



# Set Up a Web App Using AWS and VS Code

K

Kehinde Abiuwa

The screenshot shows a VS Code interface with the following details:

- Explorer View:** Shows the project structure: `NEXTWORK-WEB-PROJECT [SSH: 13.51.197.133]`. The tree includes `src/main/resources`, `src/main/webapp/WEB-INF/web.xml`, `src/main/webapp/index.jsp`, and `pom.xml`.
- Editor View:** Displays the `index.jsp` file content:

```
<html>
<body>
<h2>Hello Kenny!</h2>
<p>This is my NextWork web application working!</p>
</body>
</html>
```
- Bottom Status Bar:** Shows connection information: `SSH: 13.51.197.133`, file statistics (`0 0 0`), and code analysis results (`Ln 5, Col 16 Spaces: 4 UTF-8 LF HTML`).

# Introducing Today's Project!

In this project, I will demonstrate how to set up a Web App in the cloud... I'm doing this project to learn the basics of building a web app using AWS and another tool called VS Code.

## Key tools and concepts

Services I used were AWS EC2 for hosting my app and VS Code with the Remote - SSH extension for remote development. Key concepts I learnt include secure SSH access using key pairs, connecting to and managing cloud servers, and editing and deploying a Java web app in a remote environment.

## Project reflection

One thing I didn't expect in this project was how seamless and powerful it is to edit and manage remote server files directly from my local VS Code using the Remote - SSH extension.

This project took me approximately 30 minutes. The most challenging part was setting up the secure SSH connection correctly, and it was most rewarding to successfully connect and edit my web app files remotely with ease.



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 right away



# Launching an EC2 instance

I started this project by launching an EC2 instance because I need a server to host my cloud based app

## I also enabled SSH

SSH is a secure protocol for remotely accessing and managing servers. I enabled SSH so that I can securely connect to and control my EC2 instance from my local computer.

## Key pairs

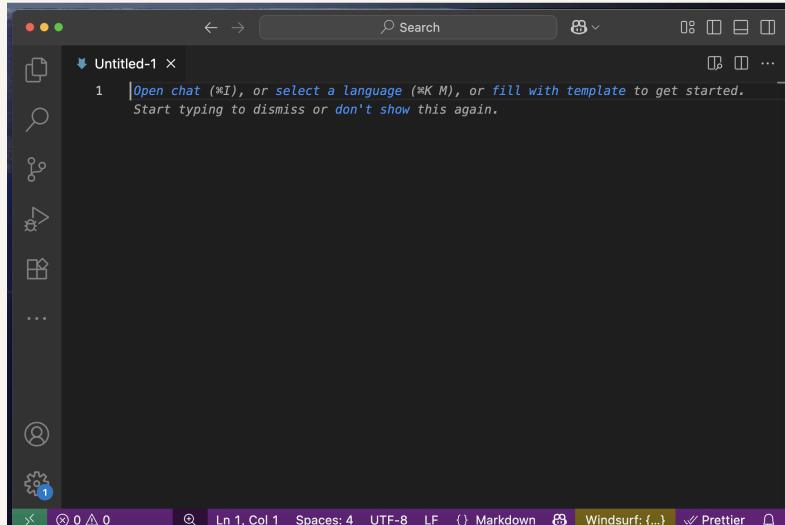
A key pair is a set of security credentials (a public key stored on the EC2 instance and a private key you keep) used to securely connect to your server via SSH.

Once I set up my key pair, AWS automatically downloaded a .pem file. The downloaded .pem file is my private key, which I will use to securely access your EC2 instance via SSH.

# Set up VS Code

VS Code is a lightweight, powerful code editor that lets you write, edit, and manage your code with built-in support for terminals and many programming tools.

