**Step 1: Create the Necessary Branches**

Ensure you have the following branches in your GitHub repository:

1. **developer** – Developers push code here.
2. **deploye** – Developers push to this branch only.
3. **preprod** – Deploye branch pushes here.
4. **main** – Preprod branch pushes here.

**Step 2: Restrict Main Branch**

1. Go to your GitHub repository.
2. Click on **Settings** > **Branches**.
3. Under "Branch Protection Rules," click **Add Rule**.
4. Set the branch name to main.
5. Enable the following:
   * ✅ **Require pull request reviews before merging**.
   * ✅ **Require status checks to pass before merging** (if you use CI/CD).
   * ✅ **Include administrators (Optional)**.
   * ✅ **Restrict who can push to matching branches** > Add only trusted users.
   * ✅ **Allow only specific branches to merge into main** → Select **preprod**.

**Step 3: Restrict Preprod Branch**

1. Add another branch protection rule for preprod.
2. Enable:
   * ✅ Require pull request reviews.
   * ✅ Require status checks.
   * ✅ Restrict push access to **only allow deploye branch**.

**Step 4: Restrict Deploye Branch**

1. Add a protection rule for deploye.
2. Enable:
   * ✅ Require pull request reviews.
   * ✅ Require status checks.
   * ✅ Restrict push access to **only allow developer branch**.

**Step 5: Allow Developers to Push to Developer Branch**

1. Do **not** add protection to the developer branch so that developers can freely push.
2. Ensure developers **cannot push directly** to deploye, preprod, or main.

**Step 6: Enforce the Workflow with Pull Requests**

1. Developers push to **developer**.
2. They create a **pull request (PR)** from developer → deploye.
3. After review, the PR is merged.
4. A new PR is created from deploye → preprod (merging only after testing).
5. Finally, a PR is created from preprod → main, merging after final approval.