

NAME : UDDHAV BAPU GUND | DESIGNATION: JR.AWS DEVELOPER

TASK : CI/CD Hello World

- Use GitHub Actions / Jenkins to build and test a small app.
- Trigger pipeline on git push.

Task Objective:

Implement a simple CI/CD pipeline using GitHub Actions:

- Build and test a small Node.js app.
- Trigger workflow automatically on git push.

Step 1: Create Project Folder

Commands:

```
mkdir ci-hello-world
```

```
cd ci-hello-world
```

```
PS D:\devops_docker> mkdir ci-hello-world

Directory: D:\devops_docker

Mode                LastWriteTime     Length Name
----                -- -- -- -- -- -- -- -- --
d----- 13-09-2025      12:10          0 ci-hello-world

PS D:\devops_docker> cd .\ci-hello-world\
```

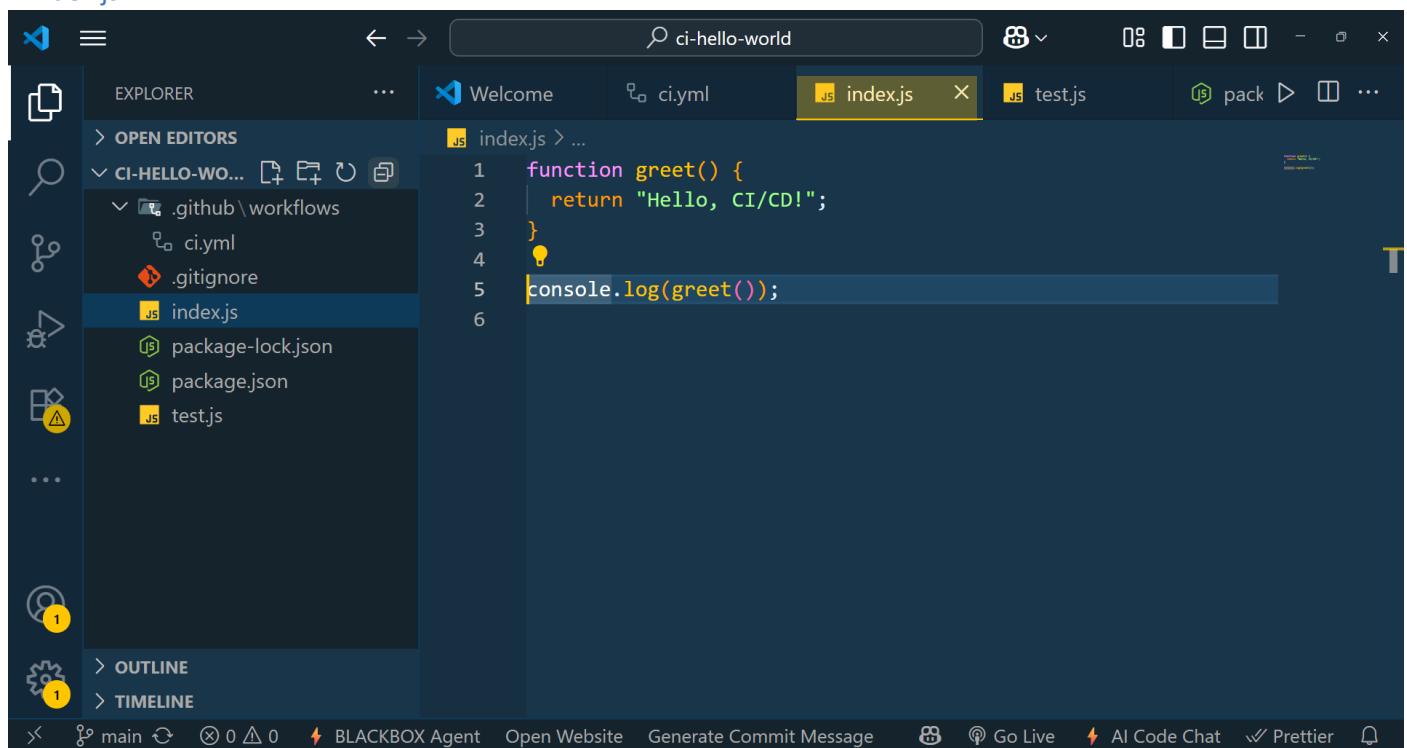
Step 2: Initialize Node.js Project

Commands:

```
npm init -y
```

Create files:

```
# index.js
```



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - CI-HELLO-WORLD folder
 - .github\workflows
 - ci.yml
 - .gitignore
 - index.js
 - package-lock.json
 - package.json
 - test.js
- WELCOME:** Shows the file list: ci.yml, index.js (selected), test.js.
- EDITOR:** Displays the content of index.js:

```
1 function greet() {
2   return "Hello, CI/CD!";
3 }
4
5 console.log(greet());
```
- STATUS BAR:** Shows the current file (index.js), line count (1), and other status indicators like BLACKBOX Agent, Open Website, Generate Commit Message, Go Live, AI Code Chat, Prettier, and a refresh icon.

