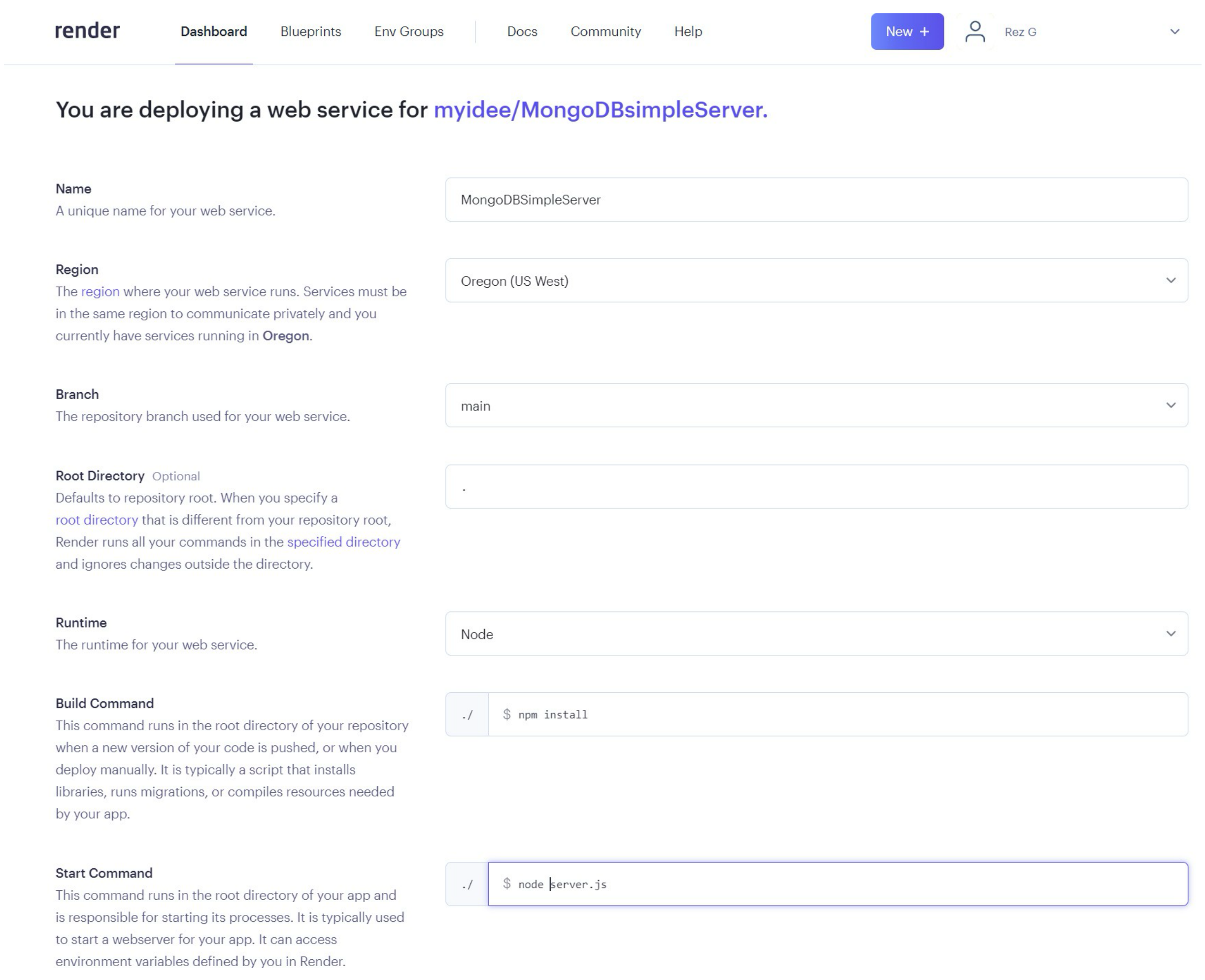
Deploying your Apps for Free

Deploying on Render.com

Push your repo to github.
Then go to render.com and create a new project. Choose webservice, and select the github repo. The following show what to put in the form.



You are deploying a web service for myidee/MongoDBsimpleServer.

Name
A unique name for your web service.
MongoDBSimpleServer

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**.
Oregon (US West)

Branch
The repository branch used for your web service.
main

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.
.

