

# Computer Network[Git]

유명성

# 0. Revision Control

---

# 0. Revision Control

---

## 0.1 버전관리

Text 파일의 수정 내역을 기록하고 싶다.



To-do.txt

# 0. Revision Control

## 0.1 버전관리

수정 사항별로 파일을 만들까?



To-do.txt



To-do\_수정.txt

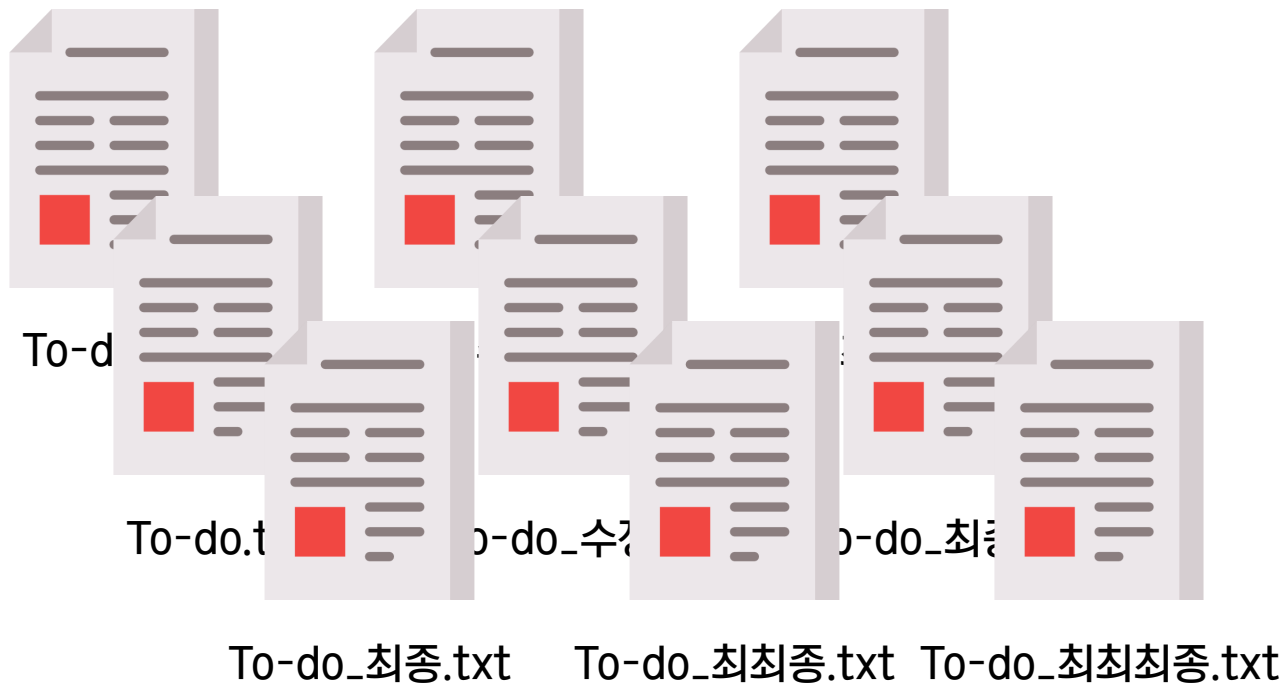


To-do\_최종.txt

# 0. Revision Control

## 0.1 버전관리

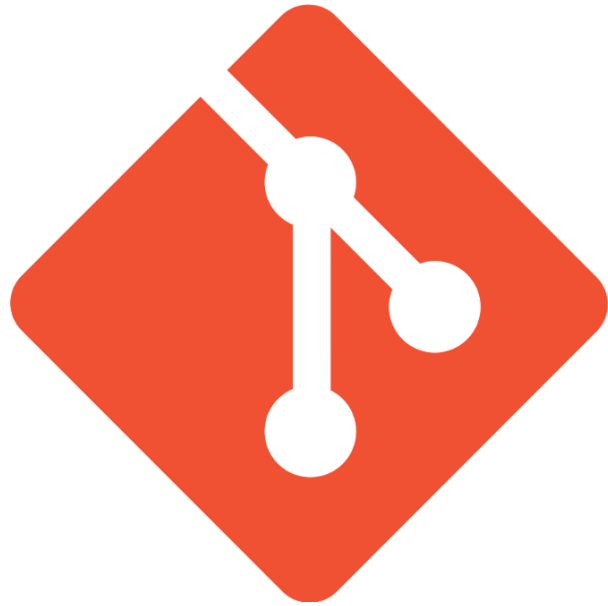
시간이 지나거나, 수정할 파일이 여러 개면?  
혹은 여러 사람이 작업한다면?



# 0. Revision Control

---

## 0.1 버전관리



**git**

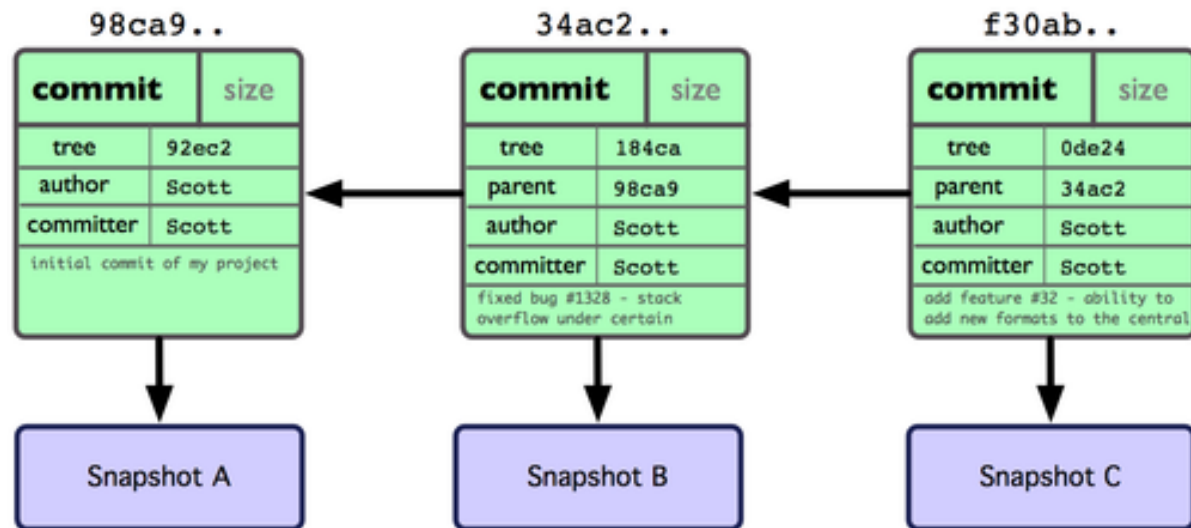
# 1. Git

---

# 1. Git

## 1.1 Commit

Commit  
[수정의 단위]





# 1. Git

## 1.1 Commit



To-do.txt

```
1 1. 에어팟 사기
2 2. 5G 공모전 서류 작성
3 3. 오토인코더 실험
```

```
1 1. 에어팟 사기
2 2. 5G 공모전 서류 작성
3 3. 오토인코더 실험
4 4. 컴퓨터네트워크 강의 준비
```

