

Gerrit Code Review 활용 가이드

2024.01.22



공통 프로젝트



Gerrit Code Review 활용 가이드

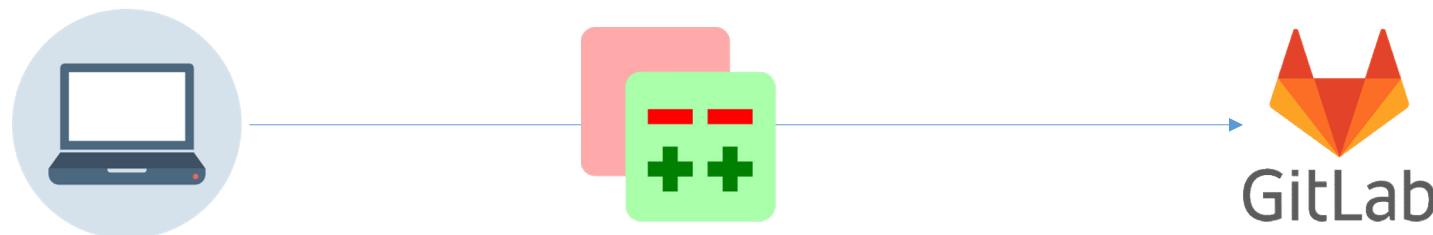
최준서

실습 코치

-
- 삼성 청년 SW 아카데미 9기 수료
 - 삼성 청년 SW 아카데미 10기 실습 코치



I Gerrit



Code Review란?



Code Review

- 개발자가 작성한 코드를 다른 개발자들이 검토하고 피드백 하는 과정



Code Review란?

Code Review

- 개발자가 작성한 코드를 다른 개발자들이 검토하고 피드백 하는 과정

코드 품질 개선

- 가독성, 유지보수성, 성능 등

Code Review란?



Code Review

- 개발자가 작성한 코드를 다른 개발자들이 검토하고 피드백 하는 과정

코드 품질 개선

- 가독성, 유지보수성, 성능 등

협업 능력 향상

- 개발자가 공통으로 코드를 개발하고 개선할 수 있음

Code Review란?



Code Review

- 개발자가 작성한 코드를 다른 개발자들이 검토하고 피드백 하는 과정

코드 품질 개선

- 가독성, 유지보수성, 성능 등

협업 능력 향상

- 개발자가 공통으로 코드를 개발하고 개선할 수 있음

코드 작성 능력 향상

- 구조 및 코딩 스타일 등을 배우며 코드 작성 능력 향상

Code Review란?



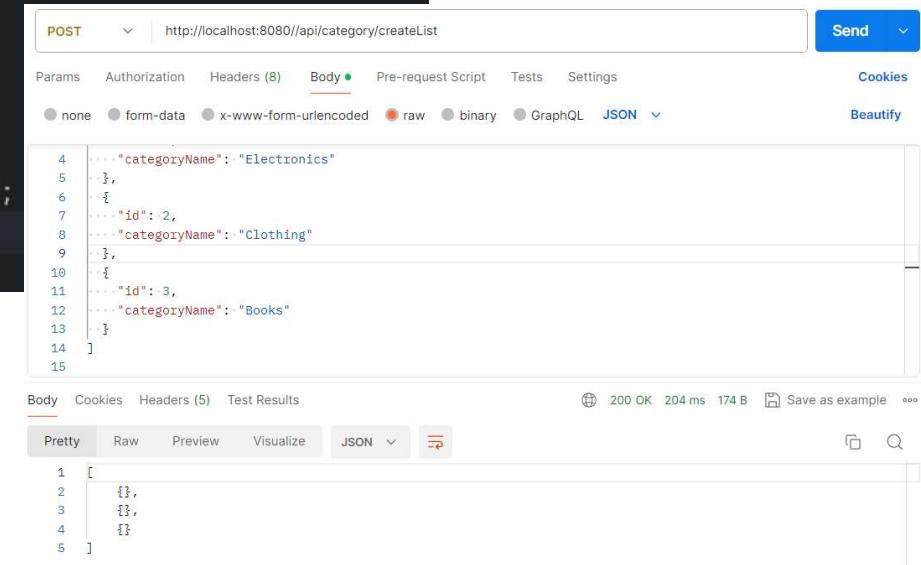
최적화

Code Review란?

✓ Code Review

- 해당 코드를 보고 생각나는 문제점에 대해서 말씀해주세요!

```
new *  
@PostMapping("/createList")  
public ResponseEntity<Object> createCategory(@RequestBody List<CategoryDto> categoryDtoList) {  
    //빠른 설명은 위해 Repository가 Controller에 있음 양해 부탁드립니다.  
    List<Category> categories = categoryDtoList.stream() Stream<CategoryDto>  
        .map(dto -> Category.builder()  
            .categoryName(dto.getCategoryName())  
            .build()) Stream<Category>  
        .collect(Collectors.toList());  
  
    return ResponseEntity.ok(categoryRepository.saveAll(categories));  
}
```



The screenshot shows a Postman interface with a successful API call. The URL is `http://localhost:8080/api/category/createList`. The request method is POST. The Body tab is selected, showing a JSON payload:

```
4 ... "categoryName": "Electronics"  
5 ... }  
6 ... {  
7 ... "id": 2,  
8 ... "categoryName": "Clothing"  
9 ... }  
10 ... {  
11 ... "id": 3,  
12 ... "categoryName": "Books"  
13 ... }  
14 ]  
15
```

The response status is 200 OK with a response time of 204 ms and a body size of 174 B. The response body is shown in Pretty, Raw, and JSON formats:

```
1 [  
2   {},  
3   {},  
4   {}  
5 ]
```

Code Review란?

Code Review

- 예외 상황을 고려한 설계가 잘 되어 있는가

Body

```
{  
    "A": "B"  
}
```

Code Review란?



Code Review

- 예외 상황을 고려한 설계가 잘 되어 있는가

Body

```
{  
    "A": "B"  
}
```

데이터가 있는데
잘못 들어왔거나

Header

Body

```
{  
    "": "B"  
}
```

Code Review란?

Code Review

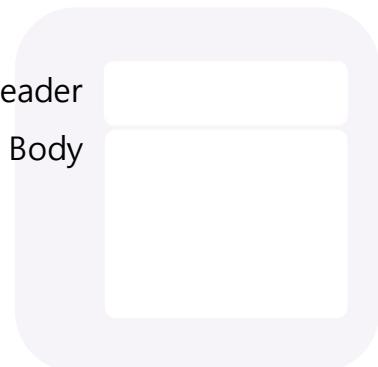
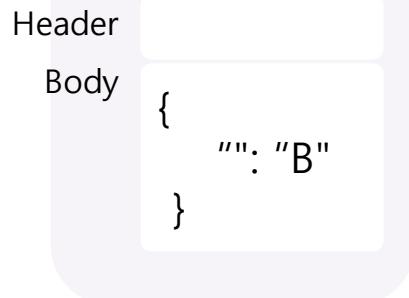
- 예외 상황을 고려한 설계가 잘 되어 있는가

Body

```
{  
    "A": "B"  
}
```

데이터가 있는데
잘못 들어왔거나

데이터가 없거나



Code Review란?



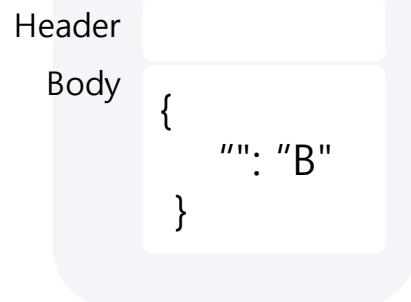
Code Review

- 예외 상황을 고려한 설계가 잘 되어 있는가

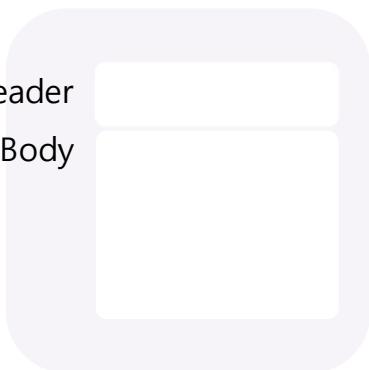
Body

```
{  
  "A": "B"  
}
```

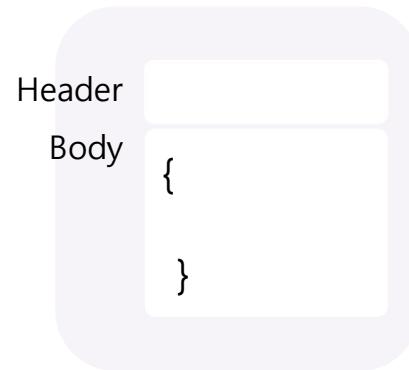
데이터가 있는데
잘못 들어왔거나



데이터가 없거나



데이터가 있는데
비었거나



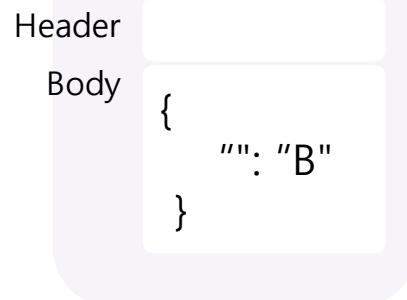
Code Review란?



