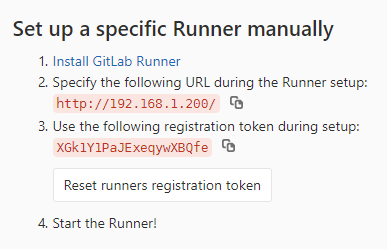
**THIẾT LẬP LUỒNG CI CHO PROJECT GATEWAY**

Đầu tiên, chúng ta thực hiện lấy code của Gateway về server 192.168.1.201, thư mục mặc định của các project cấu hình CI đó là /root/GitlabRunner/repo. Checkout code và chuyển sang nhánh dev (vì luồng CI hiện tại sẽ thiết lập cho nhánh dev).

***1. Cài đặt Gitlab Runner (xem hướng dẫn ở tài liệu CI-CD mục 2.1.2 hôm demo)***

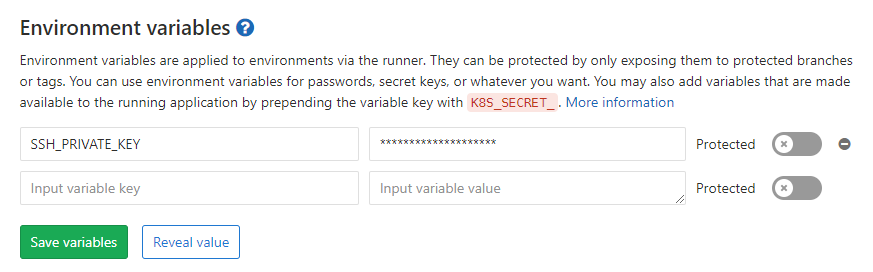
Đối với project Gateway, Gitlab runner command đã được cài sẵn trên 192.168.1.201 rồi, nên ta chỉ cần đăng kí thêm một runner mới (register a runner) cho project Gateway bằng cách sử dụng lệnh ***gitlab-runner register***.

URL và token của project ta sẽ lấy ở *Setting >> CI-CD >> Runner* của project Gateway trên git (trong quá trình đăng kí runner sẽ cần 2 thành phần này).



Sau đó ta vào phần cài đặt của runner vừa tạo, check vào mục ***Run untag jobs***.

Bước tiếp theo là phải tạo và điền SSH\_PRIVATE\_KEY cho project để runner (trên server 192.168.1.201) có thể nắm bắt được sự kiện xảy ra trên Gitlab server thông qua SSH key được mã hóa này. Nhưng trong thư mục */root/.ssh* đã có key sẵn rồi, ta copy nội dung của file ***id\_rsa\_truongbb96*** (key tạo cho các runner sử dụng) vào phần *Setting >> CI-CD >> Enviroment variables*, trường *Input variable key* ta điền “***SSH\_PRIVATE\_KEY***”, trường *Input variable value* ta điền giá trị private key vừa copy, sau đó lưu lại.



Ok, đã xong Runner, tiếp theo là cấu hình luồng CI.

***2. Cấu hình trong project***

Luồng CI của project được chia như sau:

- Khi có một push commit:

*+ checkout code*

*+ compile*

*+ package*

*+ sonnar scan*

*+ check quality gate*

- Khi có một accept merge request:

