

Puppet Enterprise vs GitLab

GitLab compared to other tools

Puppet is a configuration management tool and language that enables deployment and maintenance of state for large scale infrastructure. Puppet excels at managing legacy infrastructure like physical servers and VMs. Puppet was designed before widespread container adoption and does not implement Kubernetes natively.

GitLab is a single application for the whole DevOps lifecycle that includes not only configuration management, but also capabilities for project management, source code management, CI/CD, and monitoring. GitLab is designed for Kubernetes and cloud native applications.

GitLab can be used together with Puppet to enable VM and bare metal configuration management. For Cloud Native applications run on Kubernetes, Puppet is not required and GitLab comes with all the functionality needed built-in.

FEATURES



Built-in CI/CD

GitLab has built-in Continuous Integration/Continuous Delivery, for free, no need to install it separately. Use it to build, test, and deploy your website (GitLab Pages) or webapp. The job results are displayed on merge requests for easy access.

[Learn more about CI/CD](#)



AD / LDAP integration

Sync groups, manage SSH-keys, manage permissions, authentication and more. You can manage an entire GitLab instance through the LDAP / AD integration.

[More information about AD / LDAP integration](#)



Granular user roles and flexible permissions

Manage access and permissions with five different user roles and settings for external users. Set permissions according to people's role, rather than either read or write access to a repository. Don't share the source code with people that only need access to the issue tracker.

[Learn more about User Roles](#)



Cloud Native

GitLab and its CI/CD is Cloud Native, purpose built for the cloud model. GitLab can be easily deployed on Kubernetes and used to deploy your application to Kubernetes with support out of the box.

[Kubernetes integration](#)



Auto DevOps

Auto DevOps brings DevOps best practices to your project by automatically configuring software development lifecycles by default. It automatically detects, builds, tests, deploys, and monitors applications.



[Read more about Auto DevOps in the documentation](#)

Deploy Boards

GitLab Enterprise Edition Premium ships with Deploy Boards offering a consolidated view of the current health and status of each CI environment running on Kubernetes. The status of each pod of your latest deployment is displayed seamlessly within GitLab without the need to access Kubernetes.



[Learn more about Deploy Boards](#)

Canary Deployments

GitLab Enterprise Edition Premium can monitor your Canary Deployments when deploying your applications with Kubernetes.



[Learn more about configuring Canary Deployments](#)

Domain Specific Language

A Domain Specific Language (DSL) for defining infrastructure configuration allows thinking in resources, not files or commands to write declarative rather than procedural code.



Automated Provisioning

Find, discover, and manage bare metal and VM servers. Provision using defined policies automatically installing the correct OS or hypervisors, based on the information discovered in your environment.



Infrastructure Orchestration

Model complex infrastructures and the dependencies between distributed services and systems that make up an application or application stack using infrastructure as code. The infrastructure orchestrator uses the model to manage deployment ensuring the right services are available, at the right time, with the right information.

