

git reset

- [Git reset & three trees of Git](#)
- [Undo a change with git reset](#)
- [Undo several changes with git reset](#)
- [Pushing resets to Remote](#)
- [Undo a git reset](#)

git reset 명령은 변경 사항을 실행 취소하기 위한 복잡하고 다양한 도구입니다. 세 가지 기본 형식의 호출이 있습니다. 이러한 형식은 명령 줄 인수 --soft, --mixed, --hard에 해당합니다. 세 개의 인수는 각각 Git의 세 가지 내부 상태 관리 메커니즘인 HEAD (The Commit Tree), 스테이징 색인 및 작업 디렉토리에 해당합니다.

Git reset & three trees of Git

git reset 사용법을 제대로 이해하려면 Git의 내부 상태 관리 시스템을 먼저 이해해야 합니다. 때로는 이러한 메커니즘을 Git의 "three trees"라고 부릅니다. 나무는 엄격하게 전통적인 트리 데이터 구조가 아니므로 잘못된 이름일 수 있습니다. 그러나 Git은 편집 타임 라인을 추적하는 데 사용하는 데 사용하는 노드 및 포인터 기반 데이터 구조입니다. 이러한 메커니즘을 증명하는 가장 좋은 방법은 리포지토리에 변경 집합을 만들어 세 개의 트리를 따라 이동하는 것입니다.

Undo a change with git reset

시작하려면 역사에서 가장 최근 커밋을 실행 취소하십시오. 이 경우 Bitbucket의 CI / CD 솔루션 파이프 라인을 활성화했지만 스크립트가 울바르지 않다는 것을 알았습니다.

1. 터미널 창에 git log --oneline을 입력하십시오.
2. 두 번째 커밋의 커밋 해시를 로그에 복사합니다 (52f823c). 그런 다음 q를 눌러 로그를 종료합니다.
3. git reset --soft 52f823c를 터미널 창에 입력하십시오. 명령이 성공하면 명령이 백그라운드에서 실행됩니다. 로그 전부입니다. 당신은 당신의 첫 번째 변화를 취소했습니다. 이제 작업의 결과를 보도록 하겠습니다.
4. 터미널 창에 git status를 입력하면 커밋이 취소되었고 커밋되지 않은 변경 사항이 표시됩니다. 다음과 같이 보일 것입니다 :

```
$ git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

       new file:   bitbucket-pipelines.yml
```

5. Enter git log --oneline in your terminal window. You should see something like this:

```
$ git log --oneline
52f823c repeated quote to show how a change moves through the process
4801b87 Merged in changes (pull request #6)
1a6a403 myquote edited online with Bitbucket
3b29606 (origin/changes) myquote2.html edited online with Bitbucket
8b236d9 myquote edited online with Bitbucket
235b9a7 testing prs
c5826da more changes
43a87f4 remivng
d5c4c62 a few small changes
23a7476 Merged in new-feature2 (pull request #3)
5cc4e1e add a commit message
cbbb5d6 trying a thing
438f956 adding section for permissions and cleaning up some formatting
23251c1 updated snipptes.xml organization into resources. other files misc changes
3f630f8 Adding file to track changes
...
```

6. 분기의 새로운 HEAD가 사용자가 원하는대로 커밋 52f823c임을 알 수 있습니다.
7. q를 눌러 로그를 종료하십시오. 간단한 재설정을 수행하는 방법을 배웠으니 이제 좀 더 복잡한 것을 시도해보십시오. 터미널을 열어 두십시오.

Git 가이드

1. Git 시작하기

- [Git 저장소](#)
 - [git init](#)
 - [git clone](#)
 - [git config](#)
- [변경 저장하기](#)
 - [git add](#)
 - [git commit](#)
 - [git diff](#)
 - [git stash](#)
 - [.gitignore](#)
- [저장소 점검하기](#)
 - [git status](#)
 - [git log](#)
 - [git tag](#)
 - [git blame](#)
- [변경 취소하기](#)
 - [git 실행 취소](#)
 - [git clean](#)
 - [git revert](#)
 - [git reset](#)
 - [git rm](#)
- [Rewriting history](#)
 - [git commit --amend](#)
 - [git rebase](#)
 - [git reflog](#)

