

VIOE - Vulnerability Intelligence & Orchestration Engine

Deployment & Environment Guide

Document Version: 1.0 Classification: Internal - Operations / DevOps Last Updated: January 2026

Table of Contents

1. [Introduction](#)
 2. [Environment Definitions](#)
 3. [Deployment Architecture](#)
 4. [Release Workflow](#)
 5. [Deployment Procedures](#)
 6. [Rollback Strategy](#)
 7. [Change Controls](#)
 8. [Environment Configuration](#)
 9. [Infrastructure Requirements](#)
 10. [Monitoring & Health Checks](#)
 11. [Disaster Recovery](#)
-

1. Introduction

1.1 Purpose

This guide defines VIOE's deployment architecture, environment structure, release processes, and rollback procedures. It ensures safe, consistent, and auditable deployments.

1.2 Audience

- DevOps Engineers

- Release Managers
- System Administrators
- Operations Teams

1.3 Deployment Philosophy

VIOE follows these deployment principles:

Principle	Description
Immutable Infrastructure	Deploy new instances, don't modify running ones
Blue-Green Deployment	Zero-downtime releases
Infrastructure as Code	All configuration in version control
Automated Testing	Automated validation before deployment
Observability	Comprehensive monitoring and logging

2. Environment Definitions

2.1 Environment Overview

Environment	Purpose	Users	Data
Development	Feature development	Developers	Test data
Staging	Pre-release validation	QA, Product	Anonymized production
Production	Live customer use	All users	Real data

2.2 Development Environment

Purpose: Active development and unit testing

Characteristics:

Aspect	Configuration
Stability	May be unstable
Data	Synthetic test data
Access	Development team
Updates	Continuous
Scale	Minimal

Use Cases:

- Feature development
- Bug fixes
- Unit testing
- Integration testing

2.3 Staging Environment

Purpose: Pre-production validation and acceptance testing

Characteristics:

Aspect	Configuration
Stability	Production-like
Data	Anonymized production data
Access	QA, Product, Select stakeholders
Updates	Release candidates
Scale	Production-similar

Use Cases:

- User acceptance testing (UAT)
- Performance testing
- Integration validation
- Release verification

Staging Requirements:

- Must mirror production configuration
- Must have recent anonymized data
- Must pass all tests before production promotion

2.4 Production Environment

Purpose: Live system serving customers

Characteristics:

Aspect	Configuration
Stability	Highest priority
Data	Real customer data
Access	Authorized users
Updates	Controlled releases
Scale	Full capacity

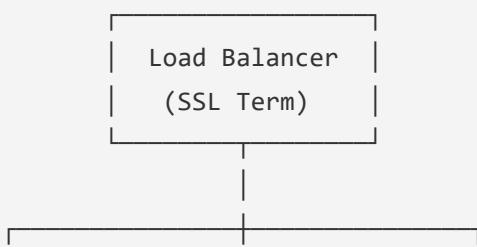
Production Controls:

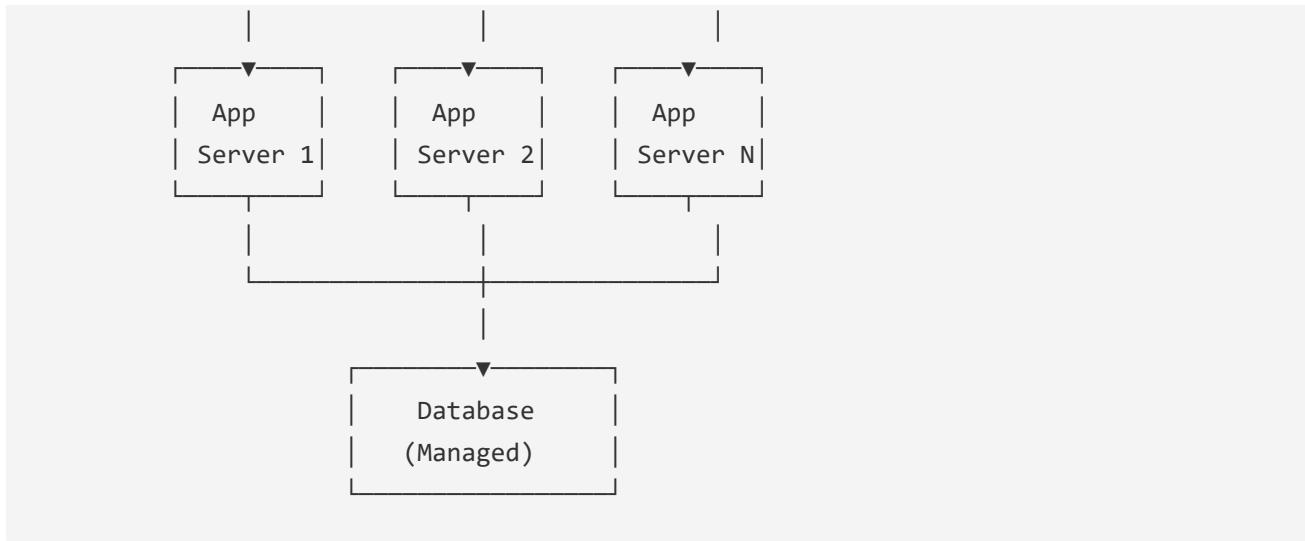
- Change management required
- Deployment windows defined
- Rollback capability ready
- Monitoring active

3. Deployment Architecture

3.1 Architecture Overview

[Diagram Placeholder: Deployment Architecture]