“강의 준비” 라는  
수정 발생


Commit의 단위는 사람이 결정

# 1. Git

## 1.1 Commit

Commit #3

2019. 03. 08 09:00  
토익 점수 제출 추가  
@유명성




```
1 1. 에어팟 사기
2 3. 오토인코더 실험
3 4. 컴퓨터네트워크 강의 준비
4 5. 공모전 서류 제출
5 6. 토익점수 제출
```

To-do.txt

Commit #2

2019. 03. 07 14:10  
공모전 서류 작성 삭제,  
공모전 서류 제출 추가  
@유명성




```
1 1. 에어팟 사기
2 3. 오토인코더 실험
3 4. 컴퓨터네트워크 강의 준비
4 5. 공모전 서류 제출
```

To-do.txt

Commit #1

2019. 03. 06 12:00  
강의 준비 추가  
@유명성



```
1 1. 에어팟 사기
2 2. 5G 공모전 서류 작성
3 3. 오토인코더 실험
4 4. 컴퓨터네트워크 강의 준비
```

To-do.txt

# 1. Git

## 1.1 Commit

파일의 추가/삭제도 commit 내용에 포함될 수 있다.

Commit #3



2019. 03. 08 09:00  
여가부 공모전 서류 추가,  
To-do.txt 삭제  
@유명성



여가부.doc



공모전.excel



SCTF.excel

Commit #2

2019. 03. 07 14:10  
SCTF 준비계획 추가  
@유명성



To-do.txt



공모전.excel



SCTF.excel

Commit #1

2019. 03. 06 12:00  
2019 공모전 일정표 추가  
@유명성



To-do.txt

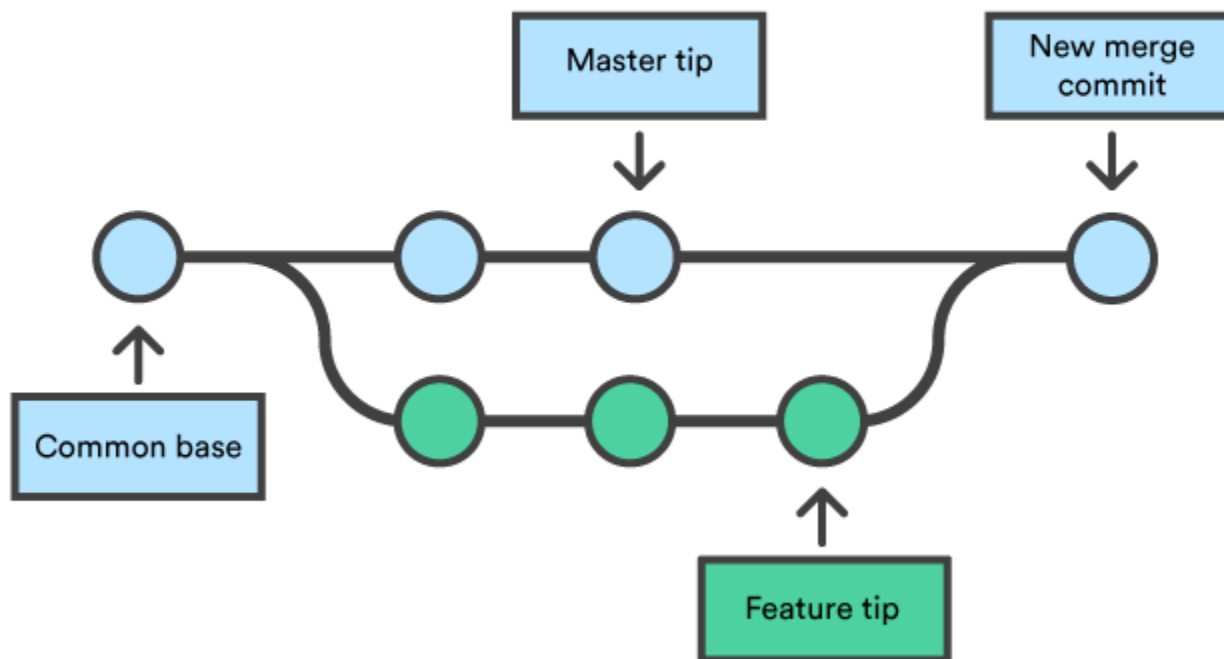


공모전.excel

# 1. Git

## 1.2 Branch

Branch  
[독립적인 작업흐름]



# 1. Git

---

## 1.2 Branch

# Init

[commit 단위 추적 시작]

