Release Report: v15.0.0

Release Date: 2022-05-22 00:00:00+00:00

Total Issues: 1

Release Description

![30 new features](https://img.shields.io/static/v1?color=108548&label;=new+features&labelColor;=525252&message;=30&style;=for-the-badge "New features added in this release") ![1671 total badges](https://img.shields.io/static/v1?color=1F75CB&label;=total+features&labelColor;=525252&message;=1671&style;=for-the-badge "Total features") [REST API for the agent for Kubernetes](https://docs.gitlab.com/ee/api/cluster_agents.html): Kubernetes Management > GitLab now includes a REST API to register and manage the agent for Kubernetes. You can view the details about your agents, register new agents, and manage agent tokens. This API supplements [the existing GraphQL API](https://docs.gitlab.com/ee/api/graphql/reference/#clusteragentconnection). You can use the REST API to automate the full agent lifecycle. > > Thank you [@tuxtimo](https://gitlab.com/tuxtimo) for your contributions! [Cluster support for Kubernetes 1.22](https://docs.gitlab.com/ee/user/clusters/agent/#supported-cluster-versions): Auto DevOps, Kubernetes Management > If you use Kubernetes, GitLab wants to ensure you have full functionality when you upgrade your clusters to the most recent Kubernetes version. While many of you use GitLab to deploy your Kubernetes clusters, until recently there was no official support for Kubernetes 1.21 and 1.22. This release brings full support for all of the Kubernetes-related features from those versions. [Terraform CI/CD template authenticates to Terraform module registry](https://docs.gitlab.com/ee/user/infrastructure/iac/#integrate-your-project-with-terraform): Infrastructure as Code > If you use Terraform, you can use the module registry to store your infrastructure modules and streamline your developer experience. GitLab ships with a set of Terraform CI/CD templates that support all of the GitLab features out-of-the-box and can help even inexperienced Terraform users get started quickly. > > Previously, if you used the Terraform module registry, you needed to authenticate to the registry as part of a custom CI job, even if you used the Terraform CI/CD templates. Thanks to community contributions from @willianpaixao and @terorie, the built-in Terraform template now automatically looks in the CI job to retrieve authorized Terraform modules from the registry. [The agent server for Kubernetes enabled by default in the Helm chart](https://docs.gitlab.com/ee/administration/clusters/kas.html) *(self-managed only)*: Infrastructure as Code > The first step for using the agent for Kubernetes in self-managed instances is to enable the agent server, a backend service for the agent for Kubernetes. In GitLab 14.8 we enabled the agent server for Omnibus based installations. The feature has matured in the past few months, so we are now making the agent server enabled by default in the GitLab Helm chart as well to simplify setup for GitLab administrators. Besides being enabled by default, the agent server accepts various configuration options to customize it according to your needs. [GitLab chart improvements](https://docs.gitlab.com/charts/) *(self-managed only)*: Cloud Native Installation > - In GitLab 15.0, the [GitLab agent server (KAS)](https://docs.gitlab.com/ee/administration/clusters/kas.html), which is required to manage the GitLab agent for Kubernetes, is enabled by default. Enabling the GitLab agent server by default allows you to benefit from the GitLab agent for Kubernetes as an active in-cluster component for solving any GitLab ↔ Kubernetes integration tasks. > - Starting with GitLab 15.0, the `AES256-GCM-SHA384` SSL

cipher will not be allowed by NGINX by default. If you require this cipher (for example, if you use [AWS's Classic Load Balancer](https://docs.aws.amazon.com/en_en/elasticloadbalancing/latest/classic/elb-ssl-security-policy.html#ssl-ciphers)), you can [add the cipher back to the allowlist](https://docs.gitlab.com/omnibus/update/gitlab_15_changes.html#aes256-gcm-sha384-ssl-cipher-no-longer-allowed-by-default-by-nginx). [Omnibus improvements](https://docs.gitlab.com/omnibus/) *(self-managed only)*: Omnibus Package > - GitLab 15.0 includes [Mattermost 6.6](https://mattermost.com/blog/mattermost-v6-6-is-now-available/), whose newest release includes triggers and actions on channel messages and the general availability of Apps Framework. This version also includes [security updates](https://mattermost.com/security-updates/) and an upgrade from earlier versions is recommended. > - In GitLab 15.0, the new [default version of PostgreSQL](https://docs.gitlab.com/ee/administration/package_information/postgresql_versions) for new installs