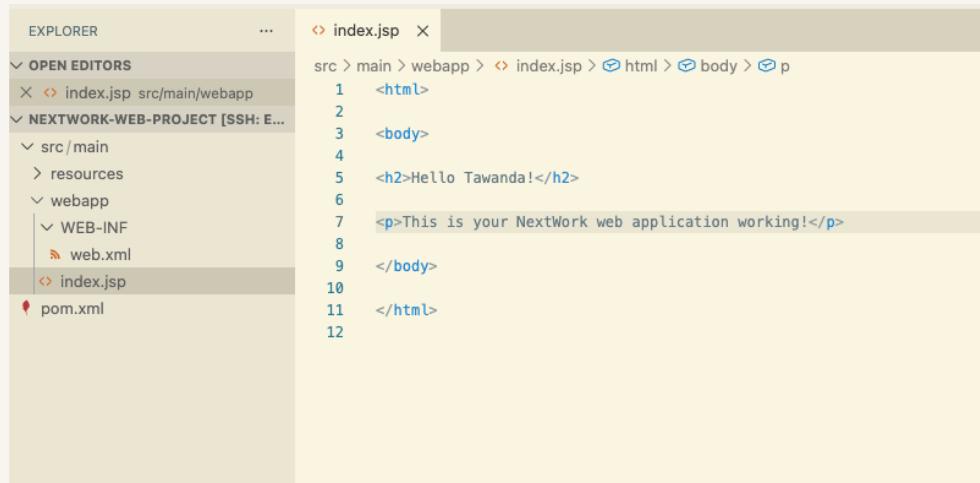




# Set Up a Web App Using AWS and VS Code



Tawanda Nigel Chitapi



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



**Tawanda Nigel Chitapi**

NextWork Student

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

# Introducing Today's Project!

In this project, I will demonstrate how to setup the foundations of a CI/CD pipeline by creating a web app from scratch. I will launch an EC2 instance and connect to the instance via VS Code to generate a web app inside.

This project is part one of a series of DevOps projects where I'm building a CI/CD pipeline!

I did this project because i wanted to understand the fundamentals of connecting to remote servers using SSH

## Key tools and concepts

Services I used were AWS EC2, VS Code, Terminal. Key concepts I learnt include connecting both my terminal (local computer) and VS Code IDE to a remote EC2 instance, I learnt how to create and organize SSH keypairs.

## Project reflection

This project took me approximately 1hour.

One thing I didn't expect in this project was the ability to edit code using terminal instead of an IDE.

**Tawanda Nigel Chitapi**

NextWork Student

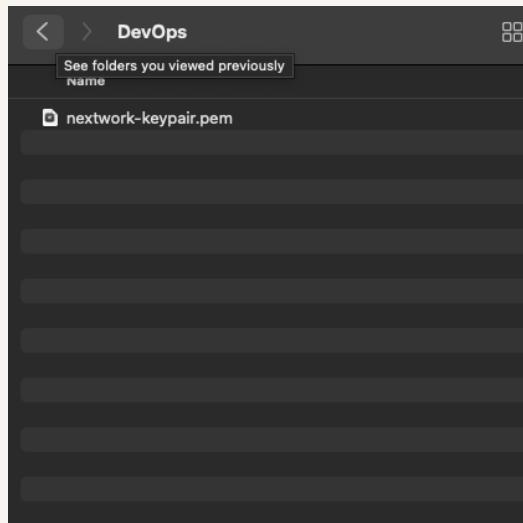
[nextwork.org](http://nextwork.org)

# Launching an EC2 instance

## What I did in this step

In this step, I will create and launch an EC2 Instance.

I started this project by launching an EC2 instance because EC2 instances are virtual computers that live in the cloud. I need the web app to live entirely in the cloud so I am launching an EC2 instance for the development of the webapp as well.





**Tawanda Nigel Chitapi**

NextWork Student

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

## I also enabled SSH

SSH is a protocol that authorizes user to access servers like EC2 instances remotely. Its also a type of traffic that enable to me to transfer data back and forth to the EC2 instance. For security purposes I have only allowed the EC2 instance to allow SSH traffic from my IP Address only and no where else.

## Key pairs

A key pair is a mechanism that enables me to gain access into an EC2 instance that i launch on AWS. I have created a keypair for the EC2 instance I am launching. Keypairs a literally a pair of keys, two halves. A public key that AWS keeps and a private key that i download. AWS will authenticate my private key by matching it to the public key that it keeps on its end.

## Downloaded key pair file

Once I set up my key pair, AWS automatically downloaded the private .pem key file for safe keeping.



**Tawanda Nigel Chitapi**

NextWork Student

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

## Set up VS Code

### What I did in this step

In this step, I will install VS Code, a tool that enables me to write and edit code. I will also set up a terminal and update the permission settings of my key.

### What is VS Code?

VS Code is an IDE (Integrated Development Environment) used to write and edit code. Developers use it to manage coding projects.