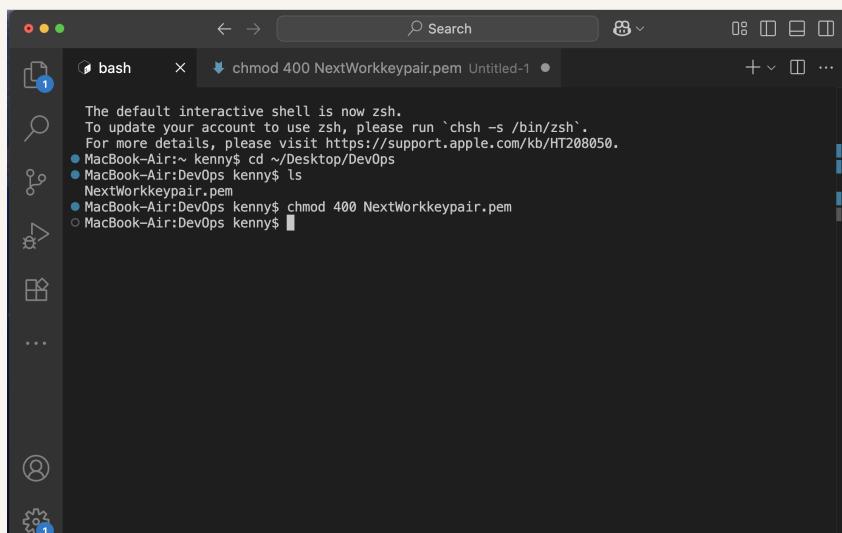
I installed VS Code to write and manage my code efficiently and to use its integrated terminal to connect to and control my EC2 instance.



# My first terminal commands

A terminal is a text-based interface that lets you enter commands to interact directly with your computer or a remote server like your EC2 instance. The first command I ran for this project is 'cd ~/Desktop/DevOps' which navigates me to my devops folder located on my local computer

I also updated my private key's permissions by running this command 'chmod 400 NextWorkkeypair.pem'. Which changes the permissions of my .pem file so that you have access to it when you connect to your EC2 instance later.



The screenshot shows a terminal window titled 'bash' with the command 'chmod 400 NextWorkkeypair.pem' being run. The terminal output includes a note about the default interactive shell being zsh, instructions to update the account to use zsh, and the command history showing the directory change and permission change.

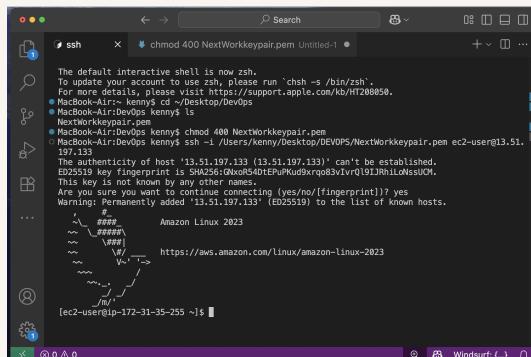
```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
● MacBook-Air:~ kenny$ cd ~/Desktop/DevOps  
● MacBook-Air:DevOps kenny$ ls  
NextWorkkeypair.pem  
● MacBook-Air:DevOps kenny$ chmod 400 NextWorkkeypair.pem  
○ MacBook-Air:DevOps kenny$
```

# SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command 'ssh -i /Users/kenny/Desktop/DEVOPS/NextWorkkeypair.pem ec2-user@13.51.197.133'

## This command required an IPv4 address

A server's IPv4 DNS is the public hostname that resolves to its IPv4 address, allowing you to connect to the server over the internet using a readable name instead of numeric IP.



The screenshot shows a terminal window titled 'ssh' with the following session details:

- The default interactive shell is now zsh.
- For more details, please visit <https://support.apple.com/kb/H7208080>.
- MacBook-Air:~ kenny\$ cd ~/Desktop/DevOps
- MacBook-Air:kenny\$ ls
- MacBook-Air:kenny\$ ls
- MacBook-Air:DevOps kenny\$ chmod 400 NextWorkkeypair.pem
- MacBook-Air:DevOps kenny\$ ssh -i /Users/kenny/Desktop/DEVOPS/NextWorkkeypair.pem ec2-user@13.51.197.133
- The authenticity of host '13.51.197.133 (13.51.197.133)' can't be established.
- ED25519 key fingerprint is SHA256:0Xor54t0PePuKu0x9qoB3yIvrQ9Jhnl0NsU0M.
- Please check the fingerprint and enter yes or no to continue connecting.
- Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
- Warning: Permanently added '13.51.197.133' (ED25519) to the list of known hosts.

