

Infrastructure Working Group Summary Report

Introduction

The LFN Technical Community is critically dependent on its software development tool chain for success. The standard tool chain today is Gerrit for Source Code Management (SCM) , Jenkins for Continuous Integration / Continuous Deliver (CI/CD) and Nexus for artifact repositories.

Some of these tools have been supplanted in popularity the larger open source development communities with more modern tools .

for example, most developers are now more familiar with GitHub/GitLab than gerrit and there is a growing need for CI/CD tooling that is readily compatible and supported with these tools.

At the same time as community and corporate developers are asking for GitHub/GitLab/Bitbucket compatibility there is a need to re-evaluate the cost structure for per project dedicated resources for the tool chain and whether moving to an "As A Service" model would be overall better for the LFN communities.

CNCF projects predominately utilize an "As a Service" approach where each project selects and utilizes Github, various CI/other services that meet its needs and the CNCF handles payment and negotiation of discounts and donations from the service providers. This has resulted in superior community experience at much lower costs and empowers the community to do what is best for each project. CNCF also actively surveys its maintainer community a few times a year to gauge any satisfaction issues and community needs to stay on top of things

The LFN TAC formed a working group to investigate the software development tooling and answer two questions:

1. Should the TAC recommend to member projects that they move off Gerrit/Jenkins/Nexus and onto a new tool chain.
2. Should the TAC recommend to member projects that they pursue with LF a "As A Service" mode of operation for the recommended tool chain.

This report summarizes the efforts from the LFN TAC and its member and the two recommendations requested of the working group by the TAC.

Criteria

The primary criteria to be used for recommending a tool will be the ease of use and ability for the community developers to get their job done. Changing tooling will take effort and impact resources so it should not be done lightly. A new tool chain must be better for today's community, stable and supported with a community of its own developers that will meet the ongoing needs of the LFN developers.

- Easy for new and existing Developers to use
- Compatible with predominate tool chains that developers are using in their corporate or community development outside of LFN
- Supported
- Up to date with the latest technologies (e.g. Docker container friendly , modern build system support etc)

The recommendation on "As A Service" should be based on an analysis of the costs to LFN and would be dependent on assumed pricing from the suppliers. A recommendation should include input from LF so that business negotiations on any final pricing and support would be done through LF.

Information/Experience Gathering

a. Prototyping

Team members using their project repositories obtained accounts on the various as a service tool chains and actually used them to build the code and generate artifacts.

Many tools were evaluated but not all tools work with each other equally well. The table below shows the tools and their respective role that were evaluated (demonstration and/or prototype)

SCM	CI/CD Pipeline	Repository
GitHub	CircleCI	DockerHub
GitLab	GitLab-CI	
BitBucket	Travic-CI	
	Zuul (Demo)	
	OPNFV CI	

	Drone/Drone.io	
--	----------------	--

POC infra results

SCM Support by CI Platform

CI SaaS Comparison : <https://wiki.lfnetworking.org/download/attachments/10552057/CI%20SaaS%20Comparison.pptx?version=1&modificationDate=1556123409612&api=v2>

CI SaaS Comparison - SCM Support

SaaS	Github	Gitlab	Bitbucket	VSTS*	Gerrit
CircleCI	✓		✓		
Gitlab-CI	✓	✓	✓		
Azure Pipelines	✓	✓	✓	✓	
Bitbucket Pipelines			✓		
Drone.io (Cloud)	✓	✓	✓		
Appveyor	✓	✓	✓	✓	

THE LINUX FOUNDATION



b. Utilization/ Minutes of Use

Build minutes for the larger projects were gathered from the existing infrastructure in order to understand rough order of magnitude costs for "As A Service"

Back of the envelope estimates per project based on list price from CircleCI :

- ODL: 3,078,224.58m x \$0.024/m - \$73,877.84/quarter
- ONAP: 790,041.78m x \$0.024/m - \$18,960/quarter
- OPNFV: 3,300,000m x \$0.024/m - \$79,200/quarter
- FD.io - 3,892,000m/year

Costs for tooling software updates and day to management are included in these cost estimates.

While these costs do not factor in all the costs that would be incurred they indicate that there is potential saving by moving to an As A Service instead of the dedicated per project resources (3rd party hosting) for at least the projects that do not need bare metal. Projects like ODL, ONAP etc are above the hypervisor layer so they need VM's and K8 while others like OPNFV need bare metal do actually do their testing of an installation.

c. Hardware Hosting

Hardware Hosting technologies have also modernized in recent year as bare metal public cloud has been emerging. While it is anticipated that there will always be **some magic** hardware, there are new opportunities that may dramatically reduce the amount/costs while improving the reliability of services.

