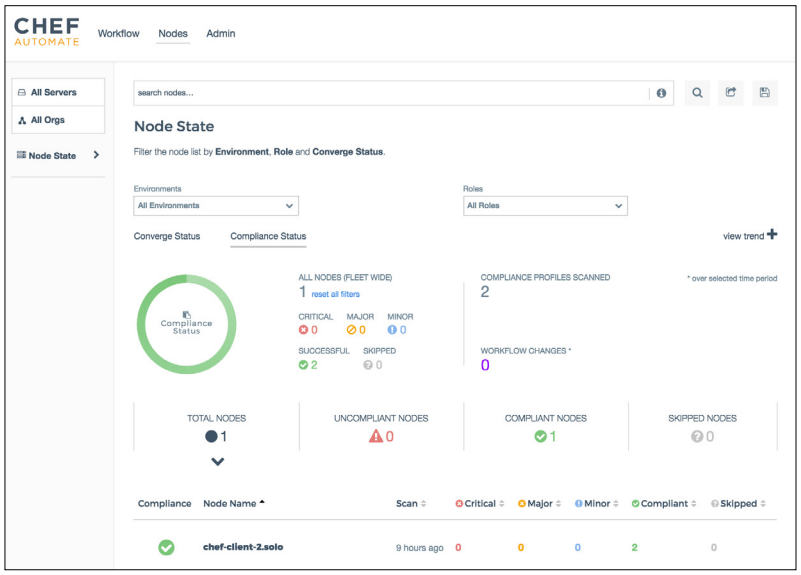
619 Western Avenue Suite 400 | Seattle, WA 98104 | PH: 206.508.4799 | sales@chef.io | www.chef.io

InSpec is fully open source with an Apache 2.0 license. It is a foundation of Chef Automate, enterprise software for high-velocity continuous deployment and compliance automation. Chef Automate integrates InSpec-based compliance automation into a DevOps operating model.

With its single, unified workflow, Chef Automate ensures compliance throughout development and deployment by making it a part of the build process. Any stage of the Chef Automate pipeline can include InSpec compliance checks to make sure that every environment is configured in accordance to policy.

The Chef Automate graphical interface and dashboards give you visibility into all aspects of your deployment process, including the effects that infrastructure automation and application automation have on your company's compliance policies. Here is an example of a Chef Automate compliance report that describes the status of the nodes in a network.





As an example of how to use this report, you can remediate any problems with the Chef development kit (Chef DK). Chef DK contains all the tools you need to create and test your code locally. You can then send the remediation through the Chef Automate pipeline to further test and then deploy your changes. The Chef Automate dashboards give you visibility into everything that's happening.

In addition, Chef Automate manages dependencies and communication across multiple feature teams and functional areas such as dev, ops, security and compliance. Everyone involved in ensuring that your applications and infrastructure follow regulatory and company standards can actively participate in a project. To control access, Chef Automate uses SAML-based authentication and authorization.

To help you quickly reach compliance at velocity, Chef Automate comes with profiles for Linux and Windows (both Computer Internet Security and base profiles). These prewritten rule sets let you start testing your systems immediately, without having to write your own InSpec tests. Profiles can be customized to suit your own company's policies. You can add, modify or remove specific requirements.

Finally, Chef Automate comes with comprehensive 24×7 support.

"The tools we use reinforce the behavior; the behavior reinforces the tool. If you want to change your behavior, change your tools."

— Adam Jacob, CTO, Chef

Learn more at www.chef.io/inspec

CHEF
619 Western Avenue Suite 400 | Seattle, WA 98104 | PH: 206.508.4799 | sales@chef.io | www.chef.io