Project Title:

Set Up a Web App Using AWS and VS Code

Objective:

The main objective of this project was to set up and deploy a simple web application using **Amazon Web Services (AWS)** and configure it through **Visual Studio Code (VS Code)** using **remote SSH connection**. This project marks the first step toward building a full **DevOps CI/CD pipeline**.

Tools and Technologies Used:

- **Amazon Web Services (AWS)** – For hosting the web application
- **Visual Studio Code (VS Code)** – For development and configuration
- **EC2 Instance** – To deploy and run the web server remotely
- **SSH (Secure Shell)** – For remote access and file transfer
- **.pem File** – For secure authentication
- **Linux Commands** – For server setup and configuration
- **Web Browser** – To test and verify deployment

Project Description:

In this project, I created and deployed a basic web application using **AWS EC2 instance** and configured it through **VS Code**.
The process began with launching an EC2 instance on AWS, followed by downloading and setting up the **.pem key file** for secure access. I then connected the EC2 instance with **VS Code Remote SSH** extension to configure and edit files directly on the server.

The web application was set up by installing and running the required dependencies on the EC2 instance (for example, a simple HTML or Node.js/Python app). Finally, I verified the setup by accessing the public IP of the instance in the browser to confirm that the web app was live.

Challenges Faced:

During the project, there were several instances when the **SSH connection** could not be established. After troubleshooting, I found that the issue was related to **incorrect configuration of the .pem file**. Adjusting file permissions and verifying the correct file path helped resolve the problem.

Another challenge was understanding how **remote connections** work between the local VS Code and AWS EC2 host. Once configured properly, it became much easier to modify and manage files directly from VS Code.

Time Taken:

This project took approximately **2 hours** to complete, including setup, debugging, and testing.

Launching an EC2 instance

I started this project by launching an EC2 Instance because EC2 instances are like virtual computers that live in the cloude want our web app to live entirely in the cloud I am launching an EC2 instance to even develop our web app's code.

I also enabled SSH

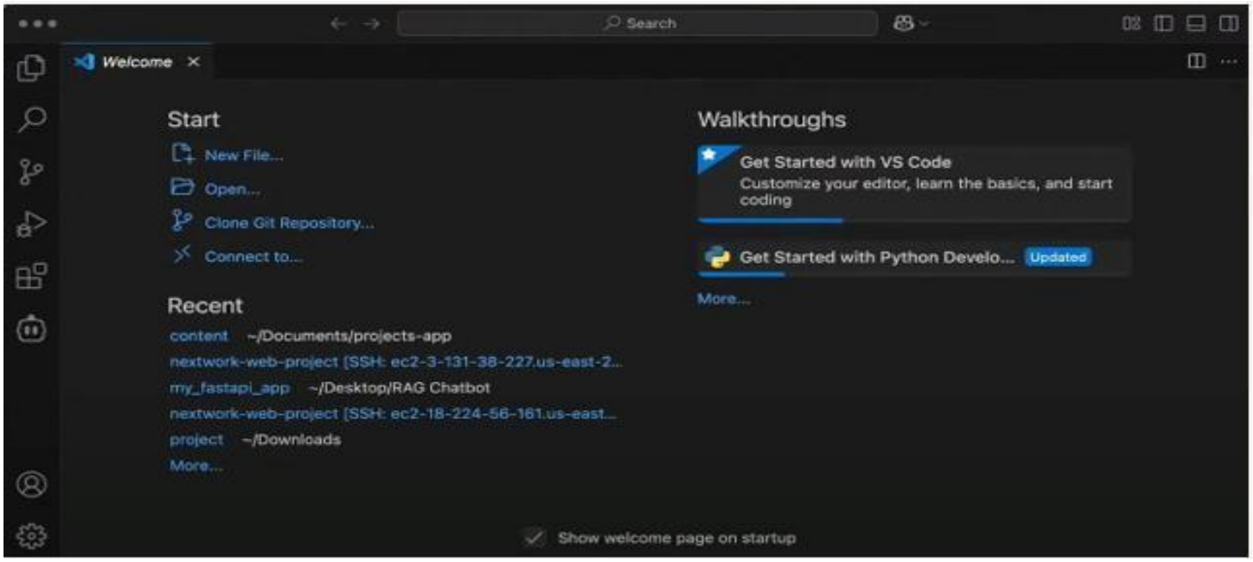
SSH is the protocol that authorizers users (like ourselves) to access remote servers (like EC2 instances). It's also a type of traffic that lets us transfer data back and forth with our EC2 instance once we're connected to it.

