Hotdog Diaries SRE Runbook

Document Version: 1.0

Last Updated: October 16, 2025 **Owner:** Platform Engineering **Review Cycle:** Monthly

Table of Contents

- 1. Overview & Environments
- 2. Secrets Management
- 3. Deploy Playbook
- 4. Rollback Procedures
- 5. Incident Response
- 6. Scheduler Operations
- 7. Backups & Disaster Recovery
- 8. Verification Checklists

Overview & Environments

System Architecture

Hotdog Diaries is a Next.js application that scans social media for hotdog content and posts it 6 times daily using a scheduled content system.

Environment Details

Environment	URL	Database	Owner	Deploy Method
Development	http://localhost:3000	SQLite (hotdog_diaries_dev.db)	Local Dev	npm run dev
Production	https://hotdog- diaries.vercel.app	Supabase PostgreSQL	Platform Team	Vercel Auto- Deploy

Key Components

• Frontend: Next.js 15.4.1 with React 19

• Database: Supabase PostgreSQL (prod) / SQLite (dev)

- Authentication: JWT with EdgeAuthUtils
- **Scheduling**: 6 daily time slots (08:00, 12:00, 15:00, 18:00, 21:00, 23:30 ET)
- Content Sources: Reddit, YouTube, Imgur, Bluesky, Tumblr, Lemmy, Pixabay

Critical Endpoints

- /admin/health/deep System health check
- /api/system/metrics Public metrics
- /admin/schedule/forecast Content forecast
- /admin/schedule/forecast/refill Schedule refill
- /admin/schedule/forecast/reconcile Schedule reconciliation

Secrets Management

Complete secrets documentation: docs/secrets.md

Environment Variables Required

Production (Vercel)

Core Application NODE_ENV=production NEXT_PUBLIC_SUPABASE_URL=https://ulaadphxfsrihoubjdrb.supabase.co SUPABASE_SERVICE_ROLE_KEY=eyJhbGci... DATABASE_URL=postgresql://postgres... JWT_SECRET=0d600b3805c95568d0a36950a7f3486f6e29fd43a6e95d6bc7ac3a282aa264947199ca5562b156dbae4bf56911467750et # Admin Credentials ADMIN_USERNAME=admin ADMIN_PASSWORD=StrongAdminPass123! ADMIN_EMAIL=admin@hotdogdiaries.com YOUTUBE_API_KEY=AIzaSyBUeB1_I_qu3Tl2zu0JD5tdC6NuVXwiKxA REDDIT_CLIENT_ID=your_reddit_client_id IMGUR_CLIENT_ID=your_imgur_client_id PIXABAY_API_KEY=your_pixabay_key GIPHY_API_KEY=your_giphy_key BLUESKY_IDENTIFIER=adampseudo.bsky.social BLUESKY_APP_PASSWORD=wiwo-vmqp-esdf-evkt

Secrets Rotation

Automated rotation: Use scripts/rotate-secrets.sh

```
# Rotate all secrets (requires confirmation)
./scripts/rotate-secrets.sh --confirm

# Rotate specific secret
./scripts/rotate-secrets.sh --secret JWT_SECRET --confirm

# Dry run (shows what would be rotated)
./scripts/rotate-secrets.sh
```

Manual rotation steps:

- 1. Generate new secret values
- 2. Update Vercel environment variables
- 3. Trigger new deployment
- 4. Verify all services using new secrets
- 5. Invalidate old secrets where possible

Deploy Playbook

Normal Deployment Process

Pre-Deploy Checklist

- All tests passing (npm test)
 OpenAPI spec validated (npm run api:validate)
 Route inventory up to date (npm run api:check-drift)
- Database migrations tested locally
- Secrets rotation scheduled (if needed)

Deployment Steps

1. Merge to main branch

git checkout main git pull origin main git merge feature-branch git push origin main

2. Vercel auto-deploys automatically

- o Monitor deployment at https://vercel.com/dashboard
- o Typical deploy time: 2-3 minutes

3. Post-Deploy Verification

Post-Deploy Smoke Testing

Automated Smoke Tests

```
# Run full smoke test suite
./scripts/smoke.sh

# Expected: Exit code 0, all checks pass

# Individual endpoint checks

curl -f https://hotdog-diaries.vercel.app/api/system/metrics

curl -f https://hotdog-diaries.vercel.app/admin/health/deep \
-H "Authorization: Bearer $AUTH_TOKEN"
```

Manual Verification Sequence

1. System Health Check

```
curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $AUTH_TOKEN" | jq '{
    database: .database.status,
    apis: .apis.status,
    scheduler: .scheduler.status
}'
```

