Mapping the Five AppSec Core Processes to a 16-Gate CI/CD Pipeline

Version 1.1

Overview

This document maps the **five core AppSec processes** (*Plan/Design, Build, Test, Release, Operate*) to **16 CI/CD gates**. Each gate lists its primary process, security intent, and example evidence.

1 Gate → AppSec Process Mapping

#	CI/CD Gate	Primary AppSec Process	What this gate enforces (AppSec intent)	Typical evidence / signals
01	Source code version control	**Build**	Protected branches; required reviews; signed commits; CODEOWNERS; secret push-protection.	Repo settings export; audit log; PR policy status.
02	Optimum branching strategy	**Build**	PR-centric flow; short-lived branches; enforced checks before merge.	Branch protection rules; PR template; required checks list.
12	Build / deploy / test each commit	**Build**	Reproducible builds; pinned actions; secretless OIDC auth; deterministic artifacts.	Workflow run logs; build provenance/attestation.
04	≥80% code coverage	**Build**	Minimum unit-test coverage threshold per service.	Coverage report artifact; hard fail if below threshold.
03	Static analysis (SAST)	**Build**	PR checks for code flaws and secrets; severity thresholds/gating.	SAST report; secret-scan report; PR check status.
05	Vulnerability scan	**Build**	Dependency/container CVE policy by severity, age, and SLA.	SBOM + scan results; allow/deny decision trail.
06	Open-source (SCA / license) scan	**Build**	License and component policy compliance.	SCA license report; approved/exception record.
07	Artifact version control	**Release**	Immutable, signed, provenance-attested artifacts (supply chain).	Image digest; signature (e.g., Sigstore); SLSA-like attestations.
08	Auto provision (IaC)	**Operate**	Baseline-hardened infrastructure via IaC; policy-as-code on plans.	OPA/Conftest results; plan/apply logs.
09	Immutable servers	**Operate**	Golden images/immutable containers; drift prevention.	Image recipe; container digest pinning; drift alerts.
10	Integration testing	**Test**	Security-relevant integration/API tests from misuse cases.	Integration test suite results; contract tests; negative tests.
11	Performance / load testing	**Test**	Performance/SLO thresholds as DoS guardrails.	Load test report vs. SLOs; error budgets.
14	Automated change order	**Release**	Change governance links risk posture to approvals using objective evidence.	Change record referencing scans, SBOM, exceptions.
15	Zero-downtime release	**Release**	Progressive rollout (blue/green, canary) with health guardrails.	Deployment strategy logs; health-gate status.

#	CI/CD Gate	Primary AppSec Process	What this gate enforces (AppSec intent)	Typical evidence / signals
16	Feature toggle	**Release**	Progressive delivery and kill-switch controls.	Toggle audit log; scoped rollout policy.
13	Automated rollback	**Operate**	Auto-revert on SLO/SI breach; incident linkage.	Rollback trigger tied to SLOs; incident/alert record.

2 Process → Gates Index

Plan/Design
Establishes policies and thresholds used by all gates (especially 01–06 and 08–16).

Build
01, 02, 12, 04, 03, 05, 06

Test
10, 11

Release
07, 14, 15, 16

Operate
08, 09, 13

3 Reusable Mapping

The following block can live in a repo/wiki and be validated by automation.

```
appsec_to_cicd_gates:
  - gate: 01
    name: Source code version control
    primary_process: Build
    intent: "Repo protections, reviews, signed commits, secret push-protection"
    evidence: ["branch_protection_export", "audit_log", "required_checks_status"]
  - gate: 02
    name: Optimum branching strategy
    primary_process: Build
    intent: "PR-centric flow; enforce checks before merge"
    evidence: ["PR_template", "branch_rules", "required_checks"]
  - gate: 12
    name: Build/deploy/test each commit
    primary_process: Build
    intent: "Reproducible, pinned, secretless builds"
    evidence: ["workflow_logs", "build_attestation"]
  - gate: 04
    name: ">=80% coverage"
    primary_process: Build
    intent: "Test coverage threshold"
    evidence: ["coverage_report"]
  - gate: 03
    name: Static analysis (SAST)
    primary_process: Build
    intent: "SAST + secrets on PR; severity gating"
    evidence: ["sast_report", "secrets_report", "check_status"]
  - gate: 05
    name: Vulnerability scan
    primary_process: Build
    intent: "Dependency/container vuln policy"
    evidence: ["sbom", "vuln_scan_results"]
  - gate: 06
    name: Open source scan
    primary_process: Build
    intent: "License/composition compliance"
    evidence: ["sca_license_report"]
  - gate: 07
    name: Artifact version control
    primary_process: Release
    intent: "Signed, immutable, provenance-attested artifacts"
    evidence: ["digest", "signature", "provenance_attestation"]
  - gate: 08
    name: Auto provision
    primary_process: Operate
    intent: "IaC security baselines; policy-as-code"
    evidence: ["opa_conftest_results", "plan_apply_logs"]
  - gate: 09
    name: Immutable servers
    primary_process: Operate
    intent: "Golden images; drift prevention"
    evidence: ["image_recipe", "container_digest", "drift_alerts"]
  - gate: 10
    name: Integration testing
    primary_process: Test
```

```yaml
    intent: "Security-relevant integration/API checks"
    evidence: ["integration_test_report"]
  - gate: 11
    name: Performance testing
    primary_process: Test
    intent: "Perf/SLO guardrails"
    evidence: ["load_test_report", "error_budget_status"]
  - gate: 14
    name: Automated change order
    primary_process: Release
    intent: "Risk-aware approvals with security evidence"
    evidence: ["change_record_with_scan_links"]
  - gate: 15
    name: Zero downtime release
    primary_process: Release
    intent: "Blue/green or canary with health gates"
    evidence: ["deployment_logs", "health_gate_status"]
  - gate: 16
    name: Feature toggle
    primary_process: Release
    intent: "Progressive delivery and kill-switch controls"
    evidence: ["toggle_audit_log"]
  - gate: 13
    name: Automated rollback
    primary_process: Operate
    intent: "Auto-revert on SLO/SI breach; incident linkage"
    evidence: ["rollback_event", "incident_record"]
```

Notes

- **Compilation**: This document uses `minted`. Compile with `-shell-escape`.
- Evidence examples are vendor-agnostic; substitute platform artifacts as needed.

Last updated: October 29, 2025.