



Set Up a Web App Using AWS and VS Code

J

Joseph Cole-Showers

```
↳ 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!

This project is day 1 of a 7 day DevOps Challenge. In this project, I'm going to set up the foundations of a CI/CD pipe by creating a webapp from scratch. I'll launch an EC2 instance and connect to the instance via VSCode to generate a webapp inside.

Key tools and concepts

Services I used were VS Code, Amazon EC2. Key concepts I learnt include, SSH connections using an IDE, launching an instance, editing index.jsp file, and using key pairs.

Project reflection

One thing I did not expect in this project today was using the terminal to edit a file.

This project took me approximately 2.30 hrs. It was most rewarding to see a successful SSH connection to the EC2 instance, both over terminal and the VS Code SSH connection.

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.



Launching an EC2 instance

I started this project by launching an EC2 instance because EC2 instances are like virtual computers that live in the cloud. I want the web app to live entirely in the cloud. So, I'm launching an EC2 instance to also develop the webapp's code.

I also enabled SSH

SSH is a protocol used to make sure only authorized users can access a remote server e.g EC2 Instances. I enabled SSH so that I can securely connect to the EC2 Instance, and it only allows SSH traffic from my ip address for security.

Key pairs

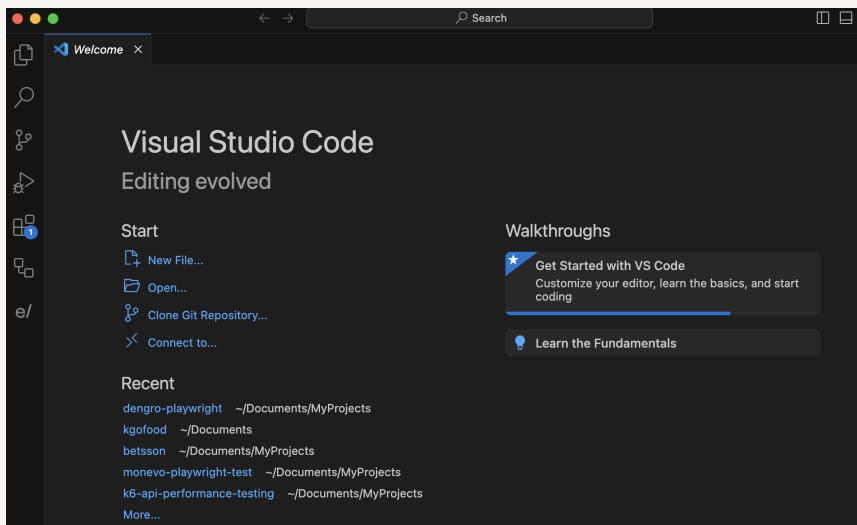
A key pair is a mechanism to connect and get access to EC2 instances launched in AWS. I created a key pair for an EC2 instance I'm launching. Key pairs work by having 2 halves. A public key AWS holds and a private key for me which AWS authenticates.

Once I set up my key pair, AWS automatically downloaded the private key file with a .pem extension. I moved this private key folder into a folder I created in my desktop.

Set up VS Code

VS Code is one of the most popular IDEs in the world. I'm using to write and edit code. It also has handy extensions that helps one to connect to an EC2 Instance.

I installed VS Code to write and edit the web apps code that will live inside the instance and it will also give me the ability to connect to an EC2 instance.



My first terminal commands

A terminal is where you send instructions to your computer using text instead of clicks. The first command I ran for this project is 'cd documents' which navigates the terminal to the documents folder.

I also updated my private key's permissions by running the command 'chmod 400 nextwork-keypair.pem'. This command gives myself access to use this file so I can use to connect to an EC2 instance.

```
~ % cd documents
documents % cd devops
devops % ls
nextwork-keypair.pem
devops % chmod 400 nextwork-keypair.pem
```

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command SSH -i path/to/private-key ec2-user@publicipv4 address.

This command required an IPv4 address

A server's IPV4 DNS is the public address that identifies where the server lives in the cloud i.e your EC2 server. The local computer you're using to do this project will find and connect to your EC2 instance through this IPV4 DNS.

