HX Infrastructure Visual Documentation

Comprehensive System Diagrams and Visual Architecture

This document provides detailed visual representations of the HX Infrastructure Ansible system architecture, workflows, and operational procedures using Mermaid diagrams.

Table of Contents

System Architecture Diagrams

1. Overall System Architecture

```
graph TB
    subgraph "HX Infrastructure Ansible Platform"
        subgraph "Control Layer"
            A[Ansible Controller]
            B[Git Repository]
            C[CI/CD Pipeline]
        end

        subgraph "Automation Layer"
            D[Playbooks]
            E[Standardized Roles]
            F[Inventories]
            G[Variables & Vault]
        end

        subgraph "Target Infrastructure"
            H[Web Servers]
            I[Application Servers]
            J[Database Servers]
            K[Load Balancers]
        end

        subgraph "External Systems"
            L[Active Directory]
            M[Certificate Authority]
            N[Monitoring Systems]
            O[Backup Storage]
        end
    end

    B --> A
    C --> A
    A --> D
    D --> E
    E --> F
    F --> G

    A --> H
    A --> I
    A --> J
    A --> K

    E --> L
    E --> M
    H --> N
    I --> N
    J --> N
    K --> N

    A --> O

    style A fill:#e1f5fe
    style E fill:#f3e5f5
    style H fill:#e8f5e8
    style L fill:#fff3e0
```

2. Role Architecture and Dependencies

```
graph TB
    subgraph "Standardized Roles Ecosystem"
        subgraph "Core Infrastructure Roles"
            A[hx_ca_trust_standardized]
            B[hx_domain_join_standardized]
            C[hx_pg_auth_standardized]
        end

        subgraph "Application Roles"
            D[hx_webui_install_standardized]
            E[hx_litellm_proxy_standardized]
        end

        subgraph "Supporting Roles"
            F[common]
            G[security_hardening]
            H[monitoring]
        end

        subgraph "External Dependencies"
            I[Certificate Authority]
            J[Active Directory]
            K[PostgreSQL Cluster]
            L[Web UI Components]
            M[LiteLLM Services]
        end
    end

    F --> A
    F --> B
    F --> C
    F --> D
    F --> E

    A --> I
    B --> J
    C --> K
    D --> L
    E --> M

    G --> A
    G --> B
    G --> C
    G --> D
    G --> E

    H --> A
    H --> B
    H --> C
    H --> D
    H --> E

    style A fill:#ffebee
    style B fill:#e8f5e8
    style C fill:#e3f2fd
    style D fill:#fff3e0
    style E fill:#fce4ec
    style F fill:#f1f8e9
```

3. Multi-Environment Architecture

```
graph LR
    subgraph "Development Environment"
        A1[Dev Web Servers]
        A2[Dev App Servers]
        A3[Dev Database]
        A4[Dev Load Balancer]
    end

    subgraph "Testing Environment"
        B1[Test Web Servers]
        B2[Test App Servers]
        B3[Test Database]
        B4[Test Load Balancer]
    end

    subgraph "Staging Environment"
        C1[Stage Web Servers]
        C2[Stage App Servers]
        C3[Stage Database]
        C4[Stage Load Balancer]
    end

    subgraph "Production Environment"
        D1[Prod Web Servers]
        D2[Prod App Servers]
        D3[Prod Database Cluster]
        D4[Prod Load Balancers]
    end

    subgraph "Shared Services"
        E1[Certificate Authority]
        E2[Active Directory]
        E3[Monitoring Stack]
        E4[Backup Systems]
    end

    A1 --> B1
    A2 --> B2
    A3 --> B3
    A4 --> B4

    B1 --> C1
    B2 --> C2
    B3 --> C3
    B4 --> C4

    C1 --> D1
    C2 --> D2
    C3 --> D3
    C4 --> D4

    E1 --> A1
    E1 --> B1
    E1 --> C1
    E1 --> D1

    E2 --> A2
    E2 --> B2
    E2 --> C2
    E2 --> D2

    E3 --> A1
    E3 --> B1
    E3 --> C1
    E3 --> D1

    E4 --> D3

    style A1 fill:#e8f5e8
    style B1 fill:#fff3e0
    style C1 fill:#e3f2fd
    style D1 fill:#ffebee
```

