Tutorial: Deploying Your Web Application With Render.com

Render.com is a service that allows one to deploy web applications on the internet. Luckily, it has a free tier so we can use it for our term project!

Before we get started, please carefully read the following warnings:

- *As usual, this service may store your personal information on servers outside the country where they may be accessible to authorities in those countries. Please talk to me ASAP if this is a concern for you.*
- *You are about to publish content to the World Wide Web where it will be accessible to everyone with an internet connection, possibly forever. Be very careful about what you publish; do not publish any information that is personal or sensitive.*
- *Your live web application may be targeted by malicious actors. We haven't focused heavily on security in this course, so I'm not able to guarantee your site will be safe from hacking, although Render.com does provide some basic security protection: proceed at your own risk. At the very least, you should apply: 1) basic data validation through both Express middleware and MongoDB collection schemas, and 2) strong passwords for your database and any other service that requires credentials.*
- *When your application is live on the internet, anyone can submit data to your database, including offensive and harmful material. We cannot stop this, so proceed at your own risk. If you have concerns, please talk to me.*
- *Render.com requires access to your GitHub account and database credentials, which generally should only be granted to organizations you trust. Render.com has a good reputation and I personally trust them, but please talk to me ASAP if you have concerns.*
- *If you provide your credit card number to Render.com, there is no limit to how much they could charge you for usage overages, as they don't have a spending cap. I strongly recommend that you DO NOT provide your credit card number; if you do, you accept full responsibility for charges. If they request a credit card for account verification, please contact me ASAP.*

Okay, let's get started!

Task 1: Preparing Your Application for Deployment

This tutorial assumes you're using the folder structure we used in our lessons and assignments with a **server** folder for the Express server and a **client** folder for the React source files and Parcel application. If you're using a different folder configuration, you'll have to adapt these instructions.

This tutorial also assumes you're working in the GitHub Codespaces environment. If you're working on your local machine, you'll have to manually set up Git and push your code to a Github repository; ask me if you need help with this.

There are a few basic things to do before we deploy our application using Render.com:

- In the root directory of your project, add a **.gitignore** file with the following content:

  ```
  server/node_modules
  server/.env
  client/node_modules
  client/.parcel-cache
  ```

  As you can see, we're excluding our database credentials, node_modules, and Parcel cache files from our GitHub repository.
- In **server/package.json,** add the following scripts:

  ```
    "dev": "node --watch --env-file=.env app.js",
    "build": "npm install",
    "start": "node app.js"
  ```

  These scripts do the following:
  - **"dev"** is the script you use to start your Express server when working in the Codespaces environment, while testing and developing it, NOT when deploying it for public use. You can run this script in the terminal using the command:

    ```
    npm run dev
    ```

  - **"build"** is the script you use to build the application if **node_modules** is not present. It relies on the **package.json** file to install the correct dependencies. You can run this script with the command:

    ```
    npm run build
    ```

  - **"start"** is the script used to run the Express application in a production setting when the project is being deployed to the web. This is a special script name that can be run without using **"run"** in the command, like this:

    npm start

- In **client/package.json,** add the following scripts:

  "dev": "npx parcel watch src/index.js src/style.css --dist-dir ../server/public",
  "start": "npx parcel build src/index.js src/style.css --dist-dir ../server/public"

  As you can see, the **start** script is exactly the same as the **dev** script, except it uses the **build** command instead of the **watch** command. From now on, you will use the **dev** script when developing the application, and **start** script when ready to deploy to production. The **build** command within the **start** script simply compiles the code one time (instead of every time you make a change), compresses the code to a smaller format for easier transmission, and disables the HMR server.

  Run the **start** script to prepare the React application for production deployment:

  npm start

- Login to MongoDB Atlas and make sure you're using a strong password for your database. You can change the password by navigating to **Database Access** and clicking **EDIT** beside the user details. Remember to click **Update User** after changing the password.
- As mentioned earlier, make sure you have at least two levels of validation for incoming data, as anyone with an internet connection will be able to submit data to your application.
- Delete any personal or sensitive information from your database and anywhere else in your application. Your app is about to be publicly accessible to everyone, maybe forever!
- If you haven't already, in your Codespace, navigate to the **Source Control** area and **Publish** your project to a new repo, just as you have with previous assignments. If you've already published to a repo, **Commit** your changes and then **Sync Changes** to make sure your repo is up to date.

Task 2: Sign Up for a Render.com Account

Navigate to https://render.com/ and then to **Sign In > Sign up.** This should go without saying, but use a strong password for your new account.

