

Day - 7

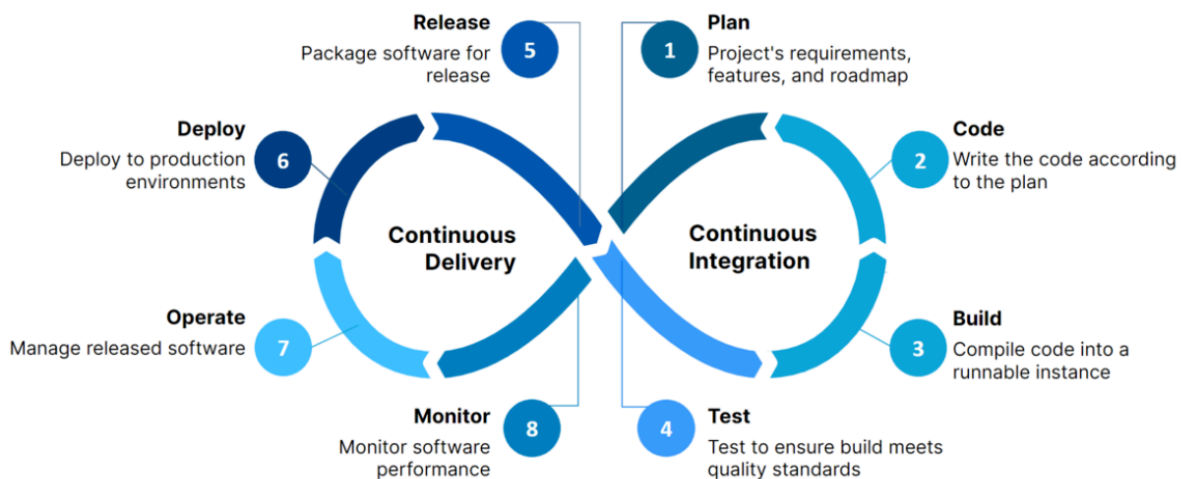
12.02.2025

CI/CD Pipeline:-

A CI/CD pipeline is a series of steps that automates the process of building, testing, and deploying software. It's a key part of the modern software supply chain.

How it works:-

- Continuous integration (CI): Builds code and runs tests
- Continuous delivery (CD): Deploys a new version of the application
- Continuous deployment: Automatically pushes code to production without human approval



Npm commands :-

npm commands

To create package.json: **npm init** | **npm init -y**

To execute scripts in package.json: **npm run <scriptName>**

To install package: **npm install <packageName>**

To install all dependencies listed in package.json: **npm install**



Test folder -> example.spec.ts

```
import { test, expect } from '@playwright/test';

test('has title', async ({ page }) => {
  await page.goto('https://playwright.dev/');

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});

test('get started link', async ({ page }) => {
  await page.goto('https://playwright.dev/');

  // Click the get started link.
  await page.getByRole('link', { name: 'Get started' }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole('heading', { name: 'Installation'
})).toBeVisible();
});
```

Test-example folder -> demo-todo-app.spec.ts

Package-lock.json ->

```
{
  "name": "auto-test",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {},
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "devDependencies": {
    "@playwright/test": "^1.50.1",
    "@types/node": "^22.13.1"
  }
}
```

Rendering Engine:-

A rendering engine is a software program that calculates the lighting and physical properties of materials. It can be used to create photorealistic visuals for games, architecture, and other fields.

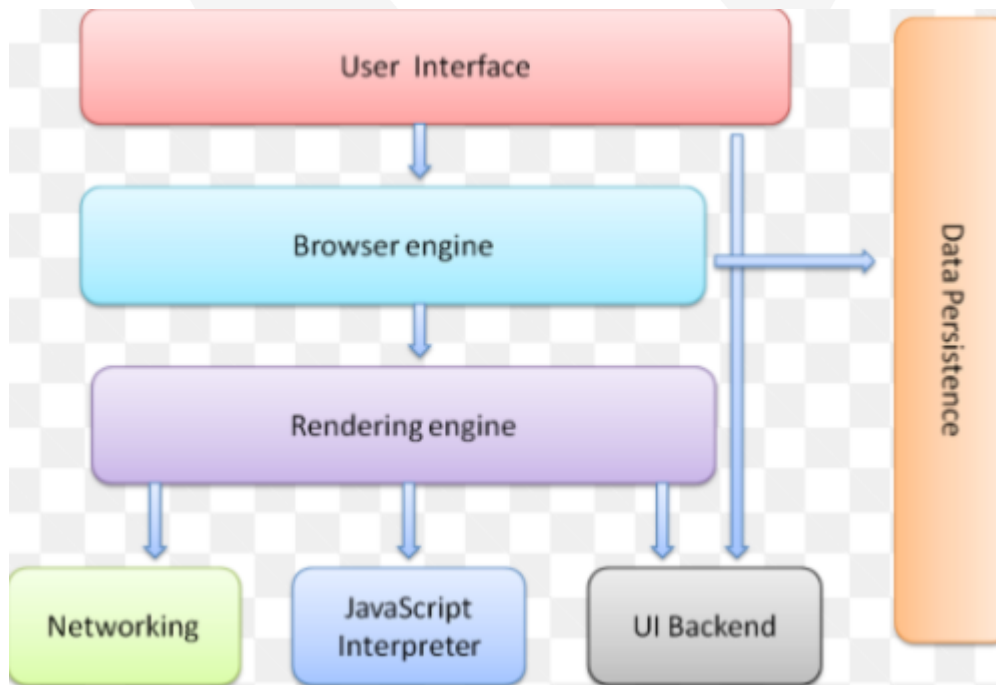
- Gecko -> open rendering engine used by firefox.
- Blink -> Chrome
- Safari -> Webkit

Playwright :-

- Cross platform
- Cross language
- One API [Application Programming Interface]

Modern browser architecture:-

A web browser is an application that serves as a gateway interface between the user and the server. It loads and compiles the results from the server in the form of an HTML web page and paints it on the screen of the user. It is an interaction tool for using the internet.



Powerful tooling in playwright:-

- **Codegen** - Generate tests by recording your actions.
- **Playwright inspector** - Inspector page, generate selector, see click.
- **Trace** - view test source , capture all info to investigate test failure.

Locators:-

- `page.getByRole()`
- `page.getByText()`
- `page.getByPlaceholder()`
- `page.getByLabel()`

Mocha - Test framework for unit and integration testing.

Commands after setup:-

`npm init playwright@latest` - install latest version

`npx playwright test` -> end to end test

`npx playwright test -- ui` -> start ui mode

`npx playwright test -- project chromium` -> run on desktop chrome

`npx playwright test example` -> specific file

`npx playwright test` -> debug mode

Output for the basic setup -

