



GitLab

Virtual Workshop:
Automating Your Organization's
Software Factory with GitLab



GitLab

Welcome

Automating Your Organization's Software Factory Workshop

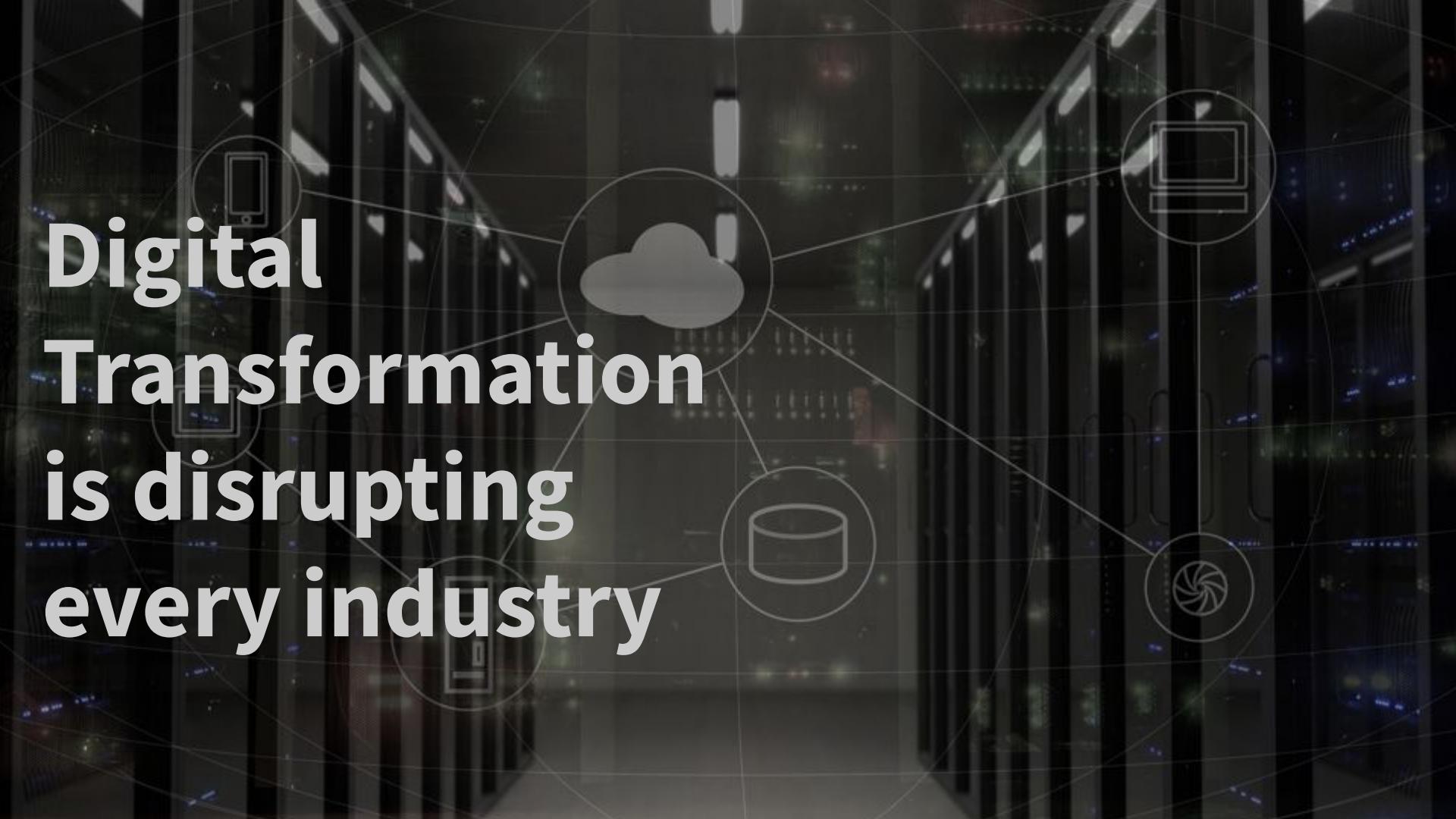


Time (PST)	Topic	Speaker
09:00 - 09:05	Welcome	Christine Saah
09:05 - 09:25	Introduction to GitLab	Joseph Valeriano
09:25 - 09:45	Using GitLab for Project Management	Joseph Valeriano
09:45 - 09:55	Lab for Project Management	Joseph Valeriano
09:55 - 10:05	Q/A and Break	Christine Saah
10:05 - 10:25	GitLab Basics for Developers	Daryl Knopf
10:25 - 10:45	Lab for GitLab Basics	Daryl Knopf
10:45 - 10:55	Q/A and Break	Christine Saah
10:55- 11:20	Digging Deeper into CI/CD	Joseph Valeriano
11:20 - 11:40	Lab for Digging Deeper into CI/CD	Joseph Valeriano
11:40 - 12:00	Q/A and Closing	Christine Saah



GitLab

Introduction to GitLab



Digital Transformation is disrupting every industry

70% of Digital Transformations fail¹ ...



Complexity, bottlenecks, and inefficiency

manual processes, functional silos, waiting for handoffs, uncertainty

Poor collaboration and communication between teams

misaligned priorities, duplication of work, unsure of who is doing what, when and why

Periodic crisis, unsatisfying tradeoffs

security breaches, bugs caught in production, downtimes, security & compliance vs. speed

Lack of visibility and traceability

siloed teams, hard to identify & fix bottlenecks, expensive to report, audit, comply

¹ <https://www.mckinsey.com/industries/retail/our-insights/the-how-of-transformation>

... resulting in



Poor quality products, over budget, unexpected costs

Frequent rework, wasted cycles and budget, lower employee productivity

Poor velocity, lost market opportunities, product launch delays

Poor business agility, unable to capture new opportunities and get quick market feedback

Poor developer experience, high employee turnover

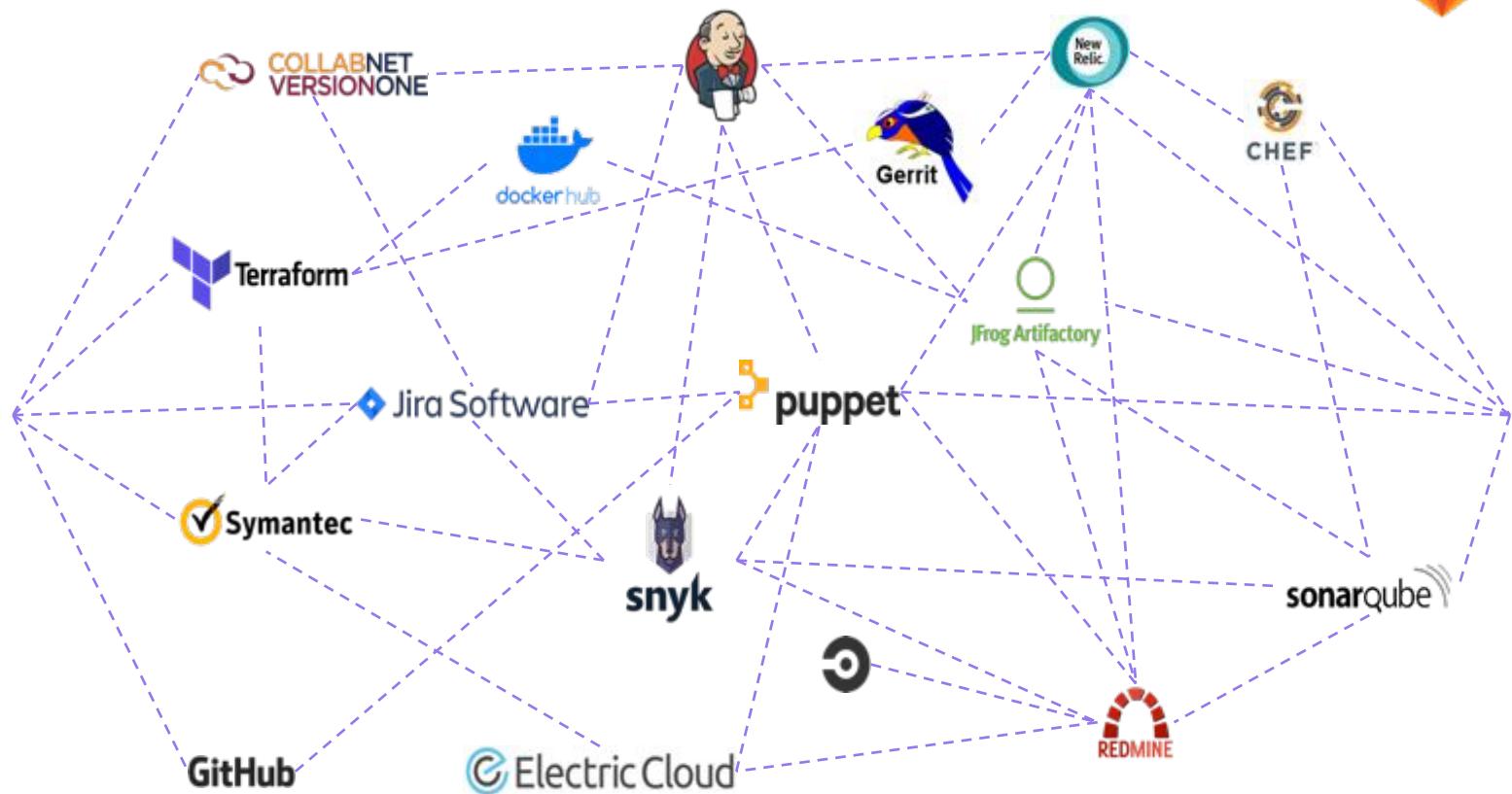
poor developer experience, high employee turnover, hard to recruit or retain talent

Business disruption, damaged reputation, customer attrition

Simplify the software development toolchain to reduce total cost of ownership

DevOps is transforming how companies deliver software

Today's software delivery landscape



Toolchain tax - multiple tools, fragile integrations, hard to scale, outages

GitLab aims to replace multiple tools across the lifecycle



Manage	Plan	Create	Verify	Package	Secure	Release	Configure	Monitor	Defend

Reducing your footprint



GITLAB IS REPLACING POINT TOOLS...

- Project management →
- SCM →
- CI/CD →
- Issue tracking →
- Container registries →
- Logging →
- Dependency scanning →
- License management →



AND ENABLING ENTIRELY NEW CAPABILITIES...

- Concurrent development
- Moving security forward (SecDevOps)
- Seamless collaboration
- Full accountability
- Cycle time measurement
- Transparency
- Real-time feedback
- ...

A single, intuitive user experience, data model and integrations

A Complete DevOps Platform Delivered as a Single Application



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Defend

Single Conversation
Single Data Store
Single Permission Model
Single Interface
Governance & Security
Team Collaboration
Lifecycle Analytics

Quantifiable benefits of simplicity



Increase Operational Efficiencies



One consistent view and collaboration space for Dev, Ops, and Security teams

Public cloud independent, deploy anywhere, SaaS and/or self-managed

Deliver Better Products Faster



Ability to work in parallel, get feedback and not have to wait on other teams

Automate testing, security, deployments to minimize manual intervention

Assess and resolve security, compliance, and code quality issues at the point of code change

Reduce Security and Compliance Risk



Decrease security exposure, cleaner and easier audits, reduce disruptions

GitLab streamlines the DevOps adoption journey



DevOps
adoption
stages

DevOps Initiatives

Better collaboration,
higher IT productivity

DevOps Maturity

Faster releases, fewer
errors, lower costs

Digital Transformation

Faster time-to-market,
increased revenue



Portfolio

Epics and roadmaps
enable visualizing and
prioritizing future work

➤ Typical starting points



Project

Backlog, sprint,
burndown, tracking
deliverable



SCM

Code reviews,
collaboration, developer
experience, productivity



CI

Automated
build/test pipeline,
quality and speed



Security

Security Dashboard,
Shift left Security



CD

Multi cloud,
flexible infrastructure,
infrastructure as code

Holistic value increases with GitLab adoption

Holistic value increases with GitLab adoption

Reduced cost

- Single license
- Reduced maintenance
- Reduced integration costs

Auditability

- Single data store

Improved toolchain security

Speed of innovation

Improved compliance

Continuous improvement visibility and analytics

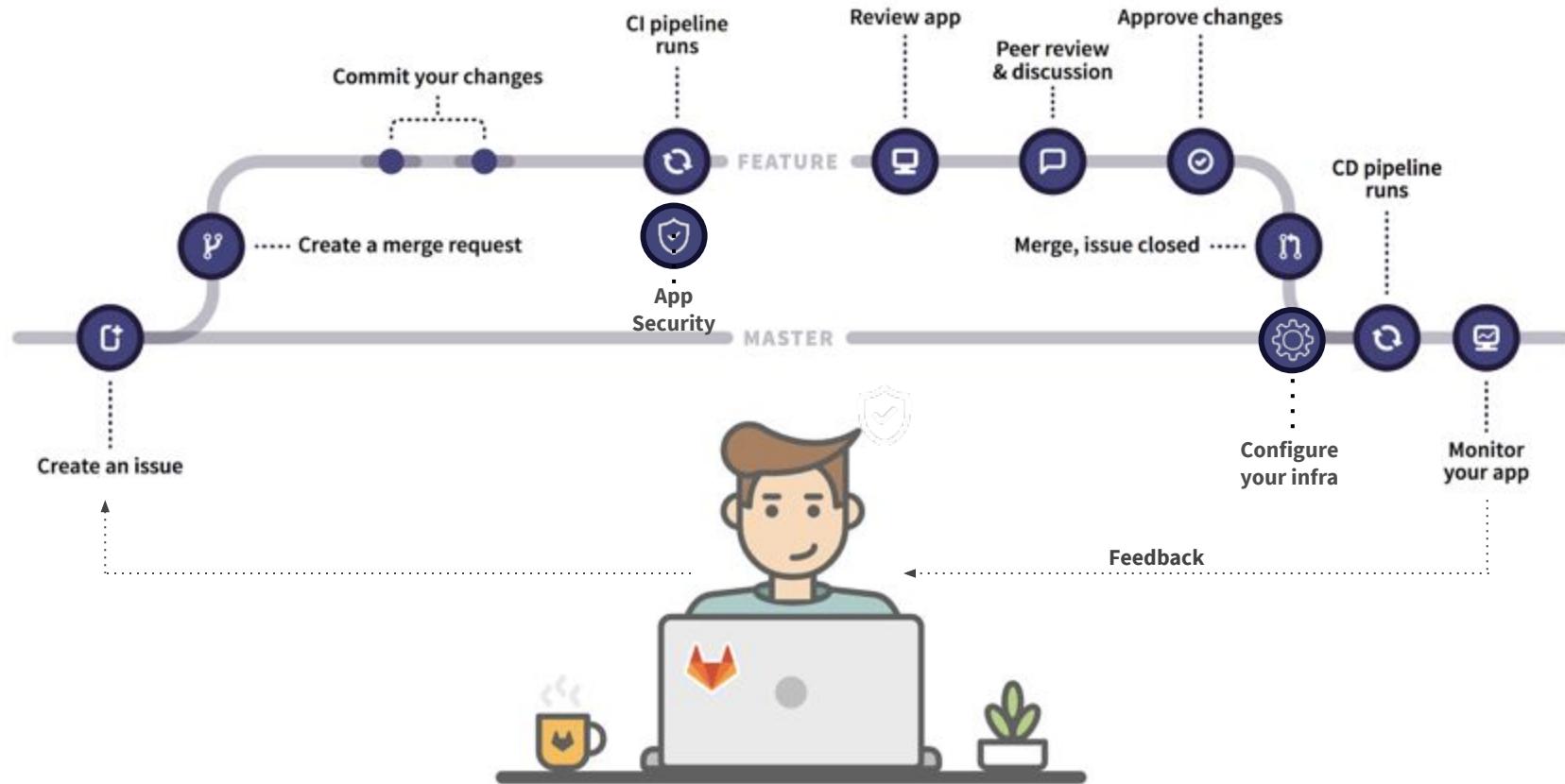
- Cycle time analytics
- Single pane of glass visibility

Seamless collaboration

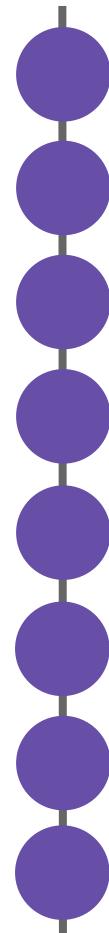
Developer experience

- Single interface
- Developer flexibility (movement between projects/new toolsets)

DevOps Best Practices Built In



Complete DevOps



- Leading SCM and CI in one application
- Built in security and compliance
- End to end insight and visibility
- Deploy your software anywhere
- Flexible GitLab hosting options
- Rapid innovation
- Open Source: Everyone can contribute
- Collaborative & transparent customer experience



GitLab

Everyone can contribute

Integrating SCM and CI doesn't have to be painful



“Learning one tool's conventions (GitLab) has been so much better than trying to glue three different tools together (e.g. Bitbucket, Jira, Jenkins, Confluence), each with their own words.

In GitLab, the repository is the center of all attention, and there's no ambiguity about how each feature (say a build or a documentation site) relates to it.”

Customer Quote from Q1 FY20 System Usability Scale Survey

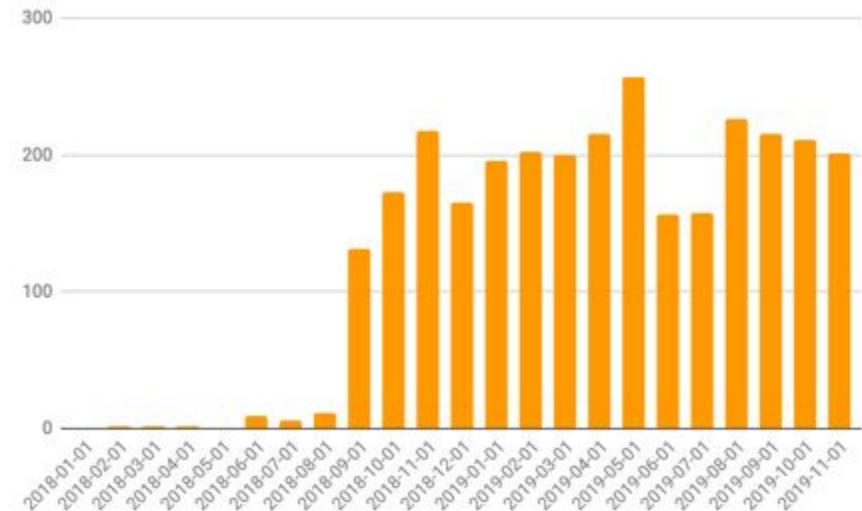
How can we achieve our vision?