+ *checkout code*

*+ compile*

*+ package*

*+ sonnar scan*

*+ check quality gate*

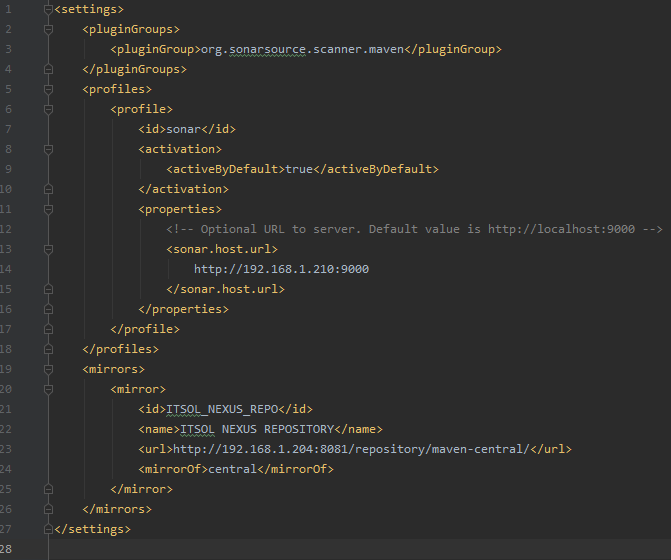
***+ docker build***

***+ docker up***

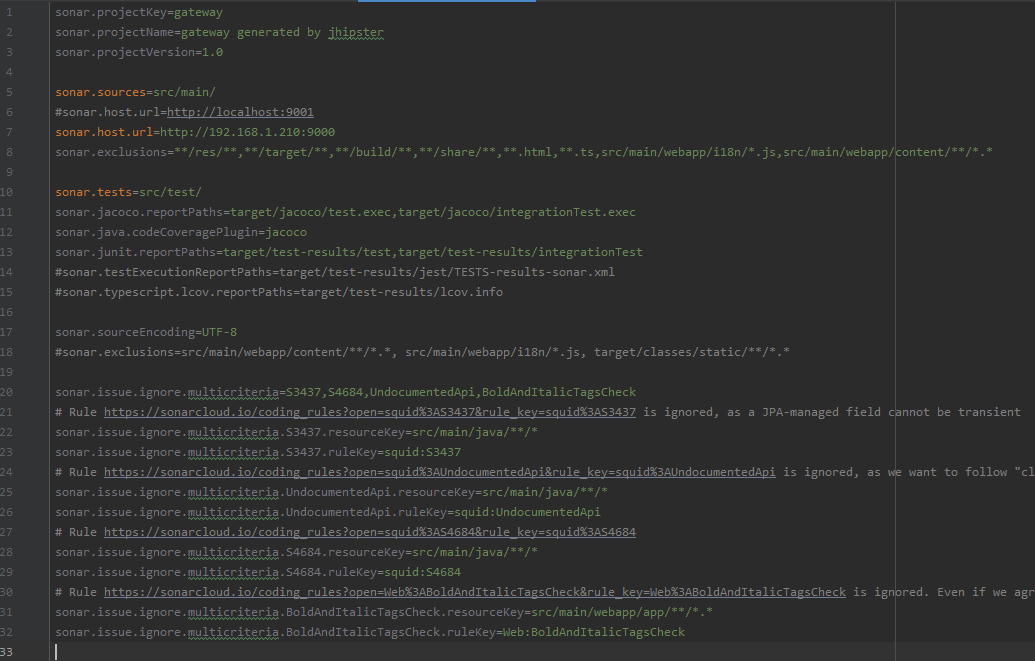
***+ docker push***

2.1 Các file cấu hình chung

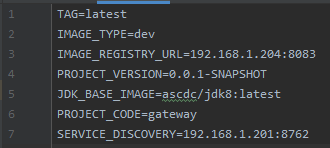
File ***settings.xml*** định nghĩa ra server sonar và Nexus repository.



File ***sonar-project.properties*** định nghĩa ra các thông tin để thực hiện *quét sonnar*.



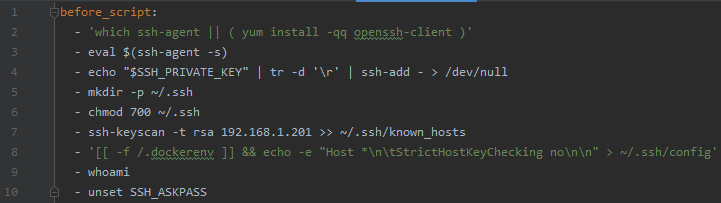
File ***.env*** chứa các biến môi trường cần cho các file cấu hình.



