

Deployment

One of the biggest things you need to do to finalize any project is to actually physically deploy it. We will be going through all of the options in a bit, but in the mean time here are some steps you want to go through to make sure everything is prepared for you. I'll also list some things to look for in a hosting site.

Hosting Options To Look Into

- [Vercel](#) - Quick hosting for personal projects
- [AWS](#) - Allows for extensive customization of database and project structure.
- [Digital Ocean](#) - Similar to AWS it allows for customization of multiple things. Larger learning curve though.
- [Microsoft Azure](#) - Microsoft's version (for lack of a better term) of AWS.

Hosting Prep Work

- Ensure that your packages are up to date in the `package.json` (run `npm update`).
- Ensure that your build and start scripts are what you need them to be.
- Make sure that your Environment variables / config are set to production.
- Ensure that you're using a correctly configured version of Docker if you're deploying via docker
- Make sure any testing is completed.
- Ensure you have tested it on all browsers you want compatibility for.
- Make sure nothing is going to be leaked to the user that isn't supposed to be.
- Remove any console logs you used for testing.
- Make sure all API routes are protected against malicious tampering.
- Ensure that your URL redirects to the `https` version if they try to go to the `http` version.

```
app.enable("trust proxy");
app.use((req, res, next) => {
  req.secure
    ? next()
    : res.redirect("https://" + req.headers.host + req.url);
});
```

- Make sure you've either built your files locally OR you're having them built after pushing to the host
- Ensure all of your configuration is correct

What to look for in a host

- Cost - There are many cost effective options out there but some of them can rack up costs based off use and can lead to unexpected charges.
- Ease of use - AWS/ Azure / Digital Ocean are much more difficult to use BUT they allow more customization and can help you get a better feel for real world deployment.
- Ease of Deployment / Continuous Integration - If this is something that you'll deploy once and be done with then you shouldn't need to worry about this.

- Servers - Some hosting sites (like Vercel) won't let you deploy a server. Instead, you have to make it "serverless".
- Security - This should be a no-brainer but I'm going to mention it anyway. Any of the above are more than secure enough.
- Database needed - Take a look at all of the options as not all of them allow for the same types of databases as easily as others.