

Assignment 5: Creating a CI Environment

Project Description

In this assignment, I set up a Continuous Integration (CI) environment using **CircleCI** to automate the build, test, and deployment processes. The project uses a Node.js-based application with a `Makefile`, unit tests written using **Jest**, and a CI pipeline configured via `.circleci/config.yml`.

The pipeline automatically triggers on **pushes to the main branch**, builds the project, runs a suite of unit tests, and stores the deployment artifacts inside a timestamped `DEPLOY-{CURRENT_DATE}` directory.

Project Description

Prerequisites

Step-by-Step Instructions

Step 1: Connect GitHub to CircleCI

Step 2: Add and Commit Your CircleCI Configuration

Step 3: Trigger the CI Pipeline

Step 4: Observe the Build Process

Pipeline Started

Build Successful and Unit Tests Passed

Deployment Artifacts Generated

Prerequisites

Before you begin, ensure the following are ready in your GitHub repository:

- Repository is hosted on GitHub
- Project includes:
 - `Makefile`
 - `jest.config.js`
 - Unit tests inside `src/tests/unit.test.js`

MakeFile

```
# Install dependencies
node_modules:
  npm install && npx browserslist@latest --update-db && npm install --save-dev

# Build the project using npm
build: node_modules
  npm run build

# Clean the build artifacts (only the build directory)
clean:
  rm -rf build
```

jest.config.js

```
module.exports = {
  reporters: [
    'default',
    ['jest-junit', {
      outputDirectory: './test-results',
      outputName: 'junit.xml'
    }]
  ],
  testMatch: [
    '**/tests/**/*.{js,jsx,ts,tsx}', // Look for test files inside the "tests" directory
    '**/src/**/*.__tests__/**/*.{js,jsx,ts,tsx}', // Keep the src directory test pattern
    '**/src/**/*.{spec,test}.{js,jsx,ts,tsx}' // Keep the existing spec/test pattern
  ],
};
```

src/tests/unit.test.js

```

// Array functions
const containsElement = (arr, elem) ⇒ arr.includes(elem);
const arr = [1, 2, 3];

// Math functions
const add = (a, b) ⇒ a + b;
const subtract = (a, b) ⇒ a - b;

// String function
const greet = (name = 'User') ⇒ `Hello, ${name}!`;

// Test Suite
describe('Combined Unit Tests', () ⇒ {

  // Array tests
  test('checks if array contains element', () ⇒ {
    expect(containsElement(arr, 2)).toBe(true);
  });

  test('checks if array does not contain element', () ⇒ {
    expect(containsElement(arr, 4)).toBe(false);
  });

  // Math tests
  test('adds 1 + 2 to equal 3', () ⇒ {
    expect(add(1, 2)).toBe(3);
  });

  test('subtracts 5 - 2 to equal 3', () ⇒ {
    expect(subtract(5, 2)).toBe(3);
  });

  // String test
  test('greet user by name', () ⇒ {
    expect(greet('CircleCI')).toBe('Hello, CircleCI!');
  });
});

```

```
});  
});
```

- When you run `jest` or `npm test`, Jest reads `jest.config.js` and identifies the test files.
 - Jest finds `src/tests/unit.test.js` since it matches the patterns in `testMatch`.
 - Jest executes the test cases and:
 - Shows results in the terminal.
 - Generates a `junit.xml` report inside `./test-results/`, which can be used in CI/CD pipelines.
-

Step-by-Step Instructions

Step 1: Connect GitHub to CircleCI

- Go to <https://circleci.com> and **sign up** or **log in** using your GitHub account.
- Navigate to the **"Projects"** section on the left-hand dashboard.
- Click **"Create Project"** on the top-right.
- Choose the repository (e.g., `devops2`) and click **"Set Up Project"**.
- Name your pipeline (e.g., `test`) and click **"Start building"**.
- Choose **"Prepare config file"** to generate a starter config. CircleCI looks for this file at `.circleci/config.yml`.
- Configure the trigger:
 - Trigger name: `GitHub App Trigger`
 - Pipeline: `test`
 - Repo: `eduardobautista-devops/devops2`
 - Run on: `All pushes`

- Click **"Commit config and run"**.

