





- 형상 관리 프로그램
- 버전 컨트롤 시스템 (VCS : Version Control System)
- 리눅스 창시자인 **리누스 토발즈**가 2005년에 개발
- 분산형 형상관리 - 서버 없이도 개발이 가능
- 현시점에서 **프로그래머의 기본 소양**
- **극단적으로 복잡하고 어렵다.**



- 원격 저장소를 제공하는 웹 서비스
- 2018년 Microsoft 사가 인수
- 전 세계에서 가장 큰 오픈 소스 저장소 – **오픈소스의 성지**
- 유사 서비스 : GitLab, Bitbucket,...
- 최근 개발자의 포트폴리오 활용



필수 개념 소개

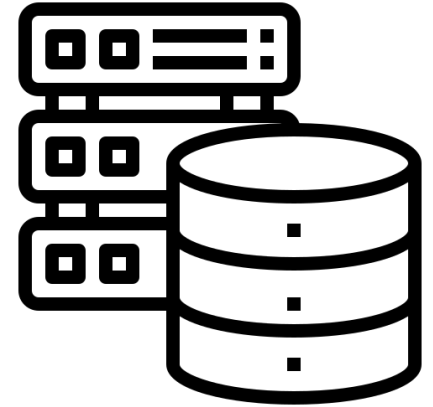
리포지토리(Repository)

- Repo, 저장소

- 프로젝트의 변경 이력을 저장하는 공간, HDD

- 저장 위치에 따른 구분

- **Local Repository** : 지역 저장소, 자신의 컴퓨터에 존재하는 저장소
- **Remote Repository** : 원격 저장소, 다른 컴퓨터(서버)에 있는 저장소



Git 3개의 영역



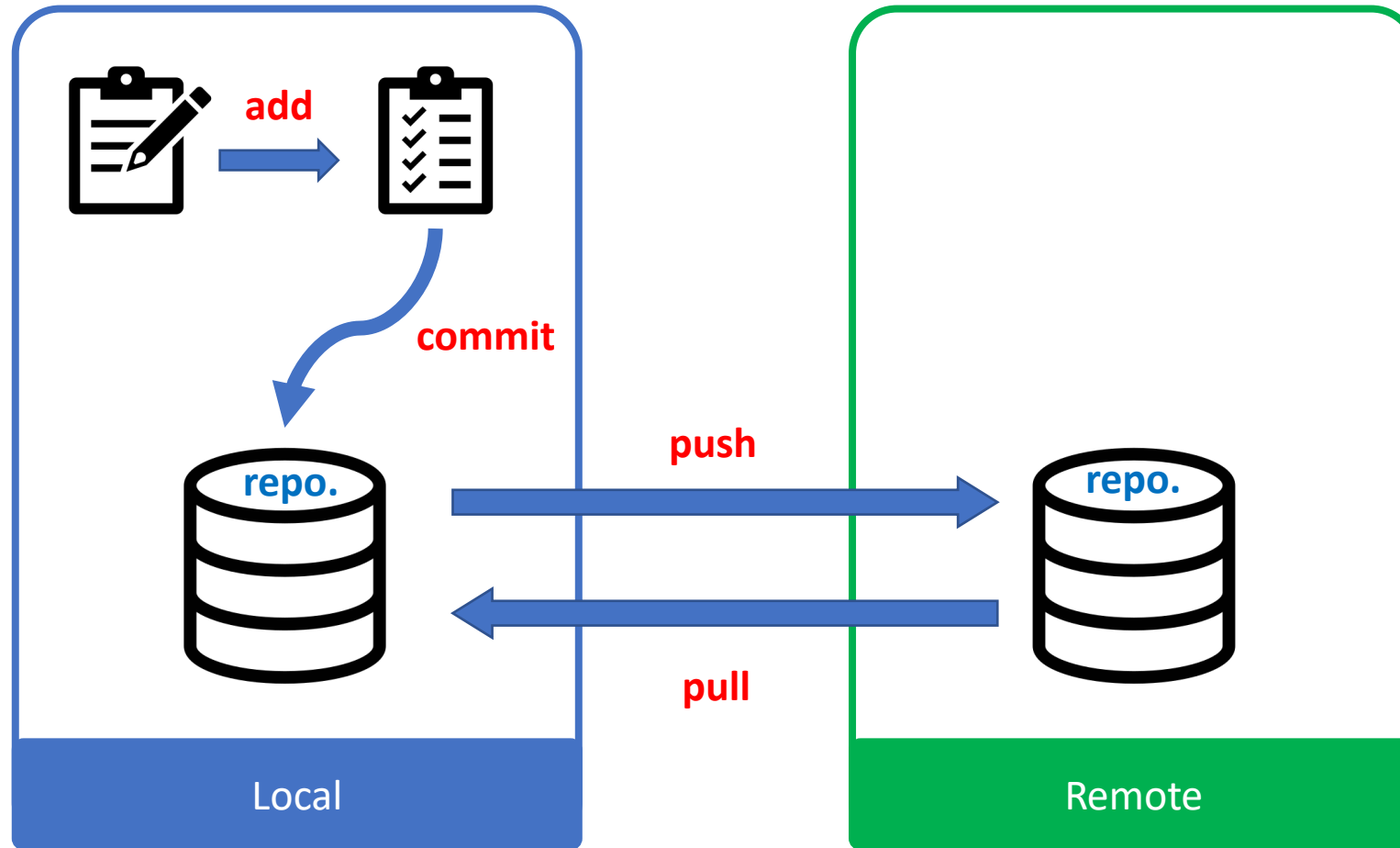
- **Working Directory**
 - 작업 폴더
- **Staging Area**
 - 저장할 대상을 임시 보관하는 영역
 - 논리적인 영역
- **Repository**
 - 변경 이력을 기록하는 영역