2. Git 협업하기

- [동기화하기](#)
 - [git remote](#)
 - [git fetch](#)
 - [git pull](#)
 - [git push](#)
- [브랜치 사용하기](#)
 - [git branch](#)
 - [git checkout](#)
 - [git merge](#)
 - [병합 충돌 해결하기 \(Merge conflicts\)](#)
 - [병합 전략 \(Merge strategies\)](#)
- [Pull request 만들기](#)

Undo several changes with git reset

당겨 받기 요청 # 6 (4801b87)을 다시 작성해야하고 git reset 명령을 사용할 때 HEAD가 1a6a403을 커밋하도록 재설정해야한다는 사실을 깨닫게되었다고 가정 해 보겠습니다.

1. git log --online을 입력하십시오.
2. 실행 취소하려는 변경 사항이있는 끌어 오기 요청 # 6 바로 아래 커밋 인 커밋 해시 1a6a403 (Bitbucket을 사용하여 온라인으로 편집 된 myquote)을 복사합니다.
3. 터미널 창에 git reset 1a6a403을 입력하십시오. 출력은 다음과 같이 보일 것입니다 :

```
$ git reset 1a6a403
Unstaged changes after reset:
M README.md
M myquote2.html
```

변경 사항이 현재 커밋되지 않은 상태에 있음을 알 수 있습니다. 이는 이제 프로젝트 기록과 준비 영역에서 몇 가지 변경 사항을 제거했음을 의미합니다.

4. Enter git status in your terminal window. The output should look something like this:

```
$ git status
On branch master
Your branch is behind 'origin/master' by 6 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

       modified:   README.md
       modified:   myquote2.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)

       bitbucket-pipelines.yml

no changes added to commit (use "git add" and/or "git commit -a")
```

5. 이제 우리가 undid (bitbucket-pipelines.yml 파일)의 첫 번째 변경 사항이 git에서 완전히 추적 할 수 없음을 알 수 있습니다. git reset을 호출하면 브랜치의 헤드와 git의 추적 영역 또는 인덱스 영역 모두에서 변경 사항이 제거되기 때문입니다. 기본 프로세스는 여기에서 다를 수 있는 것보다 조금 복잡합니다. git reset에서 더 많은 것을 읽을 수 있습니다.
6. Enter git log --online in your terminal window.

```

1a6a403 myquote edited online with Bitbucket
8b236d9 myquote edited online with Bitbucket
43a87f4 remivng
d5c4c62 a few small changes
23a7476 Merged in new-feature2 (pull request #3)
5cc4e1e add a commit message
cbbb5d6 trying a thing
438f956 adding section for permissions and cleaning up some
formatting
23251c1 updated snipptes.xml organization into resources. other
files misc changes
3f630f8 Adding file to track changes
e52470d README.md edited online with Bitbucket
e2fad94 README.md edited online with Bitbucket
592f84f Merge branch 'master' into new-feature2 Merge branch
especially if it merges an updated upstream into a topic branch.
7d0bab8 added a line
879f965 adding to the quote file
8994332 Merged in HOT-235 (pull request #2)
b4a0b43 removed sarcastic remarks because they violate policy.
b5f5199 myquote2.html created online with Bitbucket
b851618 adding my first file
5b43509 writing and using tests

```

로그 출력은 커밋 내역이 수정되어 커밋 1a6a403에서 시작됨을 보여줍니다. 데모 및 추가 예제를 위해 방금 수행 한 재설정을 실행 취소하려고한다고 가정 해 봅시다. 추가 고려 사항을 고려한 후에 풀 요청 # 6의 내용을 유지하려고했습니다.

Pushing resets to Remote

Git reset은 변경 을 취소하는 방법 중 하나입니다. git reset은 일반적으로 변경 사항을 취소하기위한 '안전하지 않은'오프셋으로 간주됩니다. 격리 된 코드에서 로컬로 작업 할 때 재설정은 좋지 만 팀 구성원과 공유 할 경우 위험 해집니다.

