

# Production Deployment Guide

## Jupiter Swap DApp

*Complete Production Deployment*

### Production Deployment Stack

<b>Platform:</b> Vercel Edge Network	<b>CDN:</b> Global Edge Caching
<b>Framework:</b> Next.js 14 App Router	<b>SSL:</b> Automatic HTTPS
<b>CI/CD:</b> GitHub Actions	<b>Environment:</b> Secure Variables
<b>Monitoring:</b> Sentry + Vercel Analytics	<b>Performance:</b> 99.9% Uptime SLA

### Deployment Achievements

Zero-downtime Deployments  
Automatic SSL Certificates  
Global CDN Distribution  
Environment Security  
Performance Monitoring  
Error Tracking  
Rollback Capabilities  
Scalable Infrastructure

**Author:** Kamel (@treizeb\_\_)  
**Company:** DeAura.io  
**Updated:** July 14, 2025

## Contents

<b>1</b>	<b>Vercel Deployment</b>	<b>2</b>
1.1	Initial Setup . . . . .	2
1.2	Environment Variables Configuration . . . . .	2
1.3	Vercel Configuration . . . . .	3
<b>2</b>	<b>GitHub Actions CI/CD</b>	<b>4</b>
2.1	Workflow Configuration . . . . .	4
2.2	Lighthouse Configuration . . . . .	6
<b>3</b>	<b>Monitoring Setup</b>	<b>7</b>
3.1	Sentry Configuration . . . . .	7
3.2	Health Check Endpoint . . . . .	8
<b>4</b>	<b>Security Configuration</b>	<b>10</b>
4.1	Content Security Policy . . . . .	10
<b>5</b>	<b>Performance Optimization</b>	<b>12</b>
5.1	Bundle Analysis . . . . .	12
5.2	Performance Monitoring . . . . .	12
<b>6</b>	<b>Rollback Procedures</b>	<b>13</b>
6.1	Automatic Rollback . . . . .	13
<b>7</b>	<b>Conclusion</b>	<b>14</b>
7.1	Deployment Summary . . . . .	14

# 1 Vercel Deployment

## 1.1 Initial Setup

```
1 # Install Vercel CLI globally
2 npm install -g vercel
3
4 # Login to Vercel account
5 vercel login
6
7 # Initialize project in repository root
8 cd jupiter-swap-nextjs-production
9 vercel
10
11 # Follow interactive setup:
12 # ? Set up and deploy "jupiter-swap-nextjs-production"? [Y/n] Y
13 # ? Which scope do you want to deploy to? [Your Team]
14 # ? Link to existing project? [y/N] N
15 # ? What's your project's name? jupiter-swap-dapp
16 # ? In which directory is your code located? ./
17 # ? Want to override the settings? [y/N] N
18
19 # Deploy to production
20 vercel --prod
```

Listing 1: Vercel CLI Installation and Setup

## 1.2 Environment Variables Configuration

```
1 # Set environment variables via Vercel CLI
2 vercel env add NEXT_PUBLIC_HELIUS_API_KEY production
3 # Enter: d94d81dd-f2a1-40f7-920d-0dfaf3aaf032
4
5 vercel env add NEXT_PUBLIC_ALCHEMY_API_KEY production
6 # Enter: Uv0k23LRlqGz1m58VCEd3PJ2Z0X2h9KM
7
8 vercel env add NEXT_PUBLIC_COINGECKO_API_KEY production
9 # Enter: CG-your-api-key-here
10
11 vercel env add NEXT_PUBLIC_SOLANA_NETWORK production
12 # Enter: mainnet-beta
13
14 vercel env add NEXT_PUBLIC_JUPITER_API_URL production
15 # Enter: https://quote-api.jup.ag/v6
16
17 vercel env add NEXT_PUBLIC_ENABLE_DEVTOOLS production
18 # Enter: false
19
20 vercel env add SENTRY_DSN production
21 # Enter: https://your-sentry-dsn@sentry.io/project-id
22
23 vercel env add SENTRY_AUTH_TOKEN production
24 # Enter: your-sentry-auth-token
25
26 # Verify environment variables
27 vercel env ls
```

