

Genesis Provenance - Vercel Deployment Guide

This comprehensive guide walks you through deploying Genesis Provenance to Vercel with all necessary configurations for production use.

Prerequisites

What You Need:

- Vercel account ([sign up here](https://vercel.com/signup) (<https://vercel.com/signup>))
 - GitHub account (or GitLab/Bitbucket)
 - GoDaddy domain (genesisprovenance.com)
 - PostgreSQL database (Vercel Postgres, Neon, or Supabase)
 - AWS S3 bucket (for file storage)
 - Microsoft 365 email account
-

Part 1: Database Setup

Option A: Vercel Postgres (Recommended for Simplicity)

1. **Navigate to Vercel Dashboard** → Your Project → Storage

2. **Create Postgres Database:**

- Click “Create Database”
- Select “Postgres”
- Choose region closest to your users
- Name: genesis-provenance-db

3. **Get Connection String:**

- Copy the `DATABASE_URL` connection string
- Format: `postgres://user:password@host:5432/database`

Option B: External PostgreSQL (Neon, Supabase, Railway)

Neon (Recommended for Scalability):

1. Go to neon.tech (<https://neon.tech>)
2. Create new project: genesis-provenance
3. Get connection string from dashboard
4. Copy `DATABASE_URL`

Supabase:

1. Create project at supabase.com (<https://supabase.com>)
 2. Go to Settings → Database → Connection string
 3. Use “Transaction” connection mode for Prisma
-

Part 2: AWS S3 Storage Setup

1. Create S3 Bucket

```
# Using AWS Console
1. Go to AWS S3 Console
2. Click "Create bucket"
3. Bucket name: genesis-provenance-assets
4. Region: us-east-1 (or your preferred region)
5. Block all public access: OFF (we'll use signed URLs)
6. Enable versioning: ON
7. Enable encryption: AES-256
```

2. Create IAM User for S3 Access

```
# AWS Console → IAM → Users → Add User
1. User name: genesis-provenance-s3-user
2. Access type: Programmatic access
3. Attach policy: AmazonS3FullAccess (or create custom policy below)
4. Download CSV with ACCESS_KEY_ID and SECRET_ACCESS_KEY
```

Custom S3 Policy (More Secure):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3>ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::genesis-provenance-assets",
        "arn:aws:s3:::genesis-provenance-assets/*"
      ]
    }
  ]
}
```

3. Configure S3 CORS

```
[  
  {  
    "AllowedHeaders": ["*"],  
    "AllowedMethods": ["GET", "PUT", "POST", "DELETE"],  
    "AllowedOrigins": [  
      "https://genesisprovenance.com",  
      "https://www.genesisprovenance.com",  
      "https://app.genesisprovenance.com"  
    ],  
    "ExposeHeaders": ["ETag"],  
    "MaxAgeSeconds": 3000  
  }  
]
```

Part 3: Vercel Project Setup

1. Push Code to GitHub

```
# In your project directory  
cd /home/ubuntu/genesis_provenance/nextjs_space  
  
# Initialize git (if not already done)  
git init  
git add .  
git commit -m "Initial commit - Genesis Provenance Phase 1+2"  
  
# Create GitHub repo (via GitHub website or CLI)  
# Then push:  
git remote add origin https://github.com/your-username/genesis-provenance.git  
git branch -M main  
git push -u origin main
```

2. Import Project to Vercel

1. Go to vercel.com/new (<https://vercel.com/new>)

2. Import Git Repository:

- Select your GitHub repository
- Click “Import”

3. Configure Project:

- Project Name: genesis-provenance
- Framework Preset: Next.js (auto-detected)
- Root Directory: ./ (if code is in root) or ./nextjs_space
- Build Command: yarn build (default)
- Output Directory: .next (default)

4. Add Environment Variables (see Part 4 below)

5. Deploy:

- Click “Deploy”
- Wait 2-3 minutes for initial deployment

Part 4: Environment Variables

Required Environment Variables for Vercel

Go to: Vercel Dashboard → Your Project → Settings → Environment Variables

Database

```
DATABASE_URL="postgresql://user:password@host:5432/database?sslmode=require"
```

Authentication (NextAuth.js)

```
NEXTAUTH_URL="https://app.genesisprovenance.com"
NEXTAUTH_SECRET="

```

AWS S3 Storage

```
AWS_ACCESS_KEY_ID=""
AWS_SECRET_ACCESS_KEY=""
AWS_REGION="us-east-1"
AWS_BUCKET_NAME="genesis-provenance-assets"
AWS_FOLDER_PREFIX="uploads/" # Optional
```

Node Environment

