

Here are 30 Git and GitHub scenario-based interview questions and answers specifically tailored for DevOps Engineers, with real-time examples:

## Basic Git

### Scenarios

1. How do you clone a specific

# branch from a remote repository?

**Scenario:** You need to work on a feature branch and avoid downloading unnecessary data.

**Answer:**

```
git clone -b <branch-name> --single-branch  
<repo-url>
```

**Example:**

```
git clone -b feature-login --
```

single-branch <https://github.com/example/repo.git>

## 2. How do you rename a local Git branch?

**Scenario:** You mistakenly named your local branch incorrectly and need to rename it.

**Answer:**

`git branch -m <new-`

branch-name>

**Example:**

git branch -m fix-login-issue

**3. How do you**

**discard**

**uncommitted**

**changes in a file?**

**Scenario:** You have made changes to app.py, but you want to revert it to

the last committed state.

**Answer:**

`git checkout -- app.py`

Or, if using newer Git  
versions:

`git restore app.py`

**4. How do you  
delete a local Git  
branch?**

**Scenario:** You finished  
working on a feature and  
merged it; now you want

to delete the local branch.

**Answer:**

git branch -d feature-  
branch

If the branch is not  
merged yet, force delete:  
git branch -D feature-  
branch

**5. How do you  
delete a remote Git  
branch?**

**Scenario:** The feature

branch is merged, and you want to clean up the remote repository.

**Answer:**

```
git push origin --delete  
feature-branch
```

## Intermediate Git

### Scenarios

6. How do you undo the last commit but keep

# the changes?

**Scenario:** You committed a change but forgot to add a file.

**Answer:**

```
git reset --soft HEAD~1
```

**7. How do you revert a specific commit?**

**Scenario:** A commit broke production, and you need

to undo it without  
affecting later commits.

**Answer:**

git revert <commit-hash>

**8. How do you  
stash and reapply  
changes?**

**Scenario:** You need to  
switch branches but have  
uncommitted changes.

**Answer:**

git stash git checkout

another-branch git stash  
pop

## 9. How do you squash multiple commits into one?

**Scenario:** Your PR has  
multiple small commits,  
and you need to merge  
them into one before  
merging.

**Answer:**

git rebase -i HEAD~3

Replace pick with squash  
(s) for unnecessary  
commits.

## 10. How do you force push a rebased branch?

**Scenario:** You rebased  
your feature branch and  
need to update the  
remote branch.

**Answer:**  
`git push origin feature-`

`branch --force`

**Caution:** This rewrites history and should be used carefully.

## Advanced Git

### Scenarios

11. How do you resolve merge conflicts?

**Scenario:** You merged a branch but encountered

conflicts.

**Answer:**

```
git merge feature-branch  
# Resolve conflicts  
manually in files git add .  
git commit -m "Resolved  
merge conflicts"
```

**12. How do you  
recover a deleted  
branch?**

**Scenario:** You  
accidentally deleted a

local branch before pushing it.

**Answer:**

```
git reflog git checkout -b  
deleted-branch <commit-  
hash>
```

**13. How do you track changes in a specific file's history?**

**Scenario:** You want to see

who modified config.yaml and what changes were made.

**Answer:**

git log -- config.yaml

**14. How do you check which files have changed between two branches?**

**Scenario:** You need to

compare changes  
between develop and  
feature-login.

**Answer:**

git diff develop..feature-  
login

**15. How do you  
push changes to  
multiple  
repositories?**

**Scenario:** You need to

push to GitHub and  
GitLab simultaneously.

**Answer:**

```
git remote set-url --add  
origin <second-repo-url>  
git push --all
```

## **GitHub Specific**

### **Scenarios**

**16. How do you**

**create a pull**

**request from the**

# command line?

**Scenario:** You completed a feature and want to create a PR.

**Answer:**

```
gh pr create --base main --  
head feature-branch --title
```

"Feature X" --body

"Implemented feature X"

(Requires GitHub CLI: gh)

**17. How do you  
fork a repository**

# and keep it up to date?

**Scenario:** You forked a repo but need the latest updates from the original repository.

**Answer:**

```
git remote add upstream  
<original-repo-url> git  
fetch upstream git merge  
upstream/main
```

## 18. How do you

# add a GitHub Action workflow for CI/CD?

**Scenario:** You need to run tests on every commit.

**Answer:**

Create .github/workflows/ci.yml:

```
name: CI
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/
```

checkout@v2 - name:  
Install dependencies run:  
npm install - name: Run  
tests run: npm test

## 19. How do you trigger a GitHub Action manually?

Scenario: You want to  
trigger a workflow on  
demand.

Answer:  
Modify .github/

`workflows/ci.yml:`

`on: [workflow_dispatch]`

Then, trigger via GitHub

UI.

**20. How do you roll**

**back a bad**

**deployment from**

**GitHub Actions?**

**Scenario:** A recent

deployment failed, and

you need to revert it.

**Answer:**

```
git revert <bad-commit-  
hash> git push origin  
main
```

## **Real-time DevOps**

### **Scenarios**

**21. How do you use**

**Git for**

**Infrastructure as**

**Code (IaC)?**

**Scenario:** You manage Terraform configurations using Git.

**Answer:**

Use Git branches for environments:

- main → Production
- staging → Staging
- feature-xyz → Development

**22. How do you perform a GitOps**

# deployment?

**Scenario:** You use ArgoCD or FluxCD for automatic deployments.

**Answer:**

Push Kubernetes manifests to a GitHub repo, and ArgoCD syncs them.

**23. How do you implement a Git pre-**

# commit hook?

**Scenario:** You want to prevent commits with lint errors.

**Answer:**

Create .git/hooks/pre-commit:

```
#!/bin/sh npm run lint ||  
exit 1
```

Then, make it executable:  
chmod +x .git/hooks/pre-commit

# 24. How do you trigger a Jenkins pipeline from GitHub?

**Scenario:** You want Jenkins to build on every push.

**Answer:**

Use a GitHub webhook pointing to:  
<http://jenkins-server/>

github-webhook/

# 25. How do you secure Git credentials?

**Scenario:** You need to  
avoid storing passwords  
in scripts.

**Answer:**

Use SSH keys instead of

HTTPS:

```
git remote set-url origin  
git@github.com:user/
```

repo.git

# Bonus Questions

- How do you sign commits with GPG?
- How do you find a bug using git bisect?
- How do you apply a single commit from another branch? (git cherry-pick)
- How do you compare changes between

# commits?

- How do you use GitHub Codespaces for DevOps?

Would you like explanations for any of these?