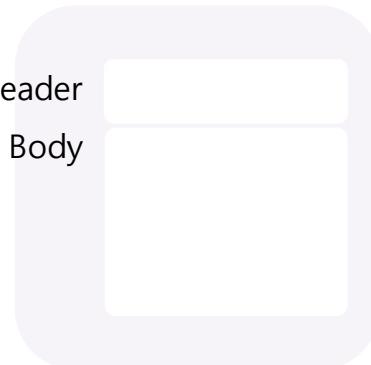
Code Review

- 예외 상황을 고려한 설계가 잘 되어 있는가

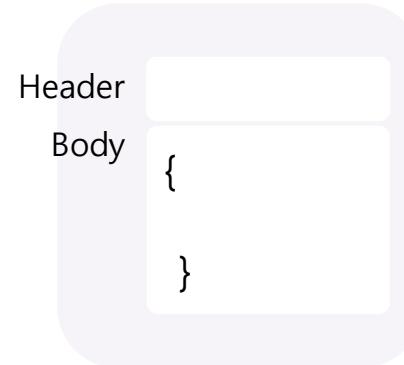
데이터가 있는데
잘못 들어왔거나



데이터가 없거나

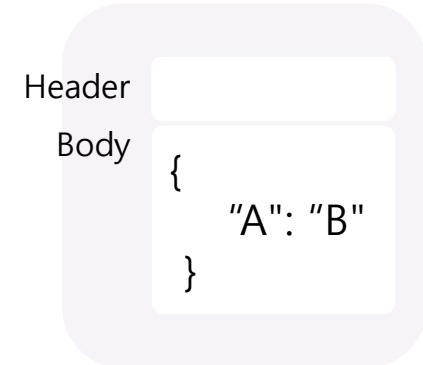


데이터가 있는데
비었거나



Body
{
 "A": "B"
}

데이터가 제대로
되어 있거나

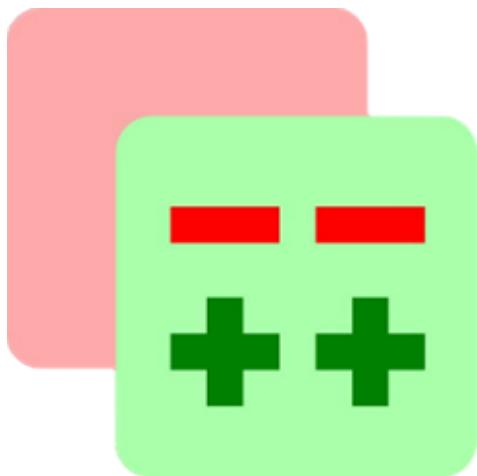


Gerrit 개념



☑ Gerrit 이란?

- 코드 리뷰 도구 중 하나
- 웹 브라우저에서 다른 개발자들의 코드를 검토하고 피드백 하는 기능을 제공(시간, 공간에 제약이 없다)
- 변경 사항 승인, 거절을 통해 소스 코드 반영 여부 결정 가능



The screenshot shows the Gerrit web interface for a code review. The top navigation bar includes links for CHANGES, YOUR, DOCUMENTATION, and BROWSE, along with user information for 'ssafy03'. The main content area displays a 'Merged' change with ID #61, titled 'modify origin test'. The 'Change Info' section shows the following details:

- Submitted: 4:46 PM
- Owner: ssafy02
- Author: 10gi_coach
- Committer: 10gi_coach
- Reviewers: ssafy03 (2)
- Repo | Branch: wishMe | master

Below this, there are sections for 'Comments' (with 1 resolved comment) and 'Submit Requirements' (Code-Review +2). The 'Files' tab is selected, showing a single file named 'README.md' with a commit message. The 'Comments' tab shows a list of changes made by 'ssafy02' and 'ssafy03'. The 'Change Log' at the bottom tracks the history of the patchset, including the merge of the patchset.

Gerrit 개념



GitHub, GitLab와의 차이점

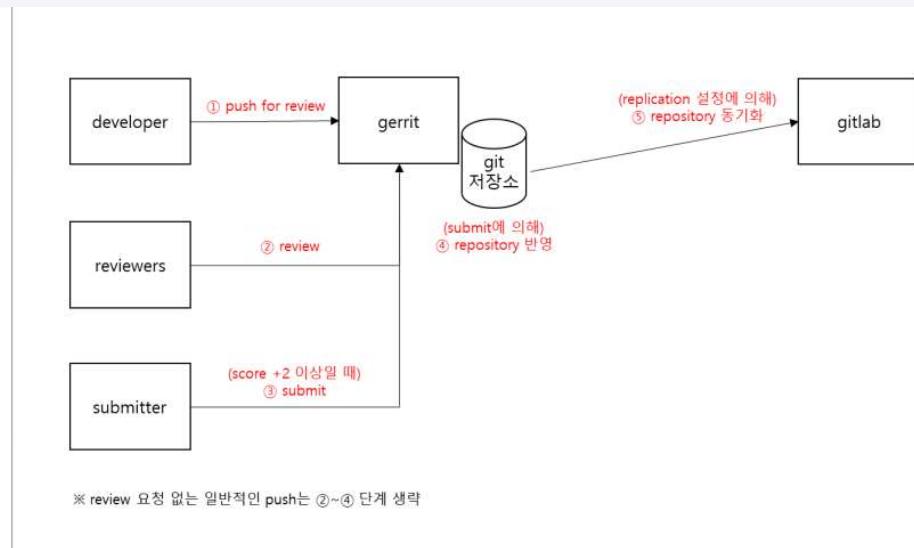
- 기존의 GitHub, GitLab PR(Pull Request)와 MR(Merge Request) 단위에서의 코드 리뷰만 가능
- Gerrit의 경우 코드 리뷰에 목적, 각각의 Commit 단위로 코드 리뷰를 진행 가능

Gerrit 개념



☑ Gerrit 구조

- Gerrit 내부에 Git 저장소가 위치해 있음
- 개발자는 소스 코드의 변경사항을 Gerrit에 Push 함
- Push 시 Review를 요청한다면 Gerrit의 변경 사항은 리뷰 오픈 상태가 됨
- 오픈된 리뷰 건을 검토하고 점수를 매길 수 있음
- 승인자는 +2 이상의 점수를 획득한 리뷰 건을 승인하여 Gerrit저장소에 반영
- GitLab와 Replication 설정이 되어있다면 Gerrit의 저장소에 반영된 변경 사항은 GitLab에 저장

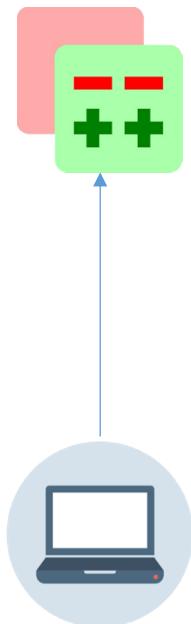


Gerrit 개념



Branch

- 새로운 브랜치 생성 전략.
- 로컬에서 브랜치 생성 직후 코드 리뷰를 요청하지 않는 경우



현재 Gerrit에 존재하는 Branch
- master

2. Gerrit Branch

1. git checkout -b develop

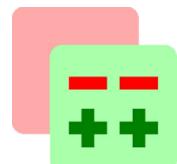
현재 local에 존재하는 Branch
- Master
- develop

Gerrit 개념



Branch

- 새로운 브랜치 생성 전략.
- 로컬에서 브랜チ 생성 직후 코드 리뷰를 요청하지 않는 경우



현재 Gerrit에 존재하는 Branch
- Master

3. Branch 반영



2. git push origin develop

현재 local에 존재하는 Branch
- Master
- develop

1. git checkout -b develop

```
SSAFY@DESKTOP-LLB32F4 MINGW64 ~/Desktop/coach12/code/test/demo (test)
$ git checkout -b develop
Switched to a new branch 'develop'

SSAFY@DESKTOP-LLB32F4 MINGW64 ~/Desktop/coach12/code/test/demo (develop)
$ git add .

SSAFY@DESKTOP-LLB32F4 MINGW64 ~/Desktop/coach12/code/test/demo (develop)
$ git commit -m "Test commit"
[develop 23231f8] Test commit
1 file changed, 1 insertion(+)

SSAFY@DESKTOP-LLB32F4 MINGW64 ~/Desktop/coach12/code/test/demo (develop)
$ git push origin develop
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 799 bytes | 799.00 KiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4)
remote: Processing changes: refs: 1, done
To https://t1017.p.ssafy.io:8989/a/test
 * [new branch]      develop -> develop
```

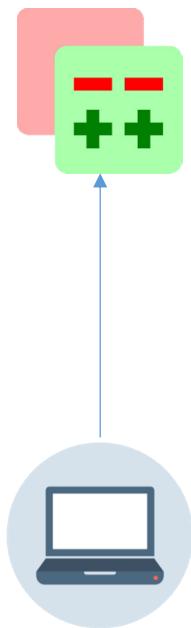
Code 수정
git add .
git commit -m “Commit Message”

