🚀 WritgoAI Deployment Guide

Platform: Render.com
Datum: 14 December 2025

📋 Prerequisites

Before deploying, ensure you have:

1. ✅ Render.com account
2. ✅ GitHub repository access (Mikeyy1405/Writgoai.nl)
3. ✅ Supabase project (database)
4. ✅ AIML API key
5. ✅ GetLate.dev API key (optional)
6. ✅ Domain: writgoai.nl (configured in Render)

🔐 Step 1: Environment Variables

In Render Dashboard, configure these environment variables:

Required Variables:

```
# Application
NODE_ENV=production
NEXTAUTH_URL=https://writgoai.nl
NEXTAUTH_SECRET=<generate: openssl rand -base64 32>

# Cron Jobs Security
CRON_SECRET=<generate: openssl rand -base64 32>

# Database (Supabase)
NEXT_PUBLIC_SUPABASE_URL=https://xxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJxxx...
SUPABASE_SERVICE_ROLE_KEY=eyJxxx...

# AI/ML API (Primary)
AIML_API_KEY=<your-aiml-api-key>
AIML_API_URL=https://api.aimlapi.com/v1

# Social Media (Optional)
LATE_DEV_API_KEY=<your-getlate-api-key>
GETLATE_API_KEY=<your-getlate-api-key>
```

Optional Variables:

```
# OpenAI (Fallback)
OPENAI_API_KEY=<your-openai-key>

# Email (SMTP)
SMTP_HOST=writgoai.nl
SMTP_PORT=587
SMTP_USER=info@writgoai.nl
SMTP_PASS=<smtp-password>

# AWS S3 (Storage)
AWS_BUCKET_NAME=<bucket-name>
AWS_FOLDER_PREFIX=writgo/

# Debug
DEBUG=false
LOG_LEVEL=info
```

📦 Step 2: Deploy Web Application

Option A: Via render.yaml (Recommended)

1. In Render Dashboard: **“New” → “Blueprint”**
2. Select repository: **Mikeyy1405/Writgoai.nl**
3. Branch: **main**
4. Blueprint file detected: **render.yaml**
5. Review services:
   - ✅ writgoai-web (Web Service)
   - ✅ writgoai-daily-content (Cron)
   - ✅ writgoai-publish-scheduled (Cron)
   - ✅ writgoai-autopilot-morning (Cron)
   - ✅ writgoai-autopilot-evening (Cron)
6. Click **“Apply”**
7. Wait for deployment (5-10 minutes)

Option B: Manual Setup

1. In Render Dashboard: **“New” → “Web Service”**
2. Connect repository: **Mikeyy1405/Writgoai.nl**
3. Configure:
   - **Name:** writgoai-web
   - **Region:** Frankfurt (or closest to users)
   - **Branch:** main
   - **Root Directory:** (leave empty)
   - **Environment:** Node
   - **Build Command:** `cd nextjs_space && npm install && npm run build`
   - **Start Command:** `cd nextjs_space && npm start`
   - **Plan:** Starter or higher
4. Add environment variables (see Step 1)
5. Click **“Create Web Service”**

⏰ Step 3: Setup Cron Jobs

Option A: Via render.yaml

Already included if you used Blueprint deployment!

Option B: Manual Setup

In Render Dashboard: **“New” → “Cron Job”**

Cron Job 1: Daily Content Generation

```
Name: writgoai-daily-content
Environment: Node
Region: Frankfurt
Schedule: 0 9 * * *
Command:
  curl -X POST \
    -H "Authorization: Bearer $CRON_SECRET" \
    -H "Content-Type: application/json" \
    $NEXTAUTH_URL/api/cron/daily-content-generation
```

Cron Job 2: Publish Scheduled Articles

```
Name: writgoai-publish-scheduled
Environment: Node
Region: Frankfurt
Schedule: 0 * * * *
Command:
  curl -X POST \
    -H "Authorization: Bearer $CRON_SECRET" \
    -H "Content-Type: application/json" \
    $NEXTAUTH_URL/api/cron/publish-scheduled-articles
```

Cron Job 3: Autopilot Morning

```
Name: writgoai-autopilot-morning
Environment: Node
Region: Frankfurt
Schedule: 0 6 * * *
Command:
  curl -X POST \
    -H "Authorization: Bearer $CRON_SECRET" \
    -H "Content-Type: application/json" \
    $NEXTAUTH_URL/api/cron/autopilot-projects
```

