



CI/CD với Jenkins

Nguyễn Hàn Duy

duy@techmaster.vn

Nội dung



- Gửi email thông báo kết quả build
- Tích hợp với Sonarqube
- Multi-branch pipeline

Gửi email thông báo

Gửi email báo kết quả build



- Đây là step luôn luôn được chạy sau mỗi lần build CI, không cần biết quá trình build là success hay failure
- Được định nghĩa trong phần post trong Jenkinsfile

Tham khảo:

- <https://www.jenkins.io/doc/pipeline/tour/post/>
- <https://www.jenkins.io/doc/book/pipeline/syntax/#post>

Task #1: Setup gửi email thông báo



Yêu cầu



- Cài đặt plugin Email Extension Plugin
- Ở giao diện Jenkins/Configure System, mục Extended E-mail Notification, cấu hình thông tin kết nối tài khoản Gmail
- Ở giao diện Jenkins/Configure System, mục Jenkins Location, cấu hình thông tin System Admin e-mail address
- Ở giao diện Pipeline Syntax, chọn Sample Step là emailext: Extended Email để tạo cú pháp cho Jenkinsfile step

Extended E-mail Notification

Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

SMTP Username

handuy1992@gmail.com

SMTP Password

.....

Use SSL

☒

Advanced Email Properties

Default user e-mail suffix

Advanced...

Default Content Type

Plain Text (text/plain)

List ID

Add 'Precedence: bulk' E-mail Header

☐

Save

Apply

Task #2: Thêm bước gửi email vào Jenkinsfile



Yêu cầu



Trong mục post của Jenkinsfile, thêm 1 step gửi email thông báo kết quả build. Step này luôn luôn được chạy

Task #3: Unit test cho ứng dụng NodeJS



Yêu cầu



Cho source code: <https://github.com/handuy/nodejs-test>

1. Clone source code, xóa thư mục .git. Tạo 1 remote repo tại Gitlab. Kết nối Gitlab repo với Jenkins
2. Viết Jenkinsfile gồm các bước chạy ở node slave:
 - Clone source code
 - Cài đặt dependencies của app bằng lệnh: **npm install**
 - Khởi động app bằng lệnh: **nohup node index.js &**
 - Chạy script kiểm thử bằng lệnh: **npm test**
 - Ghi lại kết quả test bằng plugin **junit** (kết quả test được lưu trong file test.xml đặt ở root project folder) và gửi email thông báo
 - Nếu các test case pass thì đóng gói thành docker image và push image lên Gitlab private registry
3. Push code lên Gitlab repo và kiểm tra quá trình chạy Jenkins

