# Contributing to InSpec

We are glad you want to contribute to InSpec! This document will help answer common questions you may have during your first contribution.

As a reminder, all participants are expected to follow the [Code of Conduct](https://github.com/inspec/inspec/blob/master/CODE\_OF\_CONDUCT.md).

## Submitting Issues

We utilize \*\*Github Issues\*\* for issue tracking and contributions. You can contribute in two ways:

1. Reporting an issue or making a feature request [here](https://github.com/chef/inspec/issues/new).

2. Adding features or fixing bugs yourself and contributing your code to InSpec.

We ask you not to submit security concerns via Github. For details on submitting potential security issues please see <https://www.chef.io/security/>

## Contribution Process

We have a 3 step process for contributions:

1. Commit changes to a git branch, making sure to sign-off those changes for the [Developer Certificate of Origin](#developer-certification-of-origin-dco).

2. Create a Github Pull Request for your change, following the instructions in the pull request template.

3. Perform a [Code Review](#code-review-process) with the project maintainers on the pull request.

### Pull Request Requirements

Chef Projects are built to last. We strive to ensure high quality throughout the experience. In order to ensure this, we require that all pull requests to Chef projects meet these specifications:

1. \*\*Tests:\*\* To ensure high quality code and protect against future regressions, we require all the code in Chef Projects to have at least unit test coverage. See the [test/unit](https://github.com/chef/inspec/tree/master/test/unit)

directory for the existing tests and use ```bundle exec rake test``` to run them.

2. \*\*Green CI Tests:\*\* We use [Travis CI](https://travis-ci.org/) and/or [AppVeyor](https://www.appveyor.com/) CI systems to test all pull requests. We require these test runs to succeed on every pull request before being merged.

3. \*\*Up-to-date Documentation:\*\* Every code change should be reflected in an update for our [documentation](https://github.com/chef/inspec/tree/master/docs). We expect PRs to update the documentation with the code change.

In addition to this it would be nice to include the description of the problem you are solving

with your change. You can use [Issue Template](https://github.com/chef/inspec/tree/master/ISSUE\_TEMPLATE.md) in the description section

of the pull request.

### Code Review Process

Code review takes place in Github pull requests. See [this article](https://help.github.com/articles/about-pull-requests/) if you're not familiar with Github Pull Requests.

Once you open a pull request, project maintainers will review your code and respond to your pull request with any feedback they might have. The process at this point is as follows:

1. Two thumbs-up (:+1:) are required from project maintainers. See the master maintainers document for InSpec projects at <https://github.com/chef/inspec/blob/master/MAINTAINERS.md>.

2. When ready, your pull request will be merged into `master`, we may require you to rebase your PR to the latest `master`.

3. Once the PR is merged, you will be included in `CHANGELOG.md`.

If you would like to learn about when your code will be available in a release of InSpec, read more about [InSpec Release Cycles](#release-cycles).

### Developer Certification of Origin (DCO)

Licensing is very important to open source projects. It helps ensure the software continues to be available under the terms that the author desired.

Chef uses [the Apache 2.0 license](https://github.com/chef/chef/blob/master/LICENSE) to strike a balance between open contribution and allowing you to use the software however you would like to.

The license tells you what rights you have that are provided by the copyright holder. It is important that the contributor fully understands what rights they are licensing and agrees to them. Sometimes the copyright holder isn't the contributor, such as when the contributor is doing work on behalf of a company.

To make a good faith effort to ensure these criteria are met, Chef requires the Developer Certificate of Origin (DCO) process to be followed.

The DCO is an attestation attached to every contribution made by every developer. In the commit message of the contribution, the developer simply adds a Signed-off-by statement and thereby agrees to the DCO, which you can find below or at <http://developercertificate.org/>.

```

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I

have the right to submit it under the open source license

indicated in the file; or

(b) The contribution is based upon previous work that, to the

best of my knowledge, is covered under an appropriate open

source license and I have the right under that license to

submit that work with modifications, whether created in whole

or in part by me, under the same open source license (unless

I am permitted to submit under a different license), as

Indicated in the file; or

(c) The contribution was provided directly to me by some other

person who certified (a), (b) or (c) and I have not modified

it.

(d) I understand and agree that this project and the contribution

are public and that a record of the contribution (including

all personal information I submit with it, including my

sign-off) is maintained indefinitely and may be redistributed

consistent with this project or the open source license(s)

involved.

