

Sonarqube-Github Workflow에 Sonarscanner 작업 만들기

이 문서는 Sonarqube-Gitlab Workflow에 Sonarscanner 작업 만들기 가이드를 공유하기 위해 작성되었다.

도구명	Sonarqube, Github
버전	9.5
OS	CentOS 8

- [Github Self-hosted Runner 만들기](#)
 - 초기 서버 설정 및 설치
 - Sonarscanner 설치 및 설정
 - Github에 Self-hosted Runner 등록하기
- [Github Action](#)
 - 비밀키 생성하기
 - Sonarqube 용 Branch 생성
 - Workflow 생성
 - Java Maven용 Action yml 파일
 - NodeJs용 Action yml 파일
 - C/C++/Objective-C용 Action yml 파일
 - 확인
 - Branch 삭제
- [참조 링크](#)

Github Self-hosted Runner 만들기

초기 서버 설정 및 설치

- 시간 확인 및 파일 읽기 수 설정
- JDK11
- NodeJS 18
- Maven 3.8.6 설치
- cmake 설치

```

#
date
#
timedatectl list-timezones | grep Seoul
sudo timedatectl set-timezone Asia/Seoul

vi /etc/profile
#
    ulimit -n 131072
    ulimit -u 8192

#
ulimit -n
ulimit -u

#JDK 11
wget 'https://github.com/AdoptOpenJDK/openjdk11-binaries/releases/download/jdk-11.0.7%2B10/OpenJDK11U-jdk_x64_linux_hotspot_11.0.7_10.tar.gz' -O /opt/OpenJDK11U-jdk_x64_linux_hotspot_11.0.7_10.tar.gz
#JDK 11
cd /opt
chmod 700 OpenJDK11U-jdk_x64_linux_hotspot_11.0.7_10.tar.gz
tar -xvf OpenJDK11U-jdk_x64_linux_hotspot_11.0.7_10.tar.gz
mkdir javall
mv jdk-11.0.7+10/ jdk11
mv jdk11 /opt/
#
vi /etc/profile
#
    JAVA_HOME=/opt/jdk11
    PATH=$PATH:$JAVA_HOME/bin
    export JAVA_HOME
    export PATH
source /etc/profile

#JDK
echo $JAVA_HOME
java -version

#NodeJS
sudo curl -fsSL https://rpm.nodesource.com/setup_18.x | bash -
sudo yum install -y nodejs

#Maven
cd /opt
wget https://archive.apache.org/dist/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.tar.gz
chmod 700 apache-maven-3.8.6-bin.tar.gz
tar -xvf apache-maven-3.8.6-bin.tar.gz
ln -s apache-maven-3.8.6 maven
#
    MAVEN_HOME=/opt/jdk11
    PATH=$PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin
    export JAVA_HOME MAVEN_HOME
source /etc/profile
#Maven
echo $MAVEN_HOME
mvn -v

#CFamily (cmake)
yum install cmake -y
yum install gcc-c++

```

Sonarscanner 설치 및 설정

- Maven Sonarqube 설정
- Sonarscanner 다운로드
- Sonarscanner Cfamily 다운

```

#Maven Sonarqube
cd $MAVEN_HOME/conf
vim settings.xml
#   (ex: PluginGroups  PluginGroup)
<settings>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
  <profiles>
    <profile>
      <id>sonar</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <sonar.sourceEncoding>UTF-8</sonar.sourceEncoding>
      </properties>
    </profile>
  </profiles>
</settings>

#Sonarscanner
cd /opt
wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.7.0.2747-linux.zip
chmod 755 sonar-scanner-cli-4.7.0.2747-linux.zip
unzip sonar-scanner-cli-4.7.0.2747-linux.zip
#
ln -s sonar-scanner-4.7.0.2747-linux/ sonar-scan

#Sonarscanner CFamily
cd /opt/
wget {SonarQube URL}/static/cpp/build-wrapper-linux-x86.zip
chmod 755 build-wrapper-linux-x86.zip
unzip build-wrapper-linux-x86.zip
#
ln -s build-wrapper-linux-x86 build-wrapper

#
vi /etc/profile
#
  export SONAR_SCAN=/opt/sonar-scan
  export SONAR_CSCAN=/opt/build-wrapper
  #PATH
  PATH=$PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin:$SONAR_SCAN/bin:$SONAR_CSCAN
  source /etc/profile
#
echo $SONAR_SCAN
echo $SONAR_CSCAN

#
#Runner  Command
shasum