Step 2: Add and Commit Your CircleCI Configuration

Create a folder `.circleci` in your project root, and inside it, add the `config.yml` file:

It will provide a build, then a unit test suite of 5 unit tests and a deployment in the directory `DEPLOY-{CURRENT_DATE}` automatically. running automatically for each code push,

This operation will be triggered by any code push in the main branch

`.circleci/config.yml`

```
version: 2
jobs:
  build:
    docker:
      - image: circleci/node:14 # Node.js Docker image
    working_directory: ~/devops
    steps:
      # 1. Checkout the repository from GitHub
      - checkout

      # 2. Install dependencies and build using the Makefile
      - run:
          name: Build
          command: |
            make build

      # 3. Run unit tests using Jest
      - run:
          name: Unit Tests
          command: |
            npx jest --testPathPattern=tests --outputFile=test-results/junit.json --runl
```

```

# 4. Store test results
- store_test_results:
  path: test-results

# 5. Create a deployment directory and move the build artifacts there
- run:
  name: Deployment
  command: |
    DATE=$(date '+%Y%m%d')
    DEPLOY_DIR=DEPLOY-${DATE}
    mkdir ${DEPLOY_DIR} # Create the deployment directory
    mkdir artifacts # Create the artifacts directory

    # Move the build directory to the deployment directory
    if [ -d build ]; then
      mv build/* ${DEPLOY_DIR}/
      echo -e "Build artifacts moved to ${DEPLOY_DIR}\n"
    else
      echo "Build directory not found!"
      exit 1
    fi

    echo "Contents of the deployment directory:"
    ls -la ${DEPLOY_DIR}

    # Copy the contents of the deployment directory to artifacts directory
    cp -r ${DEPLOY_DIR}/* artifacts/
    # echo "Artifacts copied to artifacts directory"

# 6. Store the deployment artifacts
- store_artifacts:
  path: artifacts

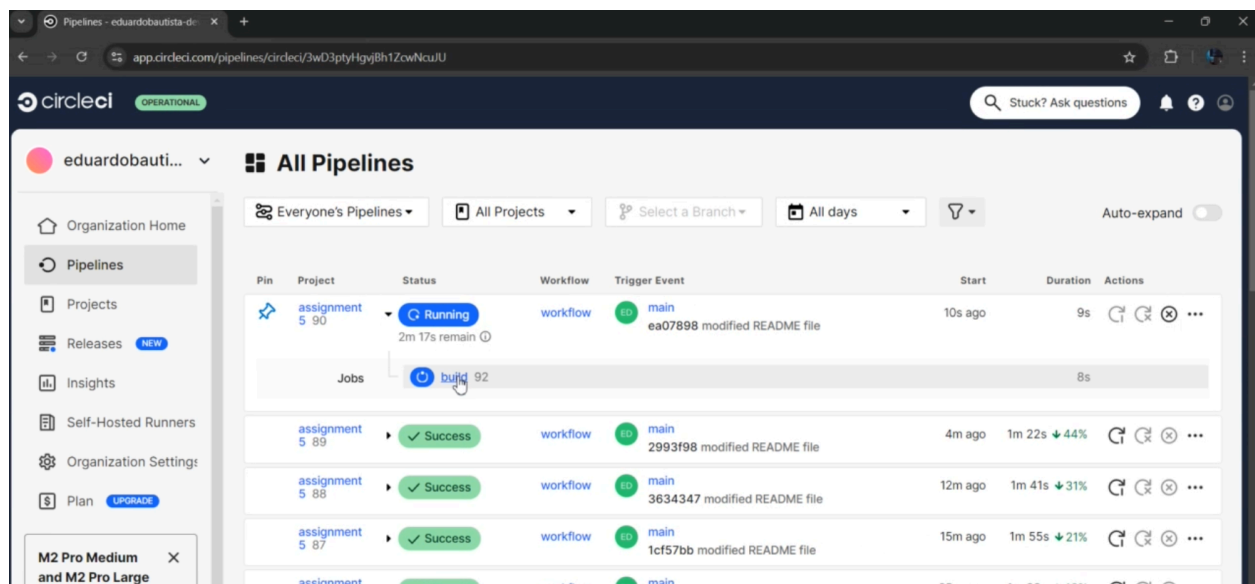
```

Step 3: Trigger the CI Pipeline

1. Confirm you're on the `main` branch: `git branch`
2. Make a small change (e.g., update `README.md`) and commit it:
`git commit -am "modify README file" && git push origin main`
3. Go to CircleCI and observe the pipeline trigger automatically.