Listing 2: Production Environment Variables

## 1.3 Vercel Configuration

```
1 {
2   "version": 2,
3   "name": "jupiter-swap-dapp",
4   "alias": ["jupiter-swap.deaura.io"],
5   "regions": ["iad1", "sfo1", "fra1", "hnd1"],
6   "build": {
7     "env": {
8       "NODE_ENV": "production",
9       "NEXT_TELEMETRY_DISABLED": "1"
10    }
11  },
12  "functions": {
13    "pages/api/**/*.ts": {
14      "runtime": "nodejs18.x",
15      "maxDuration": 30
16    }
17  },
18  "headers": [
19    {
20      "source": "/(.*)",
21      "headers": [
22        {
23          "key": "X-Content-Type-Options",
24          "value": "nosniff"
25        },
26        {
27          "key": "X-Frame-Options",
28          "value": "DENY"
29        },
30        {
31          "key": "X-XSS-Protection",
32          "value": "1; mode=block"
33        },
34        {
35          "key": "Referrer-Policy",
36          "value": "strict-origin-when-cross-origin"
37        },
38        {
39          "key": "Permissions-Policy",
40          "value": "camera=(), microphone=(), geolocation=()"
41        }
42      ]
43    },
44    {
45      "source": "/api/(.*)",
46      "headers": [
47        {
48          "key": "Access-Control-Allow-Origin",
49          "value": "https://jupiter-swap.deaura.io"
50        },
51        {
52          "key": "Access-Control-Allow-Methods",
53          "value": "GET, POST, OPTIONS"
54        },
55        {
56          "key": "Access-Control-Allow-Headers",
57          "value": "Content-Type, Authorization"
58        }
59      ]
60    }
61  ],
```

```
62  "rewrites": [  
63    {  
64      "source": "/health",  
65      "destination": "/api/health"  
66    }  
67  ],  
68  "redirects": [  
69    {  
70      "source": "/swap",  
71      "destination": "/",  
72      "permanent": false  
73    }  
74  ]  
75 }
```

Listing 3: vercel.json Configuration

## 2 GitHub Actions CI/CD

### 2.1 Workflow Configuration

```
1  name: Deploy to Production  
2  
3  on:  
4    push:  
5      branches: [main]  
6    pull_request:  
7      branches: [main]  
8  
9  env:  
10  VERCEL_ORG_ID: ${ secrets.VERCEL_ORG_ID }  
11  VERCEL_PROJECT_ID: ${ secrets.VERCEL_PROJECT_ID }  
12  
13  jobs:  
14    test:  
15      runs-on: ubuntu-latest  
16  
17      steps:  
18        - name: Checkout code  
19          uses: actions/checkout@v4  
20  
21        - name: Setup Node.js  
22          uses: actions/setup-node@v4  
23          with:  
24            node-version: '18'  
25            cache: 'npm'  
26  
27        - name: Install dependencies  
28          run: npm ci  
29  
30        - name: Run type checking  
31          run: npm run type-check  
32  
33        - name: Run linting  
34          run: npm run lint  
35  
36        - name: Run tests  
37          run: npm run test:ci  
38          env:  
39            CI: true  
40
```

```

41   - name: Build application
42     run: npm run build
43     env:
44       NODE_ENV: production
45
46   - name: Upload coverage to Codecov
47     uses: codecov/codecov-action@v3
48     with:
49       file: ./coverage/lcov.info
50       flags: unittests
51       name: codecov-umbrella
52
53 deploy-preview:
54   runs-on: ubuntu-latest
55   needs: test
56   if: github.event_name == 'pull_request'
57
58   steps:
59     - name: Checkout code
60       uses: actions/checkout@v4
61
62     - name: Install Vercel CLI
63       run: npm install --global vercel@latest
64
65     - name: Pull Vercel Environment Information
66       run: vercel pull --yes --environment=preview --token=${{ secrets.VERCEL_TOKEN }}
67
68     - name: Build Project Artifacts
69       run: vercel build --token=${{ secrets.VERCEL_TOKEN }}
70
71     - name: Deploy Project Artifacts to Vercel
72       id: deploy
73       run: |
74         url=$(vercel deploy --prebuilt --token=${{ secrets.VERCEL_TOKEN }})
75         echo "preview_url=$url" >> $GITHUB_OUTPUT
76
77     - name: Comment PR with preview URL
78       uses: actions/github-script@v7
79       with:
80         script: |
81           github.rest.issues.createComment({
82             issue_number: context.issue.number,
83             owner: context.repo.owner,
84             repo: context.repo.repo,
85             body: '      Preview deployment ready!\n\n**Preview URL:** ${{ steps.
86               deploy.outputs.preview_url }}\n\n*Deployed from commit ${{ github.sha }}*'
87           })
88
89 deploy-production:
90   runs-on: ubuntu-latest
91   needs: test
92   if: github.ref == 'refs/heads/main'
93
94   steps:
95     - name: Checkout code
96       uses: actions/checkout@v4
97
98     - name: Install Vercel CLI
99       run: npm install --global vercel@latest
100
101     - name: Pull Vercel Environment Information