Runtime
The runtime for your web service.
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.

```
./ $ npm install
```

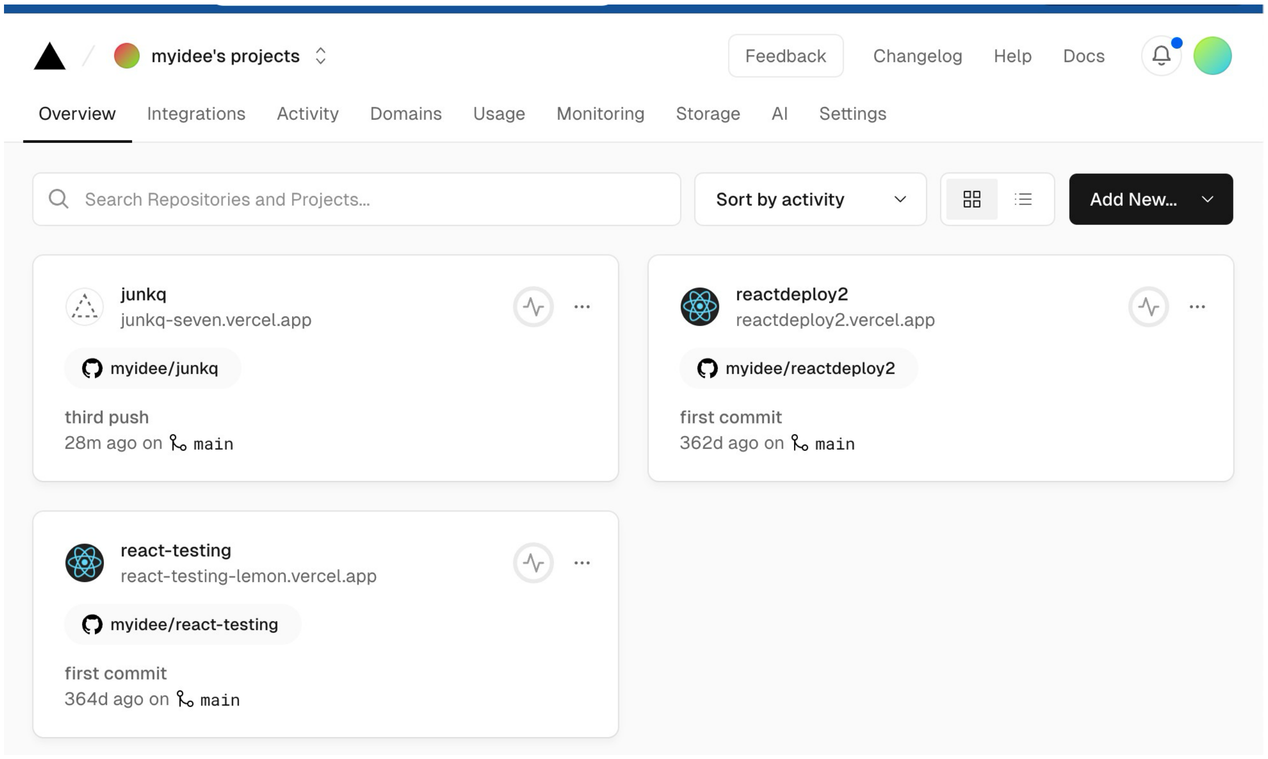
Start Command
This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

```
./ $ node server.js
```

Pay attention to use proper and Build and Start Commands.