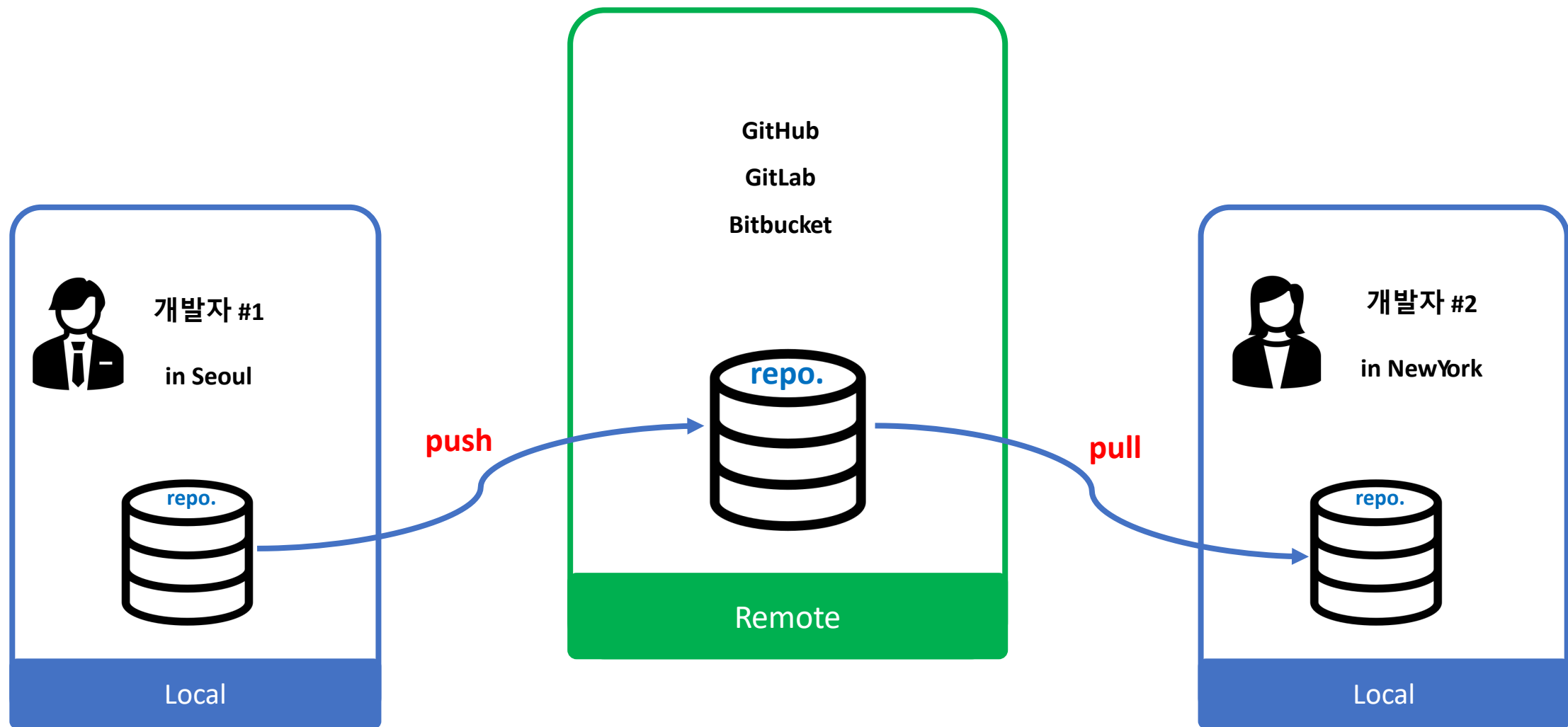
핵심 명령어

git add , git commit