Deployment Flow Diagrams

4. Standard Deployment Workflow

```
sequenceDiagram
    participant Dev as Developer
    participant Git as Git Repository
    participant CI as CI/CD Pipeline
    participant Ansible as Ansible Controller
    participant Staging as Staging Environment
    participant Prod as Production Environment
    participant Monitor as Monitoring

    Dev->>Git: Push Code Changes
    Git->>CI: Trigger Pipeline

    CI->>CI: Run Quality Gates
    Note over CI: Lint, Security Scan, Tests

    CI->>Ansible: Deploy to Staging
    Ansible->>Staging: Execute Playbooks
    Staging->>Ansible: Deployment Status

    Ansible->>CI: Staging Results
    CI->>CI: Run Integration Tests

    CI->>Dev: Request Production Approval
    Dev->>CI: Approve Production Deploy

    CI->>Ansible: Deploy to Production
    Ansible->>Prod: Execute Playbooks
    Prod->>Ansible: Deployment Status

    Ansible->>Monitor: Update Monitoring
    Monitor->>Ansible: Health Check Results

    Ansible->>CI: Production Results
    CI->>Dev: Deployment Complete
```

5. Blue-Green Deployment Process

```
graph TB
    subgraph "Load Balancer"
        A[Traffic Router]
    end

    subgraph "Blue Environment (Current)"
        B1[Blue Web Server 1]
        B2[Blue Web Server 2]
        B3[Blue App Server 1]
        B4[Blue App Server 2]
        B5[Blue Database]
    end

    subgraph "Green Environment (New)"
        G1[Green Web Server 1]
        G2[Green Web Server 2]
        G3[Green App Server 1]
        G4[Green App Server 2]
        G5[Green Database]
    end

    subgraph "Deployment Process"
        D1[Deploy to Green]
        D2[Health Check Green]
        D3[Switch Traffic]
        D4[Monitor Green]
        D5[Decommission Blue]
    end

    A -->|100% Traffic| B1
    A -->|100% Traffic| B2

    D1 --> G1
    D1 --> G2
    D1 --> G3
    D1 --> G4

    D2 --> G1
    D2 --> G2

    D3 --> A
    A -.->|Switch to| G1
    A -.->|Switch to| G2

    D4 --> G1
    D4 --> G2

    D5 --> B1
    D5 --> B2

    style B1 fill:#e3f2fd
    style B2 fill:#e3f2fd
    style G1 fill:#e8f5e8
    style G2 fill:#e8f5e8
    style A fill:#fff3e0
```

6. Rolling Deployment Strategy

```
graph TB
    subgraph "Rolling Deployment Process"
        A[Start Deployment]
        B[Select First Batch]
        C[Remove from Load Balancer]
        D[Deploy New Version]
        E[Health Check]
        F{Health Check Pass?}
        G[Add to Load Balancer]
        H{More Batches?}
        I[Select Next Batch]
        J[Deployment Complete]
        K[Rollback Previous Batch]
        L[Alert Operations]
    end

    A --> B
    B --> C
    C --> D
    D --> E
    E --> F
    F -->|Yes| G
    F -->|No| K
    G --> H
    H -->|Yes| I
    H -->|No| J
    I --> C
    K --> L

    subgraph "Server Batches"
        S1[Server 1]
        S2[Server 2]
        S3[Server 3]
        S4[Server 4]
        S5[Server 5]
        S6[Server 6]
    end

    B -.-> S1
    I -.-> S2
    I -.-> S3
    I -.-> S4

    style A fill:#e8f5e8
    style J fill:#e8f5e8
    style K fill:#ffebee
    style L fill:#ffebee
```

