

Devops Report

TOOL: Github Action



**Introduction to GitHub Actions:**

GitHub Actions is like a helper that works closely with GitHub, the place where people store their computer code. It's like having a smart assistant for your code projects. It can do all sorts of tasks automatically whenever something important happens to your code. In this report, we'll take a close look at what GitHub Actions can do and help to understand how to use it effectively.

Imagine having a magic assistant for our code projects. This assistant, GitHub Actions, can do many things automatically based on what's happening with our code. We'll explore its abilities and how it can be used efficiently in this report.

**GitHub Actions:**

1. **Building Code Automatically:**
   * Making our code work without any manual effort applying.
2. **Checking Code for Errors:**
   * **Ensuring that our code is free from any sort of mistakes and errors.**
3. **Notifying You and Your Team:**
   * Keeping us informed when any of the important changes happen.
4. **Getting Projects Online Easily:**
   * Making it simple to share our work on the internet.
5. **Automating Tasks:**
   * Saving our time by handling repetitive work and make it automated to save time, this is how GitHub actions provide efficient workflow.

**Getting Started with Exploring some GitHub Actions:**

First step is to creating the account.

**Creating GitHub Account:** Begin your GitHub Actions journey by setting up a GitHub account, if you don't have one already. This account will be your central hub for managing your code.

**Establishing a Code Repository:** Inside your GitHub account, create a new repository to store your code. This repository will be where GitHub Actions performs its magic.

**Next step is to set up the Actions.**

Now Setting Up GitHub Actions:

Once your repository is ready, follow these steps to configure GitHub Actions:

**Accessing Your Repository:** Navigate to the repository you've just created.

**Managing and creating a Workflow:** Within your repository, go to the "Actions" tab, and then select the "Create a new workflow" option. From here you can define where to perform the tasks in GitHub, basically it manages everything overall in your repository.

**Choosing any Template:** GitHub Actions offers pre-designed templates to start your workflows. You can either select a template that aligns with your needs or create a customized workflow from scratch.

**Now we can jump on to the work flow as we have chosen template for managing our work.**

Customizing our Workflow: Now, it's time to map our workflow to match our specific requirements:

**Editing the Workflow:** You'll find a file that you can easily edit. This file serves as a set of instructions for GitHub Actions, providing you with the tasks it should perform in response to particular events.

**Defining Your Workflow:** it defines when actions should run, which task should you carry out, where your notifications and alert are sent.

**Saving Your Workflow:** Don't forget to save your changes to activate them.

**Working of GitHub Actions:**

Now let’s get towards how it basically works in general:

1. **Watching Your Code**: GitHub Actions is like a digital watch for your code. It keeps a close eye on what's happening.
2. **Noticing Important Events**: When something important, like changes to your code, happens, GitHub Actions wakes up. Think of it as being alerted as alarm when something interesting occurs.
3. **Making a To-Do List**: You tell GitHub Actions what to do when these interesting things happen. This is like making a simple to-do list for it generally known as workflow in GitHub Actions.
4. **Doing Tasks (Actions)**: Inside the to-do list, you specify exactly what GitHub Actions should do. These tasks are called "actions." It's like telling it the steps to follow.
5. **Automation Flow**: GitHub Actions now automatically follows the steps you've written, just like a robot or AI following instructions.
6. **Reporting**: After finishing its tasks, GitHub Actions gives you a report card. It tells you if everything went well or if there were any problems. It's like getting feedback to make sure your code is okay.

Top of Form