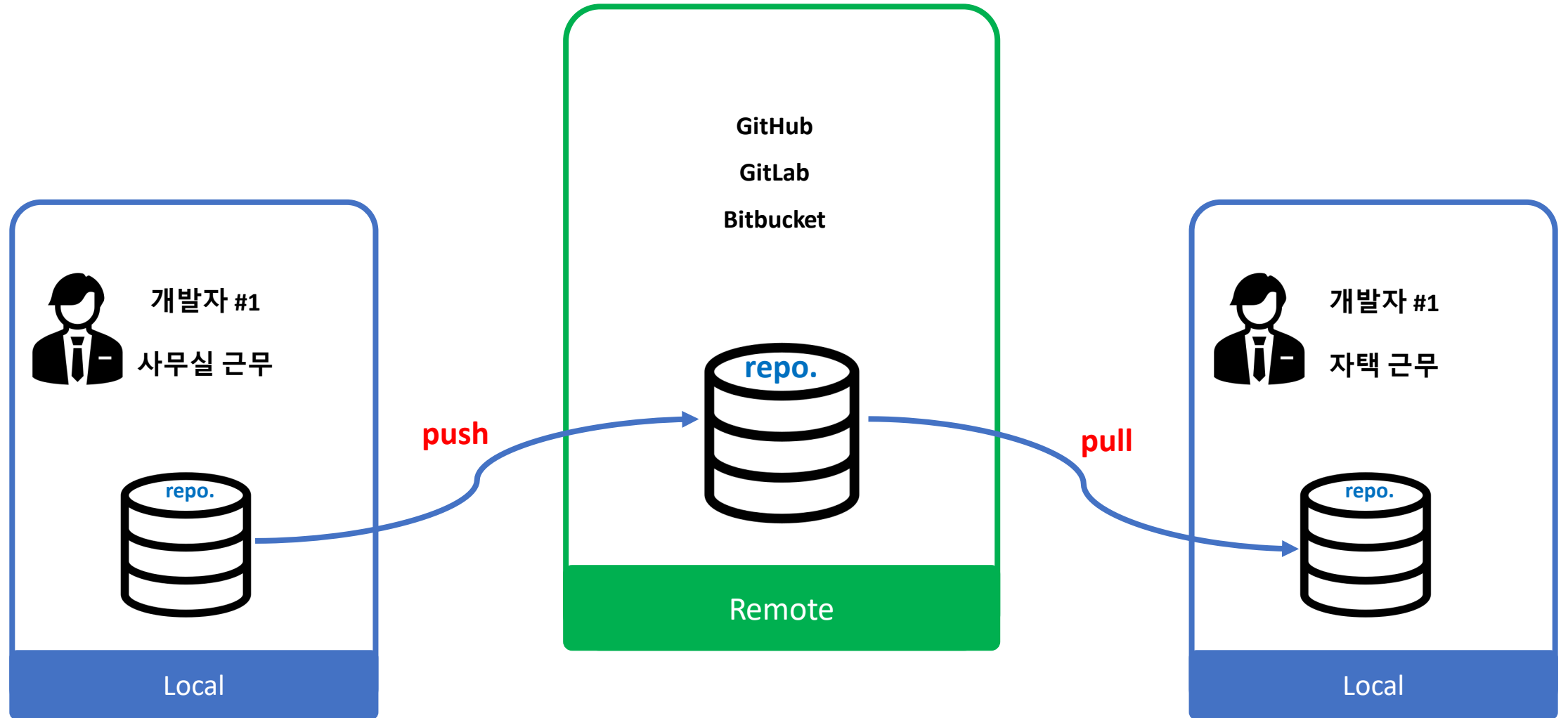
핵심 명령어 – git add, git commit, git push, git pull