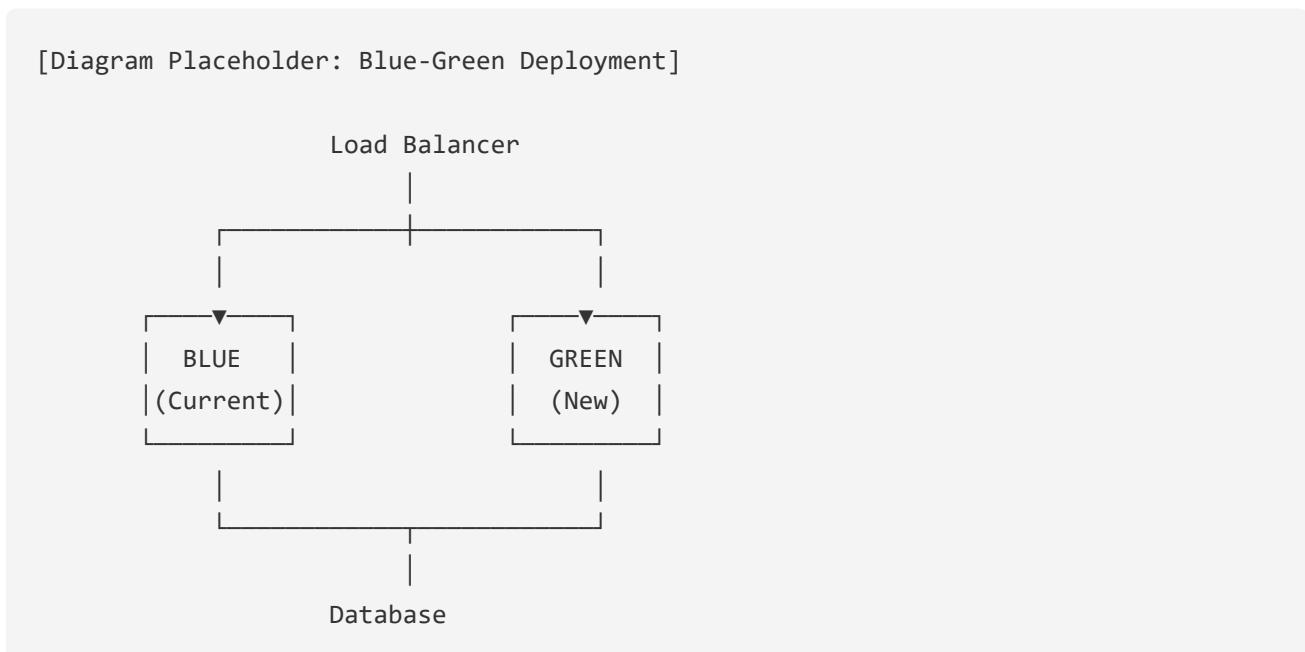




3.2 Component Distribution

Component	Deployment Model	Scaling
Frontend (SPA)	CDN	Automatic
Application Server	Container/VM	Horizontal
Database	Managed Service	Vertical/Horizontal
File Storage	Object Storage	Automatic
AI Services	Managed Service	Automatic