will be 13.6. Users currently on PostgreSQL 12 will stay on PostgreSQL 12, unless the user manually upgrades the PostgreSQL version. New installs can opt into PostgreSQL 12 during installation if desired. > - Starting with GitLab 15.0, the `AES256-GCM-SHA384` SSL cipher will not be allowed by NGINX by default. If you require this cipher (for example, if you use [AWS's Classic Load Balancer](https://docs.aws.amazon.com/en_en/elasticloadbalancing/latest/classic/elb-ssl-security-policy.html#ssl-ciphers)), you can [add the cipher back to the allowlist](https://docs.gitlab.com/omnibus/update/gitlab_15_changes.html#aes256-gcm-sha384-ssl-cipher-no-longer-allowed-by-default-by-nginx). > - Starting in Gitlab 15.0, when the version of PostgreSQL changes, `postgresql` and `geo-postgresql` services [are automatically restarted](https://docs.gitlab.com/omnibus/update/gitlab_15_changes.html#automatic-restart-of-postgresql-service-on-version-change). Restarting PostgreSQL services causes downtime, due to the temporary unavailability of the database for operations. While this restart is mandatory for proper functioning of the database services, you might want more control over when the PostgreSQL is restarted. For that purpose, you can choose to skip the automatic restarts as part of `gitlab-ctl reconfigure` and manually restart the services. Users can also skip automatic restarts as part of GitLab 15.0 upgrade. [Follow or unfollow someone from the user popup](https://docs.gitlab.com/ee/user/profile/#user-activity): User Profile > Before this release, you could only follow or unfollow a GitLab user from their user profile. It was difficult to know if you were following a specific user unless you viewed the user's profile. > > In this release, thanks to [Kev](https://gitlab.com/KevSlashNull)'s contribution, you can quickly follow and unfollow users through the user popup, no matter where you are in your GitLab workflow: in notes, issues, and more. This reduces the extra step of going to the user's profile, and makes it easier to follow and unfollow other GitLab users. [Migration support for project releases milestones](https://docs.gitlab.com/ee/user/group/import/#migrated-resources): Importers > We continue to add support for more release metadata to [GitLab migration](https://docs.gitlab.com/ee/user/group/import/). In GitLab 15.0, we've added [project releases milestone](https://docs.gitlab.com/ee/user/project/releases/#associate-milestones-with-a-release). This metadata will help you migrate more of the release data without needing to manually copy over missing release attributes. [Revoke a personal access token without PAT ID](https://docs.gitlab.com/ee/api/personal_access_tokens.html#using-a-request-header): Authentication and Authorization > In previous versions of GitLab, personal access tokens could be deleted only by the ID. Because none of the endpoints return an ID from a given value, you couldn't delete a personal access token if you only had the token value. > > You can also now use the `personal_access_tokens/self` endpoint to > revoke a PAT with a single request. The endpoint revokes the PAT used to make the request, making it easy to quickly revoke PATs in case of a leak. > > Thank you [Hemanth Krishna](https://gitlab.com/DarthBenro) for your contribution! [Access and Verify actions for environments](https://docs.gitlab.com/ee/ci/yaml/index.html#environmentaction): Pipeline Authoring, Continuous Delivery, Continuous Verification > Previously, when using environments, only one keyword existed to specify that a job is executing a task that does not trigger a deployment, or create or stop an environment. This `environment: action: prepare` keyword was intended for jobs that assist in preparing an environment. However, there are many other deployment related tasks beyond preparing an environment, and users have overloaded the `prepare` keyword to perform these tasks. In 15.0, we have added two new keywords to execute tasks that require access to environment scope variables. In the `.gitlab-ci.yaml` file, you can now add a generic `environment: action: access` keyword for a broad set of use cases and `environment: action: verify` when specifically needing to verify the results during a deployment. [Automatically create release notes from annotated tags](https://docs.gitlab.com/ee/user/project/releases/#release-notes-description): Release Orchestration > Previously, release note descriptions were empty when a release is created. With this update, when creating releases in the UI based on a tag, you can now easily include that tag's message in the release notes. You can select the checkbox option in the UI to append the tag's message to the