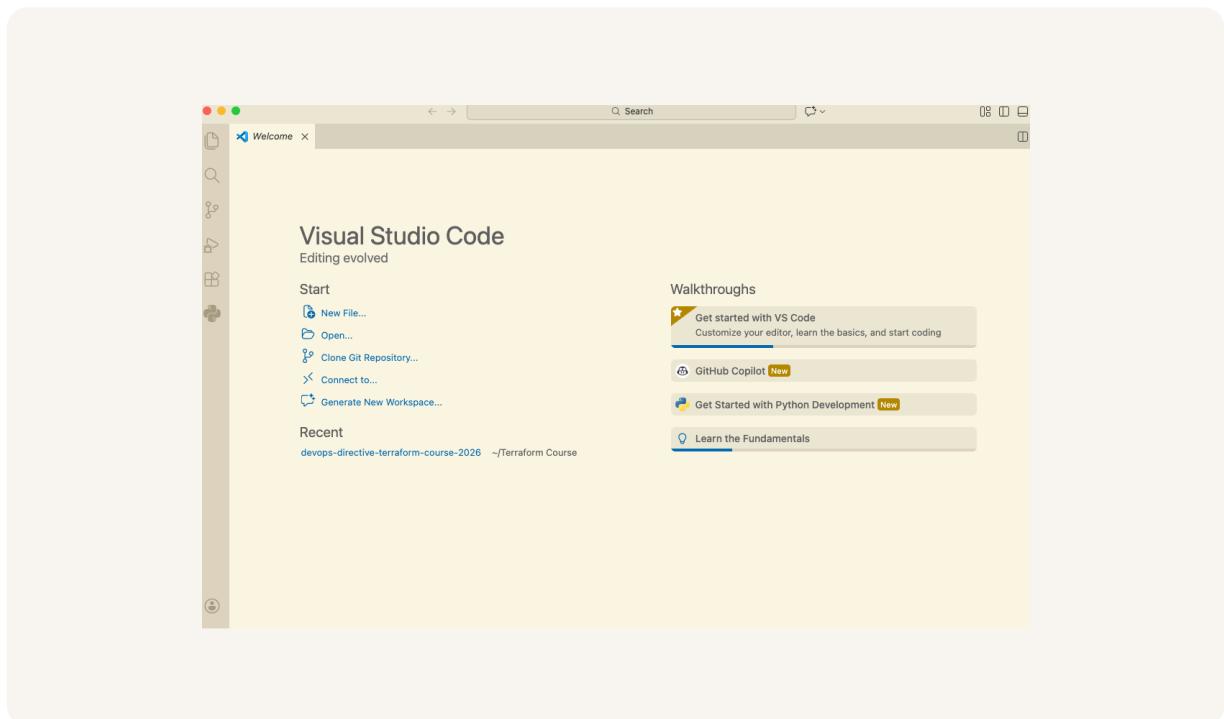
I installed VS Code to write and edit the webapp code. It also has extensions that allow and enable me to directly connect to an EC2 instance.



**Tawanda Nigel Chitapi**

NextWork Student

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



**Tawanda Nigel Chitapi**

NextWork Student

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

# My first terminal commands

A terminal is an environment that allows me to send commands to my computer. The difference between the terminal and the click drag interface is that we send text based commands in the terminal. The first command I ran for this project is cd desktop/devops which navigates to the DevOps folder on my desktop that I saved the keypair.

## Updating file permissions

I also updated my private key's permissions by running the command chmod 400 command. This command gives access to the .pem file so that I can use it to connect to the EC2 instance.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
● tawandanigelchitapi@Tawandas-MBP ~ % cd desktop/devops
● tawandanigelchitapi@Tawandas-MBP devops % ls
nextwork-keypair.pem
● tawandanigelchitapi@Tawandas-MBP devops % chmod 400 nextwork-keypair.pem
zsh: command not found: chmod
● tawandanigelchitapi@Tawandas-MBP devops % chmod 400 nextwork-keypair.pem
● tawandanigelchitapi@Tawandas-MBP devops % chmod 400 nextwork-keypair.pem
● tawandanigelchitapi@Tawandas-MBP devops % █
```



**Tawanda Nigel Chitapi**

NextWork Student

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

## SSH connection to EC2 instance

### What I did in this step

In this step, I will connect to the EC2 instance via the terminal. Once connected i can start setting up the instance inside the terminal.

### Connecting to EC2

To connect to my EC2 instance, I ran the command `ssh -i path to .pem key file ec2-user@IPV4 address`. this command sets up a direct connect between my local computer to the ec2 instance.

### This command required an IPv4 address

A server's IPV4 DNS is the public address that identifies where the server lives on the cloud. This is useful information to give to my local computer. It tells my local computer where to find the EC2 instance that I launched.

**Tawanda Nigel Chitapi**

NextWork Student

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