2. Today's Forecast

```
TODAY=$(date -u +%Y-%m-%d)

curl -s "https://hotdog-diaries.vercel.app/admin/schedule/forecast?date=$TODAY" \

-H "Authorization: Bearer $AUTH_TOKEN" | jq '.summary'
```

3. Content Queue Health

```
curl -s https://hotdog-diaries.vercel.app/api/system/metrics | jq '{
   ready_content: .queue.ready_to_post,
   diversity_score: .diversity.score,
   last_post: .posting.last_post_time
}'
```

Post-Deploy Gate Interpretation

✓ Healthy Deployment Indicators:

- All /admin/health/deep checks return "status": "healthy"
- Forecast shows 6 filled slots for today/tomorrow
- Queue metrics show >10 ready_to_post items
- Diversity score >0.7

X Unhealthy Deployment Indicators:

- Database connection failures
- Missing scheduled content (empty forecast)
- Queue exhaustion (<5 ready items)
- API authentication failures

Deployment Failure Recovery

If post-deploy checks fail:

1. Immediate Actions

```
# Check Vercel deployment logs
vercel logs https://hotdog-diaries.vercel.app

# Verify environment variables
vercel env ls
```

2. Quick Fixes

```
# Trigger emergency refill if queue empty
curl -X POST https://hotdog-diaries.vercel.app/admin/schedule/forecast/refill \
    -H "Authorization: Bearer $AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{"date": "'$(date -u +%Y-%m-%d)'", "mode": "create-or-reuse", "twoDays": true}'
```

3. Escalate to rollback if fixes don't work within 10 minutes

Rollback Procedures

Vercel Rollback

Quick Rollback (Emergency)

```
# Via Vercel CLI
vercel rollback https://hotdog-diaries.vercel.app

# Via Vercel Dashboard
# 1. Go to https://vercel.com/dashboard
# 2. Select hotdog-diaries project
# 3. Click "Deployments" tab
# 4. Find last known good deployment
# 5. Click "..." → "Redeploy"
```

Rollback Verification

After rollback, verify system health:

```
# Wait 2 minutes for deployment, then verify
sleep 120
./scripts/smoke.sh
# Check version/commit
curl -s https://hotdog-diaries.vercel.app/api/system/metrics | jq '.build'
```

Database Rollback Procedures

Supabase Migration Rollback

▲ Important: Database rollbacks are forward-fixes only. No automatic rollback.

1. Assess Migration Impact

```
# Check current schema version

curl -s https://hotdog-diaries.vercel.app/admin/health/deep \

-H "Authorization: Bearer $AUTH_TOKEN" | jq '.database.schema_version'
```

2. Create Forward-Fix Migration

```
# Create new migration to undo changes
npx tsx scripts/create-migration.ts --name "rollback_feature_xyz"
# Apply the rollback migration
npx tsx scripts/migrate-database.ts
```

3. Emergency Data Recovery

```
# If data corruption occurred
./scripts/restore-supabase.sh --backup-date YYYY-MM-DD --confirm
```

Configuration Rollback

```
# Reset environment variables to previous values
vercel env rm NEW_VARIABLE
vercel env add OLD_VARIABLE previous_value

# Trigger redeployment
vercel --prod
```

Incident Response

Severity Matrix

Severity	Definition	Response Time	Escalation
P0 - Critical	Site down, data loss	5 minutes	Immediate page
P1 - High	Major feature broken	30 minutes	Page during hours
P2 - Medium	Minor feature issues	2 hours	Slack notification
P3 - Low	Cosmetic/enhancement	Next business day	Ticket queue

Quick Triage Flow

1. Health Check Sequence

- 2. Common Failure Patterns
- Q Database Connection Issues

```
# Symptoms: health/deep returns database: "unhealthy"
# Investigation:
curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $AUTH_TOKEN" | jq '.database'

# Resolution:
# 1. Check Supabase status: https://status.supabase.com
# 2. Verify DATABASE_URL in Vercel env
# 3. Test connection: psql $DATABASE_URL -c "SELECT 1;"
```

Auth Token Mismatch

```
# Symptoms: 401 responses on admin endpoints
# Investigation:
curl -v https://hotdog-diaries.vercel.app/admin/health/auth-token \
    -H "Authorization: Bearer $AUTH_TOKEN"

# Resolution:
# 1. Generate fresh token: npm run generate:service-token
# 2. Update AUTH_TOKEN in CI/monitoring systems
# 3. Verify JWT_SECRET matches across environments
```

