

How to Deploy MERN Full-Stack to Render



Mai Vang · [Follow](#)

4 min read · Feb 20, 2023



45



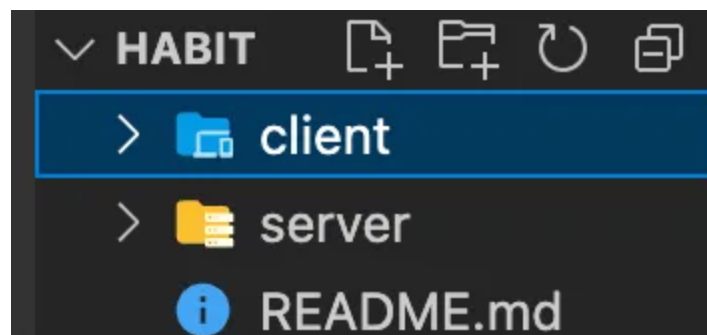
5



I recently learned how to deploy my MERN Full-stack project and can't wait to help you deploy yours too.

When deploying MERN full-stack project, the backend and frontend needs to be deployed separately. Let's walk through how to deploy the backend first:

Here's the setup of the repo I'm deploying to Render:



My folder setup

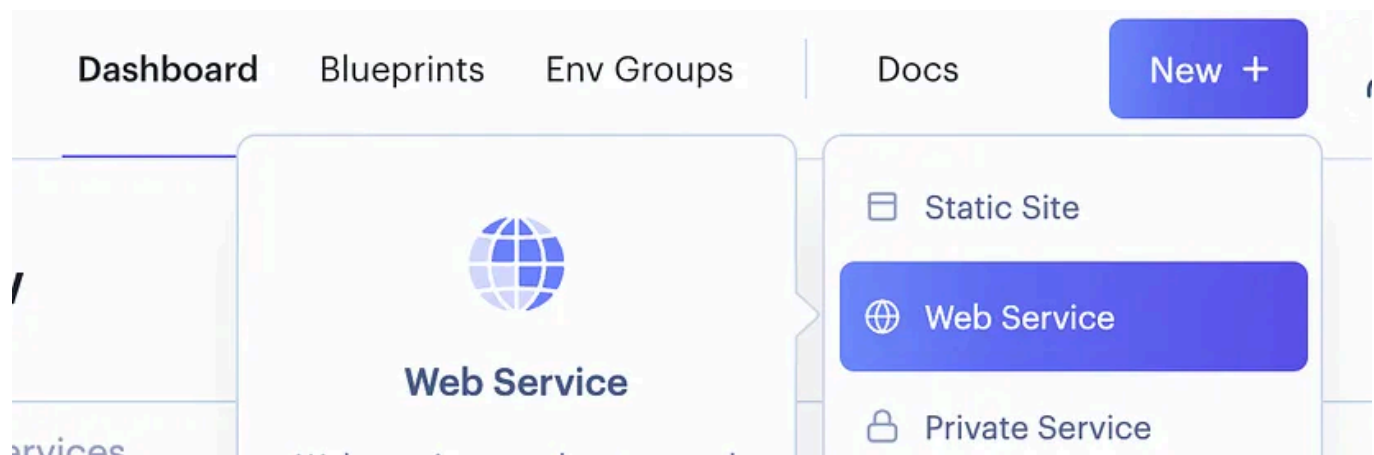
I have client folder for the frontend and server folder for the backend.

First and foremost: Sign up to Render for a free account. I recommend signing up through your GitHub account so it can access your folders faster.

