

Gerrit Code Review Plugin

Why yet another plugin?

I wanted to use the Gerrit CI validation workflow with potentially any project, including Gerrit plugins or anybody else wanting to adopt it. I could just "copy & paste" our Groovy workflow, however, that does not seem a sensible and long-term approach. I wanted to have "something more" than a pure triggering mechanism: I wanted to extend the power of Jenkinsfile with the Gerrit review workflow verbs.

Why not?

Why should I write yet another Gerrit/Jenkins plugin? Isn't Gerrit Trigger Plugin (<https://wiki.jenkins.io/display/JENKINS/Gerrit+Trigger>) enough? We couldn't use it against gerrit-review.googlesource.com because stream events are just not accessible.

There are unresolved issues about:

- **Stability**
stream-events are based on SSH, which isn't scalable or reliable against downtime, doesn't allow smart Git routing
- **Complex configuration**
requires a node-level and project-level configuration, which is orthogonal to the Jenkinsfile pipeline
- **Integration**
using it inside a Jenkinsfile isn't that straightforward and multi-branch projects aren't supported either

Why a new name?

The new name "Gerrit Code Review Plugin" indicates Gerrit as the first-class citizen of the Jenkins ecosystem, and not just as an "extra triggering or validation" logic of a Jenkins job.

Jenkinsfile / Scripted pipeline example, with Gerrit Code Review integration:

```
node {
  checkout scm
  try {
    stage('Hello') {
      echo 'Hello World'
    }
    gerritReview score:1
  } catch (e) {
    gerritReview score:-1
    throw e
  }
}
```

Jenkinsfile / Declarative pipeline example, with Gerrit Code Review integration:

```
pipeline {
  agent any
  stages {
    stage('Example') {
      steps {
        echo 'Hello World'
      }
    }
  }
  post {
    success { gerritReview score:1 }
    failure { gerritReview score:-1 }
  }
}
```

One key aspect will be: stateless, configuration-less apart the standard SCM configuration settings. That means that multiple Jobs, multiple branches of the same Job, can have their own Gerrit integration defined and working out-of-the-box.

No more people asking "how do I configure the Gerrit integration"? it will just work.

General Availability

I have presented the first prototype of this new plugin at the Jenkins World Conference in San Francisco back in 2017 inside my "Data-Driven Pipeline workshop."

(<https://jenkinsworld20162017.sched.com/event/APTd/data-driven-pipeline-workshop-free>)

The first version was published and available in the Jenkins Plugin Registry since April 2018 and has been used so far by hundreds of developers around the world that provided already very useful feedback and started contributing with pull-requests.

Release Notes and RoadMap

See below the roadmap I have envisaged for this plugin. However, I do follow a fully Agile approach where it is really the user that drives the development of the product, rather than a project plan.

The roadmap below can then change based on what are your needs and the adoption by the community.

v1.0 - Planned

Highlights

First fully functional release of the Gerrit Code Review integration that can be functionally equivalent to the historical Gerrit Trigger Plugin but with a focus on Jenkinsfile pipelines.

v0.7 - Planned

Highlights

Support for internal networks where any calls outside the Jenkins node can be made via Proxy servers.

v0.6 - Planned

Highlights

Support for builds against Gerrit replicas, where API for review feedback can be different from the one used for fetching the changes.

The challenge of this release is how to automatically discover the upstream Gerrit node purely based on the fetch URL.

v0.5 - Planned

Highlights

Introduction of the Git/SSH protocol support and use for review feedback.

v0.4 - Planned

Highlights

Support for BlueOcean, including change description, hyperlink and owner visible from Jenkins UI.

v0.3.4 - Released - 10 October 2019

New features:

- 40a5144 Set automatically change details to build environment variables, align with the Gerrit Trigger Plugin variables names
- JENKINS-54722 Generate the standard change hyperlink to the original Gerrit change that triggered a build

Fixes:

- e6c56b7 Running web hook as System ACL

v0.3.3 - Released - 8 March 2019

Fixes:

- JENKINS-5479 Fix exception thrown when trying to parse unhandled events
- 7d31723 Allow for remote URLs containing .git when matching webhook project events
- 4ee2b3f Log user account who posted the webhook
- 520c932 Fix processing WebHooks POST with unknown body length

v0.3.2 - Released - 3 November 2018

Fixes

- [JENKINS-54432](#) Restore normal behaviour when using Gerrit as anonymous SCM source

v0.3.1 - Released - 29 October 2018

Highlights

Integration of the support for Gerrit in conjunction with the Gerrit Trigger Plugin and single branch pipelines. Improvements on the support for Gerrit API with the introduction of multiple labels in a single review.

Features

- [eea176e](#) Support for multiple labels in a single `gerritReview` step invocation
- [8a9d947](#) Support for integration with the Gerrit Trigger Plugin
- [427c0fc](#) Remove the support for deprecated `GerritDSL` style invocation (e.g. `gerrit.review` step)

Fixes

- [JENKINS-54224](#) restapi: cleanup restapi usage

v0.2.1 - Released - 27 October 2018

Highlights

Bug-fix release from feedback and contributions of the adopters of the v0.2 release.

Fixes

- [3359ba3](#) - Amendments to the documentation to explain the first-time setup
- [e320d18](#) - Allow feedback to an open change without changing current score / label
- [fc0bc91](#) - Restapi: allow the use of remote Gerrit server with non-root prefix
- [JENKINS-54214](#) - Avoid NPE when credentials are not provided
- [JENKINS-54212](#) - Fix detection of project name with refupdate

v0.2 - Released - 23 October 2018

Highlights

Many bug fixes thanks to the initial adoption in real life use-cases. Improvement of the compliance of the plugin with the rest of the ecosystem of Jenkinsfile steps.

One major addition is the support for Jenkins Declarative Pipelines and the deprecation of the old DSL-based integration, which was not compatible with the way plugins should be written and integrated with the rest of the CPS scripts and pipelines.

Features

- [JENKINS-54070](#) - Support for Jenkinsfile declarative pipelines
- [JENKINS-50783](#) - Support for file-based comments from Jenkinsfile steps

Fixes

- [JENKINS-49695](#) - Could not fetch branches from source while using Discover open changes
- [JENKINS-50930](#) - Multibranch fails if gerrit lives at a subdirectory
- [JENKINS-49983](#) - NullPointerException from gerrit plugin due to lack of configuration
- [JENKINS-49985](#) - hudson.remoting.ProxyException: org.codehaus.groovy.runtime.typehandling.GroovyCastException
- [JENKINS-49713](#) - gerrit-plugin exposes change-requests as normal branches instead of pull-requests
- [JENKINS-49692](#) - Could not update folder level actions from source

v0.1.1 - Released - 4 April 2018

Initial version of the Gerrit Code Review plugin for Jenkins CI, with support for the scripted pipeline with DSL.

What about the Gerrit Trigger Plugin?

The **new plugin is not going to replace the current Gerrit Trigger Plugin**, but would rather represent an alternative to simpler scenarios when you just require a standard Jenkinsfile Gerrit validation workflow. For all the current users of the Gerrit Trigger Plugin things wouldn't change, unless they need a more Jenkinsfile-integrated experience.

Issues

Below you can see a list of open issues related to the gerrit-plugin which are not resolved yet.

key	summary	type	created	updated	due	assignee	reporter	priority	status
-----	---------	------	---------	---------	-----	----------	----------	----------	--------

 Jira project doesn't exist or you don't have permission to view it.

[View these issues in Jira](#)