You'll be asked a few questions about how you intend to use the service. I don't think your answers matter much, but choose answers that align with what we're doing in this class.

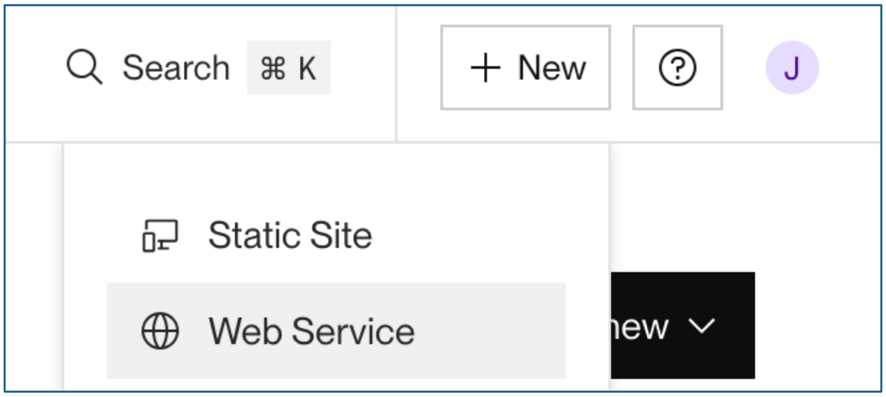
When presented with a choice of plans, choose the **Hobby Plan.** DO NOT provide a credit card number; you shouldn't need one, as this plan is free. I've heard of users in some regions being asked for a CC number for account verification, but if this happens to you, please let me know before you proceed.

After you've set up and verified your account, open a new browser tab and login to your **GitHub** account; this will make the next steps a bit easier.

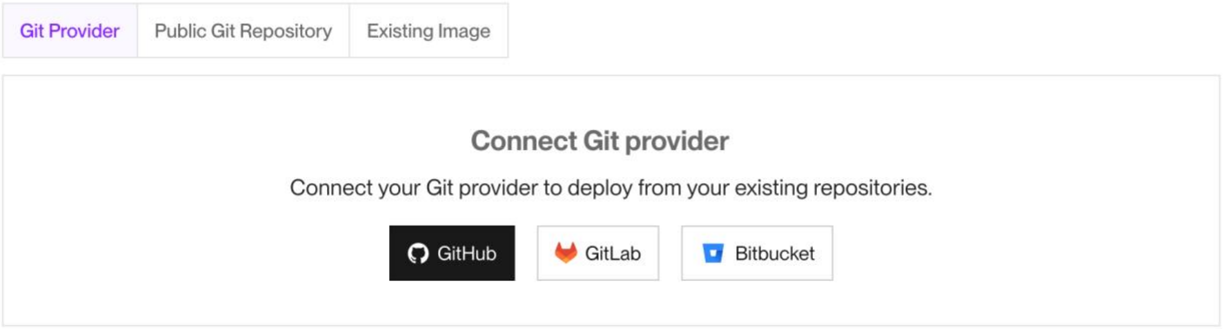
Task 3: Connect Your Render.com Account to Your GitHub

You should now be signed in to both your Render.com account AND GitHub account.

In the Render.com dashboard, click **+ New** and then **web service.**



You'll be asked to choose a Git provider; choose **GitHub:**



Connect Git provider

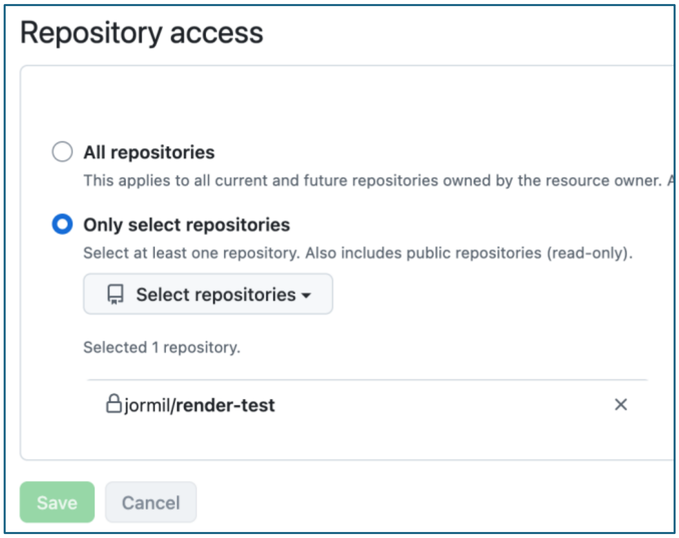
Connect your Git provider to deploy from your existing repositories.

