CI/CD and DevOps Automation Standard

Document ID: IE-DCS-2025.01
Version: 1.0
Effective Date: 2025-11-01
Department: IT & Engineering - DevOps

1. Goal

To enable high-frequency, reliable, and secure deployment of code and infrastructure changes with minimal human intervention.

2. Continuous Integration (CI) Requirements

- **Version Control:** All source code and IaC must reside in Git (private repositories).
- **Mandatory Testing:** Every pull request (PR) must pass automated unit tests, integration tests, and static code analysis (SAST) checks before merging into the main branch.
- **Artifact Registry:** Build outputs (Docker images, compiled binaries) must be immutably tagged and stored in a secure, audited container registry.
- **Dependency Scanning:** Mandatory execution of Software Composition Analysis (SCA) to check for known vulnerabilities and license compliance (per the **OSS Licensing Policy**).

3. Continuous Deployment (CD) Requirements

- **Environments:** The standard deployment flow is **Developer -> Staging -> Production**.
- **Staging Environment:** Must mirror Production infrastructure and data schema as closely as possible. All release candidates must undergo a full regression and performance test in Staging.
- **Gateways and Approval:** Deployment to Production requires automated checks (e.g., test success, security scan pass) and mandatory manual approval from the Engineering Lead and a peer from a separate team.
- **Deployment Strategy:** Prefer **Canary Deployments** or **Blue/Green** techniques to minimize downtime and facilitate rapid rollback. Direct in-place replacement is prohibited for critical services.
- **Rollback:** All deployments must have a defined, tested, and automated rollback strategy that can be executed within **5 minutes**.

4. Automation and Tooling

The use of configuration management tools (e.g., Ansible, Chef) is limited; preference is given to cloud-native orchestration (e.g., Kubernetes, serverless functions) managed via IaC.