

# 강의 02<sub>/16</sub>: 형상관리 (recap.)

---

신정규  
2017년 3월 9일











# 형상관리

---

협업을 가능하게 한 발전

# 형상 관리 도구 (SCM)

- Software Configuration Management Tools
- Version Control System (VCS)
  - 프로그램, 문서 및 모든 종류의 콘텐츠의 변경 이력을 관리하는 시스템
- “Version” “number”
  - 변경이 될 때 마다 붙는 표지 / 번호
- “Revision”
  - 새로 변경된 결과물
  - 인덱스가 붙음

Name
 발표자료 - 검수.pdf
 발표자료 - 최종.pdf
 발표자료 - 최종2.pdf
 발표자료 - 최종2(수정).pdf
 발표자료 - 최종제출 - 재제출 (수정3).pdf
 발표자료 - 최종제출.pdf
 발표자료 - 최종제출(수정2).pdf
 발표자료 - 최최종.pdf



# 형상관리도구 요소



# 버전 관리 시스템

---

- 버전 관리 시스템 (Version Control System)
  - 과거 및 현재 상태를 모두 저장
  - 상태 변화에 따른 모든 중간 단계 저장
  - 저장소: 소스 (코드, 문서, 그림 등) 를 저장
- 구조에 따른 구분
  - 중앙 집중식: CVS, subversion
  - 분산식: mercurial, git

# 버전 관리 시스템 용어 (1)

---

- 저장소 (repository)
  - 소스 (코드, 문서, 그림 등) 를 저장하는 공간
- 브랜치 (branching / branch)
  - 코드를 중간에 분기하는 행위 / 분기한 결과
- 리비전 (revision)
  - 특정 시간 / 특정 브랜치의 어떤 한 상태
  - 상태에 해당하는 인덱스를 리비전 번호 (revision number) 라고 함



# 버전 관리 시스템 용어 (2)

- 커밋 (commit)
  - 저장소에 변경된 콘텐츠를 반영하는 행위
- 커밋 로그 (commit log)
  - 커밋을 할 때 해당 커밋이 어떠한 변경을 했는지 작성한 기록
- 체크아웃 (checkout)
  - 현재 저장소를 특정한 리비전으로 업데이트
- 태그 (tag)
  - 특정 리비전을 나중에 찾거나 알아보기 쉽게 붙인 텍스트



# 중앙 집중식 버전 관리 시스템

## ■ 특징

- 중심이 되는 저장소가 존재
- 메인 트리가 존재함
- 저장소 의존도가 크므로 branching 을 쉽게 하지 않음
  - 중요한 기능 단위 / 개발 단위로만 branching
  - 의존성 문제로 한 번 만든 브랜치는 쉽게 없애지 않음
- 연속된 숫자를 인덱스로 사용
  - 예) 3125

## ■ 대표적인 오픈소스 VCS

- CVS, Subversion





# 분산식 버전 관리 시스템

## ■ 특징

- 중심이 되는 저장소가 존재하지 않음
- Branching / merging이 매우 자유로움
  - 새로운 코드를 만들거나 수정할 때 branching을 하는 것이 일반적임
- 고유의 리비전 인덱스를 사용
  - 예) 315a2baab56dc69bdd2ba653cfa2a69a8ff0e92c

## ■ 대표적인 오픈소스 VCS

- Mercurial, git



# 이슈 트래커

## ■ 이슈 트래커 (Issue tracker)

- VCS와 연동하여 VCS에 관련한 다양한 작업 담당
- 작업 분담 / 이슈 관리 / 변경 이력 추적 / 문서화

## ■ 오픈소스 이슈 트래커


- Trac (<http://trac.edgewall.org>)
- GitLab (<https://gitlab.com/groups/gitlab-org>)
- Bugzilla (<https://www.bugzilla.org>)
- Redmine (<http://www.redmine.org>)
- Jira (<https://atlassian.com/software/jira>)

## ■ 이슈 트래킹 서비스

- Github (<https://github.com>)
- Bitbucket (<https://bitbucket.org>)
- Codebase (<https://www.codebasehq.com>)
- SourceForge (<https://sourceforge.net>)



# Trac



[Login](#) | [Preferences](#) | [Help/Guide](#) | [About Trac](#)


[Home](#) | **[Trac](#)** | [Genshi](#) | [Babel](#) | [Bitten](#)


[Wiki](#) | **[Timeline](#)** | [Roadmap](#) | [Browse Source](#) | [View Tickets](#) | [New Ticket](#) | [Search](#)


[← Previous Period](#) | [Next Period →](#)