Tích hợp Jenkins + SonarQube



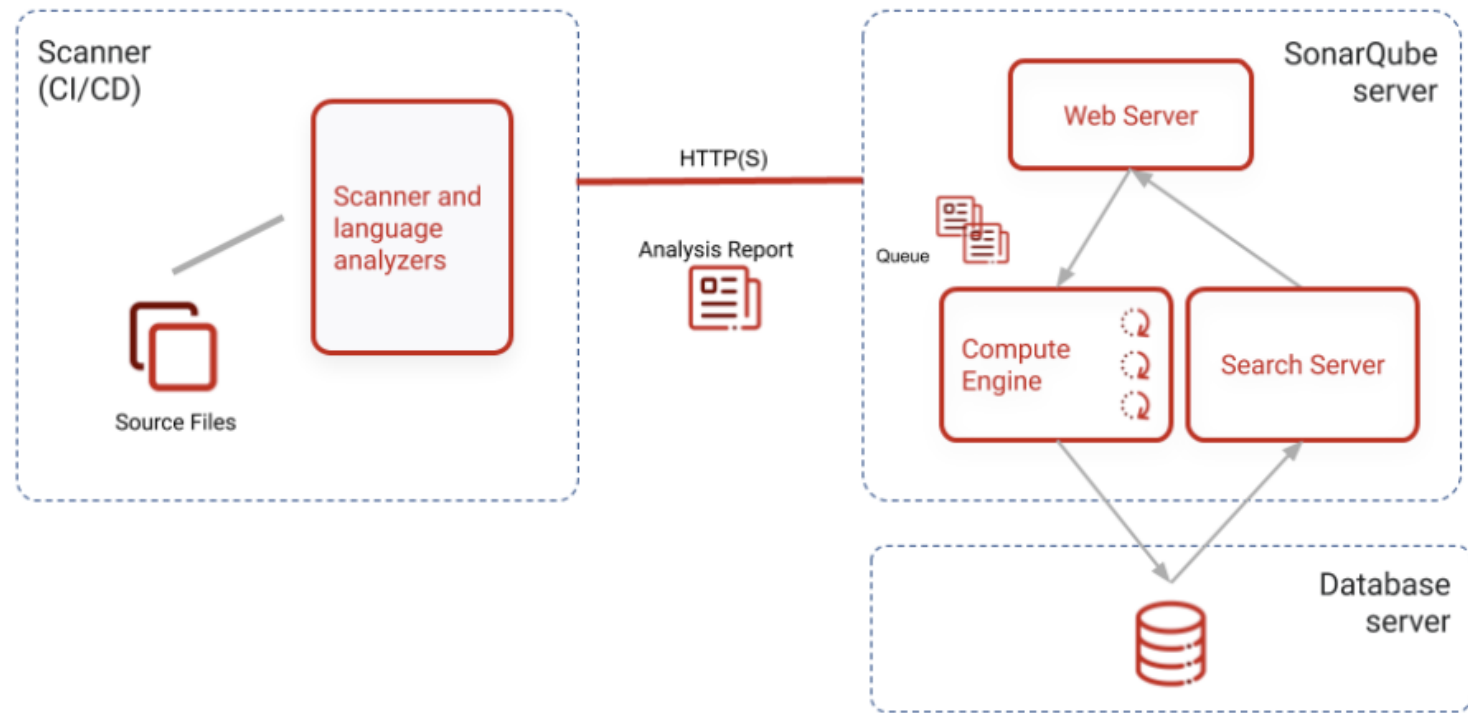
SonarQube

- Công cụ kiểm soát chất lượng mã nguồn
- Tự động chạy static code analysis, phát hiện các vi phạm liên quan đến coding convention (naming convention, security holes, potential memory leaks, ...)



