



# Deployment Guide - Turion API Nav

Created by **Bolla Network** (<https://bolla.network>)

This guide provides step-by-step instructions for deploying Turion API Nav to production.



## Table of Contents

1. [Pre-Deployment Checklist](#)
2. [Deploy to Vercel](#)
3. [Configure Environment Variables](#)
4. [Set Up Stripe Webhooks](#)
5. [Configure Custom Domain](#)
6. [Post-Deployment Verification](#)
7. [Troubleshooting](#)



## Pre-Deployment Checklist

Before deploying, ensure you have:

- [ ] Supabase project created and `supabase-schema.sql` executed
- [ ] Stripe account set up with products and prices created
- [ ] All environment variables documented
- [ ] Local build passes: `npm run build`
- [ ] GitHub repository connected (if using Git deployment)



## Deploy to Vercel

### Option 1: Deploy via Vercel Dashboard (Recommended)

#### 1. Go to Vercel

<https://vercel.com/new>

#### 2. Import Project

- Click “Import Git Repository”
- Select your GitHub repository
- Click “Import”

#### 3. Configure Project

- **Framework Preset:** Next.js (auto-detected)
- **Root Directory:** `./` (default)
- **Build Command:** `npm run build`
- **Output Directory:** `.next`

#### 4. Add Environment Variables

Add all variables from the [Environment Variables](#) section

#### 5. Deploy

Click “Deploy” and wait for the build to complete

## Option 2: Deploy via Vercel CLI

#### 1. Install Vercel CLI

```
bash
```

```
  npm install -g vercel
```

#### 2. Login to Vercel

```
bash
```

```
  vercel login
```

#### 3. Deploy

```
```bash
```

```
  # Development preview
```

```
  vercel
```

```
# Production deployment
```

```
vercel --prod
```

```
```
```

#### 1. Set Environment Variables

```
bash
```

```
  vercel env add NEXT_PUBLIC_SUPABASE_URL
```

```
  vercel env add NEXT_PUBLIC_SUPABASE_ANON_KEY
```

```
  vercel env add SUPABASE_SERVICE_ROLE_KEY
```

```
  vercel env add STRIPE_SECRET_KEY
```

```
  vercel env add NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY
```

```
  vercel env add STRIPE_WEBHOOK_SECRET
```

```
  vercel env add NEXT_PUBLIC_APP_URL
```

```
  vercel env add NEXTAUTH_URL
```

```
  vercel env add NEXTAUTH_SECRET
```

---

## Configure Environment Variables

### Required Environment Variables

| Variable                           | Where to Get                                | Production Value  |
|------------------------------------|---|---|
| NEXT_PUBLIC_SUPABASE_URL           | Supabase Dashboard → Settings → API         | <a href="https://xxx.supabase.co">https://xxx.supabase.co</a>       |
| NEXT_PUBLIC_SUPABASE_ANON_KEY      | Supabase Dashboard → Settings → API         | eyJhbGciOiJIUzI1...   |
| SUPABASE_SERVICE_ROLE_KEY          | Supabase Dashboard → Settings → API         | eyJhbGciOiJIUzI1...   |
| STRIPE_SECRET_KEY                  | Stripe Dashboard → Developers → API Keys    | sk_live_...   |
| NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY | Stripe Dashboard → Developers → API Keys    | pk_live_...   |
| STRIPE_WEBHOOK_SECRET              | Stripe Dashboard → Webhooks → Your Endpoint | whsec_...   |
| NEXT_PUBLIC_APP_URL                | Your domain                                 | <a href="https://www.turion.network">https://www.turion.network</a> |
| NEXTAUTH_URL                       | Your domain                                 | <a href="https://www.turion.network">https://www.turion.network</a> |
| NEXTAUTH_SECRET                    | Generate random string                      | <code>openssl rand -base64 32</code>                                |

### Generating NEXTAUTH\_SECRET

```
# On macOS/Linux
openssl rand -base64 32

# Or use Node.js
node -e "console.log(require('crypto').randomBytes(32).toString('base64'))"
```

### Setting Variables in Vercel Dashboard

1. Go to your Vercel project
2. Navigate to **Settings → Environment Variables**
3. Add each variable with:
  - **Key:** Variable name
  - **Value:** Variable value
  - **Environment:** Select `Production`, `Preview`, and `Development` as needed

## Set Up Stripe Webhooks

### Step 1: Create Webhook Endpoint

1. Go to [Stripe Dashboard → Developers → Webhooks](https://dashboard.stripe.com/webhooks) (<https://dashboard.stripe.com/webhooks>)
2. Click “Add endpoint”
3. Enter endpoint URL:

`https://www.turion.network/api/stripe/webhook`

Replace `www.turion.network` with your actual domain

### Step 2: Select Events

Subscribe to these events:

| Event  | Description                           |
|--|---------------------------------------|
| <input checked="" type="checkbox"/> <code>checkout.session.completed</code>    | Customer completed payment            |
| <input checked="" type="checkbox"/> <code>customer.subscription.created</code> | New subscription created              |
| <input checked="" type="checkbox"/> <code>customer.subscription.updated</code> | Plan changed or subscription modified |
| <input checked="" type="checkbox"/> <code>customer.subscription.deleted</code> | Subscription cancelled                |
| <input checked="" type="checkbox"/> <code>invoice.paid</code>                  | Successful payment                    |
| <input checked="" type="checkbox"/> <code>invoice.payment_failed</code>        | Failed payment                        |

### Step 3: Get Signing Secret

1. After creating the webhook, click on the endpoint
2. Under **Signing secret**, click “Reveal”
3. Copy the `whsec_...` value
4. Add to Vercel as `STRIPE_WEBHOOK_SECRET`

### Step 4: Verify Webhook

Test the webhook with Stripe CLI:

```
# Install Stripe CLI
brew install stripe/stripe-cli/stripe

# Login
stripe login

# Trigger a test event
stripe trigger checkout.session.completed
```

## Configure Custom Domain

### In Vercel

1. Go to **Settings → Domains**
2. Add your domain: `www.turion.network`
3. Follow the DNS configuration instructions

### DNS Configuration

Add these records to your DNS provider:

| Type  | Name | Value                |
|-------|------|----------------------|
| A     | @    | 76.76.21.21          |
| CNAME | www  | cname.vercel-dns.com |

### SSL Certificate

Vercel automatically provisions SSL certificates. No action needed.

### Update Environment Variables

After domain is connected, update:

```
NEXT_PUBLIC_APP_URL=https://www.turion.network
NEXTAUTH_URL=https://www.turion.network
```

## Post-Deployment Verification

### 1. Test Main Pages

- [ ] Homepage loads: `https://www.turion.network`
- [ ] Login page works: `https://www.turion.network/auth/login`
- [ ] Dashboard loads (after login): `https://www.turion.network/dashboard`

### 2. Test Authentication

- [ ] User can sign up
- [ ] User can log in
- [ ] User can log out
- [ ] Password reset works

### 3. Test Stripe Integration

- [ ] Pricing page shows plans
- [ ] Checkout redirects to Stripe
- [ ] Webhooks receive events (check Stripe Dashboard → Webhooks → Logs)
- [ ] Subscription updates in database

## 4. Test API

```
# Test API health
curl https://www.turion.network/api

# Test with API key (after creating one)
curl -H "Authorization: Bearer tur_your_api_key" \
https://www.turion.network/api/usage
```

## 5. Check Logs

In Vercel Dashboard:

1. Go to **Deployments**
2. Click on latest deployment
3. View **Function Logs** for any errors



## Troubleshooting

### Build Fails

**Error:** TypeScript errors

Solution: Check `if ignoreBuildErrors is disabled in next.config.ts`

**Error:** Module not found

```
# Clear node_modules and reinstall
rm -rf node_modules .next
npm install
npm run build
```

### Webhook Issues

**Error:** 400 Invalid Signature

1. **Verify** STRIPE\_WEBHOOK\_SECRET matches the endpoint's signing secret
2. Make sure you're using the production webhook secret, **not** test
3. Redeploy after updating the secret

**Error:** Webhooks not receiving

1. Check Stripe Dashboard → Webhooks → Logs **for** delivery status
2. **Verify** the URL is correct **and** accessible
3. Check Vercel function logs **for** errors

### Database Issues

**Error:** Permission denied on Supabase

1. Check RLS policies in Supabase
2. **Verify** SUPABASE\_SERVICE\_ROLE\_KEY is set correctly
3. **For** webhooks, service role key bypasses RLS

## Environment Variable Issues

**Error:** undefined environment variables

1. **Verify** all variables are set in Vercel
2. Check variable names match exactly (case-sensitive)
3. Redeploy after adding **new** variables



## Monitoring

### Vercel Analytics

1. Go to **Analytics** in your Vercel project
2. Enable Web Analytics for performance insights

### Error Tracking

Consider integrating:

- [Sentry](https://sentry.io) (<https://sentry.io>) for error tracking
- [LogRocket](https://logrocket.com) (<https://logrocket.com>) for session replay

### Stripe Dashboard

Monitor payments at:

- [Stripe Dashboard](https://dashboard.stripe.com) (<https://dashboard.stripe.com>)
- Check **Webhooks → Logs** for delivery status



## Continuous Deployment

With Git integration, Vercel automatically deploys:

- **Production:** On push to `main` branch
- **Preview:** On pull requests

### Deployment Settings

Configure in Vercel → **Settings** → **Git**:

- Production Branch: `main`
- Preview Branches: All other branches



## Support

Need help? Contact:

- [Website: bolla.network](https://bolla.network) (<https://bolla.network>)
- [Email: support@bolla.network](mailto:support@bolla.network)

---

Made with ❤️ by **Bolla Network** (<https://bolla.network>)