

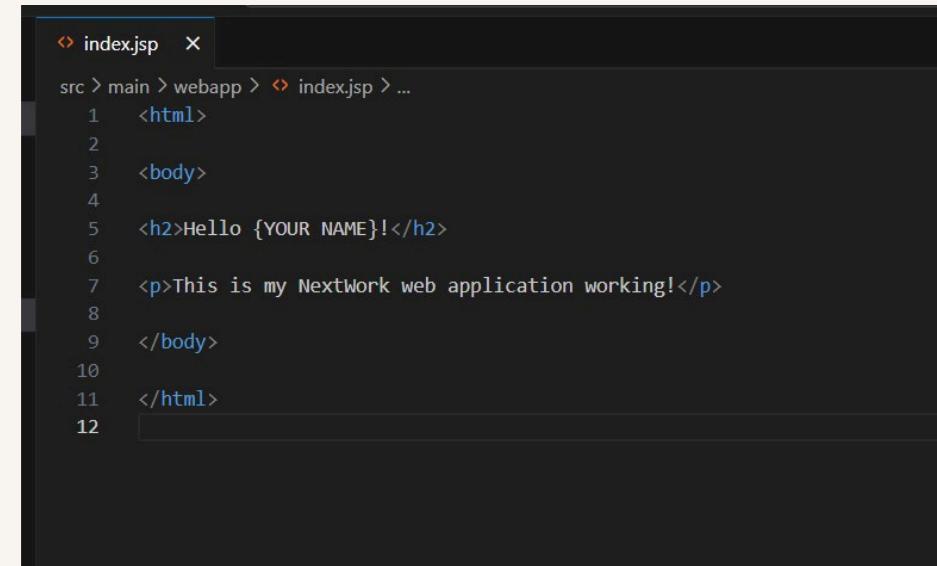


[nextwork.org](https://nextwork.org)

# Set Up a Web App Using AWS and VS Code



Gursimran SINGH



```
↳ index.jsp  X
src > main > webapp > ↳ index.jsp > ...
1   <html>
2
3   <body>
4
5   <h2>Hello {YOUR NAME}!</h2>
6
7   <p>This is my NextWork web application working!</p>
8
9   </body>
10
11  </html>
12
```



# Introducing Today's Project!

First day of the challenge! I'll be learning how to build a simple web app using AWS and VS Code — setting the foundation for hands-on DevOps work ahead.

## Key tools and concepts

Services I used were EC2, SSH, and VS Code with Remote - SSH. Key concepts I learnt include launching and connecting to a cloud server, setting up a secure development environment, and building a basic Java web app using Maven.

## Project reflection

One thing I didn't expect in this project was how smoothly VS Code's Remote - SSH extension worked—it made editing files on the EC2 instance feel just like working locally, which really boosted my productivity.

This project took me approximately 1.5 hours. The most challenging part was when my EC2 instance stopped responding—I had to restart it and set up a new SSH config for the updated IP. It was most rewarding to see the web app finally run!



**Gursimran SINGH**  
NextWork Student

[nextwork.org](http://nextwork.org)

---

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 as part of my 7-day challenge to deepen my cloud and DevOps skills.



# Launching an EC2 instance

I started this project by launching an EC2 instance because it gives me a virtual server in the cloud to run and test applications. It's a key part of learning how to deploy, manage, and automate infrastructure in a DevOps workflow.

## I also enabled SSH

SSH is a secure way to remotely access a server. I enabled SSH so that I can safely connect to my EC2 instance from my computer and manage it using the terminal without exposing sensitive data.

## Key pairs

A key pair is a set of two cryptographic keys: a public key and a private key. It's used to securely connect to your EC2 instance. I use the private key on my computer to prove my identity and access the instance without a password.

