CI-as-a-Service Platform Research

This page details and compares available CI-as-a-Service (CISaaS) platforms for open source communities. Though there is no free SaaS provider for Jenkins, it is included in the comparison as that is what most LFN projects are currently using.

Notes on Systems Not Included

There are several other CI SaaS platforms out there not included in this comparison for various reasons.

- **Travis-CI**
  After the majority of senior staff was laid off when the company was acquired, Travis-CI was essentially put into standby mode. No new major feature or changes are expected.
- **Bitbucket Pipelines**
  Bitbucket Pipelines requires code be hosted in Bitbucket and only provides 50 free minutes per-month for open source projects. This is far to few to even evaluate the system.
- **Appveyor**
  Originally the only place to get free Windows builds. There's little need to use them when Azure Pipelines provides all-if-not-more of the same services.

SCM Support by Platform

This table details the support for CI systems to report back the status of builds to SCM providers.

	Gerrit	Github	GitLab	Bitbucket	Azure Code
Jenkins	✅	✅	✅	✅	✅
CircleCI		✅		✅	
Azure Pipelines		✅		✅	✅
Drone.io		✅			
Github Actions		✅			
GitLab-CI		✅	✅	✅†	

† https://docs.gitlab.com/ee/ci/ci_cd_for_external_repos/bitbucket_integration.html
Note: a similar approach could be used for other SCMs provided there is an API for updating CI status

Build Resources and Minutes

Builds, agents, runners: these are all different names providers use to refer to the compute machine CI/CD builds run on.

	Builder Size	Max Job Time	ARM Builders	External Builders	Platform Support
Jenkins	N/A	N/A	N/A	Yes	All
CircleCI	2 vCPUs, 4-8 Gb RAM, larger sizes	5 Hours		Yes/No	All
Azure Pipelines	2 vCPUs, 7 Gb RAM	6 Hours		Yes	All
Drone.io	N/A	1 Hour	Yes	Yes/No	Linux
Github Actions	2 vCPUs, 7 Gb RAM	6 Hours		Yes	All
GitLab-CI	1 vCPU, 3.75 Gb RAM	3 Hours		Yes	Linux, Windows

Platform Support 'All' means support for Linux, OSX, Windows.
External Builders 'Yes/No' means external hardware is supported in the self-hosted (non SaaS) version.

Plans and Pricing Models

Details on 'free for open source' plans and pricing models for communities who may exceed those limits.

	Minutes (per Month)	Concurrent Builds (per Organization)	Pricing Model

Jenkins	N/A	N/A	N/A
CircleCI	1000	4	Containers + Features
Azure Pipelines	Unlimited	10	Concurrency
Drone.io	N/A	N/A	None
Github Actions	Unlimited	20	Users
Gitlab-CI	50,000*	Unlimited	Users

Gitlab-CI: Open source projects need to apply to the "Gold" level to bump the minute cap.

Platform Feature Comparison

This is not an exhaustive list of features across all the projects but a subset that maybe relevant to LFN projects.

	Container Registry	Package Hosting	Local Builds	Bring Your Own Builder	Test Result Reports	Dependency Caching	Matrix Builds	Cross Project Pipelines	SSH Debug Support
Jenkins				✓	✓		✓	✓	
CircleCI			✓			✓	✓		✓
Azure Pipeline	ℹ	✓		✓	✓	✓	✓	ℹ	
Drone.io			✓			✓	✓		
Github Actions	✓	✓		✓		✓	✓	ℹ	
Gitlab-CI	✓	✓	✓	✓	✓	✓	ℹ	✓	ℹ

Azure Pipelines: Container Register is not free; Cross Project Pipelines provide no indication in the UI
Github Actions: Cross Project Pipelines provide no indication in the UI
Gitlab-CI: Matrix builds are created with dynamic child pipelines; SSH Debug Support only on Kubernetes clusters.

Feature Descriptions

Container Registry

Registry hosted by the provider to store container build images

Package Hosting

Support for top package formats (Ex: Maven, NPM, Helm)

Bring-Your-Own-Builder

External machines (VMs, hardware, containers) can be connected to the platform

Test Result Reporting

Insights and statistics for test results

Local Builds

Reduced CI development time by supporting verifying CI configuration and builds locally

Dependency Caching

Support for caching packages or files between builds (Ex: Maven, PyPI, NPM)

Matrix Builds

Reduces job configuration needed to build the combination of multiple versions across multiple platforms

Cross Project Pipelines

Dependent builds can be easily traced across projects (not just within a project)

SSH Debugging

Pausing the execution of a build to login and debug build failures