

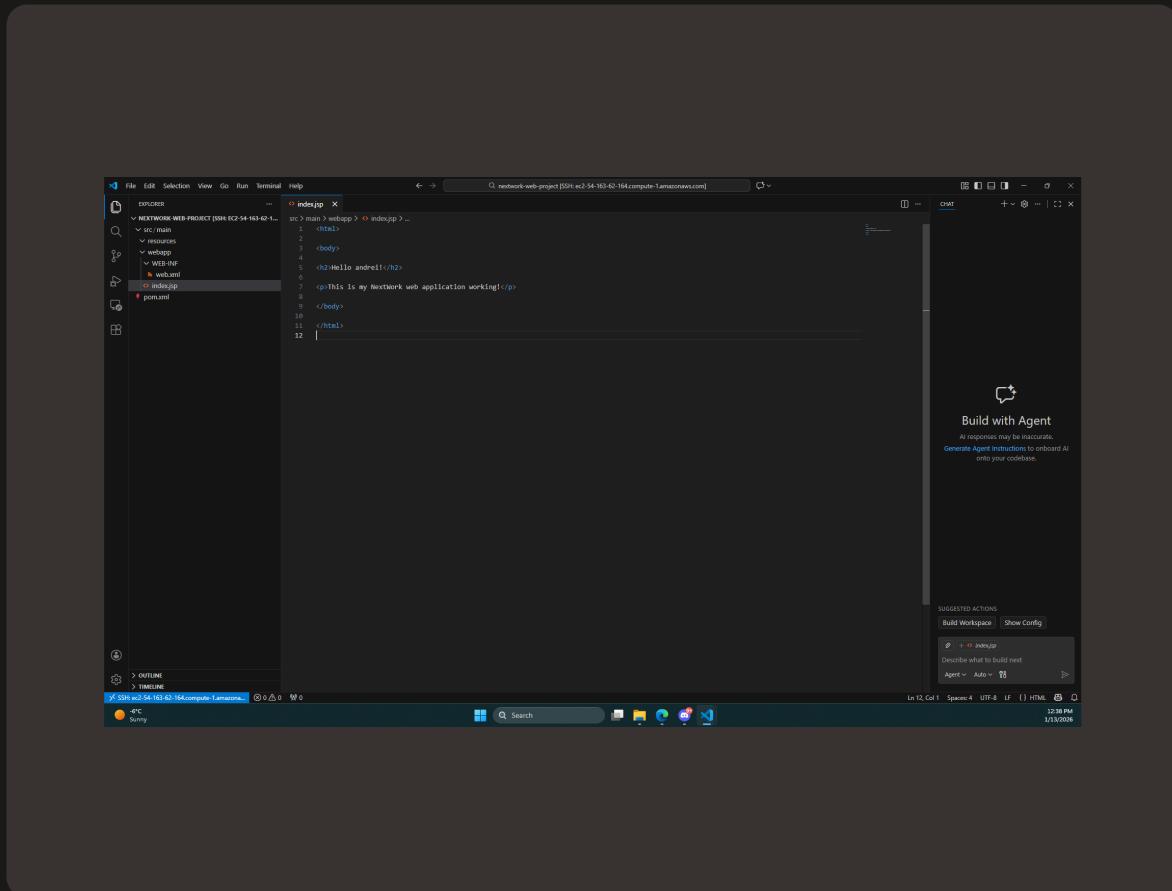


nextwork.org

Set Up a Web App Using AWS and VS Code

A

Andrei Boboc





Andrei Boboc
NextWork Student

nextwork.org

Introducing Today's Project!

In this project, I will demonstrate how to deploy a simple Java application on AWS. I am doing this project to learn how to set up IAM users, launch and connect to an EC2 instance, configure a development environment with VS Code and Maven, and build and run an application in the cloud.

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.

I did this project today to gain hands-on experience with cloud infrastructure and remote development workflows. It helped me practice launching and managing an EC2 instance, working with SSH, and building a Java web application in a real server environment.

