



# Set Up a Web App Using AWS and VS Code

I

lue

```
src > main > webapp > index.jsp > ...
1   <html>
2
3   <body>
4
5   <h2>Hello I-ueRya-n!</h2>
6
7   <p>This is my NextWork web application working!</p>
8
9   </body>
10
11  </html>
```

# Introducing Today's Project!

Today, I'll launch an AWS EC2 instance, connect to it remotely using VS Code and SSH, install Maven and Java, and generate a basic web app. I'll also try editing code without an IDE to compare the experience.

## Key tools and concepts

Services I used were AWS EC2 for hosting and VS Code Remote-SSH for editing remotely. Key concepts I learnt include secure SSH connections, Maven project structure, and how JSP files create dynamic web content compared to static HTML.

## Project reflection

One thing I didn't expect in this project was how easy it is to edit files on a remote server using VS Code's Remote-SSH extension—it felt almost like working locally, even though everything was running in the cloud!

It took me about 90 minutes to complete this project, including setting up the EC2 instance, configuring SSH access, and editing the web app files using VS Code Remote-SSH.



I

Iue  
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 this series tomorrow.

# Launching an EC2 instance

I started this project by launching an EC2 instance because it provides a scalable, on-demand virtual server in the AWS Cloud, allowing me to develop and deploy applications quickly without investing in physical hardware.

## I also enabled SSH

SSH is a secure protocol for remotely accessing servers. I enabled SSH so I can safely connect to my EC2 instance from my computer, allowing me to manage and control it without risking data exposure or unauthorized access.

## Key pairs

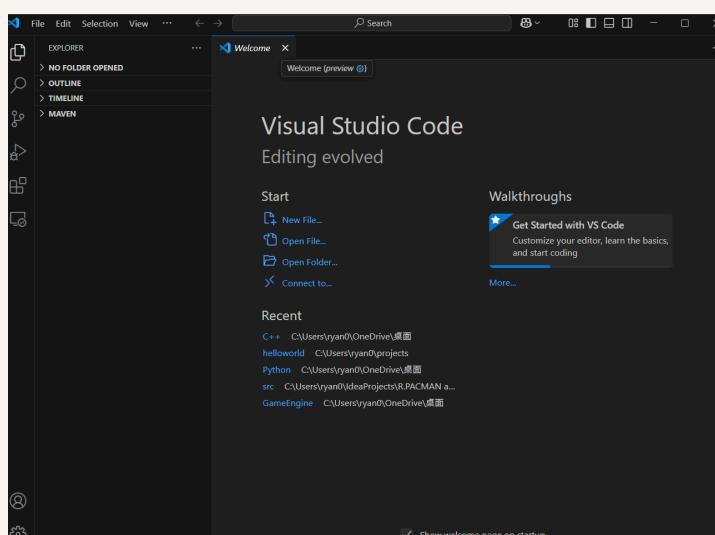
A key pair is a set of security keys—a public key stored on an EC2 instance and a private key kept by you. It enables secure SSH access, ensuring only authorized users can connect to and manage the instance safely.

Upon the creation of my key pair, AWS conveniently downloaded a private key file (.pem) onto my computer. This file is extremely important for transferring a secure connection through SSH to my EC2 instance.

# Set up VS Code

VS Code is a free, lightweight code editor by Microsoft. It supports many languages, offers features like syntax highlighting, debugging, and extensions, and includes a built-in terminal to help developers write and manage code efficiently.