```

Github에 Self-hosted Runner 등록하기

- Actions 권한 확인
Settings → Actions → General → Actions permissions

The screenshot shows the GitHub repository settings for 'wotn112/serverless-stack-client'. A red box highlights the 'Actions' section under the 'General' tab. Within this section, another red box highlights the 'Actions permissions' tab. A blue circle labeled '1' is on the 'Settings' tab at the top. A blue circle labeled '2' is on the 'Actions' tab in the sidebar. A blue circle labeled '3' is on the 'General' tab in the sidebar. A blue circle labeled '4' is on the 'Actions permissions' tab.

Actions permissions

- Allow all actions and reusable workflows

Any action or reusable workflow can be used, regardless of who authored it or where it is defined.
- Disable actions

The Actions tab is hidden and no workflows can run.
- Allow wotn112 actions and reusable workflows

Any action or reusable workflow defined in a repository within wotn112 can be used.
- Allow wotn112, and select non-wotn112, actions and reusable workflows

Any action or reusable workflow that matches the specified criteria, plus those defined in a repository within wotn112, can be used. [Learn more about allowing specific actions and reusable workflows to run](#).

Artifact and log retention

This is the duration that artifacts and logs will be retained. [Learn more](#).

90 days Save

Fork pull request workflows from outside collaborators

Choose which subset of outside collaborators will require approval to run workflows on their pull requests. [Learn more about approving workflow runs from public forks](#).

- Require approval for first-time contributors who are new to GitHub

Only first-time contributors who recently created a GitHub account will require approval to run workflows.
- Require approval for first-time contributors

Only first-time contributors will require approval to run workflows.
- Require approval for all outside collaborators

- 저장소 기준: 저장소 → Settings → Actions → Runner → New Self-hosted runner root로 실행불가

The screenshot shows the GitHub repository settings for 'wotn112/serverless-stack-client'. A red box highlights the 'Runners' section under the 'General' tab. Another red box highlights the 'New self-hosted runner' button. A blue circle labeled '1' is on the 'Settings' tab at the top. A blue circle labeled '2' is on the 'Actions' tab in the sidebar. A blue circle labeled '3' is on the 'Runners' tab in the sidebar. A blue circle labeled '4' is on the 'New self-hosted runner' button.

Runners

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. [Learn more about self-hosted runners](#).

There are no runners configured

[Learn more about using runners](#) to run actions on your own servers.

- Architecture : 해당 서버에서 uname -m 으로 확인
해당 페이지에 나오는 대로 명령어 입력

The screenshot shows the GitHub Actions Runner configuration interface. On the left, a sidebar lists various settings like General, Access, and Security. The main area is titled 'Runners / Create self-hosted runner'. It includes sections for 'Architecture' (set to x64), 'Download' (with shell scripts for macOS, Linux, and Windows), 'Configure' (with shell scripts for runner setup), and 'Using your self-hosted runner'. A red box highlights the 'Architecture' dropdown.

```
# Runner removal
# Enter runner remove token: *****
# http response code: NotFound from 'POST https://api.github.com/actions/runner-registration'
# ('message': 'Not Found', 'documentation_url': 'https://docs.github.com/rest')
# failed: Removing runner from the server
# response status code does not indicate success: 404 (Not Found).
[atlassian@jk actions-runner]$ ./config.sh remove
[atlassian@jk actions-runner]$ ./config.sh remove --token AP2IUXYHQNWNPC437VWIG53C4EHP4

# Runner removal

# Runner removed successfully
# Removed .credentials
# Removed .runner

[atlassian@jk actions-runner]$ ./config.sh --url https://github.com/wotnll2/sonarqubeCrunner --token AP2IUX73RWP2IBMEP7RFDG3C4EHRM

[GitHub logo] Self-hosted runner registration

-----[redacted]-----
# Authentication
# Connected to GitHub
# Runner Registration
Enter the name of the runner group to add this runner to: [press Enter for Default]
Enter the name of runner: [press Enter for jk] sonar
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip] sonar
# Runner successfully added
# Runner connection is good
# Runner settings
Enter name of work folder: [press Enter for _work]
# Settings Saved.

[atlassian@jk actions-runner]$
```