```
,      #
~\_ #####
~~ \#####
~~ \|##|
~~ \|#/ ___
~~ \|~'-'->
~~|   / \
~~| / \ |
~~|/ \ |
[ec2-user@ip-172-31-25-237 ~]$
```



Maven & Java

Apache Maven is a tool that helps developers build and organize Java software projects.

Maven is required in this project because it's ability to spin-up web apps using archetypes.

Java is a programming language I am using to develop the web app.

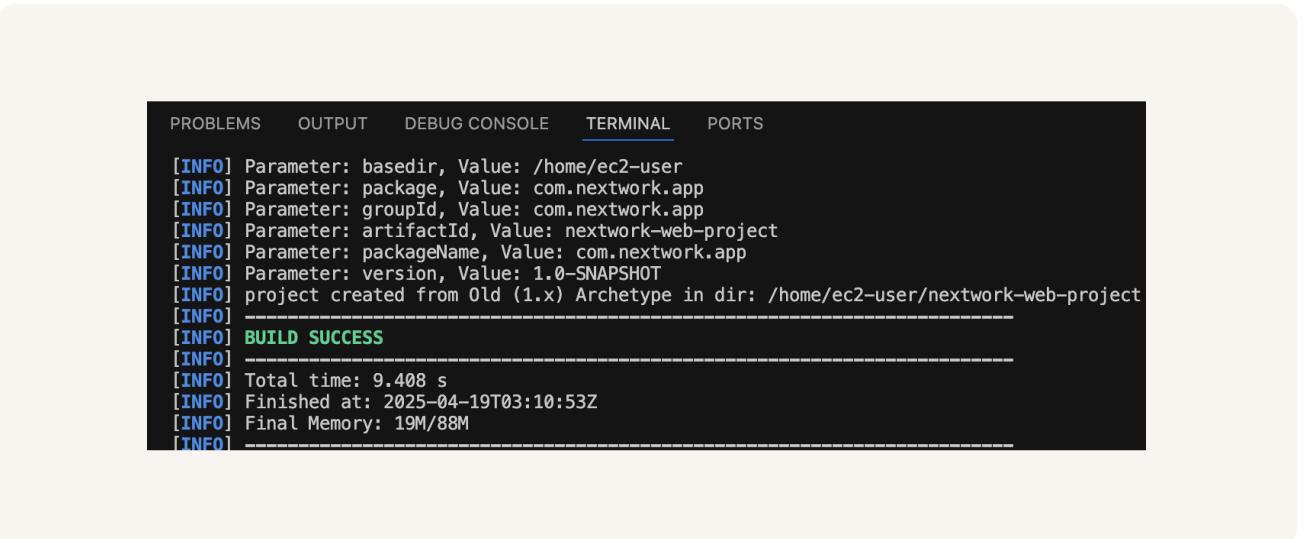
Java is required in this project because it lays the foundation for writing the web app code. Maven also needs Java in order to work.

Create the Application

I generated a Java web app using the command 'mvn archetype:generate'. This command tells Maven to generate a web app using an existing template that it has.

I installed Remote - SSH, which is an extension within VS Code. I installed it to connect VS Code to a remote server (here EC2 Instance).

Configuration details required to set up a remote connection include the Host (EC2 instance address), identityFile (location of private key) and the User.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

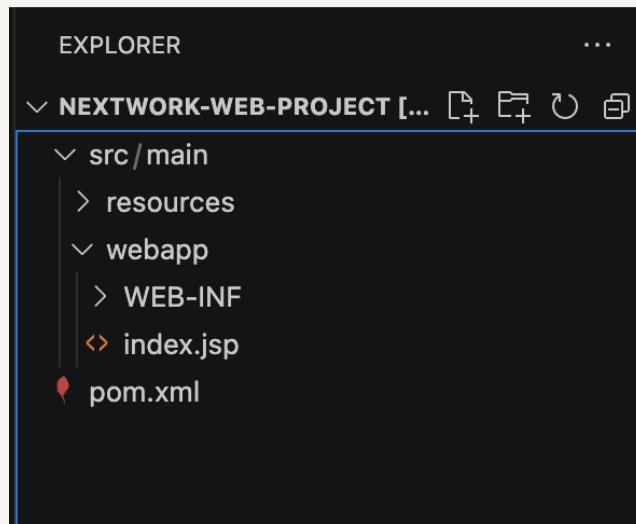
[INFO] Parameter: basedir, Value: /home/ec2-user
[INFO] Parameter: package, Value: com.nextwork.app
[INFO] Parameter: groupId, Value: com.nextwork.app
[INFO] Parameter: artifactId, Value: nextwork-web-project
[INFO] Parameter: packageName, Value: com.nextwork.app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/ec2-user/nextwork-web-project
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.408 s
[INFO] Finished at: 2025-04-19T03:10:53Z
[INFO] Final Memory: 19M/88M
[INFO] -----
```



Create the Application

Using VS Code's file explorer, I could see the projects folder and files.

Two of the project folders created by Maven are src and webapp, which can be expanded.



Using Remote - SSH

index.jsp is the file in our web app that defines html content and the static elements that goes into the web page.

I edited index.jsp by updating the HTML code.

```
↳ 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  |
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

