Host a Website on Render.com Using a Cloned GitHub Repository

This eBook will guide you step by step on how to host a website built with Node.js on the Render.com platform. We'll clone a ready-to-use GitHub repository and deploy the project to the cloud. Repository used:
https://github.com/josuefsy21/website.git

Prerequisites

1

GitHub Account

You will need a GitHub account to access the repository.

https://github.com

2

Render Account

Create a free account on the hosting platform.

https://render.com

3

Node.js Installed

Required to run the project locally.

https://nodejs.org

4

Git Installed

To clone the repository from GitHub.

https://git-scm.com

Step 1: Cloning the Repository

Create a folder, open the terminal inside it, and run the following command to clone the repository:

```
git clone https://github.com/josuefsy21/website.git
```

Navigate into the cloned folder:

```
cd site
```

Step 2: Deploying on Render

Go to Render.com

Visit **https://render.com** and click **New + > Web Service**

Connect your GitHub account

Authorize Render to access your repositories

Select the repository

Choose the cloned repository or your own repository

Configure the service:

- **Environment:** Node
- **Build Command:** `npm install`
- **Start Command:** `node app.js`
- **Branch:** main

Create the Web Service

Click "**Create Web Service**"

Render will automatically install dependencies and start your project.

Step 3: Adding Environment Variables on Render

To keep sensitive information secure (such as passwords and API keys), you should use environment variables.

1. On your service dashboard in Render, click **Environment** > **Environment Variables**.
2. Add the following variables according to your project:
3. After adding all, click **Save Changes**.

Render will automatically restart the service with the new variables.

URL Base

BASE_URL: **www.seusite.com**

Gmail SMTP Variables

- **EMAIL_USER:** **youremail@gmail.com**
- **EMAIL_PASS:** your app-specific password

> **Important:** You must enable two-factor authentication on your Gmail account and create an **app-specific password** for this. Do not use your regular password.

Stripe Variables

- **STRIPE_SECRET_KEY:** your Stripe secret key
- **STRIPE_WEBHOOK_SECRET:** the webhook secret generated by Stripe

Step 4: Using a Custom Domain URL on Render

1. Go to your service dashboard on Render and click **Settings**.
2. Scroll down to the **Custom Domains** section and click **Add Custom Domain**.
3. Enter your domain (e.g., **www.yoursite.com**).
4. Render will show the DNS records you need to configure at your domain provider:Exemplo de configuração DNS:

> - **Type**: CNAME
> - **Name**: www
> - **Value**: your-subdomain.onrender.com

5. After configuring, wait for DNS propagation (it may take up to 24 hours).
6. Render will automatically detect it and enable free HTTPS with SSL.

Your website will now be accessible through your custom domain!

Verifying the Deployment

After completing all the previous steps, you can check if your website is working properly:

1. Go to the Render.com dashboard
2. Select your Web Service
3. Check the logs to make sure there are no errors
4. Click the link provided by Render to access your website

Render provides detailed logs that can help identify issues if your website isn’t working as expected. Check them regularly during the deployment process.

Conclusion

Site Hosted

Done! Your Node.js website is now hosted in the cloud with Render.com.

Automatic Updates

Any changes made to your repository will be automatically reflected on the website after a new deploy.

Next Steps

Consider setting up CI/CD to fully automate the deployment process.

You now have a fully functional Node.js website hosted in the cloud, with environment variables configured to keep your sensitive information secure and a custom domain for a professional appearance.