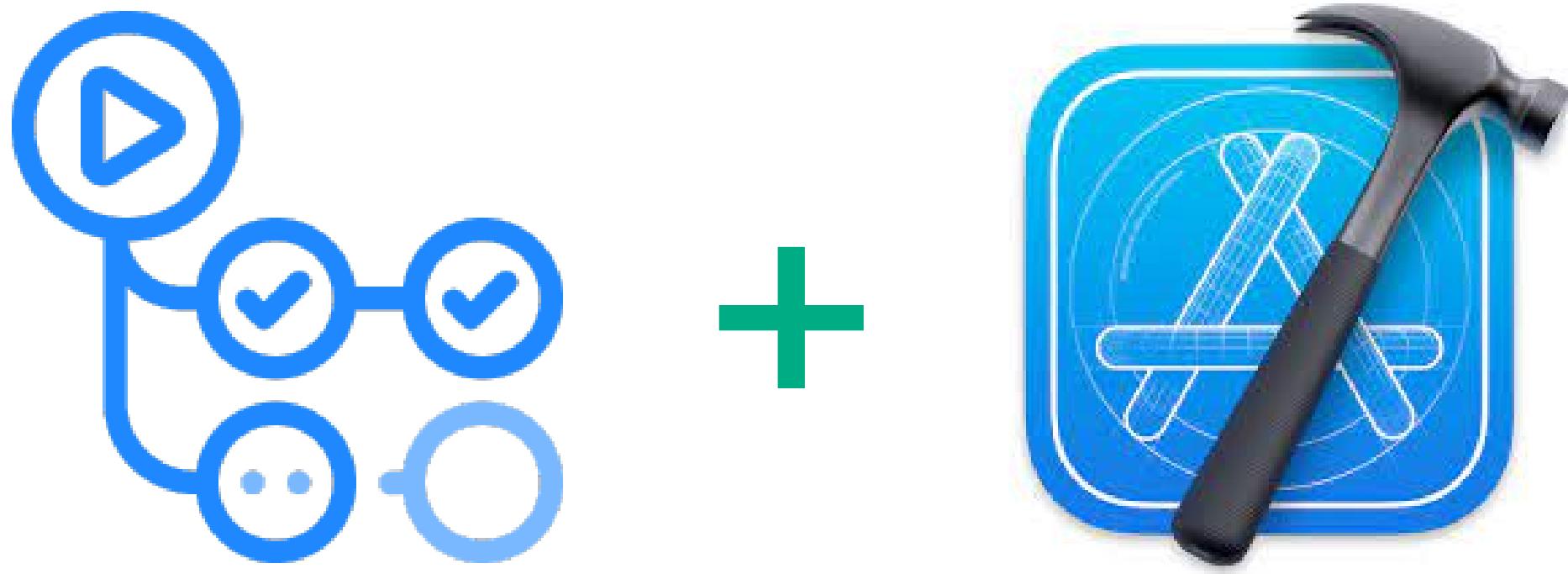


# CI/CD with GitHubActions



**SreeSaiRaghava**  
iOS Developer

# GitHubActions

GitHub Actions is a CI/CD platform by GitHub

You can create workflows to build, test and deploy changes to your repo.

GHActions provides Linux, Windows and macOS virtual machines to run your workflows.

You can also host own runners (VMs) in your cloud or data centers.

## Benefits

- 😎 Freemium model, and free for public repos :)
- 😊 Large community support and ready to use workflows



**SreeSaiRaghava**  
iOS Developer

# Demo with Xcode

## Prerequisites

1. You can create new or use existing project.
2. Add your local project to GitHub or use the existing remote repo.
3. Add UnitTest target to your project, if you don't already have it enabled while creating.

Let's start creating our workflow to build, test our changes on either push to main branch or pull request.

