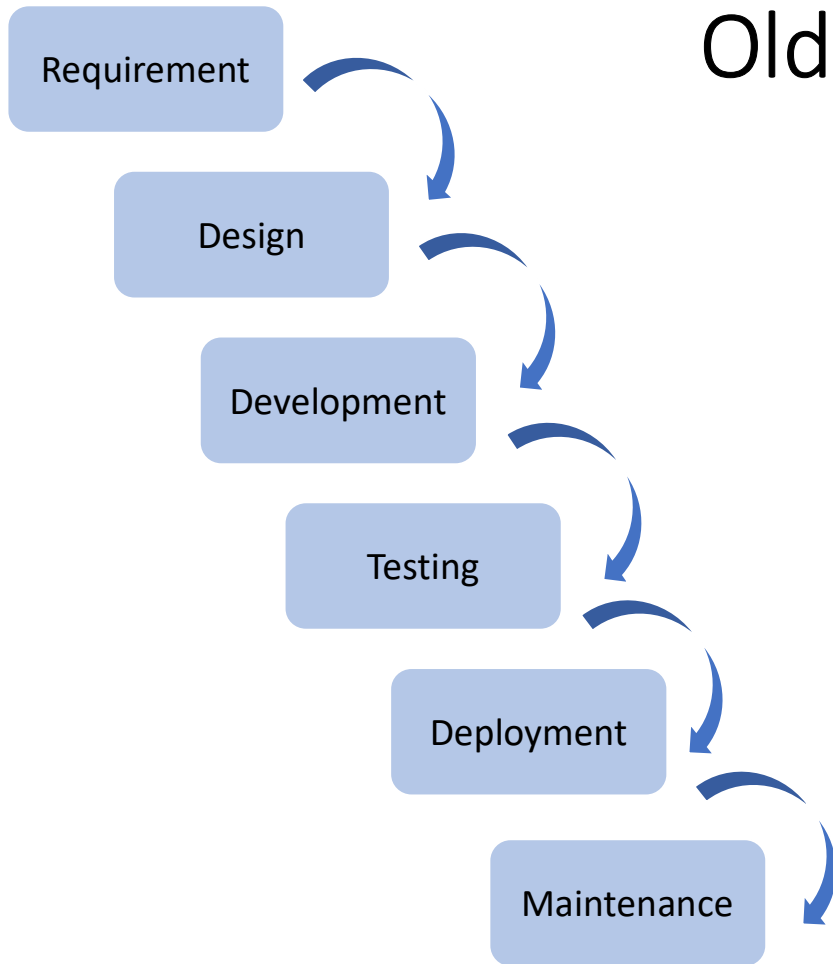


CI/CD with Jenkins

By Saurabh Dhingra

Old way of software delivery...