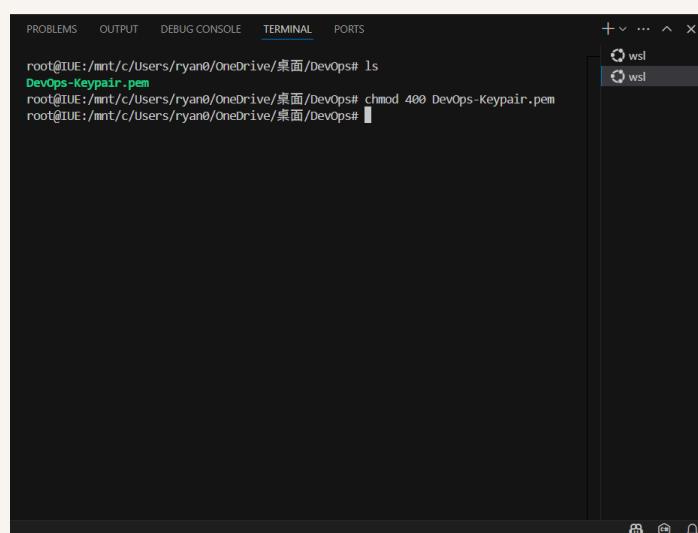
I will use VS Code to write, edit, and manage my web app's code, connect securely to my EC2 instance through its terminal, and organize project files. This makes development faster, easier, and more efficient with helpful coding features.



# My first terminal commands

Navigating the terminal means changing folders using commands like cd. Here, I move to the DevOps folder on my Desktop where my .pem file is stored, so the terminal can access it and connect securely to my EC2 instance.

I changed my file's permissions by running chmod 400 nextwork-keypair.pem, which makes the file readable only by me and blocks access for others. This keeps my private key secure for safe SSH access to my EC2 instance.



A screenshot of a terminal window within a dark-themed IDE interface. The terminal tab is selected at the top. The command history shows:

```
root@TUE:/mnt/c/Users/ryan0/OneDrive/桌面/DevOps# ls
DevOps-Keypair.pem
root@TUE:/mnt/c/Users/ryan0/OneDrive/桌面/DevOps# chmod 400 DevOps-Keypair.pem
root@TUE:/mnt/c/Users/ryan0/OneDrive/桌面/DevOps#
```

The right side of the window shows a sidebar with two entries: "wsl" and "wsl".

# SSH connection to EC2 instance

To connect to my EC2 instance, I ran `ssh -i my-keypair.pem ec2-user@my-ec2-public-dns` in Git Bash after navigating to the .pem file location and setting its permissions with `chmod 400`. This securely connected me to the server.

## This command required an IPv4 address

The server's IPv4 DNS is essentially a domain name that resolves to the server's public IPv4 address, making it easier to access the server by a human-readable name than by a numeric IP. It facilitates easy internet connectivity for my EC2 instance.

```
PS C:\Users\ryan0\OneDrive\桌面\DevOps> ssh -i C:\Users\ryan0\OneDrive\桌面\DevOps\DevOps-KeyPair.pem ec2-user@ec2-3-1-06-170-13.ap-southeast-2.compute.amazonaws.com
, #_
~\ #####      Amazon Linux 2023
~~ \####\_
~~ \###|
~~ \#/   ____ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~\  ->
~~ \_/
~~ \_/
Last login: Sat Apr 26 14:10:27 2025 from 103.131.13.226
[ec2-user@ip-172-31-15-77 ~]$
```

## Maven & Java

Apache Maven is a powerful open-source tool used to manage Java projects. It automates building, testing, and packaging code, and handles dependencies, making it easier for developers to organize and maintain their applications efficiently.

I'm using Maven in this project because it automates building, testing, and packaging Java code, and manages all project dependencies for me. This makes development faster, more organized, and ensures consistency across different environments.

Java is a popular programming language used for building many types of apps. Amazon Corretto 8 is Amazon's free, stable version of Java 8. We install it so Maven can work, and so our EC2 instance can build and run Java applications reliably.

I'm using Java in this project because it's a robust, secure, and platform-independent language. Java makes it easy to build, run, and maintain web apps, and it's required by Maven, which we're using to manage our project and its dependencies.