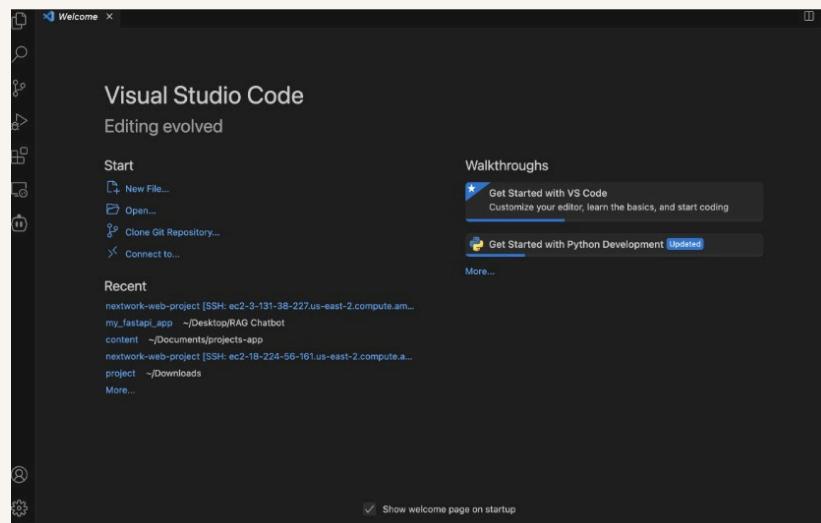
Once I set up my key pair, AWS automatically downloaded a ` `.pem` file to my computer. This private key file is needed to securely connect to my EC2 instance using SSH.



# Set up VS Code

VS Code is a lightweight and powerful code editor developed by Microsoft. It lets me write, edit, and run code easily. I'm using it to manage my files and connect to my EC2 instance through its built-in terminal.

I installed VS Code to write and manage code, use the terminal to connect to my EC2 instance, and organize my files—all in one place for a smoother DevOps workflow.





Gursimran SINGH

NextWork Student

[nextwork.org](http://nextwork.org)

# My first terminal commands

A terminal is a tool that lets me interact with my computer using text commands. The first command I ran for this project is `cd

C:\Users\gsing\OneDrive\Documents\DevOps` to navigate to my project folder.

I also updated my private key's permissions by running a few `icacls` commands in the terminal. This restricted access so only I can read the file, which is important for securely connecting to my EC2 instance using SSH.

```
PS C:\Users\gsing\OneDrive\Documents\DevOps> icacls "nextwork-keypair.pem" /reset
● >> icacls "nextwork-keypair.pem" /grant:r "gsing:R"
>> icacls "nextwork-keypair.pem" /inheritance:r
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
○ PS C:\Users\gsing\OneDrive\Documents\DevOps> █
```



# SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command `ssh -i "nextwork-keypair.pem" ec2-user@<my-IPv4-DNS>`. This securely connects my local machine to the cloud server using the private key.

## This command required an IPv4 address

A server's IPV4 DNS is a human-readable address that maps to the server's IP, making it easier to access over the internet—like a web address that points to your EC2 instance.

```
PS C:\Users\gsing\OneDrive\Documents\DevOps> ssh -i C:\Users\gsing\OneDrive\Documents\DevOps\nextwork-keypair.pem ec2-user@98.80.77.23
5
The authenticity of host '98.80.77.235 (98.80.77.235)' can't be established.
ED25519 key fingerprint is SHA256:1xtcd8DQXL/T11qhsAF99Tc280BQ1E/6rTDfpHogQLM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.80.77.235' (ED25519) to the list of known hosts.
,      #
~\###_
~~\####\ Amazon Linux 2023
~~\###_
~~ \#/   https://aws.amazon.com/linux/amazon-linux-2023
~~ \v-' .->
~~ \_/
~~ \_/
~~ \_/
[ec2-user@ip-172-31-27-104 ~]$
```

A circular profile picture of a young man with dark hair, wearing a blue hoodie and jeans, sitting outdoors on a bench or ledge.

## Maven & Java

Apache Maven is a build automation tool used mainly for Java projects. It helps manage project dependencies, compile code, and package applications, making it easier to build and deploy software consistently.

Maven is required in this project because it automates the process of building and managing Java applications. It handles dependencies, compiles the code, and packages the app, helping us streamline deployment on the EC2 instance.

Java is a popular, high-level programming language used to build a wide range of applications. It's platform-independent, meaning code written in Java can run anywhere, making it ideal for web apps, enterprise software, and cloud-based services.

Java is required in this project because the sample web application we're deploying is written in Java. It allows the app to run on the EC2 instance using tools like Maven and Amazon Corretto for building and executing the code.

