

# Hostinger Shared Hosting Deployment Guide

## Radians Automation Website - Next.js Application

---

### Prerequisites

#### Hostinger Requirements

- **Hosting Plan:** Premium or Business plan (supports Node.js)
- **Node.js Version:** 18.x or higher
- **Database:** PostgreSQL (already configured with Abacus.AI)
- **SSH Access:** Enabled in Hostinger control panel

#### Local Requirements

- Git installed
  - Node.js 18+ and Yarn installed
  - GitHub repository access
- 

### Deployment Steps

#### Step 1: Prepare Hostinger Environment

##### 1.1 Enable SSH Access

1. Log in to Hostinger control panel (hPanel)
2. Go to **Advanced** → **SSH Access**
3. Enable SSH and note your SSH credentials

##### 1.2 Enable Node.js Application

1. In hPanel, go to **Advanced** → **Node.js**
2. Click **Create Application**
3. Configure:
  - **Node.js version:** 18.x or latest
  - **Application mode:** Production
  - **Application root:** /public\_html (or your preferred directory)
  - **Application URL:** Your domain (e.g., www.radians-automation.com)
  - **Application startup file:** server.js
4. Note the **PORT** assigned by Hostinger (e.g., 3000, 4000, etc.)
5. Click **Create**

---

## Step 2: Configure Environment Variables in Hostinger

1. In the Node.js application settings, find **Environment Variables**
2. Add the following variables:

```
NODE_ENV=production
PORT=<PORT_FROM_HOSTINGER>
HOSTNAME=0.0.0.0

# Database
DATABASE_URL=postgresql://role_1141501478:Lf0HjDn4UIHMHYHtuFsyvLIFcSt
1141501478.db002.hosteddb.reai.io:5432/1141501478?
connect_timeout=15&connection_limit=1&pool_timeout=10

# NextAuth
NEXTAUTH_SECRET=H9kXs55vaTIgXftHfaqBwgRhW10sAHkM
NEXTAUTH_URL=https://www.radians-automation.com

# AWS S3 Storage
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=abacusai-apps-730023ff9c29e9d114c281d8-us-west-2
AWS_FOLDER_PREFIX=8192/
```

**Important:** Replace <PORT\_FROM\_HOSTINGER> with the actual port assigned by Hostinger.

---

## Step 3: Deploy Code to Hostinger via SSH

### 3.1 Connect via SSH

```
ssh your_username@your_server_ip
```

### 3.2 Navigate to Application Directory

```
cd public_html # or your configured application root
```

### 3.3 Clone Repository

```
git clone https://github.com/DennisDiamante/RACSI-Static.git .
```

### 3.4 Navigate to Next.js Directory

```
cd nextjs_space
```

### 3.5 Create .env File

```
cat > .env << 'EOF'
NODE_ENV=production
DATABASE_URL='postgresql://role_1141501478:Lf0HjDn4UIHMHYHtuFsyvLIFcSbcxihc@db-
1141501478.db002.hosteddb.reai.io:5432/1141501478?
connect_timeout=15&connection_limit=1&pool_timeout=10'
```

```
NEXTAUTH_SECRET=H9kXs55vaTIgXftHfaqBwgRhW10sAHkM
NEXTAUTH_URL=https://www.radians-automation.com
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=abacusai-apps-730023ff9c29e9d114c281d8-us-west-2
AWS_FOLDER_PREFIX=8192/
EOF
```

### 3.6 Install Dependencies

```
yarn install
```

### 3.7 Generate Prisma Client

```
yarn prisma generate
```

### 3.8 Build Application

```
yarn build
```

---

## Step 4: Configure Apache Proxy (if needed)

If your Hostinger plan uses Apache:

1. Update the `.htaccess` file in `public/.htaccess`
2. Replace `PORT` with your actual Hostinger Node.js port:

```
RewriteRule ^(.*)$ http://localhost:3000/$1 [P,L]
```

3. Copy `.htaccess` to document root:

```
cp public/.htaccess ../.htaccess
```

---

## Step 5: Start the Application

### Via Hostinger Control Panel

1. Go to **Node.js** section in hPanel
2. Find your application
3. Click **Restart** or **Start**
4. Application should start on the configured port

### Via SSH (if direct control needed)

```
cd nextjs_space
yarn start
```

---

## Step 6: Seed the Database

1. Visit the seed endpoint:

```
https://www.radians-automation.com/api/seed-database?
secret=RadiansSeeding2024
```

2. You should see:

```
{
  "success": true,
  "message": "Database seeded successfully!",
  "credentials": {
    "developer": "denniscortezdiamante@gmail.com /
    RadDeveloper2024!Dennis#Secure",
    "superAdmin": "info@radians-automation.com /
    RadSuperAdmin2024!Info#Secure",
    "admin": "john@doe.com / SecureTest2024!RadiansAuto"
  }
}
```

