

Development Guide

- CI/CD
 - GitHub Flow `[[[` `]]]`

CI/CD

GitHub Flow



- 1. GitHub Flow 
- 2.   
- 3.  
- 4.  Git 
- 5.   
- 6. GitLab MR(Merge Request) 
- 7.     
- 8.    

GitHub Flow

GitHub Flow     ,    (CI/CD)  .
    Main( Master)         .

GitHub Flow

- 1. **Main**     
- 2.     
- 3.      
- 4. **Merge Request(MR)**   
- 5. **Main**   



GitHub Flow     :

- 1. **Main**  :    
- 2.   : Main    ,   



GitLab Flow 是一个简单且直观的分支策略，它允许开发人员快速部署生产环境，并支持团队协作开发。其核心原则是：

- `feature/login-page` : 用于开发新功能
- `bugfix/login-error` : 用于修复生产环境中的问题
- `hotfix/security-issue` : 用于快速响应安全漏洞
- `docs/api-documentation` : 用于更新 API 文档
- `refactor/user-service` : 用于重构现有代码



GitHub Flow 是一个简单且直观的分支策略，它允许开发人员快速部署生产环境，并支持团队协作开发。其核心原则是：

1. **Main** 分支始终处于生产就绪状态
2. 开发新功能时，从 Main 分支创建 feature 分支
3. 在 feature 分支上进行开发
4. 开发完成后，将 feature 分支合并回 Main 分支
5. **Merge Request(MR)** 用于代码审查
6. 通过 MR 批准后将代码合并回 Main 分支
7. 部署到生产环境
8. **Main** 分支始终处于生产就绪状态
9. 修复生产环境中的问题
10. 修复完成后，将代码合并回 Main 分支并部署到生产环境

Git 分支策略

1. 克隆仓库

```
# 克隆仓库
git clone https://gitlab.company.com/project/repository.git
cd repository
```

```
# 克隆仓库并指定分支
git clone --single-branch --branch [branch-name] https://gitlab.company.com/project/repository.git [my-directory]
```

```
cd my-directory
```

2. 创建分支

```
# 创建分支
git branch

# 列出所有分支
git branch -a

# 切换到主分支
git checkout main
git pull

# 创建新分支
git checkout -b feature/user-authentication

# 切换到新分支
git checkout feature/user-authentication
```

3. 提交代码

```
# 查看当前状态
git status

# 添加文件
git add src/components/LoginForm.js

# 添加所有文件
git add .

# 提交代码
git commit -m "添加 UI 组件"

# 查看提交历史
git log
git log --oneline --graph
```

4. 分支 分支 分支

```
# 分支 分支 分支
git push origin feature/user-authentication

# 分支 分支 分支 分支
git pull origin feature/user-authentication

# 分支 分支 分支
git remote -v
```

5. 分支 分支

```
# Main 分支 分支
git checkout main
git pull

# 分支 分支 分支 (分支 分支 MR 分支 分支 )
git merge feature/user-authentication

# 分支 分支 (分支 )
git branch -d feature/user-authentication

# 分支 分支 (分支 )
git push origin --delete feature/user-authentication
```



分支 分支 分支 分支 分支 GitHub Flow 分支 分支 分支 :

1. 分支 分支 分支

```
# 分支 分支 分支
git checkout main
git pull
```

```
# 创建分支并切换到该分支
git checkout -b feature/login-implementation
```

2. 创建并添加文件

```
# 查看当前分支
# ...

# 查看文件状态
git status

# 添加文件到暂存区
git add src/components/Login.js
git commit -m "添加登录组件"

# 查看文件状态
# ...

# 添加文件到暂存区
git add src/api/authService.js
git commit -m "添加API服务"

# 查看文件状态
# ...

git add src/styles/login.css
git commit -m "添加登录样式"
```

3. 提交并推送到远程仓库

```
# 推送到远程仓库
git push origin feature/login-implementation
```

4. 合并分支并删除本地分支

```
# 1. Main branch을 체크아웃
git checkout main

2. 변경사항을 가져옴
git pull

3. feature/login-implementation branch를 체크아웃
git checkout feature/login-implementation

4. 변경사항을 병합
git merge main

5. 변경사항을 커밋
# 1. 커밋 메시지 작성
# ... 2. 커밋 메시지 작성 ...

git add .
git commit -m "Main branch에 feature/login-implementation을 병합"

6. 변경사항을 원격 저장소에 푸시
# 1. 원격 저장소 설정
git push origin feature/login-implementation
```

5. MR을 생성하고 리뷰를 받기

```
# MR을 생성하고 리뷰를 받기
1. MR을 생성
git checkout main
git pull
git branch -d feature/login-implementation
```

GitLab에서 MR을 생성하고 리뷰를 받기

GitLab에서 Merge Request(MR)를 생성하고 리뷰를 받기

MR을 생성하고 리뷰를 받기

1. GitLab에서 MR을 생성하기 위해 로그인
2. MR을 생성할 브랜치 "Merge Requests"를 선택
3. "New merge request" 버튼을 클릭
4. MR을 생성할 브랜치 (feature/login-implementation)를 선택 (main)을 선택
5. "Compare branches and continue" 버튼을 클릭
6. MR을 생성, 리뷰, 승인, 폐기
7. "Submit merge request" 버튼을 클릭

1. 分支 :

```
# 查看分支
git status

# 查看分支 (分支名称)
# <<<<<< HEAD (分支名称)
# ===== (分支名称)
# >>>>>> feature/branch (分支名称)

# 添加文件
git add .
git commit -m "分支名称"
```

2. 分支 Main 分支 分支

分支 : GitHub Flow 分支 Main 分支 分支

分支 :

```
# 创建分支
git checkout -b feature/intended-changes
git push origin feature/intended-changes

# Main 分支
git checkout main
git reset --hard origin/main
git pull
```

3. 分支 分支 分支

分支 : 分支 分支 分支 分支

分支 :

```
# 分支
git commit --amend -m "分支名称"
```

提交 分支 名称 (分支 : 分支 名称 名称)

git push origin feature/branch --force



1. 提交 分支 名称 : 分支 名称 名称 名称 , 提交 分支 名称
2. 提交 分支 名称 : 提交 分支 名称 名称 名称 名称
3. 提交 分支 名称 : 提交 分支 名称 名称 名称 名称
4. 提交 分支 名称 : "提交 分支 名称" 名称 "提交 分支 名称" 名称 名称 名称 名称

MR 提交 提交 提交

1. 提交 分支 名称 : MR 提交 分支 名称 名称 名称 名称
2. 提交 分支 名称 : 提交 分支 名称 名称 名称 名称 名称
3. 提交 分支 名称 : 提交 分支 名称 MR 提交 分支 名称 MR 提交 分支 名称
4. 提交 分支 名称 : 提交 分支 名称 名称 名称 名称 名称 名称

提交 分支 名称 GitHub Flow 提交 分支 名称 提交 分支 名称 提交 分支 名称 . 提交 分支 名称 提交 分支 名称 提交 分支 名称 .