release notes section of the release. This change makes it easier to incorporate important content such as a changelog or feature list into the published release. [Release API endpoint for groups](https://docs.gitlab.com/ee/api/group_releases.html): Release Orchestration > We have added a new endpoint for the Groups API that enables you to retrieve [releases](https://docs.gitlab.com/ee/user/project/releases/) for all projects within a group. This allows users or consumers of the API to conveniently get a holistic view of releases at the group level. The endpoint supports sorting by `created_at` date and pagination. [Multiple on_stop jobs for an environment](https://docs.gitlab.com/ee/ci/yaml/#environmenton_stop): Environment Management > Previously, when using the `environment:on_stop` keyword, only one job could be specified and run as part of closing an environment. In GitLab 15.0, you can now specify multiple jobs with the `on_stop` keyword in your `.gitlab-ci.yaml` file that run in parallel when closing an environment to enable more complex environment teardown procedures. [Set Environment tier via API](https://docs.gitlab.com/ee/api/environments.html#create-a-new-environment): Environment Management > Previously, the only way to set the `deployment_tier` for an Environment was to use the keyword in the `.gitlab-ci.yml` file. In 15.0, we have added an endpoint to the Environment API to set the tier. ##### [Plan](https://about.gitlab.com/stages-devops-lifecycle/plan/) [Reorganize issue description list items with drag and drop](https://docs.gitlab.com/ee/user/project/issues/managing_issues.html#edit-an-issue): Team Planning > Issue descriptions are used to capture a lot of different types of information such as checklists, outlines, and implementation details. You can now easily reorganize a description's list items by dragging and dropping them without having to edit and save the full description. [Users with the Reporter role can manage iterations and milestones](https://docs.gitlab.com/ee/user/permissions.html): Team Planning > We've changed the permissions necessary to create, edit, and delete milestones and iterations from the Developer to Reporter role. > This change better reflects the typical day-to-day Reporter responsibilities of managing and tracking planning timeboxes. [Internal notes](https://docs.gitlab.com/ee/user/discussions/index.html#add-an-internal-note): Service Desk, Team Planning > In many cases, organizations want to keep issues and epics public, but apply stricter governance to conversations within them. For example, when using GitLab issues as part of Service Desk workflows, organizations may want to make core details about an issue public, but not to expose customer-specific confidential data broadly. > > With internal notes, you can redact discussions with internal or customer data that should only be visible to certain users, while keeping the core details about an issue public. Internal notes in issues or epics can only be seen by the issue author, assignee, and group or project members with at least the Reporter role. > > Thanks [@leetickett](https://gitlab.com/leetickett) for collaborating with our team on this feature! [Link external organizations and contacts to issues](https://docs.gitlab.com/ee/user/crm/): Team Planning > GitLab 15.0 introduces the first [MVC](https://handbook.gitlab.com/handbook/product/product-principles/#the-minimal-viable-change-mvc) toward [managing and billing external customers from GitLab](https://gitlab.com/groups/gitlab-org/-/epics/5323). With the customer relations management (CRM) feature, you can: > > - Create organizations and contacts. > - Set a default bill rate for an organization. > - Add contacts to organizations. > - Link contacts to issues with the `/add_contacts` quick action. > - View issues associated with a given contact or all contacts belonging to an organization. > > The customer relations feature is not enabled by default, and can only be managed from a top-level group. If you want to help shape the future direction for customer relations management within GitLab, please [contribute to this issue](https://gitlab.com/gitlab-org/gitlab/-/issues/361946). > > Thanks [@leeticket](https://gitlab.com/leetickett) for the dozens of contributions and countless hours spent leading this effort! ##### [Create](https://about.gitlab.com/stages-devops-lifecycle/create/) [Multiple account support for GitLab Workflow in VS Code](https://gitlab.com/gitlab-org/gitlab-vscode-extension#setup): Editor Extension > When setting up [GitLab Workflow](https://marketplace.visualstudio.com/items?itemName=GitLab.gitlab-workflow) for VS Code, you must provide a token to authenticate