

Website Hosting Via Digital Ocean / Azure / Fly.io

Benefits

- You have a lot of options regarding customization and database options.
- Will give you practice for potential real world applications.
- Exposes you to more advanced hosting and more server / web connection information.

Downsides

- Most of these options cost money or are very limiting in number of connections etc.
- Can be a huge learning curve for simple, let alone more complex, deployment
- Can require additional steps / files / configuration just to be able to deploy (discounting setup on the hosting platform's side)
- Require a backend (or pseudo-backend) as well as configuring some form of proxy on the host's side.

Vercel

Vercel is probably the most straight-forward and fastest way to deploy an app. Basically, all you need to do is import the Git repository to Vercel, and every time the repo updates, it will also update on the website automatically. Here's a complete [guide](#).

The only problem with Vercel, is that it specializes in frontend deployment. If you have a backend, you would need to use serverless code. If you want to deploy to Vercel with Docker... well, you can't.

Netlify

Alternative to Vercel. Just connect a Github repo and you're done. Also requires serverless functions and you cannot use Docker.

Fly.io

Fly.io is the free-est way to deploy a frontend, backend, and database using Docker. This is the one we will be using in class because it's free and we are able to use Docker. Here is a [guide](#). Also install [flyctl](#).

Digital Ocean

Digital Ocean is a cloud based hosting service that allows for easy deployment and use for databases and/or applications. While it is pay to use, the price is incredibly reasonable for what you get. The full docs can be seen [here](#).

Deploying via Digital Ocean

Deploying via digital ocean can be very straight forward, but if you're not used to working with external hosting / server tools it can be a bit daunting at first. Rather than go through a step by step guide, there are plenty of resources located [here](#). You can also deploy with Docker.

Microsoft Azure

If you're using `.NET`, there's a solid chance that you'll be deploying to Azure. It's one of the biggest competitors to AWS for that very reason. It too has a large learning curve but overall it isn't too bad compared to AWS. Since VSCode is built by Microsoft, there are some great extensions for VSCode that can help alleviate some of the headache.

Deploying via Azure

Like with AWS there are plenty of tutorials out there on the Azure docs covering how to deploy with multiple different types of backend languages. If you're wanting to take the plunge, I also recommend learning some `.NET` and utilizing that on your backend. In some ways, that makes the deployment a ton easier as you're using Microsoft's backend language/framework to deploy directly to a Microsoft hosting service.

What is AWS and why should we care?

[AWS](#) is Amazon Web Services. It has become one of the largest hosting sites for pages, databases, static content, and many other things in the world. One of the biggest downsides with AWS is honestly one of the best things about it. You can do so much customization that you can get exactly what you want without having to settle for something you don't need or even change things too much on your end. The flip-side of that, is that there are an infinite number of things to learn, and there can be a massive learning curve for just about anything AWS related.

Deploying Via AWS

It's not the easiest / most straight forward. There are a ton of solid resources out there but I'll warn you ahead of time that it can be a massive rabbit hole. Using Lambda functions as well can be beneficial as they can make the deployment process much faster (with some tweaking to the code) as it's technically using a virtual machine as opposed to a full blown server. There are some plugins for VSCode that can help with some of the headache.