HEAD

브랜치를 가리키는 포인터



Master

Init으로 생성되는 기본 브랜치

# 1. Git

## 1.2 Branch



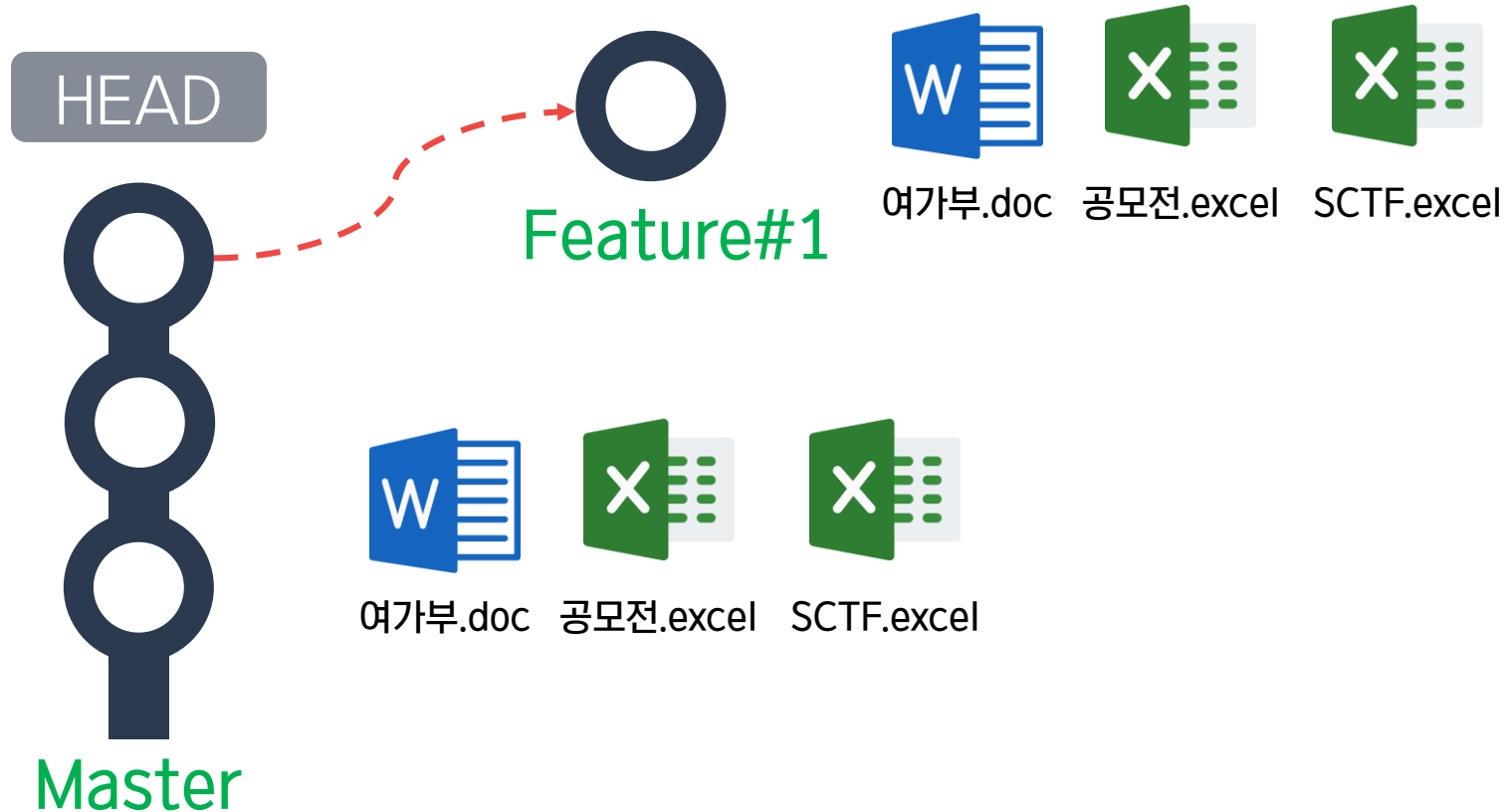
Commit은 독립적인  
각각의 브랜치 공간에 기록

# 1. Git

## 1.2 Branch

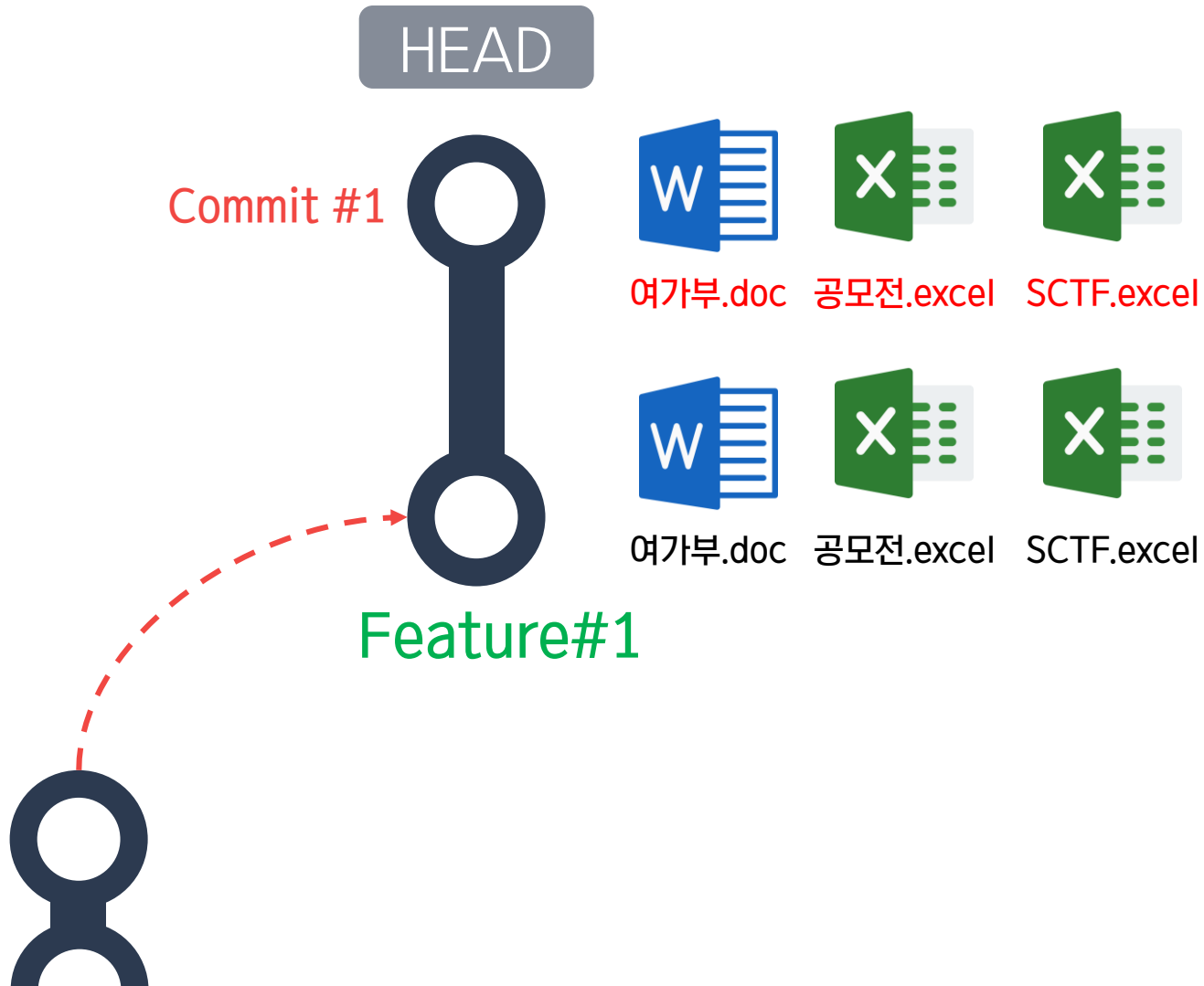
### Branch Feature#1

새로운 흐름 생성