Summarized Results

SCM

[GitHub](#) and [GitLab](#) are the dominant tools for SCM that satisfy the requirement of easy to use and popular. There are many choices but either of these would be good. [Bitbucket](#) is also very popular but is more commercially oriented. GitLab has a slight edge since it can also be run in corporate environment behind the firewall and meet corporate community developer requirements with a consistent git history.

CI/CD

[CircleCI](#) for GitHub and [GitLab-CI](#) are the dominant tools that are most compatible with their respective SCM systems. There are more options in this space than SCM and it personal project bias might influence a project to use either CircleCI or GitLab-CI

Artifact Repository

[Docker Hub](#) was the preferred docker repository hands down. Many projects are already moving to Docker Hub for the released containers.

[Packagecloud.io](#) provides apt/yum/maven repos as a service. FD.io is in the process of switching to it for apt/yum, and has had very happy experiences with it.

Recommendation

Any recommendation represents a trade off in capability that is highly dependent on a community's skill level and risk tolerance vs being on the cutting edge.

This recommendation is provided to the Technology Advisory Council for communication to the projects within Linux Foundation Networking. No specific implementation plans are included, just recommendations.

The default preferred path for new generic software projects that do not have a dependency on non-standard hardware resources and/or gerrit for legacy code is to use a pull model based SCM.

For all hosted solutions we recommend:

Github -> CircleCI -> Packagecloud

Some projects may need premises versions for compatibility with corporate build environments

Gitlab -> Gitlab-CI -> Packagecloud

For legacy projects and those who need Gerrit for legacy code (C/C++) then we recommend:

Gerrit -> Jenkins -> Nexus

If non-standard hardware resources are required in addition:

Github -> CircleCI (for normal jobs) -> [Packet](#) (several possible launching methods including terraform) -> Packagecloud

If even more specialized hardware is needed:

Github -> CircleCI -> Packagecloud
-> [drone.io](#) server launching on [nomadproject.io](#) cluster ^

The above tool chains have been used and work.

As projects move onto a new tool chain, LF business agreements are established with them and LF and the community develops more expertise with a preferred workflow then a specific path will be more clear.

Should the TAC recommend to member projects that they move off Gerrit/Jenkins/Nexus and onto a new tool chain ?

1. For new LFN Projects we would recommend "As a Service" infrastructure whether Github ecosystem + CI as a Service (Circle CI/Drone.io/Azure Pipelines/Host [Tekton](#)) or all in one GitLab/GitLab-CI as the going forward tool chain for SCM, CI/CD and Docker Hub for artifacts.
2. For existing LFN Projects we would recommend moving to "As a Service" infrastructure whether Github ecosystem + CI as a Service (Circle CI/[Drone.io](#)/Azure Pipelines/Host Tekton) or all in one GitLab/GitLab-CI if they are considering a new tool chain or to take advantage of the "As A Service" options.
3. For **magic hardware** it is anticipated that "As a Service" CI can adequately drive activities on that hardware. New projects should plan to use "As a Service" CI for managing any special hardware. Existing projects may chose to use As a Service if it fits their process/needs.

Existing projects could choose to move or not.

Should the TAC recommend to member projects that they pursue with LF a "As A Service" mode of operation for the recommended tool chain ?

The working group believes that moving to a "As A Service" model will result in better costs structure. LF should start a project to get a business case together and analyze what cost savings could be achieved by moving to a GitLab/GitLab-CI as a service model.

For Discussion

1. Community manageable and controllable.

Contributors

REVIEW FOR ACCURACY

- [Jim Baker](#)
- [Trevor Bramwell](#)
- [Casey Cain](#)
- [Wenjing Chu](#)
- [Christophe Closset](#)
- [Brian Freeman](#)
- [Thanh Ha \(zxiiro\)](#)
- [Ed Kern](#)
- [Jack Morgan](#)
- [Mohammed Naser](#)
- [Kenny Paul](#)
- [Vratko Polak](#)
- [Morgan Richomme](#)
- [user-20f1f](#)
- [Edward Vielmetti](#)
- [Ed Warnicke](#)
- [Brian Wong](#)