## Workflow

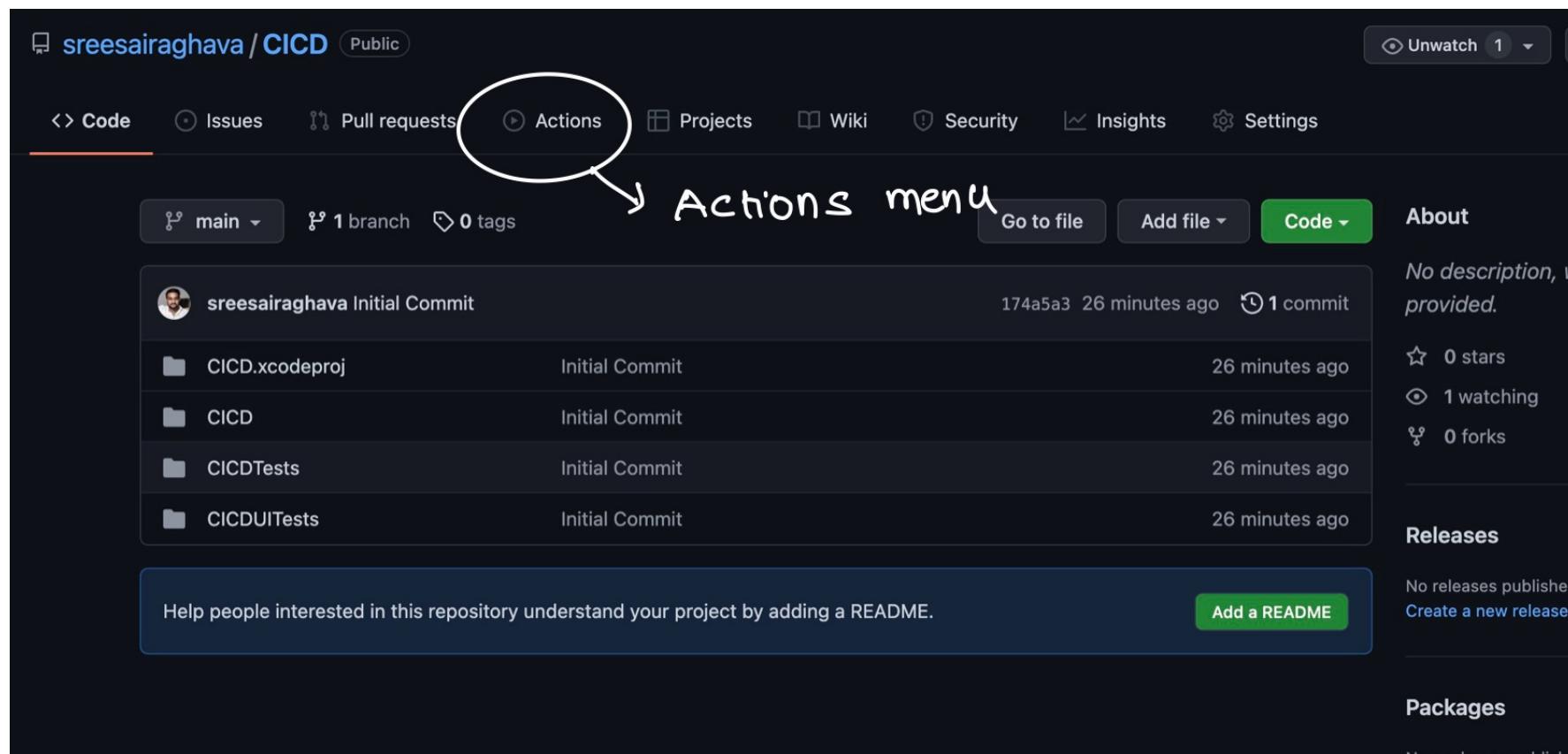
- ✓ Workflow is a configurable file created in our repo
- 🚀 We can configure to build, test, deploy and more



