**Continuous Integration and Continuous Deployment**

**Problem Statement:**

**JOB#1**

If Developer push to dev branch then Jenkins will fetch from dev and deploy on dev-docker environment.

**JOB#2**

If Developer push to master branch then Jenkins will fetch from master and deploy on master-docker environment.

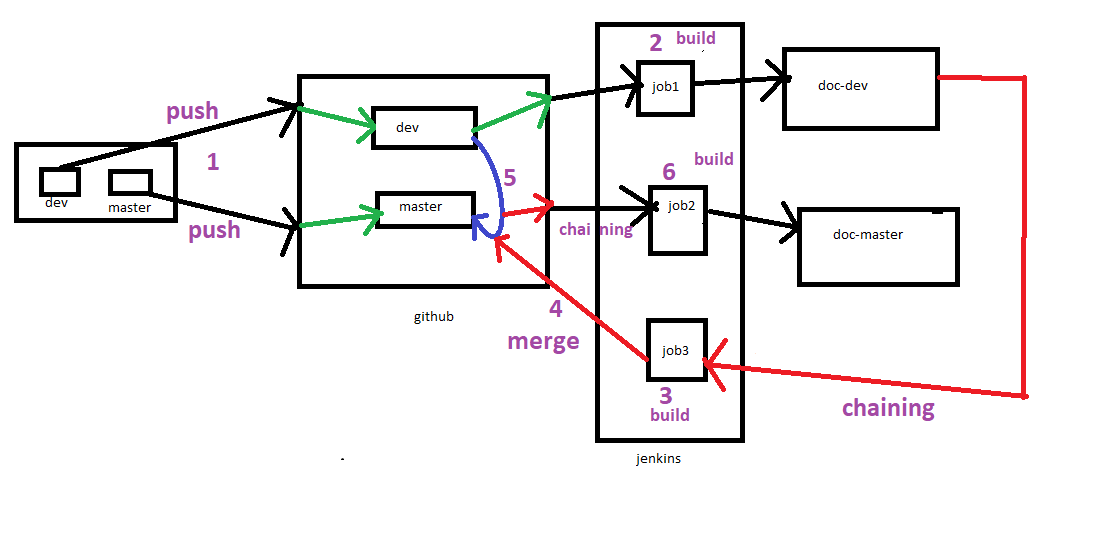
Note :-

both dev-docker and master-docker environment are on different docker containers.

**JOB#3**

Manually the QA team will check (test) for the website running in dev-docker environment. If it is running fine then Jenkins will merge the dev branch to master branch and trigger #job 2

**Flowchart :**



**Initial Steps:**

* Create a repo in local git with two branches in it
* Create post-commit hook for pushing the repo to github after commit automatically
* Commit/Push the repo to github
* Launch two containers in baseOS named as dev-doc and dev-master

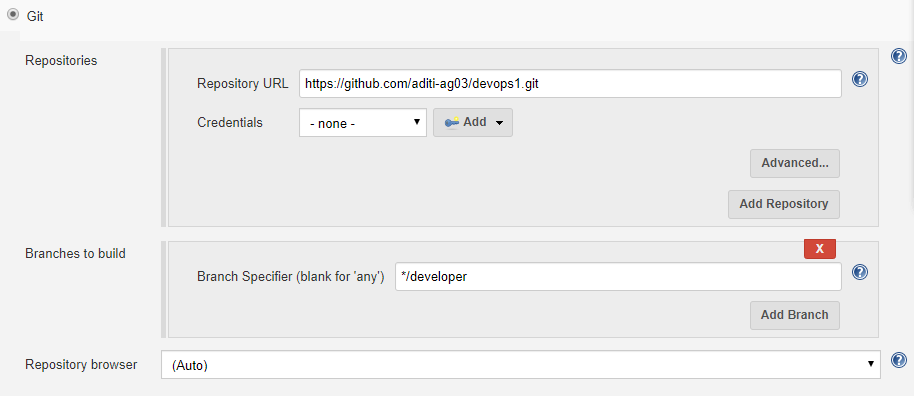
#docker run -dit --name doc-dev -v /webserverdb1:/var/www/html webserver:v1

