# PRODUCTION DEPLOYMENT READINESS GUIDE

## SKIDS Advanced Integration Infrastructure

### \*\*OVERVIEW\*\*

This document outlines the comprehensive production deployment strategy for the SKIDS Advanced integration infrastructure, including environment setup, deployment procedures, monitoring systems, and security compliance requirements.

## \*\*ENVIRONMENT SETUP PROCEDURES\*\*

### \*\*Environment Hierarchy\*\*

Development → Staging → Production
↓ ↓ ↓
localhost staging. app.
:3001 skids. skids.
clinic clinic

### \*\*Staging Environment Setup\*\*

#### \*\*Infrastructure Requirements\*\*

# staging-infrastructure.yml
Server Specifications:
CPU: 4 vCPUs
RAM: 8GB
Storage: 100GB SSD
Network: 1Gbps
OS: Ubuntu 22.04 LTS
Database:
Type: PostgreSQL 15
CPU: 2 vCPUs
RAM: 4GB
Storage: 50GB SSD
Backup: Daily automated
CDN:
Provider: Cloudflare
Cache: Global edge locations
SSL: Full (strict)

#### \*\*Environment Variables - Staging\*\*

# =============================================================================
# STAGING ENVIRONMENT CONFIGURATION
# =============================================================================
# Environment
NODE\_ENV=staging
NEXT\_PUBLIC\_APP\_ENV=staging
NEXT\_PUBLIC\_API\_BASE\_URL=https://staging-api.skids.clinic
# Database
DATABASE\_URL=postgresql://skids\_staging:secure\_password@staging-db.skids.clinic:5432/skids\_staging
# Authentication (Clerk Staging)
NEXT\_PUBLIC\_CLERK\_PUBLISHABLE\_KEY=pk\_test\_staging\_key\_here
CLERK\_SECRET\_KEY=sk\_test\_staging\_secret\_here
NEXT\_PUBLIC\_CLERK\_SIGN\_IN\_URL=/sign-in
NEXT\_PUBLIC\_CLERK\_SIGN\_UP\_URL=/sign-up
# Payment Gateways (Sandbox)
RAZORPAY\_KEY\_ID=rzp\_test\_staging\_key
RAZORPAY\_KEY\_SECRET=staging\_secret\_key
RAZORPAY\_WEBHOOK\_SECRET=staging\_webhook\_secret
RAZORPAY\_ENVIRONMENT=sandbox
STRIPE\_PUBLISHABLE\_KEY=pk\_test\_staging\_stripe\_key
STRIPE\_SECRET\_KEY=sk\_test\_staging\_stripe\_secret
STRIPE\_WEBHOOK\_SECRET=whsec\_staging\_webhook\_secret
STRIPE\_ENVIRONMENT=sandbox
# External Services (Staging)
NUTREEAI\_API\_KEY=staging\_nutreeai\_key
NUTREEAI\_BASE\_URL=https://staging.nutreeai.netlify.app
SHANTI\_API\_KEY=staging\_shanti\_key
SHANTI\_BASE\_URL=https://staging-pranayama.run.app
# Monitoring & Logging
SENTRY\_DSN=https://staging-sentry-dsn@sentry.io/project
LOG\_LEVEL=debug
ENABLE\_PERFORMANCE\_MONITORING=true
# Feature Flags
ENABLE\_VENDOR\_ONBOARDING=true
ENABLE\_AI\_RECOMMENDATIONS=true
ENABLE\_ROI\_ANALYTICS=true
ENABLE\_UNIFIED\_ANALYTICS=true
USE\_MOCK\_PAYMENTS=false
USE\_MOCK\_AI=false

### \*\*Production Environment Setup\*\*

#### \*\*Infrastructure Requirements\*\*

# production-infrastructure.yml
Load Balancer:
Type: Application Load Balancer
SSL: ACM Certificate
Health Checks: Enabled
Auto Scaling: 2-10 instances
Application Servers:
Count: 2 (minimum)
CPU: 8 vCPUs
RAM: 16GB
Storage: 200GB SSD
Network: 10Gbps
OS: Ubuntu 22.04 LTS
Database Cluster:
Primary: PostgreSQL 15
CPU: 8 vCPUs
RAM: 32GB
Storage: 500GB SSD
Read Replica: 1 instance
Backup: Continuous + Daily snapshots
Encryption: At rest and in transit
Cache Layer:
Type: Redis Cluster
CPU: 4 vCPUs
RAM: 8GB
Nodes: 3 (1 primary, 2 replicas)
CDN:
Provider: Cloudflare Pro
Cache: Global edge locations
SSL: Full (strict)
DDoS Protection: Enabled
WAF: Enabled

