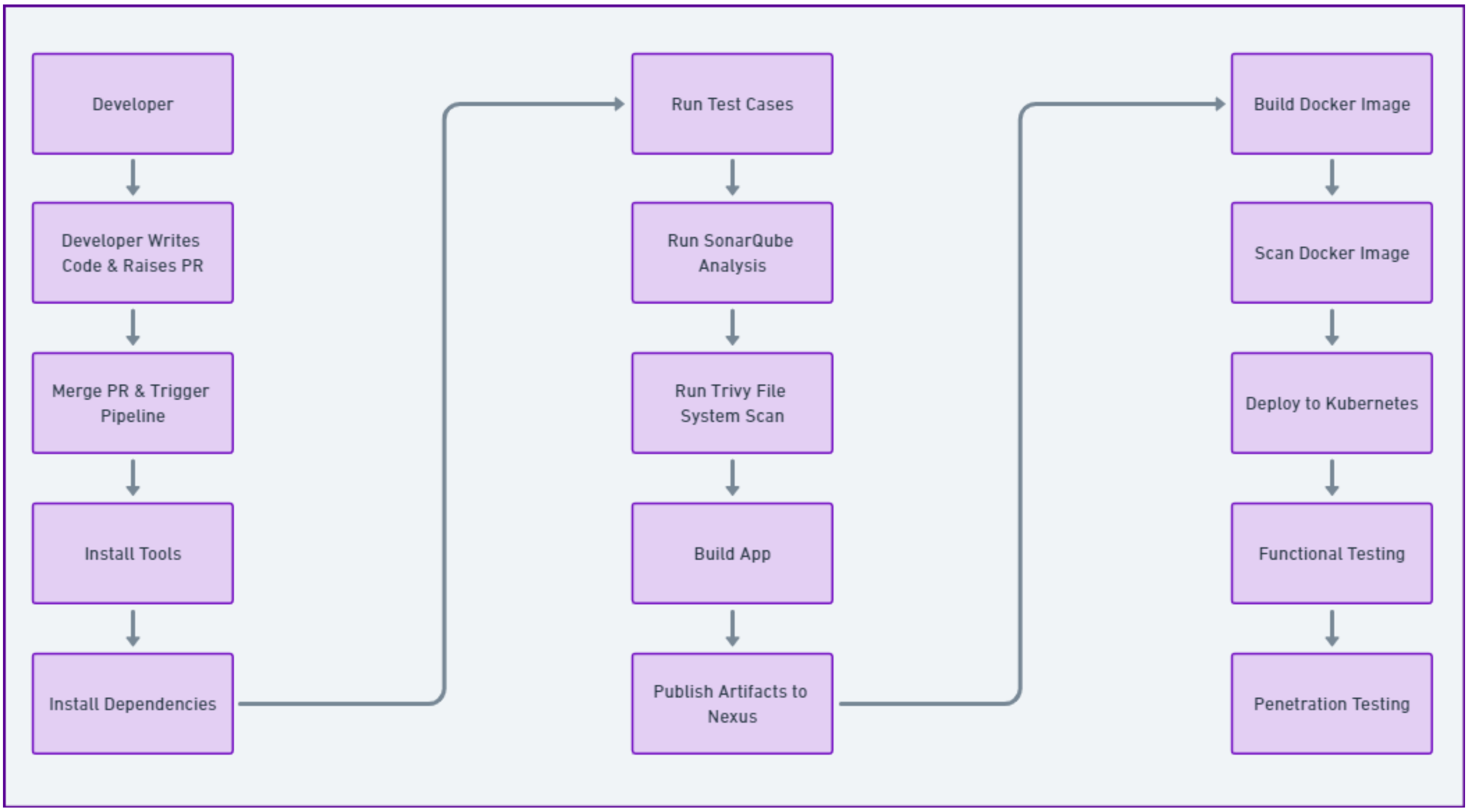
DevOps Shack

DevOps Corporate Workflow

Click Here To Enrol To Batch-6 | DevOps & Cloud DevOps



1. **Developer Writes Code & Raises PR:**
   - **Developer**: Writes new code or modifies existing code in a feature branch.
   - **Code Review**: The developer raises a Pull Request (PR) to merge their changes into a target branch (e.g., develop or main). The PR is reviewed by peers for code quality, adherence to coding standards, and potential issues.
2. **Merge & Trigger Pipeline:**
   - **Merge PR**: Once the PR is approved, it gets merged into the target branch. This merge action automatically triggers the CI/CD pipeline configured for the project.

Pipeline Stages:

3. **Install Tools:**
   - **Description**: Set up the build environment with necessary tools.
   - **Actions**:
     - Use a script or configuration file (e.g., Ansible, Chef, Puppet) to install tools like Java JDK, Node.js, Docker, Maven, etc.
     - Ensure version consistency across different environments.
4. **Install Dependencies:**
   - **Description**: Download and install all project dependencies.
   - **Actions**:
     - Use package managers such as npm for Node.js, pip for Python, or Maven for Java to install required libraries and frameworks.
     - Create a clean environment for each build to ensure no leftover dependencies affect the process.
5. **Run Test Cases:**
   - **Description**: Execute automated tests to validate the code.
   - **Actions**:
     - Unit Tests: Check individual components for correctness using frameworks like JUnit, NUnit, or Mocha.
     - Integration Tests: Validate interactions between components.
     - Code Coverage: Measure how much of the codebase is covered by tests.
6. **Run SonarQube Analysis:**
   - **Description**: Perform static code analysis for quality and security.
   - **Actions**:
     - Use SonarQube to scan the code for code smells, bugs, and vulnerabilities.
     - Generate detailed reports and ensure the code meets defined quality gates.
7. **Run Trivy File System Scan:**
   - **Description**: Scan the file system for vulnerabilities and compliance issues.