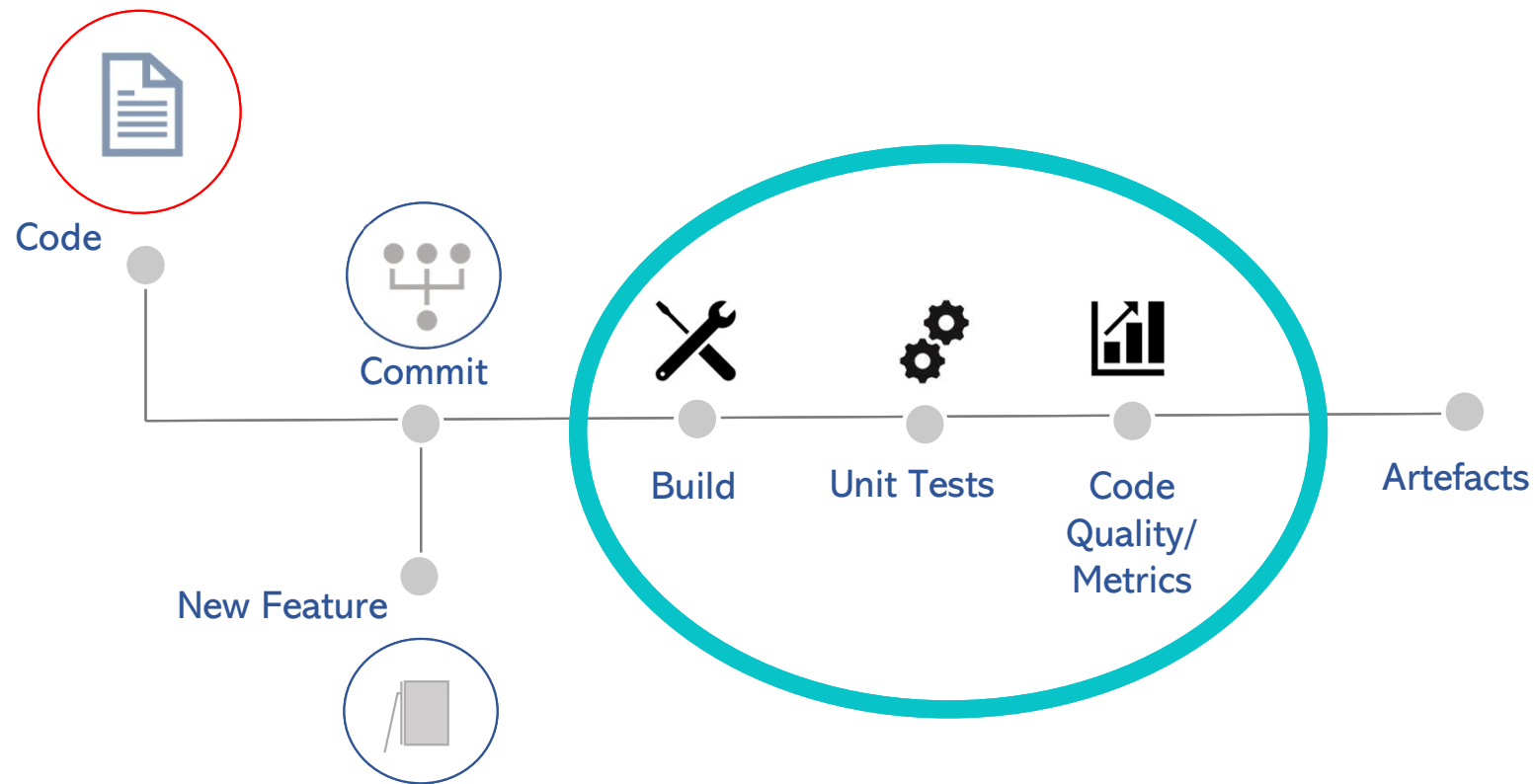


Traditional way of software delivery!

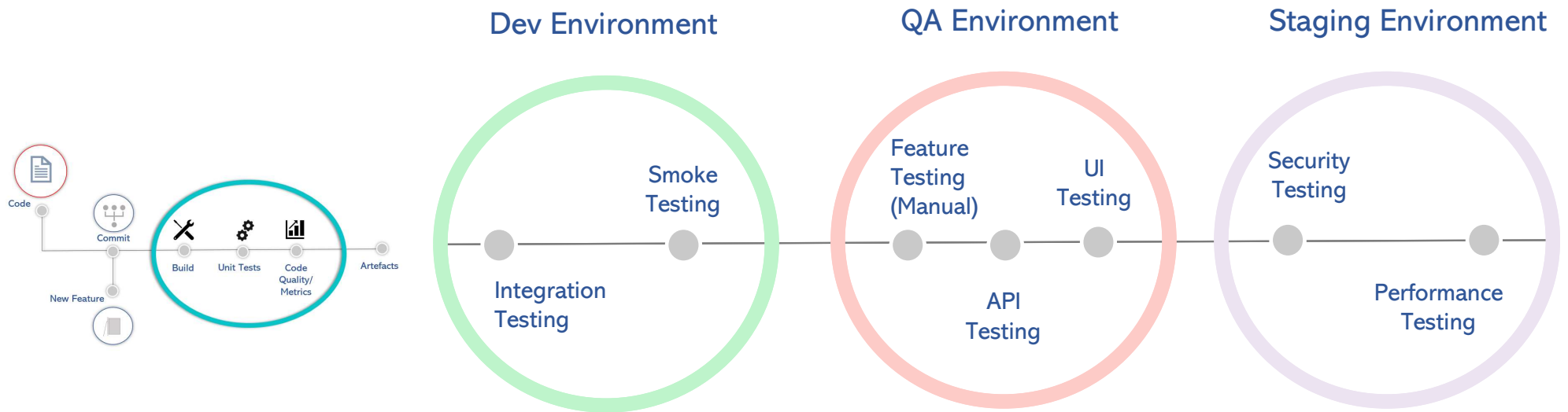
Some real-time challenges you will face in traditional way of software delivery!

