GitHub Actions: An In-depth Guide for Beginners

Actions Platform?

ALMAS

It would not be wrong to mention that GitHub Actions has significantly transformed the workflow of **web developers**. This Continuous Integration and Continuous Delivery (CI/CD) platform enables them to build, test, and deploy web codes straight from GitHub.

Are you a beginner? Do you want to learn how this tool can boost your productivity? Read this guide until the end. It makes you aware of the components and features of GitHub Actions.

What is GitHub

GitHub Actions is an automated tool powered by GitHub. It supports the automation of software building, testing, and deployment within the repositories of GitHub.

Since the user does not need to leave GitHub, it naturally enhances the workflow and productivity. Developers can perform repetitive tasks while reducing manual intervention.

GitHub Actions utilizes a YAML file to outline different steps of a workflow. These steps include running a script, testing, deploying codes, and sending notifications.

Components of GitHub Actions

GitHub Actions is a powerful tool that makes web development smooth and quick. Wondering what mechanisms make GitHub Actions work so well?
Let's learn about them.

Workflow

A workflow is a programmed process that runs one or more jobs. This configurable process is defined by a YAML file in the .github/workflows directory in a repository. This repository can have several workflows. And each workflow can perform a different set of jobs.

For instance, you can use one workflow to create and test pull requests while another to deploy your application.

Events

An event is a particular activity in a repository. It is like a trigger for workflows. When events occur within a repository, GitHub Actions respond to them. These events can push requests, pull requests, or other actions.

Jobs

Jobs are a set of steps in a workflow. They are executed under the same runner. Each step is either a shell script or an action. Scripts execute while actions run.

Action

An action is an application for the GitHub Actions. It performs frequently repeated tasks. The application helps web developers to reduce the number of repetitive codes they write in their workflow files.

Runner

A runner is a server that runs workflows when they are triggered. One runner can perform a single task at a time.

Essential Features of GitHub Actions

Though GitHub Actions offers various advantages to web developers, a few prominent features are below.

Variable in Workflows

The default GitHub actions environment variables, incorporated in every workflow, run automatically. However, users can customize the environment variables by setting them in their YAML files.

In the following example, you can see how one can create custom variables for POSTGRES_HOST and POSTGRES_PORT. These variables are available in the node client.js script.

```
jobs:
   demo-job:
      steps:
          - name: Connect to your PostgreSQL
             run: node client.js
             env:
                POSTGRES_HOST: postres
                POSTGRES_PORT: 5432
```

Addition of Scripts to Workflow

GitHub Actions allow the addition of scripts to workflow. You can employ actions for running scripts and shell commands. They get executed on the selected runner.

Find out how an action can use the run keyword to execute npm install –g bats on the runner in the flowing example.

```
jobs:
   demo-job:
      steps:
         - run: npm install -g bats
```

Sharing Data Between Jobs

One of the crucial features of GitHub Actions is that you can reuse the jobs you created earlier. You can save files for later use as artifacts on GitHub.
These files get generated while building and testing web code. These files could be screenshots, binary, test results, and package files.

You can also make your file and upload it on artifacts for later use.

```
jobs:
   demo-job:
      name: Save output
      steps:
           - shell: bash
              run: |
                 expr 1+1 > output.log
           - name: Upload output file
              users: actions/upload-artifact@v3
              with:
                 name: output-log-file
                 path: output.log
```

Step-by-Step Creation of GitHub Action File

If you want to learn the workings of the GtiHub actions workflows, here is the step-by-step guide.

You will need a GitHub repository to create the GitHub actions.

Set up of the GitHub Action File

- Make a .github/workflows directory in your repository on GitHub in case it does not already exist.

- In the directory, you may create a file name: GitHub-actions-demo.yml.

- Next, copy the following YAML content into the GitHub-actions-demo.yml file.

```
name: GitHub Actions Example
on: [push]
jobs:
  Explore-GitHub-Actions:
  runs-on: ubuntu-latest
     steps:


run: echo " The job was automatically triggered by a ${{Github.
  run: echo " This job is now running on a ${{ runner.os }} ser
  run: echo " The name of your branch is ${{ GitHub.ref }} and
  name: Check out repository code
  user: actions/checkout@v3
  run: echo " The ${{ GitHub.repository }} repository has been
  run: echo " The workflow is now ready to test your code on th
  name: List files in the repository
  run: |
     Is ${{ GitHub.workspace }}
  run: echo " This job's status is ${{ job.status }}."
```

- Create a new branch for this commit and begin a pull request.
- To create a pull request, click Propose new file.
- When you commit your workflow file to a branch within your repository,

it initiates the push event and then executes your workflow.

Run the Files

Your next step should be running the file.

- Visit github.com and go to the main page of the repository.
- Beneath your repository name, click Actions.
- On the left sidebar, hit the workflow you want
- Under Jobs, click on the Explore-GitHub-Actions job.

The above log shows the breakdown of each step carried out. You can expand these steps to view its details.

Conclusion

GitHub Actions is a robust automation tool that streamlines development workflows. Web developers can leverage its flexibility, automation, and integration within GitHub. In addition to this, the platform supports event-driven workflows.

In this blog, we learned about components of **GitHub Actions**. Also, we came to know about its essential features.

All-in-all, GitHub Actions is a versatile tool for developers that simplifies the complexities of web development. If you haven’t tried it yet, embrace its power and begin your next project with better productivity and flow.