A window will appear asking you to grant permission to Render to access your GitHub account. Read the terms carefully to make sure you understand and agree, and then click **Authorize Render.** You'll receive an email from GitHub confirming the authorization.

You will be presented with a link to a setup page in GitHub; click this link. (If you don't see the link, in GitHub navigate to the circular profile thumbnail in the top-right corner, then **Settings > Applications;** you should see **Render** in the list of applications, and you can click **Configure.**)

You may be asked to provide your password for GitHub again; go ahead and do that.

Find the **Repository Access** area, choose **Only select repositories,** and choose the repository of your project, and click **Save**.



Repository access

○ **All repositories**

This applies to all current and future repositories owned by the resource owner. A

● **Only select repositories**

Select at least one repository. Also includes public repositories (read-only).

Selected 1 repository.

jormil/**render-test** ×

Task 4: Deploy Your Application

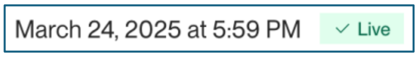
You will be directed to the **"You are deploying a web service"** area of **Render.com** where you will configure your deployment. Make sure the **Source Code** field is set to the name of your project GitHub repository.

Here's how the other fields should be set:

- **Name:** you can either use the default name of your repo, or choose a new one.
- **Project:** ignore this.
- **Language:** Node
- **Branch:** probably you'll use **main** for this, but if you created a different branch that you'd like to use instead, set it here.
- **Region:** whatever the default is.
- **Root Directory**: set this to **server,** as that's the directory where rour Express application lives. Note that the Parcel application in the **client** directory is used only for development purposes and is entirely ignored by Render.com!
- **Build Command:** npm run build
- **Start Command:** npm start
  (those scripts we created earlier are coming in handy!)
- **Instance Type:** Free
- **Environment Variables**: instead of using a **.env** file for our production build, we'll instead use Render's built-in environment variable dashboard. Copy the **name=value** pairs from your .env file and paste them into this form. For many students, only a single environment variable will be needed for the database connection string. Check carefully to make sure you've copied the right values.

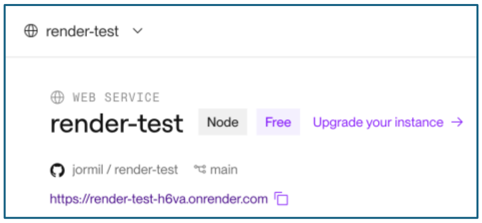
Then click **Deploy Web Service.**

You'll be directed to a page with a terminal showing the build/deploy status. You may see a progress indicator that says **"Building"** and then **"In Progress".** If any errors occur, you'll see a message appear in the terminal, and you should be able to use it to trouble-shoot and fix the problem. If all goes well and the deployment is successful, the progress indicator will say **Live.**



March 24, 2025 at 5:59 PM ✓ Live

You can now visit your live application! Find the URL above the terminal:



render-test

WEB SERVICE

render-test Node Free Upgrade your instance →

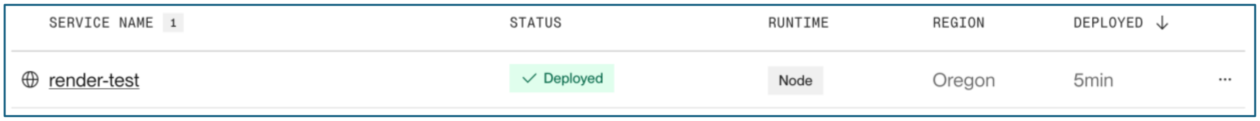
jormil / render-test main

https://render-test-h6va.onrender.com

Task 5: Continuing Development

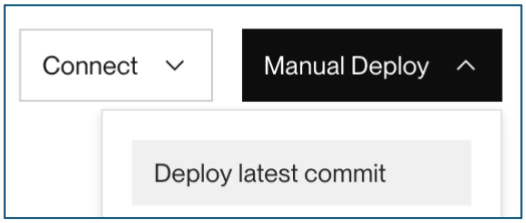
Just because you've deployed your application, it doesn't mean you have to stop developing. If you make changes to your code after deployment, first make sure you update your GitHub repo: in the Codespace, navigate to the **Source Control** area, enter a commit message to describe the changes you made, and click **Commit.** Then click **Sync Changes** to push your commit to the repo.