3.3 Blue-Green Deployment Model

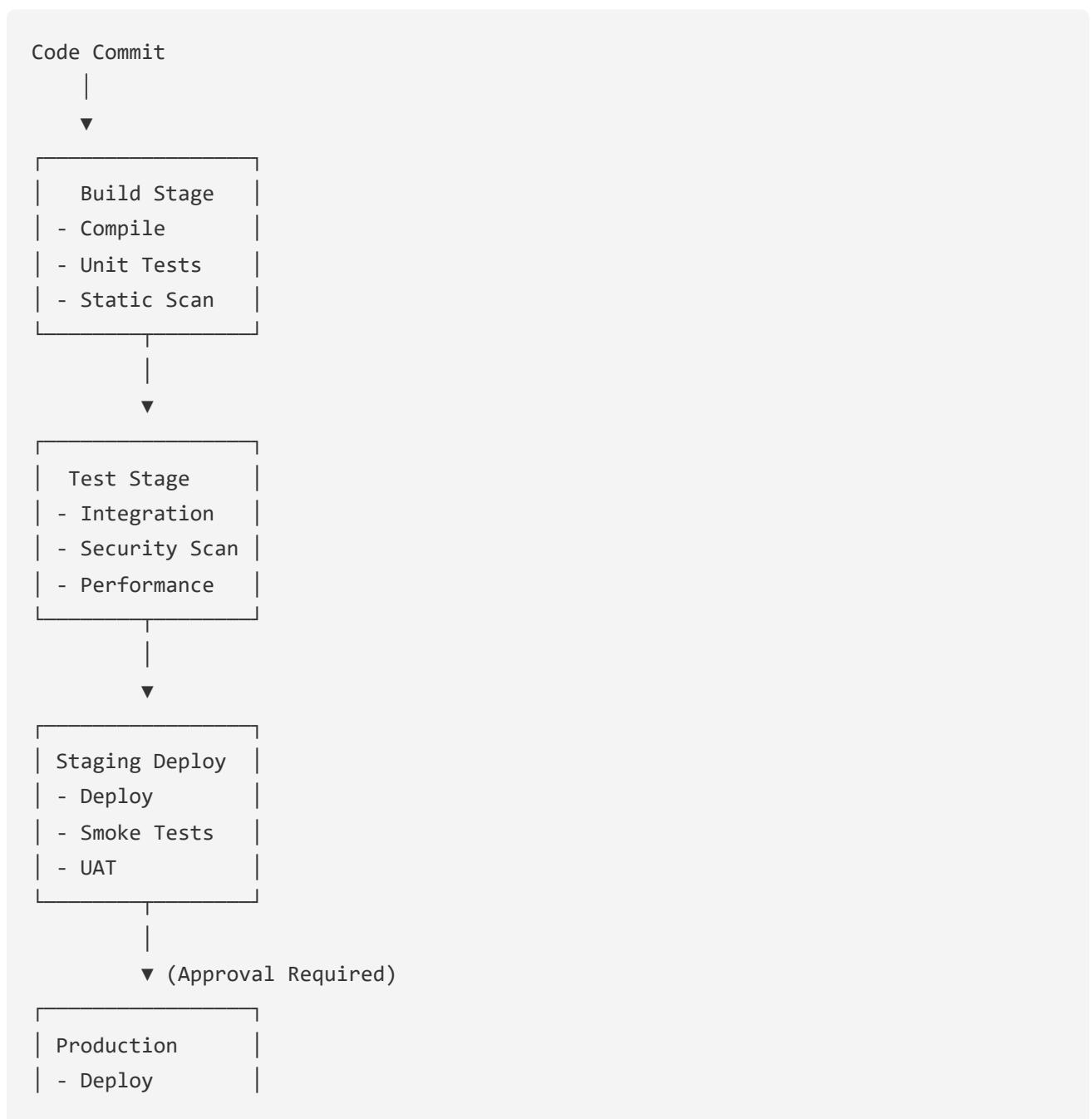


Process:

1. Green environment deployed with new version
 2. Green environment validated
 3. Traffic switched from Blue to Green
 4. Blue becomes standby for rollback
-

4. Release Workflow

4.1 Release Pipeline



- | - Verify
- | - Monitor

4.2 Release Types

Type	Description	Approval	Window
Major	New features, breaking changes	Full CAB	Planned maintenance
Minor	Enhancements, non-breaking	Team lead	Low-traffic period
Patch	Bug fixes, security	Manager	Anytime with notice
Hotfix	Critical production fix	Manager + CISO	Immediate

4.3 Release Schedule

Activity	Timing
Code freeze	Tuesday 5 PM
Staging deployment	Wednesday AM
UAT/Validation	Wednesday-Thursday
Production deployment	Thursday 10 PM
Post-deployment monitoring	Thursday-Friday

4.4 Release Checklist

Pre-Release:

- All tests passing
- Security scan complete
- Performance benchmarks met
- Release notes prepared
- Rollback plan documented
- Stakeholders notified
- Change request approved

Release:

- Staging deployment successful
- Staging validation complete
- Production deployment initiated
- Health checks passing
- Smoke tests complete

Post-Release:

- Monitoring dashboards reviewed
- No error spikes
- Performance normal
- User feedback monitored
- Release notes published

5. Deployment Procedures

5.1 Standard Deployment Procedure

Step 1: Pre-Deployment Verification

Check	Command/Action	Expected
Build status	Check CI pipeline	Green
Test results	Review test report	All passing
Security scan	Review scan results	No critical
Approval	Change management	Approved

Step 2: Staging Deployment

Action	Details
Deploy to staging	Trigger deployment pipeline
Run smoke tests	Automated test suite
Verify functionality	Manual spot checks
Performance check	Compare to baseline

Step 3: Production Deployment

Action	Timing	Owner
Notify stakeholders	1 hour before	Release Manager
Initiate deployment	Scheduled time	DevOps
Monitor deployment	During	DevOps
Run smoke tests	Post-deploy	DevOps
Verify metrics	Post-deploy	DevOps
Announce completion	After verification	Release Manager

5.2 Hotfix Deployment Procedure

For critical production issues:

Step	Action	Timeline
1	Identify and verify issue	Immediate
2	Develop fix	ASAP
3	Test fix locally	Required
4	Obtain emergency approval	Required
5	Deploy to staging	Quick validation
6	Deploy to production	After staging verify
7	Verify fix	Immediate
8	Post-incident review	Within 24 hours

5.3 Database Migration Procedure

For schema changes:

Step	Action	Verification
1	Backup database	Backup confirmed
2	Apply migration to staging	Staging works
3	Verify data integrity	Data check passes
4	Apply migration to production	During maintenance
5	Verify production data	Integrity check
6	Deploy application update	Schema compatible

6. Rollback Strategy

6.1 Rollback Triggers

Automatic Rollback:

Trigger	Threshold
Health check failures	3 consecutive failures
Error rate spike	>5% increase
Response time	>300% increase
Failed smoke tests	Any critical failure