BUILT ON OPEN SOURCE

- **THE POWER OF CONVENTION:** leverage the DevOps best practices of 100K+ organizations across the world.
- **EVERYONE CAN CONTRIBUTE:** passionate, vocal, global community of 2200+ people and organizations contribute code.
- **CO-CREATION:** with thousands of public feature proposals, GitLab delivers customer-driven innovation
- **CONTINUOUS INNOVATION:** since 2011 GitLab has consistently released new features/innovations on the 22nd of every month.
- **OPEN ECOSYSTEM:** built upon powerful open source technologies like Kubernetes and Prometheus.

Community contributions merged per month



CI



ticketmaster®

GitLab's platform helps Ticketmaster deliver higher-quality features to fans more quickly and more consistently



esa
European Space Agency

GitLab simplifies ESA's DevOps toolchain and allows team to cross borders, increase cooperation and reshape working culture



**Goldman
Sachs**

GitLab removes toolchain complexity and accelerates DevOps adoption, shortening release cycles from once every 1-2 weeks to once every few minutes



Glympse

GitLab streamlines Glympse's development processes improving compliance, security scanning and deploy time



Worldline

GitLab quickly became the backbone of Worldline's development environment, improving code review by 120x



**CLOUD NATIVE
COMPUTING FOUNDATION**

GitLab's open-source platform provides a unified CI / CD system, improves collaboration, and allows for interoperability across multiple cloud providers



GitLab

Using GitLab as a Project Management Tool

2021

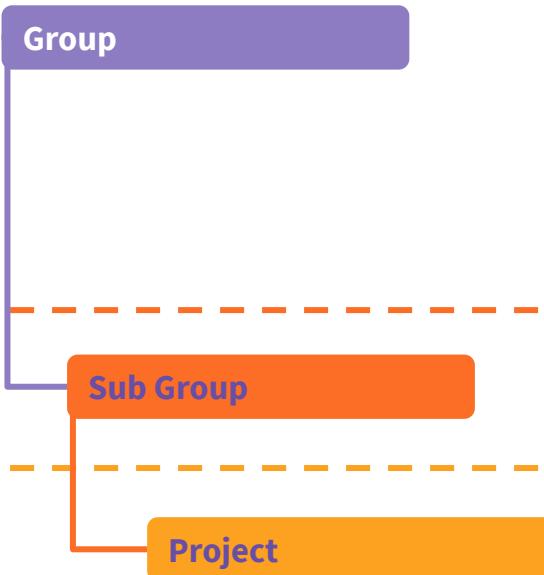
Joseph Valeriano
GitLab Solutions Architect

A Complete DevOps Platform Delivered as a Single Application



Single Conversation
Single Data Store
Single Permission Model
Single Interface
Governance & Security
Team Collaboration
Lifecycle Analytics

Organizing the work



GitLab Groups provide capabilities for assembling related projects together and grant members access to several projects at once.

The **Group** provides a layer for strategic planning, governance, and management.

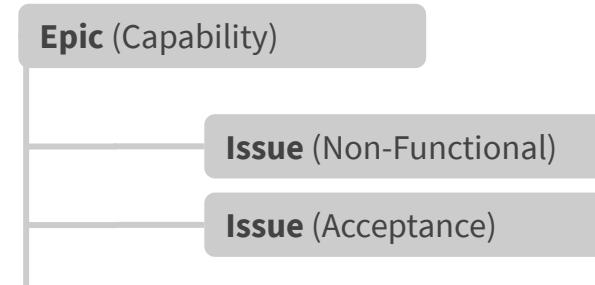
Sub Groups are nested, child Groups for additional levels of organization.

Projects are where teams collaborate, plan work, write code, and deliver applications.

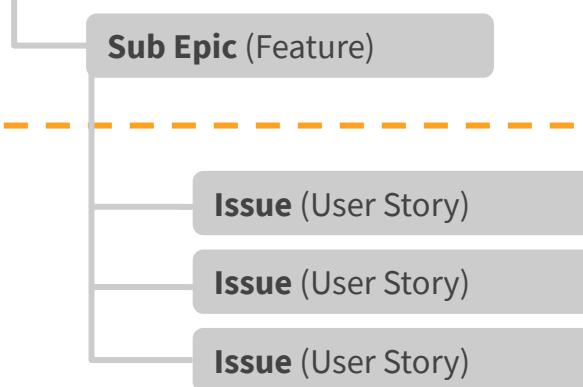
Defining the work



Epics provide grouping of Epics and Issues.



Sub Epics represent smaller pieces of a larger Epic, larger than an Issue.



Issues describe small, discrete pieces of work.

Planning the work



Roadmaps are visual representations of an Epic with a defined *Start* and *End Date* within a group, and are shown in a timeline view (e.g. a Gantt chart)

Roadmap

Milestone

Roadmaps exist at the Group and Sub Group level.

Roadmap

Milestones group and track *Issues* and *Merge Requests* against a set period of time.

Milestone

Milestones exist at the Group, Sub Group, and Project level.

Doing the work



Project

The *Git Repository* is the foundation of a **GitLab Project**.

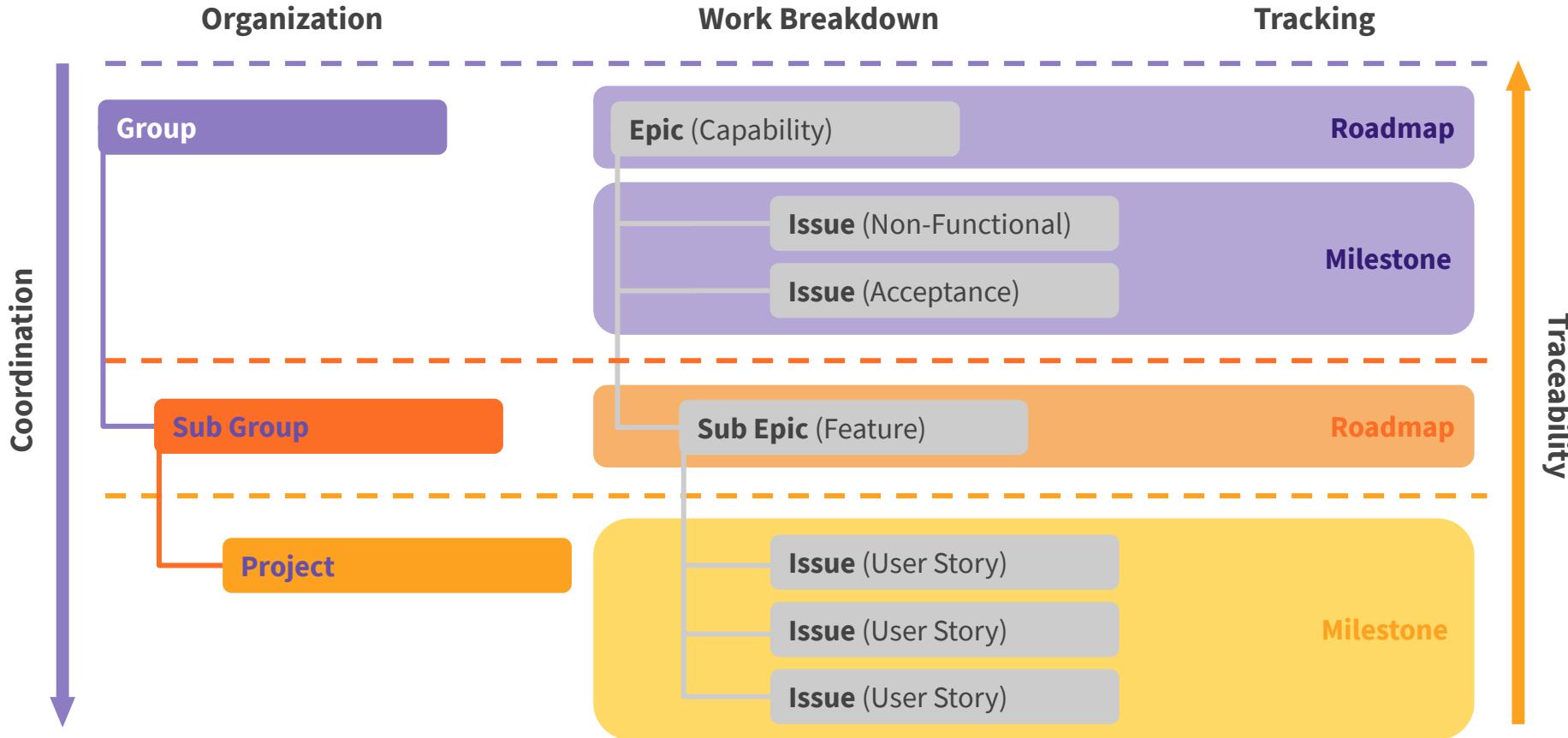
Merge Requests are project assets to visualize and collaborate on proposed changes to source code on a Git branch.

CI/CD Pipelines describe code build and test jobs and stages for a Merge Request.

Code Reviews are discussion threads on Merge Requests where developer peers review and comment on changes to the code.

Approvals are explicit acceptance events, recorded in the Merge Request, signaling sign-off on associated code changes.

GitLab Agile Planning Structure



Label It All

Labels allow you to categorize **epics**, **issues**, and **merge requests**

- Allow you to quickly and dynamically filter
- Use labels to help search
- There are two types of labels in GitLab:

Project labels and **Group labels**

The screenshot shows the GitLab Labels interface. At the top, there are tabs for 'All' and 'Subscribed'. Below that, a note says 'Labels can be applied to issues and merge requests.' and 'Star a label to make it a priority label. Order the prioritized labels to change their relative priority, by dragging.' The interface is divided into sections: 'Prioritized Labels' and 'Other Labels'. In the 'Prioritized Labels' section, there is one item: 'critical' (red circle). In the 'Other Labels' section, there are three items: 'bug' (red circle), 'confirmed' (orange circle), and 'discussion' (blue circle). Each label item includes a 'Issues · Merge requests' link, a 'Prioritized label' button, and a row of actions: 'Project label', a star icon, a pencil icon, a more options icon, and a 'Subscribe' button. A 'Prioritize' button is also visible above the 'confirmed' and 'discussion' labels.

Labels can be applied to issues and merge requests.

Star a label to make it a priority label. Order the prioritized labels to change their relative priority, by dragging.

Prioritized Labels

This screenshot shows a subset of the GitLab Labels interface. It features a 'Prioritized Labels' section with one item: 'critical' (red circle). Below it is an 'Other Labels' section with two items: 'bug' (red circle) and 'confirmed' (orange circle). Each label item has a 'Issues · Merge requests' link, a 'Prioritized label' button, and a row of actions: 'Project label', a star icon, a pencil icon, a more options icon, and a 'Subscribe' button.

GitLab Project and Group Hierarchy



Groups

Sub Groups

Epics

Milestones

Roadmap

Labels (group level)

Boards (group level)

Projects (multiple)

At the top of the GitLab model, the **Group** provides a layer for strategic planning, governance and management.

Groups can contain **SubGroups** and **Projects**

Groups = Portfolios and / or Programs

GitLab Projects are the home for *Issues* and *Discussions*



Groups

Sub Groups

Epics

Milestones

Roadmap

Labels (group level)

Boards (group level)

Projects

Repository (git)

Issues & Discussion

Labels (project level)

Boards (project level)

Milestones (Sprints)

Cycle Analytics

At the top of the GitLab model, the **Group** provides a layer for strategic planning, governance and management.

Groups can contain **SubGroups** and **Projects**

Groups = Portfolios and / or Programs

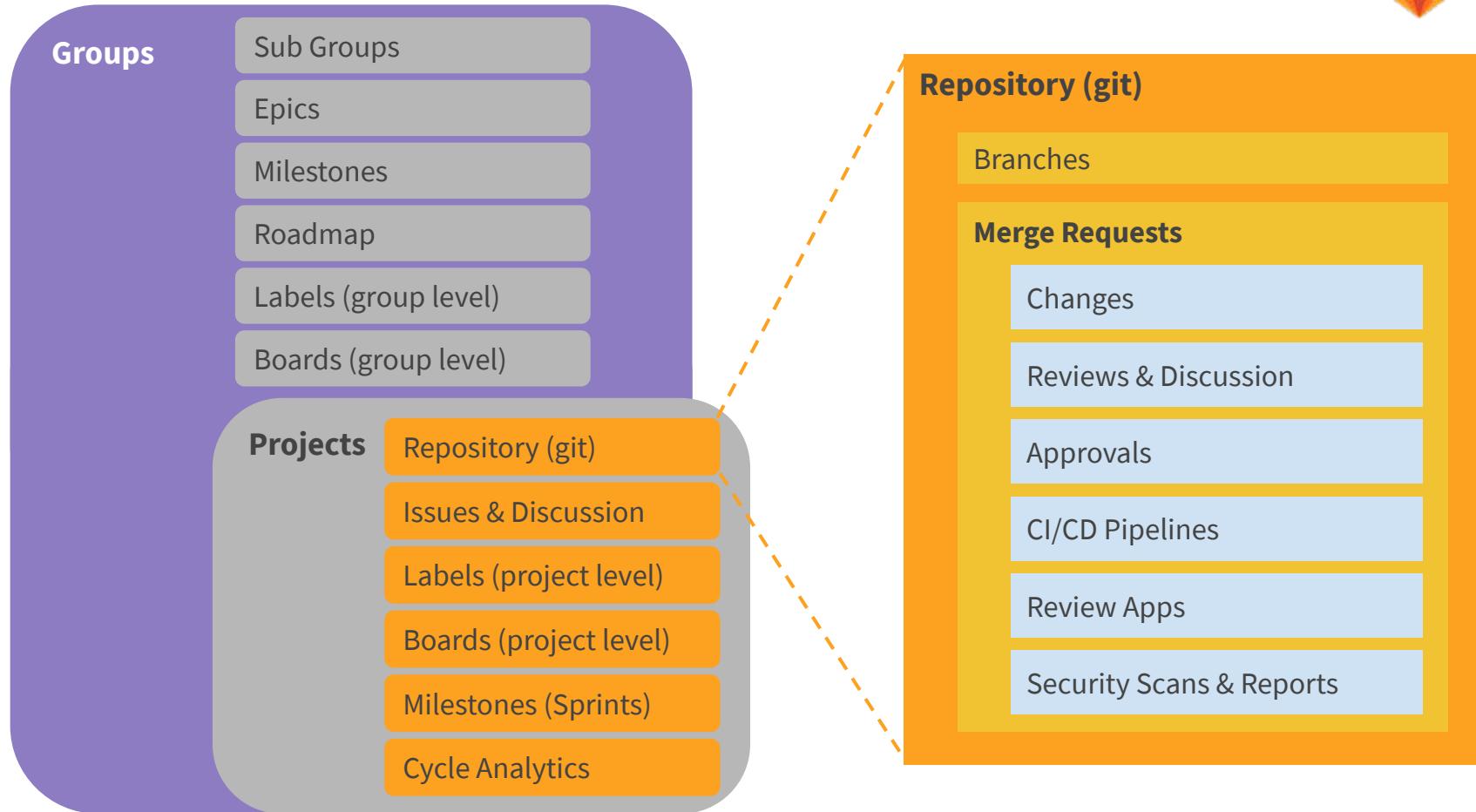
SubGroups allow for additional levels of Groups

Projects are where teams collaborate, plan work, write code, and deliver applications.

Issues = User Stories

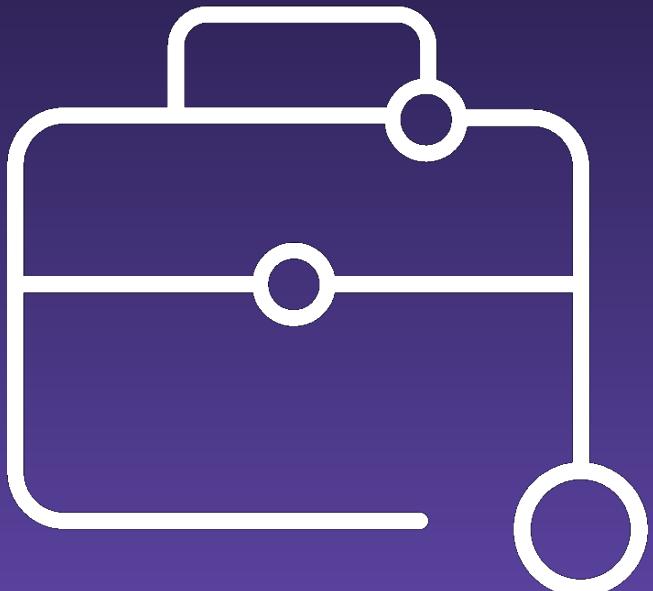
Milestones = Sprints

The Repository is the foundation of GitLab Projects





Portfolio Management



Portfolio Management

- Groups and Sub Groups
- Epics and Child Epics
- Roadmaps
- Milestones
- Burndown Charts
- Labels
- Boards, Group Level

Visualize delivery of Epic over time via Roadmaps



Initech

tech-marketing > ... > Initech > Epics Roadmap

Subgroup overview

Quarters Months Weeks Open epics Recent searches Search or filter results...

Epics 16

List

Roadmap

Issues 323

Merge Requests 2

Targeted teacher experience enhancement... 1 Teacher Services - Jan 20 - Feb 14, 2020

This is a new epic! Teacher Services - No start date - Mar 31

Decrease Time to Value (TTV) for Teach... 5 Teacher Services - Jan 20 - Feb 28, 2020

Improve adoption of existing products b... 2 Initech - Jan 20 - Jun 30, 2020

Improve conversion rates from 2% to 5% 0 Teacher Services - Feb 3 - Feb 28, 2020

Optimize performance of page rendering 0 Teacher Services - Feb 17 - Feb 28, 2020

Improve account management for pare... 0 User Management - Apr 1 - Jun 30, 2020

Switch to alternative payment gateway ... 0 Billing & Fulfillment - Jul 1 - Sep 30, 2020

Decrease costs through operational effi... 2 Initech - Jul 1 - Dec 31, 2020

Reduce infrastructure costs by 20% Teacher Services - Oct 1 - Dec 31, 2020

Switch to alternative payment gatew... Billing & Fulfillment - Jul 1 - Sep 30, 2020

Reduce infrastructure costs by 20% 0

Feb 26 2 9 16 23 1 8 15 22 29 5 12 19 26 3 10 17 24 31 7 14 21 28 5 12 19 26

Mar

Apr 19

May

Jun

Jul

Foo Bar

CY20 - Quarter 2

CY20 - Quarter 3

Epics Roadmap screenshot showing a timeline from February to July. The timeline is divided into quarters: CY20 - Quarter 2 (Mar 1 - May 31) and CY20 - Quarter 3 (Jun 1 - Jul 26). A red vertical line marks the end of Quarter 2 on May 31. A red arrow points down to the date '19' in April, indicating a specific milestone or release date. Several epics are listed along the timeline, each with a progress bar and a brief description. For example, 'Targeted teacher experience enhancement...' is listed for 'Teacher Services - Jan 20 - Feb 14, 2020' with a 13% completion. Other epics include 'This is a new epic!', 'Decrease Time to Value (TTV) for Teach...', 'Improve adoption of existing products b...', 'Improve conversion rates from 2% to 5%', 'Optimize performance of page rendering', 'Improve account management for pare...', 'Switch to alternative payment gateway ...', 'Decrease costs through operational effi...', 'Reduce infrastructure costs by 20%', and 'Switch to alternative payment gatew...'. The 'Epics' sidebar shows 16 total epics, with 16 listed here and 2 more under 'Merge Requests'.

