

## Initial Setup

### 1. Configure Git

```
git config --global user.name "Your Name"
```

```
git config --global user.email "you@example.com"
```

### 2. Generate SSH Key (if not already done)

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Add the SSH key to your Git hosting service (e.g., GitHub, GitLab).

### 3. Clone Repository

```
git clone git@github.com:username/repository.git
```

```
cd repository
```

## Workflow

### 4. Create and Switch to a New Branch

```
git checkout -b new-feature
```

### 5. Check Status

```
git status
```

### 6. Stage Changes

```
git add .
```

## 7. Commit Changes

```
git commit -m "Add new feature"
```

## 8. Pull Latest Changes from Remote

```
git pull origin main
```

## 9. Resolve Conflicts (if any)

# Manually resolve conflicts in files, then

```
git add resolved_file
```

```
git commit -m "Resolve merge conflict"
```

## 10. Push Changes to Remote

```
git push origin new-feature
```

## Code Review

### 11. Create a Pull Request (PR)

- Go to the repository on your Git hosting service and create a new PR from `new-feature` to `main`.

## Tagging and Releases

### 12. Tagging a Release

```
git tag -a v1.0 -m "Release version 1.0"
```

```
git push origin v1.0
```

### 13. Creating a Release

- On your Git hosting service, navigate to the Releases section and create a new release, attaching the tag `v1.0`.

## Ignoring Files

### 14. Add .gitignore File

```
# .gitignore
```

```
node_modules/
```

```
*.log
```

```
.env
```

```
git add .gitignore
```

```
git commit -m "Add .gitignore file"
```

## Additional Commands

### 15. Rebase Branch

```
git checkout main
```

```
git pull origin main
```

```
git checkout new-feature
```

```
git rebase main
```

## 16. View Commit History

```
git log
```

## 17. Amend Last Commit

```
git commit --amend -m "New commit message"
```

## 18. Stash Changes

```
git stash
```

## 19. Apply Stashed Changes

```
git stash apply
```

## 20. Delete a Branch

```
git branch -d new-feature
```

```
git push origin --delete new-feature
```

## 21. Reset to a Specific Commit

```
git reset --hard <commit-hash>
```

## 22. Cherry-Pick a Commit

```
git cherry-pick <commit-hash>
```

### 23. Fetch Updates from Remote

```
git fetch origin
```

### 24. Show Differences

```
git diff
```

### 25. Blame

```
git blame <file>
```