Gerrit 개념



Branch

- 새로운 브랜치 생성 전략.
- 로컬에서 브랜치 생성 직후 코드 리뷰를 요청하지 않는 경우



현재 Gerrit에 존재하는 Branch

- Master
- develop

2. git push origin develop

1. git checkout -b develop

현재 local에 존재하는 Branch

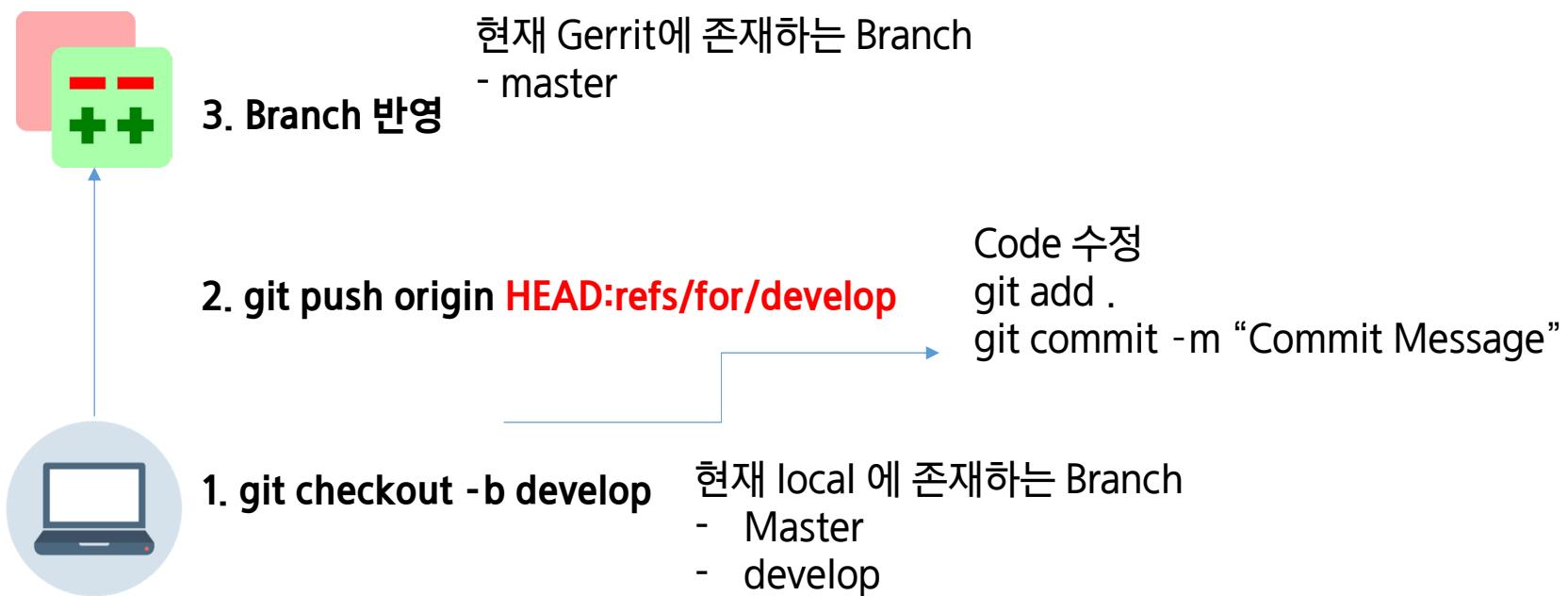
- Master
- develop

Gerrit 개념



Branch

- 새로운 브랜치 생성 전략.
- 로컬에서 브랜치 생성 직후 코드 리뷰를 요청하는 경우.
- 리뷰 요청을 하게 될 시 지정된 refspec을 맞춰 보냄.
 - Refspec 사용 시 반드시 Gerrit이 해당 브랜치가 있음을 알아야 사용 가능.

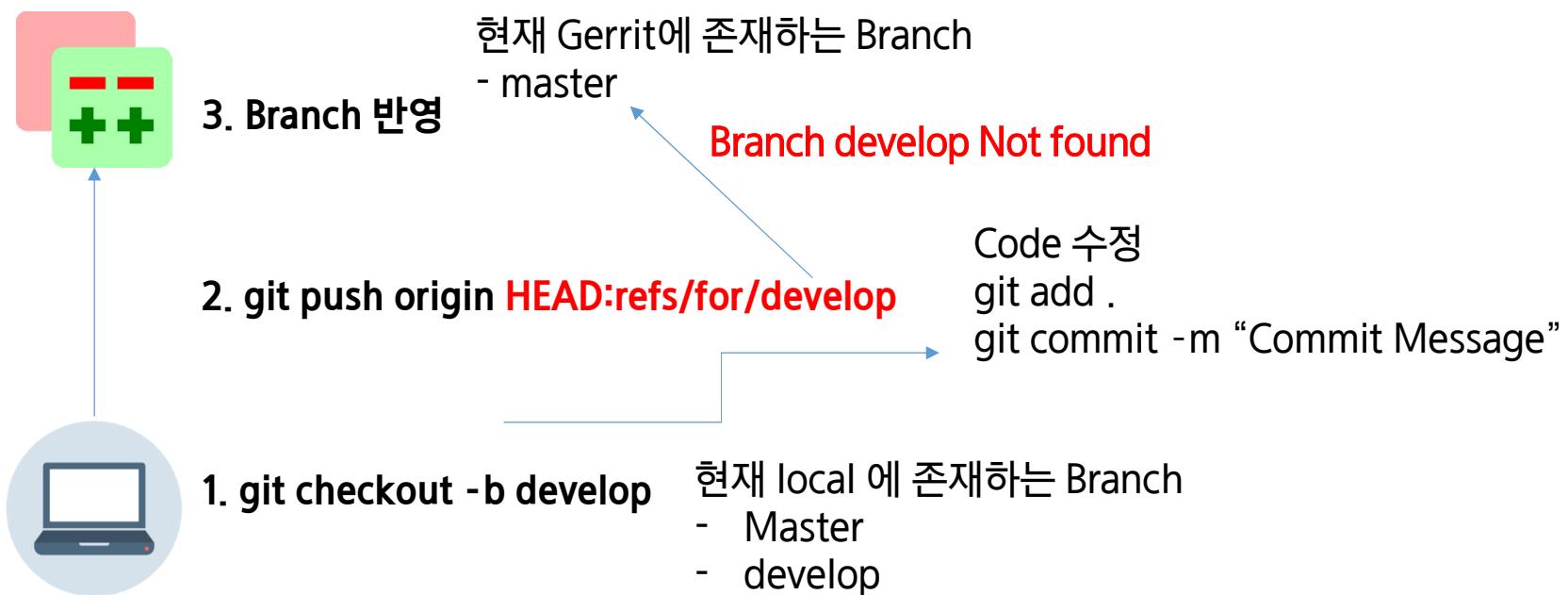


Gerrit 개념



Branch

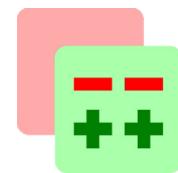
- 새로운 브랜치 생성 전략.
- 로컬에서 브랜치 생성 직후 코드 리뷰를 요청하는 경우.
- 리뷰 요청을 하게 될 시 지정된 refspec을 맞춰 보냄.
 - Refspec 사용 시 반드시 Gerrit이 해당 브랜치가 있음을 알아야 사용 가능.



Gerrit 개념

Branch

- 새로운 브랜치 생성 전략.
- 로컬에서 브랜치 생성 직후 코드 리뷰를 요청하는 경우.
- 리뷰 요청을 하게 될 시 지정된 refspec을 맞춰 보냄.
 - Refspec 사용 시 반드시 Gerrit이 해당 브랜치가 있음을 알아야 사용 가능.



3. Branch 반영



2. git push origin HEAD:refs/for/develop

1. git checkout -b develop

```
SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/test/demo (branchtest)
$ git checkout -b PRTEST
Switched to a new branch 'PRTEST'

SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/test/demo (PRTEST)
$ git add .

SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/test/demo (PRTEST)
$ git commit -m "test"
[PRTEST a0646aa] test
 1 file changed, 2 insertions(+)

SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/test/demo (PRTEST)
$ git push origin HEAD:refs/for/PRTEST
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 805 bytes | 805.00 KiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4)
remote: Processing changes: refs: 1, done
To https://t1017.p.ssafy.io:8989/a/test
 ! [remote rejected] HEAD -> refs/for/PRTEST (branch PRTEST not found)
error: failed to push some refs to 'https://t1017.p.ssafy.io:8989/a/test'
```

Code 수정
git add .
git commit -m "Commit Message"
git push origin HEAD:refs/for/develop

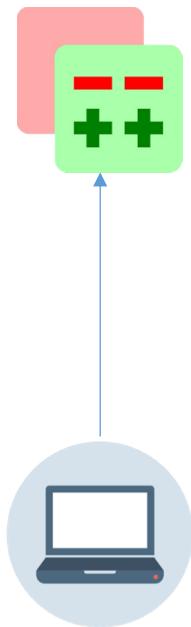


Gerrit 개념



Branch

- Gerrit이 Branch가 있음을 알아야 push가 가능하다.



3. Branch 반영

현재 Gerrit에 존재하는 Branch
- master

2. git push origin develop

1. git checkout -b develop

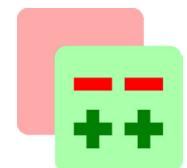
현재 local에 존재하는 Branch
- Master
- develop

Gerrit 개념



☒ Branch

- Gerrit이 Branch 가 있음을 알아야 push가 가능하다.



3. Branch 반영

현재 Gerrit에 존재하는 Branch
- Master
- develop

Code 수정
git add .
git commit -m "Commit Message"
git push origin HEAD:refs/for/develop



2. git push origin develop

1. git checkout -b develop

현재 local 에 존재하는 Branch
- Master
- develop

Gerrit 개념



☒ Branch

- Gerrit의 Branch 가 있음을 알아야 push가 가능하다.



