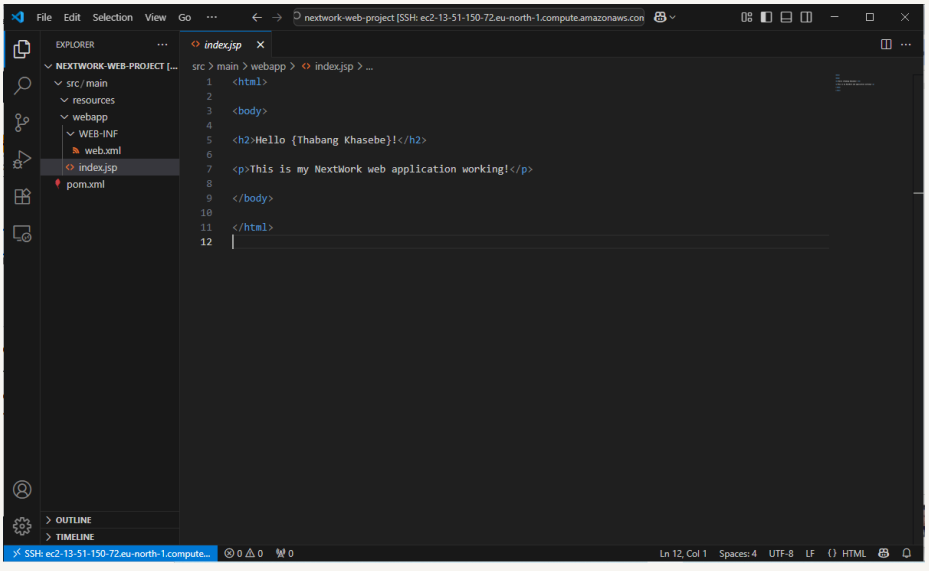
Set Up a Web App Using AWS and VS Code

Thabang Khasebe



Thabang Khasebe

Introducing Today's Project!

Today, I will build a simple web app using AWS and VS Code to learn how cloud services and development tools work together. This will help me understand the basics of deploying and managing apps in a DevOps environment.

Key tools and concepts

Services I used were AWS EC2 for hosting and VS Code for development. Key concepts I learnt include launching EC2 instances, connecting via SSH, installing Java and Maven, and generating and managing a Java web app project using Maven.

Project reflection

One thing I didn't expect was how smoothly setting up remote development with VS Code and SSH worked—it made managing and editing my web app on the EC2 instance much easier than I thought.

This project took me approximately 3 hours. The most challenging part was setting up the secure SSH connection between VS Code and the EC2 instance. It was most rewarding to successfully generate and view my Java web app files remotely.

Thabang Khasebe

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 tomorrow.

Thabang Khasebe

Launching an EC2 instance

I started this project by launching an EC2 instance because it provides a secure, scalable virtual server in the cloud to host and run my web app, allowing me to build and manage everything remotely using AWS.

I also enabled SSH

SSH is a secure protocol that allows encrypted remote access to servers. I enabled SSH so that I can securely connect to and manage my EC2 instance from my local computer.

Key pairs

A key pair is a set of security credentials consisting of a public key and a private key used to securely connect to an EC2 instance, allowing encrypted access via SSH without using passwords.

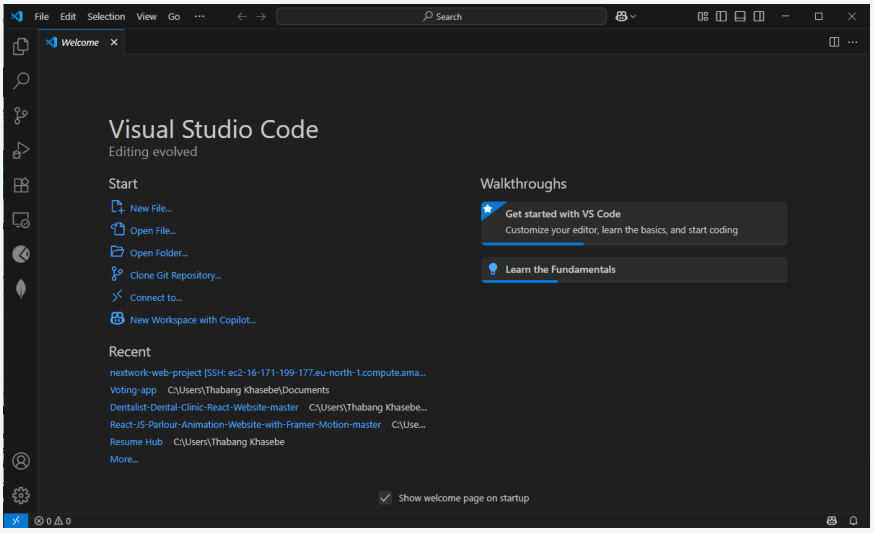
Once I set up my key pair, AWS automatically downloaded a private key file (.pem) to my local computer, which I use to securely connect to my EC2 instance via SSH.

Thabang Khasebe

Set up VS Code

VS Code is a lightweight, powerful code editor that supports many programming languages and tools. It includes an integrated terminal, making it easy to write, edit, and run code while connecting to remote servers like EC2 instances.

I installed VS Code to write and edit my web app's code and use its terminal to securely connect to my EC2 instance for managing and deploying the app directly on the cloud server.

