Step 6: CI/CDFollowing are the steps to design and setup a CI/CD pipeline for the application. We also give the manual or automated gates in the process.

We use GitHub to set up a CI/CD pipeline:

* Ensure the project exists in GitHub

If the project has not been pushed to GitHub yet, open a terminal and push the local project to a GitHub repository.

git remote add origin https://github.com/example/user-app.git

git push -u origin main

* Enable GitHub Pull Request Approval Protection

In the repository, go to Settings > Branches.  
Under Branch Protection Rules:  
Click Add branch protection rule.  
Set the Branch name pattern to main  
Enable the following options:

* Require a pull request before merging
* Require approvals (set the number of required approvals)
* Create a GitHub Actions workflow

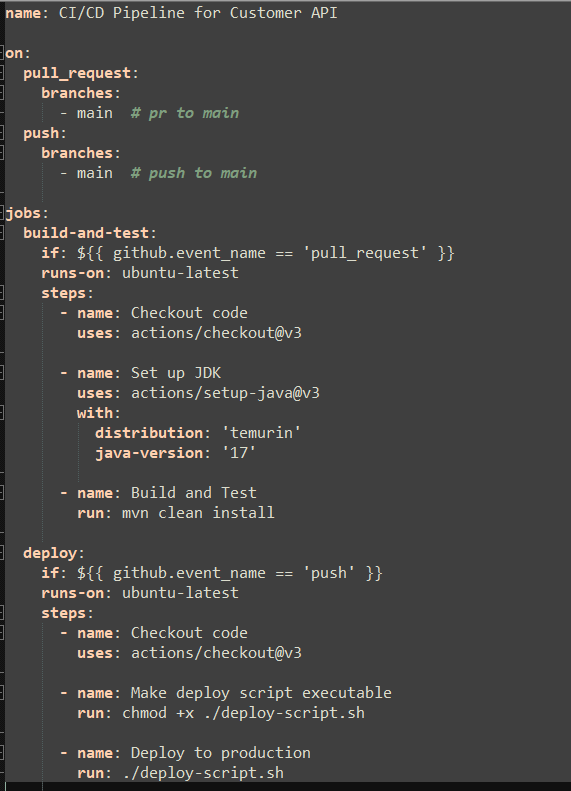
On the GitHub project page, click the **Actions** tab and New workflow.

GitHub will recommend some workflow templates. Choose a suitable template or click **Set up a workflow yourself**.

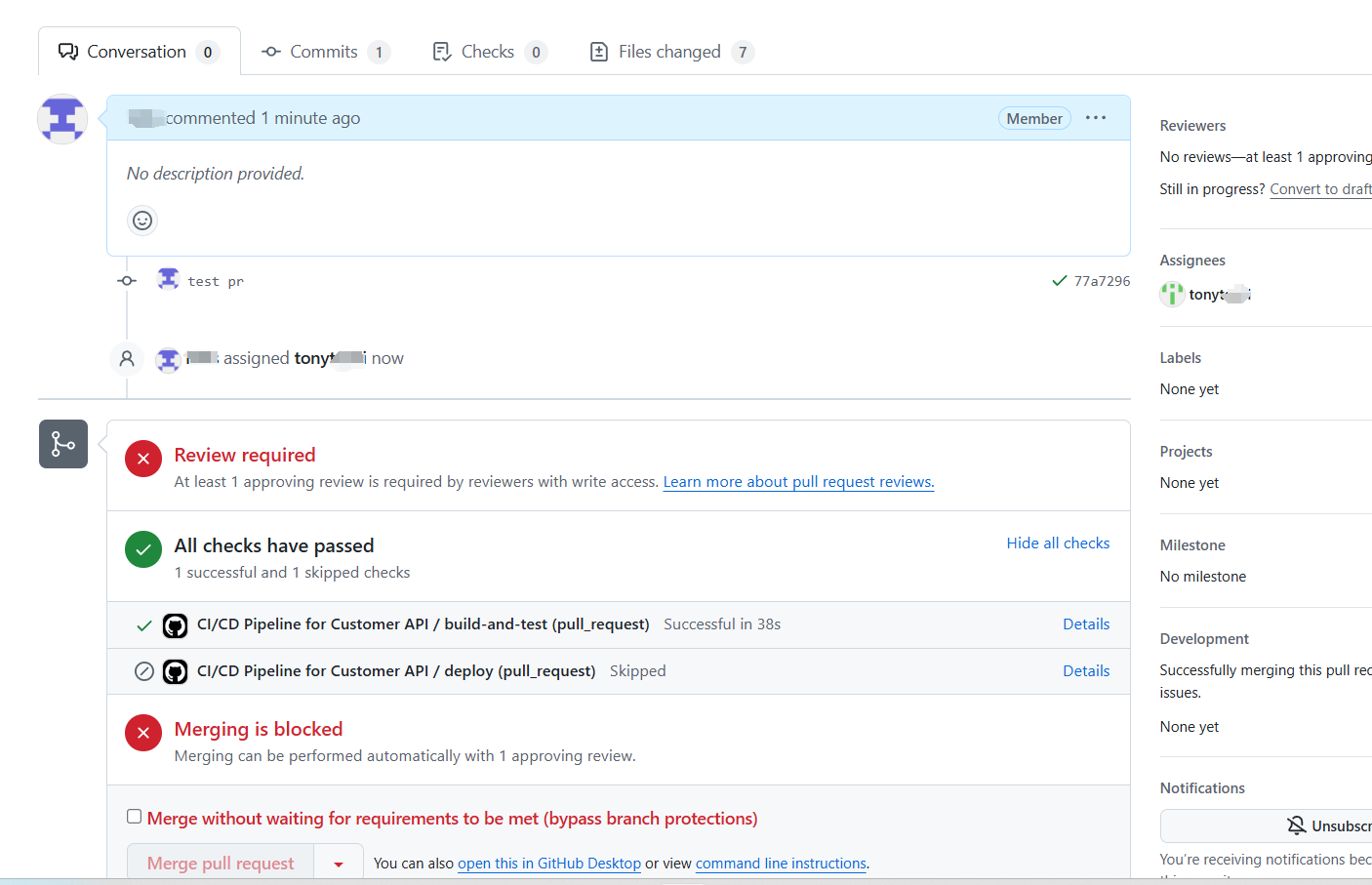
Or create a new YAML file in the .github/workflows/ directory (e.g., ci-cd.yml) and add to the repository.