3. Branch 반영



2. git push origin

1. git checkout -b

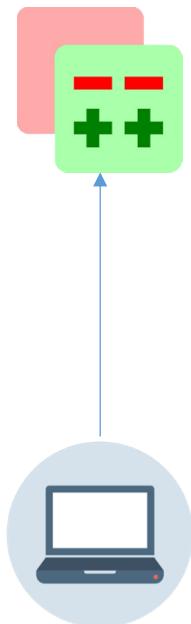
```
SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/test/demo (develop)
$ git push origin HEAD:refs/for/develop
현지화
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 792 bytes | 792.00 KiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4)
remote: Processing changes: refs: 1, new: 1, done
remote:
remote: SUCCESS
remote:
remote: https://[REDACTED]/c/test/+/112 test [NEW]
remote:
To https://[REDACTED]/a/test
 * [new reference]    HEAD -> refs/for/develop
```

Gerrit 개념



Branch

- Develop Branch에서 작업한 내용을 Master Branch로 병합



5. Branch 반영

4. git push origin master

3. git merge develop (병합될 Branch)

2. git checkout master (병합할 Branch로 이동)

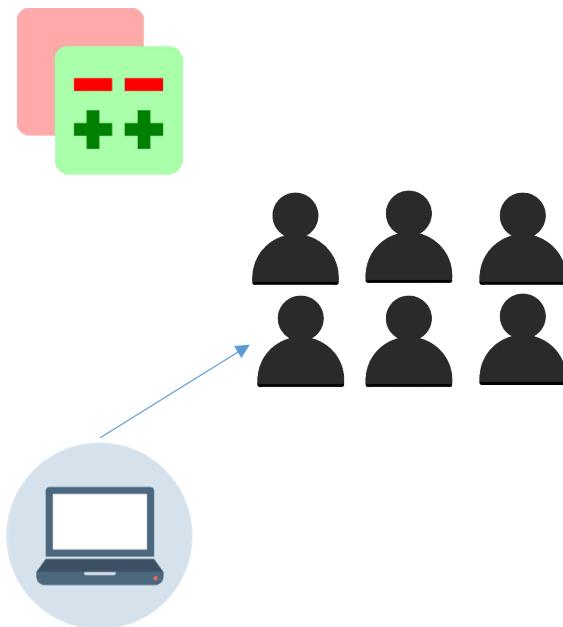
1. Develop 작업 완료 후

Gerrit 개념



Code Review

Code Review를 요청 시 리뷰어들이 코드를 리뷰

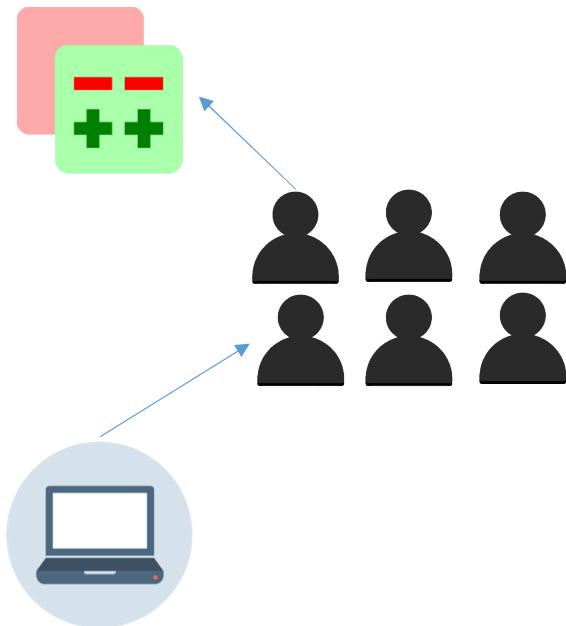


Gerrit 개념



Code Review

리뷰 결과 +2 점이 있고 -2점이 없을 경우 해당 코드를 Gerrit 저장소에 반영 가능



Gerrit 실습



Gerrit Password 설정

- 제공된 EC2에는 Gerrit이 설치된 형태로 제공

Gerrit 접속

- Password 변경 후 [http://도메인주소 :8989/](http://도메인주소:8989/) 로 접속
- Username은 1인당 1개 사용이 원칙

Git Bash 터미널 > EC2 SSH 접속

```
$ ssh -i pem파일명 ubuntu@인스턴스명.p.ssafy.io  
//SSH 접속 후 admin 비밀번호 변경
```

```
$ sudo htpasswd /opt/gerrit/etc/passwords admin  
New password:  
Re-type new password:  
Updating password for user admin
```

로그인

사용자 이름

비밀번호

로그인

취소

Gerrit 실습



☒ Gerrit Password 설정

- 제공된 EC2에는 Gerrit이 설치된 형태로 제공

☒ Gerrit 접속

- Password 변경 후 [http://도메인주소 :8989/](http://도메인주소:8989/) 로 접속
- Username은 1인당 1개 사용이 원칙

Git Bash 터미널 > EC2 SSH 접속

```
$ssh -i pem파일명 ubuntu@인스턴스명.p.ssafy.io  
//SSH 접속 후 admin 비밀번호 변경
```

```
$sudo htpasswd /opt/gerrit/etc/passwords admin  
New password:  
Re-type new password:  
Updating password for user admin
```

The screenshot shows the Gerrit 'Sign In' page. On the left, there are sections for '사용자' (User) and '비밀번호' (Password). The 'User' section contains input fields for 'Username', 'Email Address', and 'Account ID', each with a 'Become Account' button. Below these is a 'Choose:' dropdown menu with options: 'admin', 'ssafy01', 'ssafy02', 'ssafy03', 'ssafy04', 'ssafy05', 'ssafy06', and 'ssafy07'. On the right, there is a 'Register' section with a 'New Account' button.

Gerrit 실습



☑ 계정 설정 - Email 설정

- <http://도메인주소:8989/login> 유저 선택 창으로 이동 가능

The screenshot shows the Gerrit web interface. At the top, there is a navigation bar with links for CHANGES, YOUR, DOCUMENTATION, and BROWSE. A search bar and a user dropdown menu are also present. The user dropdown menu is open, showing the username "ssafy04" and four options: "Settings" (which is highlighted with a red border), "Keyboard Shortcuts", and "Sign out".

The main content area displays several sections:

- Your Turn**: Shows "No changes need your attention" with a bee icon.
- Outgoing reviews**: Shows "Push your first change for code review". It includes instructions: "Pushing a change for review is easy, but a little different from other git code review tools. Click on the 'Create Change' button and follow the step by step instructions." Below this is a "CREATE CHANGE" button.
- No outgoing changes yet**: A message indicating there are no outgoing changes.
- Incoming reviews**: Shows "No changes".
- CCed on**: Shows "No changes".

Gerrit 실습



☑ 계정 설정 - Email 설정

- <http://도메인주소:8989/login> 유저 선택 창으로 이동 가능

The screenshot shows the Gerrit web interface with the following details:

- Header:** Gerrit, CHANGES, YOUR, DOCUMENTATION, BROWSE, Search bar (ssafy04), Help icon.
- Left Sidebar:** Your Turn (No changes need your attention), Outgoing reviews (No outgoing changes yet), Incoming reviews (No changes), CCed on (No changes).
- Profile Section:** Profile, Preferences, Diff Preferences, Edit Preferences, Menu, Change Table Columns, Notifications, **Email Addresses** (highlighted with a red border), HTTP Credentials, SSH Keys, Groups, Identities, Mail Filters.
- Email Addresses Sub-section:** Preferred Email (ssafy04), New email address (email@example.com), SEND VERIFICATION button.
- HTTP Credentials Sub-section:** Username (ssafy04), GENERATE NEW PASSWORD button.
- SSH keys Sub-section:** Comment, SAVE CHANGES button, Status Public key.

Red annotations:

- A red box highlights the "Email Addresses" link in the sidebar.
- A red box highlights the "email@example.com" input field in the "Email Addresses" section.
- Red text at the bottom right of the "Email Addresses" section states: "Commit의 author 정보 검증을 위해 메일 주소를 반드시 알맞게 입력 SSAFY Git 의 이메일 및 로컬 개발 환경의 Git 설정상 이메일 정보와 일치시켜야함."

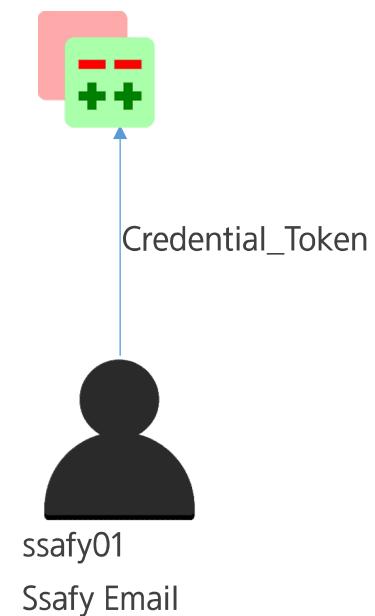
Gerrit 실습



☑ 계정 설정 - HTTP Credential Token

- Git push/pull 등 원격 인증이 필요할 때 패스워드로 사용함
- 유출 및 분실 시 재발급하여 새로운 토큰으로 교체 가능

The screenshot shows the Gerrit web interface under the 'Profile' tab. On the left sidebar, 'HTTP Credentials' is selected and highlighted with a red border. The main content area shows the 'Email Addresses' and 'HTTP Credentials' sections. In the 'HTTP Credentials' section, the username 'ssafy04' is listed. A modal window at the bottom displays a generated password: `1MANKhTUYAf18o0m9EbT8mogiIKI16gW7TJNwn7MLBC`. The modal also contains the message: "This password will not be displayed again. If you lose it, you will need to generate a new one." and a 'CLOSE' button.



Gerrit 실습



팀 내 User 정보



ssafy01
Ssafy Email
Credential_Token



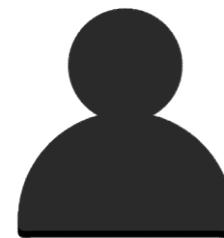
ssafy02
Ssafy Email
Credential_Token



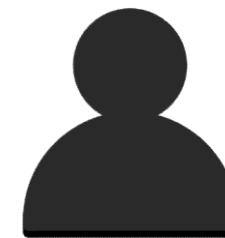
ssafy03
Ssafy Email
Credential_Token



ssafy04
Ssafy Email
Credential_Token



ssafy05
Ssafy Email
Credential_Token



ssafy06
Ssafy Email
Credential_Token

Gerrit 실습

- GitLab 연동을 위한 Repository 생성

Repository 생성



Gerrit 실습

Repository 설정

- Repository 생성 권한은 admin
- Admin 선택 후 Browse

The screenshot shows the Gerrit web interface. At the top, there are navigation links: CHANGES, YOUR, DOCUMENTATION, BROWSE (with sub-options: Repositories, Groups, Plugins), and a search bar. Below this is a section titled "Your Turn" which says "No changes need your attention". A large button labeled "CREATE CHANGE" is visible. To the right, there's a "Repositories" section with three items: All-Projects, All-Users, and two user-specific repositories (greenfingers, wishMe). At the bottom, sections for "Incoming reviews" and "Recently closed" changes are shown.