# 1. Git

## 1.2 Branch



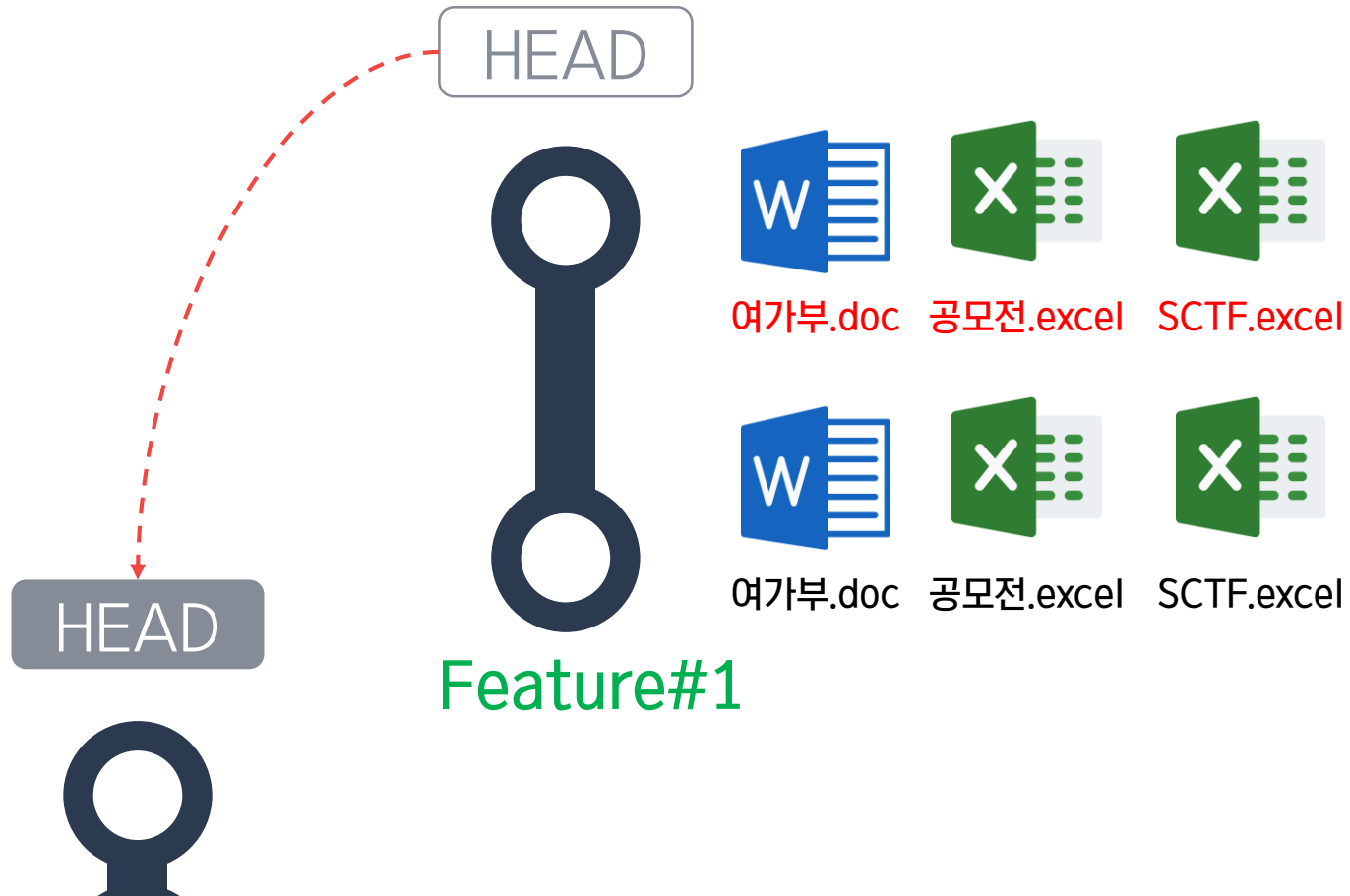


# 1. Git

## 1.3 Checkout

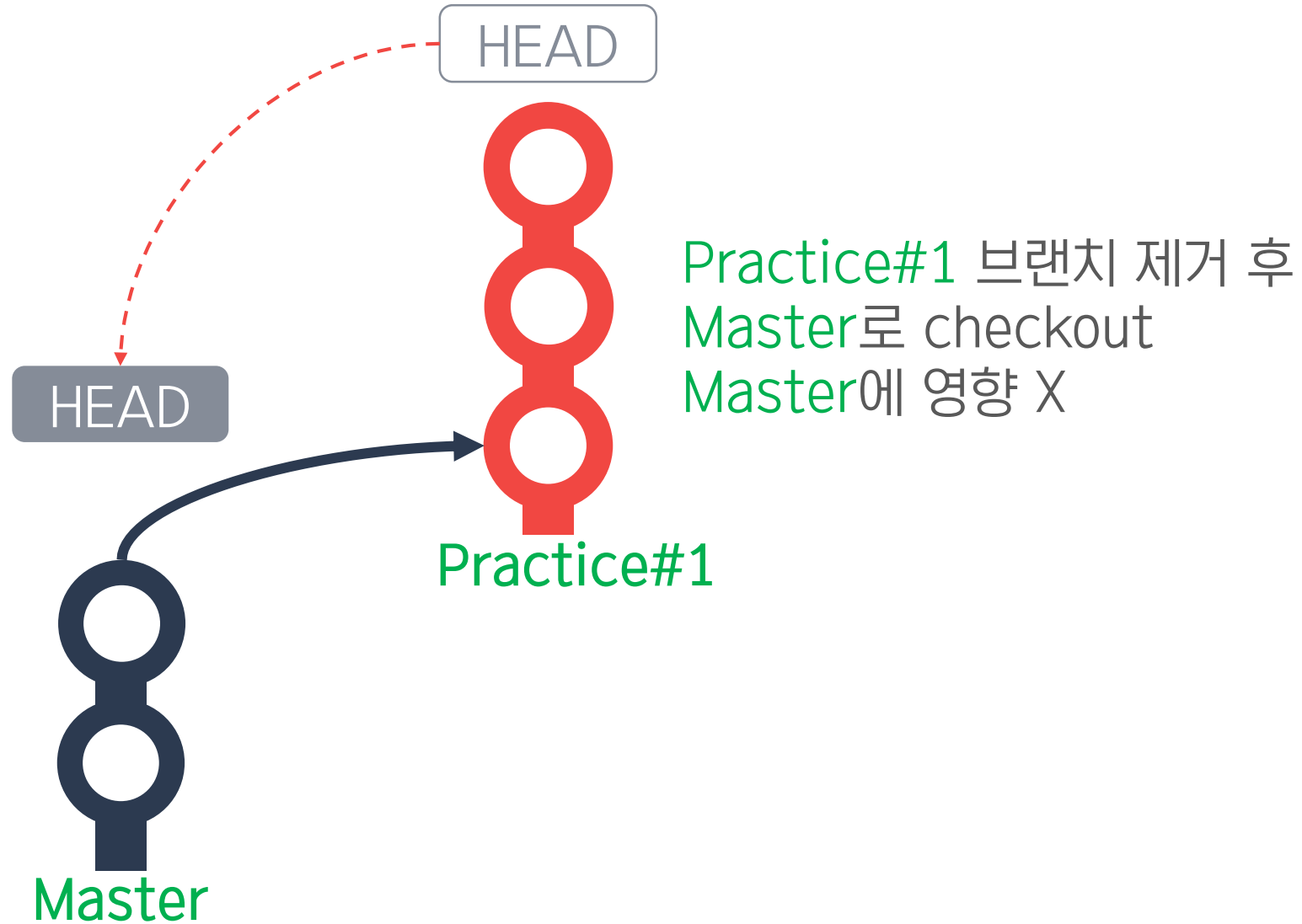
### Checkout Master

브랜치간 이동



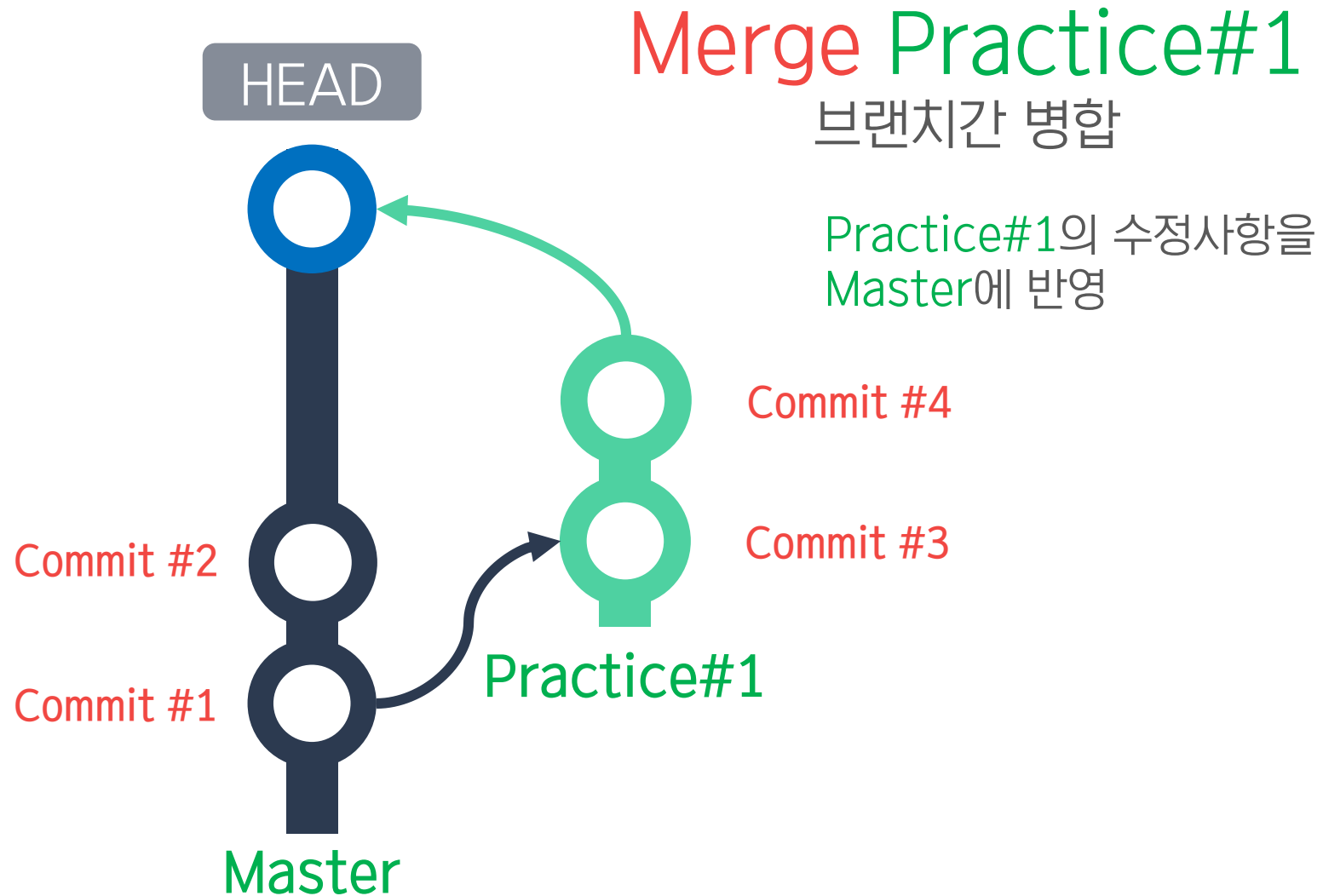
# 1. Git

## 1.3 Checkout



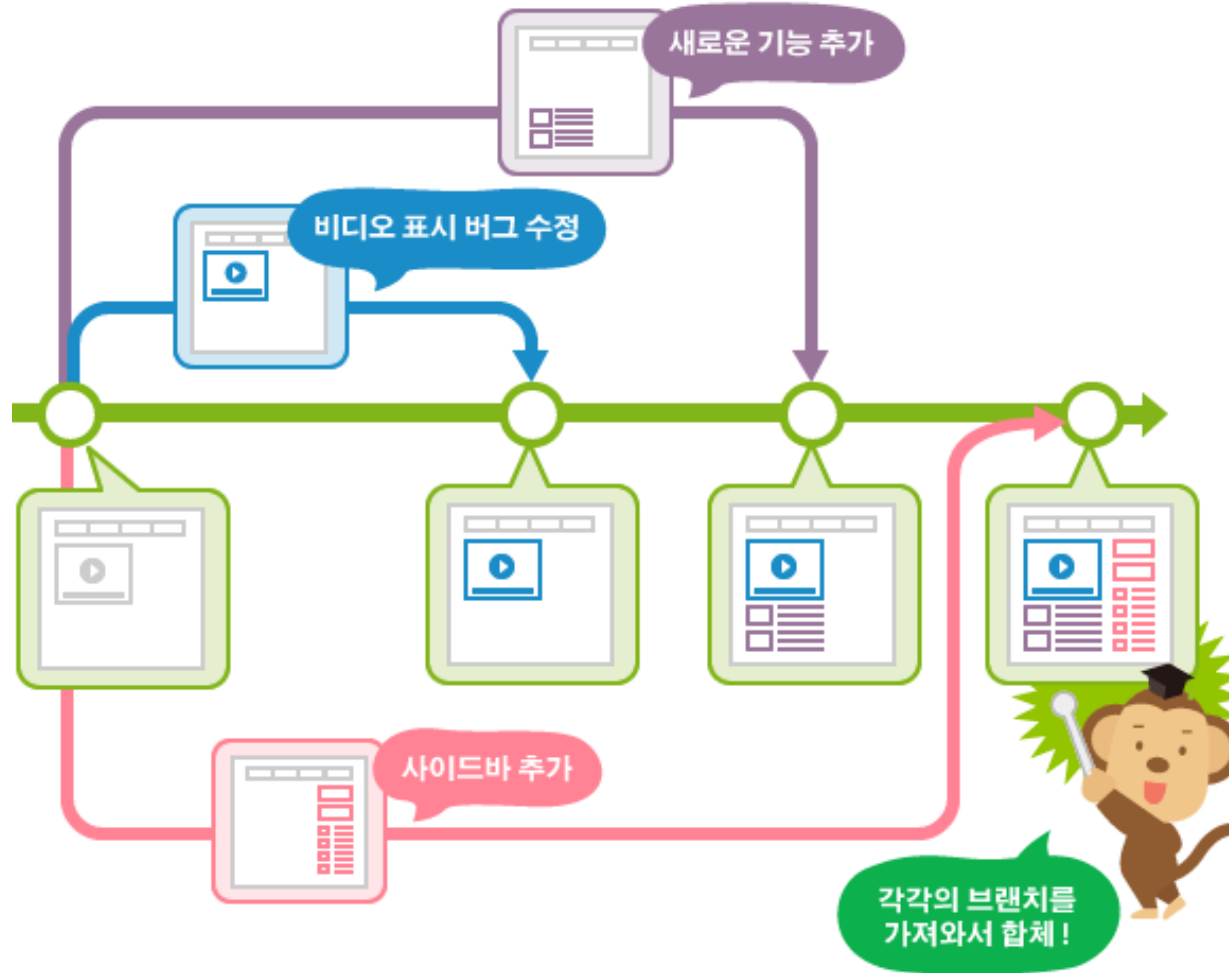