Manual Rollback:

Trigger	Decision By
User-reported critical issue	Operations + Product
Data integrity concern	Operations + DBA
Security vulnerability	Security team

6.2 Rollback Procedure

Immediate Rollback (Blue-Green):

Step	Action	Time
1	Identify rollback need	-
2	Switch traffic to previous version	2 min
3	Verify previous version serving	1 min
4	Run smoke tests	5 min
5	Monitor metrics	Ongoing
6	Notify stakeholders	Immediate

Full Rollback (If Blue-Green not available):

Step	Action	Time
1	Stop current deployment	1 min
2	Deploy previous version	10-15 min
3	Run database rollback (if needed)	Variable
4	Verify deployment	5 min
5	Run smoke tests	5 min
6	Resume traffic	1 min

6.3 Database Rollback

If database migration must be reversed:

Step	Action	Notes
1	Stop application traffic	Prevent new writes
2	Execute rollback script	Pre-written for each migration
3	Verify data state	Data integrity check
4	Deploy previous application	Compatible with old schema
5	Resume traffic	After verification

6.4 Rollback Communication

Template:

ROLLBACK NOTIFICATION

Time: [Timestamp]

Environment: Production

Reason: [Brief description]

Action Taken: Rollback to version [X.Y.Z]

Current Status: [Stable/Monitoring]

Impact: [Description of impact]

Resolution: [Next steps]

Contact: [Operations contact]

7. Change Controls

7.1 Change Classification

Classification	Description	Approval Required
Standard	Pre-approved, low risk	Team lead
Normal	Planned changes	Change Advisory Board
Emergency	Critical fixes	Manager + stakeholder

7.2 Change Request Process

Standard Change:

Step	Action	Owner
1	Submit change request	Developer
2	Team lead review	Team Lead
3	Schedule deployment	Release Manager
4	Execute deployment	DevOps

Normal Change:

Step	Action	Owner
1	Submit change request	Developer
2	Technical review	Tech Lead
3	CAB review	Change Advisory Board
4	Schedule deployment	Release Manager
5	Execute deployment	DevOps
6	Post-implementation review	All

7.3 Change Request Template

CHANGE REQUEST

Request ID: [Auto-generated]

Requestor: [Name]

Date: [Date]

CHANGE DETAILS

Description: [What is changing]

Justification: [Why this change is needed]

Risk Level: [Low/Medium/High]

IMPACT ANALYSIS

Systems Affected: [List]

Users Affected: [Count/Groups]

Downtime Required: [Yes/No, Duration]

IMPLEMENTATION PLAN

Deployment Window: [Date/Time]

Steps: [Numbered list]

Duration: [Estimated time]

ROLLBACK PLAN

Trigger Criteria: [When to rollback]

Procedure: [Steps to rollback]

Estimated Time: [Duration]

TESTING

Test Plan: [How it will be tested]

Validation: [Success criteria]

APPROVALS

Technical: [Name/Date]

Manager: [Name/Date]

CAB: [Name/Date] (if required)

7.4 Emergency Change Process

Step	Action	Timeline
1	Identify emergency	Immediate
2	Verbal approval from manager	15 min
3	Document change request	During fix
4	Implement fix	ASAP
5	Post-hoc approval	Within 24 hours
6	Retrospective	Within 1 week

8. Environment Configuration

8.1 Configuration Management

Configuration Sources:

Type	Storage	Access
Application config	Environment variables	Runtime
Secrets	Secret management service	Runtime
Feature flags	Configuration service	Runtime
Infrastructure	Infrastructure as Code	Deployment

8.2 Environment Variables

Variable	Description	Example
NODE_ENV	Environment name	production
DATABASE_URL	Database connection	(secret)
API_KEY_ENCRYPTION	Encryption key	(secret)
LOG_LEVEL	Logging verbosity	info
FEATURE_FLAGS	Enabled features	(JSON)

8.3 Secret Management

Secrets Storage:

- All secrets in dedicated secret management service
- No secrets in code or configuration files
- Secrets rotated according to policy
- Access logged and audited

Secret Types:

Secret	Rotation	Access
Database credentials	Quarterly	Application only
API keys	Annually	Application only
Encryption keys	Annually	Application only
Integration tokens	Per provider policy	Application only

8.4 Feature Flags

Flag Types:

Type	Description	Example
Release	Enable new features	NEW_DASHBOARD
Ops	Operational controls	MAINTENANCE_MODE
Experiment	A/B testing	NEW_AI_MODEL
Permission	Feature gating	PREMIUM_REPORTS

9. Infrastructure Requirements

9.1 Compute Requirements

Component	Minimum	Recommended
Application Servers	2 instances	3+ instances
CPU per instance	2 vCPU	4 vCPU
Memory per instance	4 GB	8 GB
Storage	20 GB	50 GB

9.2 Database Requirements

Aspect	Specification
Type	PostgreSQL-compatible
Minimum version	PostgreSQL 14+
Storage	100 GB minimum
IOPS	3000+
Backup	Automated daily
High Availability	Multi-AZ required

9.3 Network Requirements

Requirement	Specification
HTTPS	Required (TLS 1.3)
Load Balancer	Application load balancer
CDN	For static assets
Firewall	Web application firewall
Ports	443 (HTTPS only)

9.4 Scaling Guidelines

Metric	Trigger	Action
CPU > 70%	Sustained 5 min	Add instance
Memory > 80%	Sustained 5 min	Add instance
Response time > 1s	Sustained 5 min	Investigate/scale
Error rate > 1%	Any	Investigate

10. Monitoring & Health Checks

10.1 Health Check Endpoints

Endpoint	Purpose	Expected Response
/health	Basic health	200 OK
/health/ready	Ready for traffic	200 OK
/health/live	Process alive	200 OK
/health/db	Database connectivity	200 OK

10.2 Monitoring Metrics

Metric	Alert Threshold
HTTP 5xx rate	> 1%
Response time P95	> 2 seconds
CPU utilization	> 80%
Memory utilization	> 85%
Disk utilization	> 80%
Database connections	> 80% of max

10.3 Alerting Configuration

Severity	Response Time	Notification
Critical	5 minutes	PagerDuty + Slack
High	15 minutes	Slack + Email
Medium	1 hour	Email
Low	Next business day	Ticket

10.4 Log Aggregation

Log Type	Retention	Access
Application	30 days	Operations
Access	90 days	Security
Audit	2 years	Compliance
Debug	7 days	Development

11. Disaster Recovery

11.1 Recovery Objectives

Metric	Target
Recovery Time Objective (RTO)	4 hours
Recovery Point Objective (RPO)	1 hour

11.2 Backup Strategy

Data	Frequency	Retention
Database	Continuous + Daily snapshot	30 days
File storage	Continuous replication	30 days
Configuration	On change	90 days
Logs	Real-time	Per retention policy

11.3 Recovery Procedures

Database Recovery:

Step	Action	Time
1	Identify recovery point	5 min
2	Restore from backup	30-60 min
3	Verify data integrity	15 min
4	Update connection strings	5 min
5	Verify application	10 min

Full Environment Recovery:

Step	Action	Time
1	Provision infrastructure	30 min
2	Deploy application	20 min
3	Restore database	60 min
4	Restore file storage	30 min
5	Update DNS	5 min
6	Verify functionality	30 min

11.4 DR Testing

Test	Frequency	Scope
Backup verification	Weekly	Automated
Restore test	Monthly	Sample data
Full DR test	Annually	Complete recovery
Tabletop exercise	Quarterly	Process review

Document Control

Version	Date	Author	Changes
1.0	January 2026	Documentation Team	Initial release

This guide ensures safe and consistent deployment practices.

VIOE - Vulnerability Intelligence & Orchestration Engine Deployment & Environment Guide