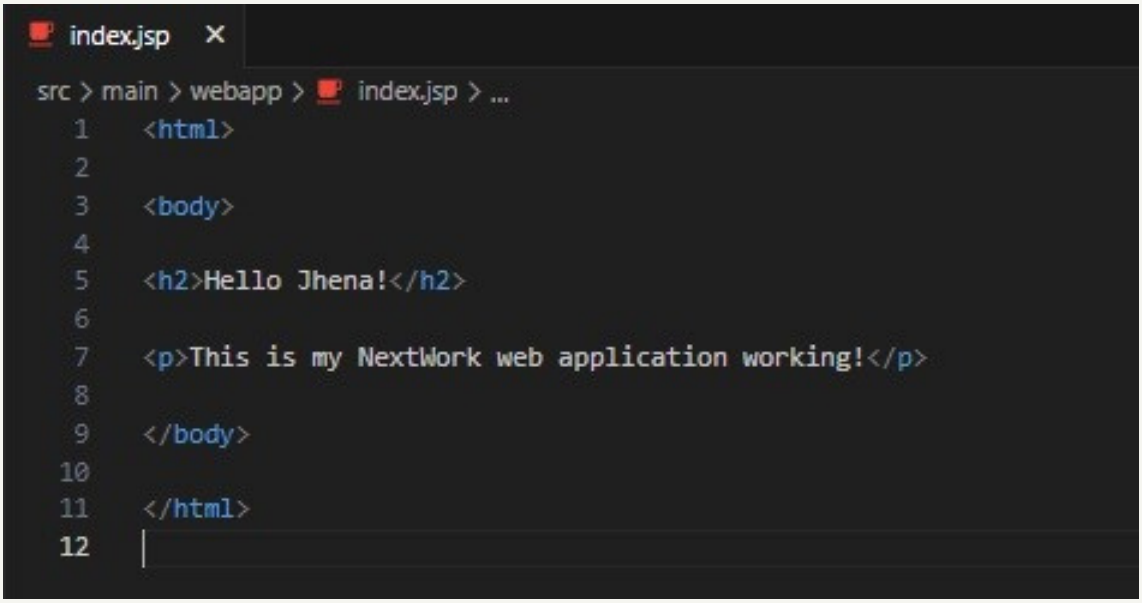
Set Up a Web App Using AWS and VS Code

SU suboljhenal yn3@gmail.com



Introducing Today's Project!

In this project, I will build a simple web app in the cloud using AWS (Amazon Web Services). This platform offers the necessary cloud resources and infrastructure, such as servers and databases, to effectively host and operate the app.

Key tools and concepts

Services I used in this project are Amazon EC2 which acts a virtual server, SSH for secure remote acces to my EC2 instance, Maven and Java to build and manage dependencies of the web app, VS Code+ Remote SSH to edit and manage code remotely from my local machine.

Project reflection

One thing I didn't expect in this project is the use of SSH to access my EC2 instance. It turns out that it is very useful and convenient when working in remote infrastracture.

This project took me approximately 3 hours. The most challenging part was connecting the VS Code to my EC2 instance as I encountered several errors. However, it was most rewarding to see my web app generated on the virtual server and edit files through VS Code.

This project is part one of a series of DevOps projects where I am building a CI/CD pipeline which is a tool that automates the build and deployment of a web app!

Launching an EC2 instance

I started this project by launching an EC2 instance as I am running the web app in the cloud. EC2 instance provides a centralized environment where all files, services and configurations of my web app are stored.

I also enabled SSH

SSH (Secure shell) is a protocol that guarantees exclusive access to my EC2 instance. I enabled SSH to establish a secure connection with the remote server, ensuring that my data remains protected while it transfers between the server and my computer.

