Deployment Runbook - HX Infrastructure Phase 3.4

Overview

This runbook provides step-by-step procedures for executing production deployments using the Phase 3.4 Production Operations and Maintenance Automation framework.

Prerequisites

System Requirements

- Ansible 2.9+ installed
- Access to production inventory
- Valid SSH keys for target hosts
- Backup verification completed within last 24 hours

Pre-deployment Checklist

- [ ] Deployment approved through change management
- [ ] Staging environment testing completed
- [ ] Rollback plan documented and tested
- [ ] Monitoring systems operational
- [ ] On-call team notified

Deployment Strategies

Blue-Green Deployment

When to Use

- Zero-downtime requirement
- Full application stack updates
- Database schema changes
- Major version upgrades

Execution Steps

1. **Pre-deployment Validation**
   ```
   bash
   ansible-playbook -i inventory/production \
     playbooks/production/deployment/blue_green_deploy.yml \
     --tags "validate" --check
   ```

2. **Execute Blue-Green Deployment**
   ```
   bash
   ansible-playbook -i inventory/production \
     playbooks/production/deployment/blue_green_deploy.yml \
     -e "app_version=v1.2.3" \
     -e "target_environment=production"
   ```

3. **Monitor Health Checks**
   - Green environment health: `http://host:8081/health`
   - Monitor logs: `/var/log/hx-infrastructure/health-*.json`
   - Verify metrics in Grafana dashboard

4. **Traffic Switch Verification**
   ```bash
   # Verify traffic routing
   curl -H “Host: production.hx-infrastructure.local” http://load-balancer/health

# Check nginx upstream configuration
nginx -T | grep upstream
```

Rollback Procedure

```bash
# Emergency rollback
ansible-playbook -i inventory/production \
  playbooks/production/deployment/rollback.yml \
  -e "rollback_to=blue" \
  -e "emergency=true"
```

Canary Deployment

When to Use

- Gradual rollout preferred
- A/B testing requirements
- Risk mitigation for new features
- Performance impact assessment

Execution Steps

1. **Deploy to Canary Instances**
   ```bash
   ansible-playbook -i inventory/production \
     playbooks/production/deployment/canary_deploy.yml \
     -e "app_version=v1.2.3" \
     -e "canary_traffic_percentage=10" \
     -e "canary_monitoring_time=900"
   ```

2. **Monitor Canary Performance**
   - Canary metrics: `http://canary-host:8082/metrics`
   - Error rate comparison
   - Response time analysis
   - User experience metrics

3. **Evaluate Canary Success**
   ```bash
   # Check canary evaluation results
   cat /var/log/hx-infrastructure/canary-evaluation-*.json
   ```

4. **Proceed with Full Deployment**
   ```bash
   # If canary successful
   ```

```
ansible-playbook -i inventory/production \
  playbooks/production/deployment/canary_deploy.yml \
  -e "proceed_full_deployment=true"
```

Monitoring During Deployment

Key Metrics to Watch

- **Response Time**: < 2000ms
- **Error Rate**: < 5%
- **CPU Usage**: < 80%
- **Memory Usage**: < 85%
- **Disk Usage**: < 90%

Monitoring Commands

```
# Real-time health check
./scripts/automation/monitoring/health_check.sh --type all --format summary

# System resource monitoring
watch -n 5 'free -h && df -h && uptime'

# Application logs
tail -f /var/log/hx-infrastructure/application.log | grep ERROR
```

Dashboard URLs

- **Grafana**: http://monitoring-host:3000/d/hx-infrastructure
- **Prometheus**: http://monitoring-host:9090/targets
- **Application Health**: http://app-host/health

Troubleshooting

Common Issues

Deployment Fails at Health Check

```
# Check service status
systemctl status nginx postgresql docker

# Verify application logs
journalctl -u hx-infrastructure-app -f

# Check network connectivity
curl -v http://localhost:8080/health
```

Traffic Switch Fails

```
# Verify nginx configuration
nginx -t

# Check upstream servers
curl -H "Host: internal" http://localhost/health

# Reload nginx configuration
systemctl reload nginx
```

Database Connection Issues

```
# Test database connectivity
psql -h db-host -U postgres -d hx_infrastructure -c "SELECT 1;"

# Check database logs
tail -f /var/log/postgresql/postgresql-*.log
```

Emergency Procedures

Immediate Rollback

```
# Stop new deployment
pkill -f "ansible-playbook.*deployment"

# Execute emergency rollback
ansible-playbook -i inventory/production \
  playbooks/production/deployment/emergency_rollback.yml \
  -e "rollback_reason='deployment_failure'"
```

Service Recovery

```
# Restart all critical services
ansible-playbook -i inventory/production \
  playbooks/production/operations/service_management.yml \
  -e "operation=restart_critical"
```

Post-Deployment

Verification Steps

1. **Health Check Verification**
   `bash`
   `ansible-playbook -i inventory/production \`
   `playbooks/production/operations/health_monitoring.yml`

2. **Performance Validation**
   - Monitor response times for 30 minutes
   - Check error rates in application logs
   - Verify database performance metrics

3. **User Acceptance Testing**
   - Execute smoke tests

- Verify critical user journeys
- Check integration points

Documentation Updates

- [ ] Update deployment log
- [ ] Record any issues encountered
- [ ] Update runbook with lessons learned
- [ ] Notify stakeholders of completion

Automation Scripts

Quick Deployment Script

```
#!/bin/bash
# Quick deployment wrapper
./scripts/automation/deployment/deploy.sh \
  --strategy blue_green \
  --version $1 \
  --environment production
```

Health Check Script

```
#!/bin/bash
# Continuous health monitoring during deployment
while true; do
  ./scripts/automation/monitoring/health_check.sh --format summary
  sleep 30
done
```

Contact Information

Escalation Contacts

- **Primary On-Call**: +1-555-0101
- **Secondary On-Call**: +1-555-0102
- **Engineering Manager**: +1-555-0103

Communication Channels

- **Slack**: #hx-infrastructure-ops
- **Email**: ops-team@hx-infrastructure.local
- **PagerDuty**: HX Infrastructure Service

References

- Architecture Documentation (../../ARCHITECTURE.md)
- Security Guidelines (../../SECURITY.md)
- Monitoring Setup (../dashboards/monitoring_setup.md)
- Incident Response Procedures (incident_response_runbook.md)