# How to write and use JJB?

## What is Jenkins Job Builder

Jenkins Job Builder(JJB) takes simple descriptions of Jenkins jobs in YAML format, and uses them to configure Jenkins jobs. JJB keeps your job descriptions in human readable format and job template system simplifies the configuration of Jenkins jobs. Upstream documentation is available at http://ci.openstack.org/jenkins-job-builder/.

## How to Write/Use Jenkins Job Builder

Job template is widely used in JJBs and makes the configuration of Jenkins jobs simple, if you need to define several jobs which are nearly identical, except perhaps in their names, SCP targets, etc., then you may use a job template to specify the particulars of the job, and then use a project to realize the job with appropriate variable substitution.

To illustrate how to configure Jenkins jobs by using job template, we can start with a simple example used in our OPNFV releng project octopus directory, which is just used to print "Hello world from Octopus", shown as below:

```
-job-template:
   name: octopus-test

   node: master

   project-type: freestyle

   logrotate:
       daysToKeep: 30
       numToKeep: 10
       artifactDaysToKeep: -1
       artifactNumToKeep: -1

   builders:
       - shell: |
           echo "Hello world from octopus"
```

the value "-1" here means keep forever, you can add this job template into the project jobs to run in jenkins:

```
- project:
    name: octopus
    jobs:
        - 'octopus-test'
```

then this job works!

Further, if you are a developer who wants to set up a job template to be used in jenkins, we should dive into much more, taking one job template in releng project used for octopus project, the 'octopus-daily-{stream}', as an example:

```
-job-template:
   name: 'octopus-daily-{stream}'

   node: master
```

```
    # Job template for daily builders
    #
    # Required Variables:
    #     stream:    branch with - in place of / (eg. stable)
    #     branch:    branch (eg. stable)

    project-type: freestyle
    varsetabove: '{somevar}'

    logrotate:
        daysToKeep: '{build-days-to-keep}'
        numToKeep: '{build-num-to-keep}'
        artifactDaysToKeep: '{build-artifact-days-to-keep}'
        artifactNumToKeep: '{build-artifact-num-to-keep}'

    parameters:
        - project-parameter:
            project: '{project}'

    scm:
        - git-scm:
            credentials-id: '{ssh-credentials}'
            refspec: ''
            branch: '{branch}'

    wrappers:
        - ssh-agent-credentials:
            user: '{ssh-credentials}'

    triggers:
        - timed: 'H H * * *'

    prebuilders:
        - test-macro

    builders:
        - shell:
            !include-raw build-upload-docu.sh

    postbulders:
        - test-macro
```

the {stream} here means when you add this into jobs, you can replace {stream} with what you want, such as:

```
stream:
      - master:
          branch: 'master'
```

the:

```
node: master
```

means to restrict this job to run in Jenkins master node. Next, several important procedures are illustrated here,

- scm, this mudule allows you to specify the source code location for the project, and it allows referencing multiple repositories in a Jenkins job.
- triggers, this module defines what causes a Jenkins job to start building.
- prebuilders and postbuilders, which define job need done pre and post the builders.
- builders, which defines actions that the Jenkins job should execute, usually the shell scripts or maven targets are existed there, e.g., build-upload-docu.sh used in our example.

Generally, the modules used in a job template is sequenced as

1. parameters, properties
2. scm
3. triggers
4. wrappers
5. prebuilders
6. builders
7. postbuilders
8. publishers, reporters, notifications

# Working with OPNFV Jenkins Jobs

By now, the releng project of OPNFV is the release engineering project for JJBs, you can clone the repo:

```
git clone ssh://YOU@gerrit.opnfv.org:29418/releng
```

make changes:

```
git commit -sv
git review
remote: Resolving deltas: 100% (3/3)
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: New Changes:
remote:   https://gerrit.opnfv.org/gerrit/51
remote:
To ssh://agardner@gerrit.opnfv.org:29418/releng.git
 "* [new branch]      HEAD -> refs/publish/master
```

Follow the link to gerrit https://gerrit.opnfv.org/gerrit/51 in a few moments the verify job will have completed and you will see Verified +1 jenkins-ci in the gerrit ui.

If the changes pass the verify job https://build.opnfv.org/ci/view/builder/job/builder-verify-jjb/ The patch can be submitted by a committer.

The verify and merge jobs are retriggerable in Gerrit by simply leaving a comment with one of the keywords listed below. This is useful in case you need to re-run one of those jobs in case if build issues or something changed with the environment.

- Verify Job: Trigger: **recheck** or **reverify**
- Merge Job: Trigger: **remerge**

You can add below persons as reviewers to your patch in order to get it reviewed and submitted.

- Ulrich Kleber (Ulrich.Kleber@huawei.com)

- Fatih Degirmenci (fatih.degirmenci@ericsson.com)

- Xinyu Zhao(Jerry) (zhaoxinyu@huawei.com)

- Aric Gardner (agardner@linuxfoundation.org)

The Current merge and verify jobs for jenkins job builder for releng project, shown in https://git.opnfv.org/cgit/releng/tree/jjb.

Assuming that you have set up some job templates and put them into a project, then the question is that how they work? Taking the jobs 'builder-verify-jjb', 'builder-merge' used in releng project as examples, 'builder-verify-jjb' is to verify jobs you commited, you will see verified '+1' jenkins-ci in gerrit if it succeed, 'builder-merge' is to set up a merge job and update all the JJBs. If you have some new jobs need to be run, you can set up your own job templates and add them into the project.

**Documentation tracking**

Revision: f93f9ba19fd8a7127274614a46965cbe8ed6fa21

Build date: Wed Nov 18 06:09:21 UTC 2015