test.js

The screenshot shows the VS Code interface with the title bar "ci-hello-world". The Explorer sidebar on the left shows a project structure for "CI-HELLO-WORLD" containing ".github\workflows", "index.js", "package-lock.json", "package.json", and "test.js". The "test.js" file is selected and open in the main editor area. The code in "test.js" is:

```
1 function greet() {
2     return "Hello, CI/CD!";
3 }
4
5 if (greet() === "Hello, CI/CD!") {
6     console.log("✅ All tests passed");
7     process.exit(0);
8 } else {
9     console.error("✗ Test failed");
10    process.exit(1);
11 }
```

The status bar at the bottom shows various extensions like BLACKBOX Agent, Go Live, AI Code Chat, and Prettier.

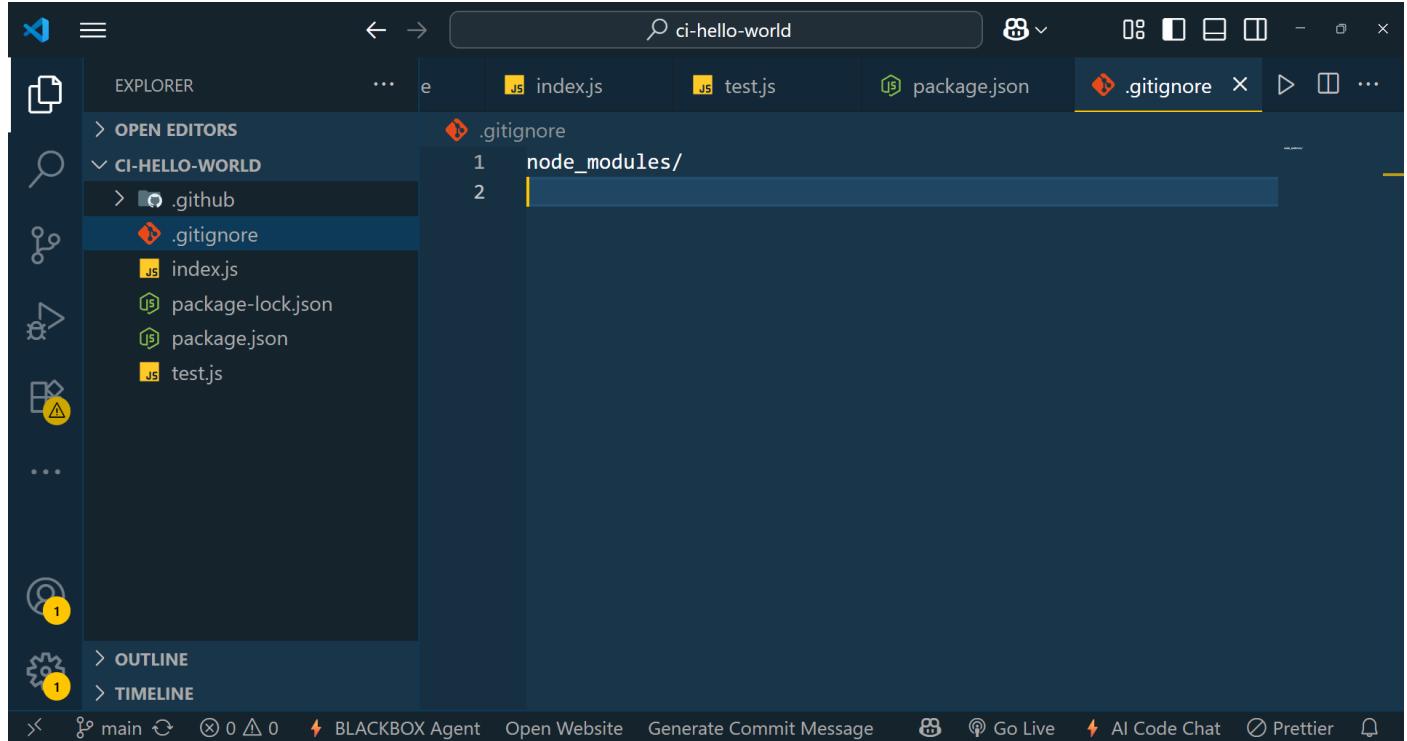
package.json

The screenshot shows the VS Code interface with the title bar "ci-hello-world". The Explorer sidebar on the left shows a project structure for "CI-HELLO-WORLD" containing ".github", ".gitignore", "index.js", "package-lock.json", "package.json", and "test.js". The "package.json" file is selected and open in the main editor area. The code in "package.json" is:

```
1 {
2     "name": "ci-hello-world",
3     "version": "1.0.0",
4     "scripts": {
5         "test": "node test.js"
6     }
7 }
```