Cron Job 4: Autopilot Evening

```
Name: writgoai-autopilot-evening
Environment: Node
Region: Frankfurt
Schedule: 0 18 * * *
Command:
  curl -X POST \
    -H "Authorization: Bearer $CRON_SECRET" \
    -H "Content-Type: application/json" \
    $NEXTAUTH_URL/api/cron/autopilot-projects
```

Note: Make sure to add environment variables to each cron job:
- NEXTAUTH_URL
- CRON_SECRET

🌐 Step 4: Configure Domain

1. In Render Dashboard → writgoai-web service
2. Click **“Settings” → “Custom Domain”**
3. Add domain: **writgoai.nl**
4. Render will provide DNS records
5. Add records to your DNS provider:
   ```
   Type: CNAME
     Name: @
     Value: writgoai-web.onrender.com
   ```
6. Wait for SSL certificate (automatic)

🧪 Step 5: Test Deployment

1. Test Web Application

```
# Visit homepage
https://writgoai.nl

# Expected: Landing page loads
```

2. Test API Health

```
curl https://writgoai.nl/api/health

# Expected: { "status": "ok", "timestamp": "..." }
```

3. Test Cron Endpoints

```
# Daily content generation
curl -X GET \
  -H "Authorization: Bearer YOUR_CRON_SECRET" \
  https://writgoai.nl/api/cron/daily-content-generation

# Expected: { "status": "operational", "autopilotProjects": N }

# Publish scheduled
curl -X GET \
  -H "Authorization: Bearer YOUR_CRON_SECRET" \
  https://writgoai.nl/api/cron/publish-scheduled-articles

# Expected: { "status": "operational", "scheduledPosts": N }
```

4. Test WordPress Integration

1. Login as admin: https://writgoai.nl/admin
2. Go to Projects → Select/Create project
3. Configure WordPress (URL, username, app password)
4. Click **"Test Connection"**
5. Expected: ✅ "Connection successful"

5. Test GetLate Integration

1. In admin panel or project settings
2. Social Media → Test GetLate Connection
3. Expected: ✅ "Connected - X accounts"

6. Test Content Generation

1. Go to Project → Content → Generate Blog
2. Input: Topic, keywords
3. Click **"Generate"**
4. Wait 2-4 minutes
5. Expected: Complete blog with images & SEO

7. Test Autopilot (Manual Trigger)

```
curl -X POST \
  -H "Authorization: Bearer YOUR_CRON_SECRET" \
  -H "Content-Type: application/json" \
  https://writgoai.nl/api/cron/autopilot-projects

# Check response for processed projects
# Check Render logs for detailed output
```

📊 Step 6: Monitor & Verify

Check Render Logs:

1. Render Dashboard → writgoai-web → Logs

2. Look for:
   - ✅ `Server started on port 3000`
   - ✅ `[Database] Connected to Supabase`
   - ✅ No critical errors

Check Cron Job Logs:

1. Render Dashboard → Cron Jobs → [Job Name] → Logs
2. After scheduled time, verify:
   - ✅ Job executed
   - ✅ Successful response from API
   - ✅ No errors

Monitor for 24 Hours:

- ✅ Web application stays up
- ✅ Cron jobs execute on schedule
- ✅ No memory leaks or crashes
- ✅ API response times < 5s

Check Database:

In Supabase dashboard:

```sql
-- Check autopilot jobs
SELECT * FROM "AutopilotJob"
ORDER BY "createdAt" DESC
LIMIT 10;

-- Check recent blog posts
SELECT * FROM "BlogPost"
WHERE status = 'published'
ORDER BY "publishedAt" DESC
LIMIT 10;

-- Check projects with autopilot
SELECT id, name, "autopilotEnabled", "autopilotLastRun"
FROM "Project"
WHERE "autopilotEnabled" = true;
```

🔧 Troubleshooting

Issue: Build fails

```bash
# Check build logs in Render
# Common causes:
- Missing environment variables
- Node version mismatch
- Dependency conflicts

# Solution:
1. Verify all env vars are set
2. Try: npm install --legacy-peer-deps
3. Check Node version (should be 18.x or 20.x)
```