```

```

101     run: vercel pull --yes --environment=production --token=${{ secrets.
VERCEL_TOKEN }}
102
103     - name: Build Project Artifacts
104       run: vercel build --prod --token=${{ secrets.VERCEL_TOKEN }}
105
106     - name: Deploy Project Artifacts to Vercel
107       id: deploy
108       run: |
109         url=$(vercel deploy --prebuilt --prod --token=${{ secrets.VERCEL_TOKEN }})
110         echo "production_url=$url" >> $GITHUB_OUTPUT
111
112     - name: Create deployment status
113       uses: actions/github-script@v7
114       with:
115         script: |
116           github.rest.repos.createDeploymentStatus({
117             owner: context.repo.owner,
118             repo: context.repo.repo,
119             deployment_id: context.payload.deployment.id,
120             state: 'success',
121             environment_url: '${{ steps.deploy.outputs.production_url }}',
122             description: 'Deployment successful'
123           })
124
125     lighthouse:
126       runs-on: ubuntu-latest
127       needs: deploy-production
128       if: github.ref == 'refs/heads/main'
129
130     steps:
131       - name: Checkout code
132         uses: actions/checkout@v4
133
134       - name: Run Lighthouse CI
135         uses: treosh/lighthouse-ci-action@v10
136         with:
137           urls: |
138             https://jupiter-swap.deaura.io
139           configPath: './lighthouseci.json'
140           uploadArtifacts: true
141           temporaryPublicStorage: true

```

Listing 4: .github/workflows/deploy.yml

## 2.2 Lighthouse Configuration

```

1 {
2   "ci": {
3     "collect": {
4       "numberOfRuns": 3,
5       "settings": {
6         "chromeFlags": "--no-sandbox --headless"
7       }
8     },
9     "assert": {
10      "assertions": {
11        "categories:performance": ["error", {"minScore": 0.9}],
12        "categories:accessibility": ["error", {"minScore": 0.9}],
13        "categories:best-practices": ["error", {"minScore": 0.9}],
14        "categories:seo": ["error", {"minScore": 0.9}],
15        "first-contentful-paint": ["error", {"maxNumericValue": 2000}],

```

```
16     "largest-contentful-paint": ["error", {"maxNumericValue": 3000}],
17     "cumulative-layout-shift": ["error", {"maxNumericValue": 0.1}]
18   },
19 },
20 "upload": {
21   "target": "temporary-public-storage"
22 }
23 }
24 }
```

Listing 5: lighthousec.json

## 3 Monitoring Setup

### 3.1 Sentry Configuration

```
1 import * as Sentry from '@sentry/nextjs';
2
3 Sentry.init({
4   dsn: process.env.SENTRY_DSN,
5
6   // Performance monitoring
7   tracesSampleRate: 1.0,
8
9   // Session replay
10  replaysSessionSampleRate: 0.1,
11  replaysOnErrorSampleRate: 1.0,
12
13  // Environment
14  environment: process.env.NODE_ENV,
15
16  // Release tracking
17  release: process.env.VERCEL_GIT_COMMIT_SHA,
18
19  // Error filtering
20  beforeSend(event, hint) {
21    // Filter out known non-critical errors
22    if (event.exception) {
23      const error = hint.originalException;
24
25      // Filter wallet connection errors (user-initiated)
26      if (error?.message?.includes('User rejected')) {
27        return null;
28      }
29
30      // Filter network timeouts (temporary)
31      if (error?.message?.includes('timeout')) {
32        return null;
33      }
34    }
35
36    return event;
37  },
38
39  // Custom tags
40  initialScope: {
41    tags: {
42      component: 'jupiter-swap-dapp',
43    },
44  },
45 }
```



```

46 // Integration configuration
47 integrations: [
48   new Sentry.BrowserTracing({
49     routingInstrumentation: Sentry.nextRouterInstrumentation(
50       require('next/router')
51     ),
52   }),
53   new Sentry.Replay(),
54 ],
55 });