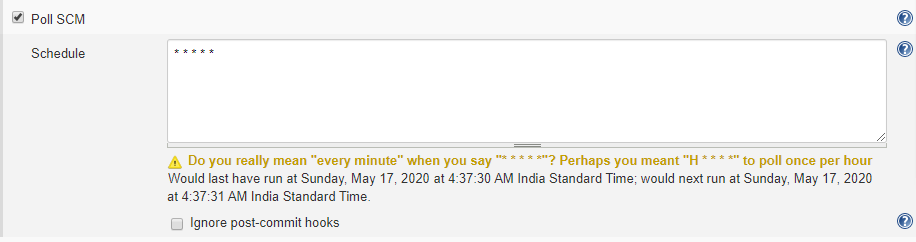
#docker run -dit --name doc-master -v /webserverdb2:/var/www/html webserver:v1

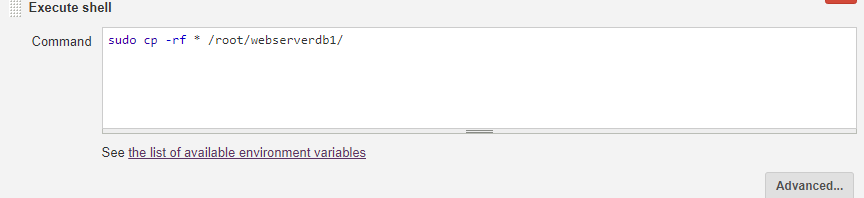
* Give all powers to jenkins user through sudoers file
* Ensure git should be installed

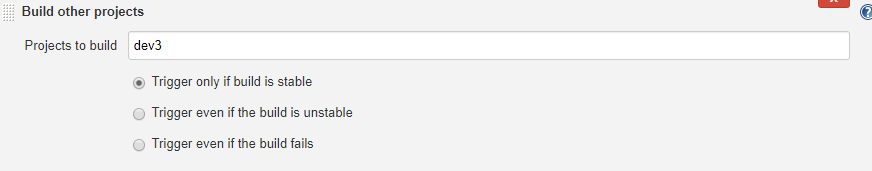
**Job 1:**

Clone the developer branch and deploy it in doc-dev container by copying all the files in a folder mounted to doc-dev container



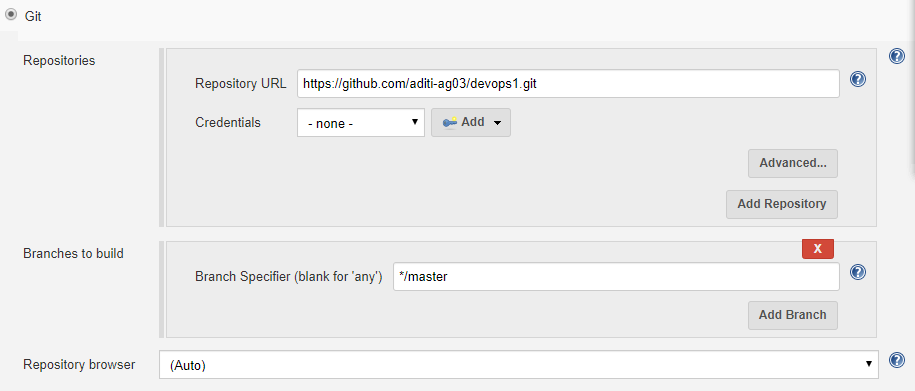






**Job 2:**

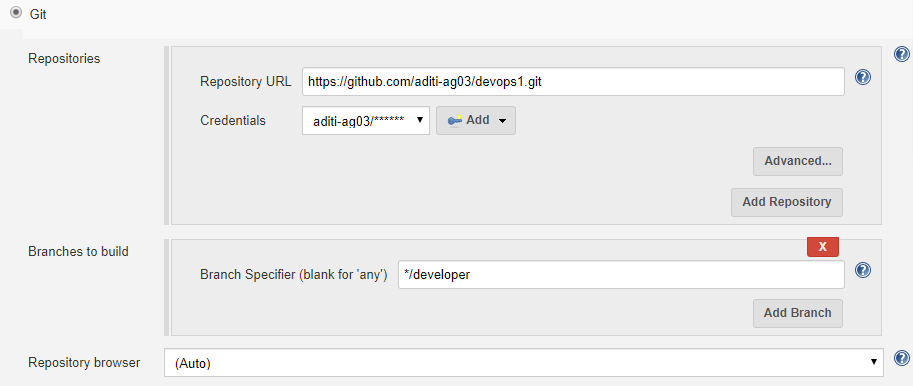
Clone the master branch and deploy it in doc-master container by copying all the files in a folder mounted to doc-master container, this job will only run when developer branch will be merged to master branch through job 3