원격 저장소가 필요한 이유 - 협업



원격 저장소가 필요한 이유 - 작업 장소의 변경



핵심 명령어 – git add

저장할 파일(또는 변경사항)을 **Staging Area**(임시 보관소)에 추가하는 명령



핵심 명령어 – git add



- 3개의 파일을 추가

핵심 명령어 – git add

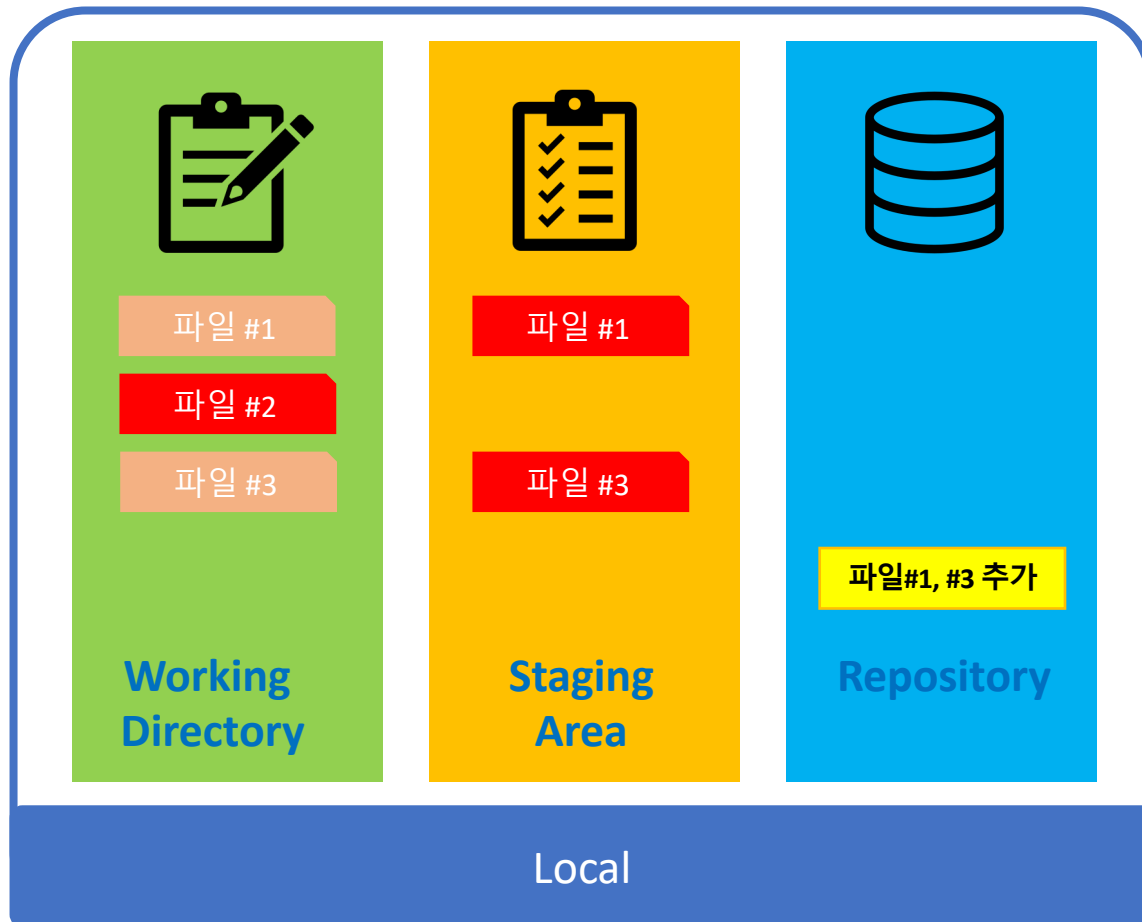


- 추가된 3개의 파일 중 저장소에 이력을 남기려는 파일만 선택해 Staging Area로 추가

```
git add 파일#1
```

```
git add 파일#3
```