# Gursimran SINGH

# NextWork Student

[nextwork.org](http://nextwork.org)

# Create the Application

I generated a Java web app using the command `mvn archetype:generate`, which sets up a basic project structure with the necessary files and folders to start building and running the application.

I installed Remote - SSH, which is a VS Code extension that lets you securely connect to remote servers. I installed it to open, edit, and manage files directly on my EC2 instance from within VS Code.

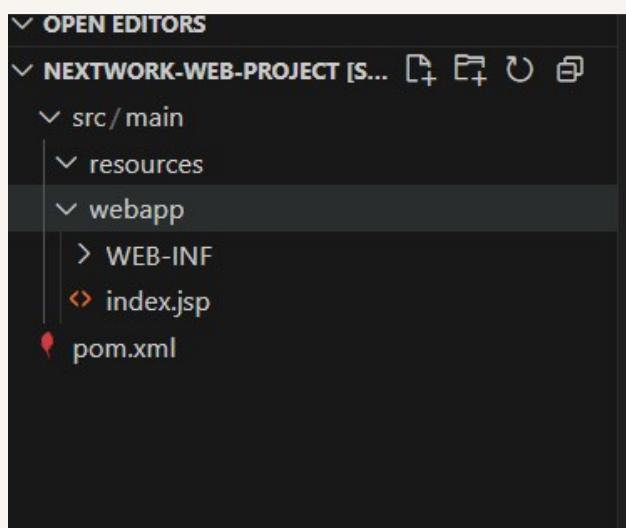
Configuration details required to set up a remote connection include the EC2 instance's public IP ('HostName'), the path to the private key file ('IdentityFile'), the SSH username ('User'), and a label to identify the connection ('Host').



# Create the Application

Using VS Code's file explorer, I could see the entire directory structure of my EC2 instance, including the Java project folders and files generated by Maven—making it easy to navigate and edit the code.

Two of the project folders created by Maven are `src` and `webapp`, which organize the application's code. `src` holds the Java source files, while `webapp` contains the web content like HTML and JSP files used in the user interface.

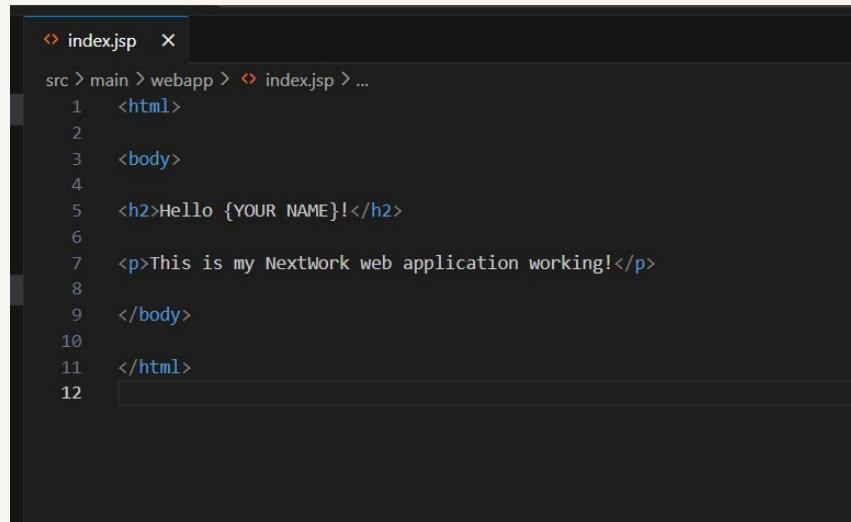




# Using Remote - SSH

index.jsp is the default landing page for a Java web application. It's a JSP (JavaServer Pages) file that combines HTML with Java to dynamically generate content users see when they visit the app.

I edited `index.jsp` by opening it in VS Code through the Remote - SSH connection to my EC2 instance. I updated the HTML content inside the file to customize what appears on the web app's main page.



```
index.jsp
src > main > webapp > index.jsp > ...
1  <html>
2
3  <body>
4
5  <h2>Hello {YOUR NAME}!</h2>
6
7  <p>This is my NextWork web application working!</p>
8
9  </body>
10 </html>
11
12
```



[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

