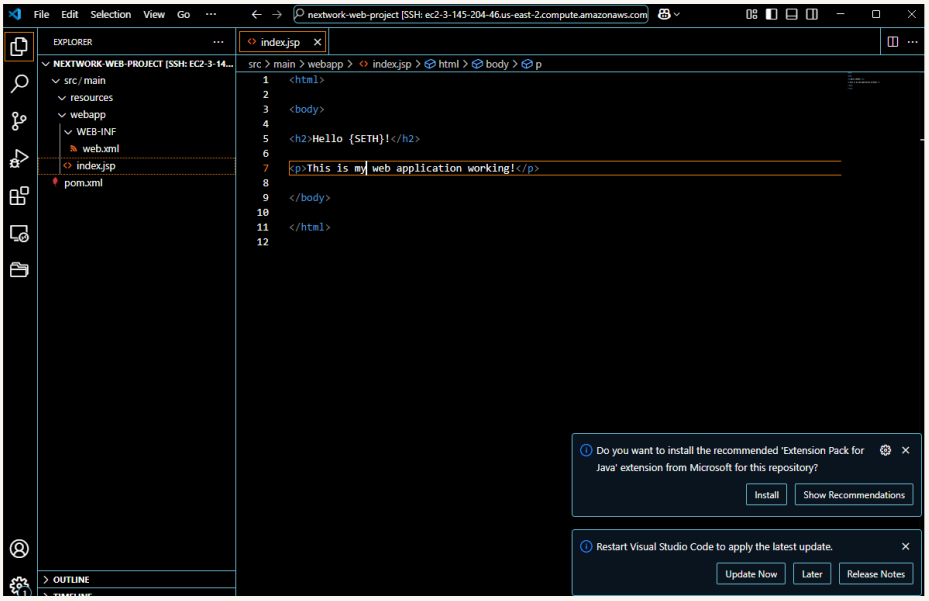
Set Up a Web App Using AWS and VS Code

Seth Sekyere



Seth Sekyere

Introducing Today's Project!

Today, I'll launch an AWS EC2 instance, connect to it via VS Code using SSH, install Java and Maven, and create a basic web app to set up the foundation for building my CI/CD pipeline.

Key tools and concepts

Services I used were AWS EC2 for hosting and VS Code Remote SSH for development. Key concepts I learnt include connecting to remote servers, editing files with an IDE, Maven project structure, and deploying Java web apps with dynamic content.

Project reflection

One thing I didn't expect was how smoothly VS Code Remote SSH integrates with an EC2 instance, making remote editing feel almost like working locally. It made managing and updating the web app much easier than I anticipated.

This project took me approximately 20 minutes. The most challenging part was setting up the remote SSH connection. It was most rewarding to successfully edit and deploy the web app on the EC2 instance using VS Code.

Seth Sekyere

This project is part one of a series of DevOps projects where I'm building a CI/CD pipeline! I'll be working on the next project in the coming days to continue enhancing my skills and automating deployments.

Seth Sekyere

Launching an EC2 instance

I started this project by launching an EC2 instance because I need a virtual server in the cloud to develop, run, and host my web app as part of building a complete CI/CD pipeline.

I also enabled SSH

SSH is a secure protocol that lets me safely connect to a remote server. I enabled SSH so that only I can access my EC2 instance using my private key, ensuring encrypted and authorized communication with the server.

Key pairs

A key pair is a set of security credentials (public and private keys) used to securely access an EC2 instance. AWS stores the public key, and you use the private key (.pem) to log in safely via SSH.

