Release Management Guide

Overview

This document outlines the release process for the agent-orchestration-ops repository, including versioning, tagging, and deployment procedures.

Versioning Strategy

We follow Semantic Versioning (SemVer) for operational releases:
- **MAJOR.MINOR.PATCH** (e.g., 1.2.3)
- **MAJOR**: Breaking changes to operational procedures or infrastructure
- **MINOR**: New features, runbooks, or non-breaking enhancements
- **PATCH**: Bug fixes, documentation updates, minor improvements

Release Types

1. Operational Releases (v1.x.x)

- Complete operational framework updates
- New runbook collections
- Major alert configuration changes
- Infrastructure procedure updates

2. Hotfix Releases (v1.x.x-hotfix.x)

- Critical security updates
- Emergency procedure fixes
- Urgent alert configuration corrections

3. Pre-release Versions (v1.x.x-alpha.x, v1.x.x-beta.x)

- Testing new operational procedures
- Experimental runbooks
- Beta alert configurations

Release Process

1. Pre-Release Checklist

- [ ] All CI checks passing (readiness:agents, alerts:validate, runbooks:index)
- [ ] Security scan completed without critical issues
- [ ] Documentation updated and reviewed
- [ ] CHANGELOG.md updated with release notes
- [ ] Version bumped in relevant configuration files

2. Release Branch Creation

```
# Create release branch from main
git checkout main
git pull origin main
git checkout -b release/v1.2.3

# Update version numbers and documentation
# Commit changes
git add .
git commit -m "Prepare release v1.2.3"
git push origin release/v1.2.3
```

3. Release Testing

- [ ] Deploy to staging environment
- [ ] Run full operational validation suite
- [ ] Execute tabletop drill scenarios
- [ ] Validate all runbooks and procedures
- [ ] Security and compliance review

4. Release Approval

- [ ] Technical review by platform team
- [ ] Security review by security team
- [ ] Operations team sign-off
- [ ] Management approval for major releases

5. Release Deployment

```
# Merge release branch to main
git checkout main
git merge release/v1.2.3
git push origin main

# Create and push tag
git tag -a v1.2.3 -m "Release v1.2.3: [Brief description]"
git push origin v1.2.3

# Clean up release branch
git branch -d release/v1.2.3
git push origin --delete release/v1.2.3
```

6. Post-Release Activities

- [ ] Update production environments
- [ ] Notify stakeholders of release
- [ ] Monitor systems for issues
- [ ] Update documentation portals
- [ ] Schedule post-release review

Rollback Procedures

Emergency Rollback

```
# Revert to previous stable tag
git checkout v1.2.2
git checkout -b hotfix/rollback-v1.2.3
# Apply necessary fixes
git commit -m "Emergency rollback from v1.2.3"
git tag -a v1.2.4 -m "Hotfix: Emergency rollback"
```

Planned Rollback

- Follow standard release process with rollback changes
- Update CHANGELOG.md with rollback rationale
- Communicate rollback to all stakeholders

Release Communication

Internal Communication

- Engineering team notification via Slack/Teams
- Operations team briefing on changes
- Security team notification of security-related changes
- Management summary for major releases

External Communication

- Customer notification for breaking changes
- Documentation updates on public portals
- API deprecation notices where applicable

Release Metrics and Monitoring

Key Metrics

- Release frequency
- Time from code complete to production
- Rollback frequency and reasons
- Post-release incident count
- Customer impact metrics

Monitoring

- Automated deployment success/failure tracking
- Post-release system health monitoring
- Performance impact assessment
- Security posture validation

Emergency Release Process

Criteria for Emergency Release

- Critical security vulnerabilities
- Production system failures
- Data integrity issues
- Compliance violations

Expedited Process

1. Create emergency branch from main
2. Implement minimal fix
3. Fast-track security and technical review
4. Deploy with enhanced monitoring
5. Schedule post-incident review

Release Calendar

Regular Release Schedule

- **Major Releases**: Quarterly (Q1, Q2, Q3, Q4)
- **Minor Releases**: Monthly
- **Patch Releases**: As needed (typically weekly)
- **Security Releases**: Immediate (within 24-48 hours)

Blackout Periods

- Holiday seasons (Dec 20 - Jan 5)
- Major business events
- Planned maintenance windows
- High-traffic periods

Tools and Automation

Required Tools

- Git for version control
- GitHub Actions for CI/CD
- Semantic versioning tools
- Automated testing frameworks
- Monitoring and alerting systems

Automation Opportunities

- Automated version bumping
- Release note generation
- Deployment automation
- Rollback automation
- Notification systems

Last Updated: 2025-09-29
Next Review: 2025-12-29
Owner: Platform Engineering Team