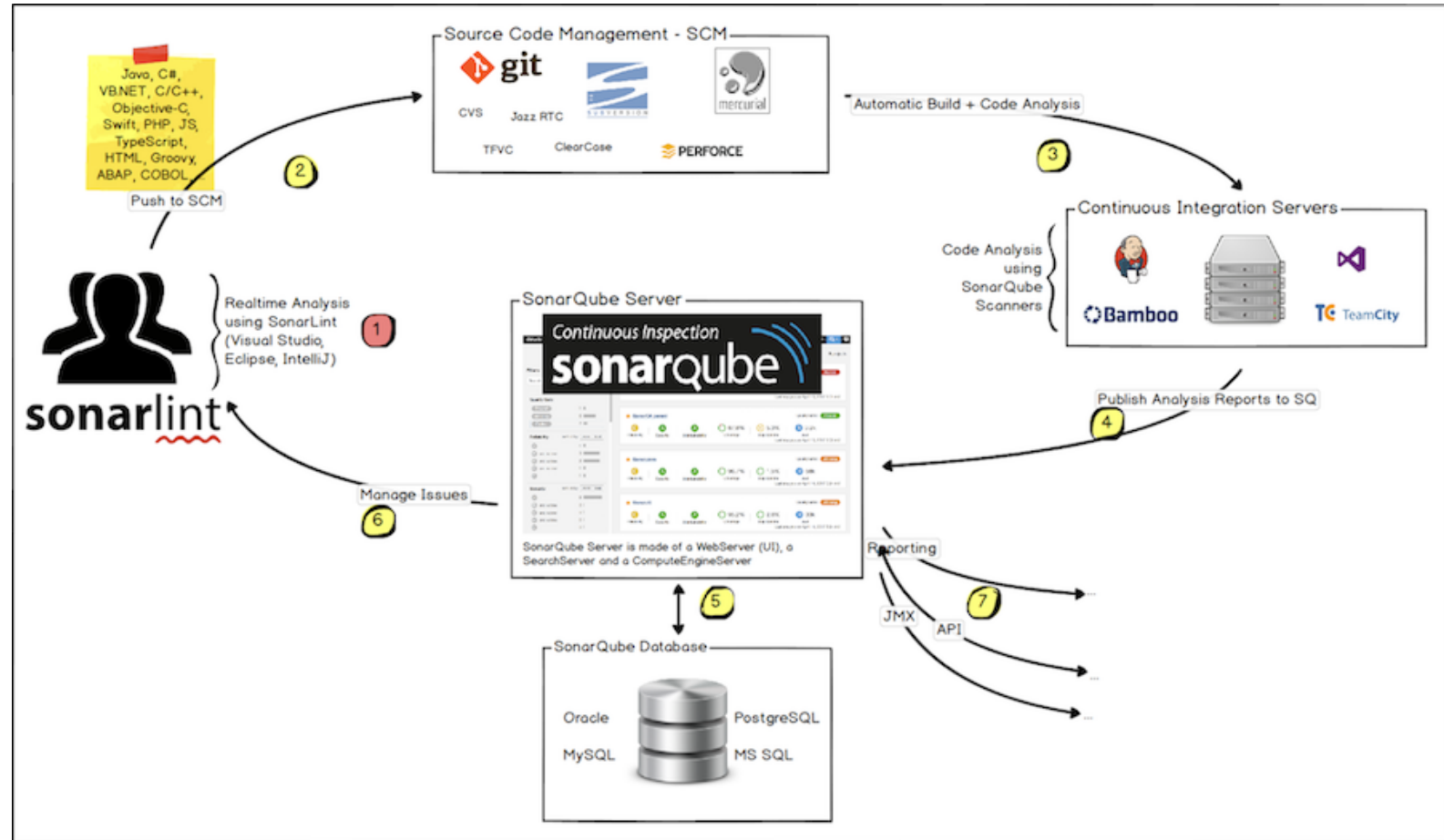
Kiến trúc SonarQube

- Scanner thực hiện phân tích source code và gửi kết quả phân tích sang server
- Server hiển thị và quản lý kết quả phân tích
- Database lưu kết quả phân tích, cấu hình server