## Timeline


**Oct 16, 2016: Yesterday**


 7:02 PM **Changeset [15184]** by Jun Omae  
1.3.1dev: merge [15183] from 1.2-stable


 7:02 PM **Changeset in mirror [11807:8f865c92c961]** trunk tip by Jun Omae  
1.3.1dev: merge [15183] from 1.2-stable


 7:01 PM **Changeset [15183]** by Jun Omae  
1.2dev: record-only merge [15182]

 7:01 PM **Changeset in mirror [11806:2ca53a00ebd8]** 1.2-stable by Jun Omae  
1.2dev: record-only merge [15182]

 7:00 PM **Changeset [15182]** by Jun Omae  
1.0.14dev/l10n: new extraction (2 added, 1 modified)

 7:00 PM **Changeset in mirror [11805:5da4a0027b69]** 1.0-stable by Jun Omae  
1.0.14dev/l10n: new extraction (2 added, 1 modified)

 4:21 PM **Ticket #12607** ([TracMasterTickets-3.0.1-py2.7] AttributeError: 'NoneType' object has ...) closed by Jun Omae  
cantfix: [PluginIssue](#) ([th:MasterTicketsPlugin](#), [th:r14515](#), [th:#12180](#))

 2:18 PM **Ticket #12607** ([TracMasterTickets-3.0.1-py2.7] AttributeError: 'NoneType' object has ...) created by matthew@...

View changes from

going back  days

by author

☒ Changesets in all repositories

☒ Milestones completed

☒ Tickets opened and closed

☐ Ticket updates

☒ Wiki changes

# 지속적 통합 도구

## ■ 지속적 통합 Continuous Integration (CI)

- 코드의 변경 내용을 확인하고, 서비스를 계속 최신 상태로 유지
- 최근 리비전의 무결성을 확인
  - 코드 오류 검사
    - 단위 테스트 (Unit test) / 기능 테스트 (functional test)
  - 코드의 호환성 검사
- 무결성 결과 보고

## ■ 오픈소스 CI

- Jenkins (<https://jenkins.io>)
- Python Buildbot
- Travis CI (<http://travis-ci.org>)
- Strider (<https://github.com/Strider-CD/strider>)
- ~~Apache Continuum~~ (<https://continuum.apache.org>)  
-은퇴!



# Jenkins

[Back to Dashboard](#)

[Manage Jenkins](#)

[New Node](#)

[Configure](#)

Build Queue

No builds in the queue.

Build Executor Status

#	Status
<a href="#">master</a>	
1	Idle
2	Idle
<a href="#">aws1</a>	
1	Idle
<a href="#">aws2</a>	
1	Idle
<a href="#">aws3</a>	
1	Idle
<a href="#">ciandroid</a>	
1	Idle
<a href="#">ciprod</a>	
1	Idle
<a href="#">ciqa</a>	
1	Idle
<a href="#">cisim1</a>	
1	Idle
<a href="#">cixas</a>	
1	Idle
<a href="#">cixcode1</a>	
1	Idle
<a href="#">cixcode2</a>	
1	Idle

S

Name

i

Architecture

Clock Difference

Free Disk Space

Free Swap Space

Free Temp Space

Response Time

Data obtained

34 min

34 min

34 min

34 min

34 min

34 min

[aws1](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

149ms

[aws2](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

148ms

[aws3](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

131ms

[ciandroid](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

126ms

[ciprod](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

122ms

[ciqa](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

119ms

[cisim1](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

118ms

[cixas](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

99ms

[cixcode1](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

92ms

[cixcode2](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

92ms

[master](#)

Mac OS X (x86\_64)

In sync

513.05 GB

64.00 MB

513.05 GB

0ms

Refresh status

[Help us localize this page](#)

Page generated: Apr 26, 2014 4:33:49 PM [REST API](#) [Jenkins ver. 1.555](#)

# 지속적 통합 도구

---

- VCS, IT, CI 연동
  - 예) git – github – travis CI
  - 예) subversion – trac – jenkins
- 예) Travis CI : 설정한 .travis.yml 을 git 프로젝트의 루트에 넣음
  - 환경 변수 설정
  - 브랜치 지정
  - 빌드 매트릭스 (build matrix) 구성
  - 데이터 스토어 소프트웨어 / 환경 지정