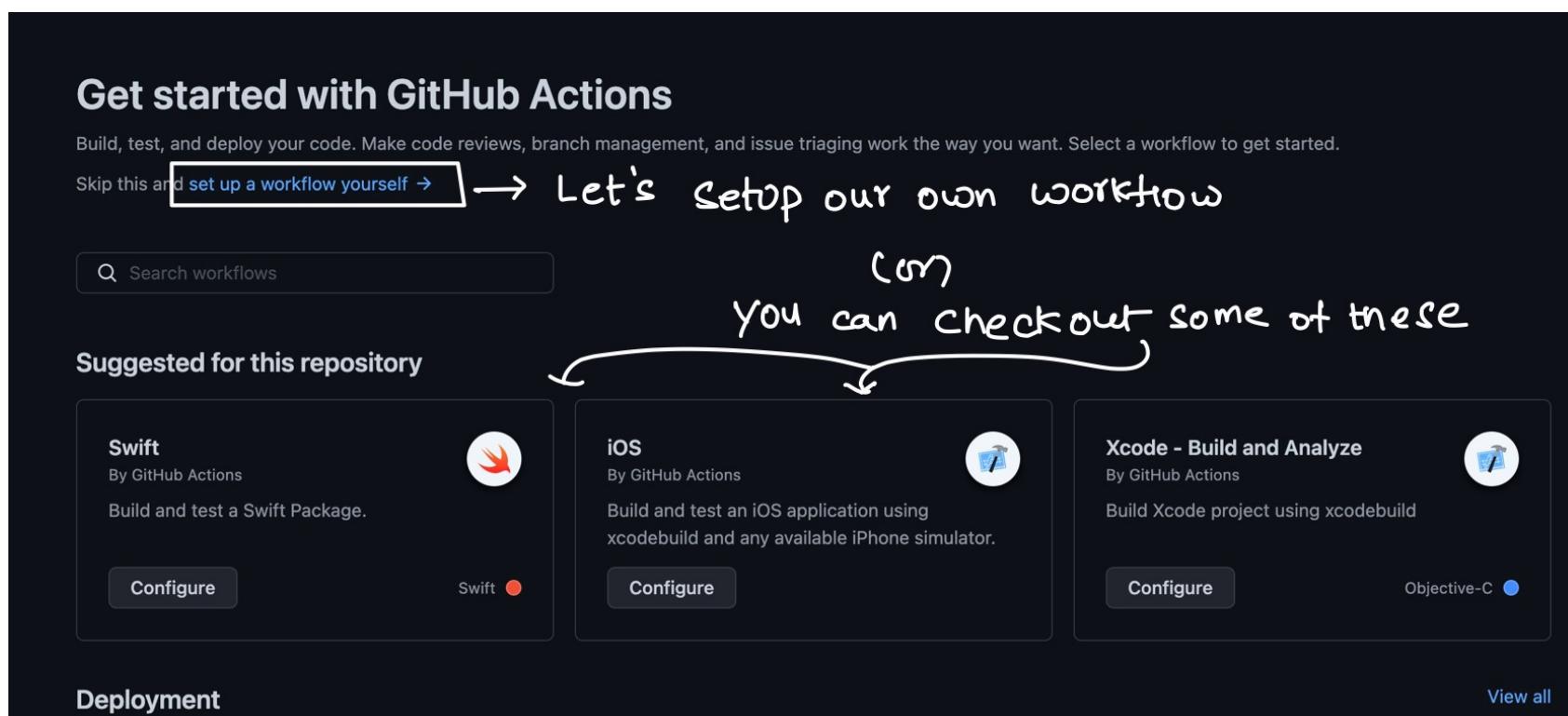
**SreeSaiRaghava**  
iOS Developer

# Let's setup workflow

Open your repo > Go to “Actions” Menu next to “Pull Requests”