Repository 설정

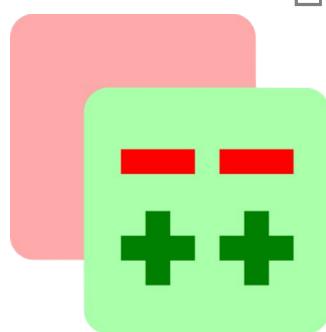
- Repository name: 동기화할 gitlab 프로젝트명과 동일하게 입력
- Create initial empty commit: False

The screenshot shows the Gerrit web interface with a modal dialog box titled "Create Repository". The dialog contains fields for "Repository name" (with a red box highlighting it), "Default Branch", "Rights inherit from" (with a note: "Optional, defaults to 'All-P'"), "Owner", "Create initial empty commit" (set to "False"), and "Only serve as parent for other repositories" (set to "False"). At the bottom, there are "CANCEL" and "CREATE" buttons. The background shows the Gerrit dashboard with the same repository list as the first screenshot.

Gerrit 실습



Gerrit Repository와 GitLab Repository 간 Replication 설정

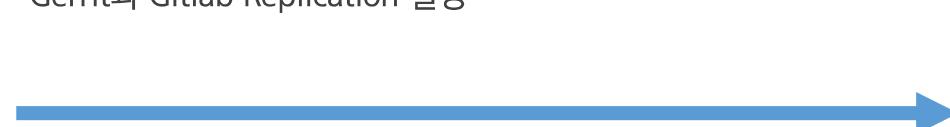


Replication.config

- Gerrit과 Gitlab Replication 설정

Secure.config

- 레포지토리 접근을 위한 설정
- GitLab의 AccessToken 등록



Gerrit 실습



Gitlab 연동

- Gerrit Repository = GitLab Repository (Replication)
- 유효기간 설정 및 권한 설정 후 Token 발급

The screenshot shows the GitLab interface for creating a project access token. On the left, the sidebar is visible with the 'Access Tokens' option selected. The main page title is 'Project Access Tokens' under the 'greenfingers' project. A form is displayed for creating a new token:

- Token name:** gerrit_repl (highlighted with a red box)
- Expiration date:** 2024-02-16 (highlighted with a red box)
- Select a role:** Maintainer
- Select scopes:** api, read_api, create_runner, read_repository, write_repository (checkboxes for 'read_repository' and 'write_repository' are checked and highlighted with a red box)

At the bottom right of the form are 'Create project access token' and 'Cancel' buttons.

On the right side of the screenshot, there are two panels showing the results of token creation:

- A message box: "Your new project access token has been created." (highlighted with a red box)
- A summary message: "Make sure you save it - you won't be able to access it again."
- A table titled "Active project access tokens" showing two entries:

Token name	Scopes	Created	Last Used	Expires	Role	Action
Gerrit	api, read_api, create_runner, read_repository, write_repository	Dec 29, 2023	3 days ago	in 2 weeks	Owner	
gerrit_repl	read_repository, write_repository	Jan 10, 2024	Never	in 1 month	Maintainer	

Gerrit 실습



Replication 설정

- Gerrit -> Gitlab 연동

```
sudo vi /opt/gerrit/etc/secure.config
```

```
[remote "{gerrit repository}"]
username = {project access token name}
password = {project access token password}
```

Gerrit 실습



☑ Replication 설정

- Gerrit Repository (Replication)

```
$sudo vi /opt/gerrit/etc/replication.config
```

```
[gerrit]
autoReload = true
replicateOnStartup = true
replicatePermissions = false
[replication]
lockErrorMaxRetries = 5
maxRetries = 5
[remote "Gerrit Repository name"]
url = https://lab.ssafy.com/{nameSpace}/{project name}
push = +refs/heads/*:refs/heads/*
push = +refs/tags/*:refs/tags/*
mirror = true
projects = gitlab 프로젝트명
```

```
sudo systemctl restart gerrit
```

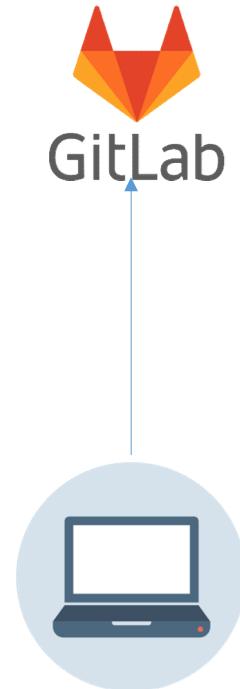
데몬 확인

```
sudo systemctl status gerrit
ps -ef | grep gerrit
```

Gerrit 실습



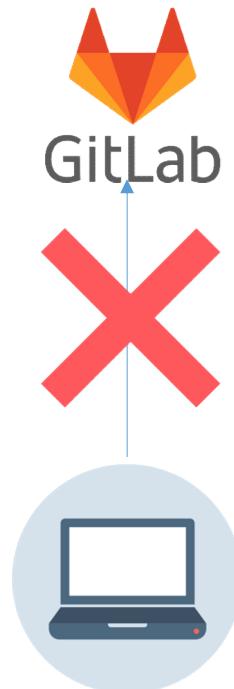
Local -> Gerrit



Gerrit 실습



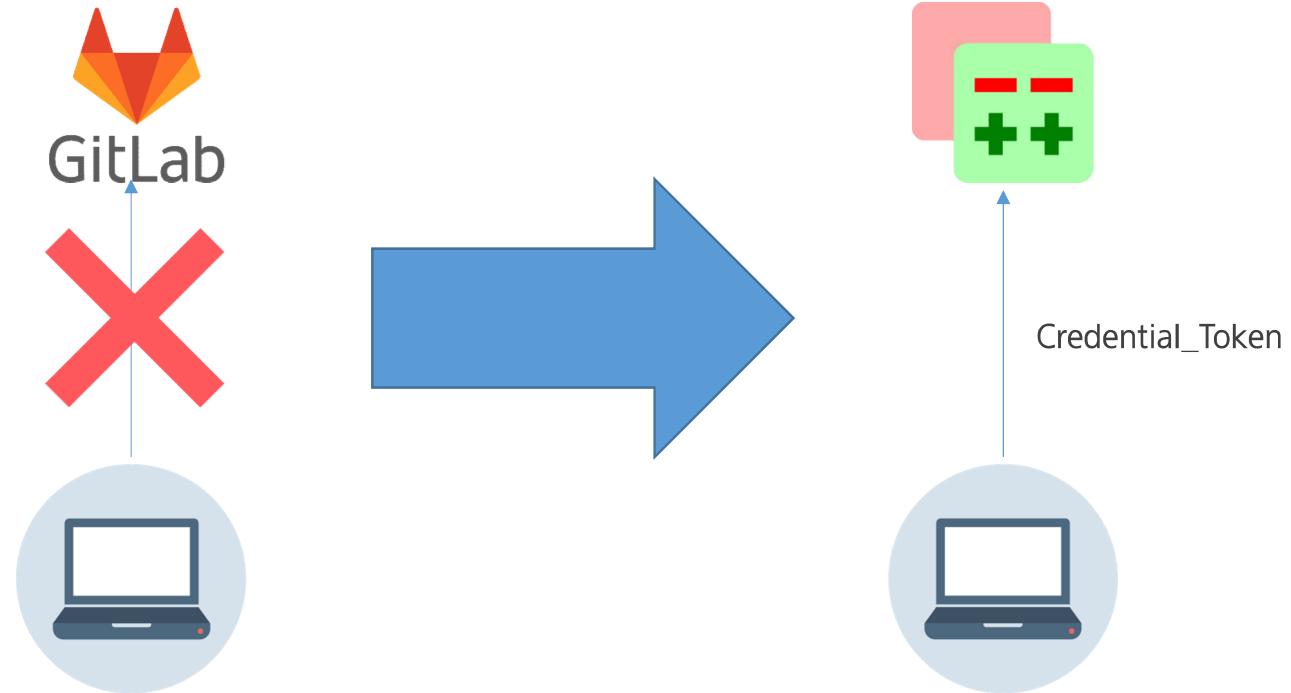
Local -> Gerrit



Gerrit 실습



Local -> Gerrit



Gerrit 실습



Gerrit Remote

- Local -> Gitlab 환경을 다음과 같이 수정
- Local -> Gerrit

```
git remote -v //현재 연결되어 있는 주소 확인
```

```
origin https://lab.ssafy.com/.....  
origin https://lab.ssafy.com/.....
```

Gerrit Push

- 팀원 중 한 명만 git push -u origin master

```
git remote set-url origin "http://<gerrit username>@<EC2 주소>:8989/a/<project명>
```

//예제

```
git remote set-url origin https://ssafy01@{도메인네임.p.ssafy.io}:8989/a/greenfingers
```

Gerrit 실습

Gerrit Remote

- Local -> Gitlab 환경을 다음과 같이 수정
- Local -> Gerrit

git remote -v //현재 연결되어 있는 주소 확인

The screenshot shows the Gerrit web interface with the 'HTTP Credentials' section highlighted. The 'Username' field contains 'ssafy04'. A modal window at the bottom displays a new password: '1MANKhTUVAf18o0m9EbT8mogiIK16gW7TJNwn7MLBQ'. The 'Email Addresses' and 'SSH keys' sections are also visible.

Gerrit Push

- 팀원 중 한 명만 git push -u origin master

The screenshot shows a terminal window with the command 'git push -u origin master' and a 'Git Credential Manager' dialog box. The dialog box prompts for credentials for 'https://lab3.p.ssafy.io:8989/' and shows the username 'ssafy01' and a masked password. The terminal output shows the command being run on a Windows machine.