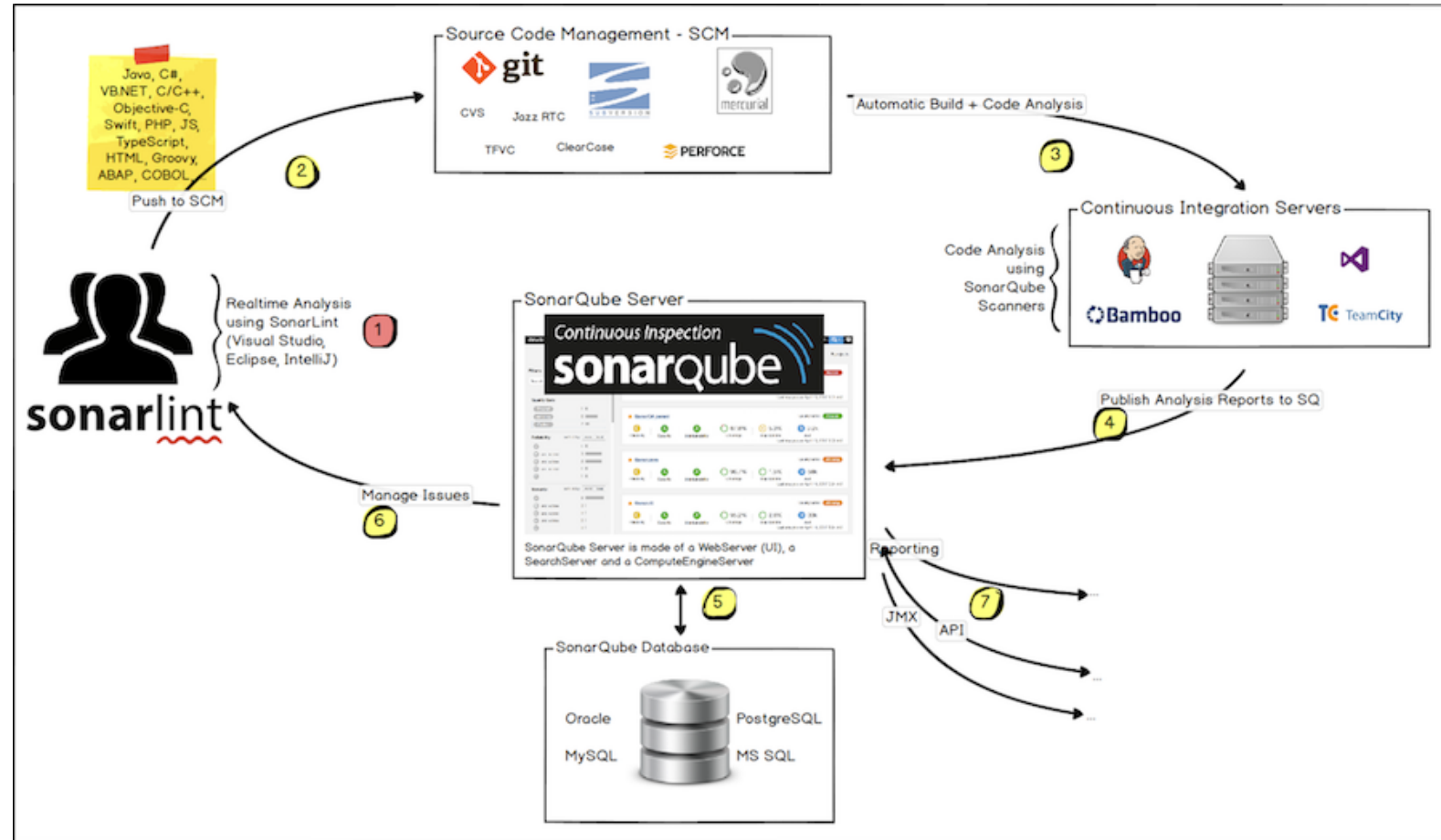
Tích hợp Jenkins + SonarQube

- Dev push code lên SCM
- Jenkins chạy sonar-scanner phân tích code
- Jenkins gửi kết quả phân tích sang Sonarqube server
- Server lưu kết quả phân tích và gửi kết quả (JSON) cho các service khác



Tích hợp Jenkins + SonarQube

- Cài đặt SonarQube server
- Cài đặt SonarQube scanner vào Jenkins
- Thiết lập kết nối giữa Jenkins và Sonarqube server
- Bổ sung step phân tích source code vào Jenkins pipeline



Task #4:

Cài đặt SonarQube server

Yêu cầu



- Cài đặt Sonarqube server trên 1 máy cloud VM bằng Docker Compose theo hướng dẫn tại:
<https://docs.sonarqube.org/latest/setup/install-server/>
- Truy cập SonarQube bằng cặp username/password là admin/admin

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 8.7 (build 41497) - LGPL v3 - Community - Documentation - Plugins - Web API - About

Task #5: Cài đặt SonarQube scanner vào Jenkins



Yêu cầu



1. Cài đặt plugin **SonarQube Scanner for Jenkins**
2. Ở giao diện Manage Jenkins/Global Tool Configuration, thêm SonarQube Scanner tool