# 1. Git

## 1.4 Merge



# 1. Git

## 1.4 Merge



# 1. Git

---

## 1.5 Summary

**Commit** 의미를 갖는 수정내용을 Snapshot에 기록한다.

**Branch** 서로 완전히 독립된 작업공간을 생성한다.

**Checkout** 원하는 브랜치로 혹은 커밋으로 자유롭게 이동  
HEAD 포인터가 변경된다.

**Merge** 서로 다른 브랜치의 내용을 하나로 병합한다.  
충돌(Conflict)가 발생할 수 있다. -> 사람이 해결해야 한다.

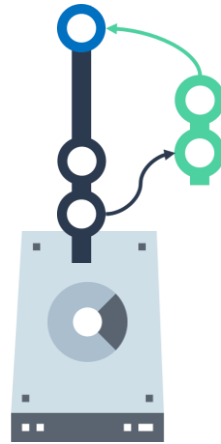
## 2. Remote Repository

---

## 2. Remote Repository

### 2.1 Collaborate with others

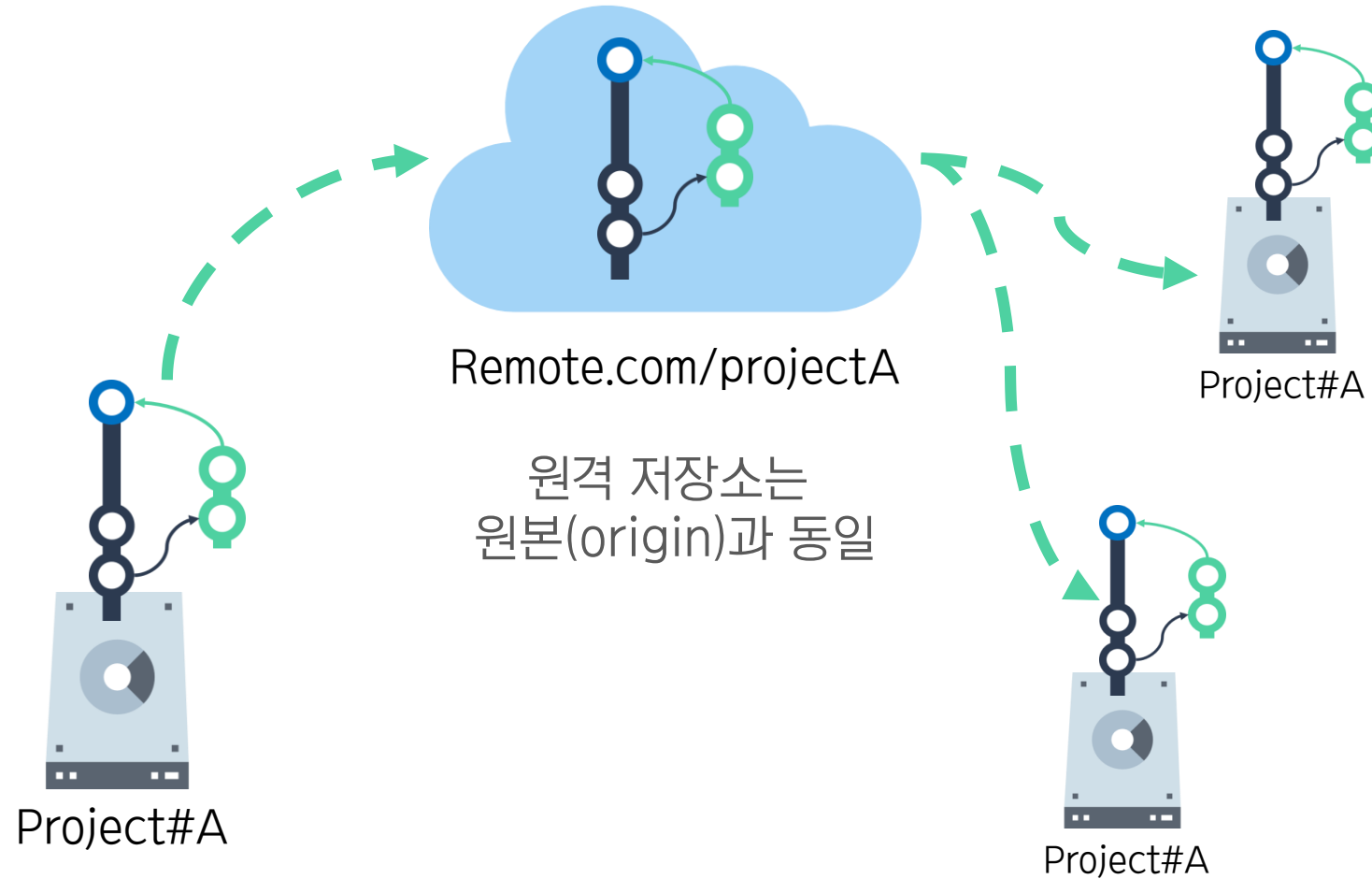
여러 사람들과 공동으로 작업하려면??



