



Set Up a Web App in the Cloud

G Gloria

The screenshot shows a code editor interface with two panes. The left pane is the Explorer view, showing a project structure for 'NEXTWORK-WEB-PROJECT [SSH: EC2-INSTA...]' with the following files:

- src/main
- resources
- webapp
 - WEB-INF
 - web.xml
- index.jsp
- target
- pom.xml

The right pane is the code editor for 'index.jsp', showing the following HTML code:

```
<html>
<body>
<h2>Hello Gloria</h2>
<p>This is my NextWork web application working!</p>
</body>
</html>
```



Introducing Today's Project!

What is VSCode and why is it useful?

I used VSCode to connect to my EC2 instance, navigate project files, and edit the web app. Its extensions, like Remote - SSH, made it easy to manage the server, while its interface allowed me to efficiently modify and update key files like index.jsp.

How I'm using VSCode in this project

I used VSCode to connect to my EC2 instance, navigate project files, and edit the web app. Its extensions, like Remote - SSH, made it easy to manage the server, while its interface allowed me to efficiently modify and update key files like index.jsp.

One thing I didn't expect...

One thing I didn't expect in this project was how smoothly VSCode's Remote - SSH extension made connecting to and managing the EC2 instance. It simplified file navigation and editing, making the development process much more efficient than anticipated.

This project took me...

This project took me about 30 minutes to complete. The clear steps, tools like VSCode, and streamlined setup process made it efficient to set up the EC2 instance, install necessary tools, and build and edit the web app.



Launching an EC2 instance

I started this project by launching an EC2 instance because it provides a virtual server to develop, test, and run my web app. It's flexible, scalable, and help to customize resources like CPU, memory, and storage to meet my project's needs.

I also enabled SSH

SSH is a secure protocol that allows remote access to servers over the internet. I enabled SSH so that I can safely connect to my EC2 instance, manage files, and execute commands as if I were directly using the server.

Key pairs

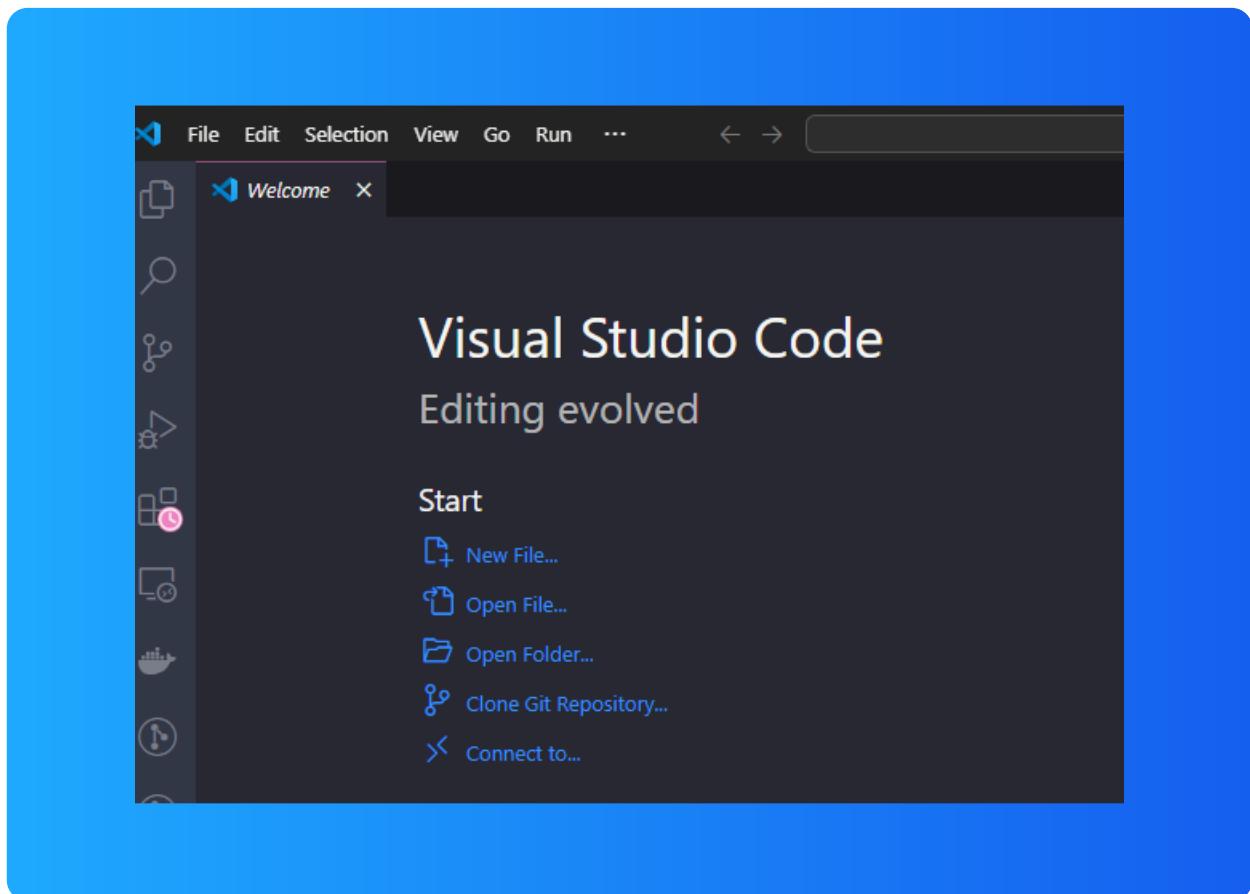
A key pair is a set of security credentials consisting of a public key and a private key. It is used to securely connect to an EC2 instance, ensuring that only authorized users with the private key can access the server.