# Create the Application

I generated a Java web app using the command mvn archetype:generate -DgroupId=com.IUE.app \ -DartifactId=IUE-web-project\ -DarchetypeArtifactId=maven-archetype-webapp \ -DinteractiveMode=false , which created the basic project structure and files

I installed Remote - SSH so I could securely connect VS Code to my EC2 instance. This lets me open, edit, and manage files directly on the remote server from my local VS Code, making development easier without needing to transfer files back and forth

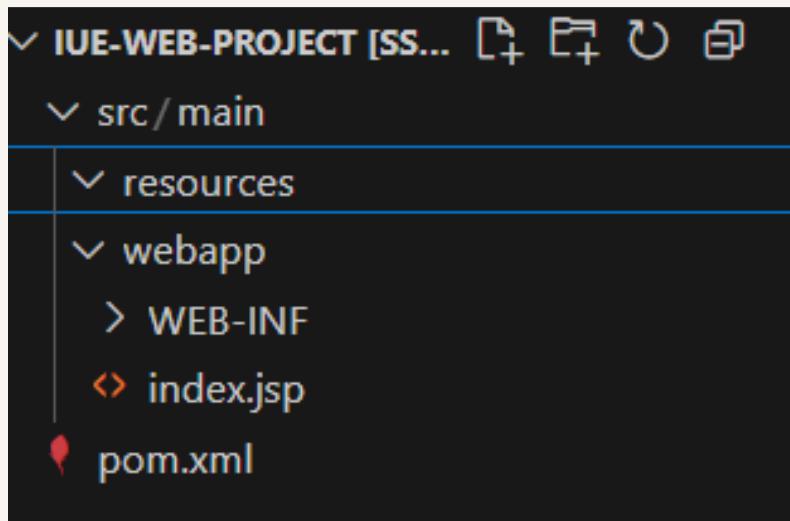
The configuration file had details for connecting to my EC2 instance: the host alias, host name (EC2 IP address), user name, and the path to my SSH private key. These settings let VS Code securely connect to and access files on the remote server.

```
[ec2-user@ip-172-31-15-77 ~]$ mvn archetype:generate \
-DgroupId=com.IUE.app \
-DartifactId=IUE-web-project \
-DarchetypeArtifactId=maven-archetype-webapp \
-DinteractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] Building Maven Stub Project (No POM) 1
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO]
[INFO] Using following parameters for creating project from old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] Parameter: basedir, Value: /home/ec2-user
[INFO] Parameter: package, Value: com.IUE.app
[INFO] Parameter: artifactId, Value: IUE-web-project
[INFO] Parameter: packagingName, Value: com.IUE.app
[INFO] Parameter: groupId, Value: com.IUE.app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from old (1.x) Archetype in dir: /home/ec2-user/IUE-web-project
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.7153 s
[INFO] Finished at: 2025-04-26T14:26:43Z
[INFO] Final Memory: 175/98M
[INFO] -----
```

# Create the Application

The IUE-web-project folder contains all parts of my Java web app. src/main/webapp holds web files like HTML, CSS, and JSP. src/main/resources has config files. pom.xml is Maven's build config file. This structure helps organize my project.

The src folder contains your project's source code. Inside it, webapp is where you put web files like HTML, CSS, JavaScript, and JSP. These files control what users see and interact with when they visit your web app in a browser.



# Using Remote - SSH

index.jsp is the default home page of your Java web app. It's a JSP (JavaServer Pages) file that combines HTML and Java code, letting you display dynamic content to users when they first visit your site.

I edited `index.jsp` in VS Code by replacing the placeholder with your custom code. Unlike HTML, `index.jsp` can include Java code for dynamic content. HTML is static, while JSP lets my web app respond to users or data changes.

```
src > main > webapp > index.jsp > ...
1   <html>
2
3   <body>
4
5   <h2>Hello I-ueRya-n!</h2>
6
7   <p>This is my NextWork web application working!</p>
8
9   </body>
10
11  </html>
```



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

# The place to learn & showcase your skills

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