Gerrit 실습

Local -> Gerrit

연동을 진행하면 Master Branch만 존재

- 아래 이미지는 TEST를 진행해서 Develop Branch 존재

The screenshot shows the Gerrit web interface. On the left, there's a sidebar with 'CHANGES', 'YOUR', 'DOCUMENTATION', 'BROWSE' (which is highlighted), and a search bar. Below that are sections for 'Repositories', 'Groups', and 'Plugins'. In the 'Repositories' section, there's a table with columns: Repository Name, Repository Browser, Changes, Read only, and Repository Description. It lists 'All-Projects' (browse, view all), 'All-Users' (browse, view all), and 'greenfingers' (highlighted browse, view all). On the right, under 'Gitles', it shows the path '/greenfingers'. Below it, there's a 'Clone this repo:' field with the command 'git clone ssh://'. Under 'Branches', it shows 'master' and 'develop' branches with their commit history: 'fe2a9aa modify.readMeTest by 10gi_coach · 8 days ago develop master', '52ddfe4 modify.readMe by 10gi_coach · 8 days ago', and 'cd13a33 imitial commit by lab1.ssafy · 2 weeks ago'. At the bottom, there's a section titled '그린핑거스' (GreenFingers) with a note about managing food intake.

Gerrit CHANGES ▾ YOUR ▾ DOCUMENTATION ▾ BROWSE ▾ Filter: ADMINISTRATOR

Repositories

Repository Name	Repository Browser	Changes	Read only	Repository Description
All-Projects	browse	view all		Access inherited by all other projects
All-Users	browse	view all		Individual user settings and preferences
greenfingers	browse	view all		

Groups

Plugins

Gitles

/ greenfingers

Clone this repo:

```
git clone ssh://
```

Branches

Branch	Commit	Author	Date
master	fe2a9aa	modify.readMeTest by 10gi_coach	· 8 days ago
develop	52ddfe4	modify.readMe by 10gi_coach	· 8 days ago
	cd13a33	imitial commit by lab1.ssafy	· 2 weeks ago

[More »](#)

그린핑거스

반11려식물 종합 관리 서비스

카메라를 통해 식물을 인식하여 올바른 관리 방법을 제공하며, 주기별 알람을 통해 식물 관리에 편의성을 높여 줍니다. 또한, 식물의 성장 기록을 관리할 수 있는 성장 일기 서비스를 제공해 식물 키우기에 대한 흥미를 잃지 않도록 돕습니다.

더 나아가, AR 식물 배치 기능, MBTI별 식물 추천 서비스 등을 통해 사용자의 흥미를 유발하고자 노력하는 서비스입니다.

Gerrit 코드 리뷰 절차



Local -> Gerrit

- 리뷰를 요청하지 않고 일반적인 방식으로 Push를 하게 되면 리뷰 오픈이 되지 않고 바로 Gerrit에 변경 사항 반영
 - git push origin <branch>
- 리뷰 오픈을 위해서는 git push 시 refspec을 지정된 패턴으로 입력해서 push를 진행

```
git push <remote명> HEAD:refs/for/ <branch명>
```

Gerrit 코드 리뷰 절차



Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

	Owner	Reviewers	Repo	Updated	Size	Status	CR
Open	irigin test	ssafy02	wishMe	master	4:31 PM	XS	∅ 1 missing
Merged							∅
Abandoned							

Gerrit 코드 리뷰 절차



Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

The screenshot shows a Gerrit Code Review interface for a patchset titled "modify origin test". The patchset has a status of "CODE-REVIEW+2" and is ready for "REBASE". The "REVIEW" button is visible. The "Files" section shows a single file, "README.md", which has been modified ("M"). The "Comments" section shows a single comment from "ssafy02" with a green checkmark and a red minus sign, indicating a positive review. The "Change Log" section is empty. At the bottom, it shows "ssafy02 Uploaded patch set 1." and "VIEW DIFF Patchset 1 | 4:31 PM".

Gerrit 코드 리뷰 절차



✓ Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

The screenshot shows the Gerrit Code Review interface with a patch set titled "modify origin test". The patch set has a score of -2, -1, 0, +1, and +2. The status message is "This shall not be submitted".

Below this, there are five more review rows, each with a different score and a corresponding status message:

- Code-Review: -2, -1, 0, +1, +2. Status: *I would prefer this is not submitted as is*
- Code-Review: -2, -1, 0, +1, +2. Status: *No score*
- Code-Review: -2, -1, 0, +1, +2. Status: *Looks good to me, but someone else must approve*
- Code-Review: -2, -1, 0, +1, +2. Status: *Looks good to me, approved*

The left sidebar shows the project navigation and the current patch set details.

Gerrit 코드 리뷰 절차



✓ Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

The screenshot shows the Gerrit web interface for a code review. At the top, there's a navigation bar with links for CHANGES, YOUR, DOCUMENTATION, and BROWSE. A red box highlights the 'SUBMIT' button, which is currently inactive (grayed out). Below the navigation is a 'Change Info' section with fields for Owner (ssafy02), Author (10gi_coach), Committer (10gi_coach), and Reviewers (ssafy03). The 'Repo | Branch' field shows 'wishMe | master'. Under 'Submit Requirements', there are two green 'Code-Review +2' buttons. The main area shows a 'Files' tab with a single file 'README.md' and a 'Comments' tab with one resolved comment. The 'Patchset 1' section shows a diff with a '+' sign and a '-' sign, indicating changes made. The 'Change Log' at the bottom tracks activity: ssafy02 uploaded patch set 1, ssafy03 was added as a reviewer, and ssafy03 submitted a code review with a +2 rating.

Comments	Size	Delta
[diff content]	+1	-1
	+1	-1

ssafy02	Uploaded patch set 1.	VIEW DIFF	Patchset 1 4:31 PM
ssafy03	Added to reviewer: ssafy03		Patchset 1 4:45 PM
ssafy03	Code-Review +2	1 test	Patchset 1 4:45 PM

Gerrit 코드 리뷰 절차



✓ Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

The screenshot shows two side-by-side Gerrit Code Review interfaces. Both are for a change titled "modify origin test".

Left Side (Initial State):

- Header: Gerrit CHANGES YOUR DOCUMENTATION
- Status Bar: Ready to submit, 61 reviews, modify origin test
- Change Info: Owner ssafy02, Author 10gi_coach, Committer 10gi_coach, Reviewers ssafy03 +2, Repo | Branch wishMe | master
- Code-Review: +2
- Files: README.md
- Comments: 1 resolved
- Base: Patchset 1, 405ee26

Right Side (After Review):

- Header: Gerrit CHANGES YOUR DOCUMENTATION BROWSE
- Status Bar: Merged, 61 reviews, modify origin test
- Change Info: Owner ssafy02, Author 10gi_coach, Committer 10gi_coach, Reviewers ssafy03 +2, Repo | Branch wishMe | master
- Code-Review: +2
- Files: README.md
- Comments: 1 resolved
- Base: Patchset 1, 405ee26
- Download: DOWNLOAD EXPAND ALL
- Change Log:
 - ssafy02: Uploaded patch set 1.
 - ssafy03: Added to reviewer: ssafy03
 - ssafy03: Code-Review +2, 1 test
 - ssafy02: Uploaded patch set 1.
 - ssafy03: Added to reviewer: ssafy03
 - ssafy03: Code-Review +2, 1 test
 - ssafy03: Change has been successfully merged
- View Diff: Patchset 1 | 4:31 PM
- Patchset 1 | 4:45 PM
- Patchset 1 | 4:45 PM
- Patchset 1 | 4:46 PM

Gerrit 코드 리뷰 절차



✓ Code Review

- 오픈 된 리뷰의 내용을 확인하고 평가
- 리뷰어로부터 받은 중 2점인 점수가 있다면 코드를 승인 할 수 있는 Submit 버튼이 활성화
- 리뷰어로 받은 점수 중 -2 점이 있다면 코드 반영 불가.

The screenshot shows the Gerrit interface for managing code reviews. On the left, there are two side-by-side pull request details:

- Left PR:** Status: Ready to submit. Change Info: modify origin test. Owner: ssafy02, Author: 10gi_coach, Committer: 10gi_coach, Reviewers: ssafy03. A green "Code-Review +2" button is present. The commit message is "modify origin test".
- Right PR:** Status: Merged. Change Info: modify origin test. Owner: ssafy02, Author: 10gi_coach, Committer: 10gi_coach, Reviewers: ssafy03. A blue "REPLY" button is present. The commit message is "modify origin test".

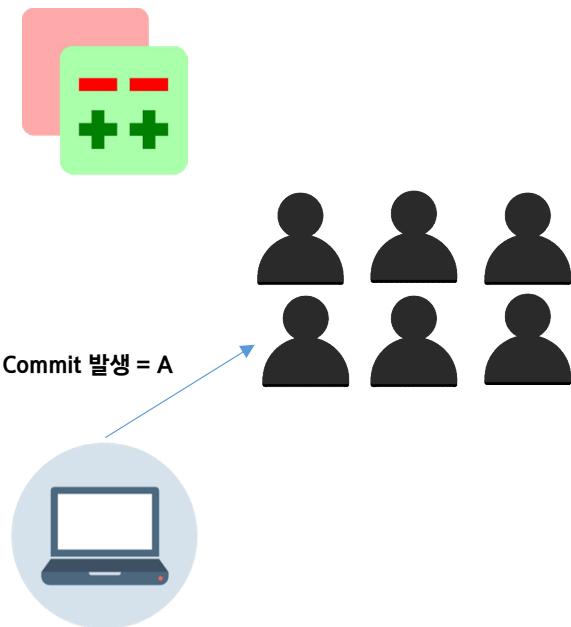
Below these, a list of changes is shown:

File	Comments	Submitted	Owner	Author	Committer	Reviewers	Repo Branch	Commit Message	Time Ago
.idea	modify readMe	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	7 hours ago
client	[README/Update] 리드미 작성2	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	2 months ago
data	Docker-Compose Setting and Jenkins PipLine Setting	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	3 months ago
docs	Update README.md, docs/Wish Me 최종 발표 자료.pdf, doc...	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	2 months ago
exec	[Docs] 시연시나리오 내용 작성	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	2 months ago
img	[README/Update] 리드미 작성4	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	2 months ago
jenkins	modify jenkins	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	4 hours ago
server	modify myletter gradle add	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	5 hours ago
기획	Update 기획/231019.md	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	3 months ago
.gitignore	DockerComposr And Pipline	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	3 months ago
README.md	modify origin test	4:46 PM	ssafy02	10gi_coach	10gi_coach	ssafy03	wishMe master	modify origin test	17 minutes ago