Once I set up my key pair, AWS automatically downloaded a private key file (.pem) to my local computer. This file is essential for securely connecting to my EC2 instance and must be kept safe to prevent unauthorized access.

Set up VSCode

VSCode is a lightweight and powerful code editor by Microsoft. It supports multiple programming languages and tools, making it ideal for development. I'm using it to connect to my EC2 instance to work with it.

I installed VSCode to manage and edit my coding projects efficiently. I'll use it to write code, connect to my EC2 instance via SSH, and execute commands remotely. Its tools and features make it ideal for developing, debugging, and managing projects.



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 ~/Desktop/DevOps`, which navigates the terminal to the DevOps folder where my `.pem` file is stored for connecting to EC2

I also updated my private key's permissions by using the command `'chmod 400 nextwork-keypair.pem'`. This ensures the file is only readable by me, which is necessary for securely connecting to my EC2 instance.

```
→ DEVOPS ls
nextwork-keypair.pem
→ DEVOPS chmod 400 nextwork-keypair.pem
→ DEVOPS █
```



SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command `ssh -i "nextwork-keypair.pem" ec2-user@ec2-3-91-216-190.compute-1.amazonaws.com`. This used my private key to establish a secure SSH connection to the server, allowing me to manage it remotely.

This command required an IPv4 address

A server's IPV4 DNS is its unique address that allows users or devices to locate and connect to it over the internet. It acts like a website address, making it easier to access the server without needing to remember its IP address.

```
➔ DEVOPS chmod 400 nextwork-keypair.pem
➔ DEVOPS ssh -i "nextwork-keypair.pem" ec2-user@ec2-3-91-216-190.compute-1.amazonaws.com
The authenticity of host 'ec2-3-91-216-190.compute-1.amazonaws.com (3.91.216.190)' can't be established.
ED25519 key fingerprint is SHA256:NbjJ8dzhRJ4yAXnkoFbYJX2lTXnzCJRC6QnxshbyCpU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-91-216-190.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
,      #
~\_ #####      Amazon Linux 2023
~~ \####\_
~~ \|##|
~~ \|#/   _--> https://aws.amazon.com/linux/amazon-linux-2023
~~|_/
~~|/_/
~~|/_/
/_m/ ._
[ec2-user@ip-172-31-18-36 ~]$ █
```



Maven & Java

Apache Maven is a build automation tool used for managing Java projects. It simplifies tasks like compiling code, running tests, and managing project dependencies, making it easier to develop and maintain Java-based applications efficiently.

Maven is required in this project because it simplifies building and organizing Java projects. It manages dependencies automatically and uses templates called archetypes to create a structured web app foundation, which makes the whole process faster

Java is a programming language used to create apps that can run on different devices. It's popular for making websites, mobile apps, and other software because it's reliable, flexible, and works on many platforms.

Java is required in this project because Maven, the tool we use to build and manage our web app, needs Java to work. Java also provides the platform to run the web app we're creating, making it essential for both building and operating the app.

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`. This created a template with all the basic files.

I installed Remote - SSH, which is a VSCode extension that allows direct access to remote servers. I installed it to connect VSCode to my EC2 instance, so I can easily navigate, edit, and manage my web app files directly on the server.