Plan Milestones via Group Issue Boards



GitLab Projects Groups Activity Milestones Snippets 🔒

Search or jump to... 0:24 1 4 ? 🚙

Create: roadmap

Edit board • Add list ▾

Open

SSH push mirroring support with public-key authentication
Create Deliverable In dev P1 backend customer customer+ devops:create direction feature proposal mirror missed-deliverable missed:11.5
gitlab-org/gitlab-ce#49565 5

Send one email notification for one published code review
Create Deliverable GitLab Premium P1 Product Vision 2018 UX backend devops:create direction emails frontend merge requests
gitlab-org/gitlab-ce#4326 3

Mirror changes from Web IDE to CI runner
Accepting merge requests Create Deliverable GitLab Ultimate P1 Requires e2e tests backend devops:create direction feature proposal release post item web ide
gitlab-org/gitlab-ce#5276 5

Multiple blocking merge request approval rules
Create Deliverable GitLab Premium P1 UX approvals backend customer customer+ devops:create direction feature proposal frontend merge requests
gitlab-org/gitlab-ce#1979 5

Block command line pushes if they have the user's private email address
Create Deliverable P1 UX backend devops:create direction feature proposal movingtogitlab repository
gitlab-org/gitlab-ce#52355 3

[Frontend] Send one email notification for one published code review
Create Deliverable GitLab Premium P1 Product Vision 2018 UX ready devops direction emails frontend merge requests
gitlab-org/gitlab-ce#8349 4

[Frontend] Multiple blocking merge request
Accepting merge requests Create Deliverable GitLab Premium Product Vision 2019 UX backend customer+ devops:create direction feature proposal repository
gitlab-org/gitlab-ce#4418

Expand diff to entire file

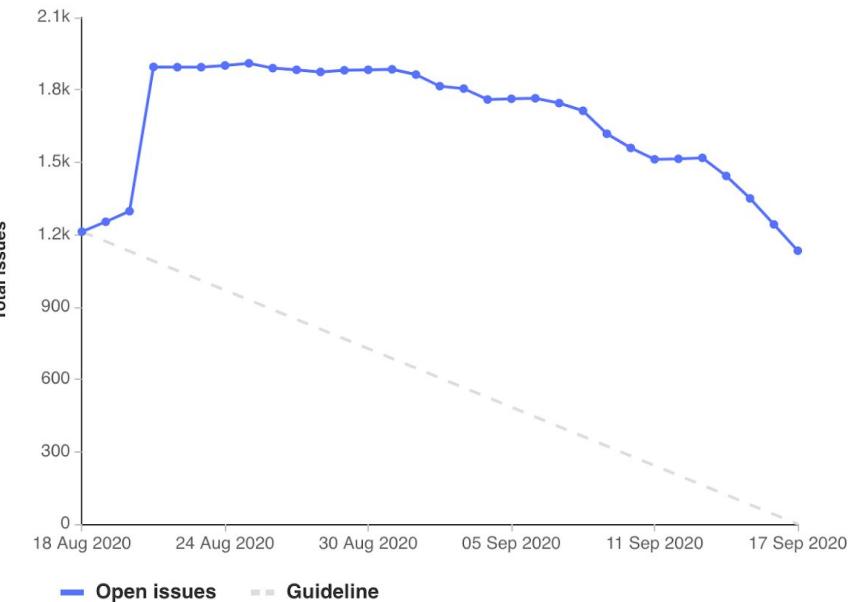
Releases Page Accepting merge re Product Vision 2019 devops:release movingto gitlab release orchestration gitlab-org/gitlab-ce

Override squash commits Accepting merge re devops:create direction frontend merge re gitlab-org/gitlab-ce

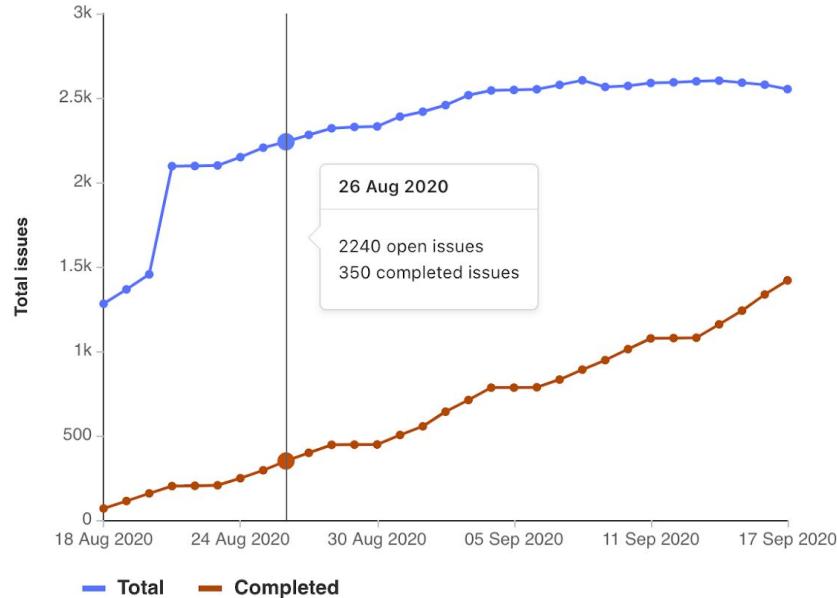
Track Milestone progress with *Burndown* and *Burnup* Charts



Burndown chart

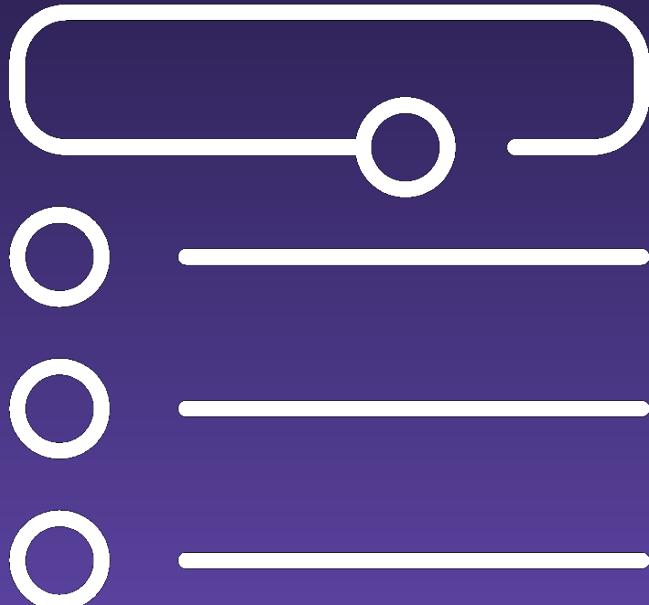


Burnup chart





Project Management



Project Management

- Projects
- Issues
- Milestones
- Burndown Charts
- Labels
- Boards, Project Level
- Cycle Analytics
- Wikis
- Service Desk

Plan a Sprint using Issues



GitLab Projects Groups Activity Milestones Snippets

Search or jump to... 0:24

GitLab Community Edition

Project Repository

Issues 13,177

Merge Requests 731

CI / CD Operations Registry Snippets Members

Closed Opened 1 year ago by Mark Fletcher

Reopen issue New issue

Status messages in GitLab

Problem to solve

Right now, a number GitLab team members add emoji to their real name when they go on vacation to indicate to other users that they may not be able to respond or for reviews for a while. Let's incorporate this into the product natively by allowing users to communicate their status.

Design

Settings page:

User Settings Edit Profile

Public Avatar

Upload new avatar

Choose file... No file chosen

The maximum file size allowed is 200KB.

Current Status

This emoji and message will appear on your profile and next to your name throughout the interface.

Your status

making magic

Main settings

This information will appear on your profile.

Name Luke Nederpenter

User ID

Labels

Community contribution Deliverable

In review Platform [DEPRECATED]

UX ready backend devops:share

direction emoji feature proposal

frontend release post item

settings user profile

Scrum and Kanban Issue Boards as Information Radiators



GitLab Projects Groups Activity Milestones Snippets

Search or jump to... D24 I5 ? v

GitLab Community Edition > GitLab.org > GitLab Community Edition > Issue Boards

Project Repository Issues 13,177 Boards Labels Service Desk Milestones Merge Requests 730 CI / CD Operations Registry Snippets Members

Workflow Search or filter results...

Rails 5.0

- regression / bug / technical debt
- Support
- Upcoming
- Upgrade to Rails 5
- victor board
- Workflow
- Create new board

Workflow 110 1501 +

In dev 42 76 +

Skip per commit Checks::ChangeAccess which have already passed on another change/branch

Accepting merge requests Create Deliverable

In dev P2 backend devops:create

missed-deliverable performance repository

#44679 3

SSH push mirroring support with public-key authentication

Create Deliverable In dev P1 backend

customer customer+ devops:create direction

feature proposal mirror missed-deliverable

missed:11.5

#849565 5

Improve consistency in the way we retrieve project & group in API endpoints

Manage Plan api missed-deliverable

technical debt

#20728

Remove all the relevant "import" columns that were migrated over to ProjectImportState from the project table

Create Deliverable In dev P2 backend

gitlab.com import mirror production request

technical debt

#850341

Add mechanism that will discard artifacts metadata if file is too large

In review 25 40 +

ExpireBuildArtifactsWorker is broken

In review P3 S3 Verify database

devops:verify missed-deliverable performance

#41057

Ensure that all CI/CD queries take less than 15 seconds to complete

Accepting merge requests In review Stretch

Verify database devops:verify meta

missed-deliverable performance

#40524

Stored XSS in merge request pages

Create Deliverable HackerOne In review P2

S2 frontend merge requests security

#51527 2

Read Name of any private groups

HackerOne In review P3 Plan S3 backend

bug security todos

#51262 Jan 31, 2019 2

Edit board Add list Add issues

Capture team Retrospectives with *Project Wikis*



Screenshot of the GitLab interface showing a Sprint 1 retrospective page.

The sidebar on the left shows project navigation links: Project, Repository, Issues, Merge Requests, CI / CD, Operations, Registry, Packages, Wiki (which is selected), Snippets, and Settings.

The main content area shows the following sections:

- Sprint 1 retrospective**: Last edited by Victor Hernandez 1 minute ago. Includes buttons for New page, Page history, and Edit.
- Purpose**: Reflect on what happened during the Sprint and identify actions for improvement going forward.
- Before we begin...**: Review as needed
 - Team Working Agreement
 - Timebox
 - Sprint Objective
 - Agenda items
- What Went Well**: Keep it positive, record observations, findings, successes, and ideas as a team.
 - 1.
 - 2.
 - 3.
- What Needs Improvement**: Surprises, obstacles, learning opportunities - actions and outcomes.
 - 1.
 - 2.

On the right side, there is a sidebar with a Clone repository button and a list of More Pages.



- GitLab Plan Phase of the DevOps Lifecycle
 - <https://about.gitlab.com/stages-devops-lifecycle/plan/>
- How GitLab's **Marketing Team** uses GitLab for Project Management
 - <https://about.gitlab.com/handbook/marketing/#-marketing-project-management-guidelines>
- How GitLab's **Quality Team** uses GitLab for Project Management
 - <https://about.gitlab.com/handbook/engineering/quality/project-management/>
- **GitLab Agile Planning**
 - <https://about.gitlab.com/solutions/agile-delivery/>
- SAFe and Agile Planning with GitLab
 - <https://youtu.be/PmFFlTH2DQk>



GitLab

Part 1: Lab Setup



Go to:

<https://gitlab.com/jvaleriano/devops-automation-workshop-resources>

Download the labs to follow along!

Gitlab Demo Login / Registration



Have an invitation code?

GitLab Demo Portal

Sign In for GitLab Team Members

Login with OKTA

New to the GitLab Demo Portal?

Have an invitation code from an event or training class? You can redeem the invitation code to create an account. Your invitation code is in the course materials with the hands-on lab instructions or it will be provided by your instructor.

If you have already redeemed your invitation code, you can use your credentials to sign in to the GitLab instance directly.

Please [email us](#) if you need assistance with your invitation code.

Redeem Invitation Code

Redemption Code:

50d13a3c

Registering



Already redeemed your invitation code?

You can sign in using the credentials that were generated for you.

Redeem an Invitation Code

Invitation Code

50d13a3c

Redeem and Create Account

Redemption Code: **50d13a3c**



Your GitLab Credentials

Please download your credentials. You will not be able to access your username and password after you navigate away from this page.

We have provisioned a user account and organizational group on our GitLab instance that you can configure with your own subgroups or projects. These credentials are only valid on our instance and will not work on gitlab.com.

GitLab URL: <https://workshop.gitlabtraining.cloud>

Username: iugyed6r

Password: [REDACTED]

2. Go to group

1. Download credentials



Download Credentials

GitLab Dashboard

My Group

Enter the Username and Password



GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

Username or email

Password

Remember me [Forgot your password?](#)

Sign in

Forgot the Username and Password?

1. The info is in the file you downloaded when created
2. Create a new registration (it's okay)

You Should Now See this Page

The screenshot shows the GitLab web interface. At the top, there's a navigation bar with the GitLab logo, a 'Menu' button, a search bar labeled 'Search GitLab', and various user icons. Below the header, the URL 'Training Users > ... > My Test Group - iugyed6r' is visible. The main content area displays a group page for 'My Test Group - iugyed6r'. The group ID is 1049, and there's a 'Leave group' link. A red oval highlights the group name. On the left, a sidebar lists various GitLab features: Subgroup information, Epics (0), Issues (0), Merge requests (0), Security & Compliance, Push Rules, Kubernetes, Packages & Registries, Analytics, Wiki, and Settings. The main content area includes sections for Recent activity (Last 90 days: 0 Merge Requests opened, 0 Issues opened), Members added (2), and tabs for Subgroups and projects, Shared projects, and Archived projects. There are also search bars for 'Search by name' and 'Name'. Below this, a large text block states: 'A group is a collection of several projects.' followed by two descriptive paragraphs and an illustration of a shield containing a GitLab logo and various icons.

A group is a collection of several projects.

If you organize your projects under a group, it works like a folder.

You can manage your group member's permissions and access to each project in the group.

The illustration shows a shield-shaped frame with a dashed border. Inside the shield, there's a central GitLab logo (a stylized cat head). Surrounding the logo are several icons: a blue circle with a white question mark, a green circle with a white checkmark, a pink circle with a white exclamation mark, a purple line, and a red circle with a white 'X'. The shield has a light gray background and a dark gray outline.



GitLab

Plan Stage

Project Management - Lab 1

Lab Assignment

Part 1

Scenario: Create a project structure and views

Key Tasks to Complete

1. Create a Group
2. Create a Project
3. Create Project Milestone
4. Create at least 3 labels
5. Create at least 2 project issues.
6. Create New Board
7. Modify board scoping

Create subgroup



From your workshop group:

1. Select the 'New subgroup' button.

Training Users > ... > My Test Group - iugyed6r



M My Test Group - iugyed6r  Group ID: 1049 [Leave group](#)

Recent activity Merge Requests opened Issues opened Members added
Last 90 days 0 0 2

Subgroups and projects Shared projects Archived projects

Search by name Name

Create subgroup cont.



From the Create group page:

- Select the 'Create group' panel.

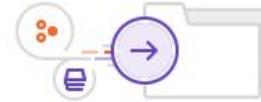


Create new group



Create group

Assemble related projects together and grant members access to several projects at once.



Import group

Export groups with all their related data and move to a new GitLab instance.

Enter subgroup details



From the Create group tab:

- Enter desired subgroup name in the '**Group name**' field.
- Select the '**Create group**' button.

FYI:

- Do not edit Group URL field.

The screenshot shows the 'Create group' interface. A red arrow points to the 'Group name' input field, which contains 'JV workshop prep'. Another red arrow points to the 'Create group' button at the bottom left. The 'Group URL' field is pre-filled with a long URL, and the 'Visibility level' section shows 'Private' selected. The 'Invite Members (optional)' section has an email address 'member1@company.com' entered in the 'Email 1' field.

New group > **Create group**

Group name

JV workshop prep

Group URL

<https://workshop.gitlabtraining.cloud/training-users/session-50d13a3c/ugyed6r/> jv-

Visibility level

Who will be able to see this group? [View the documentation](#)

Private
The group and its projects can only be viewed by members.

Invite Members (optional)

Invited users will be added with developer level permissions. [View the documentation](#) to see how to change this later.

Email 1

member1@company.com

+ Invite another member

Create group Cancel

Create project



From your workshop subgroup:

1. Select the 'New project' button.

Training Users > ... > JV workshop prep

ⓘ Group 'JV workshop prep' was successfully created.



JV workshop prep

Group ID: 1050 [Leave group](#)



New subgroup

New project

Recent activity

Last 90 days

Merge Requests opened

0

Issues opened

0

Members added

1

[Subgroups and projects](#)

[Shared projects](#)

[Archived projects](#)

Search by name

Name



Create blank project



In the ‘Create new project’ page:

1. Select ‘Create blank project’ quadrant.

Create new project



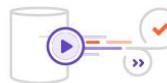
Create blank project
Create a blank project to house your files, plan your work, and collaborate on code, among other things.



Create from template
Create a project pre-populated with the necessary files to get you started quickly.



Import project
Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.



Run CI/CD for external repository
Connect your external repository to GitLab CI/CD.

You can also create a project from the command line. [Show command](#)

Enter project details



In the Create blank project page:

1. Enter desired project name in the '**Project name**' field.
2. Enter a brief project description in the '**Project description (optional)**' field.
3. Select the **Initialize repository with README** checkbox so it is enabled.
4. Select the '**Create project**' button.

New project > Create blank project

Project name 

Project URL Project slug

Project description (optional)
This project is for a workshop 

Visibility Level 
 Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
 Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project **Cancel**

FYI:

- Do not edit Project slug field.



Access project milestones



In the project repository:

1. Select 'Issues→Milestones'.

The screenshot shows the GitLab interface for a project named 'JV workshop project'. The left sidebar lists various project management sections: Project information, Repository, Issues (0), Merge requests (0), Requirements, CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, and Packages & Registries. A red arrow points from the 'Issues' section to the 'Milestones' option in a dropdown menu that appears when the 'Issues' section is selected. The dropdown menu also includes options for List, Boards, Service Desk, Iterations, and main.