Click on “Set up a workflow yourself” button as highlighted



**SreeSaiRaghava**  
iOS Developer

# Configure the YAML file

Choose your workflow file name with .yml extension > explore editor

The screenshot shows a GitHub Actions YAML configuration file named `Clyml` in the `main` branch. The code defines a workflow named `CI` that triggers on push or pull request events for the main branch. It contains a single job named `build`. A handwritten note on the right side of the screen points to the file with the text "Configurable YAML File". To the right of the code editor is the GitHub Marketplace search results page for "source workflows". An arrow points from the text "Available open source workflows" to the search bar. Below the search bar, several actions are listed:

- Cache: Cache artifacts like dependencies and build outputs to improve workflow execution time.
- Upload a Build Artifact: Upload a build artifact that can be used by subsequent workflow steps.
- Setup Go environment: Setup a Go environment and add it to the PATH.
- Download a Build Artifact: Download a build artifact.

Configure file for build, tests execution & commit to main branch

The screenshot shows a GitHub Actions configuration file named `configuration.yml` with handwritten annotations. The file defines a workflow named `CI` that triggers on push or pull request events for the main branch. It contains a job named `test` with a strategy matrix. The matrix destination is set to `['platform=iOS Simulator,OS=15.2,name=iPhone 12 Pro Max']`. The job has two steps: `Checkout` and `Build`. The `Build` step runs the command `xcodebuild clean build -project CICD.xcodeproj -scheme CICD -destination "\${destination}" CODE\_SIGN\_IDENTITY="" CODE\_SIGNING\_REQUIRED=NO ONLY\_ACTIVE\_ARCH=NO`. The file also includes a test step that runs the command `xcodebuild clean test -project CICD.xcodeproj -scheme CICD -destination "\${destination}" CODE\_SIGN\_IDENTITY="" CODE\_SIGNING\_REQUIRED=NO ONLY\_ACTIVE\_ARCH=NO`. Handwritten annotations explain the code:

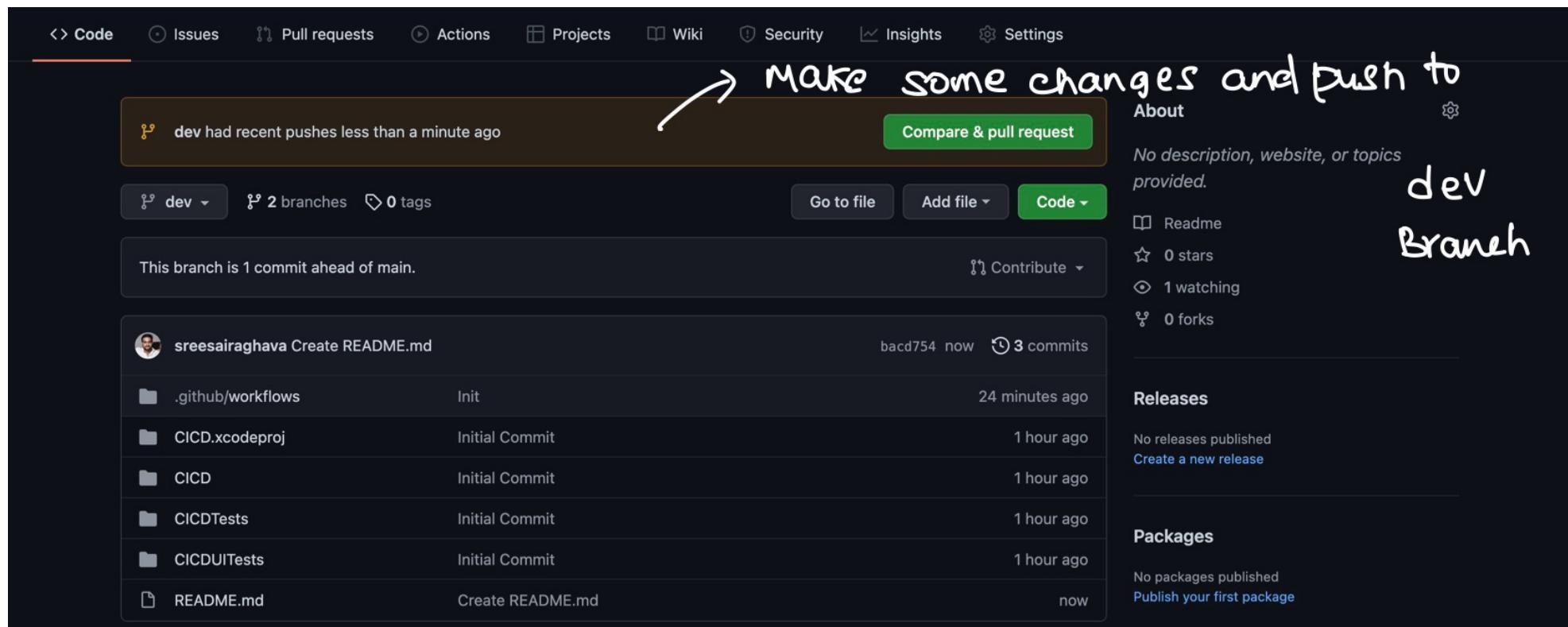
- A bracket under the `on` section is labeled "Events on which CI will be triggered".
- A bracket under the `steps` section is labeled "check out code".
- A bracket under the `matrix` section is labeled "destination".
- A bracket under the `env` section is labeled "Your project Name".
- An arrow points from the word "Build" to the command `CODE\_SIGN\_IDENTITY="" CODE\_SIGNING\_REQUIRED=NO ONLY\_ACTIVE\_ARCH=NO` with the label "Build command".
- An arrow points from the word "Test" to the command `xcodebuild clean test` with the label "Test command".



