

CI Process

Basic CI Services

- All CI services are run and maintained by the Linux Foundation
- The basic layout and operation of services is similar to other Linux Foundation-run sites (e.g. onap.org, acumos.org, etc.)
- The following services are provided:
 - **Gerrit**: for source code (gerrit.akraino.org)
 - **Jenkins**: for code builds and unit tests (jenkins.akraino.org)
 - **Nexus**: for storing artifacts (nexus.akraino.org) and Docker containers (nexus3.akraino.org)
 - **SonarQube**: for automated code inspection (sonar.akraino.org)
 - **Wiki**: for documentation (wiki.akraino.org)
 - **Jira**: for issues (jira.akraino.org)
- All services use HTTPS; in addition, Gerrit also allows ssh

A Linux Foundation ID is required for all services (identity.linuxfoundation.org)

New Projects

- The PTL reaches out to the CI/Blueprint Validation Lab sub-committee to create project under Gerrit and in other Linux Foundation infrastructure. This includes VMs or community lab share they may need.
- The CI/Validation sub-committee approves the request and assists the PTL to access the resources.
- PTL coordinates with the project team to start the development.
- Projects have a list of committers who can put code into the projects; this needs to be given to LF to assign to your LF account.
- We recommend projects be named as such: `<blueprint-name>_<subproject-name>`, so that different blueprints can be easily discerned in Gerrit.
- For new Projects: the CI/Blueprint Validation Lab sub-committee sends an email to helpdesk@akraino.org with name of project (lower case) and list of committers.
- Documentation in wiki generally refers to the latest master branch code (may be overridden by documentation subcommittee)

CI Process - Gerrit

- Code is submitted to Gerrit from the developer, using the `git review` command.
- Code is reviewed and voted on via the normal Gerrit code review process
- Reviewers need to have permissions (assigned by LF) to merge code
- LF uses a voting range of -2..+2; +2 code is eligible to merge
- LF guidelines on Gerrit use are here: <https://docs.releg.linuxfoundation.org/en/latest/gerrit.html>
- The code Review process is described here: <https://lf-releg-docs.readthedocs.io/en/latest/environment-overview.html>

CI Process - Jenkins

- Jenkins jobs are defined by Jenkins Job Builder (JJB) YAML files. Jobs are written as JJB templates, on a per-project basis.
- Jenkins jobs are primarily defined by Linux Foundation JJB code; although this is extensible.
- All Jenkins jobs are maintained in the ci-management project (<https://gerrit.akraino.org/r/#/admin/projects/ci-management>)
- LF personnel code review/approve all changes to the ci-management project
- More details are available here:
 - <https://docs.releg.linuxfoundation.org/en/latest/>
 - <https://github.com/lfci/releg-global-jjb>
 - <https://github.com/lfci/releg-common-packer.git>
- The Jenkins Sandbox (<https://jenkins.akraino.org/sandbox/>) is available to do limited testing of jobs (without Nexus). This does not require LF approval, so it is a way to test out new JJB templates.

CI Process - Nexus

- All built artifacts are stored in Nexus
 - nexus.akraino.org - .tar, .war, .jar and other files
 - nexus3.akraino.org - Docker containers
- Individual CD processes retrieve from nexus as the result of Gerrit API events
- nexus3 can be used to retrieve Docker containers (it supports Docker v2 API). There are four individual repositories here:

Repository	Purpose
nexus3.akraino.org : 10001	a read-only repository metagroup that is comprised of the local <code>docker.release</code> and docker.io repositories.
nexus3.akraino.org : 10002	<code>docker.release</code> repo which is where releases should land after they have been blessed. We replicate from this out to the Docker hub organization.

nexus3.akraino.org: 10003	the "normal" repo where snapshots would go.
nexus3.akraino.org: 10004	The docker staging repo. Stage builds go here.

CI Process - Docker multiarch support



Akraino Multiarch Support.pptx