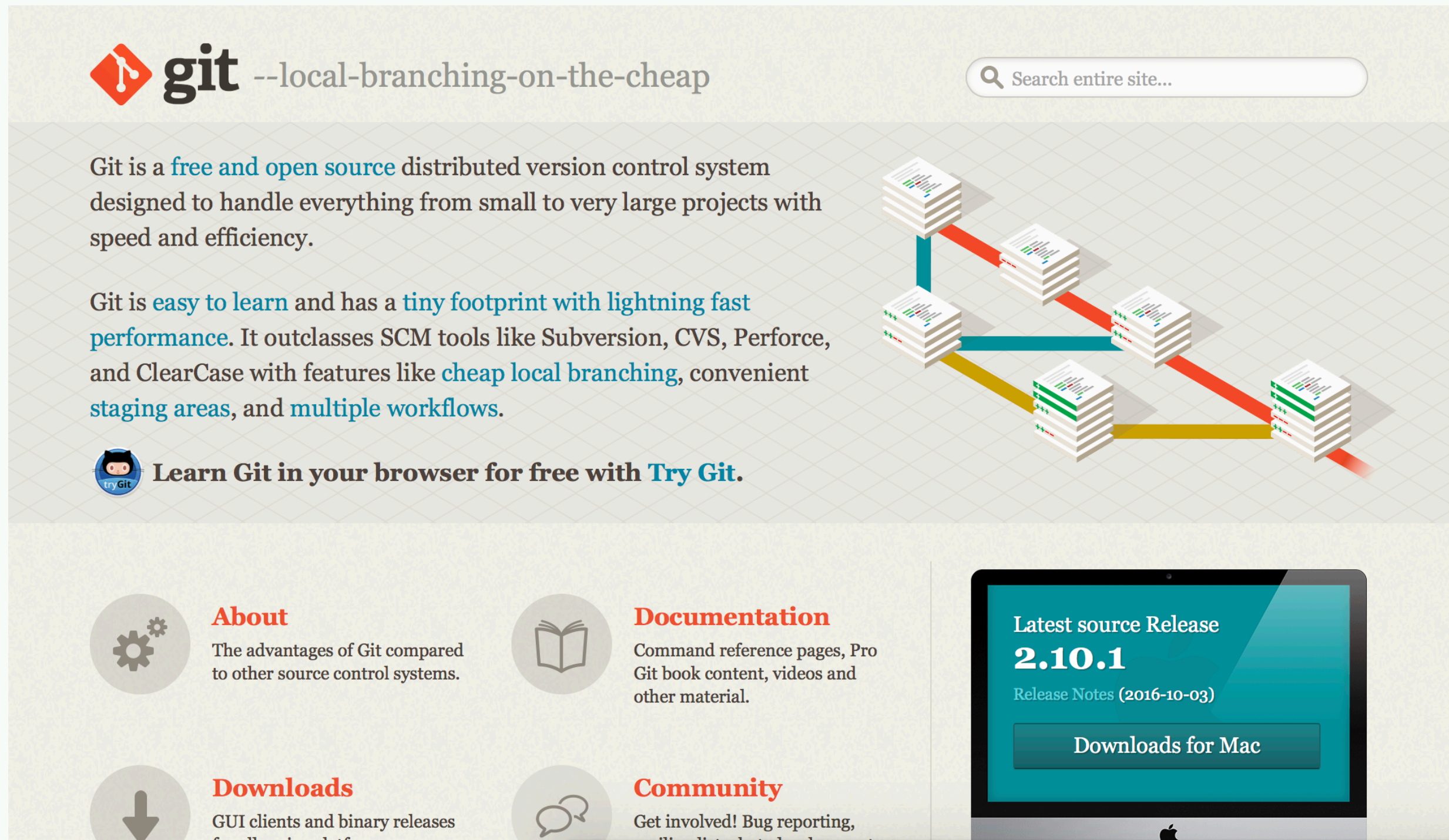
# git

---

오픈소스계의 de facto standard DVCS 소프트웨어



# git




The image shows the Git website homepage. At the top left is the Git logo (a red diamond with a white branching diagram) followed by the text "git --local-branching-on-the-cheap". To the right is a search bar with the placeholder text "Search entire site...". Below the header, there are two paragraphs of text. The first paragraph describes Git as a "free and open source" distributed version control system. The second paragraph describes Git as "easy to learn" and having a "tiny footprint with lightning fast performance". To the right of the text is a diagram showing a branching model with several stacks of papers connected by colored lines (red, blue, yellow). Below the text, there is a "Try Git" button with the GitHub logo. At the bottom, there are four sections: "About" (The advantages of Git compared to other source control systems.), "Documentation" (Command reference pages, Pro Git book content, videos and other material.), "Downloads" (GUI clients and binary releases), and "Community" (Get involved! Bug reporting, ...). On the right side of the bottom section, there is a laptop displaying the "Latest source Release 2.10.1" and a button for "Downloads for Mac".

