GitHub pull request builder plugin

You should probably migrate to GitHub Branch Source Plugin

This plugin builds pull requests in github and report results.

Plugin Information
View GitHub Pull Request Builder on the plugin site for more information.

The current version of this plugin may not be safe to use. Please review the following warnings before use:

- GitHub access tokens stored in in build.xml

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- Webhook secret stored in plain text
- CSRF vulnerability and lack of permission checks allows capturing credentials

This plugin is up for adoption. Want to help improve this plugin? Click here to learn more!

More **up to date** information is available on the:
Github README.md

Commands:

When a new pull request is opened in the project and the author of the pull request isn't white-listed, builder will ask "Can one of the admins verify this patch?".

- "ok to test" to accept this pull request for testing
- "test this please" for a one time test run
- "add to whitelist" to add the author to the whitelist

If the build fails for other various reasons you can rebuild.

- "retest this please" to start a new build

Installation:

- I recommend to create GitHub 'bot' user that will be used for communication with GitHub (however you can use your own account if you want).
- The user needs to have push rights for your repository (must be collaborator (user repo) or must have Push & Pull rights (organization repo)).
- If you want to use GitHub hooks have them set automatically the user needs to have administrator rights for your repository (must be owner (user repo) or must have Push, Pull & Administrative rights (organization repo))

- Install the plugin.
- Go to ``Manage Jenkins`` -> ``Configure System`` -> ``GitHub pull requests builder`` section.

- Add GitHub usernames of admins (these usernames will be used as defaults in new jobs).
- Under Advanced, you can modify:
    - The phrase for adding users to the whitelist via comment. (Java regexp)
    - The phrase for accepting a pull request for testing. (Java regexp)
    - The phrase for starting a new build. (Java regexp)
    - The crontab line. This specify default setting for new jobs.
- Under Application Setup
    - There are global and job default extensions that can be configured for things like:
        - Commit status updates
        - Build status messages
        - Adding lines from the build log to the build result message
        - etc.

Credentials

- If you are using Enterprise GitHub set the server api URL in ``GitHub server api URL``. Otherwise leave there ``https://api.github.com``.
- A GitHub API token or username password can be used for access to the GitHub API

- To setup credentials for a given GitHub Server API URL:
    - Click Add next to the ``Credentials`` drop down
        - For a token select ``Kind`` -> ``Secret text``
            - If you haven't generated an access token you can generate one in ``Test Credentials...``.
                - Set your 'bot' user's GitHub username and password.
                - Press the ``Create Access Token`` button
                - Jenkins will create a token credential, and give you the id of the newly created credentials. The default description is: ``serverAPIUrl + " GitHub auto generated token credentials"``.
        - For username/password us ``Kind`` -> ``Username with password``
            - The scope determines what has access to the credentials you are about to create
        - The first part of the description is used to show different credentials in the drop down, so use something semi-descriptive
        - Click ``Add``
    - Credentials will automatically be created in the domain given by the ``GitHub Server API URL`` field.
    - Select the credentials you just created in the drop down.
    - The first fifty characters in the Description are used to differentiate credentials per job, so again use something semi-descriptive
- Add as many GitHub auth sections as you need, even duplicate server URLs
- Save to preserve your changes.

Creating job:

- Create a new job.
- Add the project's GitHub URL to the "GitHub project" field (the one you can enter into browser. eg: "https://github.com/janinko/ghprb")
- Select Git SCM.
- Add your GitHub "Repository URL".
- Under Advanced, set "refspec" to

```
+refs/pull/*:refs/remotes/origin/pr/*
```

- In "Branch Specifier", enter

```
${sha1}
```

  or if you want to use the actual commit in the pull request, use

```
${ghprbActualCommit}
```

- Under "Build Triggers", check "Github pull requests builder".
    - Add admins for this specific job.
    - If you want to use GitHub hooks for automatic testing, read the help for "Use github hooks for build triggering" in job configuration. Then you can check the checkbox.
    - In Advanced, you can modify:
        - The crontab line for this specific job. This schedules polling to GitHub for new changes in Pull Requests.
        - The whitelisted users for this specific job.
        - The organisation names whose members are considered whitelisted for this specific job.
- Save to preserve your changes.
  Make sure you **DON'T** have "Prune remote branches before build" advanced option selected, since it will prune the branch created to test this build. As noted in issue #507 this may be achieved by unchecking 'Lightweight checkout'.

If you want to manually build the job, in the job setting check "This build is parameterized" and add string parameter named "sha1". When starting build give the "sha1" parameter commit id you want to build or refname (eg: "origin/pr/9/head").

Environment Variables

The plugin makes some very useful environment variables available.

- ghprbActualCommit
- ghprbActualCommitAuthor
- ghprbActualCommitAuthorEmail
- ghprbPullDescription
- ghprbPullId
- ghprbPullLink
- ghprbPullTitle
- ghprbSourceBranch
- ghprbTargetBranch
- ghprbCommentBody
- sha1

Job DSL Support

The plugin contains an extension for the Job DSL plugin to add DSL syntax for configuring the build trigger and the pull request merger post-build action.

