

Github-Action

GitHub Actions is a powerful automation tool integrated into GitHub that allows developers to automate workflows directly in their repositories. With GitHub Actions, you can define workflows that are triggered by specific events, such as code pushes, pull requests, issue creation, or even scheduled events. It supports continuous integration (CI) and continuous delivery (CD) directly within GitHub, enabling you to automate testing, building, and deployment of your code.

Key Features of GitHub Actions:

1. **Event-Driven:** GitHub Actions are triggered by various GitHub events like `push`, `pull_request`, `issue`, `release`, and more. You can also schedule workflows to run at specific intervals (using cron syntax).
2. **YAML-Based Configuration:** Workflows are defined in YAML files (typically in `.github/workflows/`). These files describe the sequence of jobs, conditions for their execution, and the steps within each job.
3. **Reusable Workflows:** You can reuse workflows across repositories, making it easy to standardize processes across teams or projects.
4. **Cross-Platform Support:** GitHub Actions supports running jobs on Linux, macOS, and Windows runners, giving flexibility depending on your target environment.
5. **Integration with GitHub Ecosystem:** It's tightly integrated with the GitHub ecosystem, allowing workflows to access repository secrets, commit information, and interact with the GitHub API. GitHub Actions can also automate GitHub operations like creating issues, managing pull requests, and labeling.
6. **Marketplace:** There is a marketplace for GitHub Actions where developers can share reusable workflows, scripts, and tools with the community.

Key Components:

1. **Workflow:** A YAML file that defines the automation process. Each workflow contains one or more jobs.

- Example Workflow File:

yaml

```
name: CI Workflow
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Set up Node.js
        uses: actions/setup-node@v2
        with:
          node-version: '14'
      - run: npm install
      - run: npm test
```

2. **Jobs:** Each job is a set of steps that run on a runner (a virtual environment). Jobs can run in parallel or be dependent on other jobs.
3. **Steps:** Steps are individual tasks performed within a job. Each step can either run a shell command or use an action from the GitHub Actions marketplace.
4. **Actions:** Actions are reusable units of code that can be included as steps in workflows. They can be built and shared within the community or private to your repository.
5. **Runners:** The environment where the jobs are executed. GitHub provides hosted runners (on Ubuntu, Windows, and macOS), but you can also use self-hosted runners for more control.

Common Use Cases:

1. **Continuous Integration (CI):** Automating code builds, running tests, and checking the quality of code whenever a change is made.
2. **Continuous Deployment (CD):** Automatically deploying your application after tests pass (to services like AWS, Azure, or Google Cloud).

3. **Automating Pull Requests:** Automatically labeling, assigning reviewers, or even merging pull requests based on criteria.
4. **Package Management:** Automating the release process for packages (e.g., npm, Docker, etc.).
5. **Issue and Project Management:** Automating issue creation, triaging, or tracking.

Example Use Case (CI for Node.js Project):

yaml

```
name: Node.js CI

on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2
      - name: Use Node.js 14.x
        uses: actions/setup-node@v2
        with:
          node-version: '14'
      - run: npm install
      - run: npm run build
      - run: npm test
```

Implementation

1. git clone url to make a copy of all the file which i want in my local machine.
2. All the files are available in the local machine.
3. Make the folder as .github/workflows.
cd .github/workflows/
4. Made a file integration.yml

```
jobs:
  unit-test:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: [18.x, 20.x, 22.x]
        # See supported Node.js release schedule at https://nodejs.org/en/about/releases/
    steps:
      - uses: actions/checkout@v4
      - name: Use Node.js ${{ matrix.node-version }}
        uses: actions/setup-node@v4
        with:
          node-version: ${{ matrix.node-version }}
          cache: 'npm'
      - run: npm i
      - run: npm run test

# this is new build
```

We have created the jobs as build and for test.

5. Git add . (to make all file to stage area)
6. Git commit -m "Integration work"
7. Git push
8. Now go in action tab in the github and check the build and test.