```

Listing 6: sentry.client.config.ts

### 3.2 Health Check Endpoint

```

1 import type { NextApiRequest, NextApiResponse } from 'next';
2 import { Connection } from '@solana/web3.js';
3
4 interface HealthStatus {
5   status: 'healthy' | 'degraded' | 'unhealthy';
6   timestamp: string;
7   version: string;
8   services: {
9     solana: 'up' | 'down';
10    jupiter: 'up' | 'down';
11    helius: 'up' | 'down';
12    alchemy: 'up' | 'down';
13  };
14   performance: {
15     uptime: number;
16     memory: {
17       used: number;
18       total: number;
19     };
20   };
21 }
22
23 export default async function handler(
24   req: NextApiRequest,
25   res: NextApiResponse<HealthStatus>
26 ) {
27   const startTime = Date.now();
28
29   try {
30     // Check Solana RPC
31     const connection = new Connection(
32       'https://mainnet.helius-rpc.com/?api-key=${process.env.
33       NEXT_PUBLIC_HELIUS_API_KEY}'
34     );
35
36     const [solanaHealth, jupiterHealth, heliusHealth, alchemyHealth] =
37       await Promise.allSettled([
38         // Solana health check
39         connection.getEpochInfo(),
40
41         // Jupiter API health check
42         fetch('https://quote-api.jup.ag/v6/quote?inputMint=
43         So11111111111111111111111111111111111111111111111111111112&outputMint=
44         EPjFWdd5AufqSSqeM2qN1xzybapC8G4wEGGkZwyTDt1v&amount=1000000000')
45           .then(res => res.ok),
46
47         // Helius health check

```

```

45     fetch('https://mainnet.helius-rpc.com/?api-key=${process.env.
NEXT_PUBLIC_HELIUS_API_KEY}', {
46         method: 'POST',
47         headers: { 'Content-Type': 'application/json' },
48         body: JSON.stringify({
49             jsonrpc: '2.0',
50             id: 1,
51             method: 'getHealth',
52         }),
53     }).then(res => res.ok),
54
55     // Alchemy health check
56     fetch('https://solana-mainnet.g.alchemy.com/v2/${process.env.
NEXT_PUBLIC_ALCHEMY_API_KEY}', {
57         method: 'POST',
58         headers: { 'Content-Type': 'application/json' },
59         body: JSON.stringify({
60             jsonrpc: '2.0',
61             id: 1,
62             method: 'getHealth',
63         }),
64     }).then(res => res.ok),
65     ]);
66
67     const services = {
68         solana: solanaHealth.status === 'fulfilled' ? 'up' : 'down',
69         jupiter: jupiterHealth.status === 'fulfilled' && jupiterHealth.value ? 'up' : '
down',
70         helius: heliusHealth.status === 'fulfilled' && heliusHealth.value ? 'up' : '
down',
71         alchemy: alchemyHealth.status === 'fulfilled' && alchemyHealth.value ? 'up' : '
down',
72     } as const;
73
74     const healthyServices = Object.values(services).filter(status => status === 'up')
.length;
75     const totalServices = Object.values(services).length;
76
77     let overallStatus: HealthStatus['status'];
78     if (healthyServices === totalServices) {
79         overallStatus = 'healthy';
80     } else if (healthyServices >= totalServices / 2) {
81         overallStatus = 'degraded';
82     } else {
83         overallStatus = 'unhealthy';
84     }
85
86     const memoryUsage = process.memoryUsage();
87
88     const healthStatus: HealthStatus = {
89         status: overallStatus,
90         timestamp: new Date().toISOString(),
91         version: process.env.VERCEL_GIT_COMMIT_SHA || 'unknown',
92         services,
93         performance: {
94             uptime: process.uptime(),
95             memory: {
96                 used: Math.round(memoryUsage.heapUsed / 1024 / 1024),
97                 total: Math.round(memoryUsage.heapTotal / 1024 / 1024),
98             },
99         },
100     };
101