핵심 명령어 – git commit

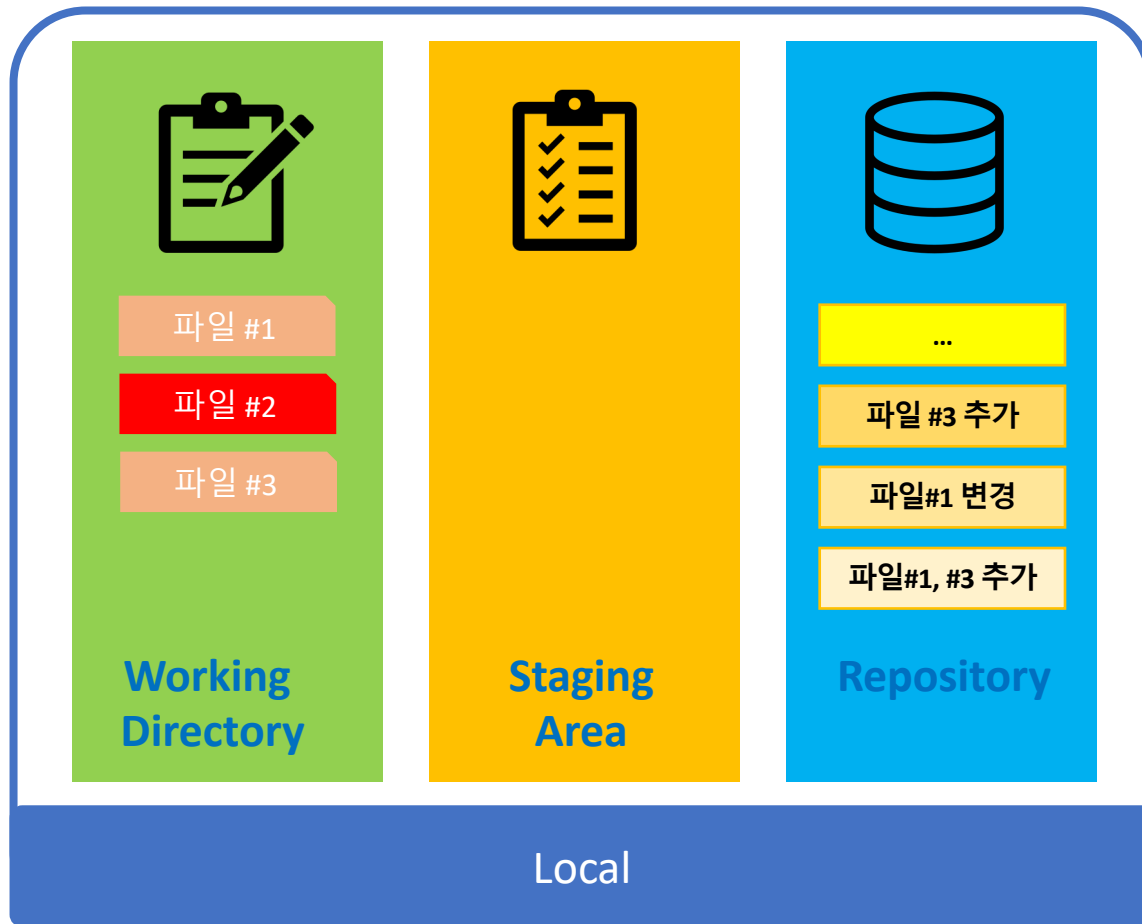


- Staging Area에 추가된 변경내역을 하나의 커밋(기록)으로 리포지토리에 저장

```
git commit -m "커밋 메시지"
```

- 커밋 메시지는 추후 롤백을 위해 의미있는 내용으로 표기해야 함.

핵심 명령어 – git commit



- 리포지토리에 기록된 커밋은 누적된다.
- 원하는 커밋으로 롤백(되돌리기)할 수 있다.