Local  
repository

## 2. Remote Repository

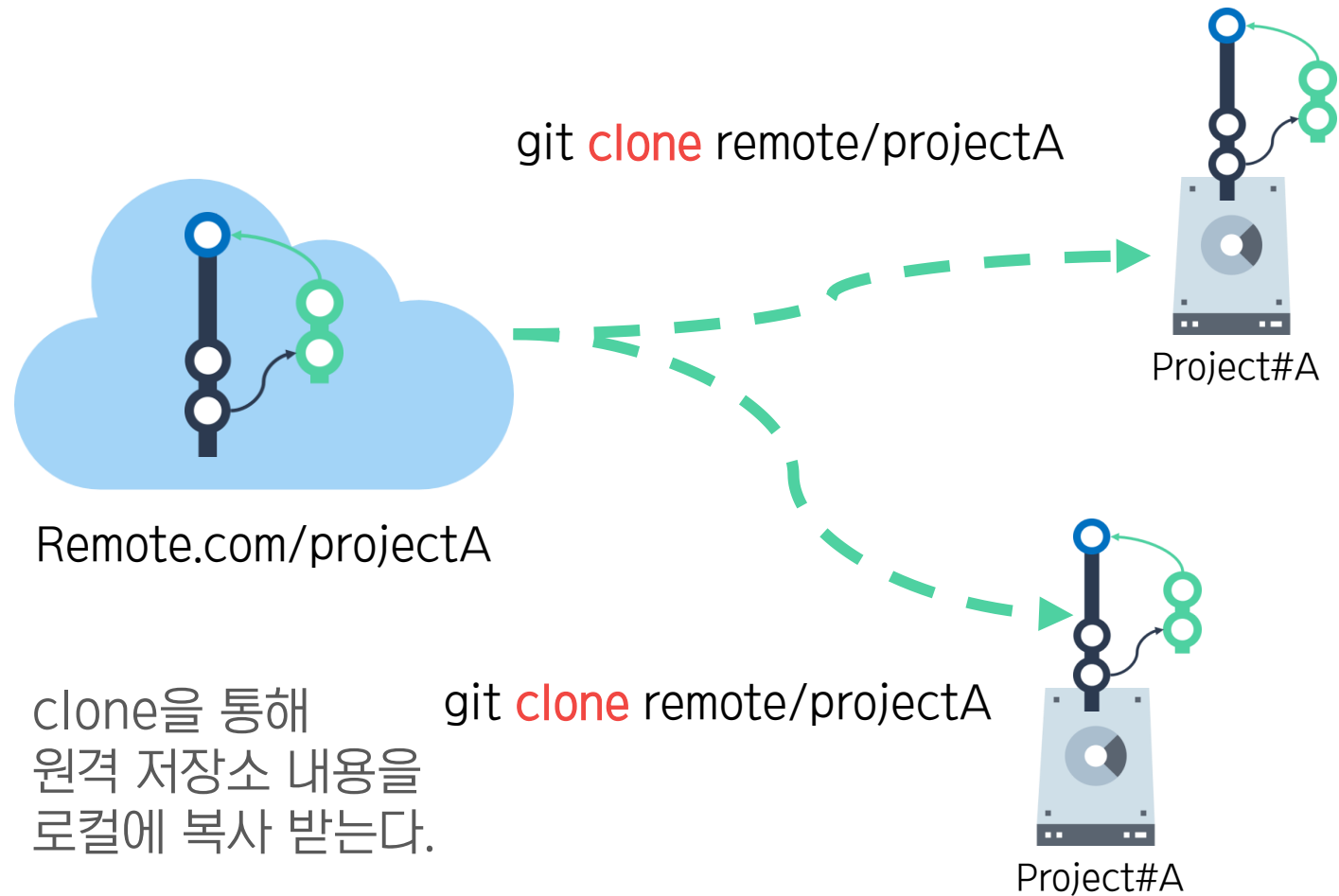
### 2.2 Remote Repository





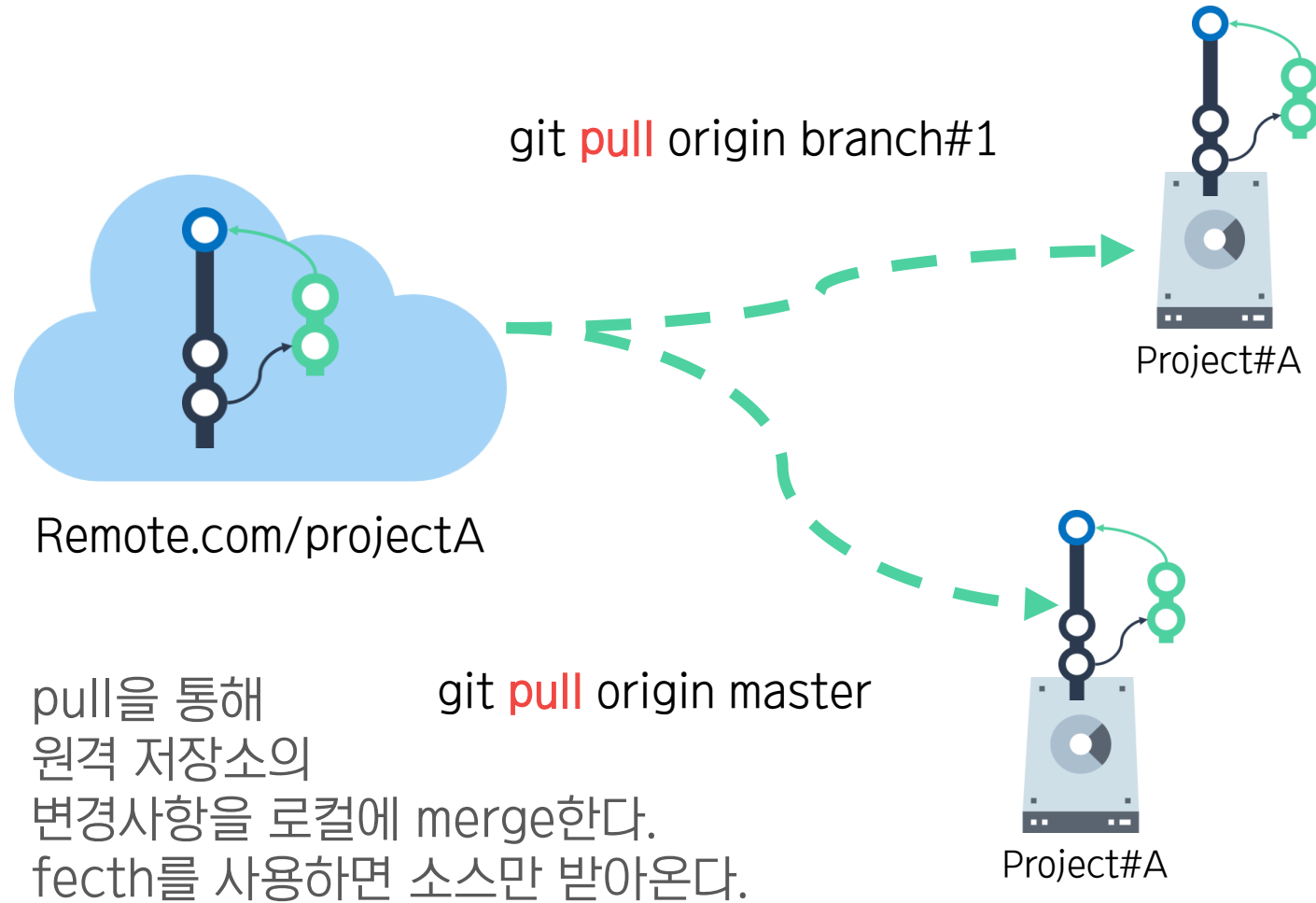
## 2. Remote Repository

### 2.3 Git Clone



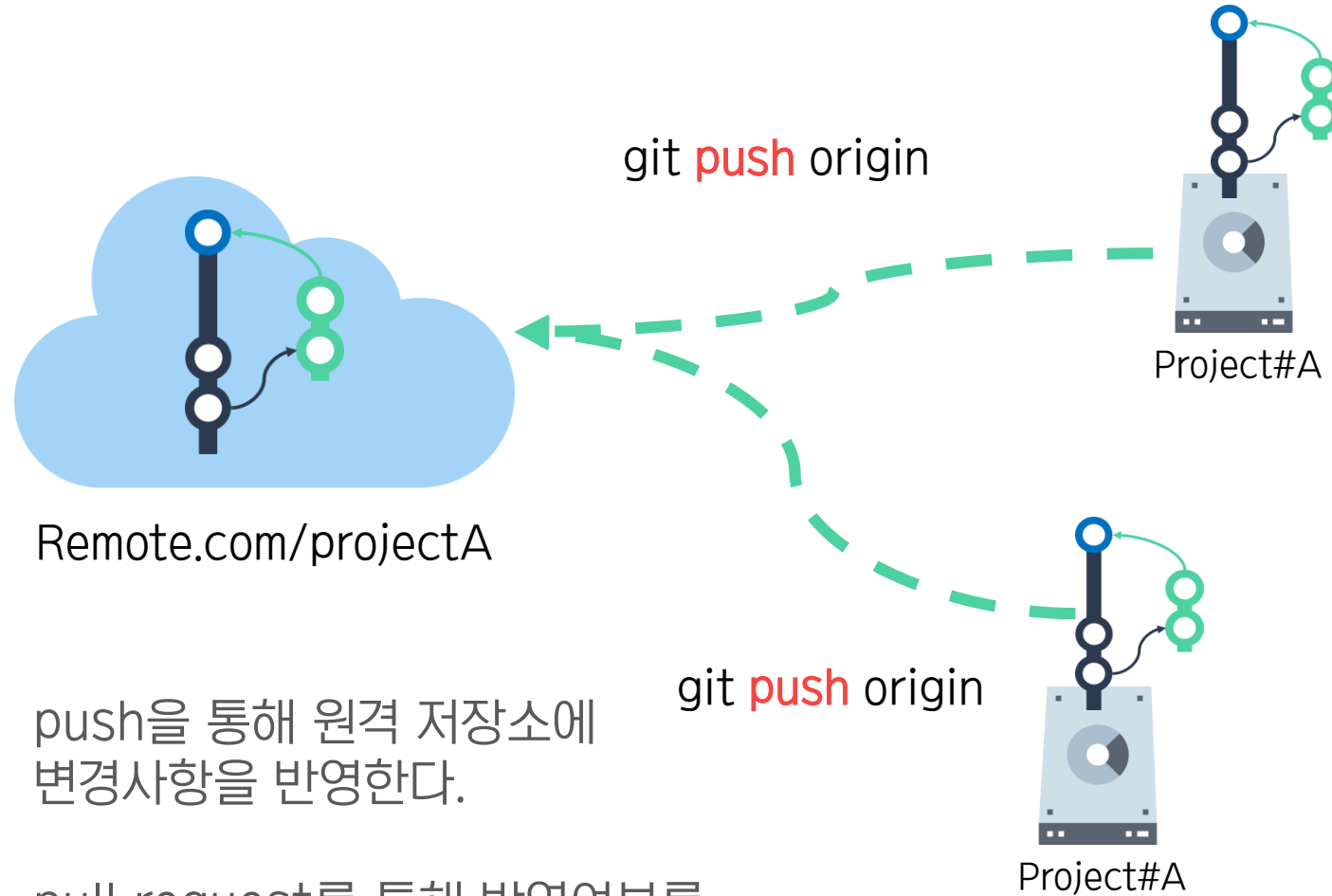
## 2. Remote Repository

### 2.4 Git Pull



## 2. Remote Repository

### 2.4 Git Push



Remote.com/projectA