Content Queue Exhaustion

Scheduler Issues

```
# Symptoms: Content not posting at scheduled times
# Investigation:
curl -s https://hotdog-diaries.vercel.app/admin/schedule/forecast \
    -H "Authorization: Bearer $AUTH_TOKEN" | jq '.summary'

# Resolution:
# 1. Check GitHub Actions cron status
# 2. Manually trigger posting: POST /admin/posting/execute
# 3. Reconcile schedule: POST /admin/schedule/forecast/reconcile
```

Incident Response Runbook

P0 - Site Down

1. Immediate (0-5 minutes)

```
# Quick status check
./scripts/smoke.sh

# Check Vercel status
curl -s https://www.vercel-status.com/api/v2/status.json | jq '.status.indicator'
```

2. Triage (5-15 minutes)

```
# Check deployment logs
vercel logs https://hotdog-diaries.vercel.app --lines 50
# Test database connectivity
psql $DATABASE_URL -c "SELECT count(*) FROM content_queue;"
```

- 3. Resolution (15+ minutes)
 - o If recent deployment: Execute rollback
 - o If infrastructure: Monitor vendor status pages
 - o If database: Check Supabase dashboard

P1 - Major Feature Broken

1. Investigation

```
# Check specific feature health

curl -s https://hotdog-diaries.vercel.app/admin/health/deep \

-H "Authorization: Bearer $AUTH_TOKEN" | jq '.features'
```

- 2. Common Resolutions
 - Auth issues: Rotate JWT_SECRET
 - Queue issues: Execute refill operation
 - o API issues: Check rate limits and credentials

Scheduler Operations

Content Forecast Management

Understanding the Forecast System

- 6 Daily Slots: 08:00, 12:00, 15:00, 18:00, 21:00, 23:30 ET
- Platform Diversity: No consecutive posts from same platform

• Content Types: Mixed text, image, video content

Two-Day Refill Recipes

Standard Refill (recommended)

```
# Refill today and tomorrow with platform diversity
curl -X POST https://hotdog-diaries.vercel.app/admin/schedule/forecast/refill \
    -H "Authorization: Bearer $AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{
        "date": "'$(date -u +%Y-%m-%d)'",
        "mode": "create-or-reuse",
        "twoDays": true
    }'
```

Force Recreation (emergency)

```
# Clear existing schedule and recreate
curl -X POST https://hotdog-diaries.vercel.app/admin/schedule/forecast/refill \
    -H "Authorization: Bearer $AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{
        "date": "'$(date -u +%Y-%m-%d)'",
        "mode": "force-recreate",
        "twoDays": true
}'
```

Tuture Date Preparation

```
# Refill specific future date
FUTURE_DATE=$(date -u -d "+7 days" +%Y-%m-%d)
curl -X POST https://hotdog-diaries.vercel.app/admin/schedule/forecast/refill \
    -H "Authorization: Bearer $AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{
        "date": "'$FUTURE_DATE'",
        "mode": "create-or-reuse",
        "twoDays": false
}'
```

Diversity Warning Policy

Warning Triggers

- Platform Concentration: >50% from single platform in 24h
- Content Type Imbalance: >70% same type (text/image/video)
- Low Queue Diversity: <3 different platforms available

Resolution Actions

```
# Check current diversity status
curl -s https://hotdog-diaries.vercel.app/api/system/metrics | jq '.diversity'

# Emergency platform rebalancing
curl -X POST https://hotdog-diaries.vercel.app/admin/platforms/scan \
    -H "Authorization: Bearer $AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{"emergency_rebalance": true}'
```

Manual Reconciliation Steps

When to Reconcile

- Posted content doesn't match forecast
- Missing posts from scheduled times
- Duplicate content detected

Reconciliation Process

```
# 1. Check discrepancies
TODAY=$(date -u +\%Y-\%m-\%d)
curl -s "https://hotdog-diaries.vercel.app/admin/schedule/forecast?date=\$TODAY" \
    -H "Authorization: Bearer \$AUTH_TOKEN" | jq '.summary.discrepancies'

# 2. Execute reconciliation
curl -X POST https://hotdog-diaries.vercel.app/admin/schedule/forecast/reconcile \
    -H "Authorization: Bearer \$AUTH_TOKEN" \
    -H "Content-Type: application/json" \
    -d '{"date": "'\$TODAY'"}'

# 3. Verify reconciliation
curl -s "https://hotdog-diaries.vercel.app/admin/schedule/forecast?date=\$TODAY" \
    -H "Authorization: Bearer \$AUTH_TOKEN" | jq '.summary.posted'
```

Backups & Disaster Recovery

Backup Strategy

What Gets Backed Up

Data Type	Frequency	Retention	Location
Database	Daily 02:00 UTC	30 days	Supabase Auto-Backup
Environment Config	On change	Indefinite	Vercel Settings
Application Code	On commit	Indefinite	GitHub Repository
API Keys/Secrets	Manual	Secure vault	Password Manager