- 해당 서버에 ./run.sh 실행(백그라운드 실행 시 ./run.sh &)

서비스로 변경(sudo ./svc.sh install USERNAME)

시작(sudo ./svc.sh start)

```
[atlassian@jk actions-runner]$ ./run.sh
✓ Connected to GitHub

Current runner version: '2.294.0'
2022-07-26 09:14:24Z: Listening for Jobs
```

- Settings → Actions → Runners에서 상태 확인

The screenshot shows the GitHub repository settings for 'wohn112/serverless-stack-client'. The 'Runners' section is highlighted with a red box. The left sidebar has 'Actions' selected under 'Runners'. The main area shows a table with one runner named 'sonar' which is 'Idle'. There is also a 'New self-hosted runner' button.

Github Action

비밀키 생성하기

- Settings → Secrets → Actions → New repository secret 선택

The screenshot shows the GitHub repository settings page for 'wotn112/serverless-stack-client'. The 'Settings' tab is selected. In the left sidebar, under the 'Security' section, the 'Secrets' item is highlighted with a red box and labeled '2'. Under 'Secrets', the 'Actions' item is also highlighted with a red box and labeled '3'. At the top right, there is a button labeled 'New repository secret' with a red box around it and the number '4' above it.

- SONAR_URL: sonarqube Url
- SONAR_TOKEN: sonarqube token
- SONAR_USER: sonarqube ID
- SONAR_PASSWORD: 해당 ID password

The screenshot shows the SonarQube Administration interface. The top navigation bar includes 'Projects', 'Portfolios', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. The user is logged in as 'Administrator'. The main content area has two tabs: 'Tokens' and 'Enter a new password'. The 'Tokens' tab displays a table of generated tokens:

Name	Type	Project	Last use	Created	Action
Bamboo	User		1 month ago	June 23, 2022	<button>Revoke</button>
test2323	User		< 1 hour ago	July 27, 2022	<button>Revoke</button>

The 'Enter a new password' tab contains fields for 'Old Password *', 'New Password *', and 'Confirm Password *', along with an 'Update' button.

The screenshot shows the GitHub repository settings for 'Actions secrets'. On the left, the sidebar has sections like General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Actions, Webhooks, Environments, Pages), Security (Code security and analysis, Deploy keys, Secrets - selected), Integrations (GitHub apps, Email notifications). The main content area is titled 'Actions secrets' and says 'There are no secrets for this repository's environments.' Below it is a section for 'Repository secrets' containing four entries:

Secret	Last Updated	Action
SONAR_PASSWORD	Updated now	<button>Update</button> <button>Remove</button>
SONAR_TOKEN	Updated 34 minutes ago	<button>Update</button> <button>Remove</button>
SONAR_URL	Updated 1 hour ago	<button>Update</button> <button>Remove</button>
SONAR_USER	Updated 10 seconds ago	<button>Update</button> <button>Remove</button>

Sonarqube 용 Branch 생성

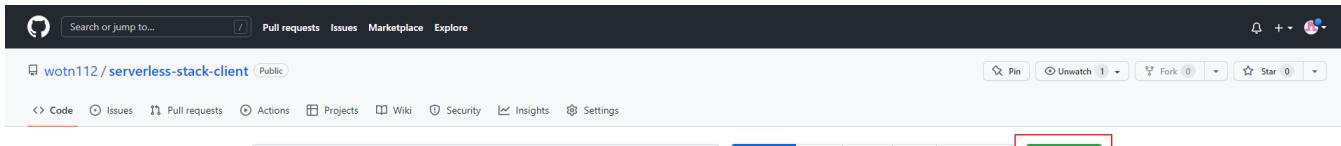
- 해당 저장소에서 branch 선택

The screenshot shows the GitHub repository page for 'wotn112/serverless-stack-client'. The top navigation bar includes Search or jump to..., Pull requests, Issues, Marketplace, Explore, and a user icon. The repository name is 'wotn112/serverless-stack-client' (Public). The main content area shows the 'Code' tab selected, displaying 1 branch and 0 tags. The branch 'master' is selected. The commit history for 'Daud Setting up our React app' is shown, with 3 commits from fcd12a0 on 5 Oct 2020. The commits are:

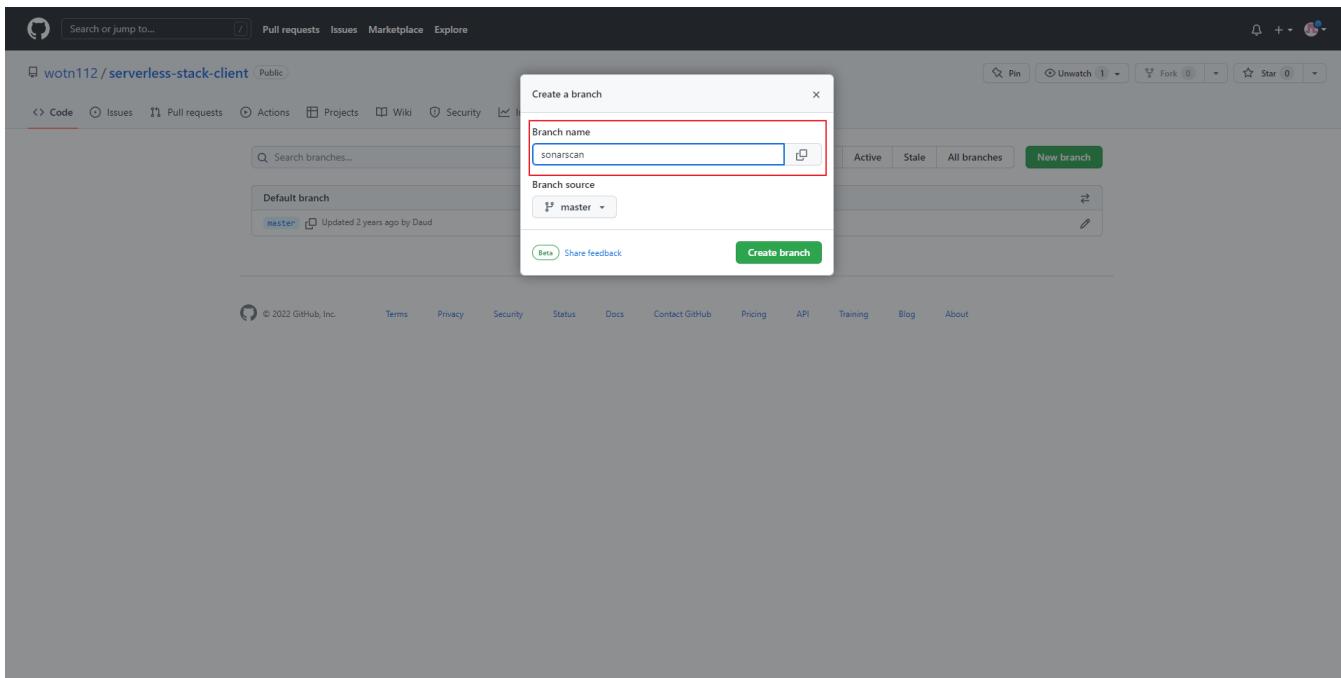
Commit	Message	Date
public	Setting up our React app	2 years ago
src	Setting up our React app	2 years ago
.gitignore	Setting up our React app	2 years ago

The 'About' section indicates 'No description, website, or topics provided.' It shows 0 stars, 1 watching, and 0 forks. The 'Releases' section shows 'No releases published. Create a new release.' The 'Packages' section shows 'No packages published. Publish your first package.' The 'Languages' section shows a chart with JavaScript at 70.7%, HTML at 18.4%, and CSS at 10.9%.

- New Branch 선택



- Branch Name: sonarscan



Workflow 생성

- 저장소에서 sonarscan Branch 선택

The screenshot shows a GitHub repository page for 'wotn112 / serverless-stack-client'. The 'Code' tab is selected. A red box highlights the dropdown menu above the file list, which shows 'master' selected. Another red box highlights the 'sonarscan' file in the list.

Branches

- master (selected)
- sonarscan

Files

- sonarscan (Setting up our React app)
- .gitignore
- README.md
- package-lock.json
- package.json

README.md

This project was bootstrapped with Create React App.

Available Scripts

In the project directory, you can run:

```
npm start
```

Runs the app in the development mode.
Open <http://localhost:3000> to view it in the browser.

■ Add File → Create New file 선택

The screenshot shows the same GitHub repository page after a new file has been created. The 'Code' tab is selected. A red box highlights the 'Create new file' button in the dropdown menu above the file list.

