**Experiment no 8**

**AIM:** To study and implement deployment of Ecommerce PWA to GitHub pages.

**THEORY:**

*Continuous deployment* (CD) is the practice of using automation to publish and deploy software updates. As part of the typical CD process, the code is automatically built and tested before deployment.

Continuous deployment is often coupled with continuous integration.

You can set up a GitHub Actions workflow to deploy your software product. To verify that your product works as expected, your workflow can build the code in your repository and run your tests before deploying.

You can configure your CD workflow to run when a GitHub event occurs (for example, when new code is pushed to the default branch of your repository), on a set schedule, manually, or when an external event occurs using the repository dispatch webhook. GitHub Actions provides features that give you more control over deployments. For example, you can use environments to require approval for a job to proceed, restrict which branches can trigger a workflow, or limit access to secrets. You can use concurrency to limit your CD pipeline to a maximum of one in-progress deployment and one pending deployment.

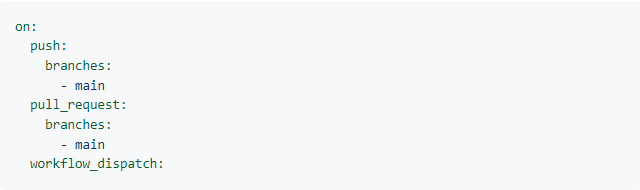
**OUTPUT:**

* **Triggering your deployment**

You can use a variety of events to trigger your deployment workflow. Some of the most common are: pull\_request, push, and workflow\_dispatch.

For example, a workflow with the following triggers runs whenever:

* There is a push to the main branch.
* A pull request targeting the main branch is opened, synchronized, or reopened.
* Someone manually triggers it.



## Using concurrency

Concurrency ensures that only a single job or workflow using the same concurrency group will run at a time. You can use concurrency so that an environment has a maximum of one deployment in progress and one deployment pending at a time.

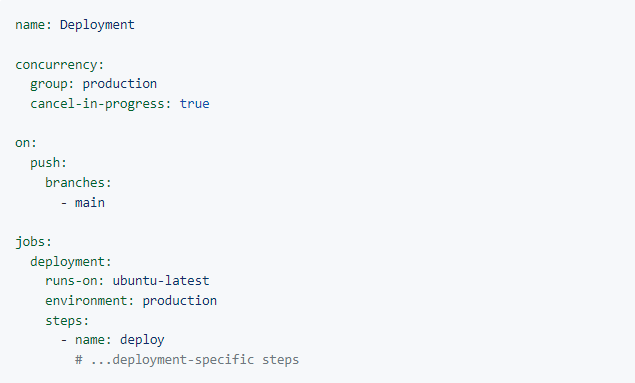
For example, when the following workflow runs, it will be paused with the status pending if any job or workflow that uses the production concurrency group is in progress. It will also cancel any job or workflow that uses the production concurrency group and has the status pending. This means that there will be a maximum of one running and one pending job or workflow in that uses the production concurrency group.



You can also specify concurrency at the job level. This will allow other jobs in the workflow to proceed even if the concurrent job is pending.



You can also use cancel-in-progress to cancel any currently running job or workflow in the same concurrency group.



**CONCLUSION:** Thus, we have studied and implemented deployment on GitHub pages.