- Demonstrating a new feature of your application to a customer on short notice
- That one critical bug in production

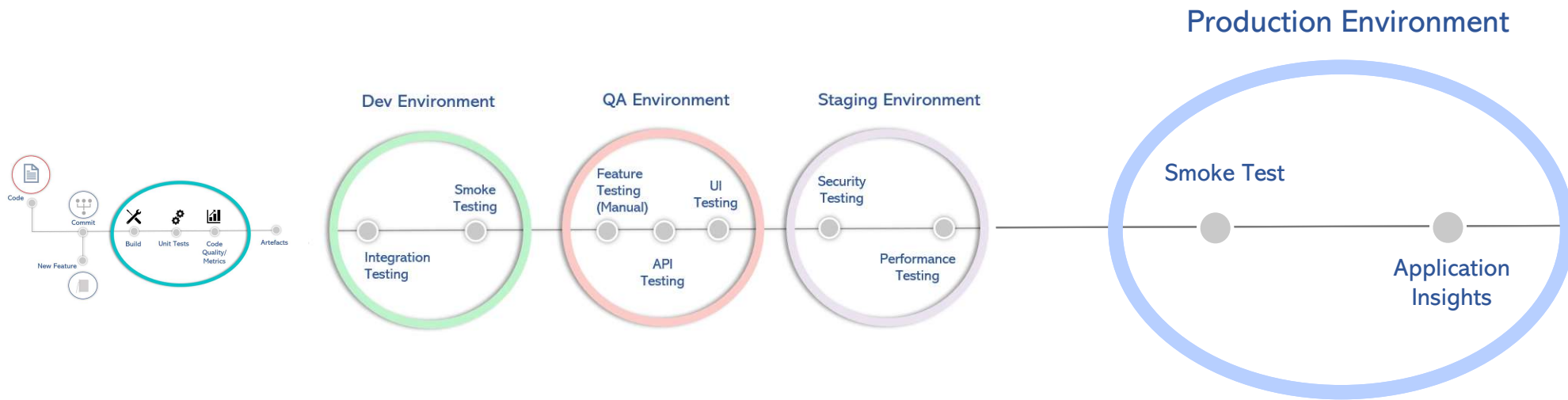
Continuous Integration

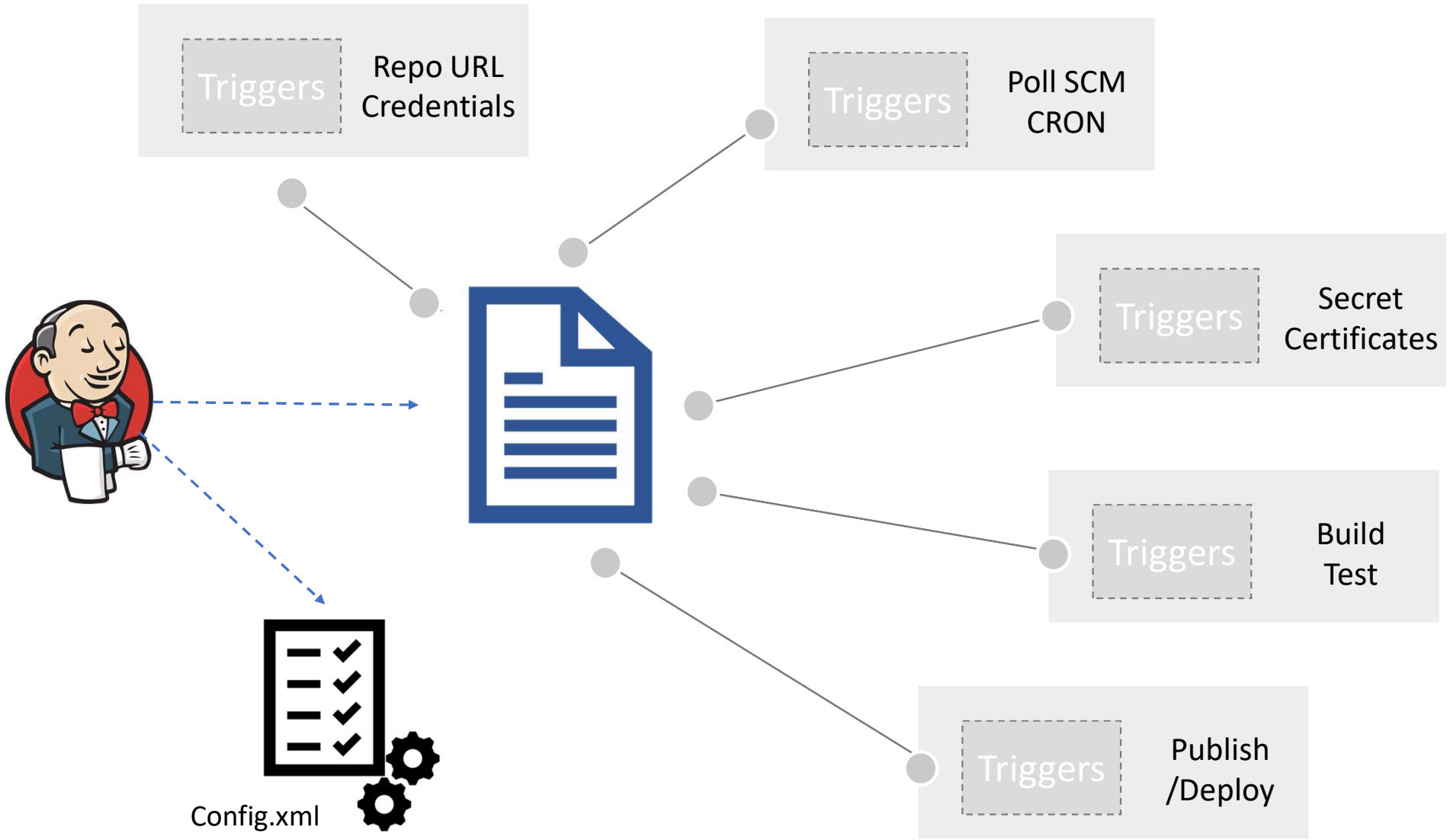


Continuous Delivery

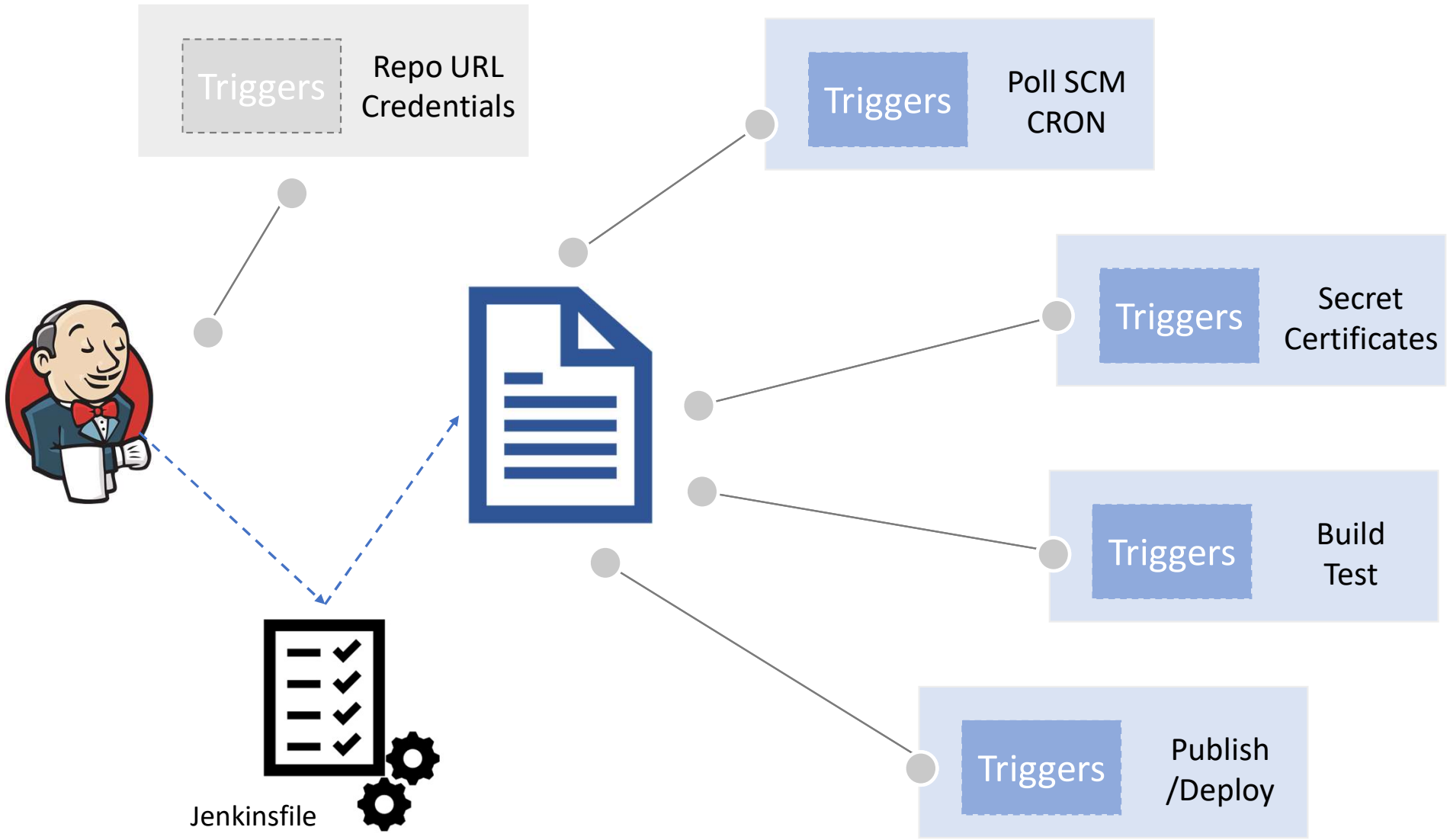


Continuous Deployment





Jenkins Pipeline



Jenkins Pipeline

Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the Pipeline domain-specific language (DSL) syntax.

The definition of a Jenkins Pipeline is written into a text file (called a Jenkinsfile) which in turn can be committed to a project's source control repository.

This is the foundation of "Pipeline-as-code"; treating the CD pipeline a part of the application to be versioned and reviewed like any other code.

Benefits of Jenkins Pipeline