Create new project milestone A



In the project Milestone view:

1. No Milestones will be listed.
2. Select the '**New milestone**' button.

The screenshot shows a user interface for managing project milestones. At the top, there is a search bar labeled "Filter by milestone name". Next to it are two buttons: "Due soon" and a blue rectangular button labeled "New milestone". A thick red arrow is drawn to the right of the "New milestone" button, highlighting it as the action to take.

In the New Milestone page:

1. Enter a milestone title into the '**Title**' field.

The screenshot shows a "New Milestone" form. It contains two main input fields: "Title" and "Start Date". The "Title" field is filled with the text "Milestone A". The "Start Date" field is filled with the date "2021-10-13". A red arrow points to the "Title" field, indicating where the user should enter the milestone title.

New Milestone	
Title	Milestone A
Start Date	2021-10-13

Enter project milestone A start and due dates



In the New Milestone page:

1. Enter a brief milestone description into the '**Description**' field.

Enter a Start Date.

1. Select the '**Start Date**' field.
2. Select **today's date**.

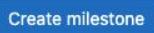
Enter a Due Date.

1. Select the '**Due Date**' field.
2. Select the date **six days from the Start Date**.

Create the Milestone

1. Select the '**Create milestone**' button.

New Milestone

Title	Milestone A
 Start Date	2021-10-13  Due Date 2021-10-19
 Description	<p>Milestone A is the first milestone created</p> <p>Write Preview </p> <p>Markdown is supported</p>
 Create milestone 	

Project milestone A details



In the Milestone details:

1. Select 'Milestones'.



Training Users > ... > JV workshop prep > JV workshop project > Milestones > **Milestone A**

Upcoming **Milestone** Oct 13, 2021–Oct 19, 2021 Edit Promote Close milestone Delete

Milestone A

Milestone A is the first milestone created

Filter by Issues Issue weight

Burndown chart

A burndown chart showing a single vertical line starting at the top and ending at the bottom, labeled with the number '1' at both ends. This indicates that all work is completed by the end of the milestone period.

Day	Issues Remaining
Start	1
End	1

Burnup chart

A burnup chart showing a single vertical line starting at the bottom and ending at the top, labeled with the number '1' at both ends. This indicates that one new issue was added at the beginning of the milestone period.

Day	Issues Added
Start	0
End	1

Create new project milestone



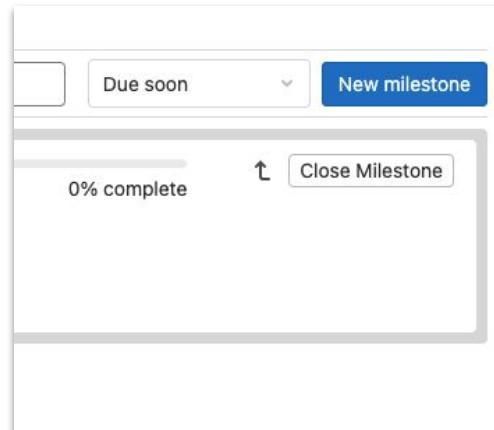
In the project Milestone view:

1. Milestone A will be listed.

A screenshot of the Jira Milestone view. At the top, there are filters: 'Open 1', 'Closed 0', and 'All 1'. On the right, there is a 'Filter by milestone' button and a status bar showing '0 Issues · 0 Me'. Below the filters, a card for 'Milestone A' is displayed. The card shows the dates 'Oct 13, 2021–Oct 20, 2021' and a status indicator 'Upcoming'. Below the date range, there is a summary: 'Training Users / Session 50d13a3c / My Test Group - iugyed6r / JV workshop prep / JV workshop project'.

Create a new Milestone:

1. Select the 'New milestone' button.



Repeat: Create a project milestone



→ Title Milestone B

→ Start Date 2021-10-20 Clear start date → Due Date 2021-10-26

→ Description Write Preview B I '' </>

Milestone B is the second milestone created

Markdown is supported

→ Create milestone Cancel

Project milestone B details



Training Users > ... > JV workshop prep > JV workshop project > Milestones > **Milestone B**

Upcoming

Milestone Oct 20, 2021–Oct 26, 2021

Edit

Promote

Close milestone

Delete

Milestone B

Milestone B is the second milestone created

Filter by

Issues

Issue weight

Burndown chart



Burnup chart



Access project labels



In the project repository:

1. Select 'Project information→Labels'.

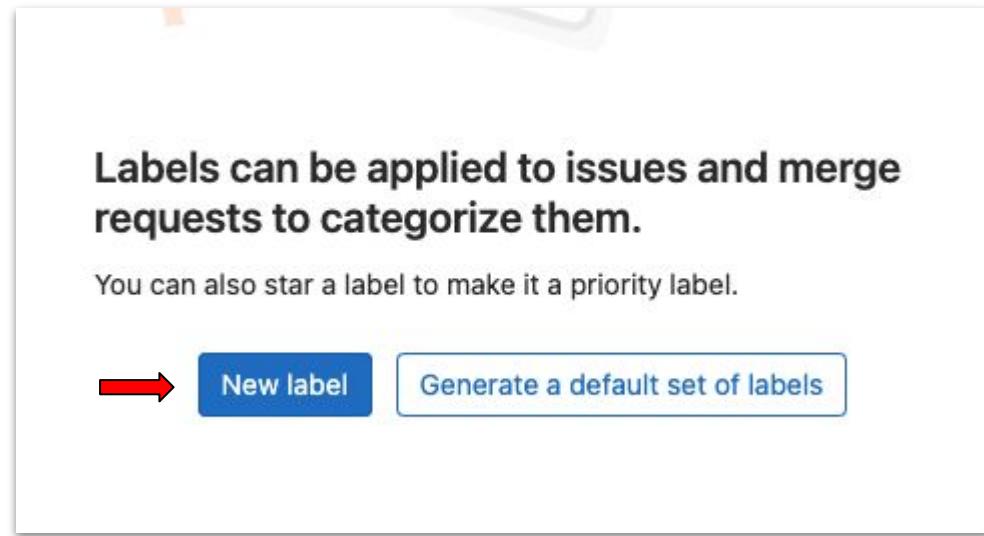
The screenshot shows the Jira interface for a project titled "JV workshop project". On the left, there's a sidebar with options like "Project information", "Repository", "Issues", "List", "Boards", and "Service Desk". A red arrow points from the text above to the "Project information" option. On the right, a context menu is open over the "Issues" button, listing "Activity", "Labels", and "Members". Another red arrow points from the text above to the "Labels" option in the menu. At the bottom, there are buttons for "Filter by", "Issues" (which is highlighted in blue), and "Issue weight".

Create new scoped project label



In the project Labels view:

1. No Labels will be listed.
2. Select the 'New label' button.



A screenshot of a software interface titled "Labels can be applied to issues and merge requests to categorize them." Below this text, it says "You can also star a label to make it a priority label." At the bottom, there are two buttons: a blue "New label" button with a red arrow pointing to it, and a white "Generate a default set of labels" button.

Labels can be applied to issues and merge requests to categorize them.

You can also star a label to make it a priority label.

New label

Generate a default set of labels

Enter new scoped project label title



In the project Labels view:

1. Enter a scoped label title in the '**Title**' field.
2. For example, enter '**'OS::Mac'**'.

FYI, scoped labels are formatted as
<value>::<value>

In the project Labels view:

1. Enter a brief scoped label description in the '**Description**' field.
2. Select a background color from the color options listed.
3. Select the '**Create label**' button.

New Label

Using :: denotes a [scoped](#)

Title	OS::Mac
Description	Scoped Mac
Background color	#6699cc

Choose any color. You can choose one of the following colors:

- Green-cyan
- Dark slate blue
- Light green
- Forest green
- Dark green
- Black

Create label **Cancel**

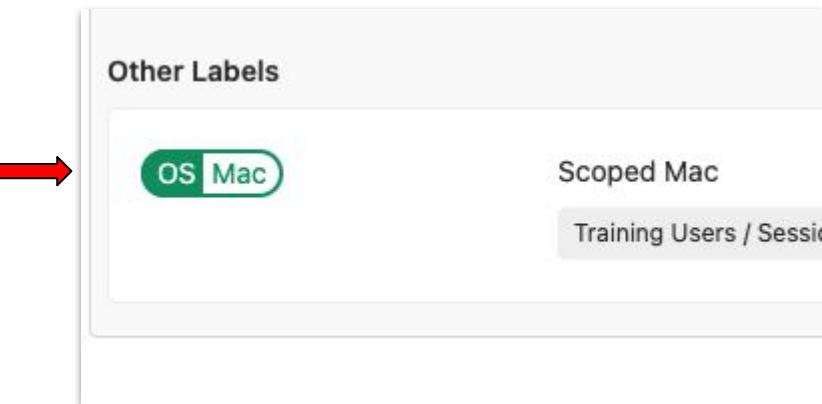
The screenshot shows a 'New Label' dialog box. The 'Title' field contains 'OS::Mac'. The 'Description' field contains 'Scoped Mac'. The 'Background color' field shows a blue square with the hex code '#6699cc'. A color picker dropdown is open, displaying a color palette with various shades of green and blue. One color, 'Green-cyan', is highlighted with a black box and a tooltip. The 'Create label' button is at the bottom left, and a 'Cancel' button is at the bottom right. Red arrows point from the text descriptions to each of the three main input fields: Title, Description, and Background color.

Project labels list



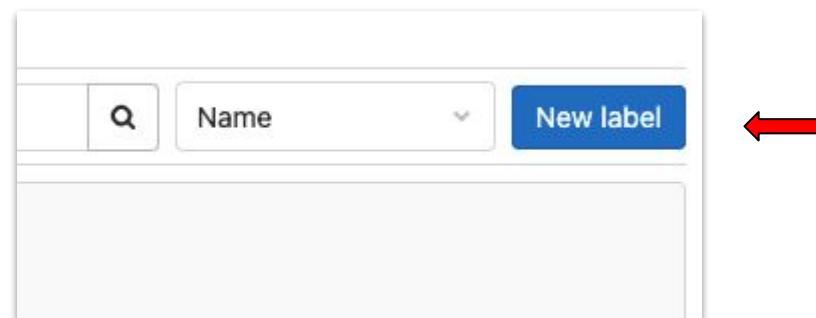
In the project Labels list:

1. Confirm 'OS::Mac' label is listed.



In the project Labels list:

1. Select the 'New label' button.



Enter another scoped project label title



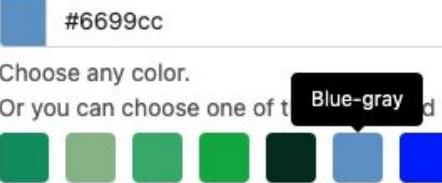
In the project Labels view:

1. Enter a scoped label title in the '**Title**' field.
2. For example, enter '**OS::Linux**'.
3. Enter a brief scoped label description in the '**Description**' field.
4. Select a background color from the color options listed.
5. Select the '**Create label**' button.

New Label

→ Title OS::Linux
Using :: denotes a scoped label set

→ Description Scoped Linux

→ Background color #6699cc
Choose any color.
Or you can choose one of the following colors:


→ Create label Cancel

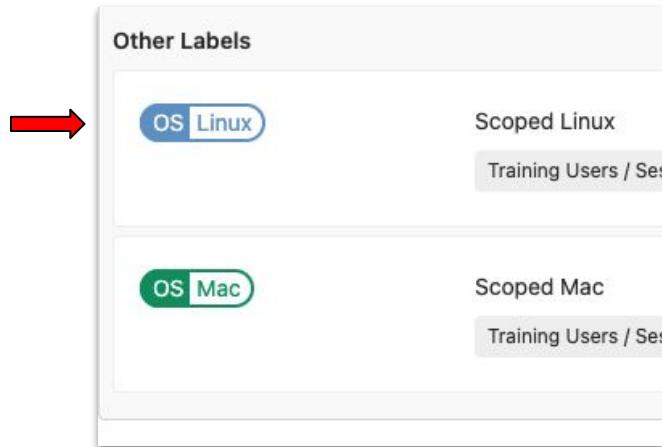


Project labels list



In the project Labels list:

1. Confirm 'OS::Linux' label is listed.



In the project Labels list:

1. Select the 'New label' button.



Enter new non-scoped project label title



In the project Labels view:

1. Enter a non-scoped label title in the '**Title**' field.
2. For example, enter '**Bugs**'.
3. Enter a brief label description in the '**Description**' field.
4. Select a background color from the color option listed.
5. Select the '**Create label**' button.

New Label

→ Title Use :: to create a scoped label set (eg. pri

→ Description

→ Background color #0000ff
Choose any color.
Or you can choose one of the suggested colors:

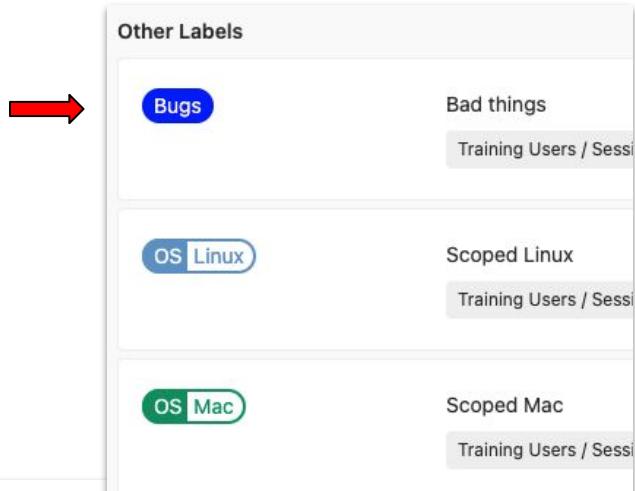

→ Create label Cancel

Project labels list



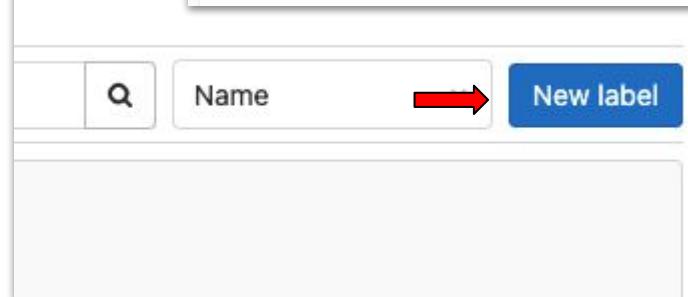
In the project Labels list:

1. Confirm ‘Bugs’ label is listed.



In the project Labels list:

1. Select the ‘New label’ button.



Enter new non-scoped project label title



In the project Labels view:

1. Enter a non-scoped label title in the '**Title**' field.
2. For example, enter '**Feature**'.
3. Enter a brief label description in the '**Description**' field.
4. Select a background color from the color option listed.
5. Select the '**Create label**' button.

New Label

→ Title Feature

Use :: to create a scoped label set (eg. priority::1)

→ Description Good things

→ Background color #e6e6fa

Choose any color.
Or you can choose one of the suggested colors.

Lavendar

→ Create label Cancel

Project labels list



In the project Labels list:

1. Confirm ‘Feature’ label is listed.

A screenshot of a software interface showing a list of labels. The top section is titled 'Other Labels' and contains three items: 'Bugs' (blue button), 'Bad things' (grey button), and 'Training Users / Ses...'. Below this is another section with three items: 'Feature' (purple button), 'Good things' (grey button), and 'Training Users / Ses...'. A red arrow points from the text above to the 'Feature' label in the second section.

In the project Labels list:

1. Select the ‘Issue’ panel.

A screenshot of a software interface showing a sidebar menu. The menu items are: 'Project information' (disabled), 'Activity' (disabled), 'Labels' (selected, highlighted in grey), 'Members' (disabled), 'Repository' (disabled), 'Issues' (selected, highlighted in grey), 'Merge requests' (disabled), 'Requirements' (disabled), and 'CI/CD' (disabled). To the right of the menu is a panel titled 'Prioritized Labels' containing 'List', 'Boards', 'Service Desk', and 'Milestones'. A red arrow points from the text above to the 'Issues' menu item.



In the project Labels view:

1. No issues will be listed.
2. Select the '**New issue**' button.

The Issue Tracker is the place to add things that need to be improved or solved in a project

Issues can be bugs, tasks or ideas to be discussed. Also, issues are searchable and filterable.



New issue

Import issues ▾

Enter issue metadata/create issue



In the project New issue view:

1. Enter an issue title in the '**Title**' field.
2. Enter an issue description in the '**Description**' field.
3. Select the '**Assign to me**' hyperlink.
4. Select the '**Milestone**' drop-down→<any milestone>.
5. Select the '**Labels**' drop-down→<any label>.
6. Select the '**Submit issue**' button.

The screenshot shows the 'New Issue' dialog in Jira. A red arrow points to the 'Title' field, which contains 'Issue A'. Another red arrow points to the 'Description' field, which contains 'Feature request'. A third red arrow points to the 'Assignees' field, which contains 'Invited User iugedy6r'. A fourth red arrow points to the 'Epic' field, which contains 'Select epic'. A fifth red arrow points to the 'Milestone' field, which contains 'Milestone A'. A sixth red arrow points to the 'Labels' field, which contains 'OS::Linux'. A final red arrow points to the 'Create issue' button at the bottom left of the dialog.

Issue details



Issue A

Feature request

Drag your designs here or [click to upload](#).

Linked issues 0 +



Oldest first

Show all activity

Create merge request



Invited User iugyed6r @iugyed6r changed milestone to %Milestone A just now



Invited User iugyed6r @iugyed6r added OS Linux scoped label just now



Invited User iugyed6r @iugyed6r assigned to @iugyed6r just now

Write Preview



Write a comment or drag your files here...



@iugyed6r

Epic

None

Milestone

Milestone A

Iteration

None

Time tracking

No estimate or time spe

Due date

None

Labels

OS Linux

Weight

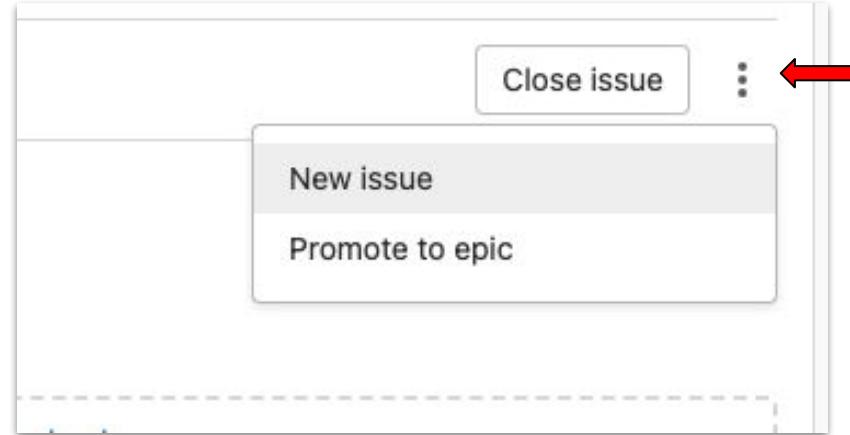
None

Create issue



In the project issue view:

1. Select the 
2. Select 'New issue' option.



Repeat: enter issue details

In the project New issue view:

1. Enter an issue title in the '**Title**' field.
2. Enter an issue description in the '**Description**' field.
3. Select the '**Milestone**' drop-down→<any milestone>.
4. Select the '**Labels**' drop-down→'<scoped label>' and '<non-scoped label>' .
5. Select the '**Submit issue**' button.