The status bar at the bottom shows various extensions like BLACKBOX Agent, Go Live, AI Code Chat, and Prettier.

```
# .gitignore
```

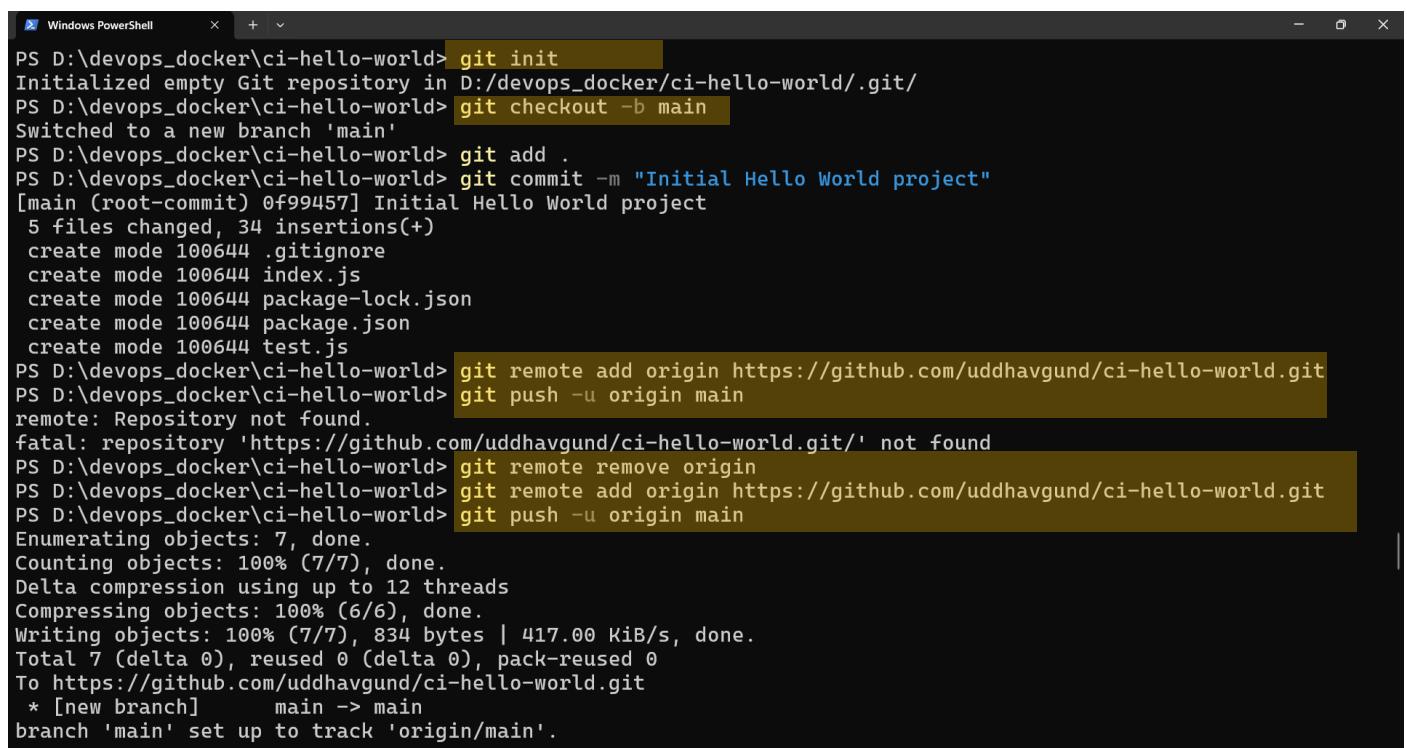


```
index.js test.js package.json .gitignore
node_modules/
```