Backend:

For backend:

1. Click on new and select Web Service



Click on 'New' and click 'Web Service' from dropdown

2. Select your repo you want to deploy. Then the next steps will occur with the page below.

Name A unique name for your web service.	<input type="text" value="example-service-name"/> Required
Region The region where your web service runs. Services must be in the same region to communicate privately and you currently have services running in Oregon .	<input type="text" value="Oregon (US West)"/>
Branch The repository branch used for your web service.	<input type="text" value="main"/>
Root Directory <small>Optional</small> Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.	<input type="text" value="e.g. src"/>
Runtime The runtime for your web service.	<input type="text" value="Node"/>
Build Command This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.	<input type="text" value="\$ yarn"/>
Start Command This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.	<input type="text" value="\$ yarn start"/> Required

3. Identify a unique name for your web service.

4. We can leave 'Region' alone

5. Leave 'Branch' as 'main'.

6. For 'Root Directory', I specified 'server' since all of my backend information is located in this folder.

7. Leave Runtime to Node since that is what we used for our backend.

8. Build command, you can leave it as 'yarn'.

9. The Start command, you can put 'npm start'.

10. Then, make sure to chose free for the Instance Type.

Instance Type	RAM	CPU	Price
<input checked="" type="radio"/> Free	512 MB	0.1 CPU	\$0 / month
<input type="radio"/> Starter	512 MB	0.5 CPU	\$7 / month
<input type="radio"/> Standard	2 GB	1 CPU	\$25 / month
<input type="radio"/> Pro	4 GB	2 CPU	\$85 / month
<input type="radio"/> Pro Plus	8 GB	4 CPU	\$175 / month
<input type="radio"/> Pro Max	16 GB	4 CPU	\$225 / month
<input type="radio"/> Pro Ultra	32 GB	8 CPU	\$450 / month

Always choose free for the instance type


11. Afterwards, click the 'advanced' button:



After clicking on the button, you will get more choices populated at the bottom of the page.

12. Click on 'Add Environment Variable':

Use environment variables to store API keys and other configuration values and secrets. You can access them in your code like regular environment variables, for example with `os.getenv()` in Python or `process.env` in Node.

Generate

Add Environment Variable

You can store secret files (like `.env` or `.npmrc` files and private keys) in Render. These files can be accessed during builds and in your code just like regular files.

All secret files you create are available to read at the root of your repo (or Docker context). They are also available to load by absolute path at `/etc/secrets/<filename>`.

Add Secret File

Click on Add Environment Variable

Add an environment variable called 'MY_MONGO_URL' and paste in your MongoDB connection URL to the value.

This step is telling Render how to connect to your MongoDB database.

13. Then click on 'Create Web Service'.

Create Web Service

Then a console log should populate telling you it is starting to clone from your GitHub file.

After a couple of runs, the log should state it is “connected to db && listening on port ###”.

Testing if Backend works:

The way to test if you know if your backend is deployed is click on the hyperlink it provided in the top upper right. It will show an error similar to this:

```
Cannot GET /
```

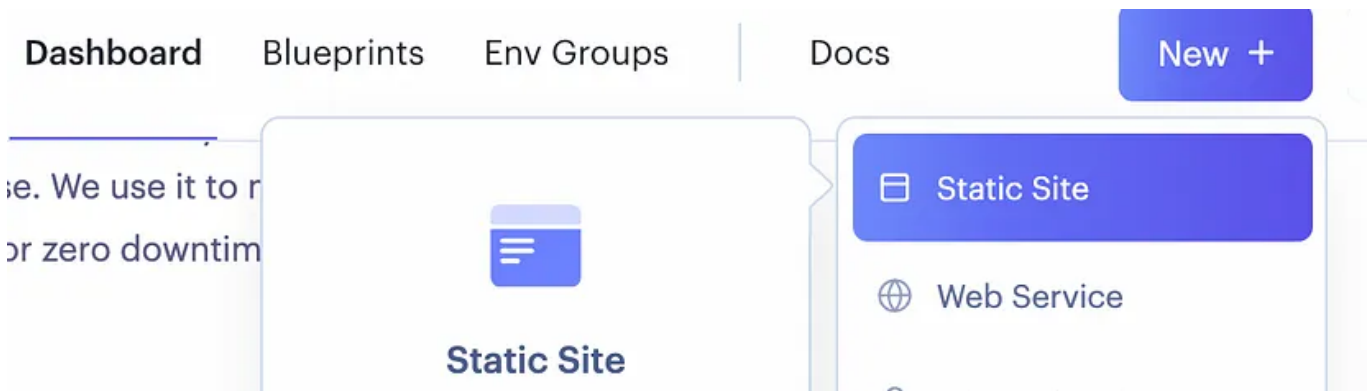
If you go to the hyperlink at the end, add in ‘api/(the end point for your route)’ to the query string, then you should see your database information in a JSON format.