**git** --local-branching-on-the-cheap

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

 **Learn Git in your browser for free with Try Git.**

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases

**Community**  
Get involved! Bug reporting, ...

Latest source Release  
**2.10.1**  
Release Notes (2016-10-03)  
Downloads for Mac

■ <https://git-scm.com>



# git

---

- 리눅스 커널 개발
  - patch (with diff) 기반의 관리
- 2002년
  - BitKeeper의 리눅스 커뮤니티로의 DVCS 기여
- 2005년
  - BitKeeper의 유료화 정책
  - 새로운 도구의 필요성

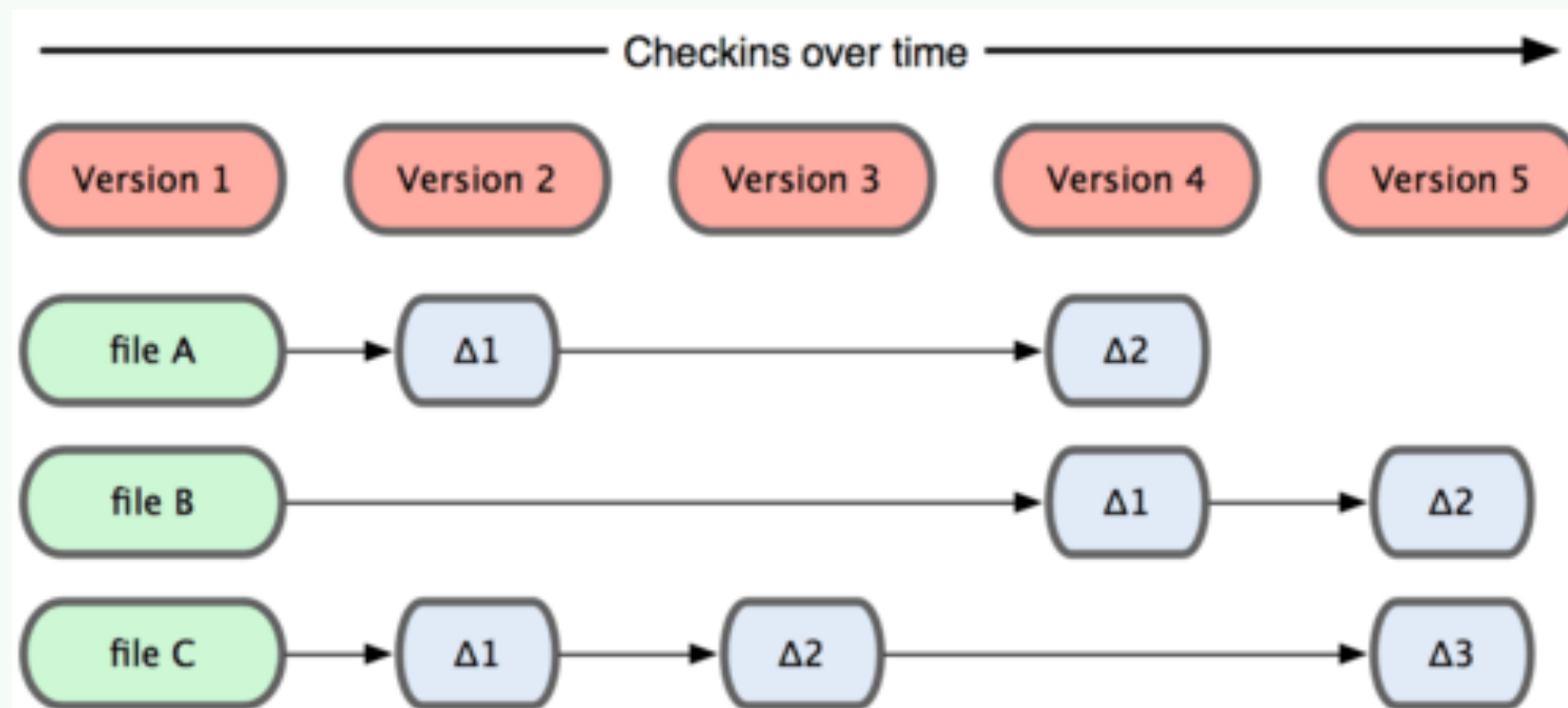
# Git의 목표

---

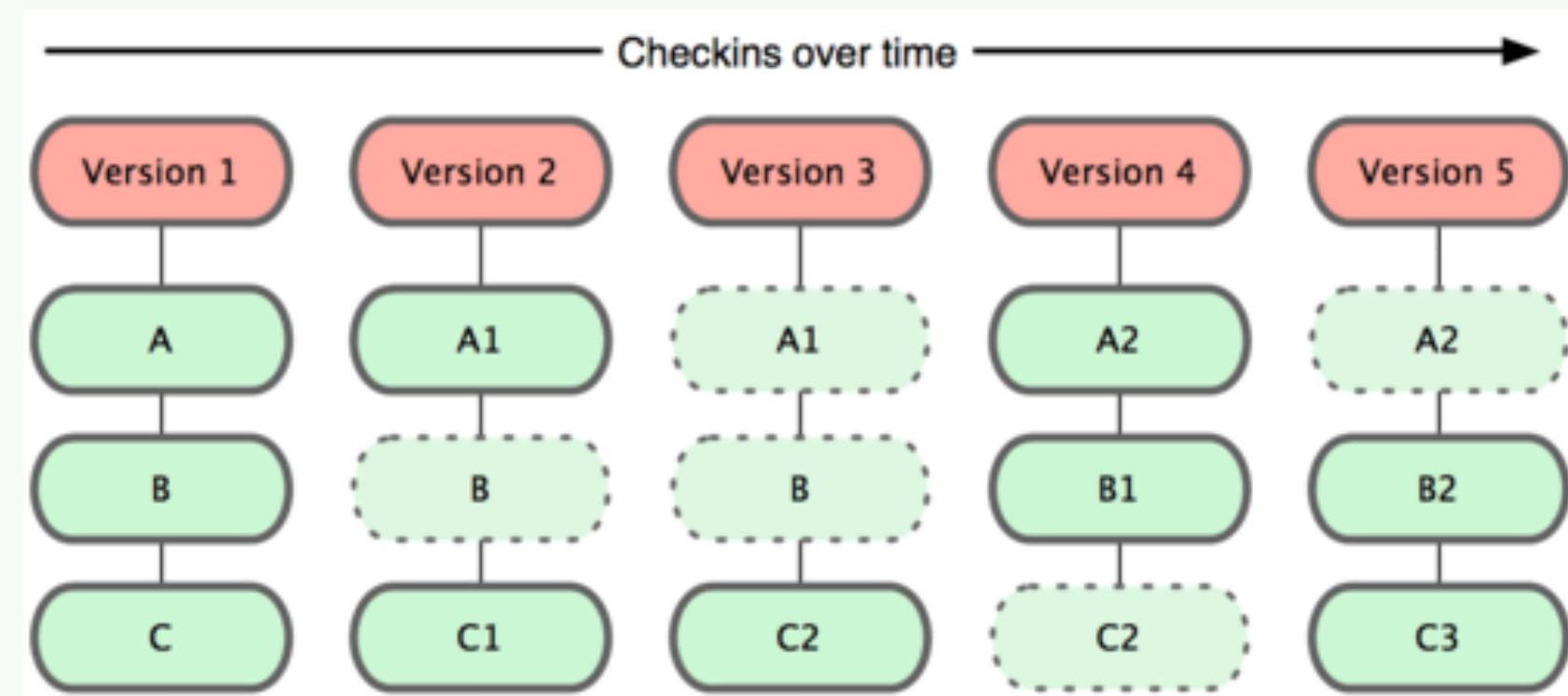
- 빠른 속도
- 단순한 구조
- 비선형적인 개발(수천 개의 동시 다발적인 브랜치)
- 완벽한 분산
- 리눅스 커널 같은 대형 프로젝트에서의 사용 편의

# Git의 특징

- 스냅샷 기반



고전적 VCS



git

- 로컬 단독 실행 지원
- 무결성 저장: 체크섬 기반의 무결성 관리
  - Git의 파일: 파일명 기반이 아닌 SHA-1 체크섬 기반의 이름으로 저장
- 삭제가 없음
  - Git에서는 삭제가 없음: 삭제하면 포인터만 사라질 뿐임
  - 프로젝트 트리를 복구하기가 용이함

