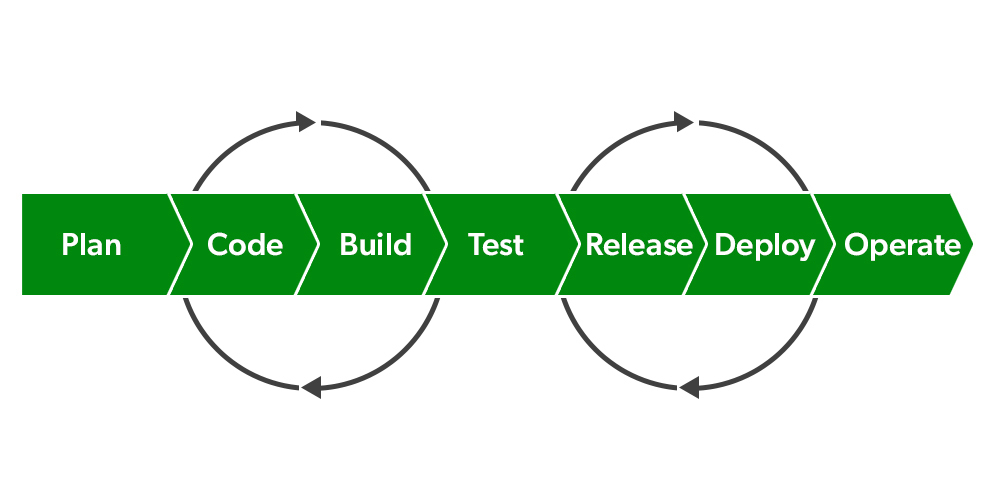
**GitHub Actions**

GitHub Actions is a CI/CD (Continuous Integration/ Continuous Deployment) platform for automating the builds, test, and deployment process. Using GitHub actions, we can build and test every pull request in the repository using workflows, or push the merged pull requests to production with workflows. GitHub Actions allows you to conduct processes in response to the events in your repository. For example, anytime someone files a new issue in your repository, you may execute a process to automatically add the required labels or if someone sends a pull request to your repository, then the automated workflows built by you can check if the pull request is eligible for merging into the main code or not. The figure below explains the flow of work done by a software engineer and the need for CI/CD pipelines.



The role of GitHub Actions comes into the scene after step 2. When you’ve finished coding the project, upload it to GitHub and build a script to test it every time you commit a change or someone else creates a pull request. If any exception occurs, GitHub Actions will fail the build, preventing the modifications from being merged. Otherwise, the build will succeed, and the modified project will be available for download.

**Components Of GitHub Actions**

When an event occurs in the repository, such as when a pull request is opened or an issue is raised, one can set up the GitHub Actions process to be triggered. The workflow includes one or more than one job that can be executed sequentially or concurrently. The components of GitHub Actions are listed below:

**1. Workflows**

A workflow is a programmable automated procedure that executes one or more tasks. Workflows are written in a YAML (Yet Another Markup Language) file that is checked into your repository. They can be triggered by an event in your repository, manually, or on a set timetable. There can be n number of workflows that can be triggered at the desired time. For example, you may have a workflow that builds tests for every pull request that is sent to your repository or a workflow that pushes your code into production once the built test passes. So as soon as someone sends a pull request to your repository, the first workflow triggers and build the code to check if it is working fine or not. All the workflows are written in a specific file and in a specific directory on GitHub that is, **.github/workflows/file.yml**. Here .github/workflows are the directory and file.yml is the file with the extension in which the script will be written.

**2. Events**

Events are the states or activities at which the workflows will be triggered. When someone pushes the newly committed code to the repository or files an issue or creates a pull request, are some events that will ultimately trigger the GitHub actions to check for workflows, if any, and do the jobs as specified in the workflow. The job can also be started manually or on a specified timetable.

**3. Jobs**

A job is like a collection of workflow stages that run on the same runner. Each step can either be a shell script or an action that will be executed. All the steps are carried out in a sequential manner and are interdependent. Being interdependent, the exchange of data between the steps can be done easily. Let us take the previous example only, which says to send the code to the production once the build passes. Now, as soon as the build passes the next step to send the code to the production will be triggered else it will fail.

**4. Actions**

An action is a GitHub Actions platform-specific custom program that performs a sophisticated but regularly performed job. Use action to help you write less repeated code in your workflow files. An action can grab your git repository from GitHub, configure your build environment’s toolchain, or configure your cloud provider’s credentials.

**5. Runners**

In simple terms, runners are the servers that execute the workflows built by you. Just like a C++ code needs to be compiled first using mingw and then run, similarly when the workflows are triggered, a runner executes them. Each runner is limited to one job at a time. To run the workflows, GitHub provides Ubuntu Linux, Microsoft Windows, and macOS runners, each workflow run takes place on a new, freshly provisioned virtual machine. You can host your own runners if you need a different operating system or a specific hardware setup.