Et voila, our backend has been deployed. Now to finish everything, let's deploy our frontend.

Frontend:

For frontend:

1. Click on ‘New’ and select ‘Static Site’ from the dropdown.



Click on 'New' and select 'Static Site' from the dropdown.

2. Select the GitHub repo you want to deploy. Then you will be directed to the next page and it should look like the following below.

Name
A unique name for your static site.

Required

Branch
The repository branch used for your static site.

Root Directory Optional
Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

Build Command
This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

Publish directory
The relative path of the directory containing built assets to publish. Examples: ./, ./build, dist and frontend/build.

Required

Advanced

Create Static Site

4. Identify a unique name for it

5. Leave Branch as 'main'.
6. For 'Root Directory', type out 'client'.
7. Build command — yarn build
8. For 'Publish directory', type in 'build'.
9. Click 'Create Static Site' and a new page should appear with the console log. On the left side, look for 'Redirects/Rewrites'.

Events

Environment

Redirects/Rewrites

Headers

PRs

Metrics

Settings

10. Afterwards, create two rule.

Redirect and Rewrite Rules

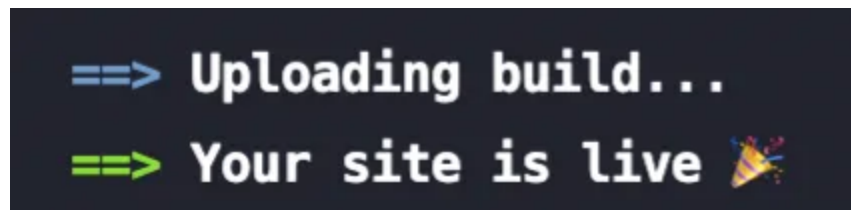
Add Redirect or Rewrite Rules to modify requests to your site without writing code. You can use URL parameters to capture path segments, and wildcards to redirect everything under a given path.

	Source	Destination	Action	
 	<input type="text" value="/api/*"/>	<input type="text" value="/foo/:bar"/>	<div>Rewrite </div>	
		Required		
 	<input type="text" value="/*"/>	<input type="text" value="/index.html"/>	<div>Rewrite </div>	

The first rule that goes on top is to rewrite the `/api/*` in your code to your new deployed backend.

The second rule is to rewrite `/*` to the `index.html` file.

Afterwards, it should try to deploy again and the message should state “uploading build...” and “your site is live”.



What other ways can you deploy full-stack MERN app?

. . .

Please feel free to contact me on LinkedIn:

<https://www.linkedin.com/in/mai-vang-software-engineer/>

[Render](#)[Deployment](#)[Mern Stack](#)[Software Engineering](#)[Job Hunting](#)

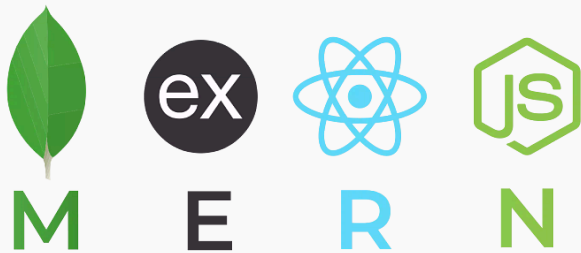
Written by Mai Vang

41 Followers · 14 Following

[Follow](#)

Software engineer Check out more here: <https://www.linkedin.com/in/mai-vang-swe/>

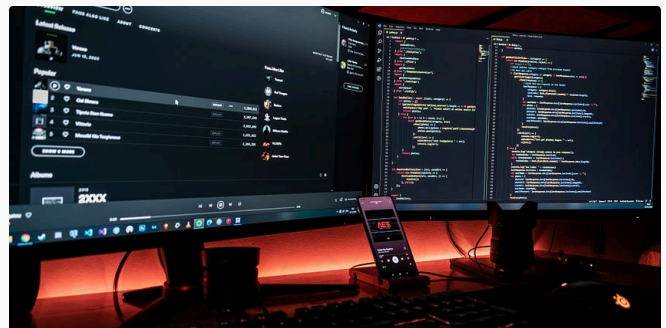
Recommended from Medium



Muhammad Sulaiman Anwar

MERN Stack

Four technologies make up the well-known web development framework known as the...