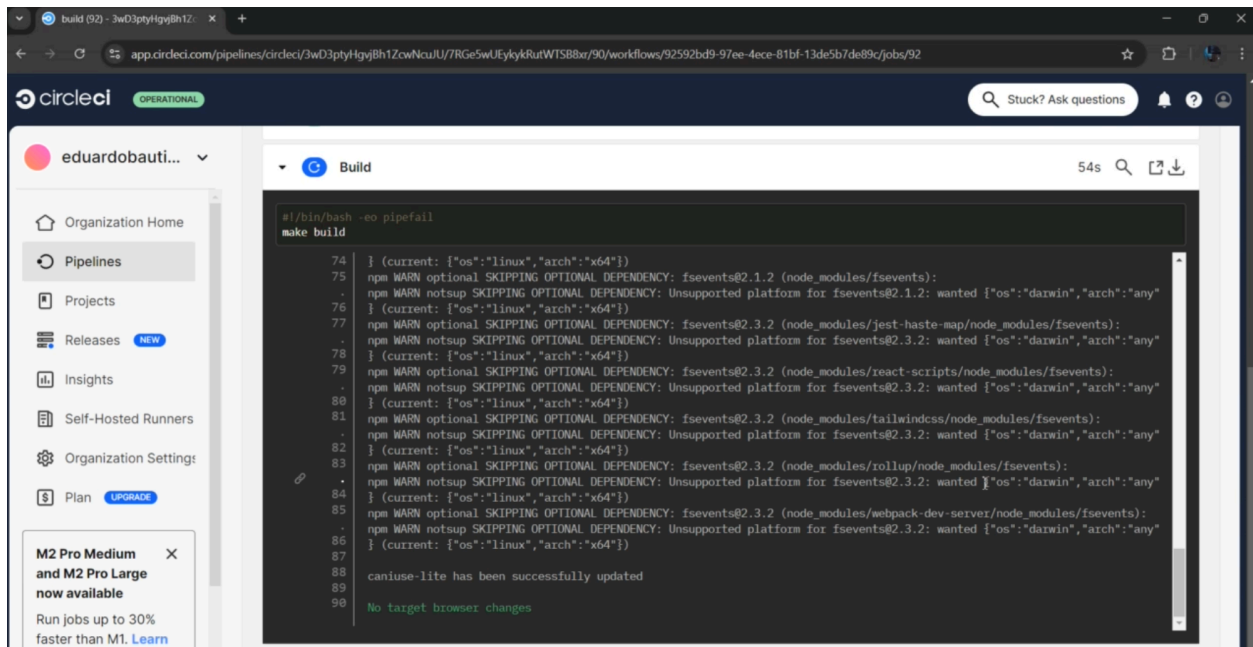
Step 4: Observe the Build Process

Pipeline Started

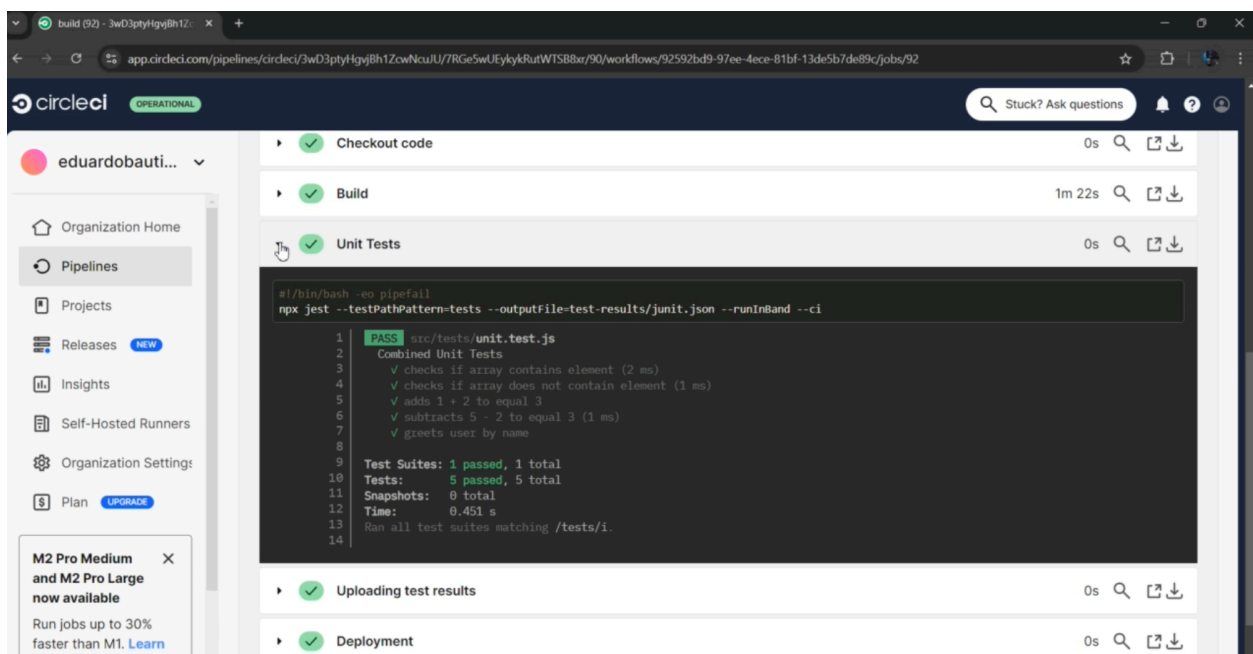


The screenshot shows the CircleCI web interface for the organization 'eduardobautista...'. The 'All Pipelines' view is active, displaying a list of pipeline runs. The top row shows a pipeline in a 'Running' state, triggered by a 'main' branch event (commit ea07898) modifying the README file. Below this, a 'Jobs' section shows a 'build' job in progress. The bottom section lists several previous pipeline runs, all of which completed successfully ('Success' status).

Pin	Project	Status	Workflow	Trigger Event	Start	Duration	Actions
assignment 5 90		Running	workflow	main ea07898 modified README file	10s ago	9s	🔄 🗑️ ⋮
Jobs							
		🔄	build 92			8s	
assignment 5 89		Success	workflow	main 2993f98 modified README file	4m ago	1m 22s ↓ 44%	🔄 🗑️ ⋮
assignment 5 88		Success	workflow	main 3634347 modified README file	12m ago	1m 41s ↓ 31%	🔄 🗑️ ⋮
assignment 5 87		Success	workflow	main 1cf57bb modified README file	15m ago	1m 55s ↓ 21%	🔄 🗑️ ⋮
assignment		Success	workflow	main	25m ago	1m 20s ↓ 10%	🔄 🗑️ ⋮



Build Successful and Unit Tests Passed



Deployment Artifacts Generated

build (92) - 3wD3ptyHgYjBh1Z...

app.circleci.com/pipelines/circleci/3wD3ptyHgYjBh1ZcwNcuU/7RGe5wUEykykRutWTS88x/90/workflows/92592bd9-97ee-4ece-81bf-13de5b7de89c/jobs/92

circleci OPERATIONAL

Stuck? Ask questions

eduardobauti...

Organization Home

Pipelines

Projects

Releases **NEW**

Insights

Self-Hosted Runners

Organization Settings

Plan **UPGRADE**

M2 Pro Medium and M2 Pro Large now available

Run jobs up to 30% faster than M1. [Learn about M2 Pro.](#)