Giao diện Manage Jenkins/Global Tool Configuration

SonarQube Scanner installations

Add SonarQube Scanner



SonarQube Scanner

Name

sonarscan



Install automatically



Install from Maven Central

Version

SonarQube Scanner 4.6.0.2311



Delete Installer

Add Installer



Delete SonarQube Scanner

Add SonarQube Scanner

List of SonarQube Scanner installations on this system

Task #6: Kết nối SonarQube server và Jenkins



Kết nối Jenkins đến Sonarqube



1. Ở giao diện Sonarqube, tạo access token cho user admin.
2. Ở giao diện Jenkins, tạo 1 credential dưới dạng **secret text** cho Sonarqube token
3. Ở giao diện Jenkins/Configure Systems, thêm thông tin kết nối Sonarqube server

Kết nối Sonarqube đến Jenkins

Ở giao diện Sonarqube, route
/admin/webhooks, tạo mới 1
webhook trỏ đến Jenkins:
<Jenkins-url>/sonarqube-webhook/

Create Webhook

Name*



URL*



Server endpoint that will receive the webhook payload, for example:
"http://my_server/foo". If HTTP Basic authentication is used, HTTPS is
recommended to avoid man in the middle attacks. Example:
"https://myLogin:myPassword@my_server/foo"

Secret

If provided, secret will be used as the key to generate the HMAC hex
(lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256'
header

Create

Cancel

Task #7: Thêm step phân tích Sonarqube vào Jenkinsfile



Yêu cầu Jenkinsfile



1. Clone source code:
2. Thêm 2 stage:
 - Sử dụng Sonarqube scanner phân tích source code
 - Ghi nhận kết quả phân tích được gửi về từ Sonarqube server
3. Gửi email thông báo kết quả

Stage: Sonarqube scanner phân tích source code



1. Định nghĩa biến môi trường lưu đường dẫn thư mục chứa sonarqube scanner

environment {

scannerHome = tool 'sonarscan'

}

2. Sử dụng step withSonarQubeEnv + tên của Sonarqube server:

withSonarQubeEnv('tên-sonarqube-server-ở-mục-jenkins-configure-system') {

sh "\${scannerHome}/bin/sonar-scanner"

}

Stage: Lấy kết quả phân tích từ Sonarqube server



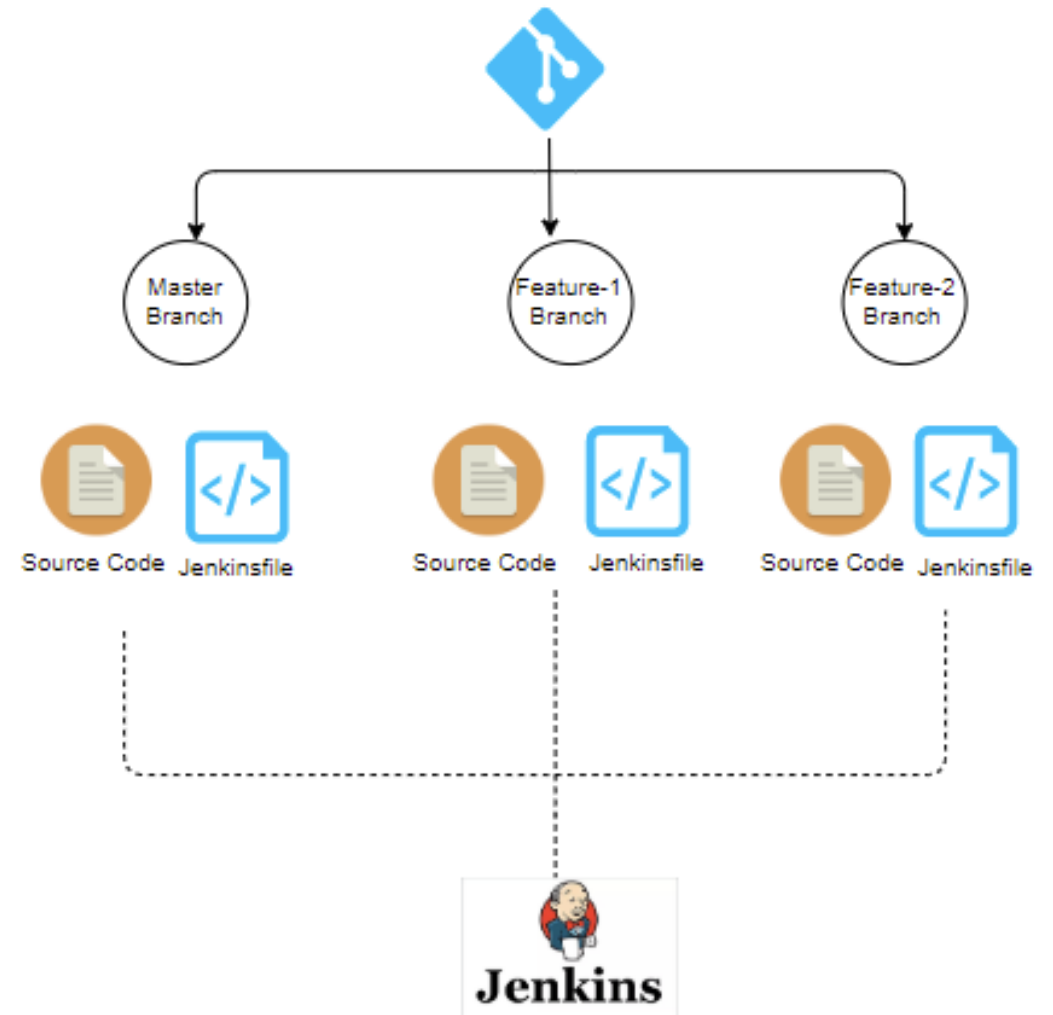
Sử dụng step `waitForQualityGate`:

`waitForQualityGate abortPipeline: true`

Multi-branch pipeline

Multi-branch pipeline

Chạy các kịch bản CI-CD khác nhau cho mỗi nhánh trong Git repo, dựa vào nội dung của file Jenkinsfile



Task #8: Tạo Jenkins multi-branch pipeline item



Yêu cầu



- Tạo 1 Github repo mới đặt tên là nodejs-todolist-multi, vẫn lấy source code của ứng dụng NodeJS todo list
- Kết nối Github repo với Jenkins thông qua webhook


Yêu cầu

- Từ giao diện trang chủ Jenkins, chọn New Item
- Điền tên item, chọn kiểu Multibranch Pipeline rồi bấm OK


Enter an item name

nodejs-todolist-multi


» Required field




Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.




Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.




Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.



Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:

OK

Type to autocomplete

Yêu cầu



- Mục Branch Sources: Chọn Github
- Repository HTTPS URL: Copy paste link Github repo nodejs-todolist-multi (ví dụ: <https://github.com/handuy/handuy.github.io.git>)
- Phần Behaviours, Discover branches: All branches
- Phần Build Configuration: Để mặc định là Jenkinsfile
- Cuối cùng bấm OK

Yêu cầu

Branch Sources

GitHub

Credentials

- none -

Add

Credentials are recommended

Repository HTTPS URL

Repository HTTPS URL

https://github.com/handuy/handuy.github.io.git

Credentials ok. Connected to https://github.com/handuy/handuy.github.io.

Validate

Repository Scan - Depreciated Visualization

Behaviours

Discover branches

Strategy

All branches

Discover pull requests from origin

Strategy

Merging the pull request with the current target branch revision

Discover pull requests from forks

Strategy

Merging the pull request with the current target branch revision

Trust

From users with Admin or Write permission

May not be supported on older versions of GitHub Enterprise. See help button.

Add

Property strategy

All branches get the same properties

Save

Apply

Add property

Task #9: Tạo thêm nhánh development cho Github repo



Yêu cầu



Jenkinsfile trên nhánh dev:

- Lấy source code từ nhánh dev
- Thực hiện phân tích Sonarqube
- Chạy unit test
- Gửi email thông báo kết quả build

Jenkinsfile trên nhánh master:

- Lấy source code từ nhánh master
- Build docker image và push lên Gitlab private registry