Key pairs

A key pair is a mechanism for us to connect and get access to EC2 instances we launch in AWS. We've created a key pair for the EC2 instance that we're launching Once I set up my key pair, AWS automatically downloaded the pem file of our key pair and I locally saved it to my system in folder named Devops

Set up VS Code

VS Code is an IDE that we're using today to write an edit our web app's code. It also has handy extensions that let us directly connect to an EC2 Instance, so that we can write and ed code that will live inside the instance. I installed VS Code to write and edit our Web's app code



Customize your editor, learn the basics, and start coding

Recent

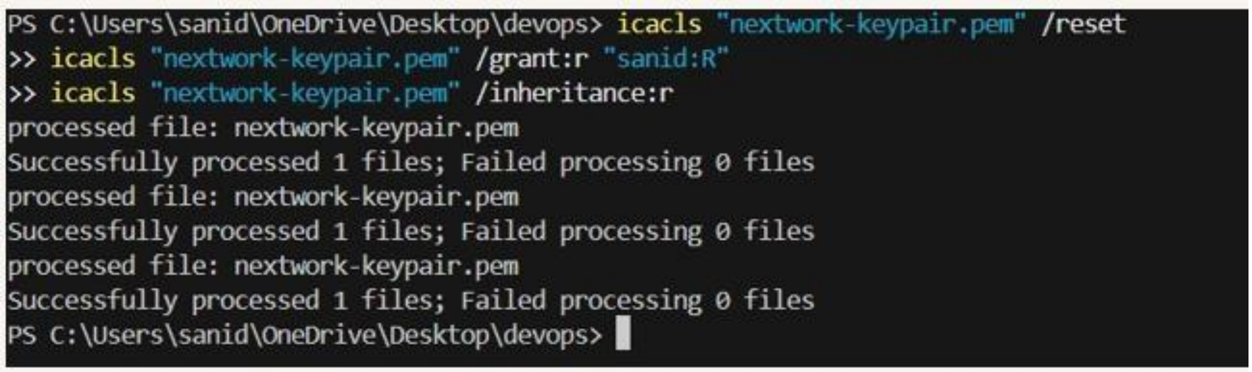
content ~/Documents/projects-app
nextwork-web-project [SSH: ec2-3-131-38-227.us-east-2...
my_fastapi_app ~/Desktop/RAG Chatbot
nextwork-web-project [SSH: ec2-18-224-56-161.us-east...
project ~/Downloads
More...

☑ Show welcome page on startup

My first terminal commands

A terminal is text based interface that allows users to interact with a computer system... The first command I ran for this project is cd C:\Users\sanid\OneDrive\Desktop\devops which navigates the terminal

I also updated my private key's permissions by runnung commnad icacls "nextworkkeypair.pem" /reset icacls "nextwork-keypair.pem" /grant:r "sanid:R" icacls "nextwork-keypair.pem" /inheritance:r which were successfully modified