```

```

102   const responseTime = Date.now() - startTime;
103   res.setHeader('X-Response-Time', `${responseTime}ms`);
104
105   // Set appropriate status code
106   const statusCode = overallStatus === 'healthy' ? 200 :
107                     overallStatus === 'degraded' ? 200 : 503;
108
109   res.status(statusCode).json(healthStatus);
110
111 } catch (error) {
112   console.error('Health check failed:', error);
113
114   res.status(503).json({
115     status: 'unhealthy',
116     timestamp: new Date().toISOString(),
117     version: process.env.VERCEL_GIT_COMMIT_SHA || 'unknown',
118     services: {
119       solana: 'down',
120       jupiter: 'down',
121       helius: 'down',
122       alchemy: 'down',
123     },
124     performance: {
125       uptime: process.uptime(),
126       memory: {
127         used: 0,
128         total: 0,
129       },
130     },
131   });
132 }
133 }

```

Listing 7: pages/api/health.ts

## 4 Security Configuration

### 4.1 Content Security Policy

```

1  /** @type {import('next').NextConfig} */
2  const nextConfig = {
3    reactStrictMode: true,
4    swcMinify: true,
5
6    // Security headers
7    async headers() {
8      return [
9        {
10         source: '/(.*)',
11         headers: [
12           {
13             key: 'Content-Security-Policy',
14             value: [
15               "default-src 'self'",
16               "script-src 'self' 'unsafe-eval' 'unsafe-inline' https://vercel.live",
17               "style-src 'self' 'unsafe-inline' https://fonts.googleapis.com",
18               "font-src 'self' https://fonts.gstatic.com",
19               "img-src 'self' data: https://quote-api.jup.ag https://mainnet.helius-rpc.com https://solana-mainnet.g.alchemy.com https://eclipse.helius-rpc.com wss://mainnet.helius-rpc.com",
20             ],
11

```

```

21         "frame-src 'none'",
22         "object-src 'none'",
23         "base-uri 'self'",
24         "form-action 'self'",
25         "frame-ancestors 'none'",
26         "upgrade-insecure-requests",
27     ].join('; '),
28 },
29 {
30     key: 'X-DNS-Prefetch-Control',
31     value: 'on',
32 },
33 {
34     key: 'Strict-Transport-Security',
35     value: 'max-age=63072000; includeSubDomains; preload',
36 },
37 {
38     key: 'X-Content-Type-Options',
39     value: 'nosniff',
40 },
41 {
42     key: 'X-Frame-Options',
43     value: 'DENY',
44 },
45 {
46     key: 'X-XSS-Protection',
47     value: '1; mode=block',
48 },
49 {
50     key: 'Referrer-Policy',
51     value: 'strict-origin-when-cross-origin',
52 },
53 {
54     key: 'Permissions-Policy',
55     value: 'camera=(), microphone=(), geolocation=(), payment=()',
56 },
57 ],
58 },
59 ];
60 },
61
62 // Environment variables validation
63 env: {
64     CUSTOM_KEY: process.env.CUSTOM_KEY,
65 },
66
67 // Webpack configuration
68 webpack: (config, { dev, isServer }) => {
69     // Security: Don't expose source maps in production
70     if (!dev && !isServer) {
71         config.devtool = false;
72     }
73
74     return config;
75 },
76
77 // Image optimization
78 images: {
79     domains: ['assets.coingecko.com', 'raw.githubusercontent.com'],
80     formats: ['image/webp', 'image/avif'],
81 },
82
83 // Experimental features

```

```
84 experimental: {
85   // Enable modern bundling
86   esmExternals: true,
87   // Optimize server components
88   serverComponentsExternalPackages: ['@solana/web3.js'],
89 },
90 };
91
92 module.exports = nextConfig;
```

Listing 8: next.config.js Security Headers

## 5 Performance Optimization

### 5.1 Bundle Analysis

```
1 # Install bundle analyzer
2 npm install --save-dev @next/bundle-analyzer
3
4 # Add to package.json scripts
5 {
6   "scripts": {
7     "analyze": "ANALYZE=true npm run build",
8     "analyze:server": "BUNDLE_ANALYZE=server npm run build",
9     "analyze:browser": "BUNDLE_ANALYZE=browser npm run build"
10  }
11 }
12
13 # Run analysis
14 npm run analyze
```