Issue: Cron jobs not running

```
# Check:
1. CRON_SECRET is set correctly
2. NEXTAUTH_URL is correct (https://writgoai.nl)
3. Cron schedule syntax is correct
4. Environment variables are added to cron job

# Test manually:
curl -X POST \
  -H "Authorization: Bearer YOUR_CRON_SECRET" \
  https://writgoai.nl/api/cron/daily-content-generation

# If 401: CRON_SECRET mismatch
# If 500: Check app logs for error
```

Issue: WordPress publish fails

```
# Check:
1. WordPress REST API is enabled
2. Application Password (not regular password!)
3. WordPress URL is correct (with https://)
4. User has admin/editor permissions

# Test in WordPress:
1. Go to Users → Profile
2. Scroll to Application Passwords
3. Generate new password
4. Use in WritgoAI config

# Check error in database:
SELECT "publishError" FROM "BlogPost" WHERE status = 'failed';
```

Issue: GetLate social media fails

```
# This is non-critical (graceful fallback)
# Check:
1. LATE_DEV_API_KEY is correct
2. Accounts are connected in GetLate
3. Rate limits not exceeded

# Workaround:
- Social media posting is optional
- Can be done manually
- App continues without blocking
```

Issue: Out of memory

```
# Symptom: App crashes, restarts frequently
# Solution:
1. Upgrade Render plan (more RAM)
2. Optimize image generation (smaller sizes)
3. Add memory limits to API routes
4. Check for memory leaks in logs
```

📈 Scaling Considerations

When to scale up:

- More than 50 autopilot projects
- API response times > 5 seconds
- Memory usage > 80%
- Cron jobs timing out

Scaling options:

1. **Vertical:** Upgrade Render plan (more CPU/RAM)
2. **Horizontal:** Multiple web service instances (Render Pro)
3. **Database:** Upgrade Supabase plan
4. **Caching:** Add Redis for API responses
5. **CDN:** Cloudflare for static assets

🔒 Security Checklist

- ✅ NEXTAUTH_SECRET is strong (32+ chars)
- ✅ CRON_SECRET is strong and secret
- ✅ Supabase keys are not exposed
- ✅ WordPress uses Application Password (not admin)
- ✅ Cron endpoints require auth
- ✅ HTTPS is enabled (automatic on Render)
- ✅ Environment variables are not in git
- ✅ Row Level Security (RLS) enabled in Supabase

📅 Maintenance Schedule

Daily:

- ✅ Check Render logs for errors
- ✅ Verify cron jobs executed
- ✅ Monitor autopilot success rate

Weekly:

- ✅ Review failed jobs (autopilot logs)
- ✅ Check credit usage per client
- ✅ Verify WordPress connections
- ✅ Review API performance

Monthly:

- ✅ Database backup (Supabase automatic)
- ✅ Update dependencies (npm update)
- ✅ Review and optimize costs

- ✅ Client feedback & feature requests

🆘 Emergency Contacts

If app goes down:

1. Check Render status: https://status.render.com
2. Check Supabase status: https://status.supabase.com
3. Review recent deployments (rollback if needed)
4. Check logs for critical errors

Rollback procedure:

```
# In Render Dashboard:
1. Go to writgoai-web → Deploys
2. Find last working deploy
3. Click "Rollback to this deploy"
4. Wait 2-3 minutes
5. Verify app is back online
```

✅ Post-Deployment Checklist

- [ ] Web application is accessible
- [ ] Admin login works
- [ ] Client login works
- [ ] WordPress connection works
- [ ] GetLate connection works (optional)
- [ ] Content generation works
- [ ] WordPress publish works
- [ ] Social media posting works (optional)
- [ ] Cron jobs are scheduled
- [ ] Logs show no critical errors
- [ ] Domain SSL certificate is active
- [ ] Database connections are stable
- [ ] All environment variables are set
- [ ] Backup strategy is in place

🎓 Additional Resources

- **Render Docs:** https://render.com/docs
- **Next.js Deployment:** https://nextjs.org/docs/deployment
- **Supabase Docs:** https://supabase.com/docs
- **AIML API Docs:** https://aimlapi.com/docs
- **GetLate.dev Docs:** https://getlate.dev/docs

Deployment Complete! 🎉

For issues or questions, check:
- IMPLEMENTATION_SUMMARY.md
- README.md
- Render Dashboard Logs

Built with ❤️ by the WritgoAI team