Deploying on Vercel

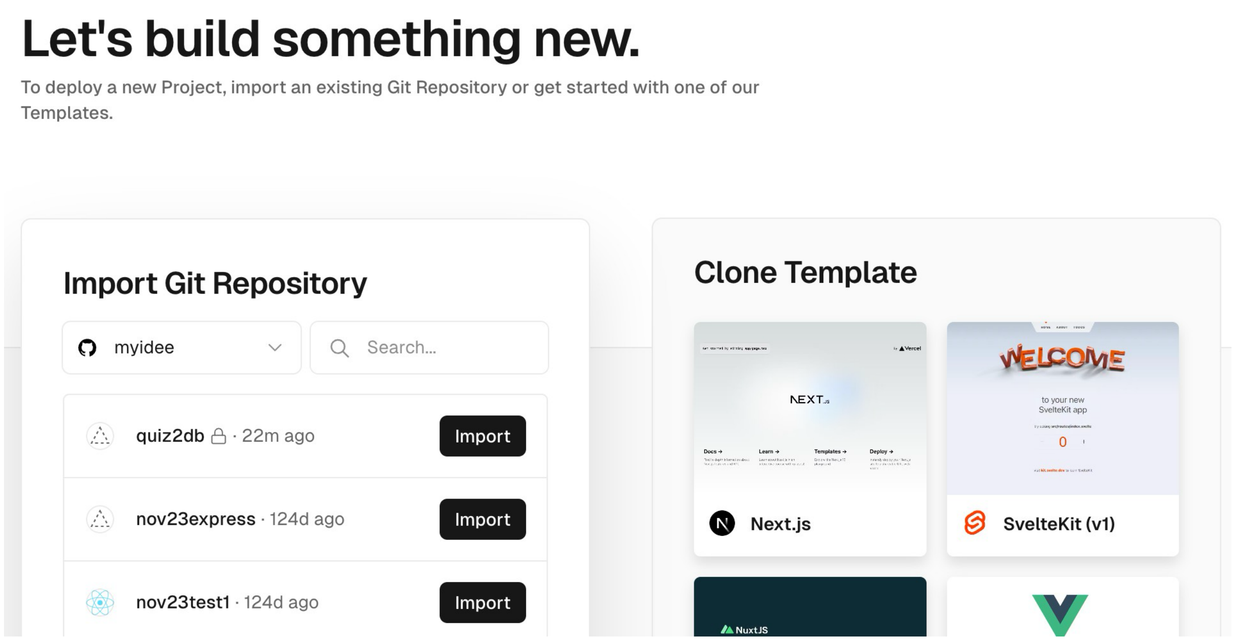
The process otherwise are similar. You can deploy from your github.



junkq
junkq-seven.vercel.app
myidee/junkq
third push
28m ago on main

reactdeploy2
reactdeploy2.vercel.app
myidee/reactdeploy2
first commit
362d ago on main

react-testing
react-testing-lemon.vercel.app
myidee/react-testing
first commit
364d ago on main



Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

Import Git Repository

quiz2db · 22m ago — Import
nov23express · 124d ago — Import
nov23test1 · 124d ago — Import

Clone Template

Vercel offers a free plaVorm and seems to be beWer than Render. Vercel process is similar to Render, but oXen you need to add a vercel.json file to make sure that on page refresh routes are s2ll working. The contents of vercel.json for apps that it has by default is:

```
{
  "routes": [
    {
      "src": "/(.*)",
      "dest": "/"
    }
  ]
}
```

For the plaVorms that Vercel does not automa2cally detect, such as Express apps, the content of **vercel.json** is:

```
{
  "version": 2,
  "builds": [
    {
      "src": "server.js",
      "use": "@vercel/node"
    }
  ],
  "routes": [
    {
      "src": "/(.*)",
      "dest": "server.js"
    }
  ]
}
```

If start file is index then use : **“src” : “index.js”** or **“./index.js”**

Deploying to Netlify:

Deploying to Netlify is also simple. You can also drop your folder instead of using Github repository. However for apps using React Router you need to add:

/* /index.html 200

Into a file named _**redirects** in the **public folder**.

Deploying on Github Pages or production web servers

Before you can deploy your React applica2on, you'll need first to create a produc2on-ready built of your app.
Build your app, by: **npm run build**

This will generate a build folder inside your project containing all the sta2c files that can be used on a produc2on web server. The build command transfers all your React code into code the browser understands, and it op2mizes your files for best performance.

It bundles up all your JavaScript files into one minified file, and minifies resources, like the HTML template and CSS to help reduce download 2mes. It's also going to provide a service worker file to help run your app offline. This will be the folder that you deploy to a server.

Now, to view the production build of your app on your machine, you'll s2ll need to run it on a server.

A quick way to do that, as suggested in the Create React App docs to use a tool called serve. You can install it by:
npx serve -s build

Now, you need to install it on Github Pages.

For github, you need to add this to **package.json**
"homepage" : "h;p://<username>.github.io/<repository name>"

Then install github pages package:
npm i gh-pages

then
in package.json add the following scripts:

"scripts": {
"predeploy": "npm run build", "deploy": "gh-pages -d build",
…
}
If your app uses React Router, change <BrowserRouter> to <HashRouter> before pushing and deploying, so the deployment doesn't lose router pages.

It is easier to push the files directly to github and then perform **npm run deploy**
if it the repository is already in github, add the git remote before running **npm run deploy**

git remote add origin https://github.com/<username>/<repositoryname>.git
the new build will be available in the github pages, in gh-pages branch.