to GitLab. This token authenticates you to your GitLab instance as a particular user for checking out code, seeing issues, reviewing merge requests, and more. > > In [GitLab Workflow 3.44](https://gitlab.com/gitlab-org/gitlab-vscode-extension/-/blob/main/CHANGELOG.md#3440-2022-05-13), you can now use multiple tokens to authenticate to the same GitLab instance. This can be great for users who have both work and personal accounts, or accounts with separated duties. > > We've also improved key storage for tokens, which will now be stored in VS Code's SecretStorage, and is backed by your operating system keychain. [Edit code blocks, links, and media inline in the WYSIWYG editor](https://docs.gitlab.com/ee/user/project/wiki/#content-editor): Wiki > GitLab 15.0 includes a few exciting improvements to speed up your workflow in the WYSIWYG Markdown editor for your wikis. > > First, you'll find no more un-styled, monochrome code blocks: choose from over 100 languages in the dropdown list above the code block so your `CSS`, `YAML`, and `Python` code are distinct from each other with accurate syntax highlighting. The code blocks will even inherit your preferred syntax highlighting theme. You can also quickly copy the code block to your clipboard for use in your code editor of choice. > > You'll also find working with links and media in the WYSIWYG editor easier than ever. Previously, you had to select from the editing toolbar to change a selected link or image on your wiki page, with some edits requiring you to delete the link or image and re-create it. Editing links and images is now easier, with a new popover menu that appears when you select a link or attached image. From the menu you can quickly edit a link's destination URL or description, copy the link or image to your clipboard, or even remove the link or image from the page. ##### [Verify](https://about.gitlab.com/stages-devops-lifecycle/verify/) [Project-level Secure Files in open beta](https://docs.gitlab.com/ee/ci/secure_files/): Continuous Integration (CI) > Previously, it was difficult to use binary files in your CI/CD pipelines because CI/CD variables could only contain text values. This made it difficult to make use of profiles, certificates, keystores, and other secure information, which are important for teams building mobile apps. > > Today we are releasing Project-level Secure Files in [open beta](https://handbook.gitlab.com/handbook/product/categories/gitlab-the-product/#open-beta). Now you can upload binary files to projects in GitLab, and include those files in CI/CD jobs to be used in the build and release processes as needed. Secure Files are stored outside of version control and are not part of the project repository. > > Please leave us your feedback about your experience with this feature in the [feedback issue](https://gitlab.com/gitlab-org/gitlab/-/issues/362407). [View more details about each runner](https://docs.gitlab.com/ee/administration/admin_area.html#administering-runners): GitLab Runner > Previously, if wanted an at-a-glance view of a runner's relevant information, you had to switch between screens or even use the API to retrieve the details. Now, administrators can view the runner's executor, architecture, and platform on the runner's detail view. These details can help you quickly determine essential details, which are critical for troubleshooting issues or managing day-to-day operations and maintenance tasks. [GitLab Runner 15.0](https://docs.gitlab.com/runner): GitLab Runner > We're also releasing GitLab Runner 15.0 today! GitLab Runner is the lightweight, highly-scalable agent that runs your CI/CD jobs and sends the results back to a GitLab instance. GitLab Runner works in conjunction with GitLab CI/CD, the open-source continuous integration service included with GitLab. > > #### What's new: > > - [Support for Windows Server 2022](https://gitlab.com/gitlab-org/gitlab-runner/-/issues/27859) > - [Add support for Ubuntu 22.04 LTS - Jammy Jellyfish](https://gitlab.com/gitlab-org/gitlab-runner/-/issues/29045) > > #### Bug Fixes: > > - [Image pull failed: Back-off pulling image error in Kubernetes executor](https://gitlab.com/gitlab-org/gitlab-runner/-/issues/27664) > - [AWS credential exposure in failed GitLab Runner cache upload to S3](https://gitlab.com/gitlab-org/gitlab-runner/-/issues/4625) > > The list of all changes is in the GitLab Runner [CHANGELOG](https://gitlab.com/gitlab-org/gitlab-runner/blob/15-0-stable/CHANGELOG.md). [Show instance CI/CD limits in /help](https://docs.gitlab.com/ee/administration/instance_limits.html#cicd-limits) > Instance Administrators can set a number of CI/CD limits in the Admin Area of their instance. It is hard for users