

Day 7 - Live Deployment and Post-Launch Practices

1. Introduction to Deployment

Deployment is the process of making your Next.js application available to users on the internet. A well-planned deployment ensures reliability, scalability, and security.

2. Choosing a Hosting Platform

Popular hosting platforms for Next.js include:

- **Vercel** (Recommended for Next.js due to seamless integration)
- **Netlify**
- **Firebase Hosting**
- **AWS, Digital Ocean, or VPS** (for advanced users)

3. Preparing for Deployment

Before deploying, ensure:

- Your code is **optimized** and **minified**
- Environment variables are properly configured
- Unused dependencies are removed
- Code is free from errors and warnings

4. Deploying to Vercel

1. Install Vercel CLI:

```
npm install -g vercel
```

2. Login to Vercel:

```
vercel login
```

3. Deploy the Project:

vercel

4. **Set Environment Variables** via Vercel dashboard

5. **Check Deployment Status** on Vercel dashboard

5. Continuous Deployment (CD) Setup

- Connect GitHub repository with Vercel for automatic deployments on every push.
- Use **GitHub Actions** for custom workflows if needed.

6. Post-Launch Practices

6.1 Monitoring & Analytics

- Use **Vercel Analytics** for performance tracking.
- Enable **Google Analytics** for user behavior insights.
- Monitor application logs using **LogRocket** or **Sentry**.

6.2 Security Best Practices

- Keep dependencies updated.
- Secure API keys and sensitive data using **environment variables**.
- Use HTTPS and enforce security headers.

6.3 Performance Optimization

- Optimize images using **Next.js Image component**.
- Implement lazy loading for large components.
- Use **Server-Side Rendering (SSR)** or **Static Site Generation (SSG)** where applicable.

6.4 Handling Downtime & Issues

- Set up **error tracking** and **alert systems**.
- Implement **automated backups** to restore data quickly.
- Maintain documentation for quick troubleshooting.

7. Conclusion

A successful deployment is just the beginning. Post-launch maintenance ensures smooth user experience, security, and continuous improvement. Keep monitoring, optimizing, and iterating to make your Next.js application robust and scalable!