The screenshot shows the 'New issue' form in Jira. The 'Title' field is populated with 'Issue B'. The 'Description' field contains the text 'There's a bug in the code'. The 'Milestone' dropdown is set to 'Milestone B'. The 'Labels' dropdown is set to 'Bugs +1 more'. A red arrow points to each of these four fields. In the top right corner of the slide, there is a small orange decorative icon.

Issue B
Add description templates to help your contributors communic

Similar issues Issue A
#1 • 2 minutes ago by [user]

Type Issue

Description Write Preview
There's a bug in the code

Markdown and quick actions are supported

This issue is confidential and should only be visible to team

Assignees Unassigned Assign to me

Epic Select epic

Milestone Milestone B

Labels Bugs +1 more

Finalize the issue creation



In the project New issue view:

1. Select the 'Create issue' button.



A screenshot of the 'New issue' creation dialog box. The dialog has four input fields: 'Assignees' (set to 'Unassigned'), 'Epic' (set to 'Select epic'), 'Milestone' (set to 'Milestone B'), and 'Labels' (set to 'Bugs +1 more'). At the bottom of the dialog are two buttons: a blue 'Create issue' button and a white 'Cancel' button.

New issue details



Issue B ←

There's a bug in the code ←

↑ Drag your designs here or [click to upload.](#)

Linked issues 0 +



Oldest first ▾

Show all activity ▾

Create merge request ▾



Invited User iugyed6r @iugyed6r changed milestone to %Milestone B just now



Invited User iugyed6r @iugyed6r added OS Mac scoped label just now



Invited User iugyed6r @iugyed6r added Bugs label just now →

Epic
None

Milestone
Milestone B ←

Iteration
None

Time tracking
No estimate or time spent

Due date
None

Labels

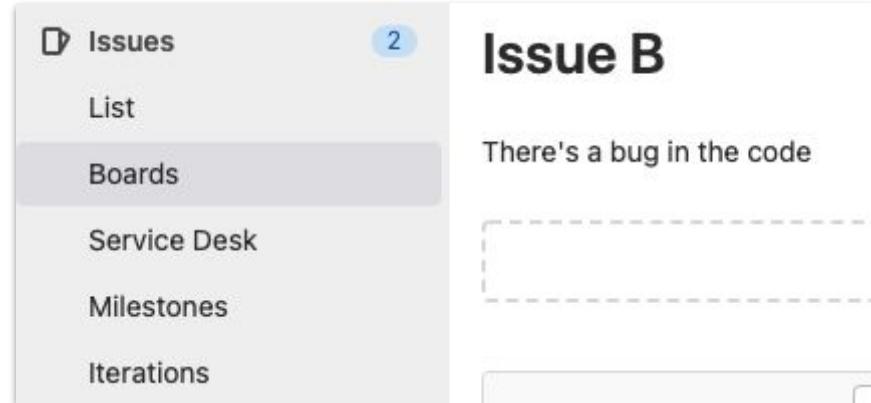
Bugs X OS Mac X

Access project issue boards



In the project repository:

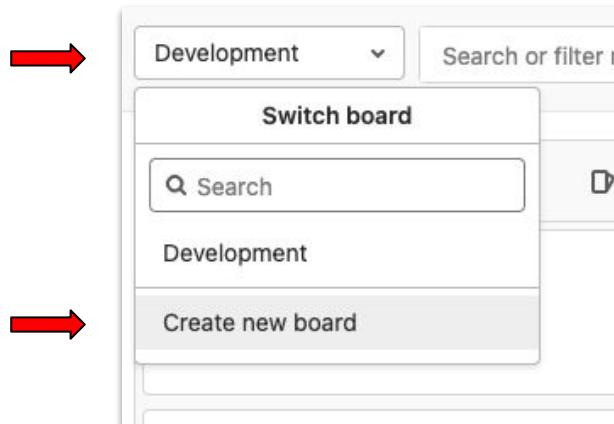
1. Select 'Issues→Boards'.



The image shows a screenshot of a software interface for managing issues. On the left, there is a sidebar with the following options: 'Issues' (with a blue circular badge showing '2'), 'List', 'Boards' (which is highlighted with a grey background), 'Service Desk', 'Milestones', and 'Iterations'. To the right of the sidebar, the main area is titled 'Issue B' and contains the text 'There's a bug in the code'. Below this text is a dashed-line box, likely representing a comment or a section of the issue.

In the project Issue Board view:

1. Select the 'Switch board' menu→ 'Create new board' option.



The image shows a screenshot of a 'Switch board' interface. At the top, there is a dropdown menu set to 'Development' and a search bar with the placeholder 'Search or filter re...'. Below this is a section titled 'Switch board' containing a search bar with the placeholder 'Search'. Further down, there are two main options: 'Development' and 'Create new board', each represented by a button-like element.

Name new project issue board



In the Create new board view:

1. Enter a board title into the 'Title' field.
2. Select the 'Create board' button.



Create new board ×

Title

List options
Configure which lists are shown for anyone who visits this board

Show the Open list
 Show the Closed list

Scope Expand

Board scope affects which issues are displayed for anyone who visits this board

Cancel Create board



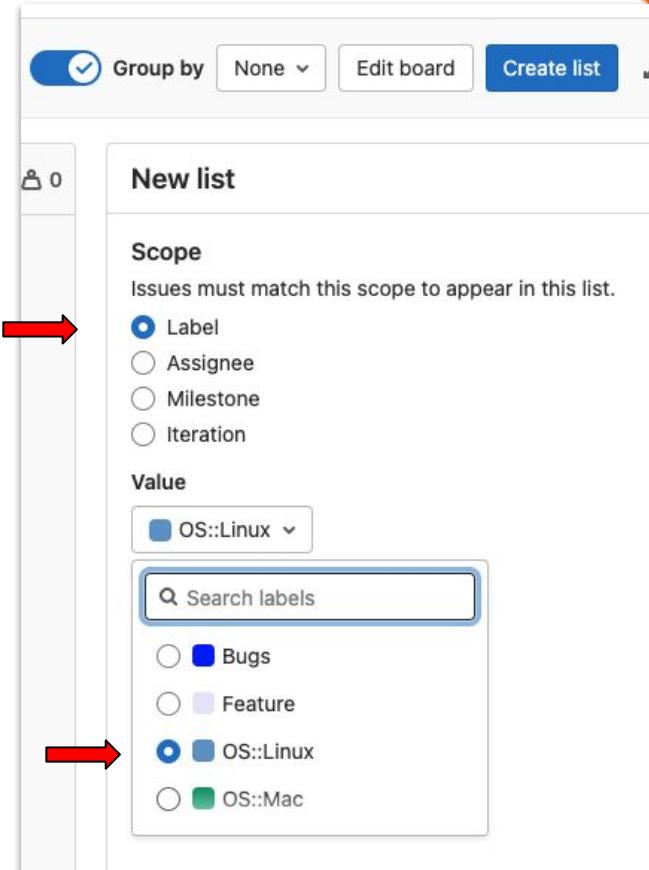
Modify issue board scope

In the project Issue Board view:

1. Select the ‘Create list’ menu→ ‘Label’ radio button.
2. Select ‘Value’ drop-down.
3. Select a ‘<scoped label>’.
4. Select the ‘Add to board’ button.

In the project Issue Board view:

5. Select the ‘Create list’ menu→ ‘Label’ radio button.
6. Select ‘Value’ drop-down.
7. Select the ‘<other scoped label>’.
8. Select the ‘Add to board’ button.



Move board columns



In the project Issue Board view:

1. Select and hold a ‘<scoped column>’ and move it before/after the other “<scoped label>” column.
2. Example: ‘OS::Mac’ before ‘OS::Linux’ column.

The screenshot shows a Jira Issue Board titled "WiP Board". At the top, there is a dropdown menu set to "WiP Board", a search bar with placeholder text "Search or filter results...", and a "Show labels" toggle switch which is turned on (indicated by a checked box). Below the header, there are two columns: "OS Mac" and "OS Linux". The "OS Mac" column contains one issue labeled "Issue B" with a "Bugs" label and ID "#2". The "OS Linux" column contains one issue labeled "Issue A" with ID "#1". Between the two columns is a horizontal separator bar with a gear icon. A large red arrow is drawn over this gear icon, pointing from the "OS Mac" side towards the "OS Linux" side, illustrating the process of moving the "OS Mac" column before the "OS Linux" column.

Move issue within board



In the project Issue Board view:

1. Select and hold '**<an issue>**' and drag into the other '**<scoped label>**' column.
2. Example: '**Issue A**' into '**OS::Linux**' column.

The screenshot shows a Jira Software Issue Board view titled 'WiP Board'. There are two columns: 'OS::Mac' and 'OS::Linux'. In the 'OS::Mac' column, there is one issue labeled 'Issue B' with the label 'Bugs' and the ID '#2'. In the 'OS::Linux' column, there is one issue labeled 'Issue A' with the ID '#1'. A red arrow points from the text 'Issue A' in the second list item of the instructions towards the 'OS::Linux' column header. The top right corner of the interface has a 'Show labels' toggle switch that is turned on.

Modify scope of existing issue board

In the project Issue Board view:

1. Select 'Edit board' button.

The screenshot shows a Jira Issue Board view. At the top, there are buttons for 'Show labels' (checked), 'Group by' (set to 'None'), and 'Edit board'. Below this, there are two issues: 'Issue A' and 'Issue B'. Issue A is categorized under 'OS Linux' and has a status of 2 and 0 assignees. Issue B is categorized under 'Bugs' and also has a status of 2. A red arrow points to the 'Edit board' button in the top right corner of the interface.

Scope existing issue board to milestone



In the Edit board view:

1. Select '**Edit**' option for the '**Milestone**' pane.
2. Select '**Any milestone**' option.
3. Select the '**Save changes**' button.

Scope
Board scope affects which issues are displayed for anyone who visits this board

Milestone Edit

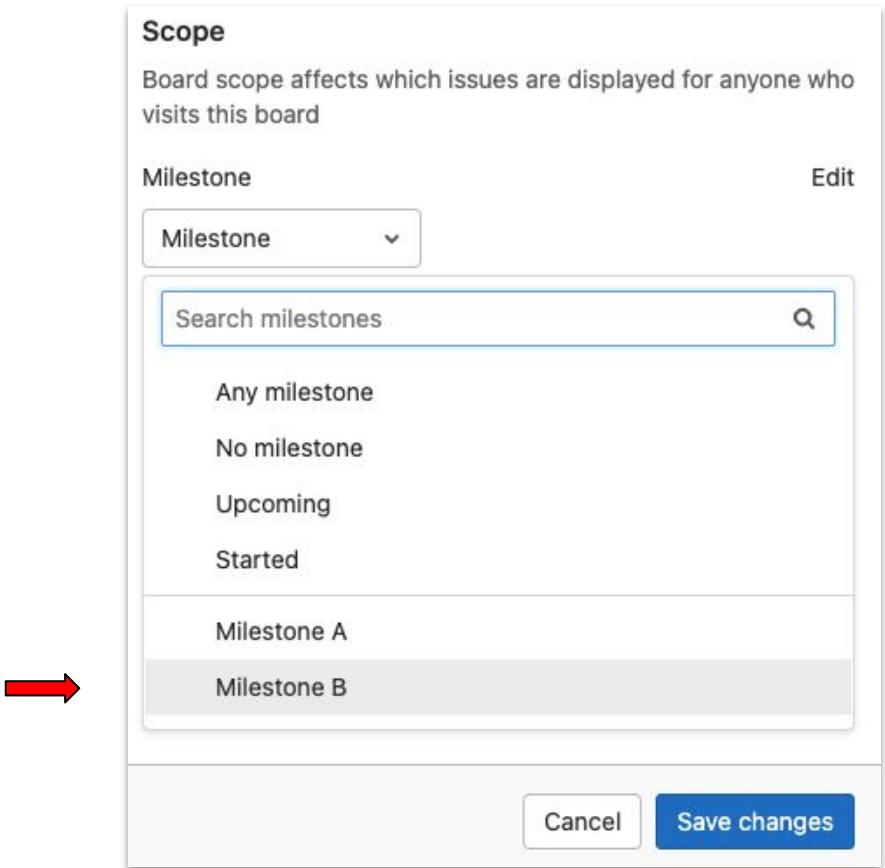
Milestone Milestone ▼

Search milestones Q

Any milestone
No milestone
Upcoming
Started

Milestone A
Milestone B

Cancel Save changes



Confirm scoping change to existing issue board



In the Issue Boards view:

1. Issue(s) matching selected milestone is/are displayed.

The screenshot shows the Jira Issue Boards interface. There are two boards visible:

- OS Mac**: This board has 0 issues and 0 attachments. It includes a '+' button and a gear icon for settings.
- OS Linux**: This board has 1 issue and 0 attachments. It includes a '+' button and a gear icon for settings.

A red arrow points to the 'Issues' section of the OS Linux board. Inside this section, there is one issue card labeled "Issue B" with the status "Bugs" and the ID "#2".



GitLab

Thank you



GitLab

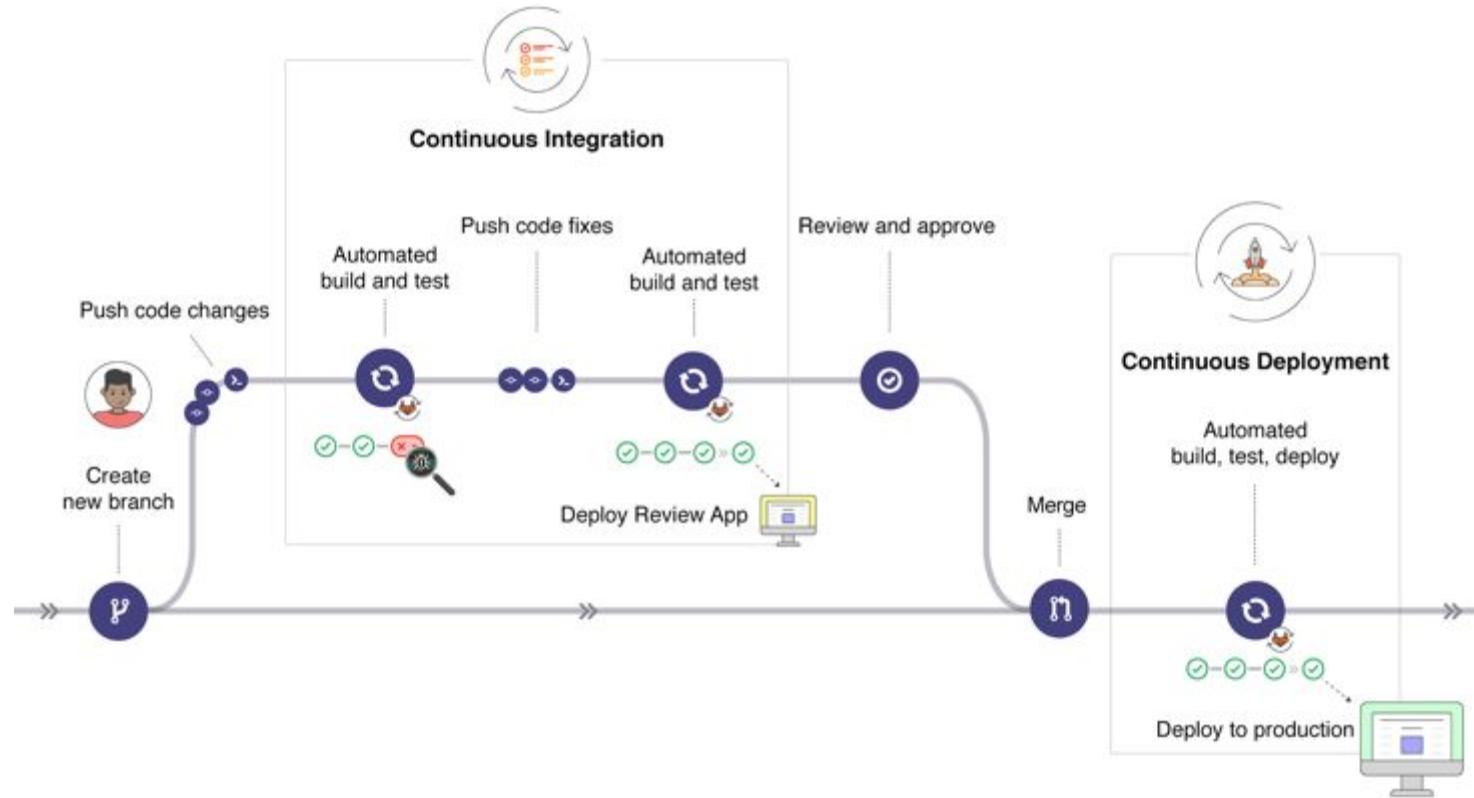
GitLab Basics for Developers

A Complete DevOps Platform Delivered as a Single Application



Single Conversation
Single Data Store
Single Permission Model
Single Interface
Governance & Security
Team Collaboration
Lifecycle Analytics

DevOps Best Practices Built In



Where do I start? - Issue Board



GitLab.org > Issue Boards

Label: -group:fulfillment X

Show labels Edit board *

Open D 1 ▲ 6 +

Backend D 9 ▲ 27 +

frontend D 5 ▲ 22 +

UX D 2 ▲ 6 +

For Seatlink / # of Seats that are in use for a Subscription, can we stamp that back to the Zuora Subscription.

