

How to Contribute

Welcome and thank you for your interest in contributing to jclouds! This guide will take you through the process of making contributions to the jclouds code base.

Contributor license agreement

All contributions and patches attached to a [JIRA](#) issue are assumed to be under the [Apache ICLA](#), so make sure you read and understand it before making your contributions.

How to contribute

We use [GitHub](#) and [JIRA](#) on the jclouds project.

0. Talk to us

If it's your first contribution or it's a particularly big/complex contribution, things typically go much more smoothly when they start off with a conversation. Visit our [Community](#) page to see how you can contact us via IRC or email.

1. Review our best practices

Read [Best Practices](#). jclouds has been developed over many years and naturally things have been done differently at different times by different people. It's hard to know what the current best practice is. If you don't read through our best practices, you might wind up having to redo a significant portion of your PR because you based it on an older way of doing things. You're also welcome to add to the best practices yourself as you discover them.

2. Create an issue in JIRA

Your first step is to create or find an issue in [JIRA](#) for your feature request or fix. If it's a simple cleanup, this isn't necessary, but it's good to see if your change fixes an existing issue anyway. If it's a formatting issue alone, consider creating an issue for us to fix our checkstyle configuration to catch it in the future.

3. Create a pull request in GitHub

Next, you'll want to create a pull request (PR) in GitHub for the change you're interested in making. The comment section of the PR must contain a link to the JIRA issue (if it has one). Please also reference the issue [in the commit message](#), and make sure it properly describes the changes that have been made and their purpose ([here](#) are some guidelines). If you need the commit backported to one of the release branches (e.g. 1.6.x), make a note about that in the comment too and set the Affects Version/s of the issue accordingly.

Some good references for working with GitHub are below. We ask that you keep your change rebased to master as much as possible, and we will ask you to rebase again if master has moved before accepting your patch.

- [Setting Up Git with GitHub](#)
- [Forking a Repository](#)
- [Submitting Pull Requests](#)
- [Rebasing your Branch](#)

Checkout our [recommended Git workflow](#) for an example workflow you can use when working in your pull request.

4. Comment the issue in JIRA

Finally, add a comment in the JIRA issue with a link to the pull request so we know the code is ready to be reviewed.

The review process

The jclouds community will review your pull request before it is merged. This process can take a while, so please be patient.

If we are slow to respond, feel free to join us in irc on [the #jclouds Slack channel](#) and ask for help. Feel free to also email our dev [mailing list](#): dev@jclouds.apache.org. To subscribe send an email to jclouds-dev-subscribe@apache.org.

During the review process you may be asked to make some changes to your submission. While working through feedback, it can be beneficial to create new commits so the incremental change is obvious. This can also lead to a complex set of commits, and having an atomic change per commit is preferred in the end. Use your best judgement and work with your reviewer as to when you should revise a commit or create a new one.

A pull request is considered ready to be merged once it gets at least one +1 from a reviewer. Once all the changes have been completed and the pull request is accepted, it **must be rebased to the latest upstream version**. It is also a good idea to squash all the commits into a single one, since this will allow us to generate a clean patch and merge it properly.