Security Architecture

7. Defense in Depth Security Model

```
graph TB
    subgraph "Security Layers"
        subgraph "Perimeter Security"
            A1[Firewall Rules]
            A2[DDoS Protection]
            A3[WAF]
        end

        subgraph "Network Security"
            B1[Network Segmentation]
            B2[VPN Access]
            B3[Network Monitoring]
            B4[Intrusion Detection]
        end

        subgraph "Host Security"
            C1[OS Hardening]
            C2[Endpoint Protection]
            C3[Patch Management]
            C4[Access Controls]
        end

        subgraph "Application Security"
            D1[Authentication]
            D2[Authorization]
            D3[Input Validation]
            D4[Session Management]
        end

        subgraph "Data Security"
            E1[Encryption at Rest]
            E2[Encryption in Transit]
            E3[Key Management]
            E4[Data Classification]
        end

        subgraph "Monitoring & Response"
            F1[SIEM]
            F2[Log Analysis]
            F3[Incident Response]
            F4[Forensics]
        end
    end

    A1 --> B1
    A2 --> B2
    A3 --> B3

    B1 --> C1
    B2 --> C2
    B3 --> C3
    B4 --> C4

    C1 --> D1
    C2 --> D2
    C3 --> D3
    C4 --> D4

    D1 --> E1
    D2 --> E2
    D3 --> E3
    D4 --> E4
    E1 --> F1
    E2 --> F2
    E3 --> F3
    E4 --> F4

    style A1 fill:#ffebee
    style B1 fill:#fff3e0
    style C1 fill:#e8f5e8
    style D1 fill:#e3f2fd
    style E1 fill:#f3e5f5
    style F1 fill:#fce4ec
```

8. Certificate Management Workflow

```
sequenceDiagram
    participant CA as Certificate Authority
    participant Ansible as Ansible Controller
    participant Vault as Ansible Vault
    participant Target as Target Servers
    participant Monitor as Certificate Monitor

    Note over CA,Monitor: Certificate Lifecycle Management

    CA->>Ansible: Generate Certificate
    Ansible->>Vault: Store Certificate Securely

    Ansible->>Target: Deploy Certificate
    Target->>Ansible: Confirm Installation

    Ansible->>Monitor: Register Certificate
    Monitor->>Monitor: Track Expiration

    Monitor->>Ansible: Certificate Expiring (30 days)
    Ansible->>CA: Request Renewal
    CA->>Ansible: New Certificate

    Ansible->>Vault: Update Certificate
    Ansible->>Target: Deploy New Certificate
    Target->>Ansible: Confirm Update

    Ansible->>Monitor: Update Certificate Info

    Note over CA,Monitor: Automated Renewal Complete
```

Monitoring and Observability

9. Monitoring Architecture

```
graph TB
    subgraph "Data Collection Layer"
        A1[System Metrics]
        A2[Application Metrics]
        A3[Log Collection]
        A4[Trace Collection]
        A5[Security Events]
    end

    subgraph "Processing Layer"
        B1[Prometheus]
        B2[Elasticsearch]
        B3[Jaeger]
        B4[SIEM]
    end

    subgraph "Storage Layer"
        C1[Time Series DB]
        C2[Log Storage]
        C3[Trace Storage]
        C4[Event Storage]
    end

    subgraph "Visualization Layer"
        D1[Grafana Dashboards]
        D2[Kibana]
        D3[Jaeger UI]
        D4[Security Dashboard]
    end

    subgraph "Alerting Layer"
        E1[AlertManager]
        E2[PagerDuty]
        E3[Slack]
        E4[Email]
    end

    A1 --> B1 --> C1 --> D1
    A2 --> B1 --> C1 --> D1
    A3 --> B2 --> C2 --> D2
    A4 --> B3 --> C3 --> D3
    A5 --> B4 --> C4 --> D4

    B1 --> E1
    B2 --> E1
    B4 --> E1

    E1 --> E2
    E1 --> E3
    E1 --> E4

    style A1 fill:#e8f5e8
    style B1 fill:#e3f2fd
    style C1 fill:#fff3e0
    style D1 fill:#f3e5f5
    style E1 fill:#ffebee
```