CSAT-Priority 1 P1 GitLab Foresystem P1
drives growth group fulfillment portal integration zuora

gitlab-org/customers-gitlab-com#1147 ▲ 5

Aggregate user usage data in the instance and provide a manual export option

Accepting merge requests Deliverable P1
drives growth customer group fulfillment gitlab.org/gitlab#213618

gitlab-org/customers-gitlab-com#1147 ▲ 5

Implementation: Self-Service Upgrades for Self-Managed

GitLab.com Growth-deliverable DRR FY20 Q2 UX
auto updated Backend customer drives growth front end group fulfillment vintage workflow validation backlog

gitlab-org/customers-gitlab-com#107 ▲ 5

GitLab.com users should be able to purchase additional users without adding members

CSAT-Priority 1 P1 GitLab Foresystem
Growth-deliverable DRR FY20 Q2 P2
drives growth customer group fulfillment portal integration

gitlab-org/customers-gitlab-com#1098 ▲ 3

API call to pull user/seat data from GitLab.com

Growth-deliverable P2 customer
drives growth gitlab.com group fulfillment parent issue raised 12.8 retention

gitlab-org/gitlab#35454 ▲ 3

Create Dynamic "Seats Currently in Use Members List" for GitLab.com

Accepting merge requests Growth-deliverable P2
customer customer drives growth front end gitlab.com group fulfillment internal customer parent issue workflow design

gitlab.org/gitlab#27074 ▲ 8

Wrong text color for .alert-info

P2 S4 bug drives growth group fulfillment regression

gitlab-org/customers-gitlab-com#946 ▲ 3

Create Dynamic "Seats Currently in Use Members List" for GitLab.com

Accepting merge requests Growth-deliverable P2
customer customer drives growth front end gitlab.com group fulfillment internal customer parent issue workflow design

gitlab-org/gitlab#27074 ▲ 8

Pajamas Update: Portal "Manage Purchases/Upgrade Subscription" Page

Customers Portal Pajamas Update For Scheduling P1
drives growth group fulfillment

gitlab-org/customers-gitlab-com#938 ▲ 5

Review Issue



GitLab.org > customers-gitlab.com > Issues > #948

Open Opened 3 months ago by Ammar Alakkad

Close issue **New issue**

Wrong text color for .alert-info

Summary

Wrong text color for .alert-info , it's showing white text instead of #0c5460 . I believe it's related to scss imports orders.

Steps to reproduce

1. Login to the customers portal with an account that doesn't have any invoice.
2. Go to the invoices page.
3. Notice the alert that says "You do not currently have any invoices."

What is the current bug behavior?

White text on a light background, barely readable.

What is the expected correct behavior?

Dark text (#0c5460) on a light background.

Relevant logs and/or screenshots

GitLab Contact Us My Account Manage Purchases Payment History Payment Methods FAQ Sign out

View Invoices

You do not currently have any invoices.

Linked issues 0 □ 0 +

Oldest first ▾ Show all activity ▾ Create merge request ▾

Discussion 2 Design 0

See Progress using Merge Requests



Open Opened just now by Josh Swann Edit Close merge request

WIP: Resolve "Put links back to the homepage from the greetings page"

Overview 0 Commits 0 Changes

Closes #18

Request to merge 18-put-links-back-to-the-homepage-from-the-greetings-page into master

Open in Web IDE Check out branch

Merge requests are a place to propose changes you have made to a project and discuss those changes with others.

Interested parties can even contribute by pushing commits if they want to.

Currently there are no changes in this merge request's source branch. Please push new commits or use a different branch.

Create file

0 0 0

Show all activity

Josh Swann @joshswann changed milestone to Version 2.0 just now

Josh Swann @joshswann added to do label just now

Write Preview

Write a comment or drag your files here...

Markdown and quick actions are supported

Attach a file

Comment Close merge request

- Merge Request to Issue Traceability
- Web IDE or Check Out locally
- Branch Creation
- Record of changes and their impact to the branch

Built-in IDE - Code Changes



The screenshot shows the GitLab IDE interface. On the left is a sidebar with project files: .gitlab-ci.yml, Dockerfile, README.md, app.json, ci_settings.xml, and pom.xml. The main area shows a code editor with GreetingController.java open. The code defines a controller with two methods: one for GET /greeting and one for POST /greeting. A conflict marker is visible in the code editor.

```
package hello;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
```

```
@Controller
public class GreetingController {
    @GetMapping("/greeting")
    public String greeting(Model model) {
        model.addAttribute("greeting", new Greeting());
        return "greeting";
    }
    @PostMapping("/greeting")
    public String greetingSubmit(@ModelAttribute Greeting greeting) {
        return "result";
    }
}
```

At the bottom, there are buttons for 'Commit...' and '0 changed files'.

Check out, review, and merge locally

Step 1. Fetch and check out the branch for this merge request

```
git fetch origin
git checkout -b "32-investigate-vulnerability-hardcoded-constant-database-password" "origin"
```

Step 2. Review the changes locally

Step 3. Merge the branch and fix any conflicts that come up

```
git fetch origin
git checkout "origin/master"
git merge --no-ff "32-investigate-vulnerability-hardcoded-constant-database-password"
```

Step 4. Push the result of the merge to GitLab

```
git push origin "master"
```

Tip: You can also checkout merge requests locally by following these guidelines.

Traceability



WIP: Resolve "Investigate vulnerability: Hardcoded constant database password"

Overview 0 · Commits 1 · Pipelines 1 · Changes 2

Closes #32

The screenshot shows a GitHub pull request interface for a merge request titled "Request to merge 32-investigate-vuln... into master". The top bar includes links for "Open in Web IDE" and "Check out branch". A red box highlights the status message "Pipeline #131563038 passed for 414431ca on 32-investigate-vuln...". Below this, another red box highlights the "Approve" button and the note "Requires approval." A third red box highlights the "Scan Results" section, which lists "Code quality improved on 52 points", "Security scanning detected 43 fixed vulnerabilities" (with a note about an out-of-date security report), and "License Compliance detected no new licenses". At the bottom, there are buttons for "Merge", "This is a Work in Progress", "Resolve WIP status", and "Closes #32". A note at the bottom states: "You can merge this merge request manually using the command line".

- CI/CD Pipeline Activity
- Approval Processes
- Scan Results
 - Code Quality
 - Security Scanning
 - Compliance
- Code Reviews
- And more!
 - Commit History
 - Comments
 - Reactions

Code Review



swann-demo > tanuki-two > Merge Requests > 130

Open Opened 3 weeks ago by Josh Swann

Edit Close merge request

Resolve "Investigate vulnerability: Hardcoded constant database password"

Overview 0 Commits 1 Pipelines 1 Changes 2

Compare latest version and master

2 files +0 -284

src/main/java/hello/CodeSmells.java deleted

```
1 - package hello;
2 -
3 - import java.lang.Integer;
4 -
5 - public class CodeSmells{
6 -
7 - //TODO: Need to flesh out these functions more.
8 - //password: IamputtingAPasswordH3R3
9 -
10 -
11 -     public Greeting drive;
12 -
13 -     public void tearDown(){
```

Why? Really? Why?

Markdown is supported

Attach a file

Start a review Add comment now Cancel

Code Review



Overview 1 Commits 1 Pipelines 1 Changes 2 0/1 thread resolved Reply Print Up

Josh Swann @joshswann restored source branch 32-investigate-vulnerability-hardcoded-constant-database-password 3 weeks ago

Josh Swann @joshswann added 1 commit 3 weeks ago

- * 414431ca - Deleted src/main/java/hello/CodeSmells.java, src/main/java/hello/ConstantPasswords.java files

[Compare with previous version](#)

Josh Swann @joshswann unmarked as a Work In Progress 2 weeks ago

Josh Swann @joshswann approved this merge request 2 weeks ago

Josh Swann @joshswann started a thread on the diff 26 seconds ago Toggle thread

[src/main/java/hello/CodeSmells.java deleted](#)

```
1 - package hello;
2 -
3 - import java.lang.Integer;
4 -
5 - public class CodeSmells{
6 -
7 - //TODO: Need to flesh out these functions more.
8 - //password: IamputtingAPasswordH3R3
9 -
10 -
11 -     public Greeting drive;
12 -
13 -     public void tearDown(){
```

Josh Swann @joshswann · 26 seconds ago

Why? Really? Why?

[Reply...](#) [Resolve thread](#)

MR Reviewer



GitLab Merge Request Reviewers easily allow authors to **request a review as well as see the status of the review**. By selecting one or more users from the Reviewers field in the merge request's right-hand sidebar, the **assigned reviewers will receive a notification of the request to review the merge request**.

This makes it easy to determine the relevant roles for the users involved in the merge request, as well as **formally requesting a review from a peer**.



The screenshot shows a GitLab Merge Request interface. At the top, there is a 'To Do' section with a button to 'Add a to do' and a '»' link. Below it is an 'Assignee' section showing Edmund Ochieng (@ochienged) with his profile picture. Further down is a 'Reviewer' section showing Georgi N. Georgiev (@ggeorgiev_) with his profile picture. A vertical scrollbar is visible on the right side of the interface.

Easy Merging



[Merge](#) [Delete source branch](#)

➤ 1 commit and 1 merge commit will be added to master. Modify merge commit

[Closes #32](#)
[Assign yourself to this issue](#)

You can merge this merge request manually using the command line

Merged by  [Josh Swann](#) just now [Revert](#) [Cherry-pick](#)

The changes were merged into [master](#) with [952e47ea](#) 

The source branch has been deleted

Closed [#32](#)

[Assign yourself to this issue](#)

CI/CD Pipelines



All 150	Pending 0	Running 1	Finished 149	Branches	Tags	Run Pipeline	Clear Runner Caches	CI Lint				
Status	Pipeline	Triggerer	Commit	Stages								
running	#135457891 latest		P' master -> b07350bb Merge branch '30-i-wanna-go-f...									
canceled	#134517448 latest		P' 18-put-link... -> b07350bb Merge branch '30-i-wanna-go-f...						⌚ 00:04:59 4 days ago			
passed	#131563038 latest		P' 32-investig... -> 414431ca Deleted src/main/java/hello/Cod...						⌚ 00:16:35 1 week ago			
failed	#131562920		P' 32-investig... -> b07350bb Merge branch '30-i-wanna-go-f...						⌚ 00:13:57 1 week ago			
passed	#131444617 latest		P' 31-add-unit... -> 33bc6e24 Deleted .example.ci.file.yml.not_i...						⌚ 00:15:24 1 week ago			
passed	#131432572 latest		P' 29-is-this... -> 8c0e37bb Deleted .example.ci.file.yml.not_i...						⌚ 00:14:43 1 week ago			
passed	#131431688 latest		P' 31-investig... -> b18f1034 Deleted src/main/java/hello/Cons...						⌚ 00:16:22 1 week ago			
failed	#131431015		P' 31-investig... -> b07350bb Merge branch '30-i-wanna-go-f...						⌚ 00:14:17 1 week ago			

Resolved MR, Closed Issue



swann-demo · Januki-teo · Merge Requests · #32

Merged · Opened 3 weeks ago by Josh Swann · Edit

Resolve "Investigate vulnerability: Hardcoded constant database password"

Overview · Commits · Pipelines · Changes · 0/0 thread resolved

Closes #32 (closed)

Edited 2 weeks ago by Josh Swann

Request to merge #32 · investigate-vulnerability-hardcoded-constant-database-password into master

Pipeline #131563838 passed for 414431ca on 32-investigate-vulnerability-hardcoded-constant-database-password

Merge request approved. Approved by Josh Swann · View eligible approvers

Code quality improved on 52 points · Expand

Security scanning detected 13 new, and 65 fixed vulnerabilities · View full report · Expand

License Compliance detected no new licenses · Manage licenses · View full report

Merged by Josh Swann 2 minutes ago · Revert · Cherry-pick
The changes were merged into master with 952e47ea · The source branch has been deleted

Closed #32 (closed)
Assign yourself to this issue

Pipeline #148454288 running for 952e47ea on master · View latest app · Review

Will deploy to production

Development · Search or filter results...

Show labels · Edit board · Add list · Add issue

Open · Closed

Click here if you forgot your password · Change the Point and Mile · Put links back to the homepage from the greetings page · I Wanna go Fast! · Update maven repo url · Update new base Docker image for JUnit · Add some code quality issues in the code · something new · Create an issue template for User Stories · Replace the login page · Move the Login Block to the center of the page · Add new code for the toolbar script

Investigate vulnerability: Password in URL · Check to see if the Template still works · Investigate vulnerability: Found Spring endpoint · Super Awesome new Feature implementation · Reset the password field in the project · Create a .github/gitlab-webhooks.yml file to enable web terminal feature · Investigate vulnerability: Hard-coded key



GitLab

Lab



GitLab

Create Stage Gitlab Basics - Lab 2

Lab Assignment

Part 2

Scenario: Use CI features to branch from, modify, and integrate updates into a known repository state.

Key Tasks to Complete

1. Review Issue
2. Create a MR
3. Open in Web IDE
4. Modify README.md file
5. Commit file changes
6. Resolve Draft Status
7. Merge changes

Access project issues



In the project repository:

1. Select 'Issues'.

A screenshot of the GitLab interface. At the top is a dark header bar with the GitLab logo and a 'Menu' button. Below it is a sidebar with the following items:

- J JV workshop project
- Project information
- Repository
- Issues** (highlighted with a blue background)
- Boards
- Service Desk
- Milestones
- Iterations
- Merge requests
- Requirements

On the right side of the sidebar, there are sections for 'WiP Board' (with counts 0, 0, 0) and a '+' button. The main content area shows a summary of project status: 0 open issues, 0 closed issues, and 2 total issues.



From the project issues list:

1. Select '<any issue>'.

A screenshot of the GitLab Issues list. At the top, there are filters for 'Open 2', 'Closed 0', and 'All 2'. Below that is a search bar with 'Recent searches' and a placeholder 'Search or filter results...'. The list contains two items:

- Issue B**
#2 · created 1 hour ago by Invited User iugyed6r ⓘ
- Issue A**
#1 · created 1 hour ago by Invited User iugyed6r ⓘ

Each item has a small red arrow pointing to its title, indicating it can be selected.



Create a Merge Request from issue



From Issues detail view:

1. Select ‘Create merge request’ button.

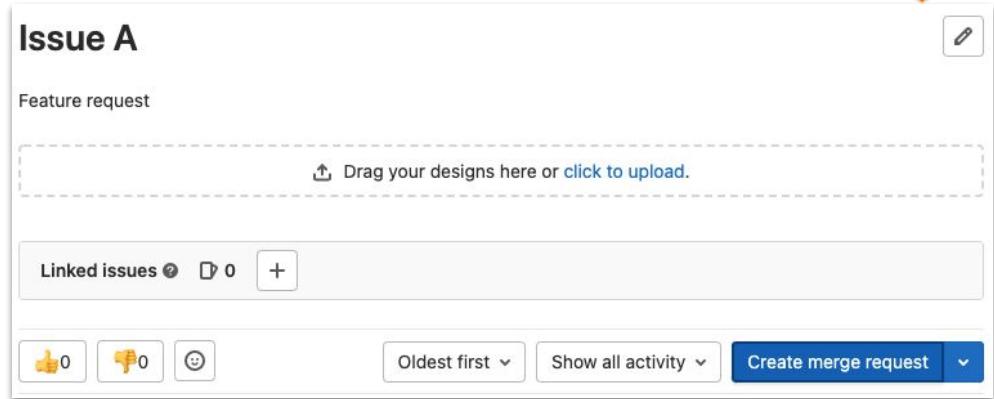
Issue A

Feature request

Drag your designs here or [click to upload](#).

Linked issues 0 +

0 0 0 Oldest first Show all activity Create merge request



From your Merge Request view:

1. Select ‘Open in Web IDE’ button.

Draft: Resolve "Issue A"

Overview 0 Commits 0 Changes 0

Closes #1

Request to merge 1-issue-a into main Open in Web IDE



FYI, the Merge Request (MR) is in “Draft” – it can not merged into a target baseline until the MR is marked as “Ready”.

Access file from Web IDE



From the Web IDE view:

1. Select ‘**README.md**’ from the list of project files.
2. File contents will be displayed in the edit pane.



JV workshop project
training-users/session-50d13a3c/iugyed6r/j...
1-is... !1

Edit

README.md

Edit Preview Markdown

```
1 # JV workshop project
2
3 This project is for a workshop
4
```

From the Web IDE edit pane:

1. Modify the contents of the ‘**README.md**’ file in the ‘**Edit**’ pane.
2. Example: “**This line has been edited**”.
3. Select the ‘**Commit**’ button.

Note: Below the commit button it tracks the number of the files that you have changed

JV workshop project
training-users/session-50d13a3c/iugyed6r/j...
1-is... !1

Edit

README.md

Edit Preview Markdown

```
1 # JV workshop project
2
3 This project is for a workshop
4
5 This line has been edited. | ←
```

Web IDE: file diffs



From your Web IDE view:

1. Confirm before/after changes to file in a side-by-side view.

```
1 # JV workshop project
2
3 This project is for a workshop
```

```
1 # JV workshop project
2
3 This project is for a workshop
4+
5+This line has been edited.
6+
```

From your Web IDE view:

2. Add text to 'Commit Message' pane.
3. Select the 'Commit' button.

Commit Message ?

README modified to reflect new edits.

Commit to 1-issue-a branch

Create a new branch

Commit **Discard draft**

Web IDE: consolidated content



From your Web IDE view:

1. Select the '**Branches/MR**' drop-down menu.
2. Select the '**Merge Requests**' tab.
3. Select the Merge Request for this change.

The screenshot shows the Atlassian Bitbucket Web IDE interface. At the top, there's a header bar with the project name 'JV workshop project' and a dropdown menu showing '1-is... !1'. Below the header is a navigation bar with two tabs: 'Branches' and 'Merge requests'. A red arrow points from the text 'Select the Merge Requests tab.' to the 'Merge requests' tab. The 'Merge requests' tab is highlighted with a grey background. Underneath the navigation bar is a search bar labeled 'Search merge requests' with a magnifying glass icon. The main content area displays a single merge request titled 'Draft: Resolve "Issue A"' with a checkmark icon and the URL 'training-users/session-50d13a3c/iugyed6r/jv-workshop-prep/jv-workshop-project!1'. A red arrow points from the text 'Select the Merge Request for this change.' to this merge request entry.

Web IDE: return to MR



From your Web IDE view:

1. Select the hyperlinked '[**<!MR>**](#)'.
2. Example: '[**<!1>**](#)'.

The screenshot shows a Jira Software interface for a project titled 'JV workshop project'. The URL in the address bar is 'training-users/session-50d13a3c/iugyed6r/j...'. Below the title, there are two buttons: 'Review' and 'Edit'. A red arrow points to the link 'Merge request (!1)' under the 'Review' button. At the bottom, there is a file list with 'README.md'.

MR details

From the MR details view, confirm these key automated details.

1. Link with issue.
2. Branching/target links.
3. Approval information.
4. Inherited milestone association.
5. Inherited labels.
6. Commit history.

The screenshot shows a GitLab Merge Request (MR) details page for a draft merge request titled "Resolve 'Issue A'".

- Closes #1**: A red arrow points to the "Closes #1" link in the top navigation bar.
- Request to merge 1-issue-a into main**: A red arrow points to the merge request summary.
- Approval is optional**: A red arrow points to the approval status section.
- Draft merge requests can't be merged**: A red arrow points to the note indicating the merge request is still a draft.
- Milestone**: A red arrow points to the "Milestone A" entry in the sidebar.
- Labels**: A red arrow points to the "OS Linux" label entry in the sidebar.
- Commit history**: A red arrow points to the commit log at the bottom of the page, which shows a single commit: "ca7b3e1d - README modified to reflect new edits."

