

Puzzles in Build and Release Management

DevOps | CI/CD | Deployment |
Release Governance

Puzzle 1: Version Confusion

- **Scenario:**
 - Production was running v2.4.9. A hotfix v2.4.10 was deployed. Later, someone deployed v2.4.8 accidentally and failures started.
- **Puzzle Question:**
 - How can this happen even with CI/CD? How to prevent deploying older versions?
- **Solution / Learning:**
 - Use artifact immutability, release tagging, deployment approvals, and enforce version constraints. Use Git tags + artifact repository policies.

Puzzle 2: Missing Artifact



- **Scenario:**
 - Build pipeline shows SUCCESS, but release pipeline fails: 'Artifact not found in repository'.
- **Puzzle Question:**
 - Why can build succeed but artifact is missing?
- **Solution / Learning:**
 - Artifact upload step may fail silently, permissions issue, retention policy cleanup, wrong artifact name, or incorrect repository path.

Puzzle 3: Works on Developer Machine

Only

- **Scenario:**
 - App builds successfully on one developer system but fails in Jenkins with dependency errors.
- **Puzzle Question:**
 - What are the reasons and how to make builds reproducible?
- **Solution / Learning:**
 - Different OS/library versions, missing lock files, local cached dependencies, environment mismatch. Use Docker builds, dependency pinning, and build scripts.

Puzzle 4: Deployment Success but App Crash

- **Scenario:**
 - Deployment completed, but app crashes with error: Environment variable DB_PASSWORD not found.
- **Puzzle Question:**
 - Why pipeline didn't catch this issue? How to validate before release?
- **Solution / Learning:**
 - Pipeline validates deployment, not runtime config. Use config validation tests, secrets management (Vault), pre-deployment checks, and smoke tests.

Puzzle 5: Ghost Release



- **Scenario:**
 - Issue occurred at 11:30 PM. Release notes show deployment at 10 PM. DevOps claims no deployment happened.
- **Puzzle Question:**
 - How can code change without recorded deployment?
- **Solution / Learning:**
 - Auto-scaling instances may pull new images, manual server patching, or config drift. Use audit logs, immutable infra, and GitOps practices.

Puzzle 6: Two Releases One Server

- **Scenario:**
 - Team A deployed at 3 PM. Team B deployed at 3:05 PM. Production shows mixed features.
- **Puzzle Question:**
 - How is it possible? How to prevent it?
- **Solution / Learning:**
 - Partial deployment or shared services. Use blue-green deployments, atomic releases, immutable artifacts, and release locking mechanisms.

Puzzle 7: Infinite Pipeline Trigger

- **Scenario:**
 - Pipeline triggers after every commit, but one commit causes infinite pipeline executions.
- **Puzzle Question:**
 - What configuration mistake causes this?
- **Solution / Learning:**
 - Pipeline commits build metadata back to same branch, triggering again. Prevent by excluding CI commits or using separate branch for pipeline updates.

Puzzle 8: Rollback Trap



- **Scenario:**
 - Rollback failed because database schema was already modified.
- **Puzzle Question:**
 - Why is rollback difficult? Best strategy?
- **Solution / Learning:**
 - DB changes are irreversible if not planned. Use backward-compatible migrations, feature flags, versioned schema changes, and rollback scripts.

Puzzle 9: Same Commit Different

Output

- **Scenario:**
 - Same commit built twice produces different binaries.
- **Puzzle Question:**
 - How is this possible? How to achieve deterministic builds?
- **Solution / Learning:**
 - Build uses timestamps, non-pinned dependencies, or random ordering. Use locked dependencies, reproducible builds, and containerized environments.

Puzzle 10: Unauthorized Release

- **Scenario:**
 - A release happened without manager approval.
- **Puzzle Question:**
 - What is missing in pipeline governance?
- **Solution / Learning:**
 - Missing approval gates/role-based access. Use manual approval stage, RBAC, audit trail, and release policies in CI/CD tools.

Puzzle 11: Configuration Drift

- **Scenario:**
 - Application works in staging but fails in production even with same artifact.
- **Puzzle Question:**
 - Why?
- **Solution / Learning:**
 - Production environment differs in config, OS patches, firewall rules, or secrets. Use IaC (Terraform), config-as-code, and environment parity checks.

Puzzle 12: Hotfix Overwrites Features

- **Scenario:**
 - Hotfix deployed to production fixed bug but removed new feature.
- **Puzzle Question:**
 - Why?
- **Solution / Learning:**
 - Hotfix branch was created from old version. Use proper branching strategy, cherry-picking fixes, and merge back into main branch.