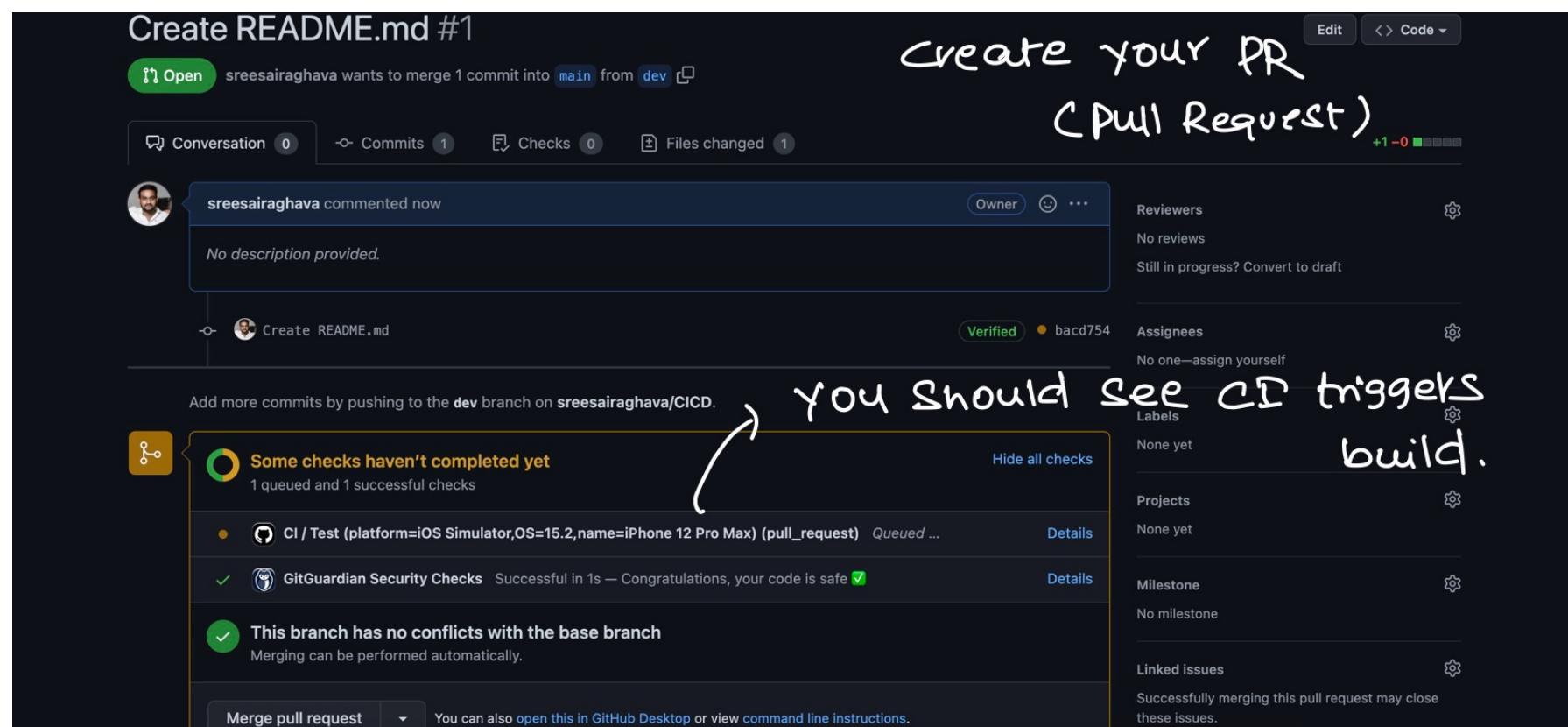
SreeSaiRaghava  
iOS Developer

# Create a PR to test CI

Create a dev branch, make some changes, push