

Merge requests in GitLab

Ivan Ivanov

15-May-2018"

Intro

Intro

- Allow to exchange changes made to source code
- Collaborate with other people on the same project
- *Request to merge* one brach into another

Benefits of using GitLab Merge requests

Benefits of using GitLab Merge requests

- Comparison between changes in two branches
- Reviews and discussions – proposed modifications inline
- Live preview of the changes when *Review Apps* configured for the project
- Builds, tests and deployment in a per-branch basis with built-in GitLab CI/CD
- Prevent the MR from being merged before it's ready with *Work In Progress MRs*

Benefits of using GitLab Merge requests

- Visible deployment process through *Pipeline Graphs*
- Automatically closure of issues that originated the implementation proposed in the MR
- Convenience by reassigning the assignee
- Convenience by assigning a milestone and keeping track of the development
- Easier organization of issues and MRs with labels

Benefits of using GitLab Merge requests

- Time Estimation & Time Spent with *Time Tracking*
- Resolve merge conflicts from the UI
- Possible *Fast-Forward MRs*

- *Semi-Linear history MRs* – another security layer for pipeline passing

Benefits of using GitLab Merge requests

- Creation of new MRs by email
- Possible edit of the MR from maintainers

Merge requests per project

Merge requests per project

- *Project > Merge Requests*
- Open/Merged/Closed Requests

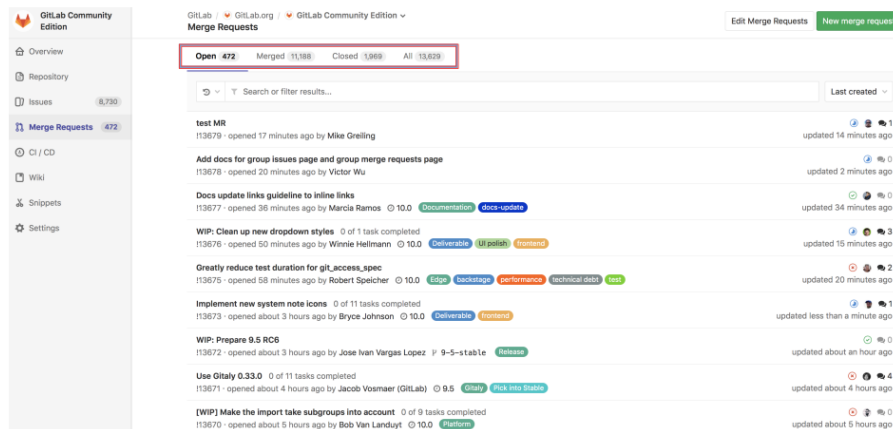


Figure 1: An example of Project MRs

Merge requests per group

Merge requests per group

- *Group > Merge Requests*

Removing the source branch

Removing the source branch

- “*Remove source branch when merge request accepted*” option on creation of a MR
- Also visible in an existing one

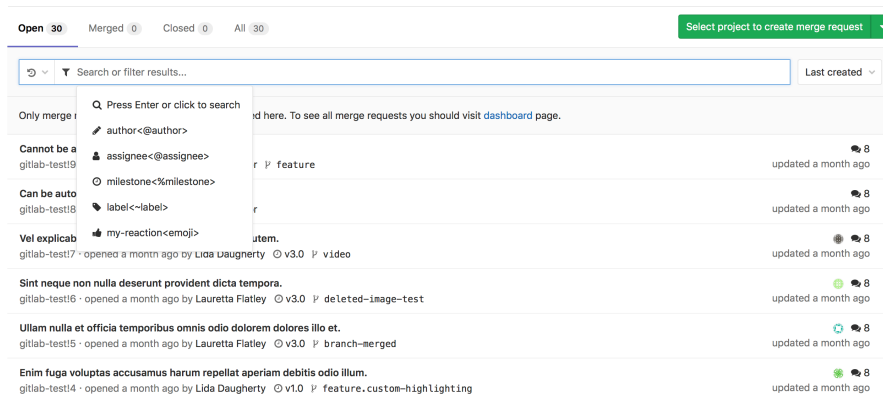


Figure 2: An example of Group MRs

- Visible only to master users



Figure 3: An example of removing source branch

Discussions

Discussions

- Comments at issues, MRs, snippets, commits, commit diffs
- Resolvable discussions

Discussions

- Commit discussions in the context of a MR

Connor Shea @connorshea started a discussion on [the diff](#) 21 minutes ago
Last updated by Luke Bennett 20 minutes ago Toggle discussion

```

app/views/admin/ip_blocking/_blocking_ips.html.html 0 → 100644
1 + %ul.content-list
2 + - if @collection.blank?
3 + %li
4 + .nothing-here-block No IP addresses to show
5 + - else
6 + .table-holder
7 + %table.table.builds
8 + %thead
9 + %tr
10 + %th IP

```

Connor Shea @connorshea commented 21 minutes ago Developer
I think this should probably be expanded to "IP Address".
Edited 6 minutes ago

Luke Bennett @lbennett commented 20 minutes ago Developer
Sounds good to me 🙌
Edited 6 minutes ago

Reply... Resolve discussion

Figure 4: An example of resolvable discussion

Discussion 39 **Commits 4** Changes 5 1/11 discussions resolved

12 Dec, 2017 1 commit

add fibonacci number generator.
michael.bergeron committed 9 minutes ago e2f9b582

09 Nov, 2017 3 commits

add six file
michael.bergeron committed a month ago 08d8bd91

Discussion 40 **Commits 4** **Changes 5** 1/12 discussions resolved

Only comments from the following commit are shown below. Show latest version of the diff

add fibonacci number generator.
michael.bergeron committed 13 minutes ago e2f9b582 View details

Showing 1 changed file with 1 additions and 0 deletions Hide whitespace changes Inline Side-by-side

```

fibonacci.rb 0 → 100644
1 + def fib n;n==0?0:n==1?0,1:(2..n-1).inject({0,1}){|a|a<<[a[-1]+a[-2]]};end

```

Administrator @root commented less than a minute ago Owner
This one-liner looks funny, can you split it?
Edited less than a minute ago by Administrator

Reply... Resolve discussion

Discussions

- Jumping between unresolved discussions
- Marking a comment or discussion as resolved
- Move all unresolved discussions in a merge request to an issue

Fast-Forward Merge requests

Fast-Forward Merge requests

- Linear Git history
- No merge commits
- *Settings* → *'Merge method'* → *Fast-forward merge*

Semi-Linear history merge requests

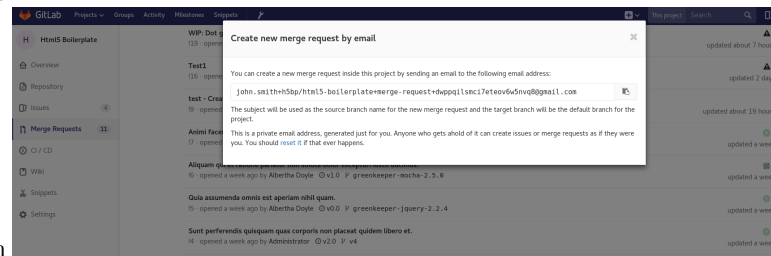
Semi-Linear history merge requests

- Merge commit for every merge
- Merged only if FF possible
- Ensures success by pipeline passing
- *Merge Requests: Merge method* → *Merge commit with semi-linear history*

Creation of new merge requests by email

Creation of new merge requests by email

- By clicking on “*Email a new merge request to this project*” button
- The subject – the source branch name



- The message body – the MR description