without administrator access to know what the limits are, especially when pipelines are failing because they bump into limits they didn't know existed. > > Now the CI/CD related limits for the instance can be seen on the instance configuration page found at `/help/instance_configuration`. Thanks to @wwwjon for this contribution! [Use nested CI/CD variables with environments in pipeline configuration](https://docs.gitlab.com/ee/ci/yaml/#environment): Pipeline Authoring > Using CI/CD variables with the `environment` keyword in your CI/CD configuration is great, because it lets you [create environments dynamically](https://docs.gitlab.com/ee/ci/environments/#create-a-dynamic-environment). While this is already a powerful feature, there were still some limitations, because you could not use nested variables to define environments. > > Starting in GitLab 15.0, you can nest variables inside other variables, and have them all expand the way you expect. This makes dynamic environments even more powerful due to the increased flexibility! ##### [Application security testing](https://about.gitlab.com/stages-devops-lifecycle/application_security_testing/) [Secure and Protect analyzer major version update](https://docs.gitlab.com/ee/user/application_security/): API Security, Container Scanning, Dependency Scanning, DAST, Fuzz Testing, License Compliance, SAST, Secret Detection > Secure and Protect features now use a new major version for all [analyzers](https://docs.gitlab.com/ee/user/application_security/terminology/#analyzer). This version change clearly distinguishes between analyzers that are: > > - Released prior to May 22, 2022, and generate reports that _are not_ subject to [strict schema validation](https://docs.gitlab.com/ee/development/integrations/secure.html#report-validation). > - Released after May 22, 2022, and generate reports that _are_ subject to [strict schema validation](https://docs.gitlab.com/ee/development/integrations/secure.html#report-validation). > > Schema validation makes GitLab analyzers and third-party integrations more reliable. > > If you use the GitLab-managed CI/CD templates, you don't have to take any action. > The analyzers used in your pipelines automatically update to the latest version. > > If you use a custom template, or if you've pinned analyzer versions, you need to update your CI/CD job definition to either remove the pinned version or update to the latest major version. > > All new bug fixes and features will now be released in the new analyzer major versions. > These improvements won't be available in the deprecated analyzer versions because we don't backport bug fixes and new features; see GitLab's [maintenance policy](https://docs.gitlab.com/ee/policy/maintenance.html). > As required, security patches will be backported within the latest 3 minor GitLab releases. > > The new versions of all analyzers are: > > - API Security: version 2 > - Container Scanning: version 5 > - Coverage-guided Fuzz Testing: version 3 > - Dependency Scanning: version 3 > - Dynamic Application Security Testing (DAST): version 3 > - Infrastructure as Code (IaC) Scanning: version 2 > - License Scanning: version 4 > - Secret Detection: version 4 > - Static Application Security Testing (SAST): version 3 for [all analyzers](https://docs.gitlab.com/ee/user/application_security/sast/#supported-languages-and-frameworks), except version 4 for the `gosec` analyzer [Static Analysis analyzer updates](https://docs.gitlab.com/ee/user/application_security/sast/analyzers): SAST, Secret Detection > GitLab Static Analysis includes [many security analyzers](https://docs.gitlab.com/ee/user/application_security/sast/#supported-languages-and-frameworks) that the GitLab Static Analysis team actively manages, maintains, and updates. The following analyzer updates were published during the 15.0 release milestone. These updates bring additional coverage, bug fixes, and improvements. > > - Brakeman analyzer updated to version 5.2.2. See [CHANGELOG](https://gitlab.com/gitlab-org/security-products/analyzers/brakeman/-/blob/master/CHANGELOG.md#v2231) for details. > - Update handling of conditionals, reflection, and nil values > - Add additional String methods for SQL injection check > - Update parser for Ruby 3.1 support > - Secrets analyzer updated. See [CHANGELOG](https://gitlab.com/gitlab-org/security-products/analyzers/secrets/-/blob/master/CHANGELOG.md#v400) for details. > - Add detection of Yandex Cloud tokens > - Remove [deprecated CI/CD variables](https://docs.gitlab.com/ee/update/deprecations#secret-detection-configuration-variables-deprecated) > > If you [include the GitLab-managed SAST