Typical CI/CD tasks

Let's take a look at typical tasks you would have your CI/CD pipelines execute, separated by the CI and CD stage.

Continuous Integration

- **Code analysis**
    - **SCA** (Software Composition Analysis): checks the *3rd party dependencies* you use in your software for known vulnerabilities (CVEs) or available updates, e.g. Renovate Bot, dependabot, safety (Python)
    - **SAST** (Static Application Security Testing): scans *your own* code for vulnerabilities (e.g. potential stack overflows, etc.), e.g. SonarQube or PMD (cross-language), Checkstyle (Java), bandit (Python), etc. See also this list.
    - **Code style** (e.g. Linting)
- **Building / compiling** your code – in general: getting it into an *executable state*. For interpreted language, such as Python, this could mean to just install the necessary 3rd party dependencies.
- **Running tests**, of those kinds that already make sense at the CI stage, e.g.
    - Unit tests: testing individual classes
    - Component tests: testing several classes which form a *module* performing a specific function
    - Integration tests: testing that multiple (but not all) components (which are already tested individually) play well together – typically requires *mocking* of those components excluded from that specific test.

Continuous Delivery & Deployment

- **Packaging** your application (e.g. as Python wheel, Docker image, Windows setup.exe, macOS App Bundle, etc.)
- **Publishing** (=uploading) the package to some server or registry (e.g. uploading a Docker image to an OCI registry)

- Software **tests**:
    - Smoke tests: simply starting your packaged application, to see whether it crashes (test passes if it doesn’t crash).
    - System tests: testing the system’s overall functionality (including *all* of its *components*)
        - E.g. End-to-end tests that test a *sequence* of input-output-input-output…. interactions (e.g. logging into a website and clicking menu items, expecting certain pages to be returned)
    - Performance tests, e.g. load tests or stress tests (they both test how your application behaves under high load – stress test are putting extreme amounts of load on the application to see whether it crashes or recovers)
    - Installation tests, which verify that your software can be installed (and upgraded) successfully
    - Security checks with DAST (Dynamic Application Security Test) tools, such as Zed Attack Proxy (ZAP), which try different attacks against a running application instance
- **Deployment** of your packaged application in an environment