***Stash Web Hook for Jenkins to trigger specific GIT branch***

* A **webhook** in [web development](https://en.wikipedia.org/wiki/Web_development) is a method of augmenting or altering the behavior of a [web page](https://en.wikipedia.org/wiki/Web_page), or [web application](https://en.wikipedia.org/wiki/Web_application), with custom [callbacks](https://en.wikipedia.org/wiki/Callback_(computer_programming))
* Webhooks are "user-defined HTTP callbacks".[[2]](https://en.wikipedia.org/wiki/Webhook#cite_note-2) They are usually triggered by some event, such as pushing code to a repository[[3]](https://en.wikipedia.org/wiki/Webhook" \l "cite_note-3) or a comment being posted to a blog.[[4]](https://en.wikipedia.org/wiki/Webhook#cite_note-4) When that event occurs, the source site makes an HTTP request to the URI configured for the webhook. Users can configure them to cause events on one site to invoke behaviour on another. The action taken may be anything. Common uses are to trigger builds with [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration) systems[[5]](https://en.wikipedia.org/wiki/Webhook" \l "cite_note-5) or to notify [bug tracking systems](https://en.wikipedia.org/wiki/Bug_tracking_system).[[6]](https://en.wikipedia.org/wiki/Webhook#cite_note-6) Since they use HTTP, they can be integrated into web services without adding new infrastructure.[[7]](https://en.wikipedia.org/wiki/Webhook#cite_note-7)However there are also ways to build a [message queuing service](https://en.wikipedia.org/wiki/Message_queuing_service) on top of HTTP—some [RESTful](https://en.wikipedia.org/wiki/Representational_state_transfer) examples include[IronMQ](https://en.wikipedia.org/wiki/IronMQ) and RestMS.

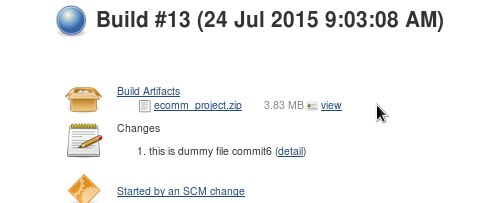
**Requirements**

* **Git Plugin** - Jenkins needs to have the [Git Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin) installed in Jenkins and the Poll SCM option must be enabled

**Setup**

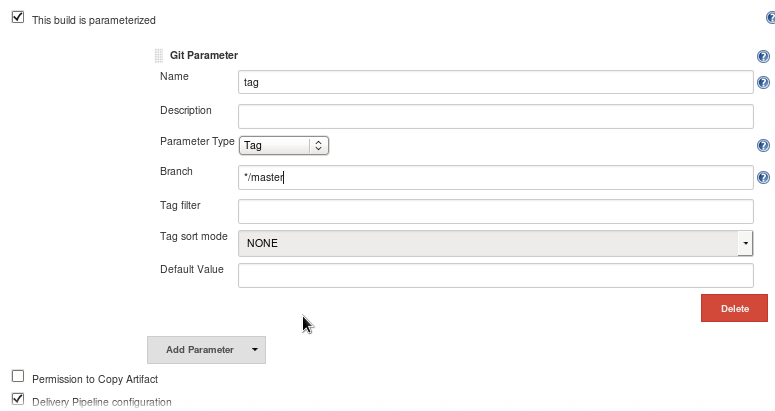
Once installed, follow these steps:

* Navigate to a repository in Stash.
* Hit the *Settings* link
* In the left-navigation, hit the *Hooks* link
* For the **Stash Webhook to Jenkins**, click the *Enable* button.
* Enter the URL to your Jenkins instance
* Select the method that clone method that Jenkins is using (HTTP or SSH).
* If using HTTP, enter the username that Jenkins is using to clone your repository.
* Submit the form.
* Commit some code and watch it trigger a build!



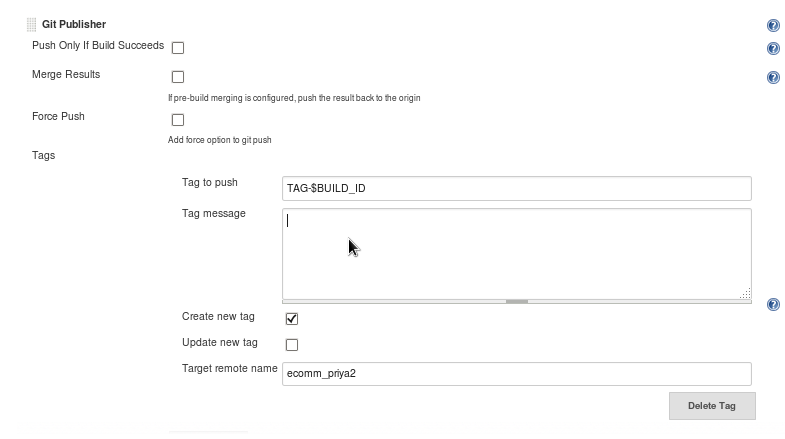
***Identify and implement tagging workflow for GIT and identify the job where we will tag the branch***

* Created git tags on my file
* Pushed those tags on repo
* Using git parameter plugin in Jenkins build the specified tag
* Mark parameterized build on Jenkins and add name as tag as well as branch(master)
* In scm block on branches to build section write “${tag}” and save and build.



