## Complete CI/DevSecOps Pipeline Steps

This sequence shows the logical progression of tasks, from environment setup through security checks and functional testing, culminating in the creation of a deployable artifact.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Step Name** | **Category** | **Status in Your YAML** | **Rationale in the CI Pipeline** |
| **1.** | **Setup Environment & Checkout** | **Preparation** | ✅ **Implemented** (as 'Checkout') | Establish the clean, isolated Python container with the PostgreSQL service, and fetch the source code. |
| **2.** | **Install Dependencies** | **Preparation** | ✅ **Implemented** | Install all base packages (requirements.txt) plus the necessary security tools (bandit, safety). |
| **3.** | **Secret Scanning** | **Security** | ❌ **Missing** | Scan repository for accidental hard-coded credentials (e.g., API keys). This is the fastest security check. |
| **4.** | **Static Code Analysis (Lint)** | **Quality** | ✅ **Implemented** (as 'Lint with flake8') | Enforce code style and catch non-security-related syntax errors and quality issues. |
| **5.** | **Static Security Testing (SAST)** | **Security** | ❌ **Missing** | Analyze *your* source code for security flaws (e.g., using **Bandit** for Python). |
| **6.** | **Dependency Scanning (SCA)** | **Security** | ❌ **Missing** | Check third-party dependencies (requirements.txt) against known vulnerability databases (e.g., using **Safety**). |
| **7.** | **Run Unit/Integration Tests** | **Validation** | ✅ **Implemented** (as 'Run unit tests with nose') | Confirm the application works correctly, including interaction with the mocked PostgreSQL database service. |
| **8.** | **Build Final Artifact** | **Build/Package** | ❌ **Missing** | Create the final runnable Docker image using the tested code. **(E.g., docker build)** |
| **9.** | **Publish Artifact** | **Release Prep** | ❌ **Missing** | Push the verified image to the Container Registry (e.g., IBM ICR), making it available for the CD process. **(E.g., docker push)** |