Parallel runs

0 01:25 Use parallelism to run faster tests
Parallelism speeds up tests by splitting them across multiple executors. [Go to Docs](#) X

Accessible step output

Step	Duration	Search	Download
Spin up environment	0s	🔍	📄
Preparing environment variables	0s	🔍	📄
Checkout code	0s	🔍	📄
Build	1m 22s	🔍	📄
Unit Tests	0s	🔍	📄
Uploading test results	0s	🔍	📄
Deployment	0s	🔍	📄
Uploading artifacts	0s	🔍	📄

build (92) - 3wD3ptyHgYjBh1Z...

app.circleci.com/pipelines/circleci/3wD3ptyHgYjBh1ZcwNcuU/7RGe5wUEykykRutWTS88x/90/workflows/92592bd9-97ee-4ece-81bf-13de5b7de89c/jobs/92/artifacts

circleci OPERATIONAL

Stuck? Ask questions

eduardobauti...

Organization Home

Pipelines

Projects

Releases **NEW**

Insights

Self-Hosted Runners

Organization Settings

Plan **UPGRADE**

M2 Pro Medium and M2 Pro Large now available

Run jobs up to 30% faster than M1. [Learn about M2 Pro.](#)

STEPS TESTS **✓ 5** TIMING **ARTIFACTS** RESOURCES

Parallel Run 0

- [artifacts/asset-manifest.json](#)
- [artifacts/gamepad.ico](#)
- [artifacts/gamepad.png](#)
- [artifacts/index.html](#)
- [artifacts/logo192.png](#)
- [artifacts/logo512.png](#)
- [artifacts/manifest.json](#)
- [artifacts/robots.txt](#)
- [artifacts/static/css/main.a620d49d.css](#)
- [artifacts/static/css/main.a620d49d.css.map](#)
- [artifacts/static/js/main.cab8bf94.js](#)
- [artifacts/static/js/main.cab8bf94.js.LICENSE.txt](#)
- [artifacts/static/js/main.cab8bf94.js.map](#)
- [artifacts/static/media/1.97ae75576d7a2f5e9933.svg](#)
- [artifacts/static/media/10.d31439c8a2e44ded4978.svg](#)
- [artifacts/static/media/11.6fe6b4cd38f155f96ecc.svg](#)
- [artifacts/static/media/13.600e75576d7a2f5e9933.svg](#)