To deploy the latest version of your application from the Render.com dashboard, find your application at the bottom of the dashboard and click its name:



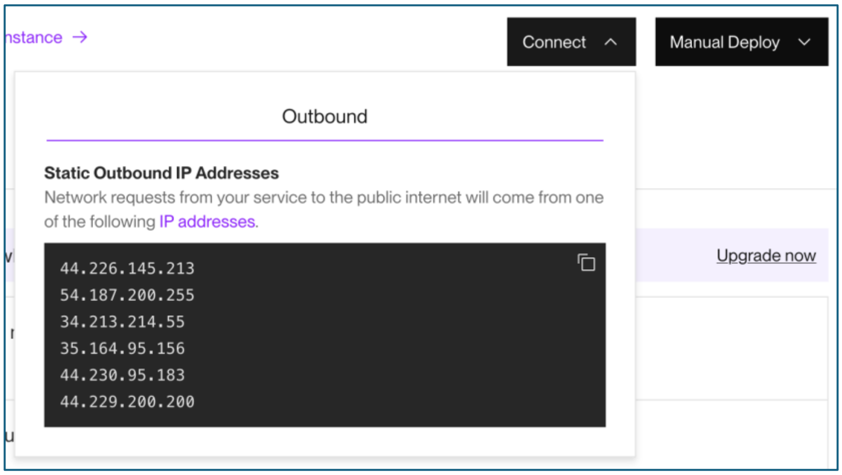
SERVICE NAME 1	STATUS	RUNTIME	REGION	DEPLOYED ↓	
render-test	✓ Deployed	Node	Oregon	5min	···

Then find the **Manual Deploy** button and click **Deploy latest commit:**



For an extra level of security, you can configure your database to only accept connections from your Render.com application. (Remember when we configured our databases to accept connections from anywhere?)

Your Render.com application actually has a set of IP address it uses that correspond to your production server. To see them, click the **Connect** button (beside the Manual Deploy button):



Static Outbound IP Addresses

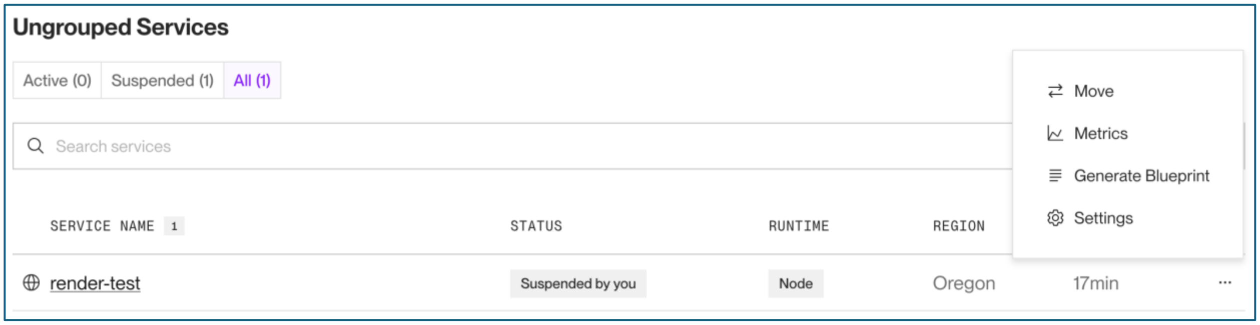
Network requests from your service to the public internet will come from one of the following IP addresses.

```
44.226.145.213
54.187.200.255
34.213.214.55
35.164.95.156
44.230.95.183
44.229.200.200
```

In the MongoDB Atlas dashboard, navigate to **Network Access** and click **+ADD IP ADDRESS.** You'll have to do this once for each IP address in the list.

You can then delete the 0.0.0.0/0 IP address, but beware: your Codespace development server won't work after you do this! Only do this when you're ready to pause development. (Of course you can always reconfigure your database to accept connections from anywhere again if you need to.)

If at any point you'd like to take your application offline (maybe after I've graded it) you can do so by finding your application in the Render.com dashboard, clicking **...** and **Settings:**



SERVICE NAME 1	STATUS	RUNTIME	REGION		
render-test	Suspended by you	Node	Oregon	17min	···

Then scroll to the bottom of the settings page and click **Suspend Web Service.** It's probably a good idea to take the web service offline when you're not using it, as it may unnecessarily provide surface area for a hack.

You can also delete the web service altogether if you want. (You can always redeploy it later if you want.)