

DevOps

GitHub Actions CI/CD



20F-1058

Muhammad Ibrahim

SE-7A

# **Introduction:**

This report provides my hands-on experience and analysis of implementing continuous integration and deployment using GitHub Actions workflows for a Node.js codebase. I configured GitHub Actions jobs to run install tasks on repository events.

# **Overview:**

Key components used:

* GitHub Actions workflow YAML syntax for defining jobs
* Virtual GitHub-hosted runners to execute jobs
* A Git repository containing Node.js code
* GitHub branch protections and pull requests to trigger jobs
* Implementation Details

The step-by-step process I followed:

* Cloned the Node.js repo from GitHub to my local system using git
* Changed directory into the local clone with cd
* Checked out a previous commit tag using git checkout
* Authored a GitHub workflow YAML file with a job to run yarn install
* Configured the install job to run on pushes to main and pull requests
* Pushed a commit to main which queued the job on GitHub's runners
* Verified the install job executed successfully on the commit
* Analysis

Key capabilities provided by GitHub Actions:

* Git commands can be used directly in GitHub workflow jobs
* Granular triggers based on Git branch events and pull requests
* Virtual hosted runners provide execution environments
* Workflows allow organizing jobs into sequences

Overall, I found GitHub Actions to provide a flexible way to implement CI/CD for a Node.js codebase. The virtual runners and event-driven execution make it easy to set up automation around Git activities. For more complex workflows, the full GitHub Actions feature set provides robust options.