

Push based builds using Jenkins and GIT

Avisi werknemer

Published: 13 January, 2012

Recently we switched to Git as our Version Control System solution and I still like it. Coming from "(good) old" CVS we used Jenkins as CI server polling CVS for changes. Besides other nice features and advantages, Git also brings nice commit hooks (well, CVS also does but I finally got to use this nice feature).

Using the post-receive hook you can use your favorite scripting language to do some kick-ass checking when a git commit/push has been made. We first started to use a common approach for every repository and created the post-receive hook like this:

```
#!/bin/bash
```

```
/usr/bin/curl --user USERNAME:PASS -s \
```

```
http://jenkinsci/job/PROJECTNAME/build?token=1qaz2wsx
```

Configure your Jenkins job to be able to "Trigger builds remotely" and use an authentication token (1qaz2wsx in this example).

Now two issues appear:

1. We need to change the PROJECTNAME for every single repository
2. In case of more projects/modules in one repository we can't figure out what specific Jenkins job has to be run.

I solved both issues using the git diff-tree feature; after a commit/push the diff-tree option will show which specific folder has been modified:

```
rodmidde-lan:confluence rodmidde$ git diff-tree --name-status HEAD
```

c299697bd62d89158fd7deb3ebd7d788a215f38d

M digiArchief

The following script does the trick for me, it uses the diff-tree option and kicks the corresponding Jenkins job (assuming the Jenkins job is named after the changed git project/module:

```
changeSet=(`git diff-tree --name-status HEAD`)
for(( i=0; i<${#changeSet[@]}; i++))
do
    if [ ${changeSet[$i]} == "M" ]
    then
        projectToBuild=${changeSet[$i+1]}
        /usr/bin/curl --user USERNAME:PASS -s \
            http://jenkinsci/job/$projectToBuild/build?token=1qzz2wsx
    fi
done
```

Feel free to leave suggestions, improvement and use the script if you like. Questions always welcomed!