These are the build and test in workflow that are going to build and test:

← Integration

✓ Integration WorkFlow with test #5

Summary

Jobs

- ✓ build (18.x)
- ✓ build (20.x)
- ✓ build (22.x)
- ✓ unit-test (18.x)
- ✓ unit-test (20.x)
- ✓ unit-test (22.x)

Run details

- Usage
- Workflow file

Triggered via push 16 hours ago

👤 RahulDubey-Devops pushed -o- 0f7f5b5 main

Status	Total duration	Artifacts
Success	46s	—

integration.yml

on: push

Matrix: build

✓ build (18.x)	36s
✓ build (20.x)	33s
✓ build (22.x)	33s

Matrix: unit-test

✓ unit-test (18.x)	20s
✓ unit-test (20.x)	20s
✓ unit-test (22.x)	17s

Linters Section is used to check the structure of the code:

```
lint:
  name: Linter
  runs-on: ubuntu-latest
  strategy:
    matrix:
      node-version: [18.x, 20.x, 22.x]
  steps:
    - name: Checkout code
      uses: actions/checkout@v2
    - name: Use Node.js ${{ matrix.node-version }}
      uses: actions/setup-node@v4
      with:
        node-version: ${{ matrix.node-version }}
    - name: Independencies
      run: npm i
    - name: Linter
      run: npx standard --fix

deploy:
  runs-on: ubuntu-latest
```

In this Job section of yaml code we implemented the scp and ssh command to connect and transfer files of code to the server:

```
deploy:
  runs-on: ubuntu-latest
  needs: ['build', 'lint'] # Wait for the build job to finish
  steps:
    - name: Checkout code
      uses: actions/checkout@v4

    - name: Setup SSH Agent
      uses: webfactory/ssh-agent@v0.5.3
      with:
        ssh-private-key: ${ secrets.SSH_PRIVATE_KEY }

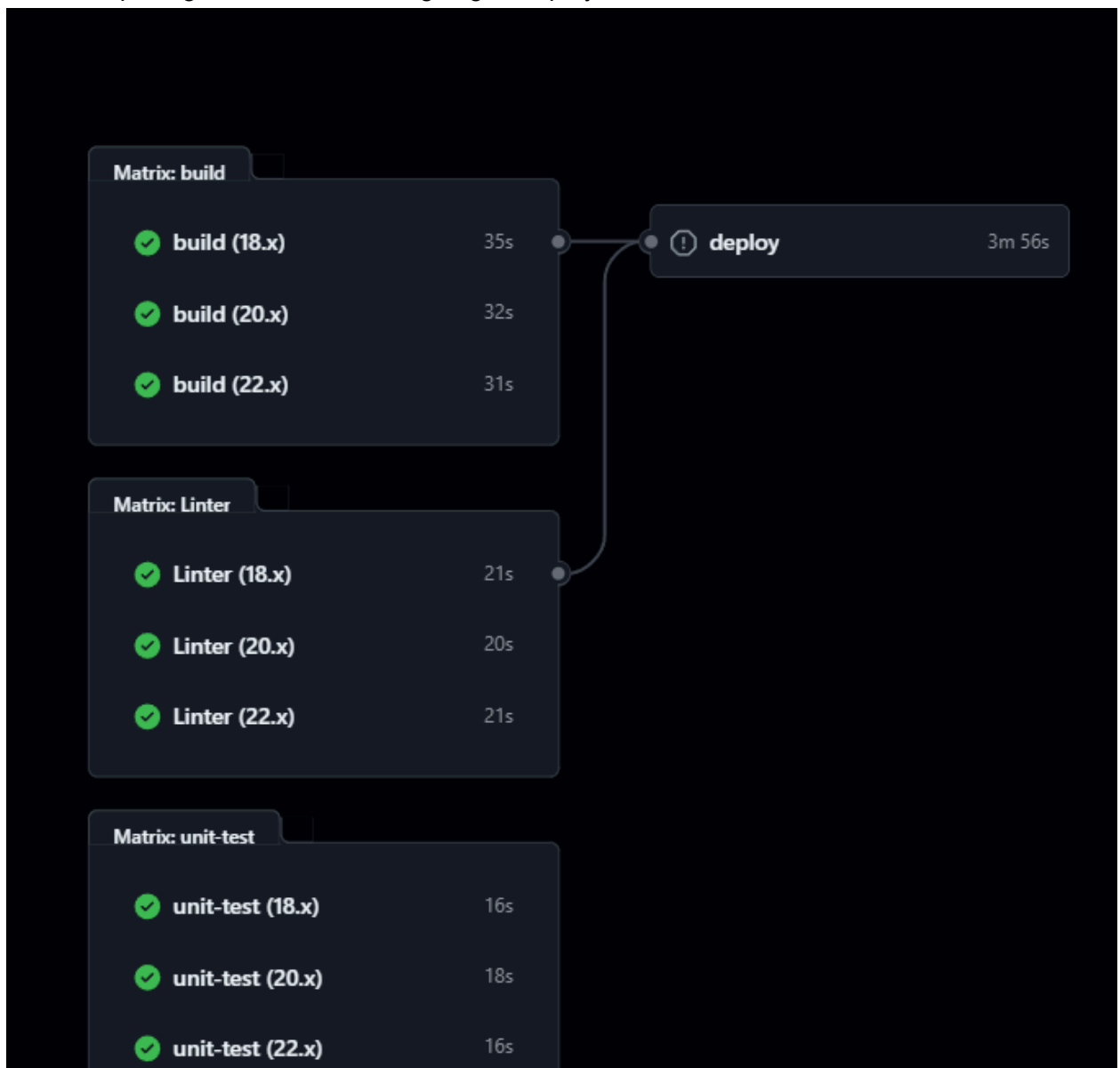
    - name: Transfer files to VM
      run: |
        scp -r -o StrictHostKeyChecking=no * ubuntu@${ secrets.VM_IP }:/home/ubuntu/apps

    - name: Connect to VM and deploy
      run: |
        ssh -tt -o StrictHostKeyChecking=no ubuntu@${ secrets.VM_IP } << 'EOF'
        cd /home/ubuntu/apps
        npm install
        npm run build
        pm2 restart all || pm2 start npm --name "my-app" -- run start
        EOF
```