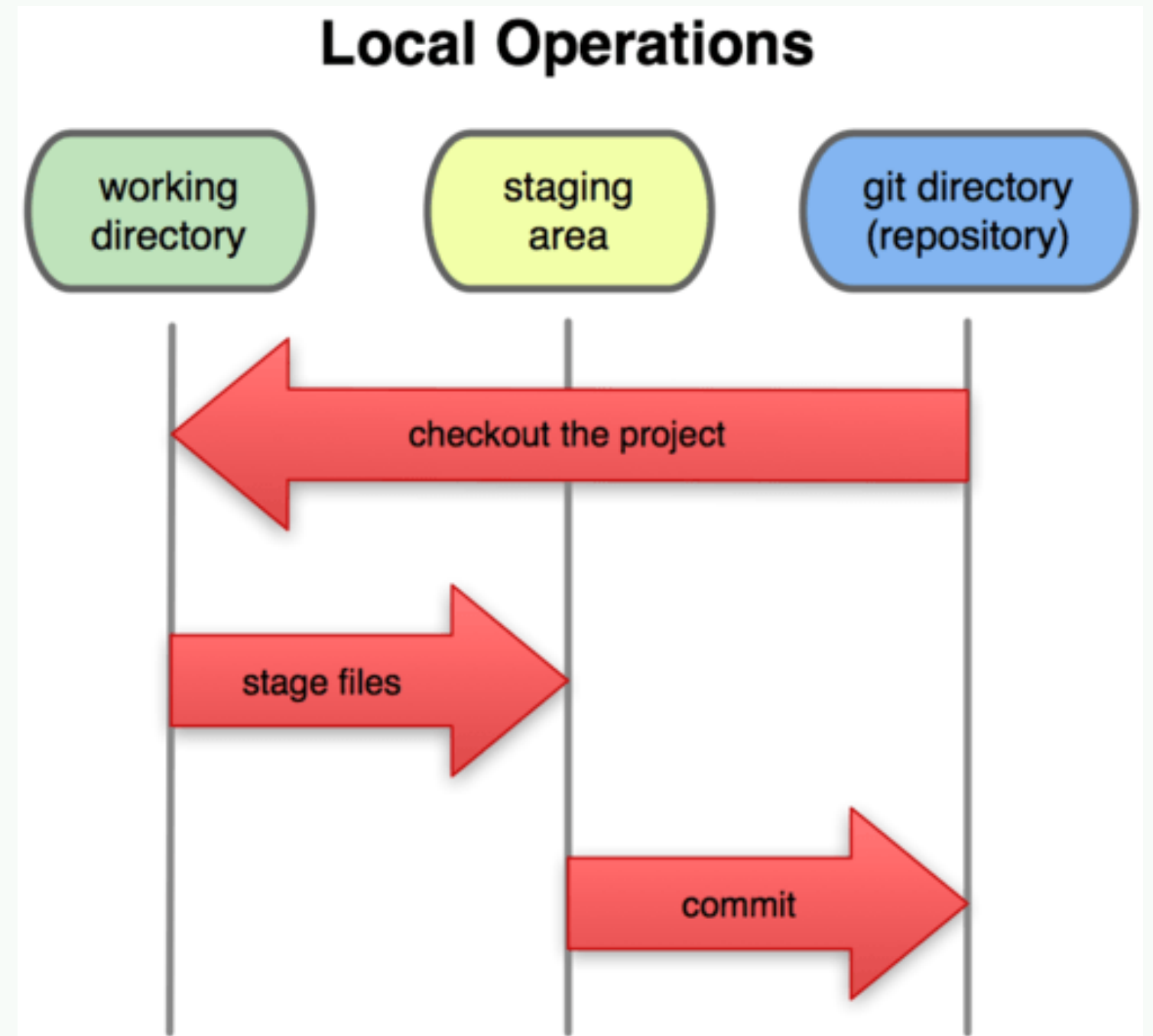
# Git 의 개념 및 구조

---

Git의 개념 및 파일 상태

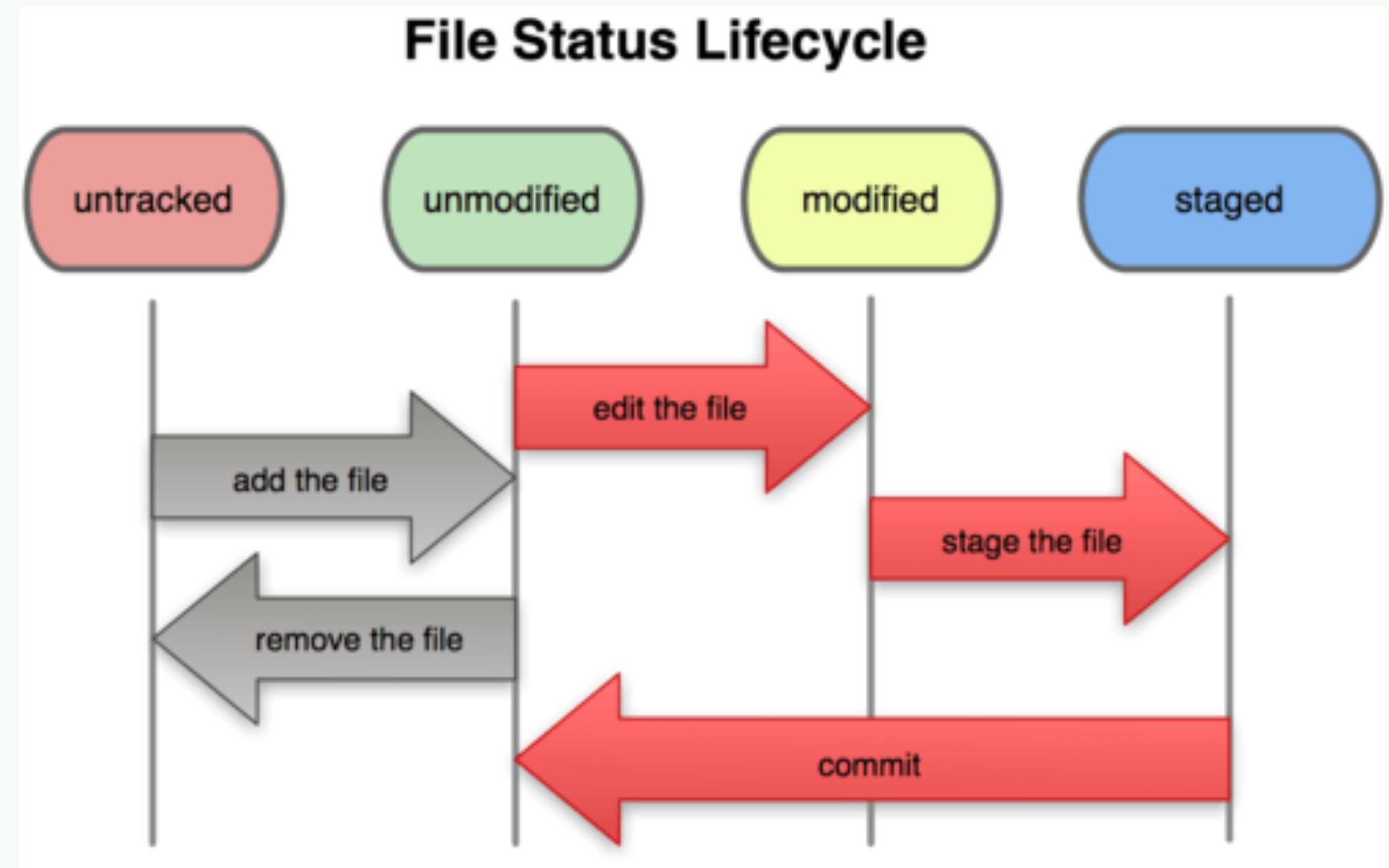
# 상태

- 세가지 상태 존재
  - Committed
    - Git에서 관리되고 있는 상태
    - 스냅샷에 포함되어 있음
  - Modified
    - 체크아웃 이후 수정한 파일들
  - Staged
    - 수정한 파일을 커밋 목록에 올린 상태



# 파일 수정시

- 추적되지 않는 파일들
  - git add로 추가할 수 있음
- 수정되지 않은 파일들
  - 따로 추적되지 않음
  - 수정하면 git add로 스테이징
- 스테이징한 파일들
  - git commit에 의해 커밋할 수 있음
- 커밋한 파일들
  - Unmodified로 표시



# 구조

- .git 디렉토리
  - Git의 실제 작업이 일어나는 공간
  - 저장소, 설정등이 모두 포함되어 있음

```
drwxr-xr-x 17 inureyes staff 578 11 1 17:52 ./
drwxr-xr-x 34 inureyes staff 1156 10 20 20:59 ../
-rw-r--r-- 1 inureyes staff 2264 7 24 10:03 COMMIT_EDITMSG
-rw-r--r-- 1 inureyes staff 2206 11 1 17:52 FETCH_HEAD
-rw-r--r-- 1 inureyes staff 23 10 20 20:59 HEAD
-rw-r--r-- 1 inureyes staff 41 10 20 20:59 ORIG_HEAD
drwxr-xr-x 2 inureyes staff 68 1 3 2014 branches/
-rw-r--r-- 1 inureyes staff 479 4 20 2015 config
-rw-r--r-- 1 inureyes staff 73 1 3 2014 description
drwxr-xr-x 11 inureyes staff 374 1 3 2014 hooks/
-rw-r--r-- 1 inureyes staff 345467 10 20 20:59 index
drwxr-xr-x 4 inureyes staff 136 3 29 2015 info/
drwxr-xr-x 4 inureyes staff 136 3 29 2015 logs/
drwxr-xr-x 260 inureyes staff 8840 5 8 2015 objects/
-rw-r--r-- 1 inureyes staff 5465 3 29 2015 packed-refs
drwxr-xr-x 5 inureyes staff 170 6 20 15:04 refs/
-rw-r--r--@ 1 inureyes staff 356 9 24 2014 sourcetreeconfig
```





