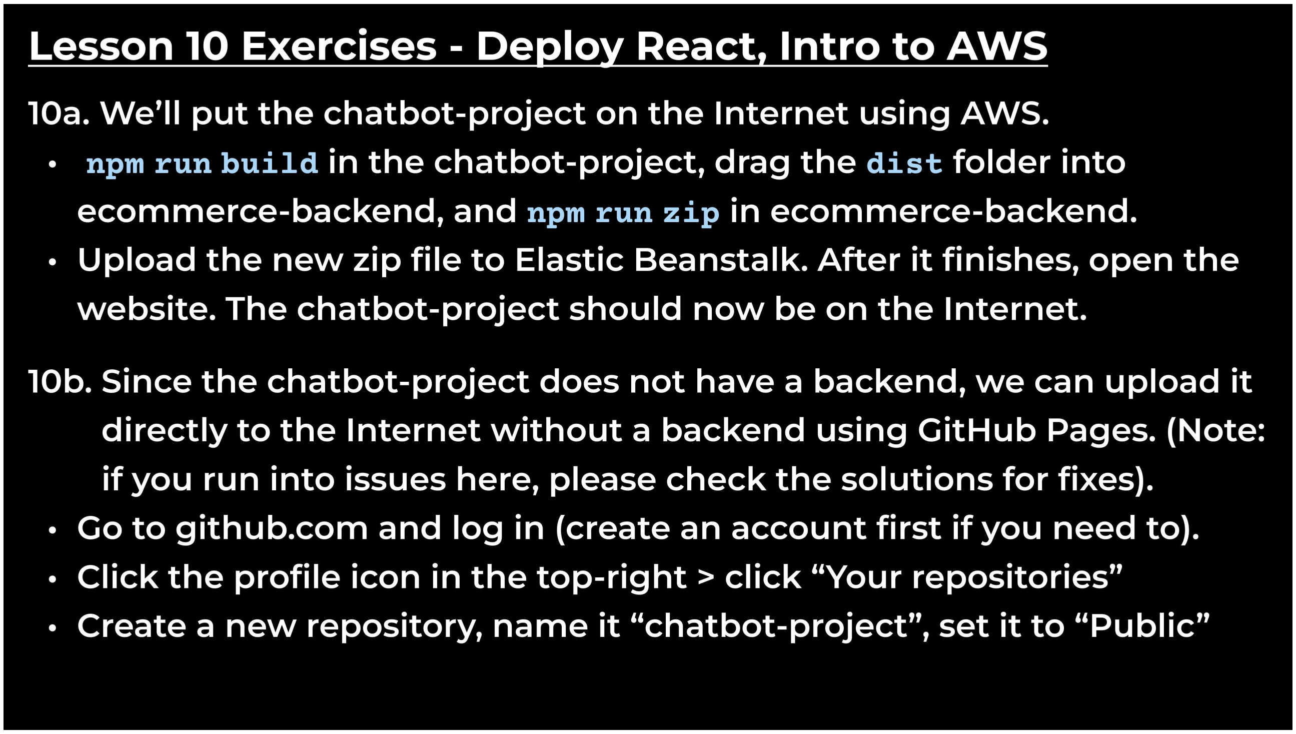


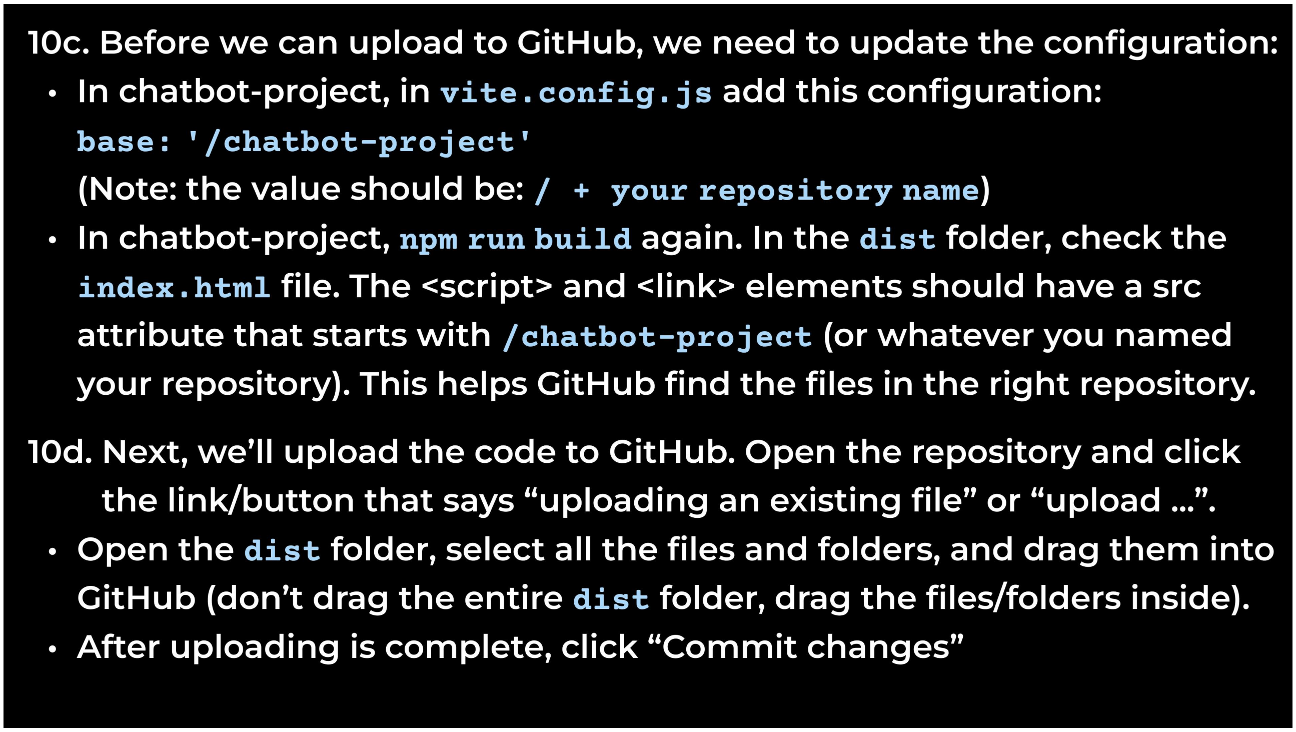
Lesson 10 Exercises - Deploy React, Intro to AWS

10a. We'll put the chatbot-project on the Internet using AWS.

- `npm run build` in the chatbot-project, drag the `dist` folder into ecommerce-backend, and `npm run zip` in ecommerce-backend.
- Upload the new zip file to Elastic Beanstalk. After it finishes, open the website. The chatbot-project should now be on the Internet.

10b. Since the chatbot-project does not have a backend, we can upload it directly to the Internet without a backend using GitHub Pages. (Note: if you run into issues here, please check the solutions for fixes).

- Go to github.com and log in (create an account first if you need to).
- Click the profile icon in the top-right > click “Your repositories”
- Create a new repository, name it “chatbot-project”, set it to “Public”