Key tools and concepts

Services I used were Amazon EC2, SSH, and VS Code Remote - SSH. Key concepts I learnt include launching and connecting to an EC2 instance, using key pairs for secure access, managing file permissions, connecting to a remote server via SSH, generating a Java web application with Maven, and editing application files both through an IDE and directly from the terminal.



Andrei Boboc

NextWork Student

nextwork.org

Project reflection

This project took me approximately 3 hours. The most challenging part was troubleshooting SSH and VS Code Remote-SSH connection issues after reconnecting to the EC2 instance. It was most rewarding to successfully set up the remote development environment and see my Java web application generated and edited directly on the EC2 instance.

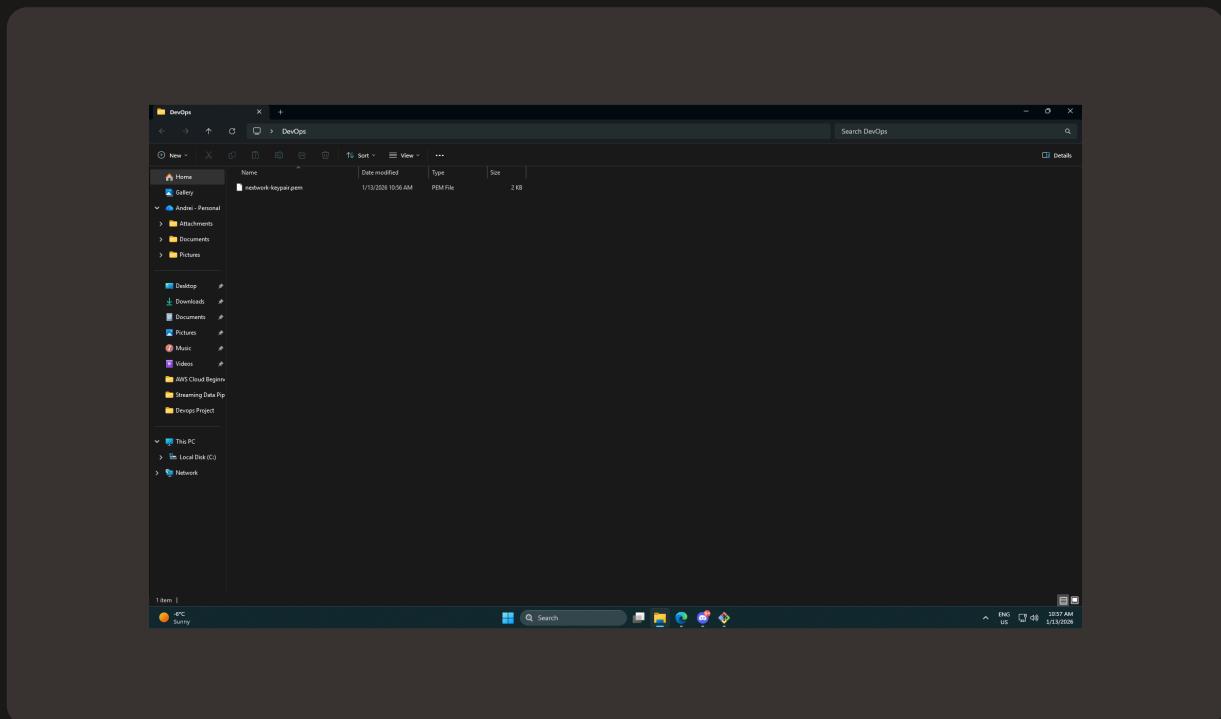
One thing I didn't expect in this project was when closing the SSH remote connection abruptly, I could not reconnect to the instance, even from the AWS console. I later learned this was caused by local SSH configuration and VS Code Remote-SSH state issues, not a problem with the EC2 instance itself. Fixing the SSH config and key permissions resolved the issue.

Launching an EC2 instance

What I did in this step

In this step, I will launch a new EC2 instance, configure a key pair, and set up the required network settings because these are necessary to securely access and run applications on a virtual server in AWS.

I started this project by launching an EC2 instance because it provides a virtual server where I can deploy, configure, and run my application in a real cloud environment.