Branches

- sonarscan (selected)
- master

Files

- sonarscan (Setting up our React app)
- .public
- .src
- .gitignore
- README.md
- package-lock.json
- package.json

README.md

This project was bootstrapped with Create React App.

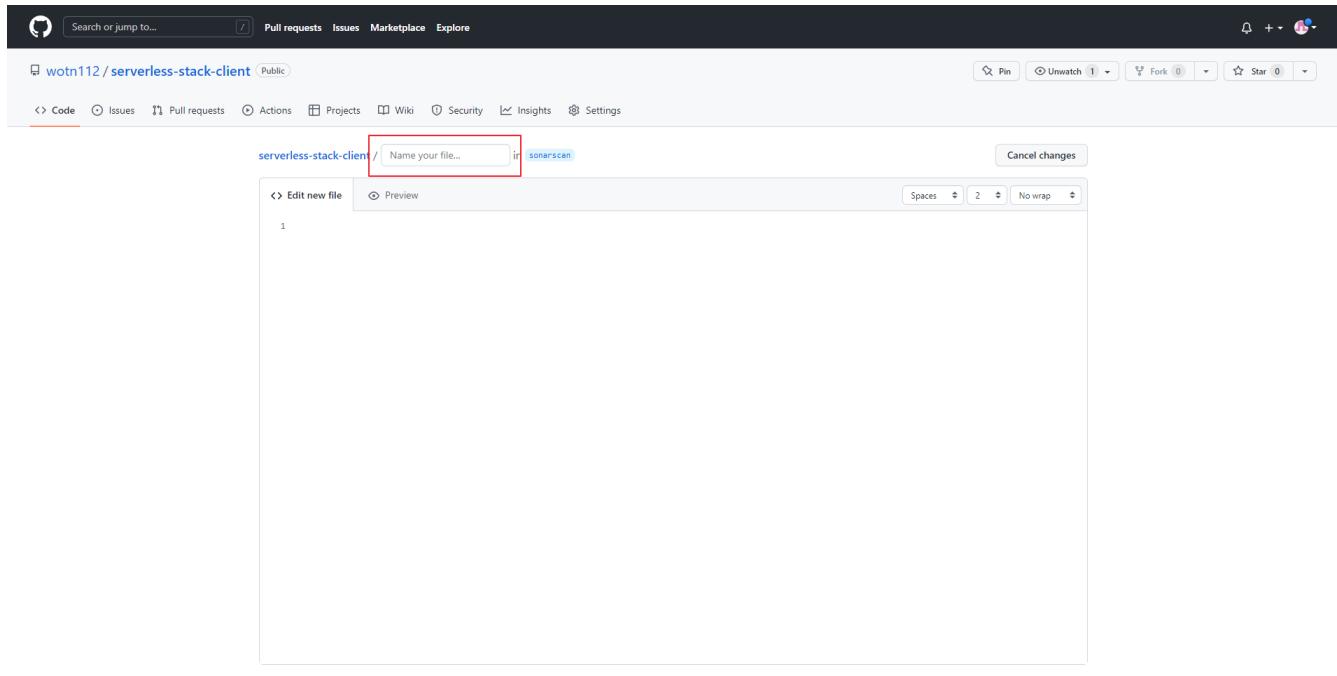
Available Scripts

In the project directory, you can run:

```
npm start
```

Runs the app in the development mode.
Open <http://localhost:3000> to view it in the browser.