Supabase Database Backups

Automated Backups

- Daily snapshots at 02:00 UTC
- Point-in-time recovery available (7 days)
- Automatic cleanup after 30 days

Manual Backup Triggers

Create on-demand backup before major changes
./scripts/backup-supabase.sh

Expected output:

Backup created: supabase_backup_2025-10-16_14-30-00.sql

Location: /backups/2025/10/supabase_backup_2025-10-16_14-30-00.sql

Content Assets Management

- Static Assets: Stored in Vercel/Next.js public directory
- External Images: Cached via proxy endpoints
- No persistent file storage: All content links to external sources

Disaster Recovery Procedures

Database Restore Process

♣ Pre-Restore Checklist

- Identify exact restore point needed
- Notify team of downtime window

- Verify backup integrity
- Prepare rollforward plan

Restore Execution

```
# 1. Dry run (shows commands, doesn't execute)
./scripts/restore-supabase.sh --backup-date 2025-10-15

# 2. Execute restore (requires confirmation)
./scripts/restore-supabase.sh --backup-date 2025-10-15 --confirm

# 3. Verify restore
curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $AUTH_TOKEN" | jq '.database'
```

Complete Environment Recovery

Environment Recreation Steps

1. Vercel Project Setup

```
# Link to existing project or create new

vercel link

# Set environment variables

vercel env add NODE_ENV production

vercel env add DATABASE_URL $SUPABASE_CONNECTION_STRING

# ... (add all required env vars)
```

2. Database Recovery

```
# Restore from latest backup
./scripts/restore-supabase.sh --latest --confirm
```

3. Deploy Application

```
# Deploy current main branch
vercel --prod

# Verify deployment
./scripts/smoke.sh
```

Recovery Time Objectives (RTO)

Scenario	Target RTO	Procedure	
Application Issues	15 minutes	Rollback deployment	
Database Corruption	2 hours	Restore from backup	
Complete Platform Loss	4 hours	Full environment recreation	
Vendor Outage	N/A	Monitor vendor status	

Verification Checklists

After Restore Verification

Database Integrity Check

Functional Verification

Admin authentication working
 Content forecast generation
 Platform diversity calculations
 Scheduled posting functionality
 API rate limiting operational

After Secrets Rotation Verification

Authentication Check

```
# 1. New JWT validation

curl -s https://hotdog-diaries.vercel.app/admin/health/auth-token \
    -H "Authorization: Bearer $NEW_AUTH_TOKEN"

# 2. API key validation

curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $NEW_AUTH_TOKEN" | jq '.apis'

# 3. Database connection with new credentials

curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $NEW_AUTH_TOKEN" | jq '.database.connection_pool'
```

Service Integration Check

All external API keys functional
 Database credentials working
 Admin panel accessible
 Scheduled jobs running
 CI/CD pipeline operational

After Migration Verification

Schema Validation

```
# 1. Migration completion check
curl -s https://hotdog-diaries.vercel.app/admin/health/deep \
    -H "Authorization: Bearer $AUTH_TOKEN" | jq '.database.migration_status'

# 2. Table structure validation
curl -s https://hotdog-diaries.vercel.app/admin/schema/verify \
    -H "Authorization: Bearer $AUTH_TOKEN"

# 3. Data integrity verification
curl -s https://hotdog-diaries.vercel.app/api/system/metrics | jq '{
    content_queue: .queue.total_content,
    scheduled_posts: .scheduler.scheduled_count,
    posted_content: .posting.total_posted
}'
```

Performance Check

Emergency Contacts & Escalation

On-Call Rotation

• Primary: Platform Engineering Team

• Secondary: DevOps Team

• Escalation: Engineering Leadership

Vendor Support Contacts

• Vercel: Support through dashboard

• **Supabase**: support@supabase.com

• GitHub: Support through dashboard

Incident Communication Channels

• Slack: #incidents (urgent) / #platform-alerts (monitoring)

• Email: platform-team@company.com

• PagerDuty: Critical alerts only

Appendix

Related Documentation

- API Documentation
- OpenAPI Specification
- Secrets Management
- Development Setup

Scripts Reference

- ./scripts/smoke.sh Health check suite
- ./scripts/backup-supabase.sh Database backup
- ./scripts/restore-supabase.sh Database restore
- ./scripts/poke-admin.sh Quick admin check
- ./scripts/rotate-secrets.sh Secrets rotation

Quick Commands Cheat Sheet

- ightharpoonup For emergencies, follow the incident response procedures above.
- For questions, consult the API documentation or contact the platform team.