Key pairs

A keypair in EC2 is a security mechanism that allows us to connect securely to our EC2 instance. It consists of 2 parts:

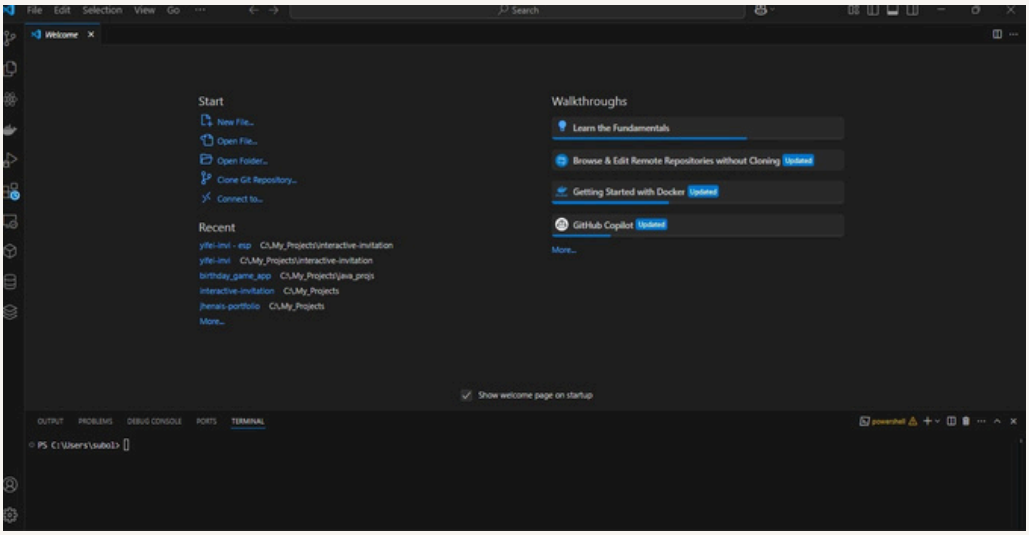
- Public key (stored by AWS)
- Private key (downloaded and kept securely in our local machine)

Once I set up my keypair, AWS automatically downloaded the private key as a *.pem file*.This file will be used to decrypt the login information that AWS has ecrypted using my public key. It guarantees that only I (or anyone possessing my private key) can access to the EC2 instance.

Set up VS Code

VS Code is one of the most popular tools (IDE) for creating and managing code projects. It comes with extra tools that lets us connect to virtual servers, i.e. EC2 instance that I will be using in this project.

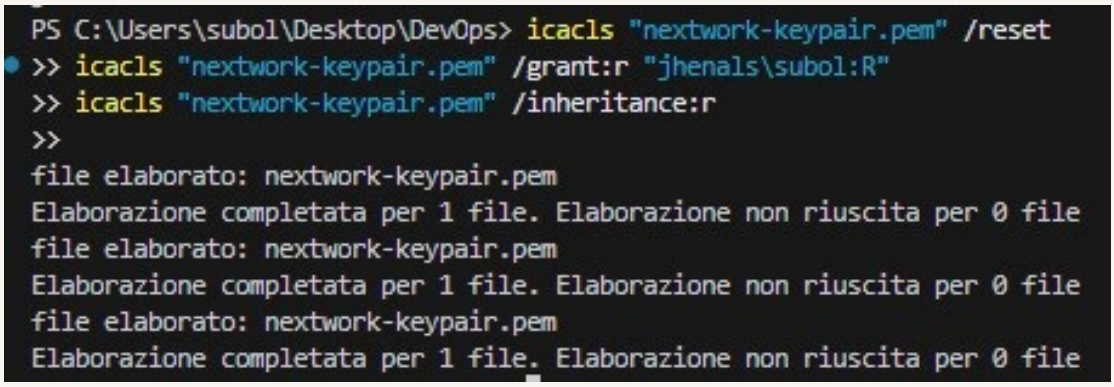
I installed VS Code to access my EC2 instance and work and edit my code remotely.



My first terminal commands

A terminal is a tool that allows us to communicate with the computer by typing commands instead of using the mouse. The first command I ran for this project is
cd ~/Desktop/DevOps
which tells the terminal to navigate from Home directory(~) to DevOps directory that is located in the Desktop.

I also updated my private key's permission using the ICACLS tool of Window. It allows me to remove the default permission setting of the .pem file (*/reset*) and gain access to the secret key (*/grant:r "%USERNAME%:R"*).