Project#A

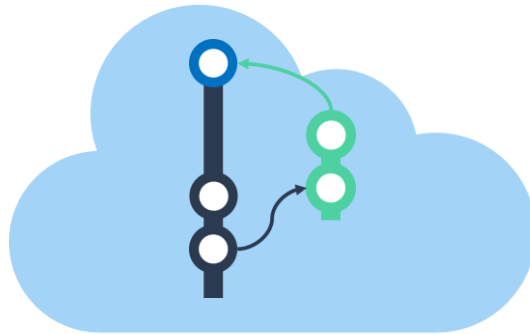
Project#A

push을 통해 원격 저장소에  
변경사항을 반영한다.

pull request를 통해 반영여부를  
다른 사람이 결정할 수 있다.

## 2. Remote Repository

### 2.5 Github

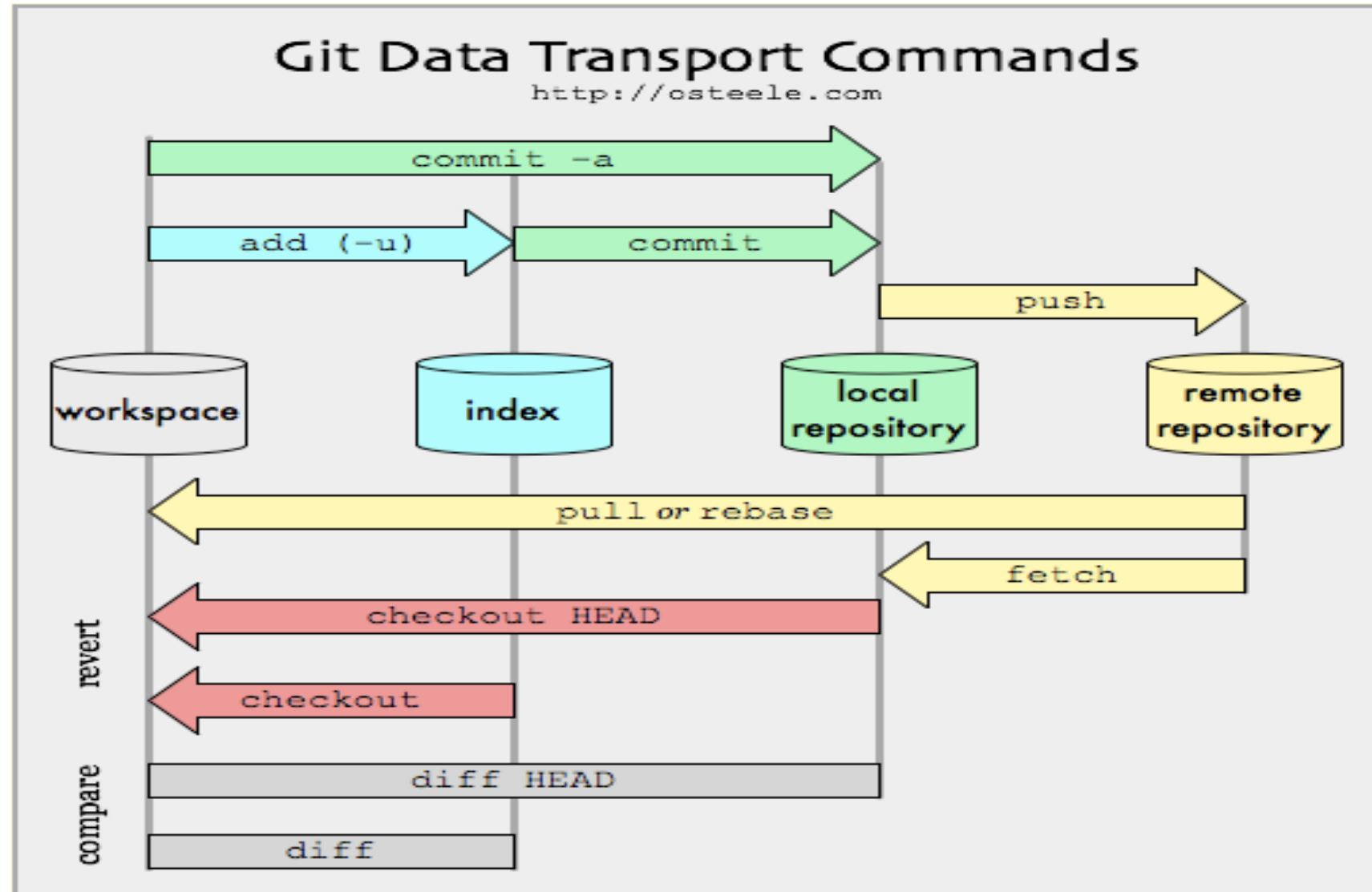


Remote.com/projectA



## 2. Remote Repository

### 2.6 Summary



# Q & A

---