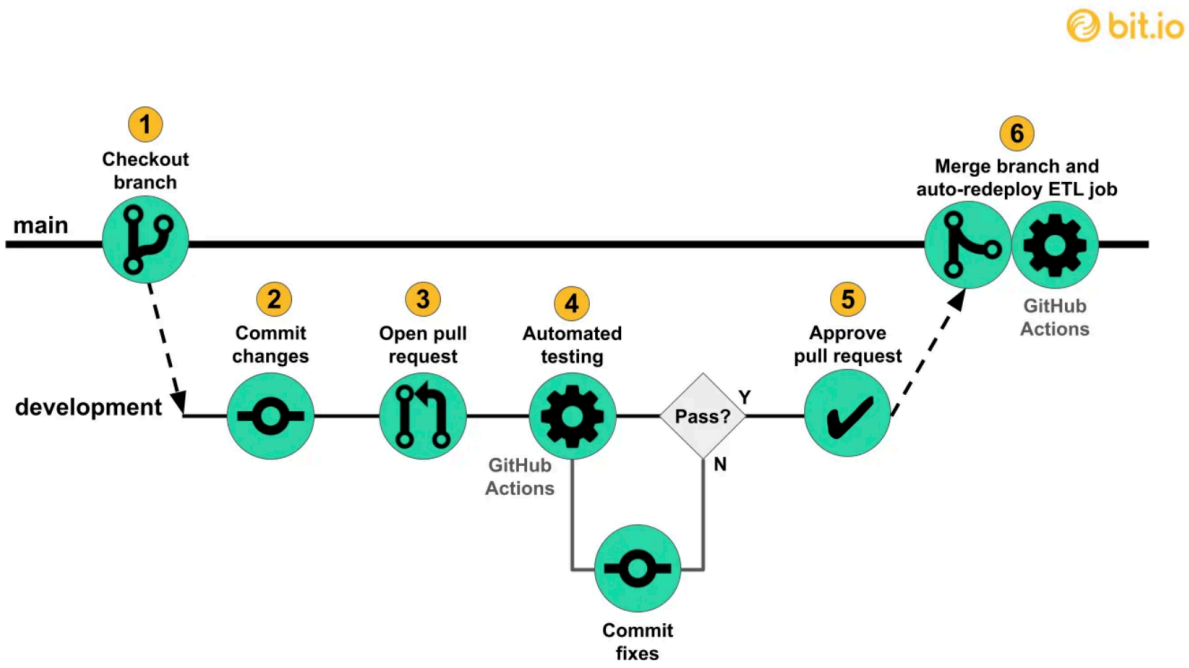
Proposed CI/CD Methodology

Platform: Github Actions, Jobs are executed by Github hosted runner (with Usage Limit)



Pipeline Stages:
(1) Developers check out main branch to start making changes
(2) Developers commit changes and create a pull request, Peer Review here
(3) Open pull request (PR), this event triggers **Github actions**
(4) Automated Testing begins with GitHub actions (Sanity Check, Smoke Test, Functional Test)
(5) Approve PR
(6) Merge branch to main, this triggers ETL job Github Actions

Continuous Integration: Build -> Test -> Artifact ->
Continuous Delivery: -> Staging -> Production
Continuous Deployment: Monitor Production release

Testing Stages
1. Unit test each component in Development Environment (each dev needs to do this before committing changes Step 2)
2. Integration testing Sanity Check, Smoke Test, Functional Test in Staging Environment in Step 4 **Need to create a testing framework*
3. Post Deploy Test in Production Environment Step 6 **Need to create a testing framework*

Creating a CI/CD pipeline:

1. Need to define workflow file (yaml) in GitHub workflows
2. Configure Coolify instance and API key in Repo Settings>Secrets
3. Configure Coolify to accept deployments via its API
4. Push yml file to repository
5. Monitor by workflow runs in Actions

Workflow yml:

- Trigger the workflow when a push is made to ‘main’ branch and when the pull request is opened against ‘main’ branch

Workflow Steps:

- Check out repo’s code so workflow can access it
- Set up Node.js [Double check: Did Rochan use Node.js for the web app?]
- Install Dependencies
- Run Tests
- Build Project
- Deploy to Coolify