The terminal then displays the Amazon Linux 2023 logo and a prompt: [ec2-user@ip-172-31-35-255 ~]\$

## Maven & Java

Apache Maven is a build automation and project management tool used primarily for Java projects to manage dependencies, compile code, and package applications.

Maven is required in this project because it uses something called archetypes, which are like templates, to lay out the foundations for different types of projects e.g. web apps. I'll use Maven later on to set up all the necessary web files to create a web app structure.

Java is a popular programming language used to build different types of applications, from mobile apps to large enterprise systems.

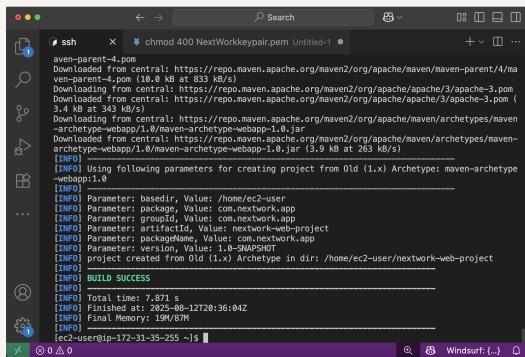
Java is required in this project because I am using a tool called Maven which depends on Java to operate

# 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 '

I installed Remote - SSH, which is a VS Code extension that lets me securely connect to another computer via SSH, so I can work on files and run programs on my EC2 instance directly from VS Code.

Configuration details required to set up a remote connection include the host, IdentityFile & User



A screenshot of a terminal window titled 'sshd'. The window shows the output of a Maven command:

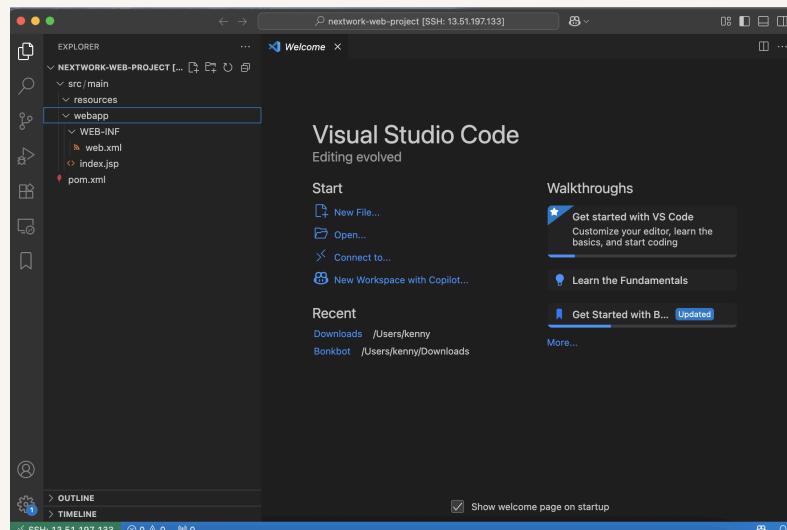
```
mvn archetype:generate \ -DgroupId=com.nextwork.app \ -DartifactId=nextwork-web-project \ -DarchetypeArtifactId=maven-archetype-webapp \ -DinteractiveMode=false
```

The terminal output shows the download of several Maven artifacts from central repositories, followed by the creation of a new Maven project named 'nextwork-web-project' in the current directory. The process is completed successfully.

# Create the Application

Using VS Code's file explorer, I could see All the files and subfolders the under nextwork-web-project which make up parts of my Web app.