```
'~\_\ ####_          Amazon Linux 2023
~~\####\
~~\###|
~~ \#/ __> https://aws.amazon.com/linux/amazon-linux-2023
~~ \~'__>
~~*_*_/
~/m/'[ec2-user@ip-172-31-17-185 ~]$ 
```

**Tawanda Nigel Chitapi**

NextWork Student

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

# Maven & Java

## What I did in this step

In this step, I will instal MAven and Java on the EC2 instance because these are important in setting a java webapp from scratch.

## Why I'm using Maven

Apache Maven is a tool that helps with managing Java projects. It has a lot of use cases like package manager. It is also a tool the uses archetypes which are like templates for spinning up webprojects.

Maven is required in this project because its needed to use its ability to spin up webapps using archetypes. I will setup the Java webapp using one of these archetypes.

## Why I'm using Java

Java is a popular programming language used to build different types of applications, from mobile apps to large enterprise systems. I will be using it to develop the webapp.



**Tawanda Nigel Chitapi**

NextWork Student

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

Java is required in this project because it lays the foundation for the webapp i am building. I will be using Maven archetypes to build the webapp. Maven needs Java in order to function.



**Tawanda Nigel Chitapi**

NextWork Student

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

# Create the Application

## What I did in this step

In this step, I will build the webapp using Maven and Java.

## Creating the Java web app

I generated a Java web app using the command mvn archetype:generate \. This command tells Maven to generate a webapp using an existing template that it has. It also tells Maven to call the generate project webapp named nextwork-web-project

## Installing Remote - SSH

I installed Remote - SSH, which is an extension within VS Code. This extension enables me to directly connect to an ssh remote server i.e my EC2 instance. I installed it to...

## SSH configuration details

Configuration details required to set up a remote connection include the host i.e the EC2 instance address, the identity file i.e the location of my private key and the user.



**Tawanda Nigel Chitapi**

NextWork Student

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

```
[INFO] -----  
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0  
[INFO] -----  
[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] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 11.013 s  
[INFO] Finished at: 2026-02-01T10:38:49Z  
[INFO] Final Memory: 19M/89M  
[INFO] -----  
[ec2-user@ip-172-31-17-185 ~]$ █
```

**Tawanda Nigel Chitapi**

NextWork Student

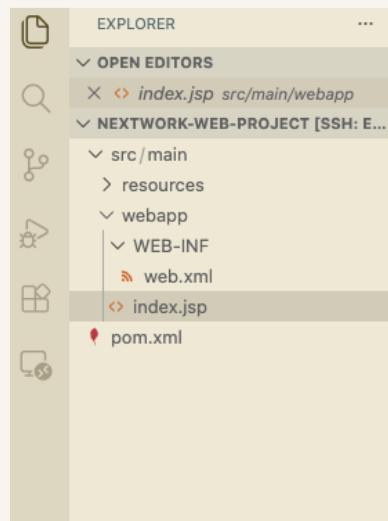
[nextwork.org](http://nextwork.org)

# Create the Application

## Exploring the project structure

Using VS Code's file explorer, I could see a bunch of folders and sub folders that define our webapp.

Two of the project folders created by Maven are resource and webapp, which organize different parts of the webapp. The Resources folders stores the connection configurations and the webapp folders stores webapp files for the look and feel of the webapp.





**Tawanda Nigel Chitapi**

NextWork Student

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

# Using Remote - SSH

## What I did in this step

In this step, I will connect VS Code with the EC2 instance. This is different from connect our local computer (terminal) to the EC2 instance.

## Updating the web app

The index.jsp is file in our webapp that define both HTML content i.e the static web content as well as any code for generate dynamic content i.e content that is always changing.

I edited index.jsp by editing the webapp header to include my name as well as add a paragraph that says " This is your NextWork web application working! "

# Tawanda Nigel Chitapi

NextWork Student

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

The screenshot shows a code editor interface with a sidebar labeled "EXPLORER" and a main editor area. The sidebar lists project files: "OPEN EDITORS" (index.jsp), "NEXTWORK-WEB-PROJECT [SSH: E...]" (src/main/resources/webapp/WEB-INF/web.xml, index.jsp), and "pom.xml". The main editor area is titled "index.jsp" and contains the following JavaServer Pages (JSP) code:

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



**Tawanda Nigel Chitapi**

NextWork Student

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

# Using nano

## Additional improvements

In this secret mission, I will edit code without an IDE because this is going to teach me how IDE's like VS Code are loved and make work so much easier.

## Terminal vs IDE

An alternative to using IDEs is within the terminal itself. To edit index.jsp, I ran cd commands to navigate through the folders in terminal and use the nano command to edit the index.jsp file.

Compared to using an IDE, editing index.jsp in the terminal felt a little less intuitive. I'd be more likely to use an IDE if I had lot of bit of code to update and delete i.e lots of editing within the code.

## Verifying my work

To verify my editing work in the terminal, I opened the VS Code window that has the ssh connection to the EC2 instance. It was possible to see my changes in VS Code right away because the window shared the same SSH connection to the instance as the terminal.



**Tawanda Nigel Chitapi**

NextWork Student

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

```
GNU nano 8.3                               index.jsp
<html>
<body>
<h2>Hello Tawanda!</h2>
<p>This is your NextWork web application working!</p>
</body>
</html>

[ Read 11 lines ]
^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^A Replace   ^U Paste    ^J Justify  ^L Go To Line
Ln 1, Col 1  Spaces: 2  UTF-8  LF  {} SSH Config  Finish Setup  ↵
```



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

# The place to learn & showcase your skills

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