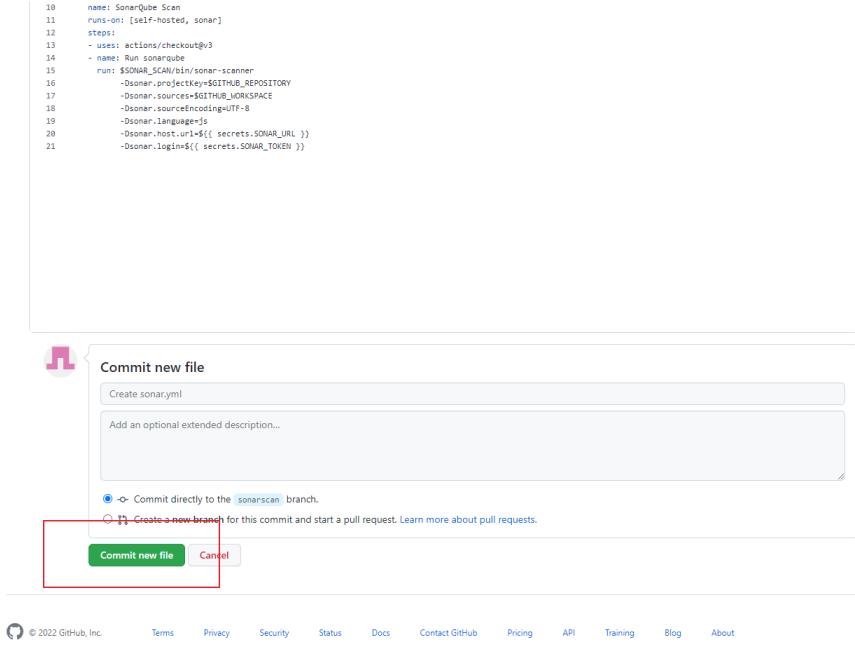
■ 해당 부분에 ".github/workflows/sonar.yml" 입력



■ 적용할 yml 파일 작성 후 commit new file(yml 내용은 아래 참조)

```

10    name: SonarQube Scan
11    runs-on: [self-hosted, sonar]
12    steps:
13      - uses: actions/checkout@v3
14      - name: Run sonarqube
15        run: $SONAR_SCAN/bin/sonar-scanner
16          -Dsonar.projectKey=$GITHUB_REPOSITORY
17          -Dsonar.sources=$GITHUB_WORKSPACE
18          -Dsonar.cs=UTF-8
19          -Dsonar.languages=js
20          -Dsonar.host.url=${{ secrets.SONAR_URL }}
21          -Dsonar.login=${{ secrets.SONAR_TOKEN }}
```



Github 자체 Runner를 통해 Sonarqube를 사용하려는 경우 Sonarqube 서버와 Github 자체 Runner 가 통신 가능해야한다

Java Maven용 Action yml 파일

■ Self-hosted Runner 사용 시

```
on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  MAVEN_HOME: /opt/maven
  SONAR_PROJECTKEY: maven
jobs:
  sonarQubeTrigger:
    name: SonarQube Scan
    runs-on: [self-hosted, sonar]
    steps:
      - uses: actions/checkout@v3
      - name: memory increase
        run: export MAVEN_OPTS="-Xmx512m"
      - name: Run sonarqube
        run: $MAVEN_HOME/bin/mvn sonar:sonar
          -Dsonar.host.url=${{ secrets.SONAR_URL }}
          -Dsonar.login=${{ secrets.SONAR_TOKEN }}
          -Dsonar.projectName=$GITHUB_REPOSITORY
          -Dsonar.projectKey=$SONAR_PROJECTKEY
          -Dsonar.sourceEncoding=UTF-8
```

- Github 제공 Runner 사용 시

```

on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  MAVEN_HOME: /opt/maven
  SONAR_PROJECTKEY: maven
jobs:
  sonarQubeTrigger:
    name: SonarQube Scan
    runs-on: [self-hosted, sonar]
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0 # Shallow clones should be disabled for a better relevancy of analysis
      - name: Set up JDK 11
        uses: actions/setup-java@v1
        with:
          java-version: 11
      - name: Cache SonarQube packages
        uses: actions/cache@v1
        with:
          path: ~/.sonar/cache
          key: ${{ runner.os }}-sonar
          restore-keys: ${{ runner.os }}-sonar
      - name: Cache Maven packages
        uses: actions/cache@v1
        with:
          path: ~/.m2
          key: ${{ runner.os }}-m2-${{ hashFiles('**/pom.xml') }}
          restore-keys: ${{ runner.os }}-m2
      - name: Build and analyze
        env:
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
          SONAR_HOST_URL: ${{ secrets.SONAR_URL }}
        run: mvn -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar
          -Dsonar.host.url=$SONAR_HOST_URL
          -Dsonar.login=$SONAR_TOKEN
          -Dsonar.projectName=$GITHUB_REPOSITORY
          -Dsonar.projectKey=$SONAR_PROJECTKEY
          -Dsonar.sourceEncoding=UTF-8