Gerrit 참조 사항



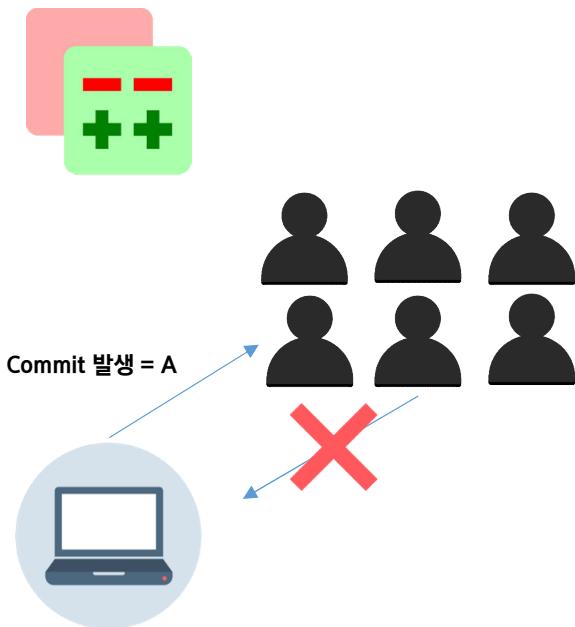
Code Review - 기본 설정



Gerrit 참조 사항



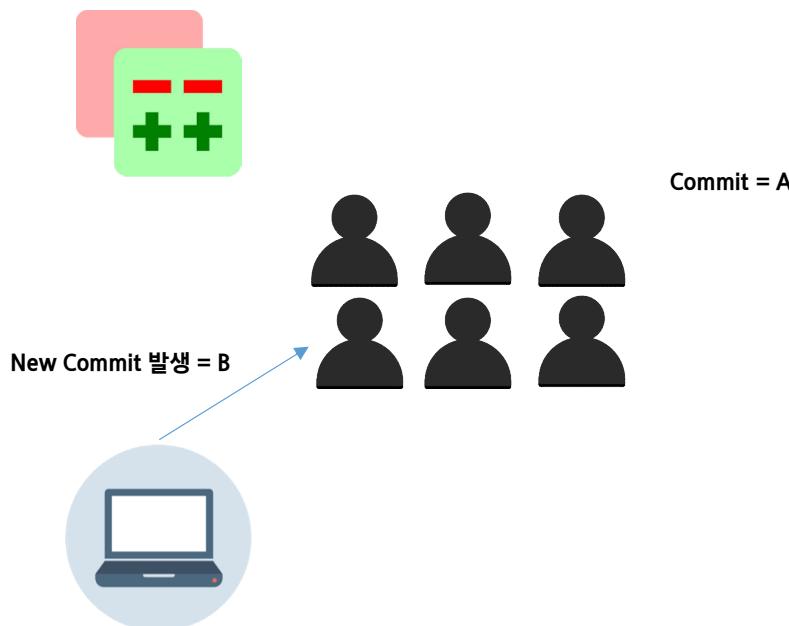
Code Review - 기본 설정



Gerrit 참조 사항



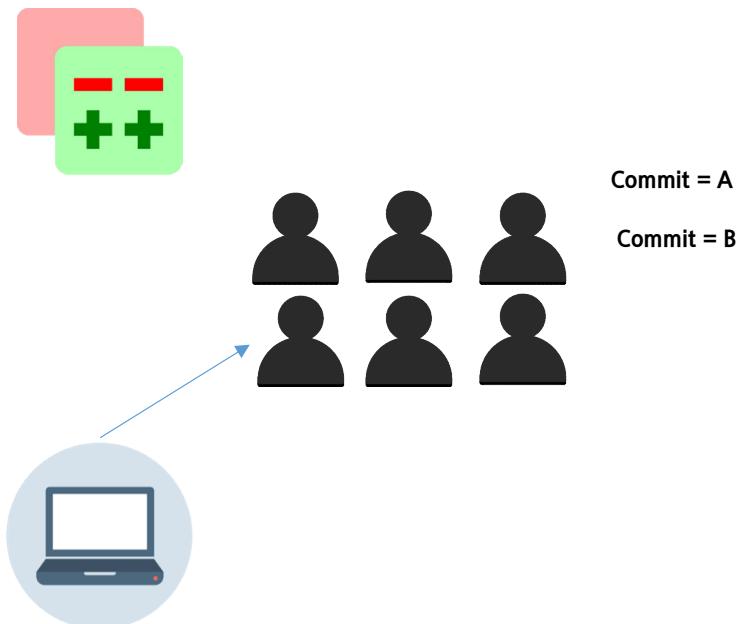
Code Review - 기본 설정



Gerrit 참조 사항

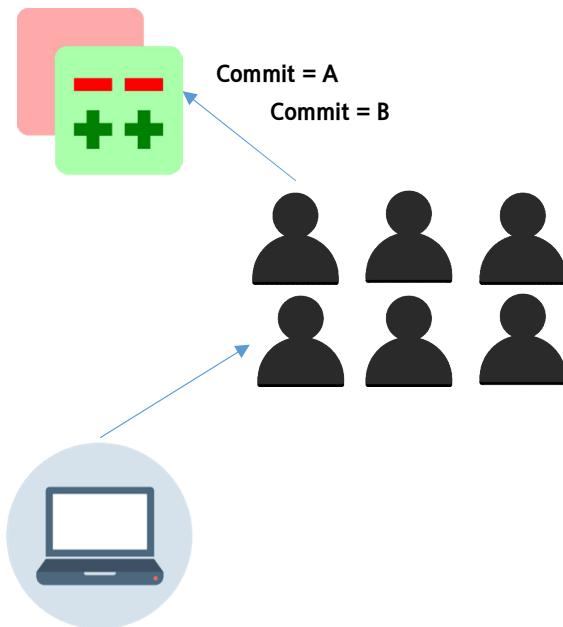


Code Review- 기본 설정



Gerrit 참조 사항

Code Review - 기본 설정



Gerrit CHANGES YOUR DOCUMENTATION BROWSE

Merged [modify origin test](#)

Change Info SHOW ALL REPLY

Submitted Jan 04 Owner [ssafy02](#)
Author [10gi_coach](#) Committer [10gi_coach](#)
Reviewers [ssafy03 -1](#) [ssafy01 +2](#) [ssafy04 +1](#)
[ssafy05 +1](#) [ssafy06 +1](#) Comments [7 resolved](#)
Repo | Branch [wishMe](#) | [master](#)

Submit Requirements

[Code-Review](#) [+2] [+1] [-1]

Relation chain

```
modify_origin_test2 ✓ (Merged)
→ modify_origin_test ✓ (Merged)

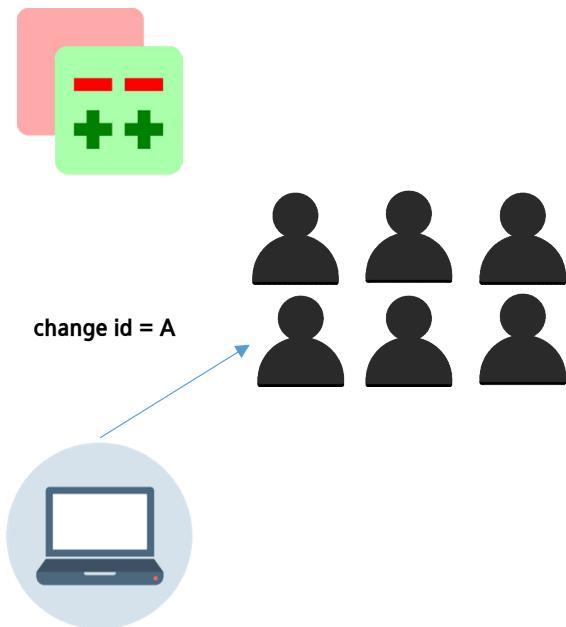
Submitted together
modify_origin_test2 wishMe | master
→ modify_origin_test wishMe | master
```

Gerrit 참조 사항



Code Review(참조) - change id 설정

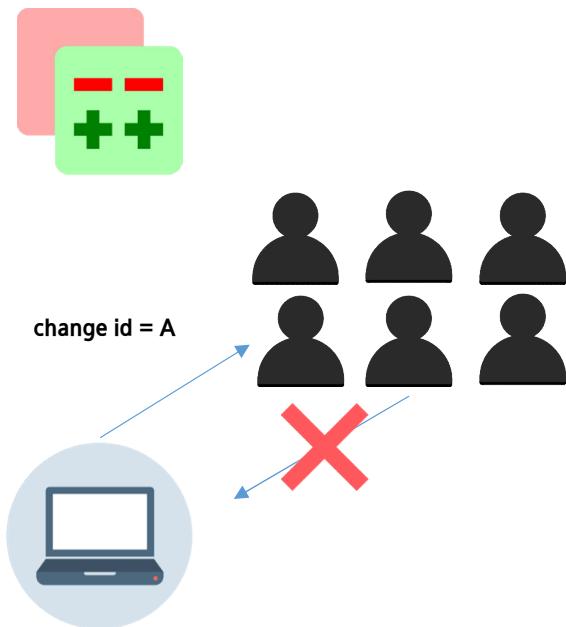
- 만약 동일 한 Commit 건에 대해서 코드 리뷰를 진행하고 싶다면?