Step 3: Initialize Git & Push Code

Commands:

```
git init
git add .
git commit -m "Initial commit - Hello World app with greet function"
git branch -M main
git remote add origin https://github.com/uddhavgund/ci-hello-world.git
git push -u origin main
```



```
PS D:\devops_docker\ci-hello-world> git init
Initialized empty Git repository in D:/devops_docker/ci-hello-world/.git/
PS D:\devops_docker\ci-hello-world> git checkout -b main
Switched to a new branch 'main'
PS D:\devops_docker\ci-hello-world> git add .
PS D:\devops_docker\ci-hello-world> git commit -m "Initial Hello World project"
[main (root-commit) 0f99457] Initial Hello World project
 5 files changed, 34 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 index.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 test.js
PS D:\devops_docker\ci-hello-world> git remote add origin https://github.com/uddhavgund/ci-hello-world.git
PS D:\devops_docker\ci-hello-world> git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/uddhavgund/ci-hello-world.git/' not found
PS D:\devops_docker\ci-hello-world> git remote remove origin
PS D:\devops_docker\ci-hello-world> git remote add origin https://github.com/uddhavgund/ci-hello-world.git
PS D:\devops_docker\ci-hello-world> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 834 bytes | 417.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/uddhavgund/ci-hello-world.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

ci-hello-world

main 1 Branch 0 Tags

Go to file Add file Code

uddhavgund Add GitHub Actions CI workflow 322b92f · 1 hour ago 2 Commits

.github/workflows Add GitHub Actions CI workflow 1 hour ago

.gitignore Initial Hello World project 1 hour ago

index.js Initial Hello World project 1 hour ago

package-lock.json Initial Hello World project 1 hour ago

package.json Initial Hello World project 1 hour ago

test.js Initial Hello World project 1 hour ago

README

Add a README

This file contains various small and big Task/Project of DevOps

Activity 0 stars 0 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages JavaScript 100.0%

Step 4: Add GitHub Actions Workflow

Create workflow file:

.github/workflows/ci.yml

ci.yml

```
name: CI
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
jobs:
  build-and-test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: '18'
      - name: Install dependencies
        run: npm install
      - name: Run tests
        run: npm test
```

Commands to commit:

```
git add .github/workflows/ci.yml
git commit -m "Add GitHub Actions CI workflow"
```

```
git push
```

```
PS D:\devops_docker\ci-hello-world> git add .github/workflows/ci.yml
PS D:\devops_docker\ci-hello-world> git commit -m "Add GitHub Actions CI workflow"
[main 322b92f] Add GitHub Actions CI workflow
 1 file changed, 24 insertions(+)
 create mode 100644 .github/workflows/ci.yml
PS D:\devops_docker\ci-hello-world> git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 596 bytes | 596.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/uddhavgund/ci-hello-world.git
 0f99457..322b92f main -> main
```

Step 5: Verify Locally

Commands:

```
node index.js      >>>Output: Hello, CI/CD!
npm test          >>>Output: ✅ All tests passed
```

```
PS D:\devops_docker\ci-hello-world> node index.js
Hello, CI/CD!
PS D:\devops_docker\ci-hello-world> npm test

> ci-hello-world@1.0.0 test
> node test.js

✅ All tests passed
```

Step 6: Verify on GitHub Actions

Go to GitHub → Actions Tab → Select the latest workflow run (named CI).

Inside, click the job build-and-test → Expand the step Run tests. → Workflow runs automatically.

The screenshot shows the GitHub Actions interface for the repository 'uddhavgund/ci-hello-world'. The left sidebar is collapsed, showing the 'Actions' tab is selected. The main area displays the 'All workflows' section, which lists a single workflow named 'CI'. Below the workflow list, there is a 'Help us improve GitHub Actions' card and a '1 workflow run' summary. The '1 workflow run' summary card is expanded, showing a single job named 'Add GitHub Actions CI workflow'. This job has a status of 'Success' and a message indicating it was triggered by a commit pushed by 'uddhavgund'. The timestamp for the run is '1 minute ago'.

The screenshot shows the GitHub Actions CI workflow summary for a repository named 'ci-hello-world'. The workflow is titled 'Add GitHub Actions CI workflow #1'. It was triggered via push 1 minute ago by the user 'uddhavgund' to the 'main' branch. The status is 'Success' with a total duration of 8s. There are no artifacts. A single job named 'build-and-test' is listed, which succeeded in 6s. The workflow file is 'ci.yml' and it triggers on pushes.

The screenshot shows the GitHub Actions CI workflow logs for the same repository and workflow. The logs for the 'build-and-test' job are displayed, showing the execution of various steps. The steps include: Set up job (0s), Run actions/checkout@v4 (1s), Setup Node.js (1s), Install dependencies (1s), and Run tests (0s). The 'Run tests' step shows the command 'npm test' being run, followed by 'All tests passed' with a green checkmark. Other steps shown are Post Setup Node.js (0s), Post Run actions/checkout@v4 (1s), and Complete job (0s).

Logs should show: All tests passed

Final Outcome:

- index.js prints Hello, CI/CD! when executed.
- test.js validates output automatically.
- GitHub Actions pipeline runs on every push.
- CI/CD pipeline successfully implemented