template](https://docs.gitlab.com/ee/user/application_security/sast/#configure-sast-in-your-cicd-yaml) ([`SAST.gitlab-ci.yml`](https://gitlab.com/gitlab-org/gitlab/blob/master/lib/gitlab/ci/templates/Security/SAST.gitlab-ci.yml)), you don't need to do anything to receive these updates. However, if you override or customize your own CI/CD template, you need to update your CI/CD configurations. To remain on a specific version of any analyzer, you can [pin to a minor version of an analyzer](https://docs.gitlab.com/ee/user/application_security/sast/#pinning-to-minor-image-version). Pinning to a previous version prevents you from receiving automatic analyzer updates and requires you to manually bump your analyzer version in your CI/CD template. > > For previous changes, see [last month's updates](https://about.gitlab.com/releases/2022/04/22/gitlab-14-10-released/#static-analysis-analyzer-updates). [Semgrep-based SAST scanning available for early adoption](https://docs.gitlab.com/ee/user/application_security/sast/analyzers.html#transition-to-semgrep-based-scanning): SAST > You can now switch to Semgrep-based scanning for many languages in GitLab SAST. > Semgrep-based scanning brings significantly faster analysis, reduced usage of CI minutes, and more customizable scanning rules compared to existing [language-specific analyzers](https://docs.gitlab.com/ee/user/application_security/sast/analyzers/). > As of GitLab 15.0, it [supports](https://docs.gitlab.com/ee/user/application_security/sast/#supported-languages-and-frameworks) C, Go, Java, JavaScript, Python, and TypeScript. > > In a future release, we'll change GitLab SAST to use only Semgrep-based scanning by default for supported languages, and we'll [remove the language-specific analyzers that also scan them](https://docs.gitlab.com/ee/update/deprecations#sast-analyzer-consolidation-and-cicd-template-changes). (This change was previously scheduled for GitLab 15.0; work to complete it is tracked in [this deprecation issue](https://gitlab.com/gitlab-org/gitlab/-/issues/352554).) > > You can now choose to [disable deprecated language-specific analyzers early](https://docs.gitlab.com/ee/user/application_security/sast/analyzers.html#activating-semgrep-based-scanning-early) and use Semgrep-based scanning instead before we change the default behavior. We've updated documentation to [explain the transition](https://docs.gitlab.com/ee/user/application_security/sast/analyzers.html#transition-to-semgrep-based-scanning), including guidance on when to make the change in your pipelines. ##### [Software supply chain security](https://about.gitlab.com/stages-devops-lifecycle/software_supply_chain_security/) [GitLab advisory data included in container scanning results](https://docs.gitlab.com/ee/user/application_security/container_scanning/#vulnerabilities-database): Container Scanning > GitLab Container Scanning relies on information from its analyzers to report vulnerabilities. Ensuring that databases have the most up to date information is important to make sure scans are returning both accurate and timeline results. > > GitLab provides our own advisory database, which sources additional information that may not be updated in common sources. These [external sources](https://gitlab.com/gitlab-org/security-products/gemnasium-db/-/blob/master/SOURCES.md#tracking-external-sources) are tracked and information is updated daily. GitLab now includes this information when the `trivy` analyzer used with in GitLab Container Scanning, to help ensure that the most comprehensive and up-to-date vulnerability data is available for identifying vulnerabilities. > > In GitLab Ultimate, the proprietary [GitLab Advisory Database](https://gitlab.com/gitlab-org/security-products/gemnasium-db) is used for these scans. In the Free and Premium tiers, the open source [GitLab Advisory Database (Open Source Edition)](https://gitlab.com/gitlab-org/advisories-community) is used, which is a one-month time-delayed clone of the proprietary database. > > These databases give you access to additional threat information identified by GitLab's research team, even if that threat data has not yet been added to other public databases. [Container Scanning available in all tiers](https://docs.gitlab.com/ee/user/application_security/container_scanning/#capabilities): Container Scanning > [Container Scanning](https://docs.gitlab.com/ee/user/application_security/container_scanning/) helps developers to easily find known security vulnerabilities in dependencies that are installed in their container images. With GitLab 15.0, we are making the basic Container Scanning features available in every [GitLab tier](/pricing/).

Tasks

ID	Title	State	Assignee

659795	Move HSTS header to Nginx?	closed	Pawe■ Chojnacki