되돌리기

git reset



Branch

필수 Git 명령어

CLI : Command Line Interface

- git init
- git clone
- git remote add
- git add
- git commit
- git status
- git log
- git push
- git pull
- git merge
- git reset / revert
- git branch
- git checkout
- git config
- git clean
- git lfs install/trck

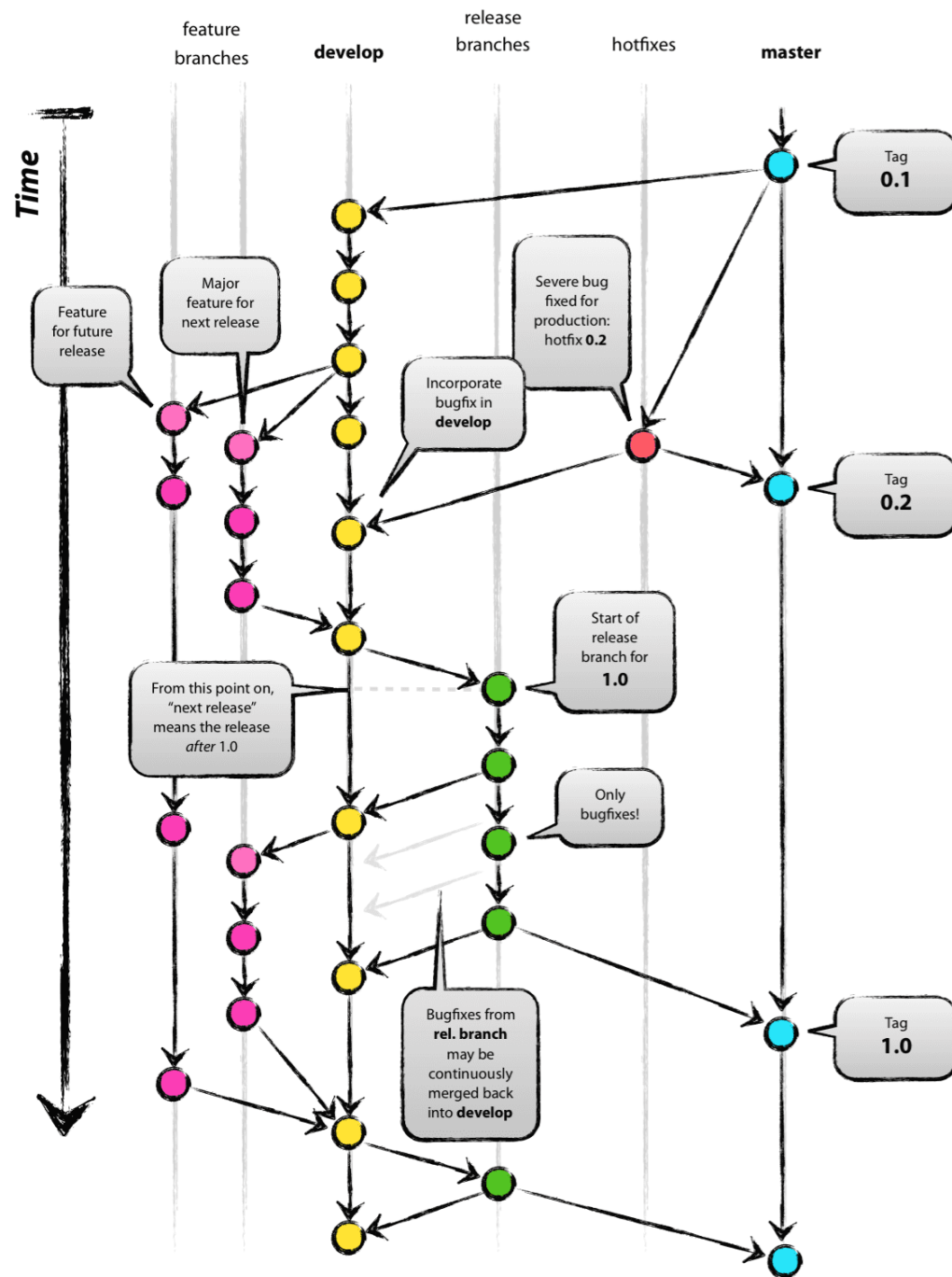
GitFlow 방법론

GitFlow 특징

- Git 을 활용해 협업하는 방법론
- Vincent Driessen 에 의해 제시된 방법
- 절대적인 방법은 아님
- 자신(팀)에 적절하게 수정

