**Step 1: Install Cypress**

If you haven’t installed Cypress yet, install it globally or in your project directory.

npm install cypress --save-dev

After installation, open Cypress using:

npx cypress open

**Step 2: Install Allure Reporter and Dependencies**

To generate Allure reports, you need the **Allure Cypress plugin** and the **Allure Command-Line Tool**.

Run the following command:

npm install -D @shelex/cypress-allure-plugin

Then, install the **Allure command-line tool**:

npm install -g allure-commandline

To verify the installation, run:

allure --version

If the command doesn’t work, try:

npx allure --version

**Step 3: Configure Allure Plugin in Cypress**

Edit cypress.config.js (for Cypress v10+) or cypress/plugins/index.js (for Cypress v9 and below).

**For Cypress v10+ (Recommended)**

Modify cypress.config.js:

const allureWriter = require('@shelex/cypress-allure-plugin/writer');

module.exports = {

e2e: {

setupNodeEvents(on, config) {

allureWriter(on, config);

return config;

},

reporter: 'cypress-mochawesome-reporter', // Set a custom reporter if needed

reporterOptions: {

reportDir: 'allure-results',

overwrite: false,

html: false,

json: true,

},

env: {

allure: true,

},

},

};

**For Cypress v9 and Below**

Edit cypress/plugins/index.js:

const allureWriter = require('@shelex/cypress-allure-plugin/writer');

module.exports = (on, config) => {

allureWriter(on, config);

return config;

};

Modify cypress/support/e2e.js (or cypress/support/index.js in older versions):

import '@shelex/cypress-allure-plugin';

**Step 4: Running Cypress with Allure**

Run your Cypress tests with:

npx cypress run --env allure=true

After the tests complete, you will find a folder called **allure-results** in your project directory.

**Step 5: Generate and View the Allure Report**

To generate and open the report, run:

allure generate allure-results --clean -o allure-report

allure open allure-report

This will generate a **beautiful, interactive HTML report** and open it in your default browser.

**Step 6: Add Allure Reports to CI/CD (Optional)**

If you are running Cypress in a **CI/CD pipeline**, you can generate and store Allure reports using GitHub Actions, Jenkins, or GitLab CI.

Example **GitHub Actions** setup:

name: Cypress Tests with Allure

on: [push, pull\_request]

jobs:

cypress-run:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v3

- name: Install dependencies

run: npm install

- name: Run Cypress tests

run: npx cypress run --env allure=true

- name: Generate Allure Report

run: |

npm install -g allure-commandline

allure generate allure-results --clean -o allure-report

- name: Upload Allure Report

uses: actions/upload-artifact@v3

with:

name: allure-report

path: allure-report

**Step 7: Troubleshooting Common Issues**

**1. allure command not found**

Try running:

npx allure --version

or reinstall:

npm install -g allure-commandline

**2. Allure report folder is empty**

Ensure your test runs include:

npx cypress run --env allure=true

**3. Allure report is not opening**

Try:

npx allure open allure-report

**Final Summary**

✅ Installed **Cypress**  
✅ Installed **Allure Cypress Plugin**  
✅ Configured **Cypress with Allure**  
✅ Ran tests and generated **Allure reports**  
✅ Opened the report in a browser

Now, you have **Cypress Allure reporting fully integrated**! 🎉