

## 2. Source control strategy

### └ 2.2 Repository Configuration and Maintenance

#### └ 2.2.4 Recover/remove data with Git commands

---

1. How do you recover a deleted branch in Git?
  2. How can you recover a deleted commit?
  3. What command is used to undo local changes in a working directory?
  4. How do you permanently remove a file from a Git repository's history?
  5. How do you use git revert and git reset to undo changes?
  6. How can you recover a file deleted in a previous commit?
  7. What is the difference between git reflog and git log for recovery?
  8. How do you use git cherry-pick to restore changes from another branch?
  9. How do you remove sensitive data (e.g., passwords) from commit history?
  10. How can you prevent accidental data loss when cleaning up branches or commits?
- 

#### 1. How do you recover a deleted branch in Git?

Find the branch's last commit hash with git reflog then recreate it:

```
git checkout -b branch_name <commit_hash>
```

---

#### 2. How can you recover a deleted commit?

Use git reflog to locate the commit hash, then checkout or branch from it:

```
git checkout <commit_hash> OR git branch <new_branch> <commit_hash>
```

---

#### 3. What command is used to undo local changes in a working directory?

git checkout -- <file> (undo changes in file) or

git restore <file> (recommended in newer Git).

---

#### 4. How do you permanently remove a file from a Git repository's history?

Use

- git filter-branch (legacy)
- git filter-repo (preferred)

to rewrite history and remove the file, then force-push:

```
git filter-repo --path <file> --invert-paths
```

---

#### 5. How do you use git revert and git reset to undo changes?

git revert <commit> creates a new commit that undoes changes.

git reset --hard <commit> moves HEAD and branch to a specific commit, discarding later commits (local use).

---

#### 6. How can you recover a file deleted in a previous commit?

Find the commit where the file existed:

```
git checkout <commit_hash> -- <file>
```

---

#### 7. What is the difference between git reflog and git log for recovery?

git log shows the commit history;

git reflog records all HEAD movements (including deleted branches/commits), useful for recovery.

**8. How do you use git cherry-pick to restore changes from another branch?**

git cherry-pick <commit\_hash> applies a specific commit from another branch onto the current branch.

---

**9. How do you remove sensitive data (e.g., passwords) from commit history?**

Use git filter-repo (or filter-branch) to remove secrets, then force-push to rewrite remote history.

---

**10. How can you prevent accidental data loss when cleaning up branches or commits?**

- Always back up with git branch <backup> <ref> or create tags before destructive operations
- Review with git reflog if needed.