1. Automatically creates a Pipeline build process for all branches and pull requests.
2. Code review/iteration on the Pipeline (along with the remaining source code).
3. Audit trail for the Pipeline.
4. Single source of truth for the Pipeline, which can be viewed and edited by multiple members of the project.

Features of a Jenkins pipeline

1. **Code:** Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.
2. **Durable:** Pipelines can survive both planned and unplanned restarts of the Jenkins controller.
3. **Pausable:** Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.
4. **Versatile:** Pipelines support complex real-world CD requirements, including the ability to fork/join, loop, and perform work in parallel.
5. **Extensible:** The Pipeline plugin supports custom extensions to its DSL and multiple options for integration with other plugins.

Jenkinsfile Example

```
pipeline {
  agent any

  stages {
    stage('Checkout'){
      steps {
        git 'https://github.com/saurabhd2106/maven-java-sample-app.git'
      }
    }
    stage('Build') {
      steps {

        // Run Maven Clean
        powershell 'mvn clean'

        powershell 'mvn compile'

        powershell 'mvn test'

        powershell 'mvn package'
      }

      post {
        // If Maven was able to run the tests, even if some of the test
        // failed, record the test results and archive the jar file.
        always {
          junit '**/target/surefire-reports/TEST-*.xml'
        }

        success {
          archiveArtifacts 'target/*.jar'
        }
      }
    }
  }
}
```