Andrei Boboc

NextWork Student

nextwork.org

I also enabled SSH

SSH (Secure Shell) is a secure network protocol used to remotely access and manage a server. I enabled SSH so that I can securely connect to my EC2 instance and run commands from my local machine.

Key pairs

A key pair is a set of security credentials made up of a public key and a private key that are used to securely connect to an EC2 instance without using a password.

Downloaded key pair file

Once I set up my key pair, AWS automatically downloaded a .pem private key file to my local computer, which is used to securely connect to the EC2 instance via SSH.

A**Andrei Boboc**

NextWork Student

nextwork.org

Set up VS Code

What I did in this step

In this step, I will install Visual Studio Code because it will be used as my code editor to write, edit, and manage application files when working with my EC2 instance.

What is VS Code?

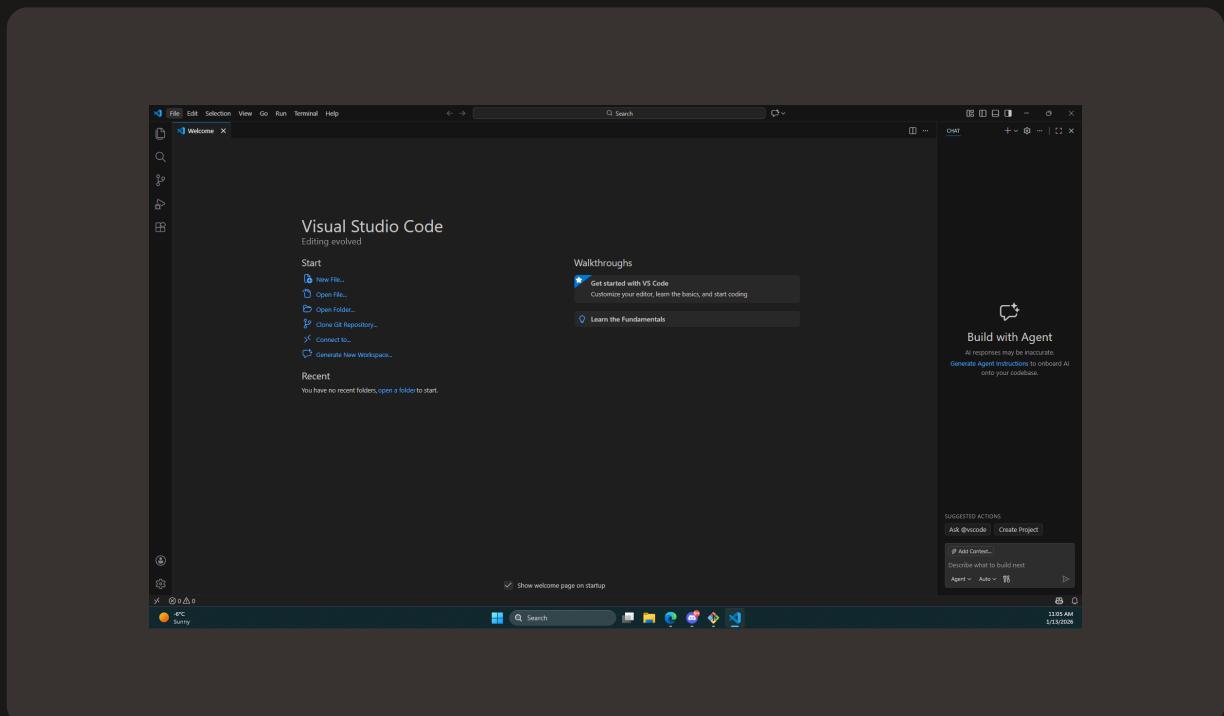
VS Code is a lightweight but powerful source-code editor developed by Microsoft that supports many programming languages and provides features like debugging, extensions, and Git integration.

I installed VS Code to write, edit, and manage my application code and to connect to my EC2 instance for remote development.

A

Andrei Boboc
NextWork Student

nextwork.org



My first terminal commands

The first command I ran in the terminal was cd to navigate to the directory where my .pem key file is stored so I could use it to connect to the EC2 instance.

