

Proposal to Move to SCM/CI As A Service

DRAFT FOR USE BY ONAP INFRASTRUCTURE WORKING GROUP

Alexis, Brian Christophe, Morgan

Background

- a. Long term proposal from the LFN TAC is to replace Gerrit and Jenkins with a more modern tool chain and use an "As A Service" approach rather than dedicated instances.
- b. General plan is to move to a large cloud hosted solution like GitLab/GitLab-CI
- c. We believe reliability will be better
- d. We calculate that costs will be lower
- e. We believe that more modern SCM will reduce the learning curve for new developers

Approach

- a. Prototype with one or two projects to work out the kinks in Frankfurt
- b. Plan to move at the start of G release if approved by TSC in time.
- c. No project moves unless we are ready

FAQ

1. Common Login
 - a. Common Login will be based on github accounts not Linux foundation accounts
 - b. Based on CLA requirements and general biterg.io tracking , contributors should use a corporate email based github account
2. Contributor License Agreements
 - a. EasyCLA supports gerrit and github right now.
 - b. gitlab is on the roadmap but not available yet.
 - c. Support ticket SUPPORT-690 created <https://jira.linuxfoundation.org/servicedesk/customer/portal/4/SUPPORT-690>
 - d. Other notes:
 - i. Contributors should use corporate email based github accounts not personal email based github accounts
3. Biterg.io
 - a. Will need to add gitlab repositories to onap.biterg.io
 - b. May need to simply have a separate biterg.io setup so we don't double count after initial upload from gerrit to gitlab
 - c. biterg probably has a recommended procedure for this type of migration
4. SCM Reviews
 - a. pull request and branches vs gerrit branches
 - i. high level change in flow with git pull vs gerrit
 - b. As a Contributor I will use a feature branch instead of working on master and submitting a gerrit review.
 - A developer makes a change in their feature branch and tests it. When they're happy they push, and make a merge request.
 - The developer assigns the merge request to a reviewer, who looks at it and makes line and design level comments as appropriate. When the reviewer is finished, they assign it back to the author.
 - The author addresses the comments. This stage can go around for a while, but once both are happy, one assigns to a final reviewer who can merge.
 - The final reviewer follows the same process again. The author again addresses any comments, either by changing the code or by responding with their own comments.
 - Once the final reviewer is happy and the build is green, they will merge.
 - c. As a PTL I will see pull requests
 - d. GUI Differences
 - i. Cherry Pick
 - ii. Abandon
 - iii. Create Branch
 - iv. Create Tag
 - v. Magic Words
5. CI Jobs
 - a. Instead of jjb on jenkins.onap.org my jobs will be
 - b. Magic Words
 - c. Seeing the job queue
 - d. Restarting a job
 - e. Seeing build errors
 - f. Seeing build status
6. Built-in Docker registry
 - a. shall we sync with Nexus/Docker hub assuming that a built-in registry is available
 - b. a security portal scanning docker vulnerabilities is also possible

7. Built-in Artifacts management
8. Built-in pages (like in github) to host static page (already used to host the pages to monitor the CI chains https://orange-opensource.gitlab.io/lfn/ci_cd/chained-ci/)
9. Enforcement of community by laws and practices
 - a. How will we publish release artifacts (nexus) and release docker containers (nexus3/dockerhub) ?
 - i. Will there be any change ?
 - b. INFO.yaml is used to track and automatically populate LDAP with PTL's and Committers , what will be the new process with gitlab as a service ?
10. [Support.linuxfoundation.org](https://support.linuxfoundation.org) and AsAService LF supported applications
 - a. How to get help through LF for SCM/CI in As A Service
 - b. Escalations

References

1. <https://about.gitlab.com/devops-tools/gerrit-vs-gitlab.html>