HX Infrastructure Ansible - Documentation Index

Overview

This repository contains the Ansible automation for HX Infrastructure, implementing a comprehensive Infrastructure as Code (IaC) solution with advanced monitoring, security hardening, and operational excellence features.

Phase 2C: Integration & Standardization

Phase 2C focuses on integrating all components built in Phase 2A and 2B, establishing standardized processes, and implementing machine-checkable quality gates.

Key Features

- **Machine-Checkable Gates**: Automated validation of integration, performance, and security
- **Golden-Path Integration Tests**: End-to-end validation of critical workflows
- **Quantified SLOs**: Hard performance thresholds with automated measurement
- **Unified CI Pipeline**: Standardized build and deployment processes
- **Enhanced Documentation**: Consolidated and structured documentation framework

Quick Start

Prerequisites

- Python 3.11+
- Ansible 2.15+
- Git
- Access to target infrastructure

Installation

```
# Clone the repository
git clone https://github.com/hanax-ai/HX-Infrastructure-Ansible.git
cd HX-Infrastructure-Ansible

# Install dependencies
pip install -r requirements.txt
ansible-galaxy install -r requirements.yml

# Validate installation
make gate-integration
```

Basic Usage

```
# Run all quality gates
make gate-integration
make gate-performance
make gate-security

# Run golden path tests
make golden-path-all

# Deploy to development
make deploy-dev

# Run performance benchmarks
make benchmark
```

Documentation Structure

Core Documentation

- README.md (../README.md) - Project overview and quick start
- ARCHITECTURE.md (./ARCHITECTURE.md) - System architecture and design
- SECURITY.md (../SECURITY.md) - Security policies and procedures
- DEVELOPMENT_GUIDE.md (./DEVELOPMENT_GUIDE.md) - Development workflows and standards

Phase 2C Documentation

- Phase 2C Plan (./phase2c_plan.md) - Detailed implementation plan with day-by-day acceptance criteria
- Validation Report Template (./validation-report.md) - Template for validation reporting
- Removal Matrix (./removal_matrix.md) - Legacy component removal mapping
- SLO Definitions (./slo_definitions.md) - Service Level Objectives and metrics

Operational Documentation

- Deployment Guide (./deployment_guide.md) - Step-by-step deployment procedures
- Monitoring Guide (./monitoring_guide.md) - Monitoring setup and troubleshooting
- Troubleshooting Guide (./troubleshooting_guide.md) - Common issues and solutions
- Runbooks (./runbooks/) - Operational procedures and emergency responses

Technical Documentation

- Role Documentation (./roles/) - Individual role documentation
- Playbook Documentation (./playbooks/) - Playbook usage and examples
- API Documentation (./api/) - API specifications and examples
- Testing Documentation (./testing/) - Testing strategies and procedures

Quality Gates

Integration Gate (`make gate-integration`)

Validates:
- Ansible syntax and structure
- Role dependencies

- Inventory configuration
- Template rendering
- Vault file security
- Golden path integration

SLO: Must pass 100% of checks

Performance Gate (`make gate-performance`)

Validates:
- P95 deploy time ≤ 8 minutes
- Playbook runtime ≤ 90 seconds
- Role execution ≤ 30 seconds
- Template render ≤ 5 seconds
- Vault decrypt ≤ 2 seconds

SLO: Must meet all performance thresholds

Security Gate (`make gate-security`)

Validates:
- Vault encryption compliance
- Sensitive data exposure
- SSH key security
- File permissions
- Security best practices
- Compliance requirements

SLO: Must pass 100% of security checks

Golden Path Tests

Blue-Green Deployment (`tests/golden_path/blue_green.sh`)

End-to-end validation of:
- Blue environment deployment
- Health check validation
- Traffic switching
- Green environment deployment
- Rollback procedures
- Performance metrics

Monitoring Pipeline (`tests/golden_path/monitoring.sh`)

End-to-end validation of:
- Metric collection
- Dashboard rendering
- Alert evaluation
- Notification delivery
- Performance validation

Self-Healing System (`tests/golden_path/self_healing.sh`)

End-to-end validation of:
- Fault detection

- Automated recovery
- System convergence
- Rollback mechanisms
- Performance metrics

Service Level Objectives (SLOs)

Metric	Threshold	Measurement
Deploy Time (P95)	≤ 8 minutes	End-to-end deployment
Playbook Runtime	≤ 90 seconds	Individual playbook execution
Role Execution	≤ 30 seconds	Per role execution time
Template Render	≤ 5 seconds	Template processing time
Vault Decrypt	≤ 2 seconds	Vault file access time
Health Check	≤ 10 seconds	Service health validation
Alert Response	≤ 5 minutes	Alert to notification time
Recovery Time	≤ 60 seconds	Fault to recovery completion

CI/CD Pipeline

The CI/CD pipeline implements the following stages:

1. **Integration Gate** - Syntax and structure validation
2. **Performance Gate** - SLO compliance validation
3. **Security Gate** - Security compliance validation
4. **Golden Path Tests** - End-to-end workflow validation
5. **Documentation Validation** - Documentation completeness
6. **Phase 2C Completion Gate** - Overall readiness validation

Branch Protection

The following contexts are required for merge:

- `Integration Gate`
- `Performance Gate`
- `Security Gate`
- `Golden Path Tests`
- `Lint and Syntax Check`
- `Security Scan`
- `Monitoring Validation`
- `Documentation Validation`

Development Workflow

1. **Feature Development**
   - Create feature branch from `develop`
   - Implement changes following coding standards
   - Run local quality gates
   - Submit pull request

2. **Quality Validation**
   - Automated CI pipeline execution
   - All gates must pass
   - Peer review required
   - Documentation updates required

3. **Integration Testing**
   - Golden path tests execution
   - Performance validation
   - Security compliance check

4. **Deployment**
   - Merge to `main` branch
   - Automated deployment to staging
   - Production deployment approval
   - Post-deployment validation

Monitoring and Alerting

Key Metrics

- Infrastructure health metrics
- Application performance metrics
- Security compliance metrics
- Operational metrics

Alert Channels

- Slack notifications
- Email alerts
- PagerDuty integration
- Webhook notifications

Dashboards

- Infrastructure overview
- Application performance
- Security compliance
- Operational metrics

Security

Security Policies

- All secrets must be encrypted with Ansible Vault

- SSH keys must have 600 permissions
- No hardcoded credentials in code
- Regular security scans required

Compliance

- SOC 2 Type II compliance
- GDPR compliance
- Industry-specific requirements
- Regular audit procedures

Support and Troubleshooting

Getting Help

1. Check the Troubleshooting Guide (./troubleshooting_guide.md)
2. Review Common Issues (./common_issues.md)
3. Check the FAQ (./faq.md)
4. Contact the infrastructure team

Emergency Procedures

- Incident Response (./runbooks/incident_response.md)
- Emergency Rollback (./runbooks/emergency_rollback.md)
- Security Incident (./runbooks/security_incident.md)
- Disaster Recovery (./runbooks/disaster_recovery.md)

Contributing

Please read our Contributing Guide (./CONTRIBUTING.md) for details on:

- Code of conduct
- Development process
- Coding standards
- Testing requirements
- Documentation standards

License

This project is licensed under the MIT License - see the LICENSE (../LICENSE) file for details.

Changelog

See CHANGELOG.md (./CHANGELOG.md) for a detailed history of changes.

Last Updated: September 26, 2025
Version: Phase 2C
Maintainer: HX Infrastructure Team