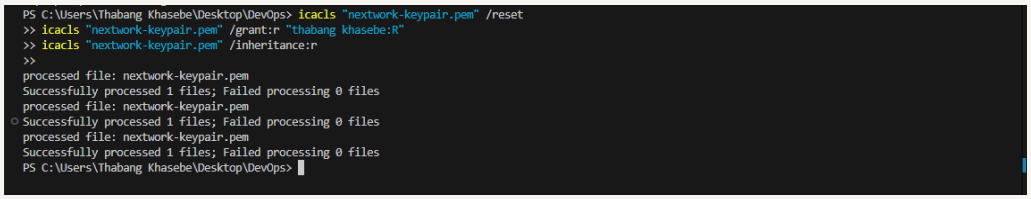


Thabang Khasebe

My first terminal commands

A terminal is a command-line tool used to interact with your computer and remote servers. The first command I ran for this project is `cd C:\Users\Thabang Khasebe\Desktop\DevOps` to navigate to the folder containing my `.pem` key file.

I updated my private key's permissions using `icacls` to restrict access. I ran `/reset`, granted read access to my user with `/grant:r`, and removed inheritance with `/inheritance:r` to secure the key for SSH access.



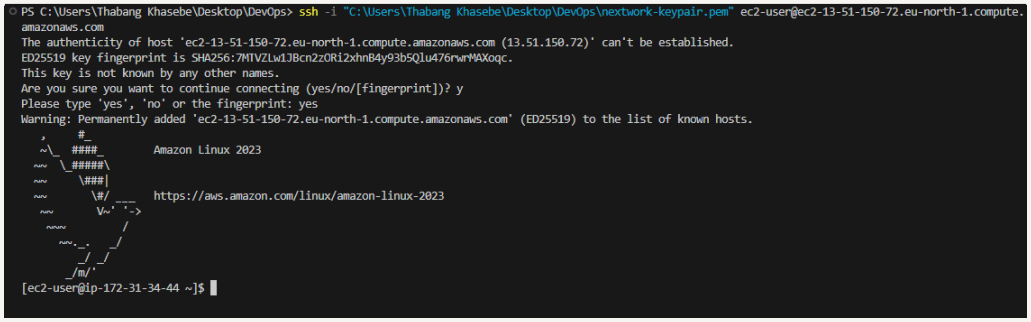
Thabang Khasebe

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command `ssh -i "C:\Users\Thabang Khasebe\Desktop\DevOps\nextwork-keypair.pem" ec2-user@ec2-13-51-150-72.eu-north-1.compute.amazonaws.com"` in the VS Code terminal, using my key and the instance's public DNS.

This command required an IPv4 address

A server's IPV4 DNS is a human-readable domain name that maps to the server's IPv4 address, making it easier to connect to the server over the internet instead of using a numeric IP address.



Thabang Khasebe

Maven & Java

Apache Maven is a build automation and project management tool used primarily for Java applications. It helps manage dependencies, compile code, run tests, and package applications efficiently.

Maven is required in this project because it automates the process of building, testing, and packaging the Java web app. It also manages dependencies, making development more efficient and organized.

Java is a high-level, object-oriented programming language used to build platform-independent applications. It's widely used for web apps, mobile apps, and enterprise software because of its reliability, portability, and strong community support.

Java is required in this project because we're building a Java-based web app, and it provides a reliable, scalable, and platform-independent environment for developing and running applications on the cloud.

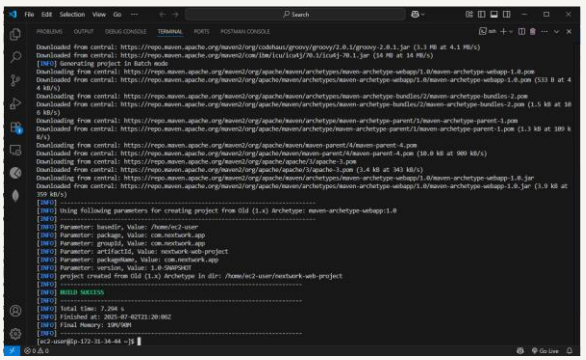
Thabang Khasebe

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 project structure.

I installed Remote - SSH, which is a VS Code extension that lets me connect to remote servers via SSH directly from the editor. I installed it to easily access and edit my Java web app files on the EC2 instance from my local machine.

Configuration details required to set up a remote connection include the EC2 instance's public DNS, the SSH username (like ec2-user), and the path to the private key (.pem) file used for authentication.

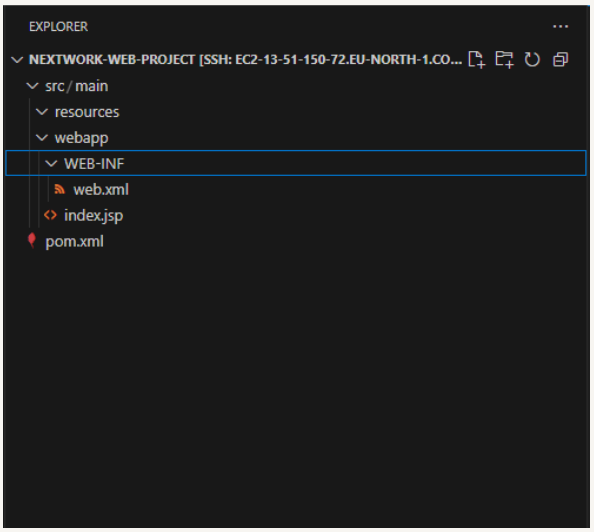


Thabang Khasebe

Create the Application

Using VS Code's file explorer, I could see the full directory structure of my Java web app on the EC2 instance, including folders like src and configuration files needed to build and run the app.

Two of the project folders created by Maven are src and webapp, which contain the source code and web resources of the Java application. The src folder holds Java classes, while the webapp folder contains JSP file.



Thabang Khasebe

Using Remote - SSH

index.jsp is the default Java Server Page file that serves as the main entry point for the web application's user interface, handling dynamic content and client requests.

I edited index.jsp by adding my name 'Thabang Khasebe'.