***Method 2:***

* Using git publisher plugin



* Click “Add Tag” to begin the process of adding a tag.
* “Tag to Push”.  This setting represents the name of the TAG as it will end up on the GIT Repository.   In this example we are using a combination of a text literal “TAG-” and a hudson variable “$BUILD\_ID”.

“Create new Tag”.  If checked the plugin will attempt to create a new tag with the name from “Tag to Push”.  IMPORTANT.  If this option is checked and the specified tag name already exists — then the plugin will fail.  If the option is not checked and the specified tag name does not exist — then the plugin will fail.  In other words…

* If Checked: The tag name cannot yet exist
* If Not Checked: The tag name has to already exist.

“Target Remote Name”.  This is the reference name of the repository that you setup above at the start of the post.  Notice that we are using the alias “RepoName” that we setup earlier.

***Check if we can create pull request from Jenkins for Stash***.

## Stash Pull Request Builder Plugin:

This Jenkins plugin builds pull requests from a Atlassian Stash server and will report the test results as a comment. This plugin was inspired by the GitHub & BitBucket pull request builder plugins.

### Prerequisites

* Jenkins 1.532 or higher.
* Git Plugin - <https://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin>
* Credentials & Stash hostname are now pulled from Git SCM

### Creating a Job

* Create a new job
* Select Git SCM
* Add Repository URL
  + Choose credentials (will be used by trigger)
  + Set refspec to: +refs/pull-requests/\*:refs/remotes/origin/pr/\*
* In Branch Specifier, type as bellow
  + \*/${pullRequest}
* Under Build Triggers, check Stash Pull Request Builder
* In Cron, enter crontab for this job.
  + e.g. every minute: \* \* \* \* \*
* In Stash BasicAuth Username - Stash username like jenkins-buildbot
* In Stash BasicAuth Password - Jenkins Build Bot password
* Supply project code (this is the abbreviated project code, e.g. PRJ)
* Supply Repository Name (e.g. myRepo)
* Save to preserve your changes

### Merge the Pull Request's Source Branch into the Target Branch Before Building

You may want Jenkins to attempt to merge your PR before doing the build -- this way it will find conflicts for you automatically.

* Follow the steps above in "Creating a Job"
* In the "Source Code Management" > "Git" > "Additional Behaviors" section, click "Add" > "Merge Before Building"
* In "Name of Repository" put "origin" (or, if not using default name, use your remote repository's name. Note: unlike in the main part of the Git Repository config, you cannot leave this item blank for "default".)
* In "Branch to merge to" put the target branch you want to merge to "${targetBranch}"
* Note that as long as you don't push these changes to your remote repository, the merge only happens in your local repository.

By default it will only accept pull requests that match the target branch specified as target branch in the 'Merge before build' extension. You can choose to disable this or choose an custom filter to limit the number of pull requests being verified. Both options are available in the advanced tab.

If you are merging into your target branch, you might want Jenkins to do a new build of the Pull Request when the target branch changes.

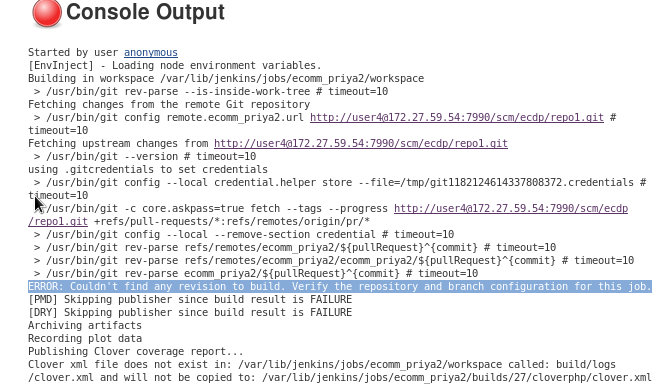
* There is a checkbox that says, "Rebuild if destination branch changes?" which enables this check.

### Notify Stash Instance

If you have enabled the 'Notify Stash Instance' Post-build Action and also enabled the 'Merge before build' extension you need to add '${pullRequestCommit}' as Commit SHA-1. Otherwise you'll notify Stash with the commit hash resulting from the merge which isn't known to Stash (since it's merged locally).

### Rerun test builds

If you want to rerun pull request test, write “test this please” comment to your pull request.



Note: Facing some issues with this task.