10. Alerting and Escalation Flow

```
graph TB
    A[Metric Threshold Exceeded] --> B{Severity Level}

    B -->|Low| C[Log Alert]
    B -->|Medium| D[Email Notification]
    B -->|High| E[Slack Alert]
    B -->|Critical| F[PagerDuty Alert]

    C --> G[Alert Dashboard]
    D --> H[Team Email List]
    E --> I[Operations Channel]
    F --> J[On-Call Engineer]

    J --> K{Acknowledged?}
    K -->|No| L[Escalate to Manager]
    K -->|Yes| M[Investigation Started]

    L --> N[Manager Notification]
    N --> O[Executive Escalation]

    M --> P{Issue Resolved?}
    P -->|No| Q[Continue Investigation]
    P -->|Yes| R[Close Alert]

    Q --> S[Update Status]
    S --> P

    R --> T[Post-Incident Review]

    style A fill:#ffebee
    style F fill:#ffebee
    style J fill:#fff3e0
    style R fill:#e8f5e8
```

Data Flow Diagrams

11. Configuration Data Flow

```
graph LR
    subgraph "Source Control"
        A[Git Repository]
        B[Branch Protection]
        C[Code Review]
    end

    subgraph "CI/CD Pipeline"
        D[Quality Gates]
        E[Security Scan]
        F[Testing]
    end

    subgraph "Ansible Controller"
        G[Playbook Execution]
        H[Variable Processing]
        I[Template Rendering]
    end

    subgraph "Target Systems"
        J[Configuration Files]
        K[Service Restart]
        L[Validation]
    end

    subgraph "Monitoring"
        M[Configuration Drift]
        N[Compliance Check]
        O[Audit Logging]
    end

    A --> D
    B --> D
    C --> D

    D --> G
    E --> G
    F --> G

    G --> J
    H --> J
    I --> J

    J --> M
    K --> N
    L --> O

    M --> A
    N --> A
    O --> A

    style A fill:#e8f5e8
    style G fill:#e3f2fd
    style J fill:#fff3e0
    style M fill:#f3e5f5
```

12. Secrets Management Flow

```
sequenceDiagram
    participant Dev as Developer
    participant Vault as Ansible Vault
    participant Controller as Ansible Controller
    participant Target as Target System
    participant HSM as Hardware Security Module

    Dev->>Vault: Encrypt Secret
    Note over Vault: AES-256 Encryption

    Vault->>Controller: Store Encrypted Secret
    Controller->>HSM: Request Decryption Key
    HSM->>Controller: Provide Key

    Controller->>Controller: Decrypt Secret in Memory
    Note over Controller: Temporary Decryption

    Controller->>Target: Deploy Secret (Encrypted Transit)
    Target->>Target: Store Secret Securely

    Controller->>Controller: Clear Memory
    Note over Controller: Security Cleanup

    Target->>Controller: Confirm Deployment
    Controller->>Dev: Deployment Complete
```

Network Architecture

13. Network Topology

```
graph TB
    subgraph "Internet"
        A[External Users]
        B[External APIs]
    end

    subgraph "DMZ"
        C[Load Balancer]
        D[Web Application Firewall]
        E[Reverse Proxy]
    end

    subgraph "Application Tier"
        F[Web Server 1]
        G[Web Server 2]
        H[App Server 1]
        I[App Server 2]
    end

    subgraph "Data Tier"
        J[Primary Database]
        K[Secondary Database]
        L[Cache Cluster]
    end

    subgraph "Management Network"
        M[Ansible Controller]
        N[Monitoring Server]
        O[Backup Server]
    end

    subgraph "Security Services"
        P[Certificate Authority]
        Q[Active Directory]
        R[SIEM]
    end

    A --> C
    B --> D
    C --> E
    D --> E

    E --> F
    E --> G
    F --> H
    G --> I

    H --> J
    I --> K
    H --> L
    I --> L

    M --> F
    M --> G
    M --> H
    M --> I
    M --> J
    M --> K

    N --> F
    N --> G
    N --> H
    N --> I

    O --> J
    O --> K

    P --> F
    P --> G
    Q --> H
    Q --> I
    R --> N

    style C fill:#ffebee
    style M fill:#e3f2fd
    style J fill:#e8f5e8
    style P fill:#fff3e0
```