2.1 File ***.gitlab-ci.yml***

Cấu hình chính cho luồng chính là file ***.gitlab-ci.yml*** (giải thích cụ thể xem ví dụ ở tài liệu CI-CD hôm demo). File ***.gitlab-ci.yml*** này sẽ gọi các hàm được viết ở ***build.sh*** để thực hiện chạy lệnh linux. Cụ thể như sau:

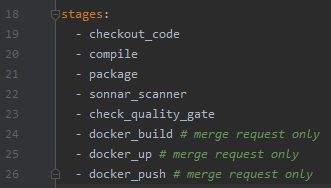
- Đầu tiên là các lệnh thực hiện kết nối tới server 192.168.1.201 được viết ở phần ***before\_script***:



- Các biến global dùng chung trong file ***.gitlab-ci.yml*** được định nghĩa ở phần ***variables***:

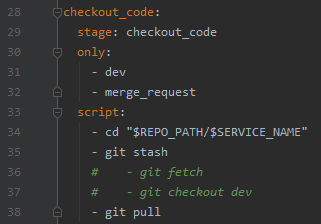


- Định nghĩa các stage sẽ được thực hiện trong luồng CI:

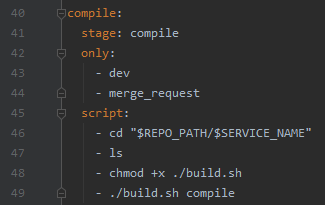


Các bước sau đó là nội dung cụ thể được thực hiện ở mỗi stage.

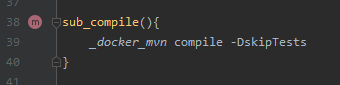
- ***Stage checkout\_code***: bước này thực hiện lấy code mới nhất về thư mục ***/root/GitlabRunner/repo/gateway*** mà ta vừa clone về bên trên. Stage này xuất hiện trong luồng CI khi có một push commit lênh nhánh dev hoặc một accept merge request xảy ra.



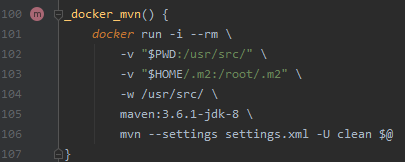
- ***Stage compile***: thực hiện biên dịch code java thành các file class. Stage này xuất hiện trong luồng CI khi có một push commit lênh nhánh dev hoặc một accept merge request xảy ra.



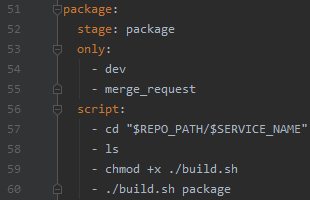
Bước này gọi hàm compile trong file ***build.sh***



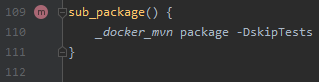
Với ***\_docker\_mvn*** là hàm chạy lệnh docker



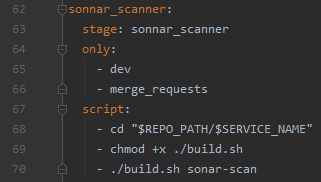
- ***Stage package***: thực hiện đóng gói thành file .jar. Stage này xuất hiện trong luồng CI khi có một push commit lênh nhánh dev hoặc một accept merge request xảy ra.