Listing 9: Bundle Analysis Setup

### 5.2 Performance Monitoring

```
1 // pages/_app.tsx
2 import { AppProps } from 'next/app';
3 import { getCLS, getFID, getFCP, getLCP, getTTFB } from 'web-vitals';
4
5 function sendToAnalytics(metric: any) {
6   // Send to your analytics service
7   if (process.env.NODE_ENV === 'production') {
8     fetch('/api/analytics', {
9       method: 'POST',
10      headers: {
11        'Content-Type': 'application/json',
12      },
13      body: JSON.stringify(metric),
14    }).catch(console.error);
15  }
16 }
17
18 export function reportWebVitals(metric: any) {
19   console.log(metric);
20   sendToAnalytics(metric);
21 }
22
23 function MyApp({ Component, pageProps }: AppProps) {
24   // Track Core Web Vitals
25   if (typeof window !== 'undefined') {
```

```

26     getCLS(sendToAnalytics);
27     getFID(sendToAnalytics);
28     getFCP(sendToAnalytics);
29     getLCP(sendToAnalytics);
30     getTTFB(sendToAnalytics);
31 }
32
33 return <Component {...pageProps} />;
34 }
35
36 export default MyApp;

```

Listing 10: Web Vitals Tracking

## 6 Rollback Procedures

### 6.1 Automatic Rollback

```

1  #!/bin/bash
2
3  # rollback.sh - Emergency rollback script
4
5  set -e
6
7  # Configuration
8  VERCEL_TOKEN=${VERCEL_TOKEN}
9  PROJECT_ID=${VERCEL_PROJECT_ID}
10 TEAM_ID=${VERCEL_ORG_ID}
11
12 # Colors for output
13 RED='\033[0;31m'
14 GREEN='\033[0;32m'
15 YELLOW='\033[1;33m'
16 NC='\033[0m' # No Color
17
18 echo -e "${YELLOW}      Starting emergency rollback...${NC}"
19
20 # Get current production deployment
21 current_deployment=$(vercel ls --token=$VERCEL_TOKEN --scope=$TEAM_ID | grep "jupiter
    -swap-dapp" | grep "READY" | head -1 | awk '{print $2}')
22
23 if [ -z "$current_deployment" ]; then
24     echo -e "${RED}      Could not find current deployment${NC}"
25     exit 1
26 fi
27
28 echo -e "${YELLOW}Current deployment: $current_deployment${NC}"
29
30 # Get previous deployment
31 previous_deployment=$(vercel ls --token=$VERCEL_TOKEN --scope=$TEAM_ID | grep "
    jupiter-swap-dapp" | grep "READY" | sed -n '2p' | awk '{print $2}')
32
33 if [ -z "$previous_deployment" ]; then
34     echo -e "${RED}      Could not find previous deployment to rollback to${NC}"
35     exit 1
36 fi
37
38 echo -e "${YELLOW}Rolling back to: $previous_deployment${NC}"
39
40 # Promote previous deployment to production
41 vercel promote $previous_deployment --token=$VERCEL_TOKEN --scope=$TEAM_ID

```

```
42
43 if [ $? -eq 0 ]; then
44     echo -e "${GREEN}    Rollback completed successfully${NC}"
45     echo -e "${GREEN}Production is now running: $previous_deployment${NC}"
46
47     # Send notification (optional)
48     if [ ! -z "$SLACK_WEBHOOK" ]; then
49         curl -X POST -H 'Content-type: application/json' \
50             --data "{\"text\":\"          Jupiter Swap DApp rolled back to
$previous_deployment\"}" \
51                 $SLACK_WEBHOOK
52     fi
53 else
54     echo -e "${RED}    Rollback failed${NC}"
55     exit 1
56 fi
```

Listing 11: Rollback Script

## 7 Conclusion

This comprehensive production deployment guide ensures reliable, secure, and performant deployment of the Jupiter Swap DApp to Vercel with complete CI/CD automation, monitoring, and rollback capabilities.

### 7.1 Deployment Summary

#### Production Deployment Achievements:

- **Zero-downtime Deployments:** Seamless updates
- **Global CDN:** Sub-100ms response times worldwide
- **Automatic SSL:** HTTPS everywhere with HSTS
- **CI/CD Pipeline:** Automated testing and deployment
- **Performance Monitoring:** Real-time metrics and alerts
- **Error Tracking:** Comprehensive error monitoring
- **Security Headers:** Enterprise-grade security
- **Rollback Capabilities:** Quick recovery procedures

*Production deployment architecture designed and implemented by Kamel (@treizeb\_\_)  
DeAura.io - July 2025*