#### \*\*Environment Variables - Production\*\*

# =============================================================================
# PRODUCTION ENVIRONMENT CONFIGURATION
# =============================================================================
# Environment
NODE\_ENV=production
NEXT\_PUBLIC\_APP\_ENV=production
NEXT\_PUBLIC\_API\_BASE\_URL=https://api.skids.clinic
# Database (Encrypted)
DATABASE\_URL=postgresql://skids\_prod:${DB\_PASSWORD}@prod-db-cluster.skids.clinic:5432/skids\_production
# Authentication (Clerk Production)
NEXT\_PUBLIC\_CLERK\_PUBLISHABLE\_KEY=${CLERK\_PUBLISHABLE\_KEY}
CLERK\_SECRET\_KEY=${CLERK\_SECRET\_KEY}
# Payment Gateways (Live)
RAZORPAY\_KEY\_ID=${RAZORPAY\_LIVE\_KEY\_ID}
RAZORPAY\_KEY\_SECRET=${RAZORPAY\_LIVE\_SECRET}
RAZORPAY\_WEBHOOK\_SECRET=${RAZORPAY\_WEBHOOK\_SECRET}
RAZORPAY\_ENVIRONMENT=production
STRIPE\_PUBLISHABLE\_KEY=${STRIPE\_LIVE\_PUBLISHABLE\_KEY}
STRIPE\_SECRET\_KEY=${STRIPE\_LIVE\_SECRET\_KEY}
STRIPE\_WEBHOOK\_SECRET=${STRIPE\_WEBHOOK\_SECRET}
STRIPE\_ENVIRONMENT=production
# External Services (Production)
NUTREEAI\_API\_KEY=${NUTREEAI\_PRODUCTION\_KEY}
NUTREEAI\_BASE\_URL=https://nutreeai.netlify.app
SHANTI\_API\_KEY=${SHANTI\_PRODUCTION\_KEY}
SHANTI\_BASE\_URL=https://pranayama-coach-shanti-969652507861.us-west1.run.app
# Security
ENCRYPTION\_KEY=${ENCRYPTION\_KEY\_32\_CHARS}
JWT\_SECRET=${JWT\_SECRET\_STRONG}
WEBHOOK\_SECRET\_KEY=${WEBHOOK\_SECRET\_KEY}
# Monitoring & Logging
SENTRY\_DSN=${SENTRY\_PRODUCTION\_DSN}
LOG\_LEVEL=info
ENABLE\_PERFORMANCE\_MONITORING=true
ENABLE\_ERROR\_TRACKING=true
# Feature Flags
ENABLE\_VENDOR\_ONBOARDING=true
ENABLE\_AI\_RECOMMENDATIONS=true
ENABLE\_ROI\_ANALYTICS=true
ENABLE\_UNIFIED\_ANALYTICS=true
USE\_MOCK\_PAYMENTS=false
USE\_MOCK\_AI=false
# Performance
REDIS\_URL=${REDIS\_CLUSTER\_URL}
CDN\_BASE\_URL=https://cdn.skids.clinic
CACHE\_TTL=3600
# Compliance
HIPAA\_ENCRYPTION\_ENABLED=true
AUDIT\_LOGGING\_ENABLED=true
DATA\_RETENTION\_DAYS=2555

## \*\*DEPLOYMENT CHECKLISTS\*\*

### \*\*Pre-Deployment Checklist\*\*

#### \*\*Code Quality & Testing\*\*

* • All unit tests passing (90%+ coverage)
* • Integration tests passing
* • E2E tests passing across browsers
* • Performance benchmarks met
* • Security scanning completed
* • Code review completed and approved
* • Documentation updated

#### \*\*Infrastructure Preparation\*\*

* • Server provisioning completed
* • Database setup and migration scripts ready
* • Load balancer configured
* • SSL certificates installed
* • CDN configuration completed
* • Monitoring tools installed
* • Backup systems configured

#### \*\*Environment Configuration\*\*

* • Environment variables configured
* • Secrets management setup
* • API keys and credentials verified
* • External service integrations tested
* • Feature flags configured
* • Logging and monitoring enabled

#### \*\*Security Verification\*\*

* • Security scanning completed
* • Vulnerability assessment passed
* • Penetration testing completed
* • HIPAA compliance verified
* • Data encryption enabled
* • Access controls configured

### \*\*Deployment Execution Checklist\*\*

#### \*\*Pre-Deployment Steps\*\*

* • Maintenance window scheduled
* • Stakeholders notified
* • Rollback plan prepared
* • Database backup completed
* • Traffic routing configured