Updating file permissions

I updated my private key's permissions by using the icacls command to remove inherited permissions and grant read access only to my user, ensuring the .pem file is secure.

```
PS C:\Users\andre\Desktop\Devops> icacls "nextwork-keypair.pem" /reset
>> icacls "nextwork-keypair.pem" /grant:r "andre:R"
>> icacls "nextwork-keypair.pem" /inheritance:r
>>
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
PS C:\Users\andre\Desktop\Devops> █
```



Andrei Boboc

NextWork Student

nextwork.org

SSH connection to EC2 instance

What I did in this step

In this step, I will connect to my EC2 instance using SSH because this allows me to securely access the server and begin configuring and deploying my application.

Connecting to EC2

To connect to my EC2 instance, I ran the SSH command using my private key file and the instance's IPv4 DNS name: ssh -i

```
C:\Users\andre\Desktop\DevOps\nextwork-keypair.pem ec2-user@ec2-107-23-65-250.compute-1.amazonaws.com
```

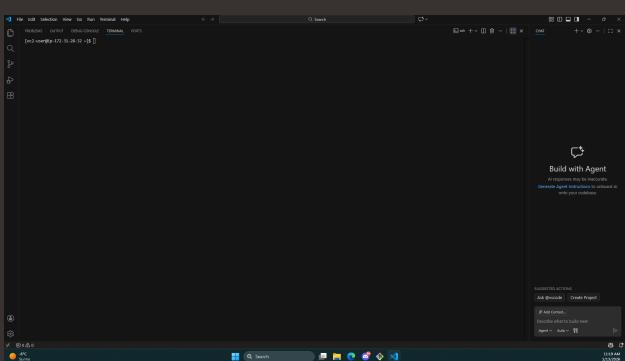
This command required an IPv4 address

A server's IPv4 DNS is a public hostname that maps to the server's IPv4 address and is used to connect to the server without needing to remember the numeric IP address.

A

Andrei Boboc
NextWork Student

nextwork.org



Maven & Java

What I did in this step

In this step, I will install Apache Maven and Amazon Corretto 8 because they are required to build and run my Java application on the EC2 instance.

Why I'm using Maven

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

Maven is required in this project because it is used to manage dependencies and build the Java application in a consistent and automated way.

Why I'm using Java

Java is a programming language and computing platform used to build applications that can run on many different operating systems.



A

Andrei Boboc

NextWork Student

nextwork.org

Java is required in this project because the application is written in Java and needs the Java runtime to be built and executed.



Andrei Boboc

NextWork Student

nextwork.org

Create the Application

What I did in this step

In this step, I will run Maven commands to generate a Java web application because Maven helps create the project structure and required files automatically.

Creating the Java web app

I generated a Java web app using the Maven archetype command: mvn archetype:generate \ -DgroupId=com.nextwork.app \ -DartifactId=nextwork-web-project \ -DarchetypeArtifactId=maven-archetype-webapp \ -DinteractiveMode=false mvn archetype:generate \ Runs Maven (mvn) and tells it to use the archetype plugin to generate a new project from a template. -DartifactId=nextwork-web-project \ Sets the artifactId, which is basically the project name. -DarchetypeArtifactId=maven-archetype-webapp \ Chooses the template (archetype) to use. maven-archetype-webapp generates a basic Java web application structure (WAR-style project). -DinteractiveMode=false Disables prompts/questions during generation.

Installing Remote - SSH

I installed Remote - SSH, which is a VS Code extension that allows remote connections over SSH. I installed it to connect VS Code directly to my EC2 instance so I can edit and manage files using IDE features.

SSH configuration details

Configuration details required to set up a remote connection include the EC2 instance's hostname (IPv4 DNS), the SSH user, and the path to the private key file.



