

GitHub Workflow Guide — 1 Page Notes

◆ 1. Overview

GitHub workflow defines how you collaborate, version-control, and deploy code using Git and GitHub.

Main stages:

➡ *Create → Clone → Branch → Commit → Push → Pull Request → Review → Merge → Deploy.*

◆ 2. Basic Workflow Steps

Step 1: Create / Clone Repository

```
# Create new repo on GitHub → Copy HTTPS/SSH URL
```

```
git clone <repo_url>      # Clone locally
```

```
cd <repo_name>
```

Step 2: Create a Branch

Keep the main branch stable. Work in a feature branch.

```
git checkout -b feature-login
```

Step 3: Make Changes & Commit

Edit files → Stage → Commit

```
git add .
```

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

Step 4: Push to GitHub

```
git push origin feature-login
```

◆ 3. Collaboration Using Pull Requests (PR)

- Go to the GitHub repository.
- Click “Compare & pull request.”
- Add a title and description → Submit PR.
- Reviewer checks and comments → Approve or request changes.
- Once approved, Merge into main or dev.

Pro Tip:

Use “Draft Pull Request” for ongoing work.

◆ **4. Syncing with Remote**

Keep your local branch updated.

`git fetch origin`

`git pull origin main`

If merge conflicts occur:

- Open files → Resolve conflicts manually.
 - Stage & commit resolved files.
-

◆ **5. Issue Tracking**

- Use Issues tab to report bugs, plan features, or assign tasks.
 - Link PRs to issues using:
👉 Fixes #<issue_number> in your commit or PR description.
-

◆ **6. Code Review Best Practices**

- ✓ Use clear PR titles
 - ✓ Write meaningful commit messages
 - ✓ Keep PRs small and focused
 - ✓ Test before merging
 - ✓ Delete merged branches
-

◆ **7. GitHub Actions (Automation)**

- Automate tests, builds, and deployments.
- Defined in `.github/workflows/` directory using YAML files.
Example:

`on: [push, pull_request]`

`jobs:`

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3
 - run: npm install && npm test
-

◆ **8. Common Commands Recap**

git checkout -b new-feature

git add .

git commit -m "message"

git push origin new-feature

git pull origin main

git merge main

◆ **9. Typical Team Branch Model**

main → production code

dev → integration branch

feature/* → new features

hotfix/* → urgent fixes

 **Summary:**

Code → Commit → Push → Pull Request → Review → Merge → Deploy.

That's the heart of the GitHub Workflow.