

Recover from an Accidental `git push --force` (README or Other Files)

If you've accidentally force-pushed an update (e.g., to `README.md`) and want to **reverse it**, don't panic—there are a few ways to recover the lost commit or content, depending on your setup and timing.

Quick Summary

You may be able to **recover the previous commit** in one of the following ways:

- Use `git reflog` (if you're on the same machine where the force push was done).
 - Find the previous commit on GitHub (if it's still listed).
 - Ask a collaborator for a local copy (if someone pulled it before the force push).
 - If all else fails, manually re-create the file(s).
-

Option 1: Use `git reflog` (Best Option)

If you force-pushed from your local machine, you can use Git's local history to recover the previous commit.

Steps:

1. Open your terminal in the repo directory.
2. Run:

```
git reflog
```

This shows a list of recent actions in your Git repository, including commits, resets, rebases, and force-pushes.

3. Look for the entry just before the force push. It may look like:

```
abc1234 HEAD@{1}: push: forced update
```

4. Once you've identified the correct commit hash (e.g., `abc1234`), reset to it:

```
git reset --hard abc1234
```

5. Push it back to GitHub (if needed):

```
git push origin main --force
```

Tip: You can also create a branch from the old commit if you're unsure:

```
git checkout -b recover-old-readme abc1234
```

Option 2: Recover via GitHub UI

GitHub might still have a reference to the previous commit, especially if someone forked or cloned the repository before the force push.

Steps:

1. Go to your repo on GitHub.
2. Click on the **"Commits"** button near the top of the repository (under the "Code" tab).
3. Look for the previous commit before the force push.
4. Click on it to:
 - Browse files and copy content manually.
 - Or click **Revert** (if available) to undo that commit via a new commit.

Option 3: Ask Collaborators

If others are working on the same repo and pulled it **before** the force push:

- Ask them to run:

```
git log
```

and look for the missing commit.

- They can create a patch or branch from it and share it with you.

In our class, the GAs likely pulled your repo. Ask them right away.

What if It's Gone?

If the commit is no longer available in:

- `git reflog`
- GitHub's commit history
- Any collaborator's local repo

...then it is most likely lost permanently (Git will eventually garbage-collect unreachable commits).

In that case:

- Manually recreate the file(s), if possible.
- Use backups, previous PRs, or documentation to restore the lost content.

How to Check If the Commit Is Still Available

Check `reflog`:

```
git reflog
```

Look for any entry labeled as:

```
HEAD@{n}: push: forced update
```

Check GitHub:

- Go to <https://github.com/your-username/your-repo/commits/main>
- See if the commit is listed.
- If you know the commit hash, try going to:

```
https://github.com/your-username/your-repo/commit/<commit-hash>
```

If the page is 404, GitHub no longer has a reference to it.

Final Tips

- Use `git push --force-with-lease` instead of `--force` to avoid unintentional overwrites.
- Consider protecting the `main` branch to prevent force pushes unless explicitly allowed.
- Always check `git status` and `git log` before pushing.

Questions? Share your `git reflog` output or commit hashes to get help from a teammate.