



jak przy commit tagu sprawdzić czy commit znajduje się na odpowiedniej gałęzi

Aby sprawdzić, czy commit z tagiem znajduje się na odpowiedniej gałęzi (np. master) w GitLab CI/CD, musisz użyć dodatkowego skryptu weryfikacyjnego, ponieważ zmienna `$CI_COMMIT_BRANCH` jest pusta dla pipeline'ów uruchamianych przez tagi. Oto kilka sprawdzonych metod: [\[1\]](#) [\[2\]](#)

Metoda 1: git merge-base --is-ancestor

Najprostsze rozwiązanie używa komendy `git merge-base` do sprawdzenia, czy commit jest przodkiem danej gałęzi: [\[2\]](#) [\[1\]](#)

```
release-job:
  stage: release
  rules:
    - if: $CI_COMMIT_TAG
  before_script:
    - git fetch origin master:master
  script:
    - |
      if ! git merge-base --is-ancestor $CI_COMMIT_SHA origin/master; then
        echo "Tag nie znajduje się na gałęzi master"
        exit 1
      fi
    - echo "Tag $CI_COMMIT_TAG jest na master - kontynuuję deployment"
    - echo "Deploying production version $CI_COMMIT_TAG"
```

Ta metoda najpierw pobiera aktualny stan brancha master, a następnie sprawdza czy commit oznaczony tagiem (`$CI_COMMIT_SHA`) jest przodkiem gałęzi master. [\[3\]](#) [\[2\]](#)

Metoda 2: git log z grep

Alternatywnie możesz użyć `git log` z `grep` do sprawdzenia, czy tag należy do określonej gałęzi: [\[1\]](#)

```
release-job:
  stage: release
  rules:
    - if: $CI_COMMIT_TAG
  before_script:
    - apk add --no-cache git # jeśli używasz obrazu bez git
    - git fetch origin --tags --prune
  script:
```

```
- |
  if ! git log -1 --pretty='%D' $CI_COMMIT_TAG | grep -q 'origin/master'; then
    echo "Tag nie należy do gałęzi master"
    exit 1
  fi
- echo "Tag jest na master - deployment..."
```

Ta metoda sprawdza dekorator commita i szuka w nim odniesienia do `origin/master`.^[1]

Metoda 3: `git branch --contains`

Możesz również użyć komendy `git branch --contains` do wylistowania wszystkich gałęzi zawierających dany commit:^[4] ^[3]

```
release-job:
  stage: release
  rules:
    - if: $CI_COMMIT_TAG
  script:
    - git fetch origin
    - |
      if ! git branch -r --contains $CI_COMMIT_SHA | grep -E '^([:space:]]origin/master$'; then
        echo "Commit nie znajduje się na gałęzi master"
        exit 1
      fi
    - echo "Deployment wersji $CI_COMMIT_TAG"
```

Flaga `-r` sprawdza gałęzie zdalne, a `grep -E` używa dokładnego dopasowania z wyrażeniem regularnym, aby uniknąć fałszywych pozytywnych dla gałęzi o podobnych nazwach.^[3]

Najlepsza praktyka

Najprostszym i najbezpieczniejszym rozwiązaniem jest połączenie `git merge-base --is-ancestor` z odpowiednią polityką ochrony brancha. Powinieneś również skonfigurować w GitLab ochronę brancha master, aby tagi mogły być tworzone tylko na tym branchu przez upoważnionych użytkowników.^[2]



1. <https://stackoverflow.com/questions/78201569/how-to-run-a-gitlab-ci-job-only-if-a-tag-is-created-on-the-main-branch>
2. <https://www.codegenes.net/blog/how-to-run-a-gitlab-ci-yml-job-only-on-a-tagged-branch/>
3. <https://stackoverflow.com/questions/43535132/given-a-commit-id-how-to-determine-if-current-branch-contains-the-commit>
4. <https://labex.io/tutorials/git-how-to-check-if-a-git-commit-is-in-a-specific-branch-560059>
5. <https://graphite.com/guides/how-to-check-out-git-tags>
6. <https://stackoverflow.com/questions/72895047/gitlab-ci-check-if-branch-is-a-tag-or-not-and-then-execute-commands-based-on-branch>
7. <https://docs.gitlab.com/user/project/repository/tags/>

8. <https://travis-ci.community/t/if-tag-is-present-does-not-work/11934>
9. <https://stackoverflow.com/questions/1862423/how-to-tell-which-commit-a-tag-points-to-in-git>
10. <https://gitlab.com/gitlab-org/gitlab/-/issues/297508>