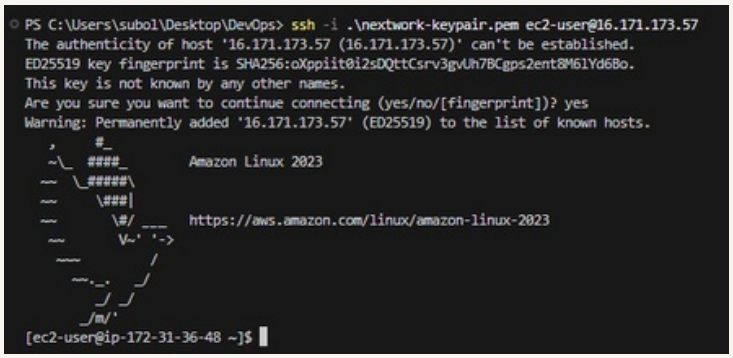
The third command is used to make sure changes in the permissions of other files and the DevOps folder won't change the permission settings for this file.

SSH connection to EC2 instance

To connect to my EC2 instance, I ran the command
ssh -i .\nextwork-keypair.pem ec2-user@16.171.173.57
where

- *ssh :* starts a secure shell connection to the EC2 instance
- *-i :* specifies the identity file (.pem file) to authenticate a connection
- *ec2-user@16.171.173.57 :* specifies the username (ec2-user) and the address of the EC2 instance (public DNS) to connect to.

This command required an IPv4 address



A server's IPV4 DNS is the public address of my EC2 server that the internet uses to find and connect to it. My local computer will find and connect to the EC2 instance through this IPv4 DNS.

Maven & Java

Apache Maven is a tool that helps developers automate the process of building a software. Building a software means producing a final product, such as web app, from our code, that is ready to be hosted in virtual machines like EC2 instance.

Maven is required in this project as it streamlines the software development process. This includes compiling our code, linking external dependencies, packaging the files into a cohesive unit (i.e. JAR), and automating testing procedures.

Java is a popular programming language used to build different applications, from mobile apps to large enterprise systems.

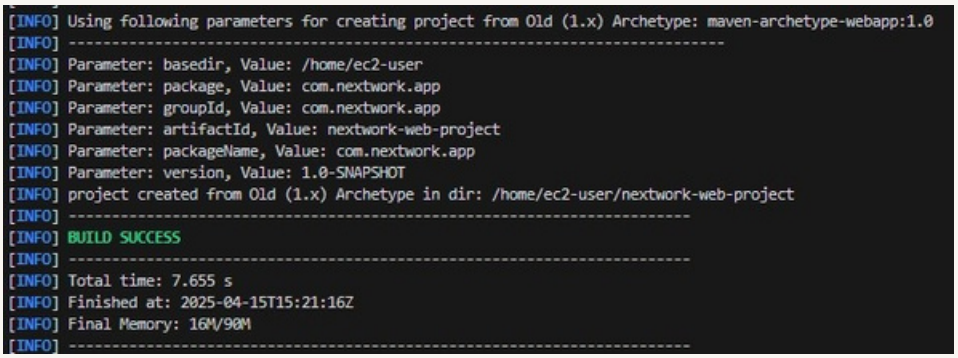
Java is required in this project since Amazon offers a free and dependable version of Java.

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

I installed Remote-SSH in VS Code to connect securely and directly to my EC2 instance via the internet. This allows me to access and manage my web app as if I were working on my local machine, even though everything is operating in the cloud.