```

NodeJs용 Action yml 파일

- Self-hosted Runner 사용 시

```

on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  SONAR_SCAN: /opt/sonar-scan
  SONAR_PROJECTKEY: nodejs
jobs:
  sonarQubeTrigger:
    name: SonarQube Scan
    runs-on: [self-hosted, sonar]
    steps:
      - uses: actions/checkout@v3
      - name: memory increase
        run: export SONAR_SCANNER_OPTS="-Xmx512m"
      - name: Run sonarqube
        run: $SONAR_SCAN/bin/sonar-scanner
          -Dsonar.projectBaseDir="${GITHUB_WORKSPACE}/src"
          -Dsonar.host.url=${{ secrets.SONAR_URL }}
          -Dsonar.login=${{ secrets.SONAR_USER }}
          -Dsonar.password=${{ secrets.SONAR_PASSWORD }}
          -Dsonar.projectName=$GITHUB_REPOSITORY
          -Dsonar.projectKey=$SONAR_PROJECTKEY
          -Dsonar.sources=$GITHUB_WORKSPACE
          -Dsonar.sourceEncoding=UTF-8
          -Dsonar.language=js

```

- Github 제공 Runner 사용 시

```

on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  SONAR_PROJECTKEY: nodejs
jobs:
  sonarQubeTrigger:
    name: SonarQube Scan
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: SonarQube Scan
        uses: sonarsource/sonarqube-scan-action@master
        env:
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
          SONAR_HOST_URL: ${{ secrets.SONAR_URL }}
        with:
          projectBaseDir: ./src
        args: >
          -Dsonar.projectName=$GITHUB_REPOSITORY
          -Dsonar.projectKey=$SONAR_PROJECTKEY
          -Dsonar.sourceEncoding=UTF-8
          -Dsonar.language=js

```

C/C++/Objective-C용 Action yml 파일

 CFamily의 경우 먼저 Build가 되어야한다

- Self-hosted Runner 사용 시

```
on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  SONAR_PROJECTKEY: cfamily
  SONAR_CSCAN: /opt/build-wrapper
  SONAR_SCAN: /opt/sonar-scan/
jobs:
  sonarQubeTrigger:
    name: SonarQube Scan
    runs-on: [self-hosted, sonar]
    steps:
      - uses: actions/checkout@v3
      - name: create property file
        run: |
          echo "sonar.host.url=${{ secrets.SONAR_URL }}" >> sonar-project.properties
          echo "sonar.login=${{ secrets.SONAR_TOKEN }}" >> sonar-project.properties
          echo "sonar.projectKey=$SONAR_PROJECTKEY" >> sonar-project.properties
          echo "sonar.projectName=$GITHUB_REPOSITORY" >> sonar-project.properties
          echo "sonar.sources=$GITHUB_WORKSPACE" >> sonar-project.properties
          echo "sonar.cfamily.build-wrapper-output=${GITHUB_WORKSPACE}/build" >> sonar-project.properties
          echo "sonar.sourceEncoding=UTF-8" >> sonar-project.properties
      - name: build sonarqube
        run: |
          mkdir build
          cmake -S . -B build
          $SONAR_CSCAN/build-wrapper-linux-x86-64 --out-dir ${GITHUB_WORKSPACE}/build cmake --build build/ --
config Release
  - name: scanner start
    run: $SONAR_SCAN/bin/sonar-scanner
```

