Deploy a Node.js app

There are different ways to deploy. If there's just Node (MongoDB is a different story), it's relatively simple. For testing/prototyping Glitch or Google Cloud Platform are two interesting ones you can try.

Note that the free tiers for many serverless cloud options are **very slow to spin up from inactivity so for your capstone and deployed apps in your portfolio, I recommend upgrading to the paid tier** just so they run more quickly (very slow apps or apps that fail to load on first page load doesn't look good for employers).

Render (https://render.com)

This is a serverless option that is relatively simple to set up. There is a free tier.

1. Create an account and log in (or use one of the other login options).
2. Create a new Web Service.
3. Assuming your app is in a Github repo, choose to build and deploy from a GitHub repo and click **Next**.
    - If this is your first time, you'll get a couple prompts asking you to login to GitHub and to ask you to authorize Render to access GitHub. It will also ask if you want to install Render in your GitHub repos. You can choose to install on all or pick and choose which repo will have the Render deploy tool installed.
4. Once you're connected to GitHub through Render, you'll see a list of repos that Render can connect to. Select the one you want to deploy.
5. You'll next need to set up your app.
    - For the **Build Command**, type `npm i`.
    - For the **Start Command**, type `npm run dev` (if that's what you have in your *package.json* file).
6. Under the instance type, choose the free or one of the paid tiers (keeping in mind the free tier is slow to spin up).
7. Under **Environment Variables**, you can add them one at a time, or choose **Add from .env** then copy and paste your environment variables in.
8. Click **Create Web Service**. If your settings were correct, it should deploy successfully. If there's an issue, check the logs.

Google Cloud Platform

- There is a free tier and it automatically upgrades to the paid option if exceeding your free tier limits so Google Cloud does require you to enter credit card info, but apps you just need to deploy but you don't use option would stay within free limits.
- There are two options:
    - Google App Engine: https://cloud.google.com/appengine/docs/standard/nodejs/building-app
        - This is faster because it's always running, so you *may* hit the free tier limit. This would be useful if you wanted something to just be more reliable over a shorter period.

    - Google Cloud Run: https://cloud.google.com/run/docs/quickstarts/build-and-deploy/deploy-nodejs-service
        - This is fast while it is running, but it is serverless so it's slower to start from inactivity. This is fine for prototyping/testing but slow if you wanted to

Netlify (https://netlify.com)

- https://www.netlify.com/blog/2018/09/13/how-to-run-express.js-apps-with-netlify-functions/

Glitch (https://glitch.com/)

- For prototyping or just playing around, there's an option to create a small anonymous app (app expires in a short time).
- You can also sign up for non-anonymous apps though there are limits for the free tier.

Paid options

Traditional hosting (shared)

If using shared hosting already, check to see if your host supports Node apps. If using a Linux-based shared host, you'll probably be using cPanel as the admin interface. If Node is supported, there's probably a one-click installer for you to set up your app. Just follow the instructions for your host. Here's a good read:

https://medium.com/@pampas93/host-your-node-js-app-on-shared-hosting-go-beyond-localhost-73ab923e6691

Cloud hosting (e.g. Digital Ocean)

For cloud servers, Digital Ocean is a good one to try. For Digital Ocean you get a droplet which is a virtual machine but acts like a separate server. This means that you need to set it up but it does give you more control. **Digital Ocean is paid but you can start with the cheaper tier ($5/mo).**

More recently, Digital Ocean also offers a one-click setup to make it easier for you.

- Create and deploy Node app to Digital Ocean (older documentation but may be useful): https://www.digitalocean.com/community/tutorials/deploying-a-node-app-to-digital-ocean
- Create a one-click Node.js app: https://marketplace.digitalocean.com/apps/nodejs
- Managed MongoDB (more expensive): https://www.digitalocean.com/blog/introducing-digitalocean-managed-mongodb/
    - If you just need MongoDB online, you can use MongoDB Atlas instead.