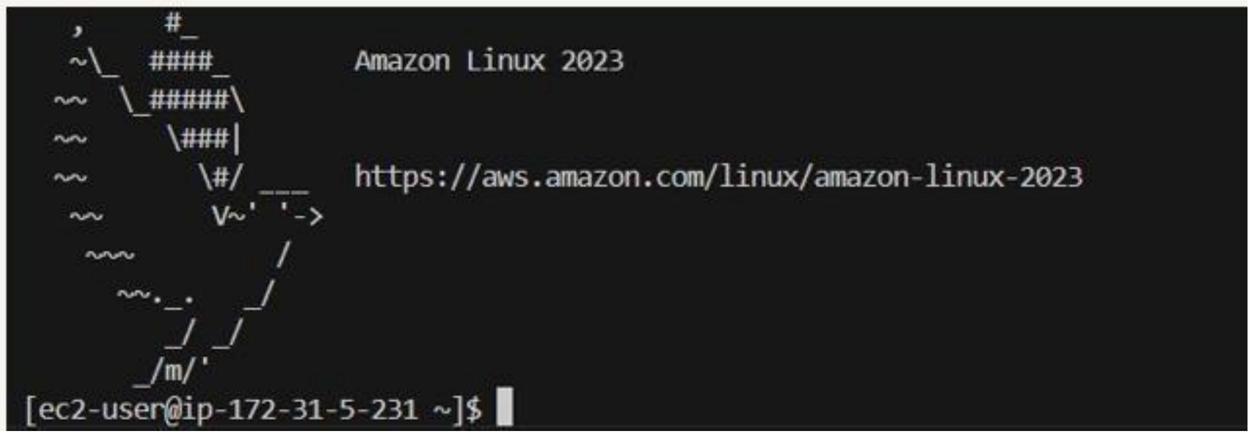
```
PS C:\Users\sanid\OneDrive\Desktop\devops> icacls "nextwork-keypair.pem" /reset
>> icacls "nextwork-keypair.pem" /grant:r "sanid:R"
>> icacls "nextwork-keypair.pem" /inheritance:r
processed file: nextwork-keypair.pem
Successfully processed 1 files; Failed processing 0 files
processed file: nextwork-keypair.pem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

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command C:\Users\sanid\OneDrive\Desktop\devops/nextwork-keypair.pem ec2-user@ec2-43204-103191.ap-south-1.compute.amazonaws.com which consist of my location of my pem file ,public IPv4 address of my instance

This command required an IPv4 address

A server's IPV4 DNS is is like its public address that identifies wher the server lives in the cloud. In our case, an EC2 Aistance's IPv4 DNS is useful information to give to our local computer- it tells our local computer where to find the EC2



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-5-231 ~]$ 
```

Maven & Java

Apache Maven is a tool that helps us in creating and organizing java projects like this one it comes with a lot of use cases, like being a package manager (downloading esternal pieces of code) and the tool that uses archetypes (templates)

Maven is required in this project because we want to use its ability to spin web apps usin archetypes! We're about to set up our a web app using web app arechetype

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

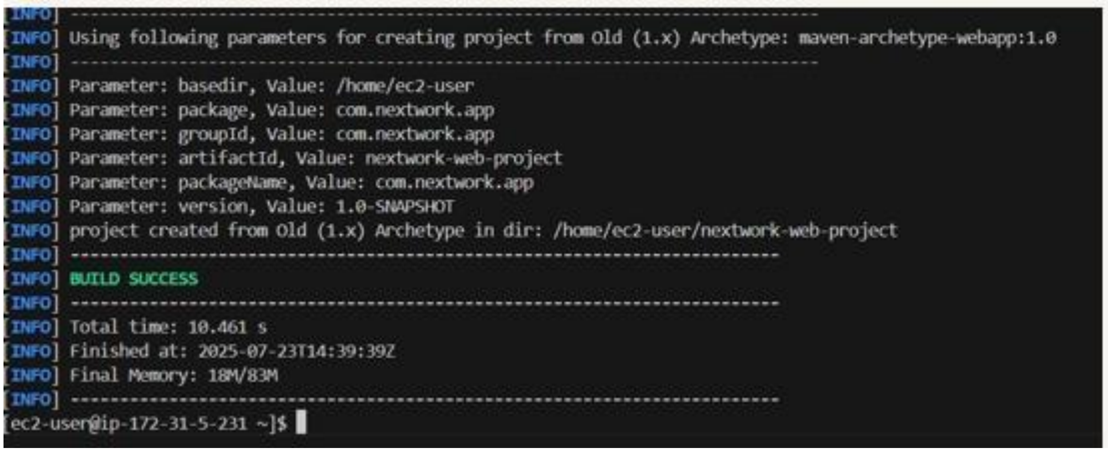
Java is required in this project because it sets a foundation to our project and maven also needs java to work on it

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 which created our project

I installed Remote - SSH, which is a VS Code extension that helps us to connect our VS code directly to a remote server like EC2 instance

Configuration details required to set up a remote connection include The host (ie. the EC2 instance's address), the identity file (i.e. the location of our private key) and the user (ie, the user that we're logging into for our instance