* The CI/CD configuration file

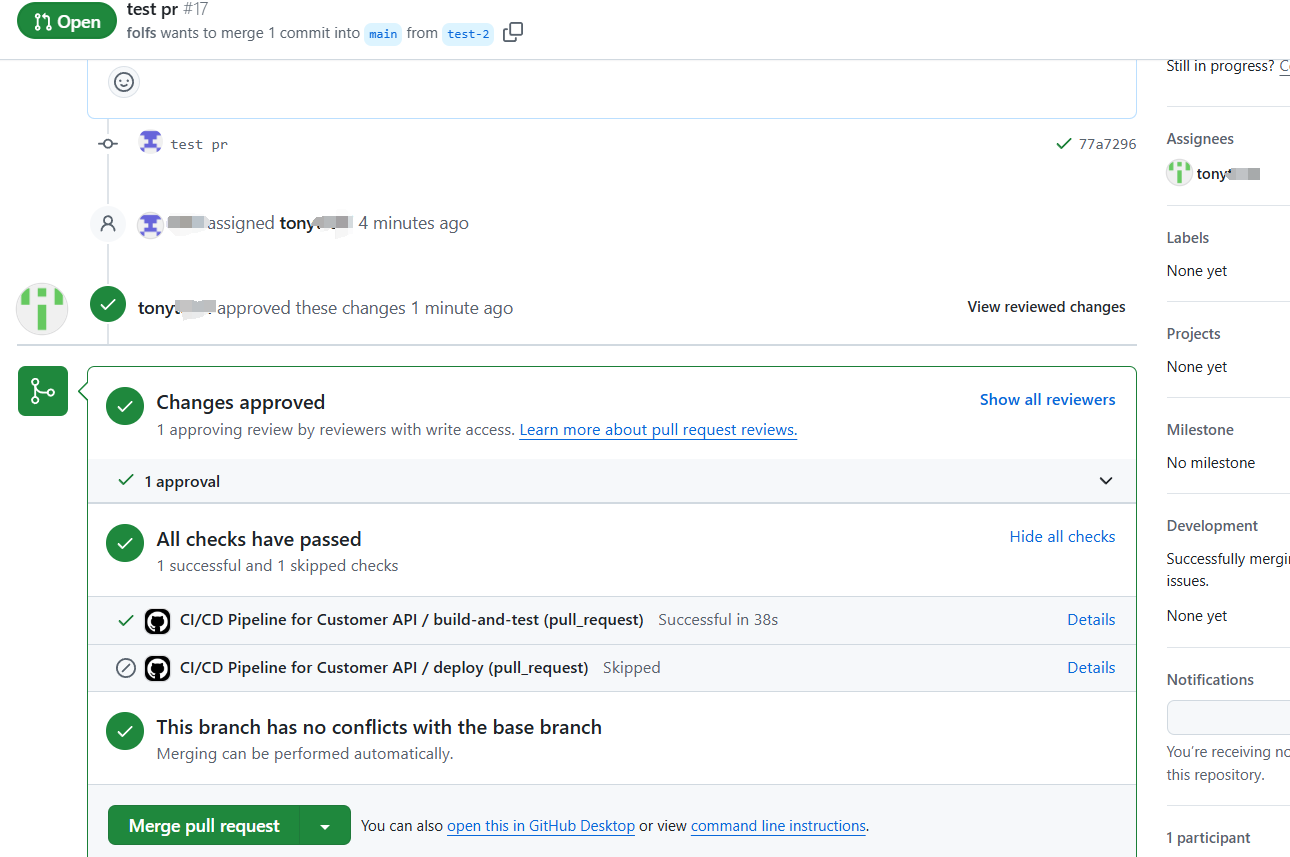
GitHub Actions uses YAML files to define workflows. We need define triggers and jobs. For example the following is a simple example (./github/workflows/ci-cd.yml) for our project:

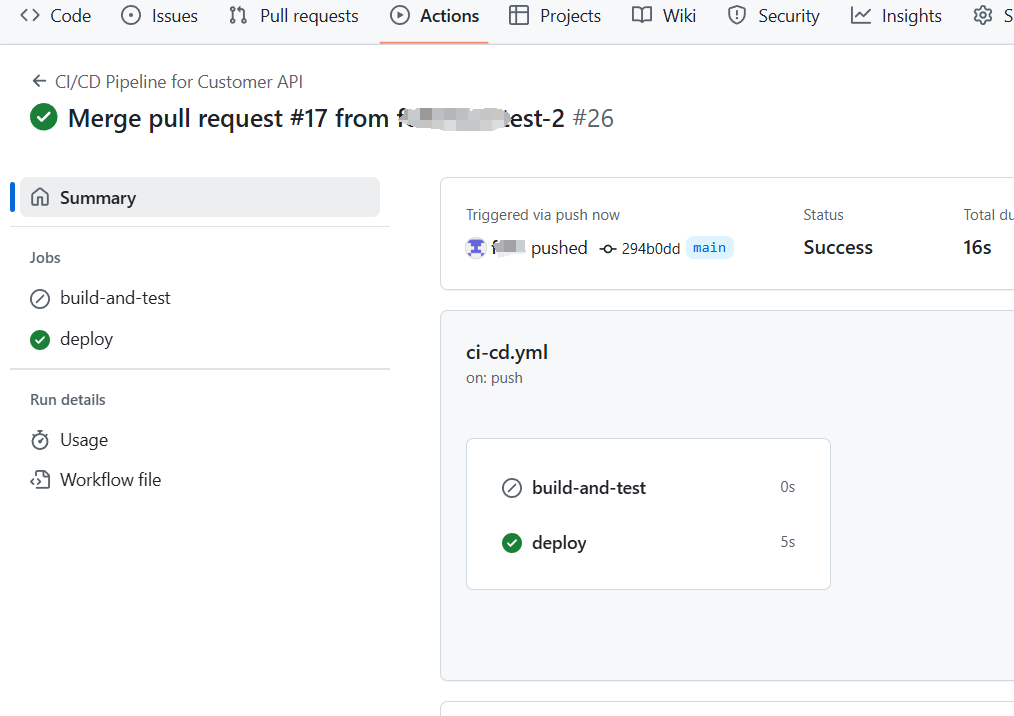


For this configuration file, the 'build-and-test' job will be triggered once there is a pull request to the main branch. If the build or tests fail, an email can be sent to the related developers. Additionally, the CI pipeline will be blocked until the code review is approved by the reviewers.



After the code review approved, then we can click Merge pull request to deploy the service to cloud environment defined by deploy.sh.





* **Manual and Automated Gates in the Pipeline**
* **Automated Gates**:
  + **Build and Test**: The build-and-test job automatically checks whether the code compiles and passes unit tests. If this step fails, the pipeline stops, and the deployment does not proceed. This is a typical automated gate that ensures the quality of the code.
  + **Deployment**: The deployment only occurs if the PR has been reviewed, merged and the build/test steps are successful. This ensures that only reviewed and tested code gets deployed.
* **Manual Gates**:
  + **PR Review and Approval**: For PRs, we added a **manual approval gate** for code review. It is essential to ensure that the code is reviewed by the team before being merged into the main branch.
  + **Approval for Deployment**: We need click Merge pull request and Confirm merge to proceed to deployment stage.