```
NODE_ENV="production"
```

Future: Stripe (Phase 4)

```
STRIPE_SECRET_KEY="sk_live_..."
STRIPE_PUBLISHABLE_KEY="pk_live_..."
STRIPE_WEBHOOK_SECRET="whsec_..."
```

Future: AI Service (Phase 3)

```
AI_SERVICE_API_KEY=""
AI_SERVICE_ENDPOINT="https://api.openai.com/v1" # or your provider
```

Future: n8n Webhooks (Phase 3)

```
N8N_WEBHOOK_URL="https://your-n8n-instance.com/webhook/"
N8N_API_KEY=""
```

Part 5: Domain Configuration

A. Configure Custom Domains in Vercel

1. **Go to:** Vercel Dashboard → Your Project → Settings → Domains

2. Add Domains:

- genesisprovenance.com (root)
- www.genesisprovenance.com (www)
- app.genesisprovenance.com (application)

3. Vercel will provide DNS records for each domain

B. Update GoDaddy DNS Records

Login to GoDaddy:

1. Go to: <https://dcc.godaddy.com/manage/genesisprovenance.com/dns>
2. Click “Manage” → “DNS”

Add/Update DNS Records:

Root Domain (genesisprovenance.com)

```
Type: A
Name: @
Value: 76.76.21.21 (Vercel's IP - check Vercel for current IP)
TTL: 600 seconds
```

WWW Subdomain (www.genesisprovenance.com)

```
Type: CNAME
Name: www
Value: cname.vercel-dns.com
TTL: 600 seconds
```

App Subdomain (app.genesisprovenance.com)

```
Type: CNAME
Name: app
Value: cname.vercel-dns.com
TTL: 600 seconds
```

API Subdomain (api.genesisprovenance.com) - Future Use

```
Type: CNAME
Name: api
Value: cname.vercel-dns.com
TTL: 600 seconds
```

Note: Vercel-specific values may differ. Always use the exact records shown in your Vercel dashboard.

Part 6: SSL/TLS Certificates

 **Automatic:** Vercel automatically provisions and renews SSL certificates for all domains.

Verify SSL:

1. Wait 10-15 minutes after DNS propagation

2. Visit: <https://genesisprovenance.com>
 3. Check for padlock icon in browser
-

Part 7: Database Migrations on Vercel

After first deployment:

Option A: Via Vercel CLI (Recommended)

```
# Install Vercel CLI
npm i -g vercel

# Login
vercel login

# Link to your project
vercel link

# Run migrations
vercel env pull .env.local
npx prisma migrate deploy
npx prisma generate

# Seed database (if needed)
npx prisma db seed
```

Option B: Via GitHub Actions (Automated)

Create `.github/workflows/deploy.yml`:

```

name: Deploy with Migrations

on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install dependencies
        run: yarn install

      - name: Run Prisma migrations
        run: npx prisma migrate deploy
        env:
          DATABASE_URL: ${{ secrets.DATABASE_URL }}

      - name: Deploy to Vercel
        uses: amondnet/vercel-action@v25
        with:
          vercel-token: ${{ secrets.VERCEL_TOKEN }}
          vercel-org-id: ${{ secrets.VERCEL_ORG_ID }}
          vercel-project-id: ${{ secrets.VERCEL_PROJECT_ID }}

```

Part 8: Post-Deployment Checklist

1. Test All Functionality

- [] Visit <https://genesisprovenance.com> (marketing site)
- [] Test signup: <https://app.genesisprovenance.com/auth/signup>
- [] Test login with admin: john@doe.com / johndoe123
- [] Navigate through dashboard
- [] Submit contact form
- [] Check admin console

2. Verify Database Connection

```

# Check database via Vercel CLI
vercel env pull .env.local
npx prisma studio

```

3. Monitor Logs

- Go to: Vercel Dashboard → Your Project → Logs
- Check for errors during first few hours

4. Set Up Monitoring

Vercel Analytics:

1. Go to: Vercel Dashboard → Your Project → Analytics
2. Enable: Web Analytics
3. Enable: Speed Insights

Sentry (Optional for Error Tracking):

```
yarn add @sentry/nextjs
npx @sentry/wizard -i nextjs
```

Part 9: Microsoft 365 Email Setup

For sending emails from your domain:

DNS Records for Microsoft 365

Add these in GoDaddy DNS:

MX Records:

```
Type: MX
Name: @
Value: genesisprovenance-com.mail.protection.outlook.com
Priority: 0
TTL: 3600
```

SPF Record:

```
Type: TXT
Name: @
Value: v=spf1 include:spf.protection.outlook.com -all
TTL: 3600
```

DKIM Records (Get from Microsoft 365 Admin):

```
Type: CNAME
Name: selector1._domainkey
Value: selector1-genesisprovenance-com._domainkey.genesisprovenance.onmicrosoft.com
TTL: 3600

Type: CNAME
Name: selector2._domainkey
Value: selector2-genesisprovenance-com._domainkey.genesisprovenance.onmicrosoft.com
TTL: 3600
```

DMARC Record:

```
Type: TXT
Name: _dmarc
Value: v=DMARC1; p=quarantine; rua=mailto:dmarc@genesisprovenance.com
TTL: 3600
```

Get Exact Values:

1. Go to: [Microsoft 365 Admin Center](https://admin.microsoft.com) (<https://admin.microsoft.com>)
2. Navigate to: Settings → Domains → genesisprovenance.com
3. Copy exact DNS records provided

Part 10: Continuous Deployment

Automatic: Vercel automatically deploys when you push to `main` branch.

Workflow:

```
# Make changes locally
git add .
git commit -m "Add feature X"
git push origin main

# Vercel automatically:
# 1. Detects push
# 2. Runs build
# 3. Deploys to production
# 4. Updates DNS
```

Preview Deployments:

- Every branch/PR gets a unique preview URL
- Test before merging to main

Part 11: Troubleshooting

Common Issues:

1. Database Connection Errors

```
Error: P1001: Can't reach database server
```

Solution:

- Check `DATABASE_URL` is correct
- Verify database allows connections from `0.0.0.0/0` (Vercel)
- Add `?sslmode=require` to connection string

2. NextAuth Session Errors

```
Error: [next-auth][error][JWT_SESSION_ERROR]
```

Solution:

- Generate new `NEXTAUTH_SECRET` : `openssl rand -base64 32`
- Verify `NEXTAUTH_URL` matches actual domain
- Clear browser cookies

3. S3 Upload Failures

```
Error: Access Denied
```

Solution:

- Verify AWS credentials are correct
- Check S3 bucket policy allows uploads
- Verify CORS configuration

4. Build Failures

```
Error: Module not found
```

Solution:

- Run `yarn install` locally
- Check `package.json` dependencies
- Clear Vercel build cache: Settings → General → Clear Cache

Part 12: Performance Optimization**1. Enable Vercel Speed Insights**

```
yarn add @vercel/speed-insights
```

In `app/layout.tsx`:

```
import { SpeedInsights } from '@vercel/speed-insights/next';

export default function RootLayout({ children }: { children: React.ReactNode }) {
  return (
    <html>
      <body>
        {children}
        <SpeedInsights />
      </body>
    </html>
  );
}
```

2. Configure Next.js for Production

`next.config.js`:

```
module.exports = {
  images: {
    domains: ['genesis-provenance-assets.s3.amazonaws.com'],
    formats: ['image/avif', 'image/webp'],
  },
  compress: true,
  poweredByHeader: false,
};
```

Part 13: Security Best Practices

1. Environment Variables

- Never commit `.env` files
- Use Vercel dashboard for secrets
- Rotate keys every 90 days

2. API Routes

- Implement rate limiting
- Validate all inputs
- Use HTTPS only

3. Database

- Use connection pooling
- Enable SSL mode
- Regular backups

Part 14: Maintenance & Updates

Weekly:

- [] Check Vercel logs for errors
- [] Review analytics
- [] Monitor S3 usage/costs

Monthly:

- [] Review and rotate API keys
- [] Database performance check
- [] Security updates (`yarn upgrade-interactive`)

Quarterly:

- [] Full security audit
- [] Cost optimization review
- [] User feedback review

Support Resources

- **Vercel Docs:** <https://vercel.com/docs>
 - **Next.js Docs:** <https://nextjs.org/docs>
 - **Prisma Docs:** <https://www.prisma.io/docs>
 - **AWS S3 Docs:** <https://docs.aws.amazon.com/s3/>
-

Deployment Complete!

Your Genesis Provenance application is now live at:

- <https://genesisprovenance.com>
- <https://www.genesisprovenance.com>
- <https://app.genesisprovenance.com>

Next Steps:

- Continue with Phase 2 features (asset onboarding, file uploads)
 - Set up n8n workflows for AI integration
 - Configure Stripe for subscriptions
-

Last Updated: November 29, 2025