#### \*\*Deployment Steps\*\*

# 1. Deploy to staging first
npm run deploy:staging
# 2. Run staging verification tests
npm run test:staging
# 3. Deploy to production
npm run deploy:production
# 4. Run production smoke tests
npm run test:production:smoke
# 5. Gradually increase traffic
# Monitor metrics at each step

#### \*\*Post-Deployment Verification\*\*

* • Application health checks passing
* • Database connectivity verified
* • External integrations working
* • Payment gateways functional
* • Monitoring alerts configured
* • Performance metrics within targets
* • User acceptance testing completed

### \*\*Rollback Procedures\*\*

#### \*\*Automated Rollback Triggers\*\*

# rollback-triggers.yml
Error Rate: >5% for 5 minutes
Response Time: >3 seconds for 10 minutes
Database Errors: >10 errors in 1 minute
Payment Failures: >20% failure rate
External Service Failures: >50% failure rate

#### \*\*Rollback Execution\*\*

# 1. Immediate rollback
npm run rollback:immediate
# 2. Database rollback (if needed)
npm run db:rollback
# 3. Traffic routing rollback
npm run traffic:rollback
# 4. Verify rollback success
npm run verify:rollback

## \*\*MONITORING & ALERTING SYSTEMS\*\*

### \*\*Application Performance Monitoring (APM)\*\*

#### \*\*Metrics to Monitor\*\*

# monitoring-metrics.yml
Application Metrics:
- Response Time (p50, p95, p99)
- Throughput (requests/second)
- Error Rate (%)
- Availability (%)
- Memory Usage (MB)
- CPU Usage (%)
Business Metrics:
- Active Users
- Vendor Onboarding Rate
- Payment Success Rate
- ROI Analysis Accuracy
- Staff Productivity Score
Infrastructure Metrics:
- Server Health
- Database Performance
- Cache Hit Rate
- CDN Performance
- Network Latency

#### \*\*Alert Configuration\*\*

# alerts.yml
Critical Alerts:
- Error Rate >5%
- Response Time >3s
- Database Down
- Payment Gateway Down
- Security Breach
Warning Alerts:
- Error Rate >2%
- Response Time >2s
- High Memory Usage >80%
- High CPU Usage >80%
- Cache Miss Rate >20%
Info Alerts:
- Deployment Completed
- Backup Completed
- Certificate Renewal
- Scheduled Maintenance

### \*\*Monitoring Tools Setup\*\*

#### \*\*Application Monitoring\*\*

// monitoring/setup.ts
import \* as Sentry from '@sentry/nextjs'
import { Analytics } from '@vercel/analytics/react'
// Error tracking
Sentry.init({
dsn: process.env.SENTRY\_DSN,
environment: process.env.NODE\_ENV,
tracesSampleRate: 1.0,
beforeSend(event) {
// Filter sensitive data
if (event.request?.headers) {
delete event.request.headers.authorization
delete event.request.headers.cookie
}
return event
}
})
// Performance monitoring
export const performanceMonitor = {
trackPageLoad: (page: string, loadTime: number) => {
// Track page load performance
},
trackAPICall: (endpoint: string, duration: number, status: number) => {
// Track API performance
},
trackUserAction: (action: string, metadata: any) => {
// Track user interactions
}
}

#### \*\*Infrastructure Monitoring\*\*

# infrastructure-monitoring.yml
Tools:
- Datadog: Application and infrastructure monitoring
- New Relic: APM and real user monitoring
- Pingdom: Uptime monitoring
- CloudWatch: AWS infrastructure monitoring
- Grafana: Custom dashboards
Dashboards:
- System Overview
- Application Performance
- Business Metrics
- Security Events
- Vendor Performance

## \*\*SECURITY COMPLIANCE REQUIREMENTS\*\*

### \*\*HIPAA Compliance Checklist\*\*

#### \*\*Administrative Safeguards\*\*

* • Security Officer designated
* • Workforce training completed
* • Access management procedures
* • Security incident procedures
* • Contingency plan developed
* • Regular security evaluations

#### \*\*Physical Safeguards\*\*

* • Data center security controls
* • Workstation access controls
* • Device and media controls
* • Facility access controls

#### \*\*Technical Safeguards\*\*

* • Access control systems
* • Audit controls implemented
* • Integrity controls
* • Person or entity authentication
* • Transmission security