In Code Smarter by Joana Parente

How to Deploy a Full-Stack MERN App for Free on Render

Create and Deploy a React App with a Node Backend

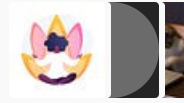


Lists



General Coding Knowledge

20 stories · 1800 saves



Stories to Help You Grow as a Software Developer

19 stories · 1511 saves



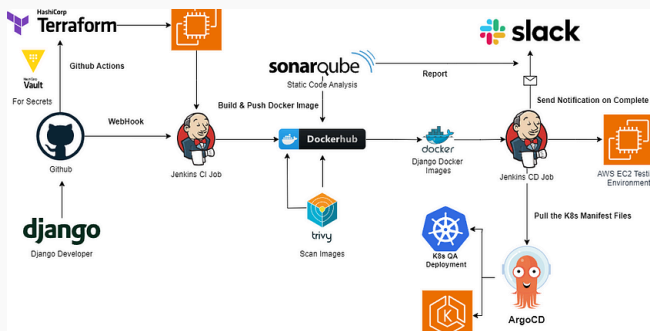
Leadership

62 stories · 498 saves



Good Product Thinking

13 stories · 780 saves

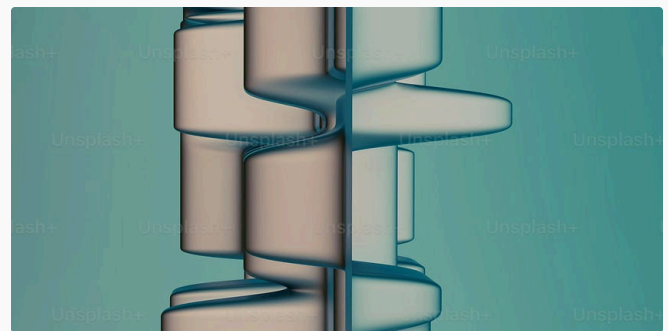


In Django Unleashed by Joel Wembo

Technical Guide: End-to-End CI/CD DevOps with Jenkins, Docker,...

Building an end-to-end CI/CD pipeline for Django applications using Jenkins, Docker,...

★ Apr 12 👏 960 💬 20



Aditya Bhuyan

The Advantages of the MERN Stack for Full-Stack Development

Introduction

★ Oct 24



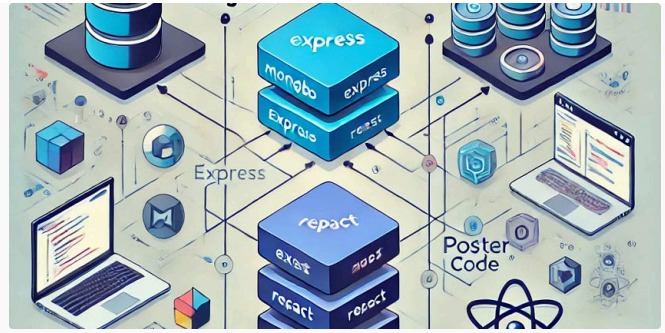


Ravi Patel

Deploying a MERN App to GitHub Actions through Local Runner on...

Deploying a MERN (MongoDB, Express, React, Node.js) stack application and...

Oct 6



Imila Maheshan

Introduction to the MERN Stack: Building a Full-Stack CRUD Project

The MERN stack, a combination of MongoDB, Express, React, and Node.js, has quickly...



Nov 12



14



See more recommendations