These are Successfully build and now going for testing :

The screenshot displays the GitHub Actions interface for a workflow named 'integration.yml'. The top bar indicates the workflow was triggered by a push to the 'main' branch by user 'RahulDubey-Devops' and is currently in a 'Cancelled' state, having run for 4 minutes and 46 seconds. The left sidebar lists the jobs in the workflow: 'build' (18.x, 20.x, 22.x), 'unit-test' (18.x, 20.x, 22.x), 'Linters' (18.x, 20.x, 22.x), and 'deploy'. The main panel shows a graph of the workflow steps. The 'Matrix: build' step is completed with 3 jobs. The 'Matrix: Linter' step is also completed with 3 jobs. The 'Matrix: unit-test' step is completed with 3 jobs. The 'deploy' step is currently in progress, indicated by a warning icon and a duration of 3m 56s. The workflow file 'integration.yml' is shown with the trigger 'on: push'.

After Completing all test the code is going to deploy on server:



The code is deployed on the remote server you mentioned in deploy Section:

The screenshot displays the GitHub Actions workflow run interface. On the left sidebar, the 'Summary' tab is active, showing a list of jobs: 'build (18.x)', 'build (20.x)', 'build (22.x)', 'unit-test (18.x)', 'unit-test (20.x)', 'unit-test (22.x)', 'Linter (18.x)', 'Linter (20.x)', and 'Linter (22.x)', all of which are marked as successful with green checkmarks. The 'deploy' job is highlighted with a blue bar and a warning icon, indicating it failed. Below the job list, there are links for 'Run details', 'Usage', and 'Workflow file'.

The main content area shows the 'Annotations' section with '2 errors'. Below this, the 'deploy' job details are shown, indicating it was 'cancelled yesterday in 3m 56s'. A list of steps follows:

- > ✓ Set up job
- > ✓ Checkout code
- > ✓ Setup SSH Agent
- > ✓ Transfer files to VM
- > ⚠ Connect to VM and deploy
- > ✓ Post Setup SSH Agent
- > ✓ Post Checkout code
- > ✓ Complete job