원격 팀으로 리셋 된 지점을 공유하려면 '강제 푸시'가 실행되어야합니다. 'forced push'는 git push -f를 실행하여 시작됩니다. 강제 푸시는 푸시 시점 이후에 만들어진 branch의 모든 기록을 삭제합니다.

'안전하지 않은'시나리오의 예는 다음과 같습니다.

- Dev A는 새로운 기능을 개발하는 브랜치에서 작업하고 있습니다.
- Dev B는 별도의 기능을 개발하는 동일한 브랜치에서 작업하고 있습니다.
- Dev B는 Dev A와 Dev B가 모두 작동하기 전에 브랜치를 이전 상태로 재설정하기로 결정합니다.
- Dev B 그러면 강제로 재설정 브랜치를 원격 저장소로 푸시합니다.
- Dev A는 분기를 가져 와서 업데이트를받습니다. git pull 동안 Dev A는 강제 업데이트를받습니다. 이것은 기능 작업이 완료되고 커밋을 오픈기 전에 Dev A의 로컬 브랜치를 원래대로 되돌립니다.

Undo a git reset

지금까지 우리는 git commit에 Sha 해쉬를 보내고있다. 이제 git log 출력에 재설정 된 커밋이 없습니다. 어떻게 그 커밋을 되 찾을 수 있습니까? Git은 커밋을 지우지 않습니다. 또한 git은 "reflog"라고 불리는 모든 ref 동작의 개별 로그를 저장합니다. git reflog를 실행하여 reflog를 검사 할 수 있습니다.

```

1a6a403 HEAD@{0}: reset: moving to 1a6a403
1f08a70 HEAD@{1}: reset: moving to origin/master
1f08a70 HEAD@{2}: clone: from git@bitbucket.org:dans9190/tutorial-
documentation-tests.git

```

git reflog의 결과는 위와 비슷해야합니다. 레포에서 수행 한 내역을 볼 수 있습니다. 맨 위 줄은 풀 요청 # 6을 재설정하기 위해 수행 한 재설정에 대한 참조입니다. 이제 풀 리퀘스트 # 6을 리셋하기 위해 리셋을 리셋하자. 이 reflog 출력의 두 번째 열은 repo에서 수행하는 수정 작업에 대한 참조 포인터를 나타냅니다. 여기서 HEAD @ {0}은 이전에 실행 한 재설정 명령에 대한 참조입니다. 리셋 명령을 회신하지 않으려면 HEAD @ {1} (으)로 리포를 복원하십시오.

```

$ git reset --hard HEAD@{1}
HEAD is now at 1f08a70 Initial Bitbucket Pipelines configuration

```

이제 git log --oneline을 사용하여 repos commit history를 살펴 보겠습니다:

```
$git log --online
1f08a70 Initial Bitbucket Pipelines configuration
52f823c repeated quote to show how a change moves through the process
4801b87 Merged in changes (pull request #6)
1a6a403 myquote edited online with Bitbucket
3b29606 myquote2.html edited online with Bitbucket
8b236d9 myquote edited online with Bitbucket
235b9a7 testing prs
c5826da more changes
43a87f4 remivng
d5c4c62 a few small changes
23a7476 Merged in new-feature2 (pull request #3)
5cc4e1e add a commit message
cbbb5d6 trying a thing
438f956 adding section for permissions and cleaning up some formatting
23251c1 updated snipptes.xml organization into resources. other files misc
changes
3f630f8 Adding file to track changes
e52470d README.md edited online with Bitbucket
e2fad94 README.md edited online with Bitbucket
592f84f Merge branch 'master' into new-feature2 Merge branch  especially
if it merges an updated upstream into a topic branch.
7d0bab8 added a line
:
```

여기서 우리는 repo의 커밋 내역이 이전 버전으로 복원되었음을 알 수 있습니다. 우리는 4801b87이 첫 번째 재 설정 작업에서 손실 된 것처럼 보였다고하더라도이를 복구 할 수 있음을 알 수 있습니다. git reflog는 저장소의 변경 사항을 취소하기위한 강력한 도구입니다. 자세한 내용은 git reflog 페이지에서 자세히 알아보십시오.