CI/CD Pipeline Visualization

14. Complete CI/CD Pipeline

```
graph TB
    subgraph "Source Control"
        A[Developer Commit]
        B[Pull Request]
        C[Code Review]
        D[Merge to Main]
    end

    subgraph "Build Stage"
        E[Syntax Check]
        F[Lint Validation]
        G[Security Scan]
        H[Unit Tests]
    end

    subgraph "Test Stage"
        I[Integration Tests]
        J[Security Tests]
        K[Performance Tests]
        L[Compliance Tests]
    end

    subgraph "Deploy Stage"
        M[Deploy to Dev]
        N[Deploy to Test]
        O[Deploy to Staging]
        P[Deploy to Production]
    end

    subgraph "Validation Stage"
        Q[Health Checks]
        R[Smoke Tests]
        S[Monitoring Setup]
        T[Rollback if Failed]
    end

    A --> B
    B --> C
    C --> D

    D --> E
    E --> F
    F --> G
    G --> H

    H --> I
    I --> J
    J --> K
    K --> L

    L --> M
    M --> N
    N --> O
    O --> P

    P --> Q
    Q --> R
    R --> S
    S --> T

    T -.->|Rollback| O
    T -.->|Rollback| N
    T -.->|Rollback| M

    style A fill:#e8f5e8
    style P fill:#ffebee
    style T fill:#fff3e0
```

Disaster Recovery Workflows

15. Disaster Recovery Process

```
graph TB
    A[Disaster Detected] --> B{Severity Assessment}

    B -->|Low| C[Automated Recovery]
    B -->|Medium| D[Guided Recovery]
    B -->|High| E[Manual Recovery]
    B -->|Critical| F[Emergency Procedures]

    C --> G[Service Restart]
    D --> H[Partial Restore]
    E --> I[Full System Restore]
    F --> J[Disaster Declaration]

    G --> K[Health Check]
    H --> L[Data Validation]
    I --> M[System Validation]
    J --> N[Activate DR Site]

    K --> O{Recovery Successful?}
    L --> O
    M --> O
    N --> P[Failover Complete]

    O -->|Yes| Q[Resume Operations]
    O -->|No| R[Escalate Recovery]

    R --> S[Expert Team]
    S --> T[Advanced Recovery]
    T --> O

    Q --> U[Post-Incident Review]
    P --> V[Primary Site Recovery]

    V --> W[Failback Planning]
    W --> X[Failback Execution]
    X --> U

    style A fill:#ffebee
    style F fill:#ffebee
    style Q fill:#e8f5e8
    style U fill:#e3f2fd
```

16. Backup and Restore Workflow

```
sequenceDiagram
    participant Scheduler as Backup Scheduler
    participant Ansible as Ansible Controller
    participant DB as Database Server
    participant App as Application Server
    participant Storage as Backup Storage
    participant Monitor as Monitoring

    Note over Scheduler,Monitor: Daily Backup Process

    Scheduler->>Ansible: Trigger Backup Job
    Ansible->>DB: Create Database Backup
    DB->>Ansible: Backup Complete

    Ansible->>App: Create Application Backup
    App->>Ansible: Backup Complete

    Ansible->>Storage: Store Backups
    Storage->>Ansible: Storage Confirmed

    Ansible->>Monitor: Update Backup Status
    Monitor->>Monitor: Validate Backup Integrity

    Note over Scheduler,Monitor: Restore Process (When Needed)

    Ansible->>Storage: Retrieve Backup
    Storage->>Ansible: Backup Retrieved

    Ansible->>DB: Stop Database Service
    Ansible->>DB: Restore Database
    DB->>Ansible: Restore Complete

    Ansible->>App: Stop Application
    Ansible->>App: Restore Application
    App->>Ansible: Restore Complete

    Ansible->>DB: Start Database Service
    Ansible->>App: Start Application

    Ansible->>Monitor: Validate Restore
    Monitor->>Ansible: Validation Results
```

This comprehensive visual documentation provides detailed diagrams for understanding the HX Infrastructure Ansible system architecture, workflows, and operational procedures. Each diagram serves as a reference for system design, troubleshooting, and operational planning.