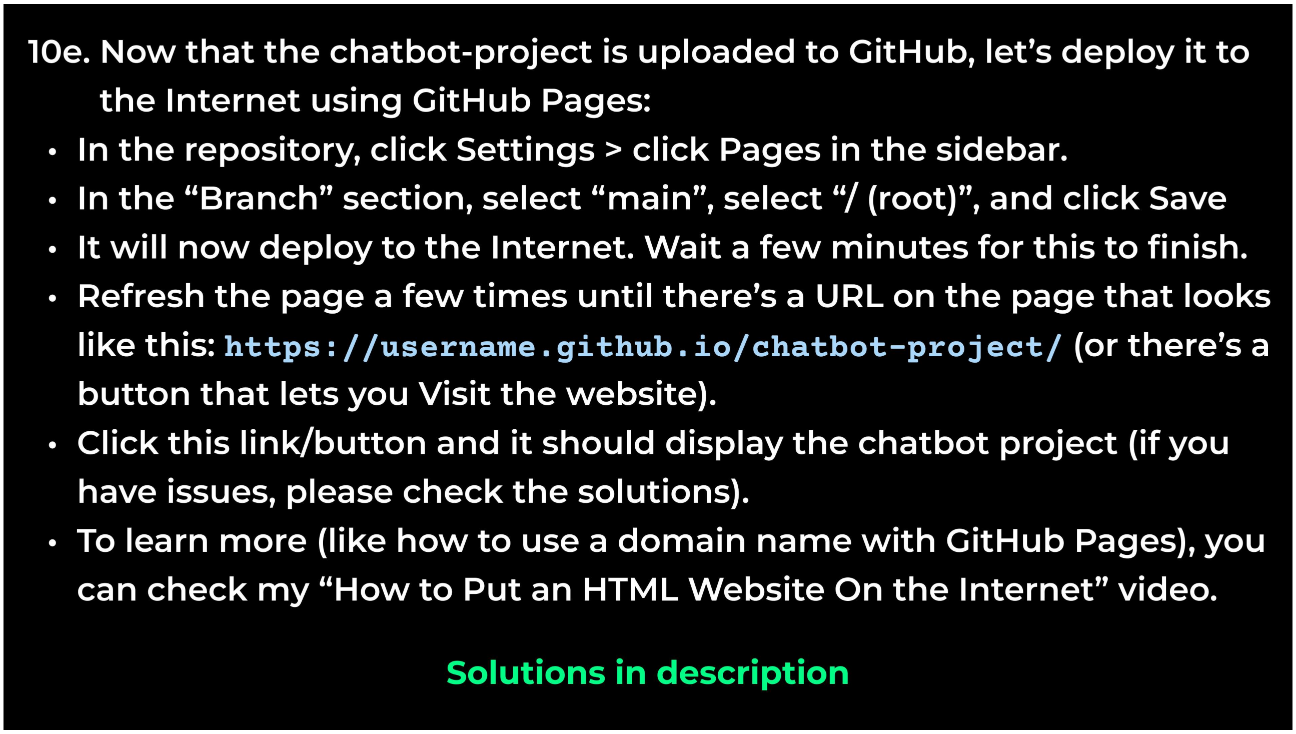


10c. Before we can upload to GitHub, we need to update the configuration:

- In chatbot-project, in `vite.config.js` add this configuration:
  `base: '/chatbot-project'`
  (Note: the value should be: `/ + your repository name`)
- In chatbot-project, `npm run build` again. In the `dist` folder, check the `index.html` file. The <script> and <link> elements should have a src attribute that starts with `/chatbot-project` (or whatever you named your repository). This helps GitHub find the files in the right repository.

10d. Next, we'll upload the code to GitHub. Open the repository and click the link/button that says “uploading an existing file” or “upload ...”.