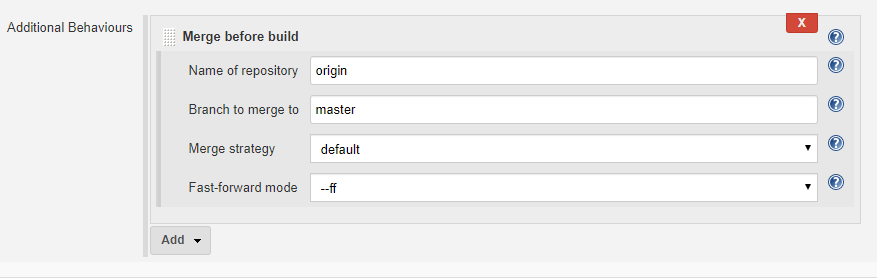


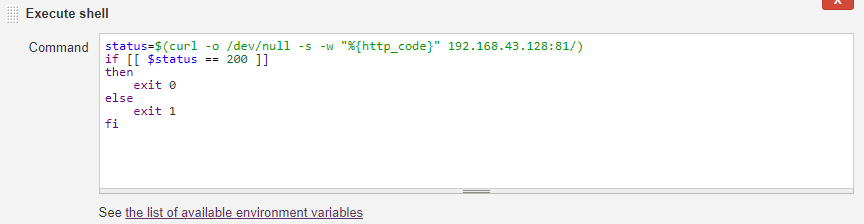


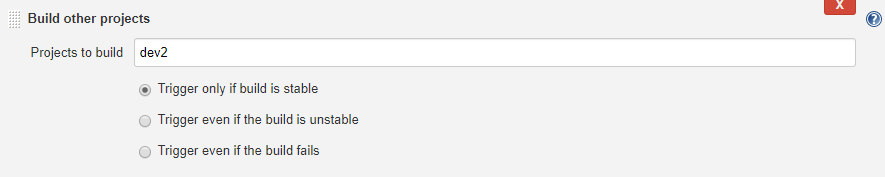
**Job 3:**

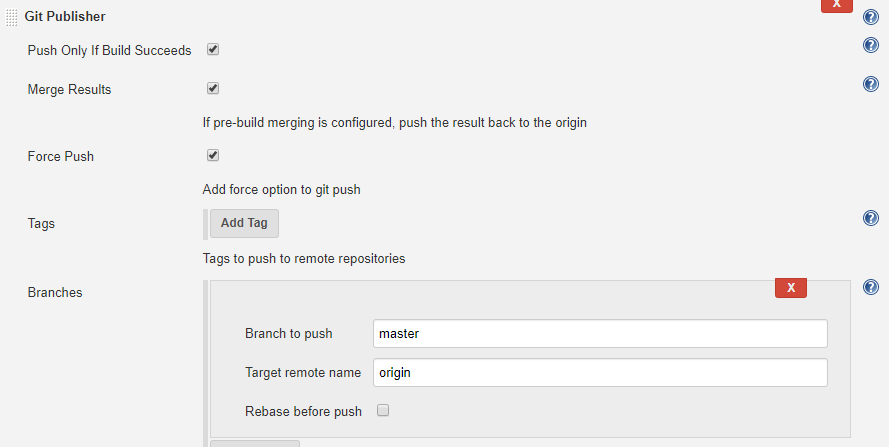
This job will check the code of developer branch is running fine or not, if it is running fine this job will merge developer branch into master branch and will push the changes to GitHub











**Build Pipeline:**