Andrei Boboc

NextWork Student

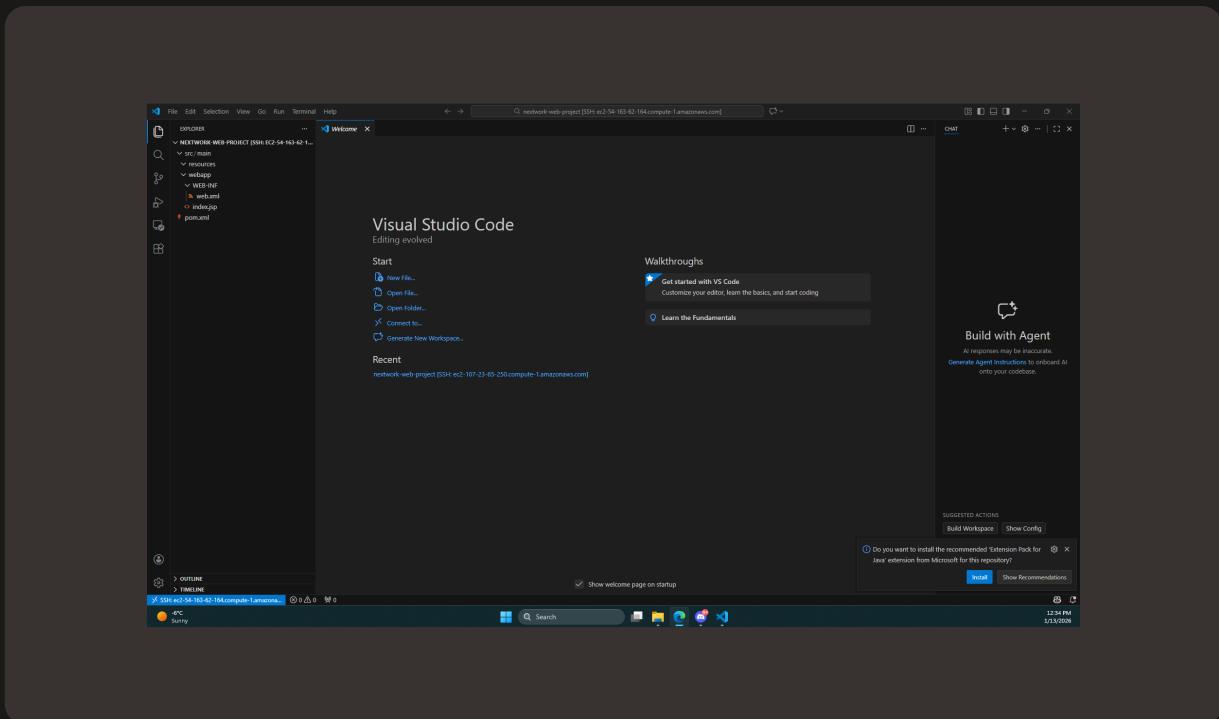
nextwork.org

Create the Application

Exploring the project structure

Using VS Code's file explorer, I could see the project's folder structure, including the src directory, the webapp folder, and files such as index.jsp, web.xml, and pom.xml on the EC2 instance.

Two of the project folders created by Maven are src and webapp, which store the application's source code and web resources. The src folder contains the project's source files, while the webapp folder contains web content such as JSP files and configuration needed for the web application.



A**Andrei Boboc**

NextWork Student

nextwork.org

Using Remote - SSH

What I did in this step

In this step, I will connect VS Code to my EC2 instance using SSH because this allows me to use VS Code's IDE features to edit and manage my application files directly on the server.

Updating the web app

The index.jsp is the main page of the Java web application that is displayed when users access the app, and it can contain both HTML and Java code.

I edited index.jsp by opening it in VS Code through the remote SSH connection and updating the file's content directly on the EC2 instance.