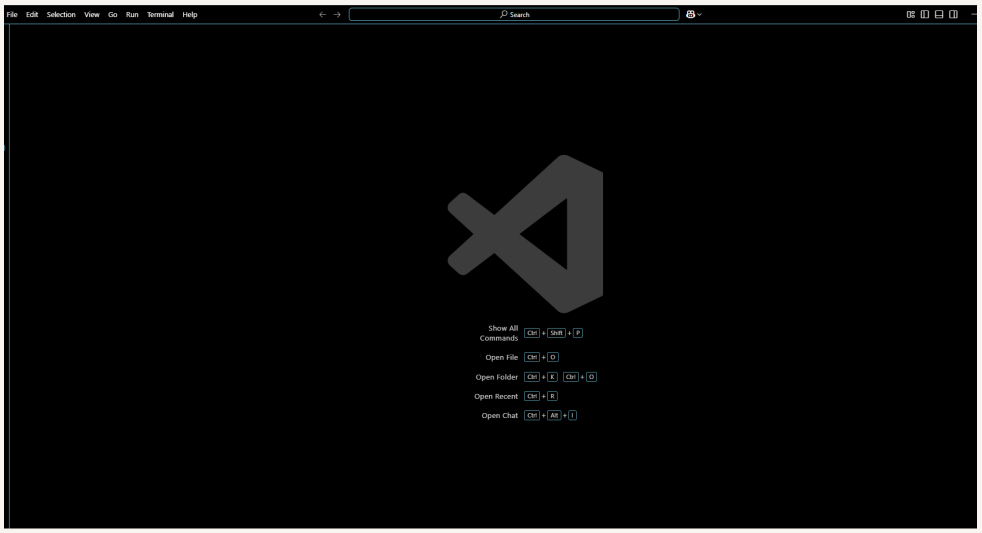
Once I set up my key pair, AWS automatically downloaded a `.pem` file to my computer. This private key file allows me to securely connect to my EC2 instance using SSH.

Seth Sekyere

Set up VS Code

VS Code is a free, lightweight code editor by Microsoft that supports many programming languages. It lets you write, edit, and manage code, and connect remotely to servers like EC2, making development easier and more efficient.

I installed VS Code to connect securely to my EC2 instance via SSH, so I can write, edit, and manage my web app's code directly on the cloud server with a user-friendly interface.

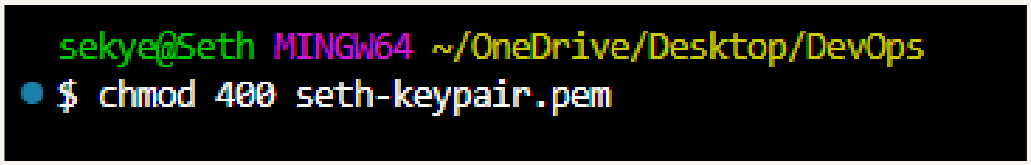


Seth Sekyere

My first terminal commands

A terminal is a text-based tool to control your computer. The first command I ran for this project is `cd ~/Desktop/DevOps` to move into the folder where my `.pem` file is stored so I can connect to my EC2 instance.

I also updated my private key's permissions by running `chmod 400 seth-keypair.pem` in the terminal, which makes the file readable only by me and keeps it secure from unauthorized access.



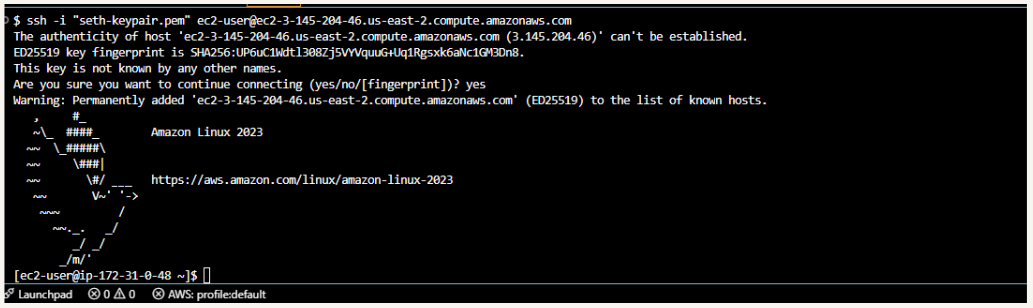
Seth Sekyere

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command `ssh -i ~/Desktop/DevOps/seth-keypair.pem ec2-user@[My_Public_IPv4_DNS]` in the terminal, using my private key and the instance's public DNS to securely access the server.