Here is an example showing all DSL syntax elements:

```
job('example') {
    scm {
        git {
            remote {
                github('test-owner/test-project')
                refspec('+refs/pull/*:refs/remotes/origin/pr/*')
            }
            branch('${sha1}')
        }
    }
    triggers {
        githubPullRequest {
            admin('user_1')
            admins(['user_2', 'user_3'])
            userWhitelist('you@you.com')
            userWhitelist(['me@me.com', 'they@they.com'])
            orgWhitelist('my_github_org')
            orgWhitelist(['your_github_org', 'another_org'])
            cron('H/5 * * * *')
            triggerPhrase('OK to test')
            onlyTriggerPhrase()
            useGitHubHooks()
            permitAll()
            autoCloseFailedPullRequests()
            allowMembersOfWhitelistedOrgsAsAdmin()
            extensions {
                commitStatus {
                    context('deploy to staging site')
                    triggeredStatus('starting deployment to staging site...')
                    startedStatus('deploying to staging site...')
                    statusUrl('http://mystatussite.com/prs')
                    completedStatus('SUCCESS', 'All is well')
                    completedStatus('FAILURE', 'Something went wrong. Investigate!')
                    completedStatus('PENDING', 'still in progress...')
                    completedStatus('ERROR', 'Something went really wrong. Investigate!')
                }
            }
        }
    }
}
```

See the README on GitHub for the latest information.

Changelog

>= 1.25

Changelog for `1.25` and upward is maintained on the plugins github page

1.24

Use signature checking for webhooks if desired.
Add custom messages to status updates and a custom url field.

1.23

Use credentials plugin for username/password combinations and tokens.
Support multiple GitHub endpoints.

1.22.x

Fix issue where if a project was disabled the Jenkins Trigger process would crash
Fix commit status context
Add one line test results for downstream builds
Miscellaneous bug fixes

1.22

Move commit status over to extension form. It is now configurable, satisfying #81, #73, and #19 at least.

1.21

Move all commenting logic out into extensions.

1.20.1

Null Pointer fix for trigger.
Added clarity to error message when access is forbidden.

1.20

PullRequestMerger now notifies the taskListener of failures.
AutoCloseFailedPullRequest has been extracted from the published URL check.

1.19

More work for disabled builds.
Unified tabs to spaces.
Updates to the tests, and added some tests.

1.18

Add support for folder projects.
Correcting issue with default credentials.
Correcting issue with ghRepository being null when it shouldn't be.
Ignoring case when matching repo to url.
Changing the wording for pull requests that are mergeable.
Change requestForTesting phrase to a textArea.
Check if project is disabled, if it is then don't do anything.

1.14

A comment file can be created during the build and added to any comment made to the pull request. podarok#33
Added a [skip ci] setting, that can be changed. Adding the skip statement to the pull request body will cause the job not to run. sathiya-mit#29
Escaping single quotes in log statements tIGO#38
Fixed owner name deduction from url on github hook handling nikicat#40
Removed unused Test field from the config

1.13-1

Replacing deprecated Github.connect method. tIGO#39
Added a merge plugin for post build. If the build is successful, the job can specify conditions under which the pull request "button" will be pressed.

1.12

Fixes issue with checking rate limits for GitHub Enterprise
Internal refactorings
Fixes checking of TriggerTimer (JENKINS-22550)

1.11.2

Major reduction in the number of requests made to GitHub to avoid rate limit issues
Fixed changelog for a build so it reports differences between Pull Request build
Fixes bug on getting author email address from GitHub APIs
Updates warning message is URL isn't set on the GitHub project plugin
Adds some additional parameters to the build, such as the source branch, pull request link, etc.
Success/failure message now put into the comment on Pull Requests

1.10 - 1.11.1

Not publicly released.

1.9

Added job specific triggers.
Added more informations as job parameters.
Fix support for gh-ent token generation. JENKINS-17334
Lookup PR detailed information for each PR. JENKINS-17852
Fix closing PR.
various bugfixes

1.8

Support auto-close at build level
add pull id and target branch to environment variables
fix GitHub Hooks when authentication is required on Jenkins
various bug fixes and improvements

1.7

implemented support for GitHub hooks
added button for generating API token

1.6

option for specifying how to mark unstable builds in GitHub
Close failed pull requests automatically if plugin admin configured it
customizeable pass/fail comments
Refactoring of code
help description in Jenkins configuration

1.5

checkbox enabling sending comments when update of comment status fails
option to enter organisations which members are whitelisted
support for Enterprise GitHub
link to pull request into build description
fixes

1.4

Changed comment commands - phrases for testing once, pull request or add user to whitelist
"test this please" starts new build
"ok to test" allow pull request for building
"add to whitelist" add user to whitelist

1.3

Add published URL feature.
Enable connection to GitHub Enterprise instances.
Enable authentication via a token.
When automatic merge is possible, build the merge.

1.2

Use commit status API instead of comments for reporting results.