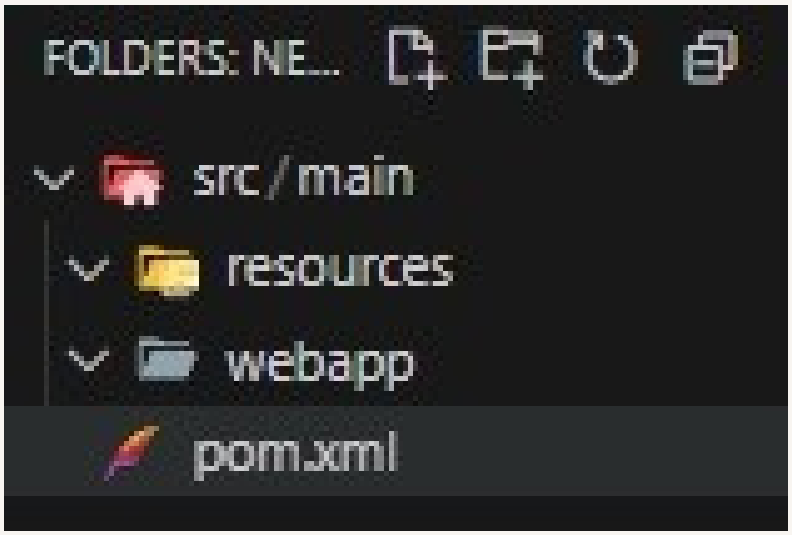
Configuration details required to set up a remote connection include information about Host, Hostname, Identity file and User.



Create the Application

Using VS Code's file explorer, I can view the web app's files, folders and subfolders just like a project on my local machine.

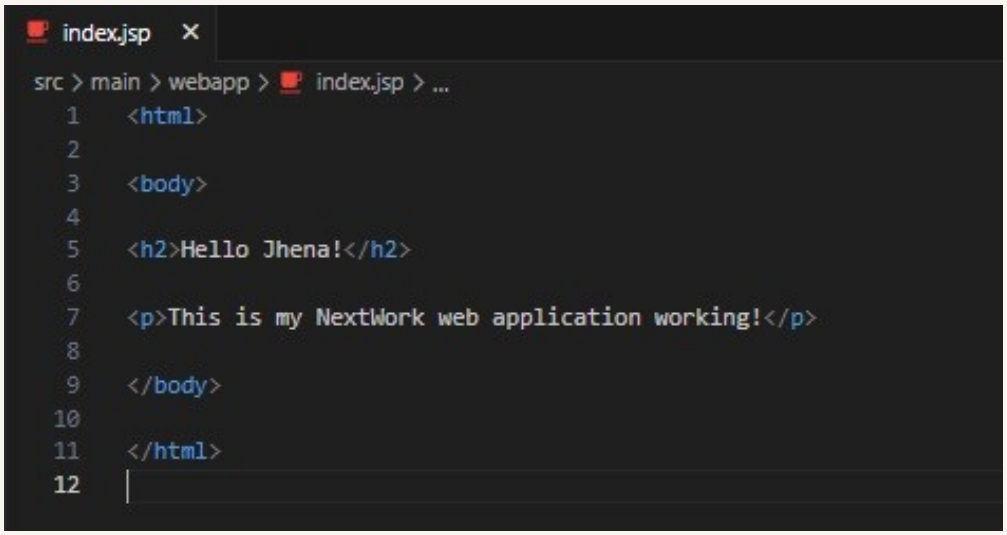
Two of the project folders created by Maven are *src* which holds all the source code files and *webapp* which is a subfolder that contains the web app's files such as HTML, CSS, JS and JSP files.



Using Remote - SSH

index.jsp file is used in Java web apps, similar to an HTML because it contains markup to display web pages. However, it also incorporates java code, allowing it to generate dynamic content.

I edited index.jsp by changing the h2 element from "Hello World!" to "Hello Jhena!" then adding a paragraph element after.



Conclusion

This project provided a hands-on introduction to deploying a web application in the cloud using AWS EC2 and tools such as Java, Maven and Visual Studio Code with Remote-SSH. Throughout the process, I not only acquire basic knowledge of cloud infrastructure and secure remote connections but also experience the real-world challenges of troubleshooting connectivity and permissions.

One of the most rewarding aspects (as I have said in the introduction) was seeing the web app successfully run on the virtual server and being able to manage it directly on my local machine.

Completing the project has laid a strong foundation for the next phases in my DevOps learning journey, and I'm excited to continue building more advanced automation and deployment workflows.

Resources/ References

- https://learn.nextwork.org/projects/aws-devops-vscode

Everyone should be in a job they love.

Check outnextwork.orgfor more projects