Github Actions — CI/CD Pipelines

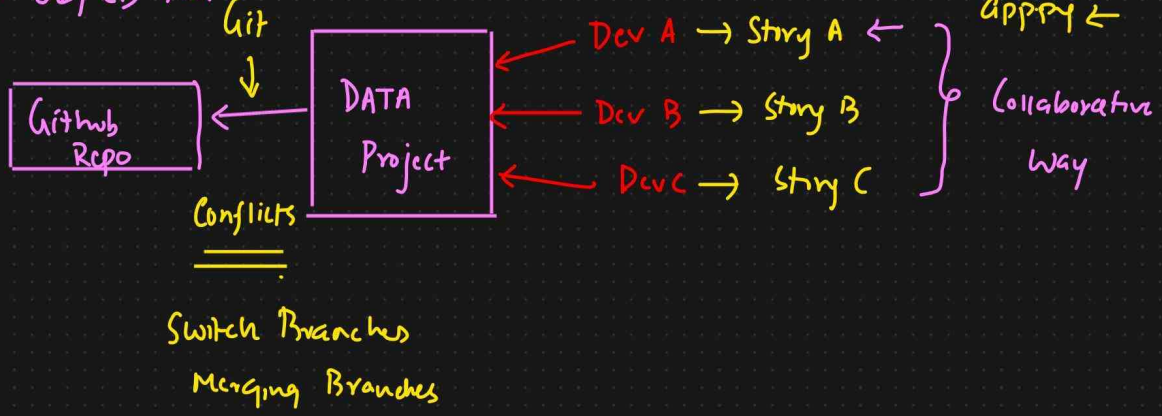
GitHub Actions is a powerful automation tool provided by GitHub that enables developers to automate tasks related to their software development workflows. It allows for the creation of custom workflows, which can be triggered by various events, such as pushes to a repository, pull requests, or scheduled events. These workflows can be used for various purposes, including Continuous Integration (CI), Continuous Deployment (CD), testing, and more.

Github Action → CI/CD Tool
↓
Github.