# Git 응용예

---

Git을 이용한 개발 방법론

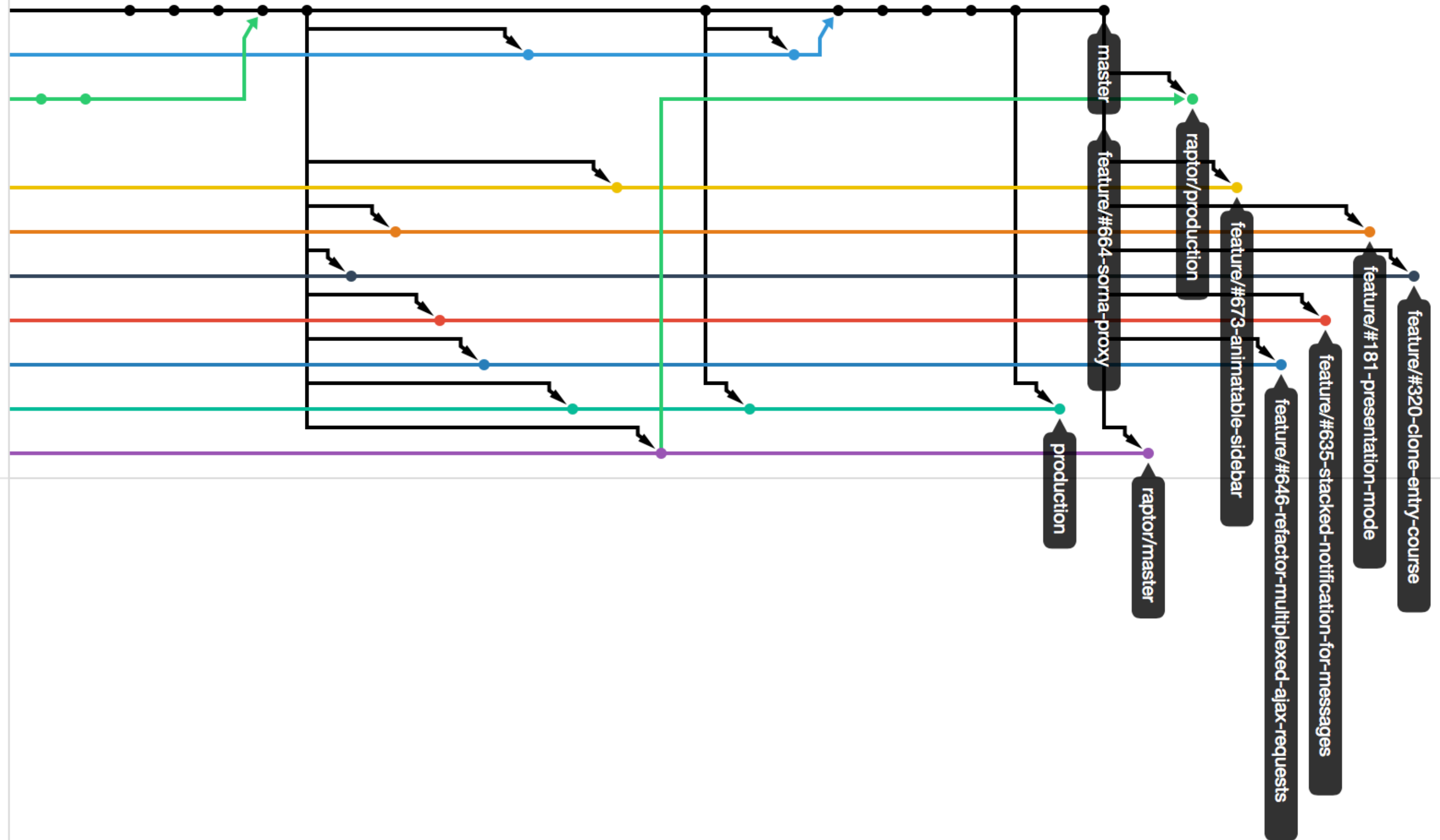
# Git flow

---

- Vincent Driessen의 브랜칭 모델을 git 명령들을 이용하여 구현
- Cheet sheet: [http://danielkummer.github.io/git-flow-cheatsheet/index.ko\\_KR.html](http://danielkummer.github.io/git-flow-cheatsheet/index.ko_KR.html)
- Feature (prefix) / hotfix (prefix) / release / master 브랜치를 사용



lablup



# Next is...

## 3/16: Basic Open-source tech stack

@inureyes

Questions? [inureyes@gmail.com](mailto:inureyes@gmail.com)

