

Production Deployment

Drop AWS Amplify Deployment Guide

“ **Rebrand note (2026-02-14):** Originally titled "FontelePay". Product rebranded to **Drop**. Some env var references (Swan, Stripe) are FUTURE integrations — Drop uses a PSD2 pass-through model. See [Drop CLAUDE.md](#).

This guide covers deploying Drop to AWS Amplify in the Frankfurt (eu-central-1) region.

Prerequisites

- AWS Account with Amplify access
- GitHub repository with Drop code
- Environment variables ready (see `.env.example`)

Step 1: Create Amplify App

1. Go to [AWS Amplify Console](#)
2. Ensure you're in **eu-central-1 (Frankfurt)** region
3. Click **Create new app**
4. Select **Host web app**

Step 2: Connect Repository

1. Choose **GitHub** as your Git provider
2. Authorize AWS Amplify to access your GitHub account
3. Select the **Drop** repository
4. Choose the branch to deploy (e.g., `main` or `production`)

Step 3: Configure Build Settings

Amplify will auto-detect Next.js. Verify the settings match `amplify.yml`:

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
      - .next/cache/**/*
```

Step 4: Configure Environment Variables

In Amplify Console, go to **App settings** > **Environment variables** and add:

Required Variables

Variable	Description	Example
<code>NODE_ENV</code>	Environment	<code>production</code>
<code>NEXT_PUBLIC_APP_URL</code>	Your app URL	<code>https://drop.amplifyapp.com</code>

Swan BaaS

Variable	Description
----------	-------------

SWAN_API_URL	https://api.swan.io (production)
SWAN_CLIENT_ID	OAuth2 Client ID
SWAN_CLIENT_SECRET	OAuth2 Client Secret
SWAN_PROJECT_ID	Project ID
SWAN_WEBHOOK_SECRET	Webhook validation secret

Stripe

Variable	Description
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY	Publishable key (pk_live_...)
STRIPE_SECRET_KEY	Secret key (sk_live_...)
STRIPE_WEBHOOK_SECRET	Webhook secret (whsec_...)

Sumsup KYC

Variable	Description
SUMSUB_APP_TOKEN	App token
SUMSUB_SECRET_KEY	Secret key
SUMSUB_WEBHOOK_SECRET	Webhook secret
SUMSUB_LEVEL_NAME	KYC flow level

Database

Variable	Description
DATABASE_URL	PostgreSQL connection string
REDIS_URL	Redis connection string

Authentication

Variable	Description
JWT_SECRET	Min 32 characters
SESSION_SECRET	Min 32 characters

Step 5: Configure Next.js for Standalone Output

Update `next.config.ts` to enable standalone output for optimal Amplify deployment:

```
import type { NextConfig } from "next";

const nextConfig: NextConfig = {
  output: 'standalone',
};

export default nextConfig;
```

Step 6: Deploy

1. Click **Save and deploy**
2. Monitor the build in the Amplify Console
3. Once complete, your app will be available at `https://<branch>.<app-id>.amplifyapp.com`

Step 7: Configure Custom Domain (Optional)

1. Go to **App settings > Domain management**
2. Click **Add domain**
3. Enter your domain (e.g., `app.getdrop.no`)
4. Follow DNS configuration instructions
5. SSL certificate is automatically provisioned

Step 8: Set Up Branch Deployments

For staging/production workflows:

1. Go to **App settings > General**
2. Click **Edit**
3. Enable **Branch auto-detection**

4. Configure branch patterns:

- `main` -> Production
- `staging` -> Staging
- `feature/*` -> Preview environments

Monitoring & Health Checks

Health Endpoint

The app exposes `/api/health` for load balancer health checks:

```
curl https://your-app.amplifyapp.com/api/health
```

Response:

```
{
  "status": "healthy",
  "timestamp": "2026-02-05T12:00:00.000Z",
  "version": "0.1.0",
  "uptime": 3600,
  "checks": {}
}
```

CloudWatch Logs

1. Go to **App settings > Monitoring**
2. View build logs and access logs
3. Set up CloudWatch alarms for errors

Troubleshooting

Build Fails

1. Check build logs in Amplify Console
2. Verify `package.json` scripts are correct
3. Ensure all dependencies are in `package.json`

Environment Variables Not Working

1. Verify variables are set in Amplify Console
2. Remember: `NEXT_PUBLIC_` prefix required for client-side access
3. Redeploy after changing environment variables

502/503 Errors

1. Check `/api/health` endpoint
2. Review CloudWatch logs
3. Verify database connections are correct
4. Check memory limits (adjust if needed)

Cold Starts

For serverless functions, cold starts may occur. Mitigate by:

1. Using connection pooling for databases
2. Keeping functions warm with scheduled pings
3. Optimizing bundle size

Security Checklist

- All secrets in Environment Variables (not in code)
- HTTPS enforced (automatic in Amplify)
- CORS configured correctly
- Rate limiting implemented
- Webhook signatures validated
- No sensitive data in logs

Cost Optimization

- Use `cache.paths` in `amplify.yml` to speed up builds
- Enable CloudFront caching for static assets
- Monitor build minutes usage
- Consider reserved concurrency for predictable traffic

Rollback

To rollback to a previous deployment:

1. Go to **Deployments** in Amplify Console
2. Find the previous successful deployment
3. Click **Redeploy this version**

Support

- [AWS Amplify Documentation](#)
- [Next.js on AWS Amplify](#)
- [Drop Internal Docs](#)

Revision #4

Created 2026-02-18 08:44:29 UTC by John

Updated 2026-05-23 10:58:21 UTC by John