Github → Code Repository → Commit Code → Collaborative

Git → Distributed version control system [Developers] → Track source code files

Github Actions → CI/CD Tool

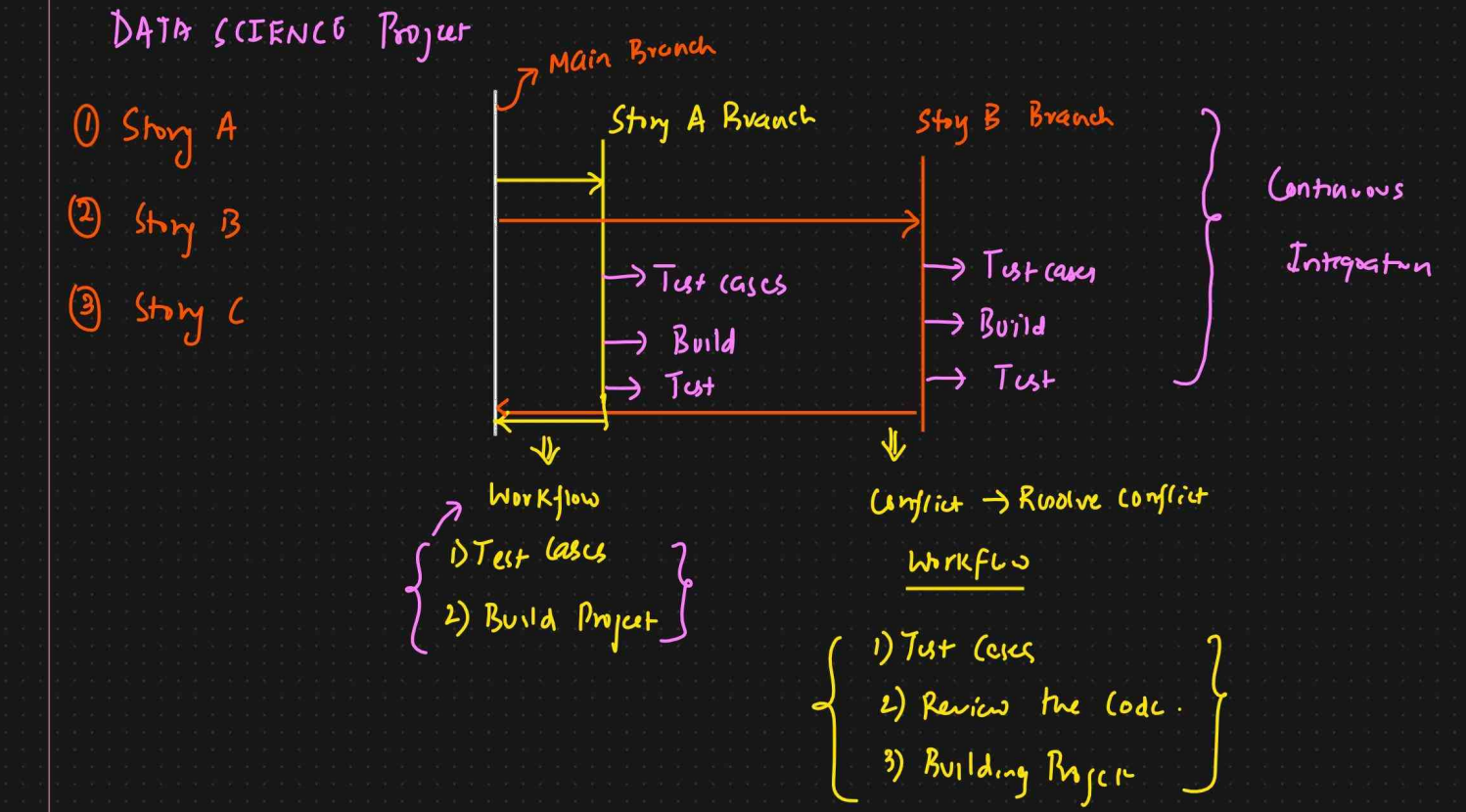


Continuous Integration And Continuous Deployment

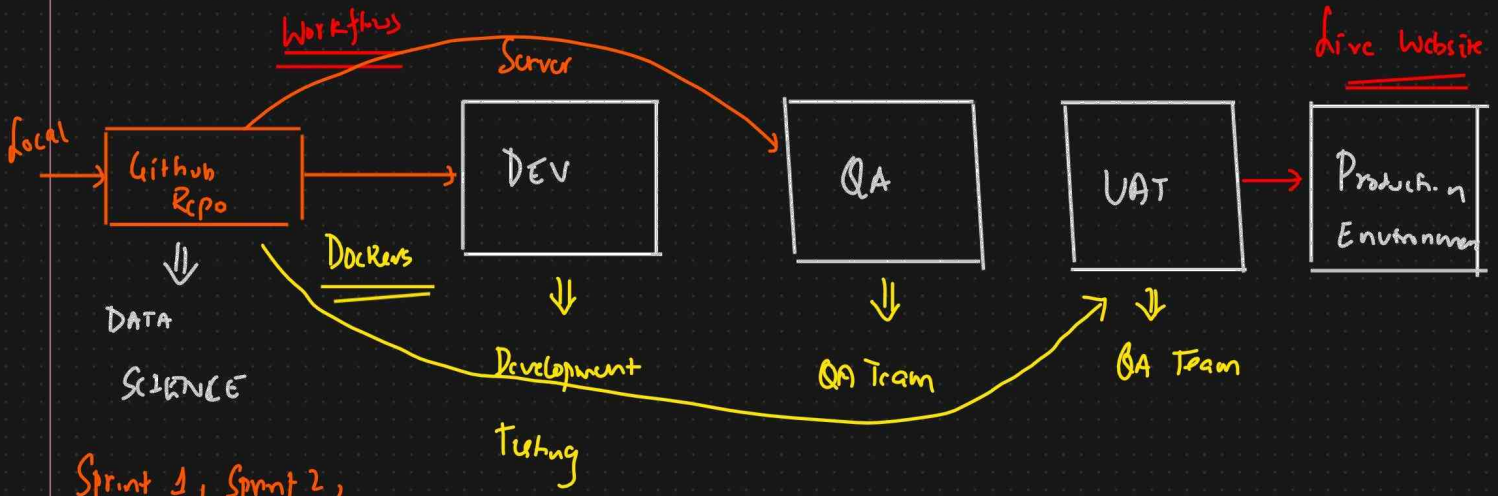
Continuous Integration (CI) and Continuous Deployment (CD) are two key practices in modern software development, and GitHub Actions can facilitate both:

