

2019 Self Release process

Overview and Support

- Self serve release is now possible in global-jjb
 Support for Java artifacts releases (Nexus)
 Support for container releases (Nexus3, DockerHub)
- Global-jjb updates are a must to ensure the latest fixes/improvements.

https://github.com/lfit/releng-global-jjb → Github Global-jjb

https://docs.releng.linuxfoundation.org/projects/global-jjb/en/latest/jjb/lf-release-jobs.html → Documentation

support.linuxfoundation.org → For further assistance



Before using self serve releases ...

Teams must make sure:

- Committer rights are up to date (INFO.yaml)
- Global-jjb has been updated to the latest version
- Understand the DO'S and DON'TS of this training
- Release verify and merge jobs are added for your repo
- Ask any questions



After your release is completed ...

DONT's:

- Do not revert files in releases/ folder
- Do not modify releases files
- Do not cherry-pick releases files between master and release branches
- Do not Re-release same versions
- Do not Re-use same stage job (Jar artifacts)
- Do not combine several releases in one Gerrit
- Do not overwrite Jenkins failures. Report failures to support team
- Running re-merge on merged changes will cause job failures

SELF SERVE RELEASES

Process steps

Releasing a JAR artifact to Nexus ...

Releasing a JAR artifact to Nexus ...

- 1- Build a release candidate from their code Produce a staging repo (autorelease)
- 2- Create the release file. (<repo>/releases/<name>.yaml)

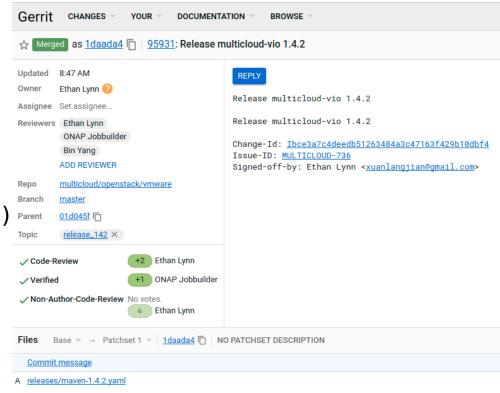
distribution_type: 'maven'

version: (Semantic version)

project: (Information on the project name)

log dir: (link to Jenkins job with build number)

- 3- Push the change to Gerrit and let verify run Triggers supported: recheck|reverify
- 4- Get the change +2/merged by a committer



```
File

1 distribution_type: 'maven'
2 version: '1.4.2'
3 project: 'multicloud-openstack-vmware'
4 log_dir: 'multicloud-openstack-vmware-maven-stage-master/146/'
5
```

SELF SERVE RELEASES

Process steps

Releasing an image to Nexus3 ...

Releasing an image to Nexus3 ...

1- Build a release candidate from their code Produce container images in Nexus3

2- Create the release file. (<repo>/releases/<name>.yaml)

distribution type: 'container'

version: (Semantic version)

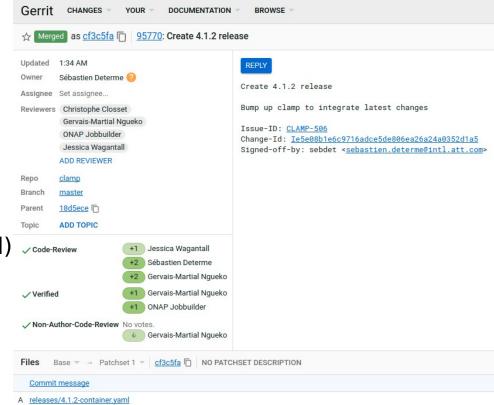
project: (Information on the project name)

log_dir: (link to Jenkins job with build number)

ref: (SHA1 to be tagged)

containers: (list of container and versions to pull)

- 3- Push the change to Gerrit and let verify run Triggers supported: recheck|reverify
- 4- Get the change +2/merged by a committer



```
File
   1 distribution_type: 'container
  2 container_release_tag: '4.1.2'
  3 project: 'clamp'
  4 log_dir: 'clamp-maven-docker-stage-master/121/
   5 ref: 9fb87912f025e85b26c238d286d87e0e1c9c767a
  6 containers:
         - name: 'clamp-backend
           version: '4.1-STAGING-20190916T192828Z'
         - name: 'clamp-frontend'
  10
          version: '4.1-STAGING-20190916T192828Z'
 11
         - name: 'clamp-dashboard-logstash'
          version: '4.1-STAGING-20190916T192828Z
 12
 13
         - name: 'clamp-dashboard-kibana'
  14
          version: '4.1-STAGING-20190916T192828Z'
 15
```

SELF SERVE RELEASES

QUESTIONS?

support.linuxfoundation.org