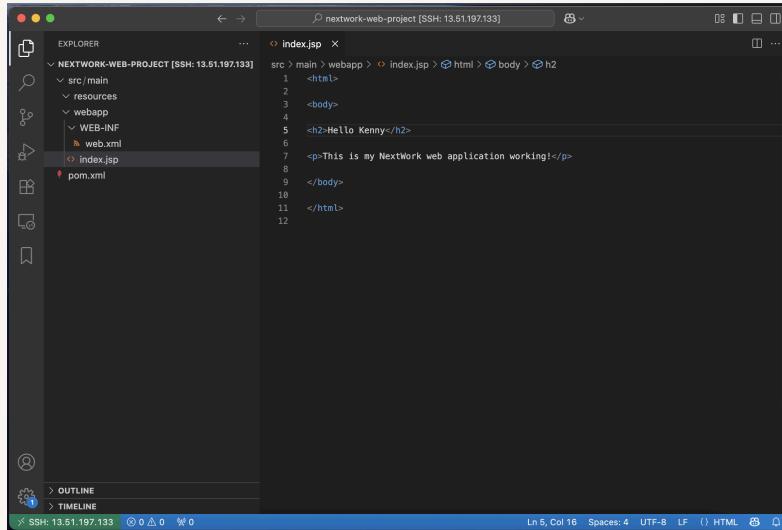
The src (source) folder contains all the source code files that define how your web app looks and works, including the webapp folder, which holds the web app's files like HTML, CSS, JavaScript, and JSP, and the resources folder, which includes configuration files such as database connection settings.



# Using Remote - SSH

The index.jsp is the main entry page of a Java web application that combines HTML with Java code to dynamically generate content for users when they visit the site or webapp

I edited index.jsp by clicking into it from the explorere tab in vscode and changing its HTML code

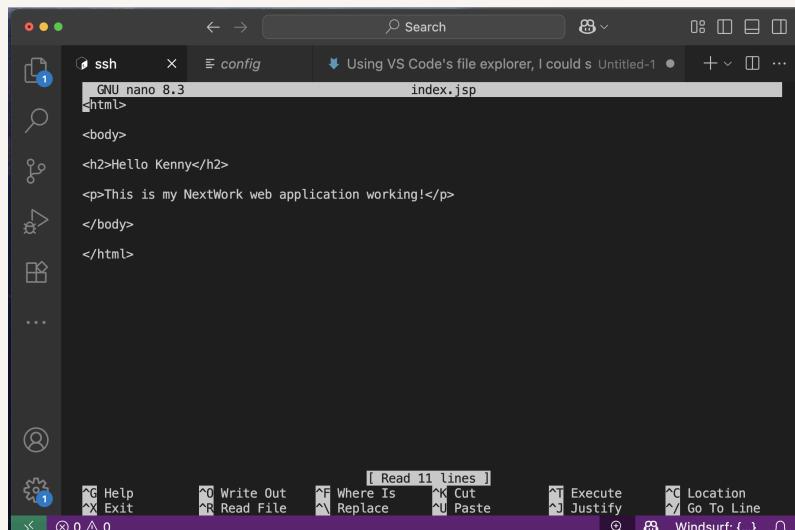


```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <title>NextWork Web Application</title>
    </head>
    <body>
        <h2>Hello Kenny</h2>
        <p>This is my NextWork web application working!</p>
    </body>
</html>
```

# Using nano

An alternative to using IDEs is the terminal; to edit index.jsp, I navigated to its directory with cd ~/nextwork-web-project/src/main/webapp and opened it for editing using the nano index.jsp command.

Compared to using an IDE, editing index.jsp in the terminal felt less convenient and harder to navigate, and I'd be more likely to use an IDE if I want features like syntax highlighting, code completion, and easier file management.



```
GNU nano 8.3
<html>
<body>
<h2>Hello Kenny</h2>
<p>This is my NextWork web application working!</p>
</body>
</html>
```



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

# The place to learn & showcase your skills

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