Continuous Integration (CI): CI is a practice where developers frequently merge their code changes into a shared repository. Each merge triggers an automated build and testing process to ensure the changes do not break the existing functionality. With GitHub Actions, developers can set up workflows to automatically build and test their code every time they push changes to the repository or create a pull request.

Continuous Deployment (CD): CD extends the concept of CI by automating the deployment of code to production environments after it passes all required tests. GitHub Actions can be configured to automatically deploy applications to various environments, such as staging or production, once the code passes CI checks. This practice reduces the time between development and deployment, allowing for faster delivery of features and updates.



② Continuous Deployment (CD) ←



2. Workflow - Developers Workflow

A developer's workflow refers to the series of steps, practices, and tools that a developer or a team of developers follows to write, test, collaborate on, and deploy code effectively. This workflow is designed to streamline the software development process, enhance productivity, reduce errors, and ensure high-quality code output. A well-defined developer's workflow incorporates various stages of development, from coding to deployment, and often integrates tools for version control, code review, testing, and continuous integration/continuous deployment (CI/CD).

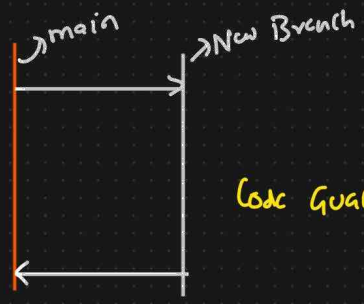
Key Stages

1) Coding → IDE (VS code) → Python, javascript → Coding standards, Best practise

2) Version Control : Git → Manage the codebase. Multiple developer to Collaborate

- Commit
- Branches
- push
- pull
- Resolve conflict

3) Code Review



Peer Code Review → Best Practices

4) Testing :

Automated Testing

1) Unit Testing
2) Integrating Testing
3) End to End Test Cases

5) Continuous Integration (CI)

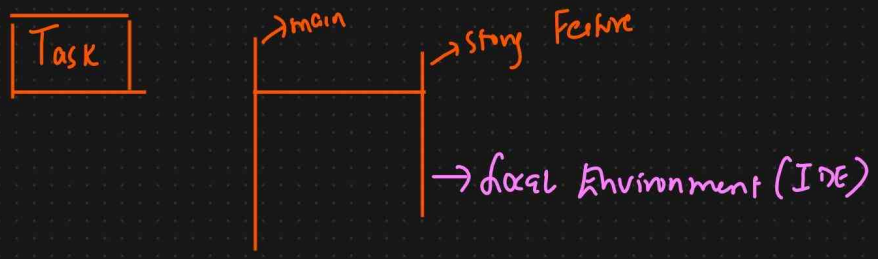
Building, Testing → Notified → Fixing the Issues

6) CD : Deploy Different Server (Dev, QA, UAT, Prod)