A

Andrei Boboc

NextWork Student

nextwork.org

The screenshot shows a terminal window titled "SSH ec2-54-162-62-164.compute-1.amazonaws.com". The window displays a file browser and a code editor. The file browser shows a directory structure for a "NEXTWORK WEB PROJECT" with subfolders like "src/main/resources", "webapp", and "src/main/webapp". The code editor shows the content of the "index.jsp" file:

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
<title>Hello andrei!</title>
</head>
<body>
<p>This is my NextWork web application working!</p>
</body>
</html>
```

Below the code editor, there is a "Build with Agent" interface with a message: "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." A "SUGGESTED ACTIONS" section includes a "Build Workspace" button and a "Show Config" link. The bottom status bar indicates "Line 12, Col 1" and "12:38 PM 1/13/2020".



Andrei Boboc
NextWork Student

nextwork.org

Using nano

Additional improvements

In this secret mission, I will edit the index.jsp file using the terminal instead of an IDE because it helps me practice managing and modifying application files directly from the command line.

Terminal vs IDE

An alternative to using IDEs is editing files directly from the terminal. To edit index.jsp, I ran the command nano index.jsp.

Compared to using an IDE, editing code in the terminal felt more manual and less convenient, but it was useful for making quick changes directly on the server. I'd be more likely to use an IDE for larger projects because it provides features like syntax highlighting, file navigation, and debugging tools.

Verifying my work

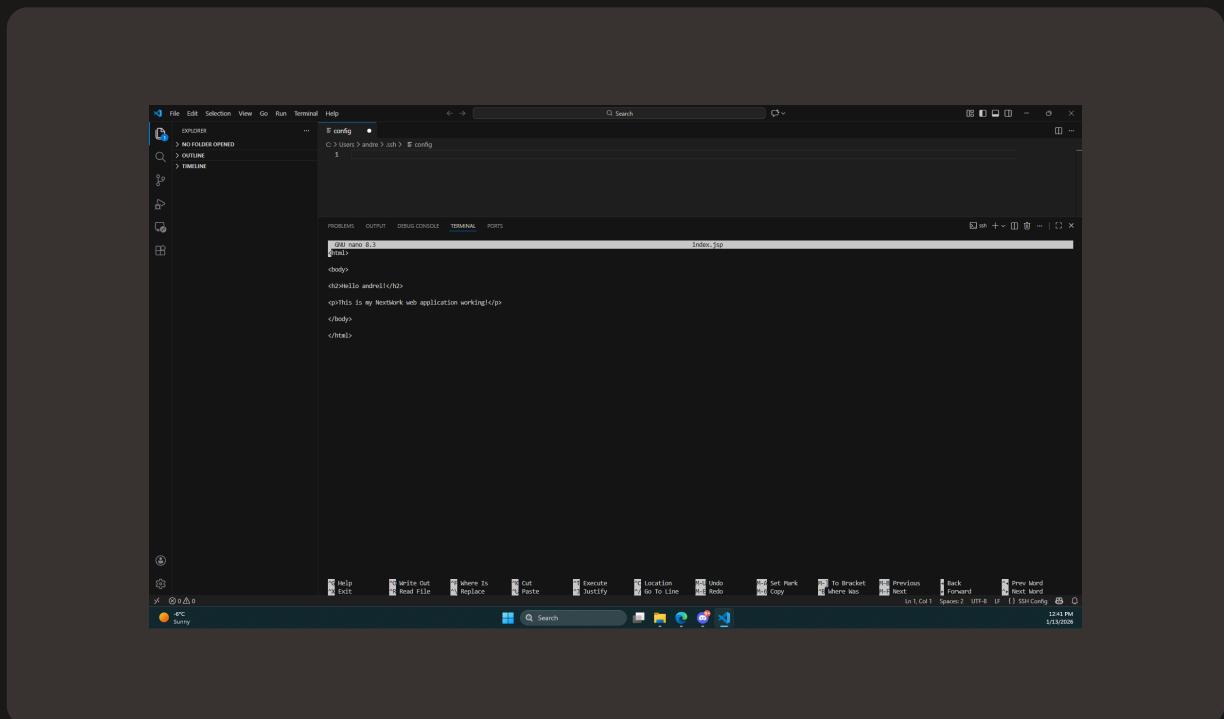
To verify my editing work in the terminal, I edited index.jsp using a terminal-based editor and then checked the file contents. It was possible to see my changes in VS Code right away because both the terminal SSH session and VS Code Remote-SSH are connected to the same EC2 instance and working on the same files in real time.

A

Andrei Boboc

NextWork Student

nextwork.org





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