   - **Actions**:

    - Use Trivy to scan for known vulnerabilities in OS packages, application dependencies, and configuration files.
    - Review and address any identified issues before proceeding.

8. **Build App:**
    - **Description**: Compile the source code into a deployable artifact.
    - **Actions**:
        - Use build tools like Maven, Gradle, or npm to compile the code.
        - Generate artifacts such as JAR, WAR, or binary files.
9. **Publish Artifacts to Nexus:**
    - **Description**: Store the built artifacts in a repository manager.
    - **Actions**:
        - Upload artifacts to Nexus Repository Manager.
        - Version control the artifacts for traceability and rollback capabilities.
10. **Build Docker Image:**
    - **Description**: Package the application into a Docker image.
    - **Actions**:
        - Use a Dockerfile to define the environment and dependencies.
        - Build the Docker image and tag it with appropriate version numbers.
11. **Scan Docker Image:**
    - **Description**: Ensure the Docker image is secure and free of vulnerabilities.
    - **Actions**:
        - Use tools like Trivy, Clair, or Aqua Security to scan the Docker image.
        - Address any vulnerabilities before proceeding.
12. **Deploy to Kubernetes:**
    - **Description**: Deploy the Docker image to a Kubernetes cluster.
    - **Actions**:
        - Use Kubernetes manifests or Helm charts to define the deployment.
        - Deploy the application to the cluster, managing pods, services, and ingress rules.
13. **Functional Testing:**
    - **Description**: Validate the application’s functionality in the deployed environment.
    - **Actions**:
        - Use tools like Selenium, Postman, or Cucumber to run automated functional tests.
        - Ensure the application meets all functional requirements and behaves as expected.
14. **Penetration Testing:**
    - **Description**: Perform security testing to identify potential vulnerabilities.
    - **Actions**:
        - Use tools like OWASP ZAP, Burp Suite, or Nessus to conduct penetration testing.
        - Identify and mitigate any security vulnerabilities found.