Gerrit 참조 사항



Code Review(참조) - change id 설정

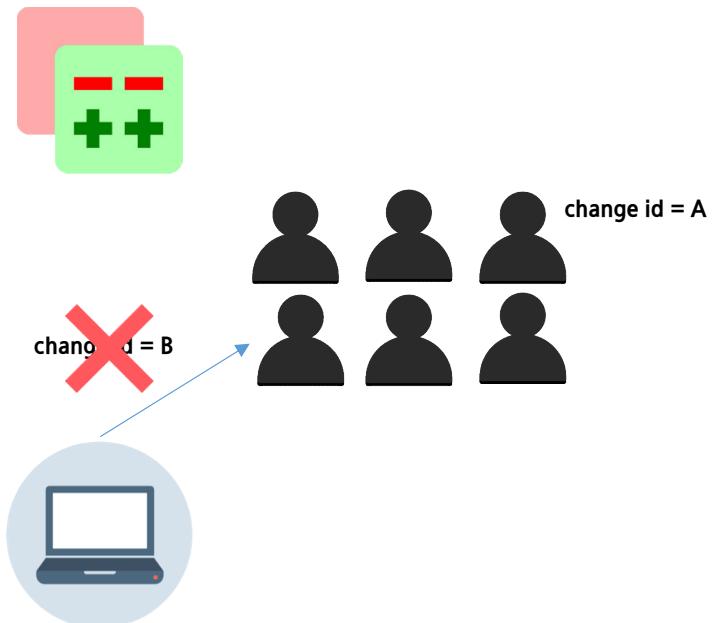


Gerrit 개념



Code Review(참조) - change id 설정

- 코드 수정 후 다른 Commit 을 다시 보내는 것이 아님
- Change id 를 입력해서 해당 Commit 을 수정함을 명시

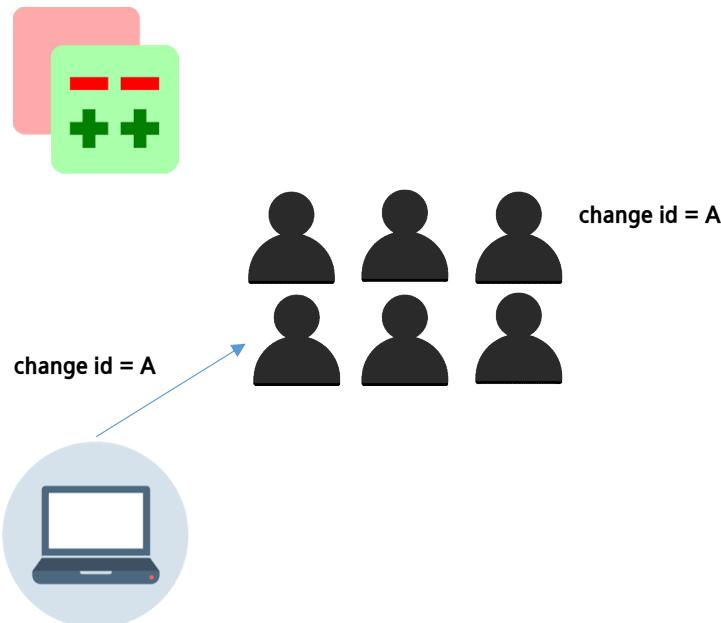


Gerrit 개념



Code Review(참조) - change id 설정

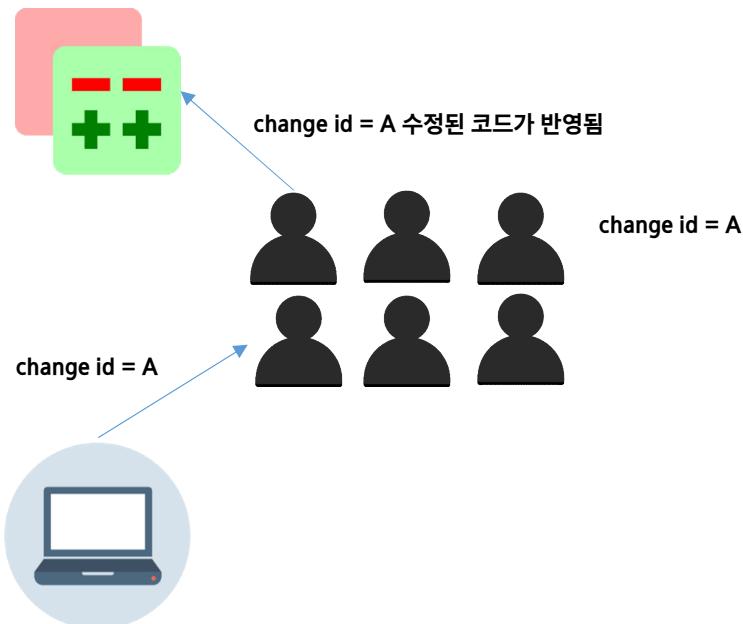
- 코드 수정 후 다른 Commit 을 다시 보내는 것이 아님
- Change id 를 입력해서 해당 Commit 을 수정함을 명시



Gerrit 개념

Code Review(참조) - change id 설정

- 코드 수정 후 다른 Commit 을 다시 보내는 것이 아님
- Change id 를 입력해서 해당 Commit 을 수정함을 명시



Merged ★ 103 code modify

Change Info SHOW ALL ↗ REPLY

Submitted Jan 13
Owner ssafy01 +2
Author 10gi_coach
Committer 10gi_coach
Reviewers
Repo | Branch test | master

Comments No comments

Submit Requirements

Code-Review +2

code modify

Change-ID: I9cb1fb69ee4f161ea521365a08f8137b2a95aeaf

Gerrit 개념



Code Review(참조) - change id 설정

- Gerrit에서 Change id 활성화
- Local PC Hook 설정

The screenshot shows the Gerrit web interface with the following details:

- Header:** Gerrit, CHANGES, DOCUMENTATION, BROWSE.
- Left Sidebar (Repositories):** All-Projects, General (highlighted), Access, Commands, Branches, Tags, Dashboards.
- Central Content:**
 - All-Projects:** BROWSE, VIEW CHANGES.
 - Download:** Clone with commit-msg hook:
\$ git clone "https://[REDACTED].afy.io:8989/All-Projects" && (cd "All-Projects" && [REDACTED])
Clone:
\$ git clone "https://[REDACTED].afy.io:8989/All-Projects"
 - Configurations:** Description: Access inherited by all other projects.
 - Repository Options:**
 - State: Active
 - Submit type: Merge if necessary
 - Allow content merges: True
 - Create a new change for every commit not in the target branch: Inherit
 - Require Change-Id in commit message: True (highlighted with a red box)
 - Reject implicit merges when changes are pushed for review: Inherit
 - Enable adding unregistered users as reviewers and CCs on: Inherit

Gerrit 개념



Code Review(참조) - change id 설정

- Gerrit에서 Change id 활성화
- Local PC Hook 설정

The screenshot shows the Gerrit web interface for the 'wishMe' repository. The 'General' tab is selected under 'Repositories'. In the 'Clone with commit-msg hook' section, there is a red box around the command:

```
s git clone "https://[REDACTED].p.ssafy.io:8989/a/wishMe" && (cd "wishMe" && m
```

Below this, another command is shown:

```
s git clone "https://[REDACTED].p.ssafy.io:8989/a/wishMe"
```

On the right side, a terminal window displays the execution of the command:

```
SSAFY@DESKTOP-LL832F4 MINGW64 ~/Desktop/coach12/code/wishMe/server/myLetter (master)
$ git clone "https://[REDACTED].p.ssafy.io:8989/a/wishMe" && (cd "wishMe" && mkdir -p `git
Cloning into 'wishMe'...
remote: Counting objects: 7763, done
remote: Finding sources: 100% (7763/7763)
Receiving objects: 100% (7763/7763), 118.75 MiB | 9.30 MiB/s, done.

Resolving deltas: 100% (3961/3961), done.
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100  3013  100  3013    0     0   8996      0 --:--:-- --:--:-- 8994
```

Gerrit 개념



☑ Code Review(참조) - change id 설정

- git commit --amend를 통해서 change-id 수정

The screenshot shows the Gerrit web interface for a code review. On the left, the 'Change Info' page is displayed with the following details:

- Owner: ssafy01
- Author: 10gi_coach
- Committer: 10gi_coach
- Reviewers: ssafy02 (2 pending reviews)
- Repo / Branch: test / master
- Submit Requirements: None
- Code-Review: 2 pending reviews

Below this, the 'Files' tab is selected, showing a single file: demo/src/main/java/com/example/demo/config/RedisConfig.java.

On the right, the terminal output of the 'git commit --amend' command is shown:

```
amend ver

Change-Id: I586a41239aeae61fc611f5691dc36cd31d63a39f

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sat Jan 20 18:11:29 2024 +0900
#
# On branch master
# Your branch is ahead of 'origin/master' by 5 commits.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:   src/main/java/com/example/demo/config/RedisConfig.java
```

Gerrit 개념



Code Review(참조) - change id 설정

- git commit --amend를 통해서 change-id 수정

Ready to submit ★ 110 amend ver

Change Info SHOW ALL REPLY

Owner ssafy01
Author 10gi_coach
Committer 10gi_coach
Reviewers ssafy02 +2

Repo | Branch test | master

Submit Requirements

Code-Review +2

Files Comments

Base → Patchset 2 → 5569230

File Commit message

M demo/src/main/java/com/example/demo/config/RedisConfig.java

Change Log

Show all entries (1 hidden)

ssafy02 Added to reviewer: ssafy02

ssafy02 Code-Review -2

ssafy01 Uploaded patch set 2. Copied Votes: * Code-Review-2 (copy condition: "changekind:NO_CHANGE OR changekind:TRIVIAL_REBASE OR **is:MIN**")

ssafy02 Code-Review +2

실습



실습

- 코드 리뷰 실습
- 동일 커밋 건에 대한 코드 리뷰

10기 여러분의 “폭풍 성장을 응원합니다!!”

궁금한 사항은 각 반 코치님들 또는 제게 MM 주세요 😊

