

Deployment Checklist

Deployment Checklist

Project: {{PROJECT_NAME}} Version: {{VERSION}} Date: {{DATE}}
Author: {{AUTHOR}} Status: Draft | In Review | Approved Reviewers:
{{REVIEWERS}}

Document History

| Version | Date | Author | Changes |
|---------|----------|------------|---------------|
| 0.1 | {{DATE}} | {{AUTHOR}} | Initial draft |

Deployment Metadata

| Field | Value |
|--------------------|-----------------------|
| Version | {{VERSION}} |
| Target Environment | {{ENVIRONMENT}} |
| Deployment Date | {{DATE}} |
| Deployment Time | {{TIME}} {{TIMEZONE}} |
| Deploy Engineer | {{ENGINEER}} |
| Approver | {{APPROVER}} |
| Deployment Type | {{TYPE}} |
| Change Request # | {{CR_NUMBER}} |
| Rollback Version | {{ROLLBACK_VERSION}} |

1. Pre-Deployment Checklist

Code Quality Gates

- Code reviewed and approved** — all PRs in this release have \geq `{{REVIEW_COUNT}}` approvals
- All unit tests passing** — CI pipeline green on the release branch
- All integration tests passing** — CI pipeline integration stage green
- Code coverage meets minimum** — \geq `{{COV_GATE}}`% (current: `{{CURRENT_COV}}`%)
- No HIGH/CRITICAL security findings** — SAST and SCA scans clean
- No secrets detected** — secret scanning clean
- Linting passes** — no errors in CI lint stage

Staging Verification

- Staging deployment complete** — this exact artifact has been deployed to staging
- E2E tests passing on staging** — automated suite green (`{{PASS_COUNT}}`/`{{TOTAL_COUNT}}` tests)
- Manual QA sign-off obtained** — QA team has reviewed new features
- Performance baseline not degraded** — P95 within `{{PERF_GATE}}`% of baseline
- Visual regression checks passed** (if UI changes)

Database & Migrations

- Database migrations reviewed** — SQL reviewed by a second person
- Migrations tested on staging** — ran successfully, duration recorded: `{{MIGRATION_TIME}}`min
- Down/rollback migration tested** — `migrate down` executes without errors
- Migration script idempotent** — safe to run twice without errors
- Estimated migration time on production** documented: `{{PROD_MIGRATION_TIME}}`min
- Data backup taken** (if destructive migration) — backup ID: `{{BACKUP_ID}}`

Environment & Configuration

- All environment variables documented** and updated in `{{SECRET_STORE}}`
- New secrets provisioned** in production secret store
- Configuration changes reviewed** — no unintended changes from staging
- External API keys** are live (not sandbox) credentials
- DNS records verified** (if changes required)

Feature Flags

- Feature flags configured** — new features default OFF in production
- Kill switches in place** for all new significant features
- Rollout plan documented** — which flags, in what order, over what timeline
- Feature flag audit complete** — no stale flags from previous releases

Rollback Readiness

- Rollback plan documented** — see Section 5 and [rollback-plan.md](#)
- Previous version artifact available** — `{{PREV_IMAGE}}` in registry
- Rollback tested** on staging (if DB migrations are included)
- Rollback owner assigned** — `{{ROLLBACK_OWNER}}` is available during deployment

Operational Readiness

- On-call engineer notified** — `{{ONCALL}}` is aware and available
- Deployment window confirmed** — `{{TIME}}` - `{{END_TIME}}` (off-peak)
- Monitoring dashboards open** and ready
- War room set up** — `{{WAR_ROOM_LINK}}`
- Change request approved** — CR-`{{CR_NUMBER}}` approved by `{{APPROVER}}` on `{{DATE}}`

2. During Deployment

| Step | Time | Actor | Status | Notes |
|---------------------------------------|------|---------------------------|--------|-------|
| Announce deployment start in war room | | <code>{{ENGINEER}}</code> | | |

| Step | Time | Actor | Status | Notes |
|--|------|--------------|--------|-------------------------------------|
| Enable maintenance mode (if required) | | {{ENGINEER}} | | |
| Trigger deployment pipeline: <code>{{PIPELINE_LINK}}</code> | | {{ENGINEER}} | | |
| Monitor deployment progress | | {{ENGINEER}} | | Watching for errors |
| Run database migrations (if applicable) | | {{DB_OWNER}} | | Duration: <code>{{TIME}}</code> min |
| Verify migration success: <code>{{VERIFY_CMD}}</code> | | {{DB_OWNER}} | | |
| Confirm new instances healthy (health checks green) | | {{ENGINEER}} | | |
| Confirm all instances running new version | | {{ENGINEER}} | | |
| Run smoke tests: <code>bash scripts/smoke-tests.sh {{ENV}}</code> | | {{QA}} | | |
| Verify health endpoint: <code>curl {{URL}}/health</code> | | {{ENGINEER}} | | |
| Disable maintenance mode (if enabled) | | {{ENGINEER}} | | |