- Github 제공 Runner 사용 시

```

on:
  push:
    branches:
      - "sonarscan"
    paths:
      - ".github/workflows/sonar.yml"

name: SonarQube
env:
  SONAR_PROJECTKEY: cFamily
jobs:
  Sonarscan:
    name: Build
    runs-on: ubuntu-latest
    env:
      SONAR_SCANNER_VERSION: 4.7.0.2747
      SONAR_SERVER_URL: ${{ secrets.SONAR_URL }}
      BUILD_WRAPPER_OUT_DIR: sonar_build
      SONAR_PROJECTKEY: cFamily
    steps:
      - uses: actions/checkout@v2
        with:
          fetch-depth: 0 # Shallow clones should be disabled for a better relevancy of analysis
      - name: Set up JDK 11
        uses: actions/setup-java@v1
        with:
          java-version: 11
      - name: Download and set up sonar-scanner
        env:
          SONAR_SCANNER_DOWNLOAD_URL: https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-
scanner-cli-${{ env.SONAR_SCANNER_VERSION }}-linux.zip
        run: |
          mkdir -p $HOME/sonar
          curl -sSLo "$HOME/sonar/sonar-scanner.zip" ${{ env.SONAR_SCANNER_DOWNLOAD_URL }}
          unzip -o "$HOME/sonar/sonar-scanner.zip" -d $HOME/sonar/
          echo "$HOME/sonar/sonar-scanner-${{ env.SONAR_SCANNER_VERSION }}-linux/bin" >> $GITHUB_PATH
      - name: Download and set up build-wrapper
        env:
          BUILD_WRAPPER_DOWNLOAD_URL: ${{ env.SONAR_SERVER_URL }}/static/cpp/build-wrapper-linux-x86.zip
        run: |
          curl -sSLo $HOME/sonar/build-wrapper-linux-x86.zip ${{ env.BUILD_WRAPPER_DOWNLOAD_URL }}
          unzip -o $HOME/sonar/build-wrapper-linux-x86.zip -d $HOME/sonar/
          echo "$HOME/sonar/build-wrapper-linux-x86" >> $GITHUB_PATH
      - name: create property file
        run: |
          echo "sonar.host.url=${{ secrets.SONAR_URL }}" >> sonar-project.properties
          echo "sonar.login=${{ secrets.SONAR_TOKEN }}" >> sonar-project.properties
          echo "sonar.projectKey=$SONAR_PROJECTKEY" >> sonar-project.properties
          echo "sonar.projectName=$GITHUB_REPOSITORY" >> sonar-project.properties
          echo "sonar.sources=$GITHUB_WORKSPACE" >> sonar-project.properties
          echo "sonar.cfamily.build-wrapper-output=${GITHUB_WORKSPACE}/build" >> sonar-project.properties
          echo "sonar.sourceEncoding=UTF-8" >> sonar-project.properties
      - name: Run build-wrapper
        run: |
          mkdir ${env.BUILD_WRAPPER_OUT_DIR}
          cmake -S . -B ${env.BUILD_WRAPPER_OUT_DIR}
          build-wrapper-linux-x86-64 --out-dir ${env.BUILD_WRAPPER_OUT_DIR} cmake --build build/ --config
Release
  - name: Run sonar-scanner
    run: sonar-scanner

```

확인

- Actions → All workflows에 해당 작업의 성공여부를 확인 할 수 있다

The screenshot shows the GitHub Actions workflow runs page for the repository 'wotn112/sonarqubeCrunner'. A red box highlights the 'Actions' tab in the top navigation bar. Another red box highlights the 'All workflows' section in the main content area, which displays a single workflow run titled 'Update sonar.yml' triggered by SonarQube. The run was completed 31 minutes ago and took 39s.

<https://github.com/wotn112/sonarqubeCrunner/actions/runs/274504553>

Branch 삭제

- 해당 작업을 완료한 후 해당 Branch를 삭제하고자 하는 경우
Branch 선택 → 삭제하고자 하는 Branch 오른쪽 휴지통 선택

The screenshot shows the GitHub repository page for 'wotn112/sonarqubeCrunner'. A red box highlights the 'Code' tab in the top navigation bar. In the main content area, under the 'Branches' section, it shows 'main' and '3 branches'. The '3 branches' link is also highlighted with a red box. On the right side of the page, there are sections for 'About', 'Releases', 'Packages', and 'Contributors'.

<https://github.com/wotn112/sonarqubeCrunner/branches>

The screenshot shows a GitHub repository page for 'wotn112 / sonarqubeCrunner' (Public). The main navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The top right features icons for pinning, unwatching, forking, and starring the repository. Below the header, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A search bar for branches is present. The 'Overview' tab is selected, showing tabs for 'Yours', 'Active', 'Stale', 'All branches', and a green 'New branch' button. The 'Default branch' section shows the 'main' branch, which is updated 2 days ago by SonarTech. A warning message states 'Your main branch isn't protected' with a note to 'Protect this branch'. The 'Your branches' section lists 'sonarscan' (updated 16 minutes ago) and 'geoffrey/coverage_conf_after_config' (updated 3 months ago). The 'Active branches' section also lists these two branches. A red box highlights the delete icon for the 'geoffrey/coverage_conf_after_config' branch.

참조 링크

Sonarqube

- Sonarqube Doc-Maven scanner
- Sonarqube Doc-JS, CSS scanner
- Sonarqube Doc-Cfmaily scanner

Github

- Github Doc-Self_hosted_runner
- Github Doc-Workflow