This command required an IPv4 address

A server's IPV4 DNS is the public internet address that directs users and devices to the server. It translates the server's IP number into a readable name, allowing easy connection to the server over the web or network.



Seth Sekyere

Maven & Java

Apache Maven is a build automation tool used primarily for Java projects. It manages project dependencies, compiles code, runs tests, and packages applications, making the build process easier and more consistent.

Maven is required in this project because it automates building, testing, and packaging the Java web app, managing dependencies efficiently and ensuring the project is built consistently every time.

Java is a popular, versatile programming language used to build applications from mobile apps to large systems. It’s platform-independent, meaning code runs anywhere with a Java Virtual Machine (JVM), making it widely used in software development.

Java is required in this project because Maven depends on Java to build and run the web app. Without Java installed, we can't compile or manage the project, making it essential for creating and deploying our Java-based web application.

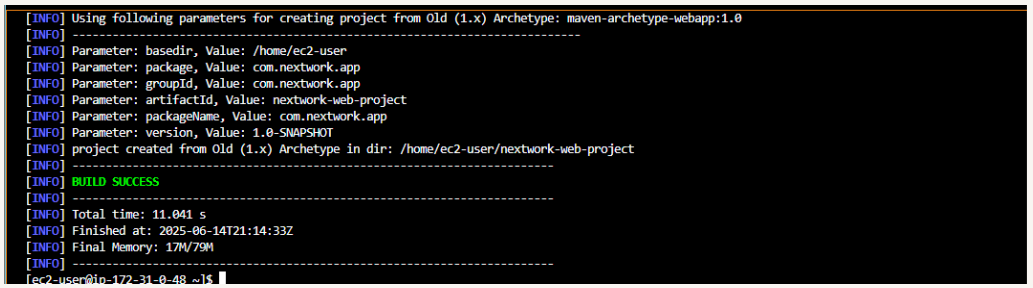
Seth Sekyere

Create the Application

I generated a Java web app using the command `mvn archetype:generate -DgroupId=com.nextwork.app -DartifactId=nextwork-web-project -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false` to create a basic web app structure automatically.

I installed Remote - SSH, which is a VS Code extension that lets me securely connect to my EC2 instance and work on files directly there. I installed it to easily edit and manage my web app on the remote server using VS Code's features.

Configuration details required to set up a remote connection include the Host name (EC2 instance's IPv4 DNS), the User (ec2-user), and the IdentityFile (path to the private key .pem file) for secure SSH access.

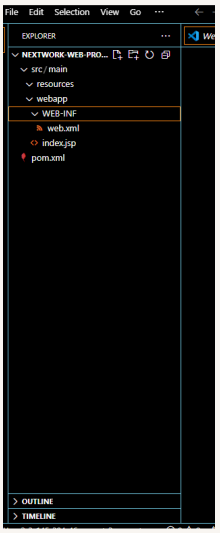


Seth Sekyere

Create the Application

Using VS Code's file explorer, I could see the full structure of the web project, including folders like `src`, `webapp`, and `resources`, as well as files like `pom.xml` that define the project's configuration and dependencies.

Two of the project folders created by Maven are `src` and `webapp`, which hold the source code. `src` is the main folder, and `webapp` contains the frontend files like HTML, CSS, JavaScript, and JSP for the web application.



Seth Sekyere

Using Remote - SSH

index.jsp is a Java Server Page file used in Java web apps. It combines HTML and Java code to create dynamic content that can change based on user input or data, unlike static HTML files.

I edited index.jsp by opening it in VS Code's editor view, replacing the placeholder code updating with my name to customize the web page content.