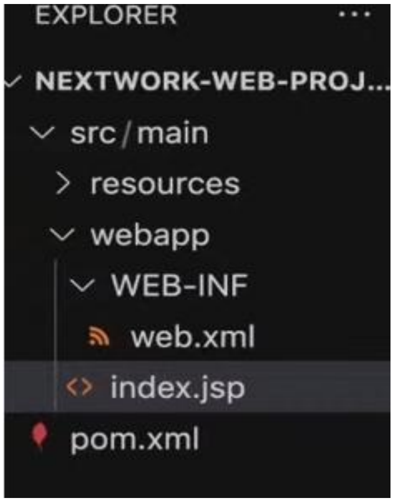


```
[INFO] ------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] ------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /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: 10.461 s
[INFO] Finished at: 2025-07-23T14:39:39Z
[INFO] Final Memory: 18M/83M
[INFO] ------------------------------------------------------------------------
[ec2-user@ip-172-31-5-231 ~]$
```

Create the Application

Using VS Code's the explorer, I could see a bunch of folders and subfolders that Define with app. These folders organise different parts of the web app for example, the resources sub-folder

These folders organise different parts of the web app, for example, the resources subfolder store connection details, while the webapp subfolder store web app files for the look and feel of the web app

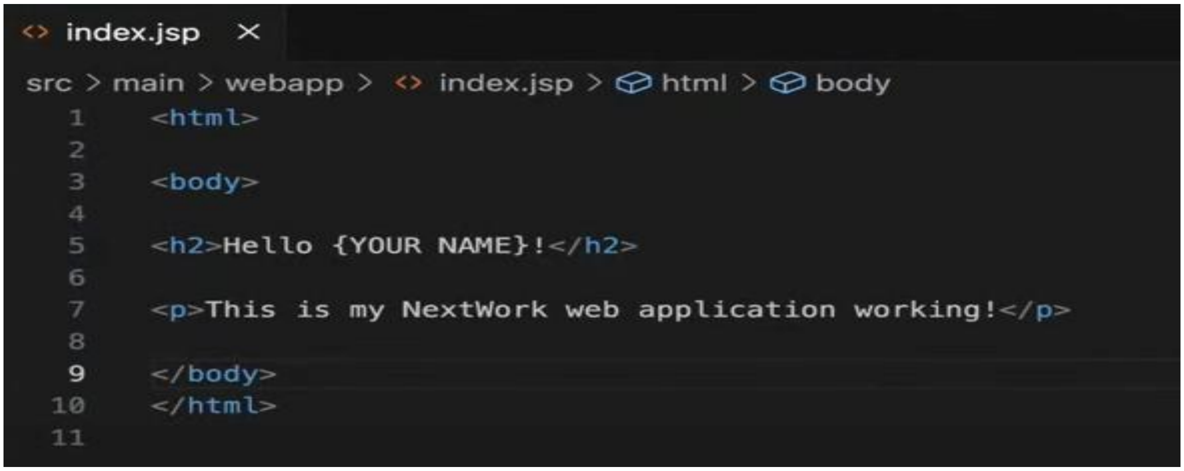


```
EXPLORER ...
NEXTWORK-WEB-PROJ...
  src / main
    resources
    webapp
      WEB-INF
        web.xml
      index.jsp
  pom.xml
```

Using Remote - SSH

Index.jsp is the file in our web app that defines both HTML content (ie. the static elements that go into our web app's page), as well as any code for generating dynamic content (i.e. content that's always changing

We updated the index.jsp by changing and editing the HTML code



index.jsp

src > main > webapp > index.jsp > html > body

```
<html>

<body>

<h2>Hello {YOUR NAME}!</h2>

<p>This is my NextWork web application working!</p>

</body>
</html>

```

Learning Outcomes:

- **Learned how to connect a local development environment (VS Code) to a remote AWS EC2 instance.**
- **Gained practical understanding of SSH authentication and .pem file usage.**
- **Experienced basic web application deployment on AWS.**
- **Improved confidence in troubleshooting connection issues.**
- **Understood the foundational steps required for building a DevOps pipeline.**

Reflection:

The most challenging part of this project was establishing a stable connection between the **remote SSH** and the **AWS host**. However, it was also the most rewarding to finally see my **web app code configured and running through VS Code**. This experience gave me valuable insight into real-world DevOps workflows and cloud deployment.

This project is the **first part of a DevOps project series** in which I plan to build a complete **CI/CD pipeline**. I will continue working on the next phases to strengthen my understanding of automation, integration, and deployment in DevOps environments.

Conclusion:

Setting up a web app using AWS and VS Code was an excellent learning experience. It helped me bridge the gap between cloud computing and development tools, and provided a strong foundation for upcoming DevOps projects.