Assignee: Invited User iugyed6r @iugyed6r

Reviewers: None

Milestone: Milestone A

Time tracking: No estimate or time spent

Labels: OS Linux

Lock merge request: Unlocked

1 participant: Invited User iugyed6r @iugyed6r

Notifications: Reference: training-users/se...

Source branch: 1-issue-a

MR view: commit details



From the MR details view:

1. Select 'Commits' tab.

Draft: Resolve "Issue A"

Overview 0 **Commits 1** Changes 1

09 Oct, 2021 1 commit



README modified to reflect new edits.

Invited User iugedy6r authored 4 minutes ago

Add previously merged commits

From the MR details view:

1. Select 'Changes' tab.
2. Confirm before/after changes to file in red/green view.

Draft: Resolve "Issue A"

Overview 0 Commits 1 **Changes 1**



Compare main and latest version

README.md

1	1	# JV workshop project
2	2	
3		- This project is for a workshop
		\ No newline at end of file
	3	+ This project is for a workshop
	4	+
	5	+ This line has been edited.

MR view: aggregated data



From the MR details view:

1. Select 'Overview' tab.



Draft: Resolve "Issue A"

Overview 0 Commits 1 Changes 1

Closes #1

Request to merge 1-issue-a into main [Open in Web IDE](#)

Approval is optional

> View eligible approvers

Merge This merge request is still a draft. [Mark as ready](#)

Draft merge requests can't be merged.

Closes #1

Older

From the MR details view,
confirm these key *automated*
details.

2. 'Approval' panel for status of outstanding approvals.
3. 'Merge' button is "grayed out" to prevent accidental merging of unfinished work.



From the MR details view:

4. Select the 'Mark as ready' button.



MR view: ready for integration



**From the MR details view,
confirm the MR is ready for
integration.**

1. Banner alerts viewers of ability to merge this MR.
2. 'Draft' prefix is dropped from MR title.
3. 'Merge' button is colored blue.



The screenshot shows a GitLab Merge Request (MR) details page. At the top, a banner says "The merge request can now be merged." Below this, the MR is listed as "Open" and "Created 13 minutes ago by Invited User". The title of the MR is "Resolve 'Issue A'". There are three tabs: "Overview 0", "Commits 1", and "Changes 1". A note below the tabs says "Closes #1". In the main content area, there is a "Request to merge 1-issue-a into main" section with a "Merge" button. Below this, there is a note "Approval is optional" and a link "View eligible approvers". At the bottom, there is a "Merge" button with a checkmark and a checkbox for "Delete source branch". Red arrows from the text points are pointing to the "Merge" button in the banner, the "Merge" button in the main content area, and the "Merge" button at the bottom of the page.

From the MR details view:

4. Select the 'Merge' button.



MR view: merge changes



From the MR details view,
confirm the MR has been
integrated.

1. 'Merged' badge replaces 'Open' badge.
2. 'Merged by' reflects integrator and branch/target information.



The screenshot shows a pull request titled "Resolve 'Issue A'" in a "Merged" state, created 13 minutes ago by an invited user. The pull request details include an overview, commits, and changes. It also shows it closes issue #1. The merge section indicates a request to merge the source branch into the main branch, with approval being optional and a link to view eligible approvers. The final status shows the pull request was merged by the invited user just now, with the commit hash 25d8dabf and a note that the source branch has been deleted.

Merged Created 13 minutes ago by Invited User iugyed6

Resolve "Issue A"

Overview 0 Commits 1 Changes 1

Closes #1

Request to merge 1-issue-a into main

Approval is optional

View eligible approvers

Merged by Invited User iugyed6r just now Revert

The changes were merged into main with 25d8dabf

The source branch has been deleted





GitLab

Everyone can contribute



GitLab

GitLab CI/CD

A Complete DevOps Platform Delivered as a Single Application



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Defend

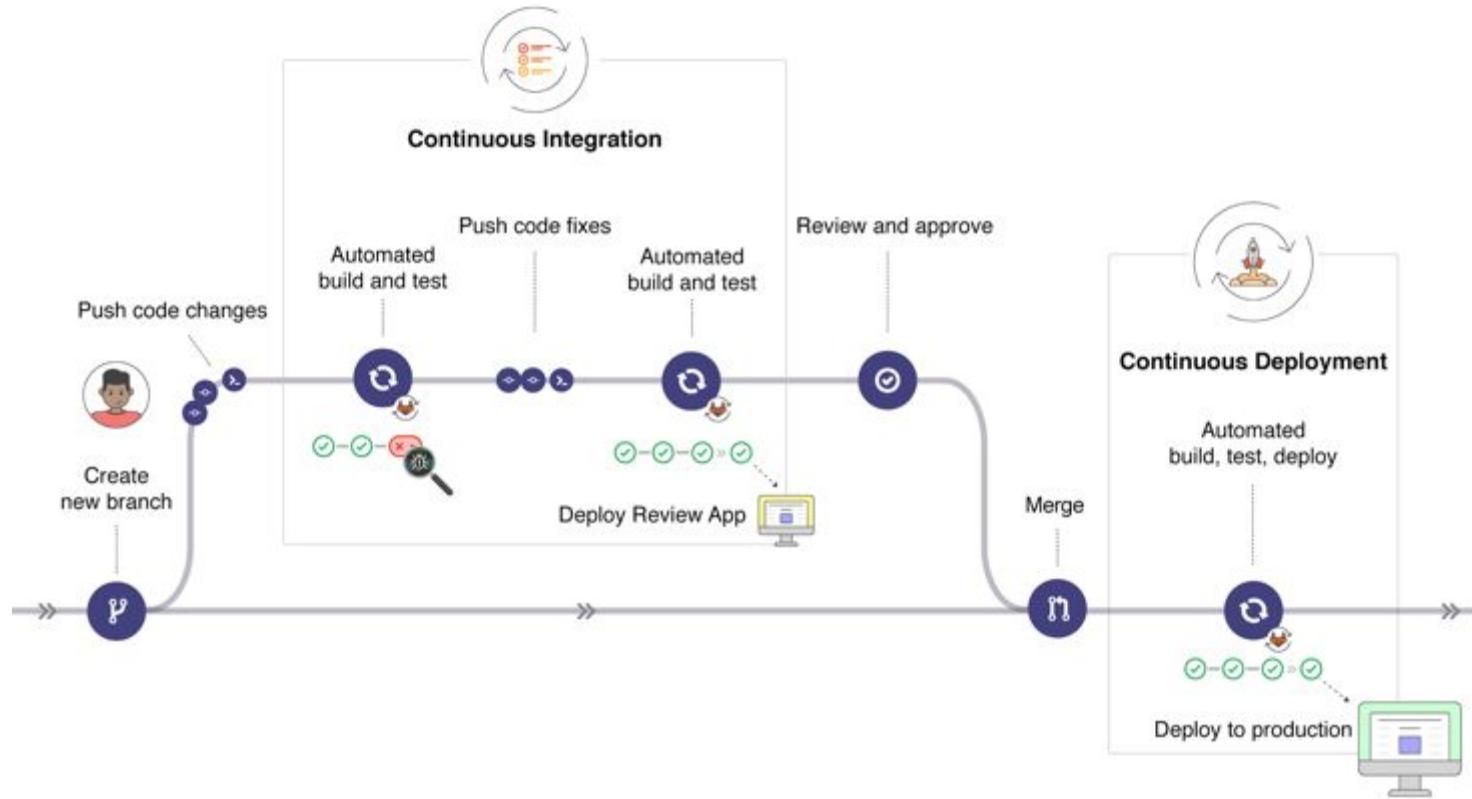
Single Conversation
Single Data Store
Single Permission Model
Single Interface
Governance & Security
Team Collaboration
Lifecycle Analytics



**“Cycle time compression
may be the most
underestimated force in
determining winners &
losers in tech.”**

— Marc Andreessen

“Shifting Left” is Key



YAML Configuration



The screenshot shows a GitLab repository interface. On the left, there's a sidebar with project navigation links: Project, Repository (selected), Files, Commits, Branches, Tags, Contributors, Graph, Compare, Charts, Locked Files, Issues (22), Merge Requests (2), CI / CD (selected), Operations, Registry, and a Collapse sidebar button. The main content area shows the repository path: fpotter > tools > busy > Repository. A dropdown menu shows 'master' selected. Below it, a file list shows '.gitlab-ci.yml' with a size of 1.33 KB. A message indicates 'Include the SAST template' by Francis Potter, authored 2 months ago. A validation message says '✓ This GitLab CI configuration is valid. Learn more'. The code editor displays the following YAML configuration:

```
include:
  template: SAST.gitlab-ci.yml

stages:
- test
- build
- publish

Test:
image: python
script: "make test"

Test Coverage:
image: python
script: "make cover"
before_script:
- "pip3 install coverage"
coverage: '/TOTAL +\d+ +\d+ +(\d+)\%\s*/'

Style Check:
image: python
script: "make style"
before_script:
- "pip3 install pycodestyle"
```

Simple Job



Before Script



```
Build Package:  
  stage: build  
  image: python  
  script: "make build"  
  only:  
    variables:  
      - $CI_COMMIT_TAG =~ /^v\d+\.\d+\.\d+$/  
  artifacts:  
    paths:  
      - dist/
```

Which Stage

Conditions to Run

Artifacts to
Generate

Define Stages



Define Your Stages

```
stages:  
  - test  
  - build  
  - publish
```

Include other CI Configurations



```
include:  
  template: SAST.gitlab-ci.yml
```

GitLab's standard
SAST job
(Ultimate tier)

CI/CD Pipeline



GitLab Projects Groups Activity Milestones Snippets

Search or jump to... Pipeline #66796601 triggered 1 minute ago by Francis Potter

Labore tempora sed voluptatem eius est ut

11 jobs for 36-labore-tempora-sed-voluptatem-eius-est-ut (queued for 2 seconds)

Latest Auto DevOps 48aa31ec ...

Pipeline Jobs 11

Build Test Review Dast Performance

build code_quality review dast performance

container_scan... dependency_sc... license_manag... sast test

Diagram illustrating a CI/CD pipeline. The pipeline consists of several stages: Build, Test, Review, Dast, and Performance. The Build stage contains a job named 'build'. The Test stage contains six jobs: 'code_quality', 'container_scan...', 'dependency_sc...', 'license_manag...', 'sast', and 'test'. The Review stage contains a job named 'review'. The Dast stage contains a job named 'dast'. The Performance stage contains a job named 'performance'. All jobs are currently in a 'passed' state, indicated by green checkmarks. The pipeline is currently queued for 2 seconds.

YAML Templates



gitwg.org · GitHub Community Edition · Repository

master · githab-ci · lib · githab · ci · templates · + · × · Lock · History · ⌂ · Find file · Web IDE · ⌂ · ×

Merge branch 'bump-auto-devops-helm-2-14-0' into 'master' · 4c27abb4 · 6 hours ago · Misra Cabrera

Name	Last commit	Last update
Jobs	add postgres version to subsequent helm deploys	1 week ago
Pages	Merge branch '86748-update-nodes-to-10.15.2 ...	1 month ago
Security	Disable proxy in container scanning template	6 days ago
Android-Fastlane.githab-ci.yml	Merge branch 'require-all-templates-to-include-...	1 month ago
Android.githab-ci.yml	Merge branch 'require-all-templates-to-include-...	1 month ago
Auto-DevOps.githab-ci.yml	Bump Helm version in Auto-DevOps.githab-ci.yml	20 hours ago
Batch.githab-ci.yml	Tidy up CI templates	2 months ago
C++-githab-ci.yml	Tidy up CI templates	2 months ago
Chef.githab-ci.yml	Merge branch 'require-all-templates-to-include-...	1 month ago
Closure.githab-ci.yml	Tidy up CI templates	2 months ago
Code-Quality.githab-ci.yml	Merge branch 'fix-github-action-issues'	2 months ago
Crystal.githab-ci.yml	Resolve "Un-resolvable dependency graph"	2 months ago
Django.githab-ci.yml	Tidy up CI template	2 months ago
Docker.githab-ci.yml	Jobs should be defined in the Dockerfile	2 months ago
Elixir.githab-ci.yml	Resolve "Un-resolvable dependency graph"	2 months ago
Go.githab-ci.yml	Merge branch 'fix-github-action-issues'	2 months ago
Gradle.githab-ci.yml	Tidy up CI template	2 months ago
Grails.githab-ci.yml	Tidy up CI template	2 months ago
Julia.githab-ci.yml	Tidy up CI template	2 months ago
LaTeX.githab-ci.yml	Resolve "Un-resolvable dependency graph"	2 months ago
Laravel.githab-ci.yml	Tidy up CI template	2 months ago
Maven.githab-ci.yml	Clarify error message for failed builds	2 months ago
Mono.githab-ci.yml	Merge branch 'fix-github-action-issues'	2 months ago
Nodejs.githab-ci.yml	Resolve "Un-resolvable dependency graph"	2 months ago

Merge Requests



The screenshot shows a GitLab Merge Request page for a project named "panda". The merge request is titled "WIP: Resolve 'Labore tempora sed voluptatem eius est ut'" and was opened 1 hour ago by Francis Potter. It is currently in progress and assigned to Francis Potter. The merge request has been reviewed by a pipeline and deployed to a review branch. There are no approvals required, and the code quality check is successful. However, security scanning detected 87 new vulnerabilities, and license management found no new licenses. The merge request has 1 participant and notifications are turned on.

fpotter > tour > panda > Merge Requests > #36

Open Opened 1 hour ago by Francis Potter

Edit Close merge request

WIP: Resolve "Labore tempora sed voluptatem eius est ut"

Closes #36

Edited just now by Francis Potter

Request to merge 36-labore-tempora-s... into master

Open in Web IDE Check out branch

Pipeline #66796691 passed for 48aa31ec on 36-labore-tempora-s...

Deployed to review/36-labore-tempora-s... 1 hour ago

No approval required

No changes to code quality

Security scanning detected 87 new, and 51 fixed vulnerabilities

License management detected no new licenses

Manage licenses View full report

Todo Add todo

Assignee Francis Potter @francispotter

Milestone 3.6.4

Time tracking Spent: 4h

Labels In Progress Tour

Lock merge request Unlocked

1 participant

Notifications

Reference: fpotter/tour/panda...

Deployment environments in a single view



Screenshot of the GitLab interface showing deployment environments for the project "chipmunk".

The left sidebar shows project navigation: Project overview, Repository, Issues (36), Merge Requests (8), CI / CD, Security & Compliance, Operations (Metrics, Tracing), Environments (selected), Error Tracking, Serverless, and Pod logs.

The main area displays two environments:

- staging**: Deployment #57 by fporter, staging #39504...
Commit: master → 9af71d0a, 1 month ago
Merge branch '13-temp...'.
- production**: Deployment #47 by fporter, rollout 100% #35...
Commit: master → 36cf9b6e, 2 months ago
Merge branch '5-neque...'.

Each environment section includes a summary table with columns: Environment, Deployment, Job, Commit, Updated, and Auto stop in. Below each summary is a detailed status table for "Instance (1)" showing metrics: 100% Complete, Succeeded (green), Running (green), Failed (red), Pending (grey), Unknown (white), and Canary (yellow).

A "New environment" button is located in the top right corner of the main content area.

Pipelines list



GitLab Next Projects Groups More Search or jump to... Run Pipeline CI Lint

All 1,000+ Pending 88 Running 61 Finished 1,000+ Branches Tags

Status	Pipeline	Triggerer	Commit	Stages	Duration	Actions
failed	#112964640	detached	[`12318 <-- 447e7fa8	Apply suggestion to d...	0:00:04:26	13 minutes ago
cancelled	#112957353	detached	[`123458 <-- d339af97	Cache merged branch...	0:00:11:26	26 minutes ago
failed	#112952059	detached	[`21453 <-- fa239bb5c	Merge branch 'master'...	0:00:04:48	37 minutes ago
failed	#112945220	detached	[`123832 <-- 6fba2814	Update test description	0:00:50:56	12 seconds ago
failed	#112936137	detached	[`123862 <-- 23fc8fe5	Add job-execution-tim...	0:01:20:30	6 minutes ago
passed	#112935325	detached	[`master <-- 87e579ce	Merge branch 'mattkas...	0:01:16:23	12 minutes ago
failed	#112933010	detached	[`123458 <-- 18b652c2	Re-add line accidentally...	0:01:08:43	27 minutes ago
failed	#112930595	detached	[`123861 <-- dcf35018	First pass updating MR...	0:00:04:44	1 hour ago

Operations Dashboard



GitLab Projects Groups More Search or jump to... 0 1 2 3 4 5 6 7 8 Add projects

Operations Dashboard

The Atomic Lab / The Quark 4pp 3 months ago 0 Alerts failed	The Atomic Lab / The Quantum App 3 months ago 0 Alerts passed
The Atomic Lab / The Neutrino App 3 months ago 0 Alerts passed	The Atomic Lab / The Quark App 4 months ago 0 Alerts passed
Joel Krooswyk / SpringSample 0 Alerts blocked	GitLab.org / GitLab 0 Alerts running
GitLab.org / gitlab-runner 14 minutes ago 0 Alerts failed	GitLab.org / GitLab FOSS 0 Alerts running

Kubernetes!



kubernetes

GitLab Projects Groups More This project Search 🔍 📁 🚀 🛡️ 🗃 🏷️ 🎯 🙋‍♂️

Applications

Install applications on your Kubernetes cluster. Read more about [installing applications](#)

Helm Tiller	Helm streamlines installing and managing Kubernetes applications. Tiller runs inside of your Kubernetes Cluster, and manages releases of your charts.	Installed
Ingress	Ingress gives you a way to route requests to services based on the request host or path, centralizing a number of services into a single endpoint.	Installed
	Ingress IP Address 35.184.119.234	ⓘ
	Point a wildcard DNS to this generated IP address in order to access your application after it has been deployed. More information	
Prometheus	Prometheus is an open-source monitoring system with GitLab Integration to monitor deployed applications.	Manage Installed
GitLab Runner	GitLab Runner connects to this project's repository and executes CI/CD jobs, pushing results back and deploying, applications to production.	ⓘ Installing



Step 1: Configure Kubernetes integration



Integrate Kubernetes cluster automation

Kubernetes clusters allow you to use review apps, deploy your applications, run your pipelines, and much more in an easy way. Adding an integration to your group will share the cluster across all your projects. Learn more about group Kubernetes clusters.

Add Kubernetes cluster

Step 2: Turn on Auto Devops

Auto DevOps

Auto DevOps will automatically build, test, and deploy your application based on a predefined Continuous Integration and Delivery configuration. Learn more about Auto DevOps.

