<p>So like I said, this site is primarily a place for me to play around with Amazon Web Services, and the site is built exclusively using them. So let me take a few minutes to explain at a high-level how it's all put together.

<p>If we rewind the tape all the back to the beginning of the “Let’s Build A Web Application” story, it starts with being able to actually write some code, and then have that code deployed and hosted somewhere so – in my case – people can find my web site.

<p>For me it starts with using a MacBook Pro for development & Visual Studio Code as my IDE. I run Git locally and have a GitHub account to persist the web application’s Git repository in the cloud. The GitHub repo comes in especially handy when using something like AWS Amplify to host & provision your web application. Amplify lets me create a simple workflow where I write code, save changes, check-in to Git, and sync with GitHub. The checkin then triggers Amplify’s build and deploy process. I even get email notifications from Amplify when a new build starts and when it completes. It's not perfect though, I don't have a test environment so essentially I just push everything "Production" :-)

<p> <strong>So here's 10,000 foot view </strong>

<p>Out of the myriad ways to build a full-stack app, the patch I choose is pretty straightforward – plain vanilla JavaScript + AWS.

<p>As I said, <strong>AWS Amplify</strong> is used to host and manage the web application. I’ve connected it to my GitHub repo so I run automated builds & deployments, I even used it to leverage AWS’s Route 53 service to secure a domain name and set up hosting. One stop shop. I set this up first, giving me an empty shell to build from as I started to add more AWS features.

<p>For storage I use <strong>AWS DynamoDB</strong> NoSQL database. Two main reasons I choose DynamoDB – I didn’t have much hands-on experience with NoSQL DBs, and having spent 20+ years using RDBMS as the main persistence layer this is a mind-bending concept that I am curious to understand. As an Architect I always ask the same main questions when looking at a new technology: What’s it for? Why would I want to use it? What’s it good at? And more importantly what’s it <i>not</i> good at? The second reason was speed and my simplistic data model. OK, maybe that’s three reasons. Either way, DynamoDB is what I hung my hat on.

<p>We've been using <strong>AWS Lambda</strong> functions for quite a while with clients, so it was an easy choice for me to use them to glue the front-end of the application to the back-end. I have a few services defined for my DynamoDB CRUD operations, as well as to fetch data. The functions are written in JavaScript and use Lambda's node.js 14.x runtime, and AWS manages the Node environment for me. One less thing to do…

<p>The Lambda functions are exposed as REST services using <strong>AWS API Gateway</strong> (the ones I wrote at least). API Gateway also leverages AWS AIM to make sure the services hitting the DynamoDB are allowed to. Which is always nice. To be honest I don't use AIM for anything more than that right now, so I've only figured out enough to write the policies (in JSON) that are needed to make sure only my services can hit my DB.

<p>I use an <strong>AWS S3 Bucket</strong>s for images – it’s rudimentary for now – I upload images using the AWS S3 console, then just call the URL from the Application. Again, I wanted to play around with S3 and this was an easy way to do it.

<p>And the front-end is written in<strong> plain JavaScript</strong> for now. I use the <a link=”https://openweathermap.org” target=”\_blank”> OpenWeatherMap API</a> to get weather data, basically because we all talk about the weather, and I wanted to find a free external API to integrate into the web app.

<p>And last but not least I use <strong>Cognito</strong> to provide controlled access to a specific page, which is for me only. It was super-easy to setup & configure, comes with its own UI (which can be styled with CSS), and although I’d likely not use it for anything but a greenfield app, it’s pretty nice.

<p>And that’s about it – all that comes together to create what you see here – a place to blog about things, and a sandbox to try out stuff.

<p>Enjoy!