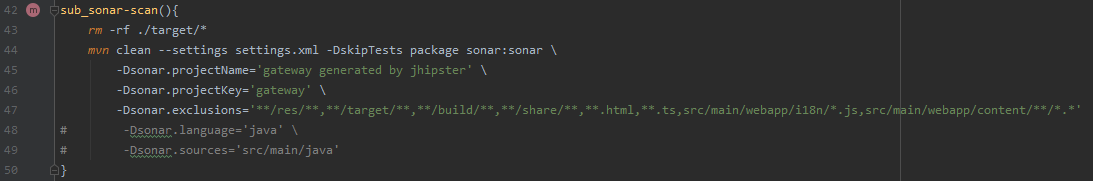
Với hàm package được gọi từ ***build.sh***



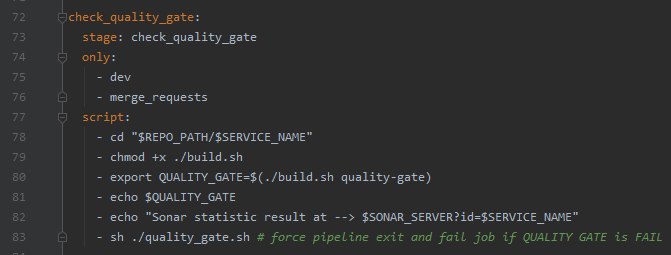
- ***Stage sonar\_scanner***: Thực hiện quét sonnar với lệnh quét được viết trong hàm sonar-scan của file ***build.sh***. Stage này xuất hiện trong luồng CI khi có một push commit lênh nhánh dev hoặc một accept merge request xảy ra.



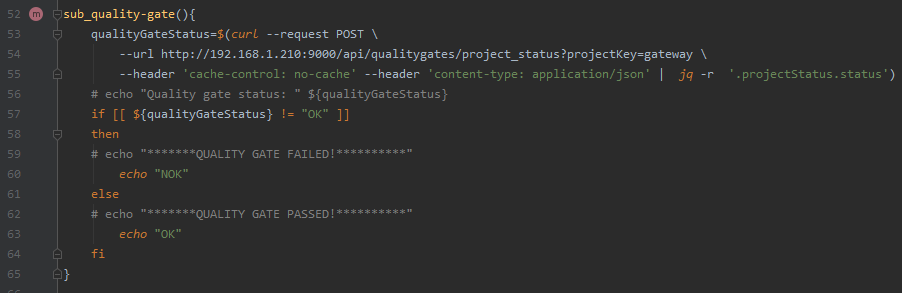
Hàm này có các tham số cấu hình thực hiện quét. Mặc dù đã định nghĩa ở file ***sonar-project.properties*** nhưng không nhận, phải truyền vào câu lệnh như hình dưới đây (vì thấy config của file ***sonar-project.properties*** không ăn nên đã thử xóa file đó đi và kết quả là luồng CI chết ở bước compile :v 🡪 không hiểu lắm)



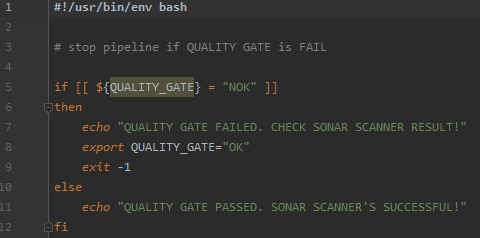
- ***Stage check\_quality\_gate***: Thực hiện kiểm tra kết quả quét sonar ở bước trên bằng một WS được gọi ở bên ***buid.sh***. Stage này xuất hiện trong luồng CI khi có một push commit lênh nhánh dev hoặc một accept merge request xảy ra.



WS bên ***build.sh***



Và sử dụng file ***quality\_gate.sh*** để kiểm tra kết quả



- ***Docker stages***: Cả 3 bước docker phía sau (dành cho sự kiện accept MR) đều gọi hàm tương ứng được viết bên ***build.sh***. Quan trọng nhất là docker được cấu hình tại file ***src/main/docker/Dockerfile*** và ***src/main/docker/app.yml***. Bước ***docker-build*** sẽ thực hiện build project thành docker image và tag image đó. Bước ***docker-up*** sẽ thực hiện start web lên cổng 80 (được cấu hình trong ***app.yml***), bước ***docker-push*** sẽ thực hiện push docker image vừa build lên Nexus repository.