to origin and raise PR



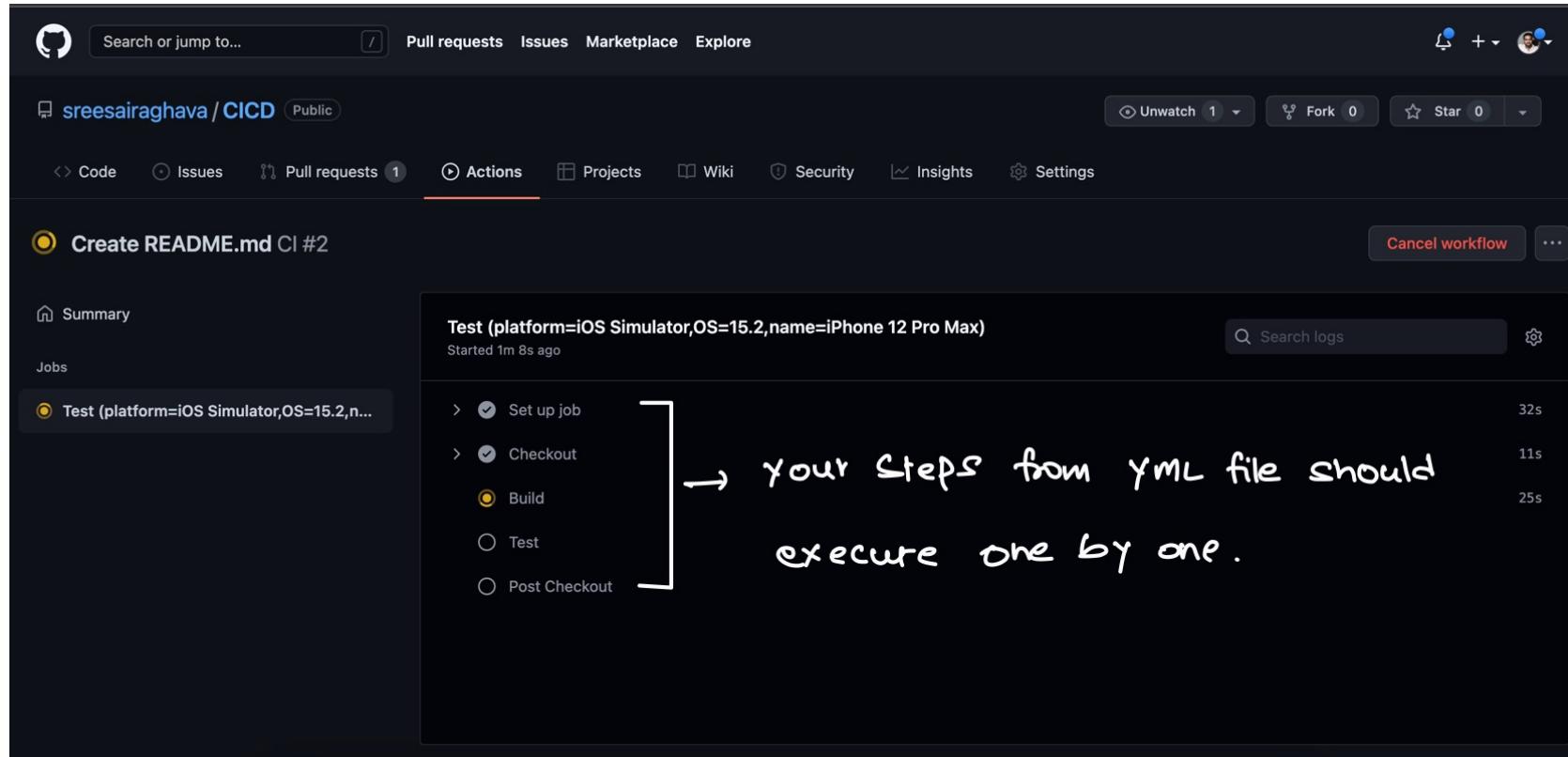
You might notice our CI has triggered build and test execution