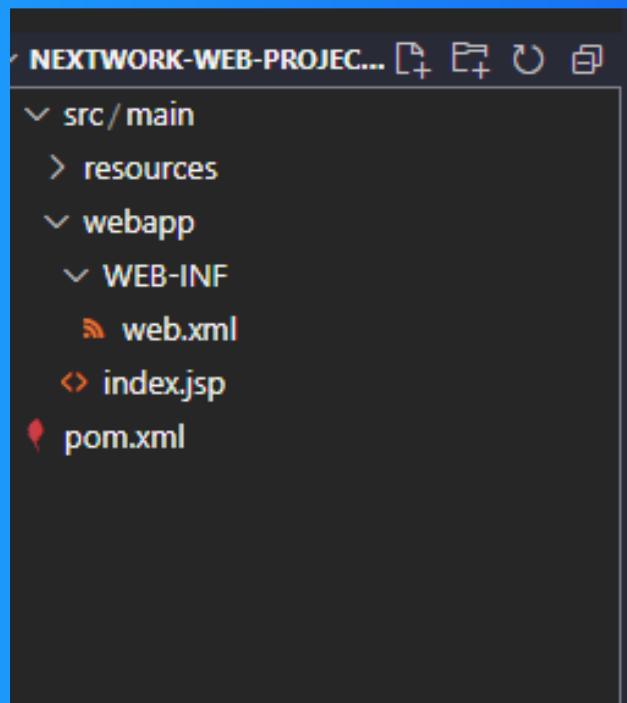
Configuration details for a remote connection include the Host (IPv4 DNS of the EC2 instance), IdentityFile (path to the private key, i.e nextwork-keypair.pem, and User (ec2-user). These ensure secure and proper SSH access to manage the server.

```
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: 8.765 s
[INFO] Finished at: 2024-12-23T20:20:51Z
[INFO] Final Memory: 16M/93M
[INFO] -----
[ec2-user@ip-172-31-18-36 ~]$ █
```

Create the Application

Using VSCode's file explorer, I could see the project structure, including the src/main folder, the webapp folder with WEB-INF containing web.xml and index.jsp, and the pom.xml file. These are the key files for managing and building the web app.

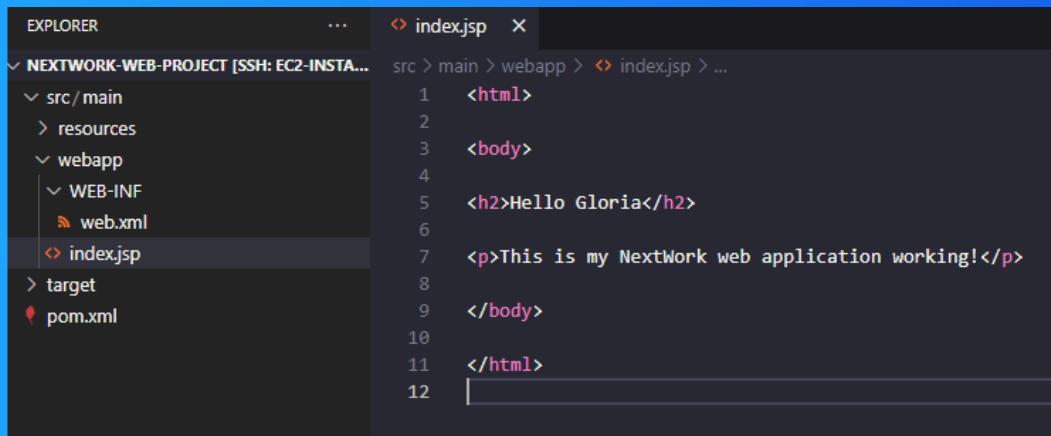
Two of the project folders created by Maven are src and webapp, which organize the project's code and resources. The src folder contains the main code and resources, while the webapp folder holds files like index.jsp and web.xml.



Using Remote - SSH

index.jsp is the main web page of the Java web app. It is a JSP (Java Server Pages) file that combines HTML and Java code, allowing dynamic content to be generated and displayed in the browser. It serves as the starting point for the web application.

I edited index.jsp by opening it in VSCode, making changes to its HTML and Java content, and saving the file. This allowed me to update the web app's main page with custom text, layout, or features to meet project requirements.



```
EXPLORER      ...      index.jsp  X
NEXNETWORK-WEB-PROJECT [SSH: EC2-INSTA...
src/main
  resources
  webapp
    WEB-INF
      web.xml
  index.jsp
target
pom.xml

src > main > webapp > index.jsp > ...
1  <html>
2
3  <body>
4
5  <h2>Hello Gloria</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