3. Post-Deployment Checklist

Immediate Verification (First 15 Minutes)

- Health checks passing** — all instances healthy per load balancer
- Smoke tests pass** — automated smoke suite green
- Critical user journey manual test** — manually verify `{{CRITICAL_JOURNEY}}`
- Error rate normal** — $< \text{{ERROR_THRESHOLD}}\%$ (check `{{DASHBOARD_LINK}}`)
- P99 latency normal** — $< \text{{P99_THRESHOLD}}$ ms
- Database connections normal** — no connection pool saturation
- No unexpected errors in logs** — `{{LOG_DASHBOARD_LINK}}` shows clean logs

Feature Validation (First 30 Minutes)

- New features functional** — quick manual test of each feature in scope
- Feature flags enabled per rollout plan** — `{{FF_DASHBOARD}}` shows correct state
- Third-party integrations working** — payments, email, external APIs responding
- Version confirmed** — `GET /api/version` or equivalent returns `{{VERSION}}`

Monitoring Setup (First 60 Minutes)

- Performance baseline compared** — P95 matches or beats staging baseline
- CPU and memory normal** — no upward trend
- Alerting verified** — test alert confirms delivery to on-call
- Stakeholders notified** — release announcement sent to `{{NOTIFY_LIST}}`
- Documentation updated** — release notes published

4. Rollback Procedure (Quick Reference)

Rollback triggers (any one sufficient):

- Smoke tests fail after deployment
- Error rate > `{{ROLLBACK_ERROR}}`% for > `{{ROLLBACK_DURATION}}` consecutive minutes
- P99 > `{{ROLLBACK_P99}}`ms sustained
- Data integrity issue detected

Rollback authorization: `{{ROLLBACK_AUTHORITY}}` or senior engineer on duty

Rollback steps:

```
# 1. Announce in war room: "Initiating rollback to {{ROLLBACK_VERSION}}"  
# 2. Trigger rollback:  
{{ROLLBACK_CMD}}  
  
# 3. If DB migration was included – run down migration:  
{{DOWN_MIGRATION_CMD}}
```

```
# 4. Verify rollback:
curl {{URL}}/health
bash scripts/smoke-tests.sh {{ENV}}
```

Expected rollback time: {{ROLLBACK_TIME}} minutes **Full rollback procedure:** [rollback-plan.md](#)

5. Emergency Deployment Process

Emergency deployments require:

1. Sign-off from {{EMERGENCY_AUTHORITY}} (Engineering Manager or above)
2. At least {{EMERGENCY_REVIEWS}} code reviewer (can be async if truly urgent)
3. Staging deployment and smoke test (cannot be skipped)
4. Enhanced post-deploy monitoring for {{EMERGENCY_MONITOR}}h
5. Full retroactive change request within {{CR_SLA}}h of deployment

Emergency deployment allowed: Security vulnerabilities, data loss bugs, P1 service outages

6. Deployment Window & Blackout Periods

Standard deployment window: {{DEPLOY_WINDOW}} **Emergency deployments:** Any time, with approvals

Blackout periods (no production deployments):

| Period | Dates | Reason |
|-------------|-----------------------------|-------------------------------|
| End of year | {{EOY_START}} - {{EOY_END}} | High traffic, minimal support |
| {{EVENT_1}} | {{DATE_1}} | {{REASON_1}} |
| {{EVENT_2}} | {{DATE_2}} | {{REASON_2}} |

Sign-Off

Pre-deployment confirmed by: {{ENGINEER}} on {{DATE}} at {{TIME}} **Deployment completed by:** {{ENGINEER}} on {{DATE}} at {{TIME}} **Post-deployment verified by:** {{ENGINEER}} on {{DATE}} at {{TIME}}

Related Documents

- [Release Notes](#)
 - [Rollback Plan](#)
 - [Go-Live Runbook](#)
 - [Operational Runbook](#)
-

Approval

| Role | Name | Date | Signature |
|----------|------|------|-----------|
| Author | | | |
| Reviewer | | | |
| Approver | | | |

Revision #3

Created 2026-02-24 14:54:20 UTC by John

Updated 2026-05-25 07:34:38 UTC by John