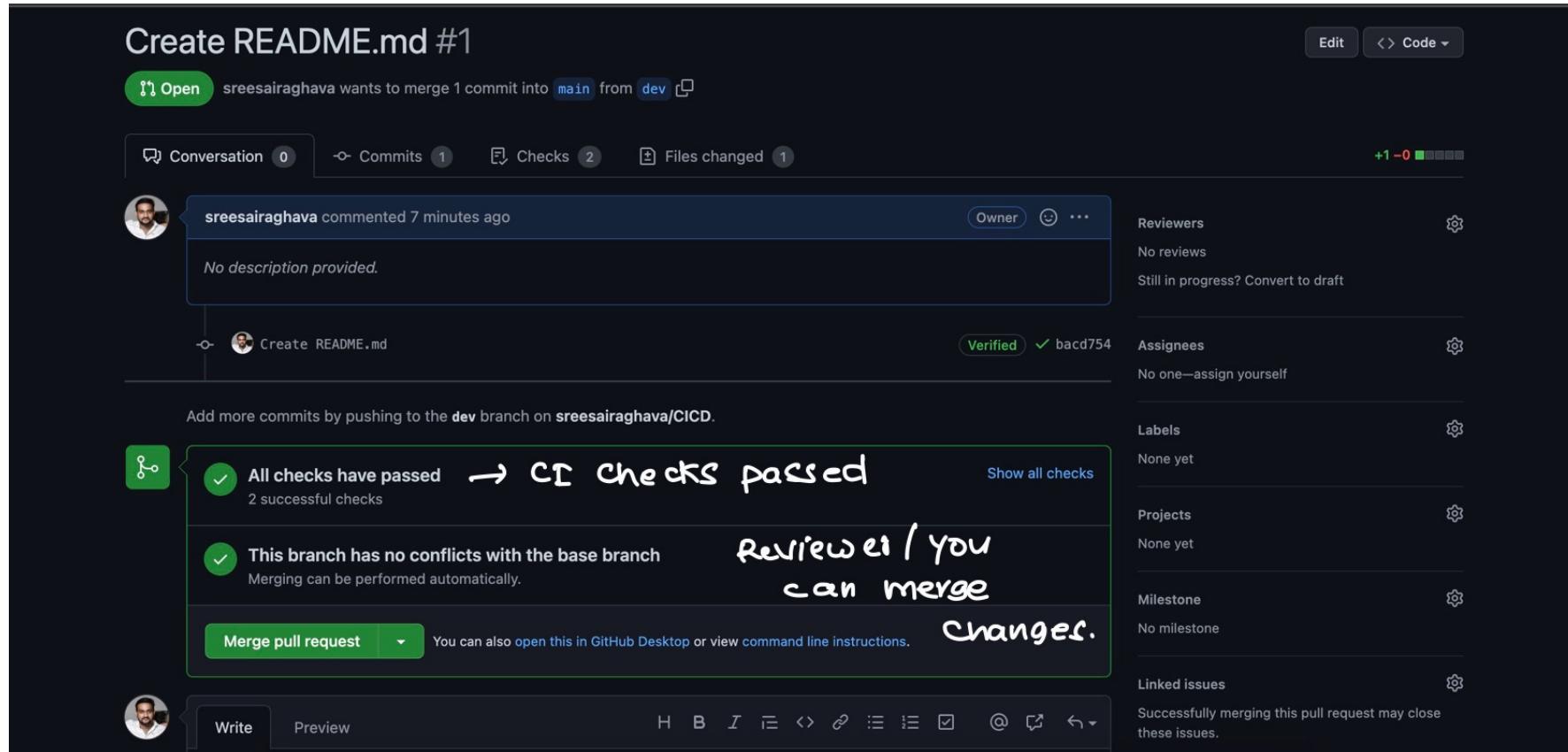
SreeSaiRaghava  
iOS Developer

# Verify your CI checks

Click on details from checks page > CI steps execute sequentially



Hola, once our code passes all checks, Merge into main branch :)



**SreeSaiRaghava**  
iOS Developer

# Conclusion

GitHubActions offer freemium model if you would like to use this for private repos with limited build minutes.

macOS build minutes are multiplied by 10 so 100 build minutes of macOS would be 1000 mins from your account.