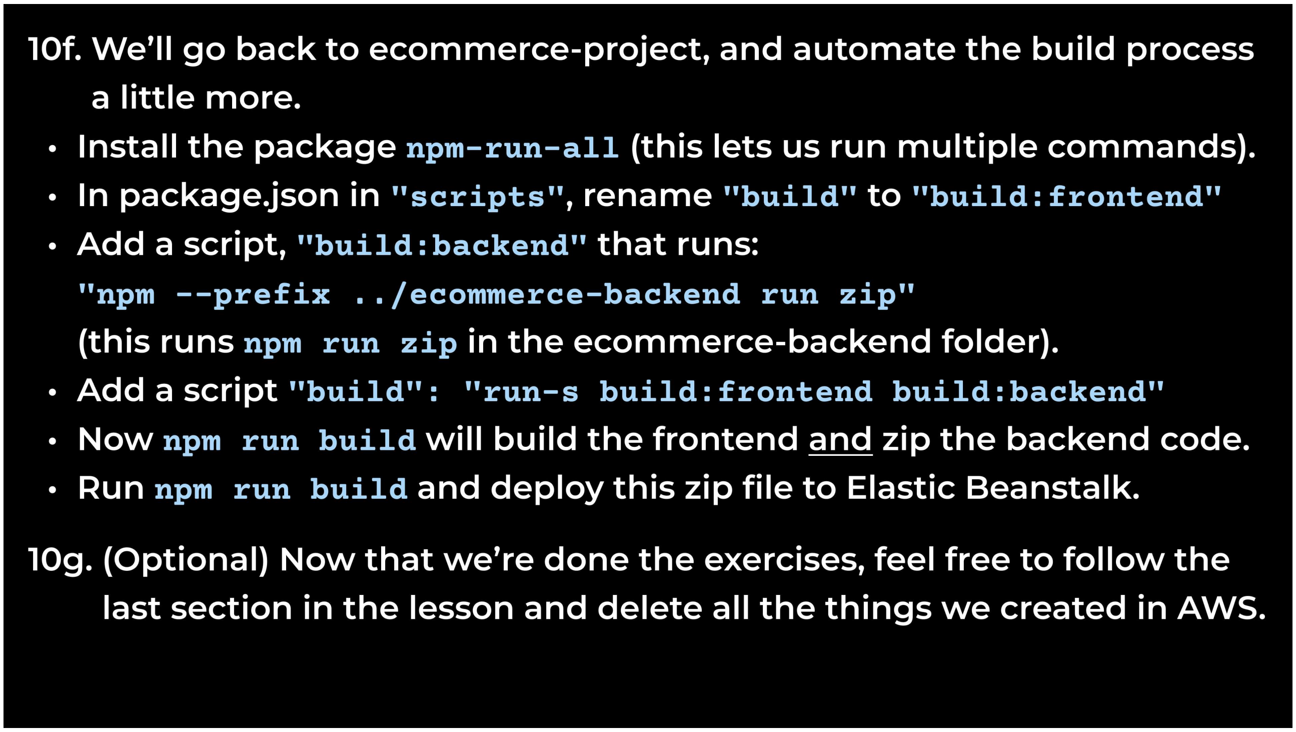
- Open the `dist` folder, select all the files and folders, and drag them into GitHub (don't drag the entire `dist` folder, drag the files/folders inside).
- After uploading is complete, click “Commit changes”



10e. Now that the chatbot-project is uploaded to GitHub, let's deploy it to the Internet using GitHub Pages:

- In the repository, click Settings > click Pages in the sidebar.
- In the "Branch" section, select "main", select "/ (root)", and click Save
- It will now deploy to the Internet. Wait a few minutes for this to finish.
- Refresh the page a few times until there's a URL on the page that looks like this: `https://username.github.io/chatbot-project/` (or there's a button that lets you Visit the website).
- Click this link/button and it should display the chatbot project (if you have issues, please check the solutions).
- To learn more (like how to use a domain name with GitHub Pages), you can check my "How to Put an HTML Website On the Internet" video.

Solutions in description



10f. We'll go back to ecommerce-project, and automate the build process a little more.

- Install the package `npm-run-all` (this lets us run multiple commands).
- In package.json in `"scripts"`, rename `"build"` to `"build:frontend"`
- Add a script, `"build:backend"` that runs:
  `"npm --prefix ../ecommerce-backend run zip"`
  (this runs `npm run zip` in the ecommerce-backend folder).
- Add a script `"build": "run-s build:frontend build:backend"`
- Now `npm run build` will build the frontend and zip the backend code.
- Run `npm run build` and deploy this zip file to Elastic Beanstalk.

10g. (Optional) Now that we're done the exercises, feel free to follow the last section in the lesson and delete all the things we created in AWS.