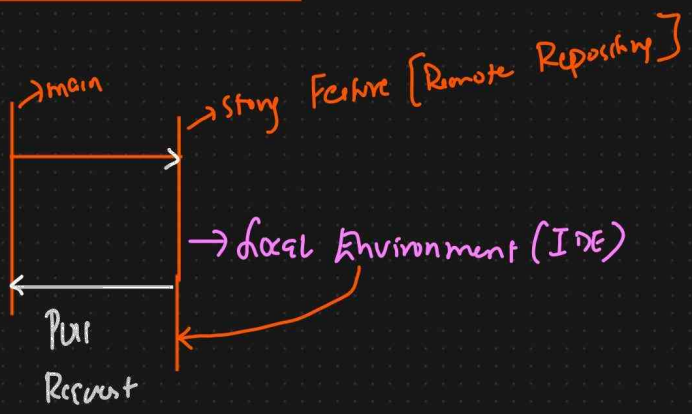
Developer Workflow

Developer A Team → Data Science Project

1) Feature Development



② Push And Pull Request (PR)



CI

Other Team member Review the PR for code quality, style and any other issue

4) Automated CI Pipeline :

PR → Event → Workflow is Triggered

1) Once the PR is opened, A CI pipeline is automatically triggered.
2) This pipeline builds the application and runs all test cases.
3) If pipeline passes, the PR is approved and merged in the main branch

5) Continuous Deployment

1) Upon merging the PR, a CD pipeline is triggered

2) The application is automatically deployed to Dev or Staging Environment.

3) Production Environment

Benefits of a Developer's Workflow

Improved Collaboration: Clear workflows and processes facilitate better collaboration among team members, making it easier to understand who is responsible for what.

Higher Code Quality: Code reviews, automated tests, and CI/CD practices help maintain a high standard of code quality.

Reduced Errors: Automated testing and deployment reduce the likelihood of human error, ensuring more reliable releases.

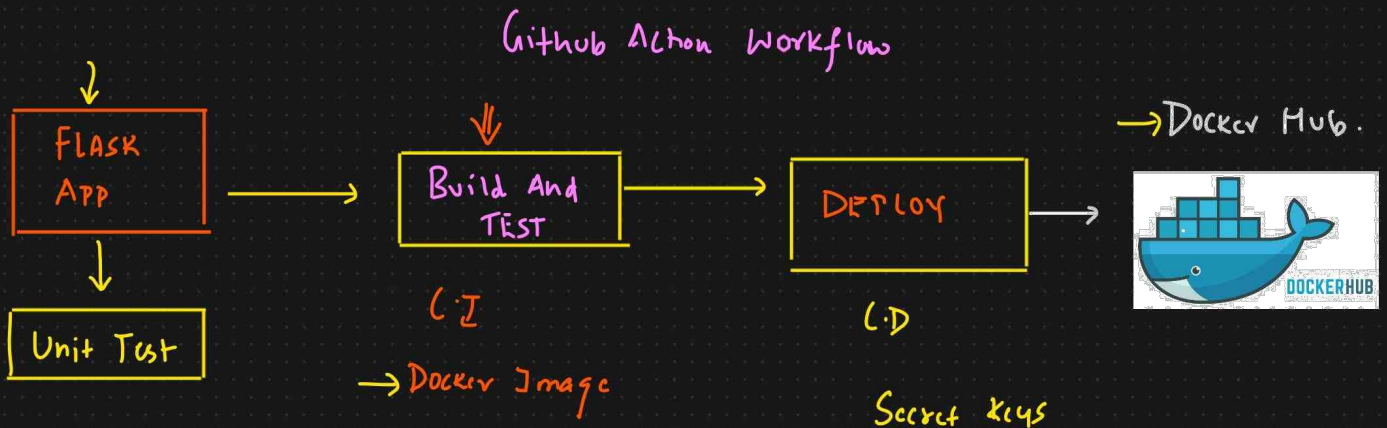
Faster Delivery: Streamlined workflows enable faster development and release cycles, allowing teams to deliver new features and fixes more quickly.

Continuous Feedback Loop: Regular monitoring and feedback help teams quickly identify and address issues, continuously improving the product.

Github Action Workflow

Project : Automate Testing for a Python Project.

Github Action Workflow → CI-CD Pipeline



1. Git
2. Github
3. Docker
4. PyTest
5. FLASK

Secret Keys

1. Docker username
2. Docker password [Token]