You can configure your own host runners if you are working for a team and need more build minutes at low price point.

## Benefits

- 😎 You can integrate apps like Slack to notify team on success
- 😊 Choose and customize from different pre-defined workflows



**SreeSaiRaghava**  
iOS Developer

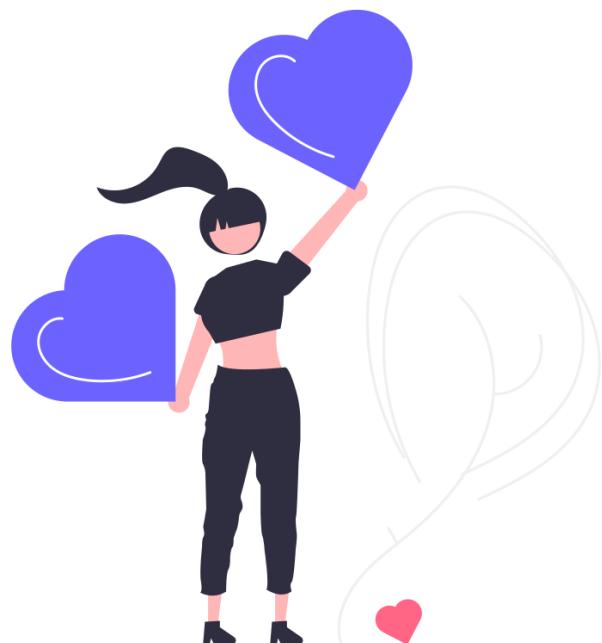
# **Thank You for coming till here :)**

**Let me know what you would like to  
see for next post in comments!**



**SreeSaiRaghava**

iOS Developer



Do follow @raghava.codes on Instagram  
For more iOS related content