### \*\*Security Implementation\*\*

#### \*\*Data Encryption\*\*

// security/encryption.ts
import crypto from 'crypto'
export class DataEncryption {
private static algorithm = 'aes-256-gcm'
private static key = Buffer.from(process.env.ENCRYPTION\_KEY!, 'hex')
static encrypt(text: string): string {
const iv = crypto.randomBytes(16)
const cipher = crypto.createCipher(this.algorithm, this.key)
cipher.setAAD(Buffer.from('SKIDS-HIPAA', 'utf8'))
let encrypted = cipher.update(text, 'utf8', 'hex')
encrypted += cipher.final('hex')
const authTag = cipher.getAuthTag()
return iv.toString('hex') + ':' + authTag.toString('hex') + ':' + encrypted
}
static decrypt(encryptedData: string): string {
const parts = encryptedData.split(':')
const iv = Buffer.from(parts[0], 'hex')
const authTag = Buffer.from(parts[1], 'hex')
const encrypted = parts[2]
const decipher = crypto.createDecipher(this.algorithm, this.key)
decipher.setAAD(Buffer.from('SKIDS-HIPAA', 'utf8'))
decipher.setAuthTag(authTag)
let decrypted = decipher.update(encrypted, 'hex', 'utf8')
decrypted += decipher.final('utf8')
return decrypted
}
}

#### \*\*Audit Logging\*\*

// security/audit-logger.ts
export interface AuditEvent {
userId: string
action: string
resource: string
timestamp: Date
ipAddress: string
userAgent: string
success: boolean
details?: any
}
export class AuditLogger {
static async log(event: AuditEvent): Promise<void> {
// Encrypt sensitive data
const encryptedEvent = {
...event,
details: event.details ? DataEncryption.encrypt(JSON.stringify(event.details)) : null
}
// Store in secure audit database
await auditDatabase.insert('audit\_logs', encryptedEvent)
// Send to SIEM system
await siemIntegration.send(encryptedEvent)
}
}

### \*\*Security Verification Procedures\*\*

#### \*\*Penetration Testing\*\*

# penetration-testing.yml
Scope:
- Web application security
- API security testing
- Database security
- Infrastructure security
- Social engineering testing
Tools:
- OWASP ZAP
- Burp Suite Professional
- Nessus
- Metasploit
- Custom scripts
Schedule:
- Pre-deployment: Required
- Quarterly: Ongoing
- Post-incident: As needed

#### \*\*Vulnerability Management\*\*

# vulnerability-management.yml
Scanning:
- Daily: Automated vulnerability scans
- Weekly: Dependency vulnerability checks
- Monthly: Infrastructure scans
- Quarterly: Comprehensive assessments
Response Times:
- Critical: 4 hours
- High: 24 hours
- Medium: 7 days
- Low: 30 days
Tools:
- Snyk: Dependency scanning
- Qualys: Infrastructure scanning
- GitHub Security: Code scanning
- SonarQube: Code quality and security

## \*\*DEPLOYMENT SUCCESS CRITERIA\*\*

### \*\*Technical Metrics\*\*

* • \*\*Uptime\*\*: 99.9% availability
* • \*\*Performance\*\*: <2s page load time
* • \*\*Error Rate\*\*: <1% application errors
* • \*\*Security\*\*: Zero critical vulnerabilities
* • \*\*Compliance\*\*: 100% HIPAA compliance

### \*\*Business Metrics\*\*

* • \*\*User Adoption\*\*: 90% staff using system within 1 week
* • \*\*Vendor Onboarding\*\*: 100% completion rate for test vendors
* • \*\*Payment Processing\*\*: 99% success rate
* • \*\*ROI Analysis\*\*: Accurate predictions within 10% variance
* • \*\*Staff Productivity\*\*: 20% improvement in task completion

### \*\*Operational Metrics\*\*

* • \*\*Monitoring\*\*: All alerts configured and tested
* • \*\*Backup\*\*: Daily backups completing successfully
* • \*\*Documentation\*\*: 100% documentation coverage
* • \*\*Training\*\*: All staff trained and certified
* • \*\*Support\*\*: 24/7 support coverage established

## \*\*SUPPORT & ESCALATION\*\*

### \*\*Production Support Team\*\*

* • \*\*Level 1\*\*: Operations Team (24/7)
* • \*\*Level 2\*\*: Development Team (Business hours)
* • \*\*Level 3\*\*: Senior Architects (On-call)
* • \*\*Level 4\*\*: External Vendors (SLA-based)

### \*\*Emergency Contacts\*\*

* • \*\*System Administrator\*\*: +91-XXXX-XXXX-XX
* • \*\*Lead Developer\*\*: +91-XXXX-XXXX-XX
* • \*\*Security Officer\*\*: +91-XXXX-XXXX-XX
* • \*\*Business Owner\*\*: +91-XXXX-XXXX-XX

### \*\*Escalation Matrix\*\*

*This production deployment guide ensures a secure, scalable, and compliant deployment of the SKIDS Advanced integration infrastructure.*