---

## Step 7: Configure Domain (if needed)

1. In hPanel, go to **Domains**
  2. Point your domain to the application:
    - If using subdomain: Create A record pointing to server IP
    - If using main domain: Update nameservers or A record
  3. Update NEXTAUTH\_URL environment variable with your domain
  4. Restart the application
- 

## Testing

### Test Public Pages

- **Home:** <https://www.radians-automation.com/>
- **About:** <https://www.radians-automation.com/about>
- **Projects:** <https://www.radians-automation.com/projects>
- **Contact:** <https://www.radians-automation.com/contact>

### Test Admin Panel

1. Go to: <https://www.radians-automation.com/admin/login>
  2. Login with:
    - **Email:** info@radians-automation.com
    - **Password:** RadSuperAdmin2024!Info#Secure
  3. Test all admin features:
    - Dashboard
    - Projects management
    - Inquiries
    - About Us editor
    - User management
- 

## Updates and Maintenance

### Deploying Updates

1. SSH into server:

```
ssh your_username@your_server_ip
cd public_html/nextjs_space
```

2. Pull latest changes:

```
git pull origin main
```

3. Install new dependencies (if any):

```
yarn install
```

4. Regenerate Prisma Client (if schema changed):

```
yarn prisma generate
```

5. Rebuild application:

```
yarn build
```

6. Restart application:

- Via hPanel: Node.js → Restart
- Or via SSH: `pm2 restart all` (if using PM2)

## Viewing Logs

In Hostinger hPanel: 1. Go to **Node.js** section 2. Click on your application 3. View **Application Logs**

Or via SSH:

```
tail -f logs/nodejs.log
```

---

## 🔧 Troubleshooting

### Issue: Application won't start

**Solution:** 1. Check Node.js version: `node --version` (should be 18+) 2. Check logs in Hostinger control panel 3. Verify all environment variables are set 4. Ensure PORT matches Hostinger assignment

### Issue: Database connection timeouts

**Solution:** With traditional server hosting (not serverless), connection pooling should work properly. If issues persist: 1. Check `DATABASE_URL` has connection pooling parameters 2. Verify database is accessible from Hostinger IP 3. Check Prisma client is initialized properly

### Issue: Static assets not loading

**Solution:** 1. Verify build completed successfully 2. Check `.next/static` directory exists 3. Ensure `.htaccess` is configured correctly 4. Clear browser cache

### Issue: API routes returning 404

**Solution:** 1. Verify application is running on correct port 2. Check .htaccess proxy configuration 3. Restart the Node.js application 4. Check application logs for errors

---

## 🔧 Support

### Admin Credentials

**Developer Account:** - Email: denniscortezdiamante@gmail.com - Password: RadDeveloper2024!Dennis#Secure - Role: DEVELOPER (full access)

**Super Admin Account:** - Email: info@radians-automation.com - Password: RadSuperAdmin2024!Info#Secure - Role: SUPER\_ADMIN (full access)

**Test Admin Account:** - Email: john@doe.com - Password: SecureTest2024!RadiansAuto - Role: ADMIN (standard admin access)

### Technical Details

- **Framework:** Next.js 14.2.28
  - **Database:** PostgreSQL (Abacus.AI hosted)
  - **Authentication:** NextAuth with JWT
  - **File Storage:** AWS S3 (Abacus.AI managed)
  - **Deployment Mode:** Standalone (traditional Node.js server)
- 

## ✅ Checklist

Before going live:

- ☐ Node.js application created in Hostinger
  - ☐ All environment variables configured
  - ☐ Code deployed via Git
  - ☐ Dependencies installed (yarn install)
  - ☐ Prisma client generated
  - ☐ Application built (yarn build)
  - ☐ Database seeded
  - ☐ Domain configured and pointing to application
  - ☐ SSL certificate active (Hostinger provides free SSL)
  - ☐ All public pages tested
  - ☐ Admin panel tested
  - ☐ Login/authentication working
  - ☐ File uploads working (projects, images)
  - ☐ Contact form submitting successfully
- 

## 🌟 Benefits of Hostinger over Vercel

1. **No Connection Pool Issues:** Traditional server = single

persistent connection

2. **Lower Cost:** Fixed monthly price vs. usage-based billing
3. **Simpler Configuration:** No serverless complexity
4. **Better for Small/Medium Traffic:** Predictable performance
5. **Full Server Control:** SSH access, custom configurations
6. **Included Database:** Some plans include PostgreSQL

---

**Deployment completed successfully! Your Next.js application is now running on Hostinger shared hosting.** 🚀