Master 브랜치

- 항상 출시 가능한 완성본
- 접근 및 머지는 팀장만 관리

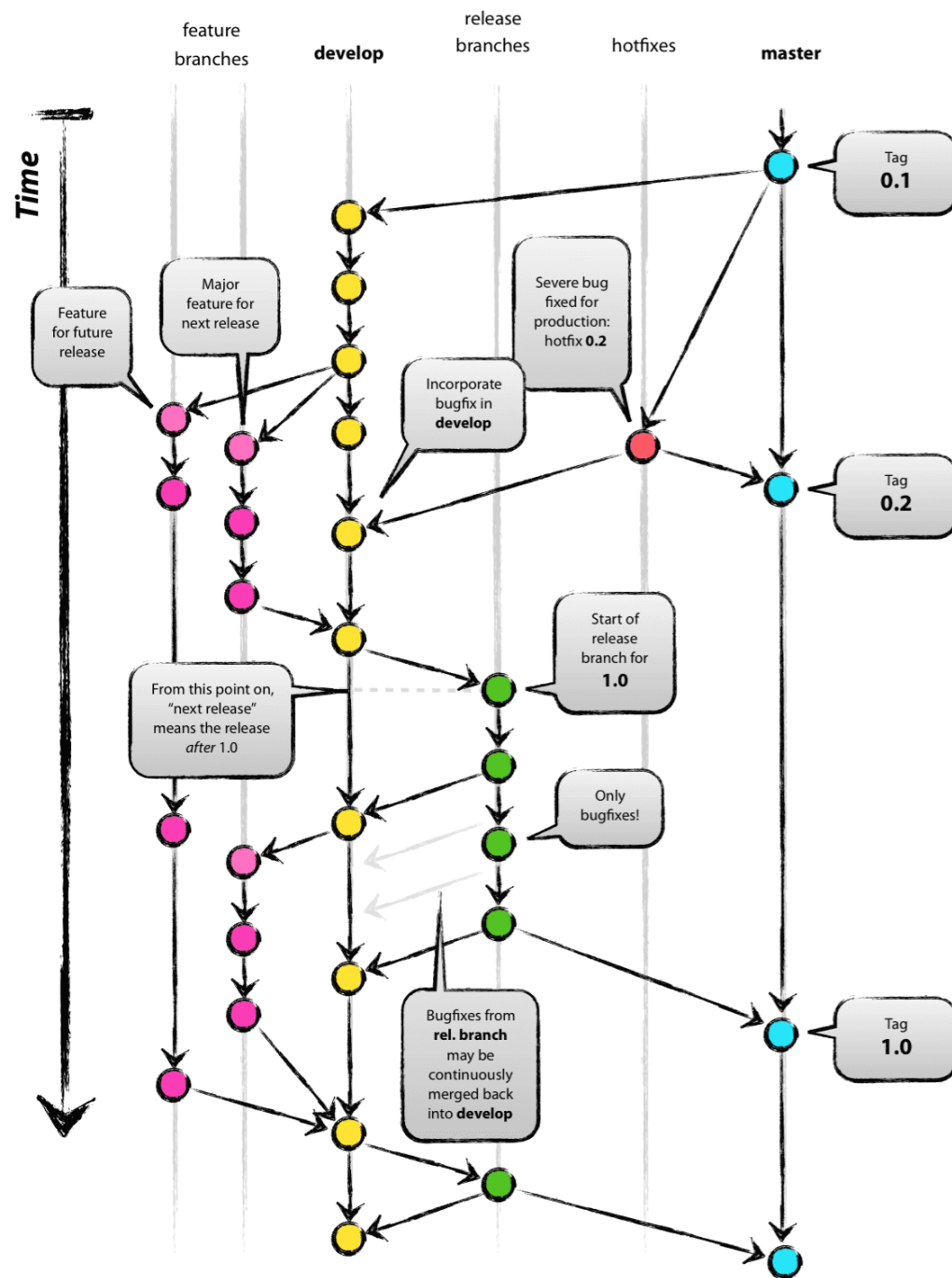


Develop 브랜치

- 팀원 간의 작업물을 병합하는 브랜치
- 오류가 없어야 하는 브랜치
- 주기적으로 Master 브랜치로 병합

개인 개발 브랜치 (dev/LJH)

- 팀원 개인이 작업하는 브랜치
- 오류가 없는 코드를 Develop 브랜치로 병합
- 매일 pull / push 가 이루어져야 함



최종 프로젝트 Git Flow 방법론