```

For more information on the change see the Chef Blog post [Introducing Developer Certificate of Origin](https://blog.chef.io/2016/09/19/introducing-developer-certificate-of-origin/)

#### DCO Sign-Off Methods

The DCO requires a sign-off message in the following format appear on each commit in the pull request:

```

Signed-off-by: Julia Child <juliachild@chef.io>

```

The DCO text can either be manually added to your commit body, or you can add either \*\*-s\*\* or \*\*--signoff\*\* to your usual git commit commands. If you forget to add the sign-off you can also amend a previous commit with the sign-off by running \*\*git commit --amend -s\*\*. If you've pushed your changes to Github already you'll need to force push your branch after this with \*\*git push -f\*\*.

### Obvious Fix Policy

Small contributions, such as fixing spelling errors, where the content is small enough to not be considered intellectual property, can be submitted without signing the contribution for the DCO.

As a rule of thumb, changes are obvious fixes if they do not introduce any new functionality or creative thinking. Assuming the change does not affect functionality, some common obvious fix examples include the following:

- Spelling / grammar fixes

- Typo correction, white space and formatting changes

- Comment clean up

- Bug fixes that change default return values or error codes stored in constants

- Adding logging messages or debugging output

- Changes to 'metadata' files like Gemfile, .gitignore, build scripts, etc.

- Moving source files from one directory or package to another

\*\*Whenever you invoke the "obvious fix" rule, please say so in your commit message:\*\*

```

------------------------------------------------------------------------

commit 370adb3f82d55d912b0cf9c1d1e99b132a8ed3b5

Author: Julia Child <juliachild@chef.io>

Date: Wed Sep 18 11:44:40 2015 -0700

Fix typo in the README.

Obvious fix.

------------------------------------------------------------------------

```

## Release Cycles

### Release Formats

Our primary shipping vehicle is operating system specific packages that includes all the requirements of InSpec. We call these Omnibus packages, and they are available from [downloads.chef.io](https://downloads.chef.io/inspec). InSpec is also bundled with recent Chef Client and ChefDK toolkits.

InSpec is also available as a [Docker image](https://hub.docker.com/r/chef/inspec) and a [Habitat package](https://bldr.habitat.sh/#/pkgs/chef/inspec/latest).

Finally, we also release our software as gems to [Rubygems](https://rubygems.org/) but we strongly recommend using one of the previously packages to ensure a smooth experience.

### Release Timing

We typically aim to make a release every Thursday; however, if there are no changes, the release will be skipped. Releases may also be delayed due to holidays, staffing constraints, or technical problems.

### Versioning

This information is provided for context only. Contributors are not expected to manage the version number of InSpec - the InSpec team and the CI system will handle any version changes needed.

Our version numbering roughly follows [Semantic Versioning](http://semver.org/) standard. Our standard version numbers look like X.Y.Z which mean:

- X is a major release, which may not be fully compatible with prior major releases

- Y is a minor release, which adds new features and may include bug fixes

- Z is a patch release, which adds just bug fixes

After shipping a release of InSpec we bump at least the `patch` version by one to start development of the next release. We do a release approximately every week. Announcements of releases are made to the [InSpec mailing list](https://discourse.chef.io/c/chef-release) when they are available.

## InSpec Community

InSpec is made possible by a strong community of developers, system administrators, auditor and security experts. If you have any questions or if you would like to get involved in the InSpec community you can check out:

- [InSpec Mailing List](https://discourse.chef.io/c/inspec)

- [Chef Community Slack](https://community-slack.chef.io/)

Also here are some additional pointers to some awesome Chef content:

- [InSpec Docs](https://www.inspec.io/docs/)

- [Learn Chef](https://learn.chef.io/)

- [Chef Website](https://www.chef.io/)