Default to Auto DevOps pipeline

The Auto DevOps pipeline will run if no alternative CI configuration file is found. More information

Deployment strategy

Continuous deployment to production

Continuous deployment to production using timed incremental rollout

Automatic deployment to staging, manual deployment to production

Save changes

Step 3: Let GitLab do everything

1. Build the Docker container

- Detects the language/platform of the code (if no Dockerfile provided)
- Performs normal build steps for that language/platform
- Adds Docker image to container registry

2. Run your unit tests and integration tests

3. Run static scans and reports to MR

- Code Quality Scan
- Static Application Security Scan
- Dependency Scan
- Container Scan
- Licence Management Scan

4. Deploy to review app with hostname and SSL

5. Run review app scans and reports to MR

- Dynamic Application Security Scan
- Browser Performance Tests

6. Deploy to Staging using Helm Chart

- Only after merge
- Runs post-deployment steps e.g. database migrations

7. Deploy to Production on command

- Supports Canary deployment option
- Supports incremental rollout option

8. Instrument for monitoring using Prometheus

9. Record all deployments on the Deploy Board for the project



GitLab

Lab



Release Stage

CI/CD

Lab Assignment

Part 3

Scenario: Create a `.gitlab-ci.yml` file to visualize an automated CI process.

Key Tasks to Complete

1. Create a `.gitlab-ci.yml` File
2. View the CI pipeline
3. Update the CI pipeline

Access repository files



In the MR view:

1. Select **Repository**→**Files**.

J JV workshop project Training Users > ... > JV wo
Project information
Repository
Issues 2
Merge requests 0
Requirements
CI/CD
Security & Compliance
Commits
Branches
Tags
Contributors
Graph

In the repository file view:

2. Select the '+' → 'New file'

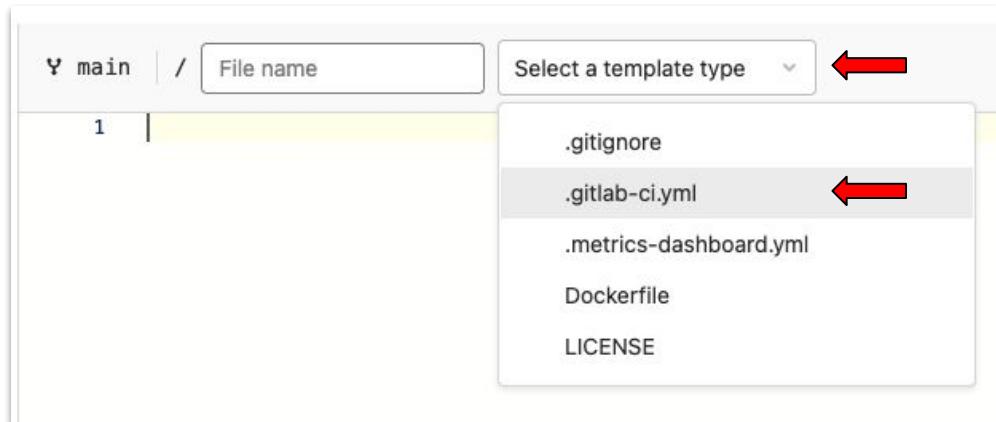
jv-workshop-project / +
This directory
New file
Upload file
New directory

Select file template type



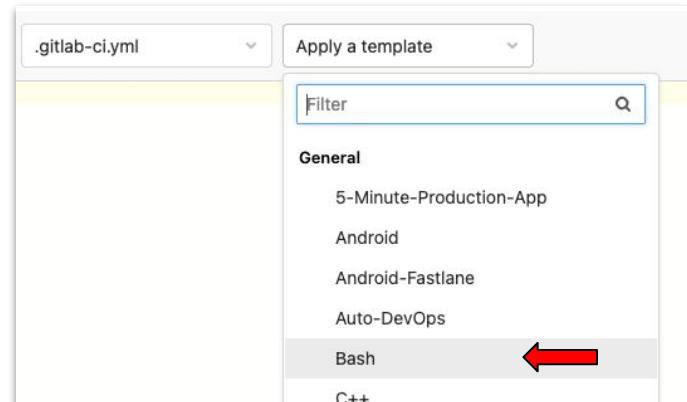
In the New file view:

1. Select the '**Select a template type**' drop-down→ '**.gitlab-ci.yml**' option.



In the New file view:

2. Select the '**Apply a template**' drop-down→ '**Bash**' option.



File title and contents populated with template



main / .gitlab-ci.yml .gitlab-ci.yml Apply a template

```
1 # This file is a template, and might need editing before it works on your project.
2 # To contribute improvements to CI/CD templates, please follow the Development guide at:
3 # https://docs.gitlab.com/ee/development/cicd/templates.html
4 # This specific template is located at:
5 # https://gitlab.com/gitlab-org/gitlab/-/blob/master/lib/gitlab/ci/templates/Bash.gitlab-ci.yml
6
7 # See https://docs.gitlab.com/ee/ci/yaml/README.html for all available options
8
9 # you can delete this line if you're not using Docker
10 image: busybox:latest
11
12 before_script:
13   - echo "Before script section"
14   - echo "For example you might run an update here or install a build dependency"
15   - echo "Or perhaps you might print out some debugging details"
16
17 after_script:
18   - echo "After script section"
19   - echo "For example you might do some cleanup here"
20
21 build1:
22   stage: build
23   script:
24     - echo "Do your build here"
25
26 test1:
27   stage: test
28   script:
```

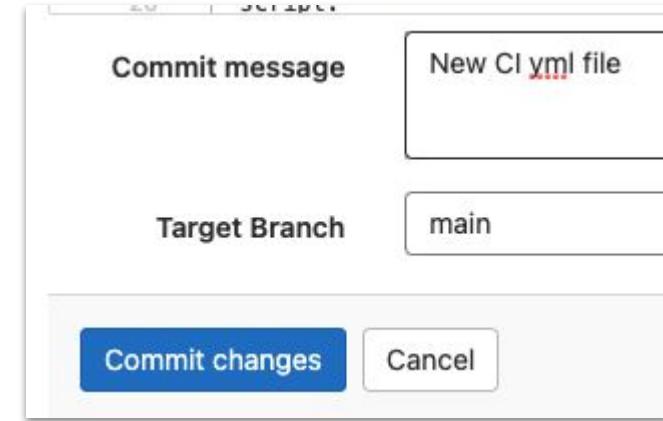


Add commit message



In the New file view:

1. Enter <commit message text> into the 'Commit message' pane.
2. Select the 'Commit changes' button.



Components of this .gitlab-ci.yml



In the file view, confirm the following components.

1. Leverages Docker image.
2. 'before_script' block.
3. 'after_script' block.
4. Pipeline elements of 'stage' and 'script'.

File: .gitlab-ci.yml | 1.15 KB

Edit | Web IDE

```
1 # This file is a template, and might need editing before it works on
2 # To contribute improvements to CI/CD templates, please follow the D
3 # https://docs.gitlab.com/ee/development/cicd/templates.html
4 # This specific template is located at:
5 # https://gitlab.com/gitlab-org/gitlab/-/blob/master/lib/gitlab/ci/t
6
7 # See https://docs.gitlab.com/ee/ci/yaml/README.html for all availab
8
9 # you can delete this line if you're not using Docker
10 image: busybox:latest
11
12 before_script:
13   - echo "Before script section"
14   - echo "For example you might run an update here or install a buil
15   - echo "Or perhaps you might print out some debugging details"
16
17 after_script:
18   - echo "After script section"
19   - echo "For example you might do some cleanup here"
20
21 build1:
22   stage: build
23   script:
24     - echo "Do your build here"
25
```

Access repository pipelines



In the file view:

1. Select CI/CD→Pipelines.

The screenshot shows the Bitbucket sidebar with several options: Requirements, CI/CD, Security & Compliance, Deployments, Pipelines, Editor, and Jobs. The 'Pipelines' option is highlighted with a red arrow pointing to it.

In the Pipelines view:

2. Confirm the pipeline elements below.

- Pipeline status.
- Pipeline ID.
- Pipeline triggerer.
- Pipeline target.
- Pipeline stages and status.
- Pipeline time.

Status	Pipeline	Triggerer	Commit	Stages	Duration
passed	#132290 latest		main -> 1fb67868 New CI yml file	✓ ✓ ✓	00:00:22 1 minute ago

Red arrows point from the following labels to specific elements in the table:
Status → Pipeline status (green passed box)
Pipeline → Pipeline ID (#132290)
Triggerer → Pipeline triggerer (green circle icon)
Commit → Pipeline target (main branch, commit hash 1fb67868)
Stages → Pipeline stages and status (green checkmarks)
Duration → Pipeline time (00:00:22)

Access pipeline details



In the Pipelines view:

1. Select the '**<pipeline status>**' button, or
2. Select the '**<pipeline ID>**' hyperlink.



Status	Pipeline	Triggerer	Co
passed	#132290 	latest 	

In the pipeline view:

3. Confirm the additional pipeline elements.

- 'latest' flag,
- Related merge requests.
- Additional tabs for context.
- Expanded pipeline view.

① 4 jobs for main in 22 seconds (queued for 4 seconds)

latest

↳ 1fb6786 ↳

No related merge requests found.

Pipeline Needs Jobs 4 Tests 0

Build	Test	Deploy
build1 	test1 	deploy1 
	test2 	

Access job details



In the pipeline view:

1. Select '**<individual job>**' button.

Pipeline Needs Jobs 4 Tests 0

Build	Test
build1	test1
	test2

In the job view:

2. Confirm the following elements.

- Job name.
- Job duration and runner name.
- Job artifacts (if avail.).
- Pipeline stage drop-down.
- List of pipeline jobs.
- Terminal output.

build1

Duration: 9 seconds

Timeout: 1h (from project)

Runner: #2 (-dma5bc6) work gitlab-runner-manager

Commit 1fb67868

New CI yml file

Pipeline #132290 for main

Preparing the "docker+machine" executor
Using Docker executor with image busybox:latest ...
Pulling docker image busybox:latest ...
Using docker image sha256:16ea53ea7c652456803632d67517b78a4f9075a10bfcd4fc6b7b4cbf2
bc98497 for busybox:latest with digest busybox@sha256:f7ca5a32c10d51aeda3b4d01c61c60
61f497893d7f6628b92f822f7117182a57 ...
Preparing environment
Running on runner--dma5bc6-project-528-concurrent-0 via runner-dma5bc6-1633712545-2
b1c5853...
Getting source from Git repository
Fetching changes with git depth set to 50...
Initialized empty Git repository in /builds/training-users/session-50d13a3c/iugyed6

Access repository files



In the job view:

1. Select **Repository**→**Files**.

J JV workshop project

Project information

Repository

Issues 1

Merge requests 0

Requirements

Commits

Branches

Tags

5 Preparing

Files

In the Repository files list view:

2. Select '.gitlab-ci.yml' file name.

Name	Last comm
New CI yml file	Invited User iugyed6r authored 11 minutes ago
.gitlab-ci.yml	New CI ym
README.md	README n
README.md	README

Open .gitlab-ci.yml file to edits



In the file content view:

1. Select the 'Edit' button.



.gitlab-ci.yml 1.15 KB

Edit Web IDE

```
1 # This file is a template, and might need editing before it works on
2 # To contribute improvements to CI/CD templates, please follow the De
3 # https://docs.gitlab.com/ee/development/cicd/templates.html
4 # This specific template is located at:
5 # https://gitlab.com/gitlab-org/gitlab/-/blob/master/lib/gitlab/ci/te
6
7 # See https://docs.gitlab.com/ee/ci/yaml/README.html for all available
```

In the file edit view:

2. Copy the contents of the 'test1:' element.



```
20
21  build1:
22    stage: build
23    script:
24      - echo "Do your build here"
25
26  test1:
27    stage: test
28    script:
29      - echo "Do a test here"
30      - echo "For example run a test suite"
31
32  test2:
33    stage: test
34    script:
35      - echo "Do another parallel test here"
```

Modify the .gitlab-ci.yml file - part 1



In the file edit view:

1. Scroll to the bottom of the file.
2. Insert a space after the last row in the file.
3. **Paste** the copied contents to the end of the `.gitlab-ci.yml` file.

```
37
38 deploy1:
39   stage: deploy
40   script:
41     - echo "Do your deploy here"
42
43 test1:
44   stage: test
45   script:
46     - echo "Do a test here"
47     - echo "For example run a test suite"
```



In the file edit view:

4. Modify the pasted contents to reflect:

```
test3:
  stage: test
  script:
    - cat README.md
```

```
37
38 deploy1:
39   stage: deploy
40   script:
41     - echo "Do your deploy here"
42
43 test3:
44   stage: test
45   script:
46     - cat README.md
47
```



Add commit message for .gitlab-ci.yml file



In the file edit view:

1. Enter <commit message text> into the 'Commit message' pane.
2. Select the 'Commit changes' button.

A screenshot of the GitLab commit dialog. At the top, there is a code editor window showing a snippet of a .gitlab-ci.yml file with a yellow highlight over the stage definition. Below the code editor are two input fields: 'Commit message' containing 'Created test3 job for test stage in gitlab-ci.yml' and 'Target Branch' containing 'main'. At the bottom are two buttons: 'Commit changes' (highlighted in blue) and 'Cancel'.

In the file view:

3. Verify the following components.
 - Commit status message.
 - Configuration lint.

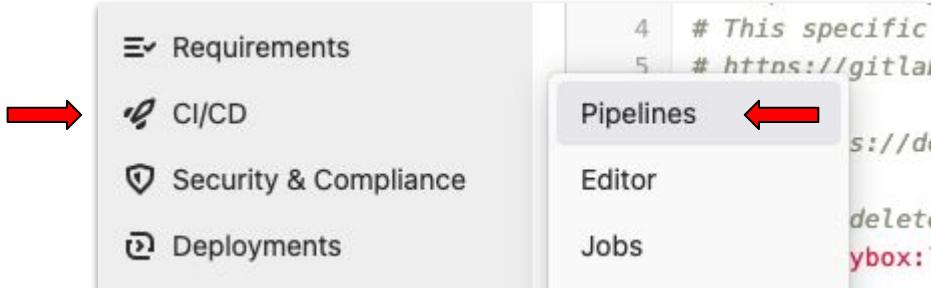
A screenshot of the GitLab commit status page. At the top, it shows a message: 'Your changes have been successfully committed.' Below this, it displays the commit message 'Created test3 job for test stage in gitlab-ci.yml' and the author information 'Invited User iugyed6r authored just now'. At the bottom, it shows a green checkmark icon and the message '✓ This GitLab CI configuration is valid. Learn more'.

Access repository pipelines



In the file view:

1. Select CI/CD→Pipelines.



In the Pipelines view:

2. Confirm updates to the following pipeline elements.

- Pipeline status.
- Pipeline ID.
- Pipeline triggerer.
- Pipeline target.
- Pipeline stages and status.
- Pipeline time.

Status	Pipeline	Triggerer	Commit	Stages	Duration
passed	#132292 latest	●	▶ main → b1b2318b Created test3 job for ...	✓ ✓ ✓	⌚ 00:00:23 🕒 2 minutes ago
failed	#132291 yaml invalid error	●	▶ main → 27184cff Created test3 job to t...		🕒 5 minutes ago
passed	#132290	●	▶ main → 1fb67868 New CI yml file	✓ ✓ ✓	⌚ 00:00:22 🕒 22 minutes ago



Access pipeline details



In the Pipelines view:

1. Select the '**<pipeline status>**' button, or
2. Select the '**<pipeline ID>**' →

Status	Pipeline	Triggerer	Commit	Stages	Duration
passed	#132292 latest		main -> b1b2318b Created test3 job for ...	✓ ✓ ✓	00:00:23 2 minutes ago

In the pipeline view:

3. Confirm the following pipeline elements.
 - Commit message listed.
 - Additional job added to “Test” stage.

→

Created test3 job for test stage in gitlab-ci.yml

5 jobs for main in 23 seconds (queued for 4 seconds)

latest

-> b1b2318b

No related merge requests found.

Pipeline Needs Jobs 5 Tests 0

Build	Test	Deploy
build1	test1	deploy1
	test2	
	test3	

Access job details



In the pipeline view:

1. Select ‘<individual job>’ button.

Example:

1. Select the ‘**test3**’ button.

Created test3 job for test stage in gitlab-ci.yml

⌚ 5 jobs for [main](#) in 23 seconds (queued for 4 seconds)

⟳ [latest](#)

→ [b1b2318b](#) ↗

↑↑ No related merge requests found.

Pipeline Needs Jobs 5 Tests 0

Build

build1

Test

test1

Deploy

deploy1

test2

test3



Job details

In the job view:

1. Confirm the following elements.

- Job name reflects added job.
- Job duration and runner name.
- Job artifacts (if avail.).
- Pipeline stage drop-down includes added job.
- Terminal output reflects “script” for ‘**test3**’ job.



The screenshot shows the 'test3' job details page. At the top, there's a navigation bar with a logo, a back arrow, and a 'Retry' button. Below it, the job name 'test3' is displayed with a red arrow pointing to it. To the right of the job name are 'Duration: 8 seconds', 'Timeout: 1h (from project)', and a help icon. Further down, it shows the 'Runner: #2 (-dma5bc6) workshop-gitlab-runner-manager'. On the left, there's a terminal window showing the build script output. The output includes steps like 'Preparing the "docker+machine" executor', 'Using Docker executor with image busybox:latest ...', 'Pulling docker image busybox:latest ...', 'Using docker image sha256:16ea53ea7c652456803632d67517b78a4f9075a10bfd4fc6b7b4cbf2 bc98497 for busybox:latest with digest busybox@sha256:f7ca5a32c10d51aeda3b4d01c61c60 61f497893d7f6628b92f822f7117182a57 ...', 'Preparing environment', 'Running on runner--dma5bc6-project-528-concurrent-0 via runner-dma5bc6-1633744046-0 8037a5c...', 'Getting source from Git repository', 'Fetching changes with git depth set to 50...', 'Initialized empty Git repository in /builds/training-users/session-50d13a3c/iugyed6 r/jv-workshop-prep/jv-workshop-project/.git', 'Created fresh repository.', 'Checking out blb2318b as main...', 'Skipping Git submodules setup', 'Executing "step_script" stage of the job script', 'Using docker image sha256:16ea53ea7c652456803632d67517b78a4f9075a10bfd4fc6b7b4cbf2 bc98497 for busybox:latest with digest busybox@sha256:f7ca5a32c10d51aeda3b4d01c61c60 61f497893d7f6628b92f822f7117182a57 ...', '\$ echo "Before script section"', 'Before script section', and '\$ echo "For example you might run an update here or install a build dependency"'. On the right, there's a pipeline summary for Pipeline #132292 for main, showing 'test' is running, 'test1' is succeeded, 'test2' is succeeded, and 'test3' is succeeded. A red arrow points to the 'test' status in the pipeline summary.



- CI/CD example projects: <https://docs.gitlab.com/ee/ci/examples/#cicd-examples>
- CI/CD templates: <https://docs.gitlab.com/ee/ci/examples/#cicd-templates>



GitLab

Everyone can contribute