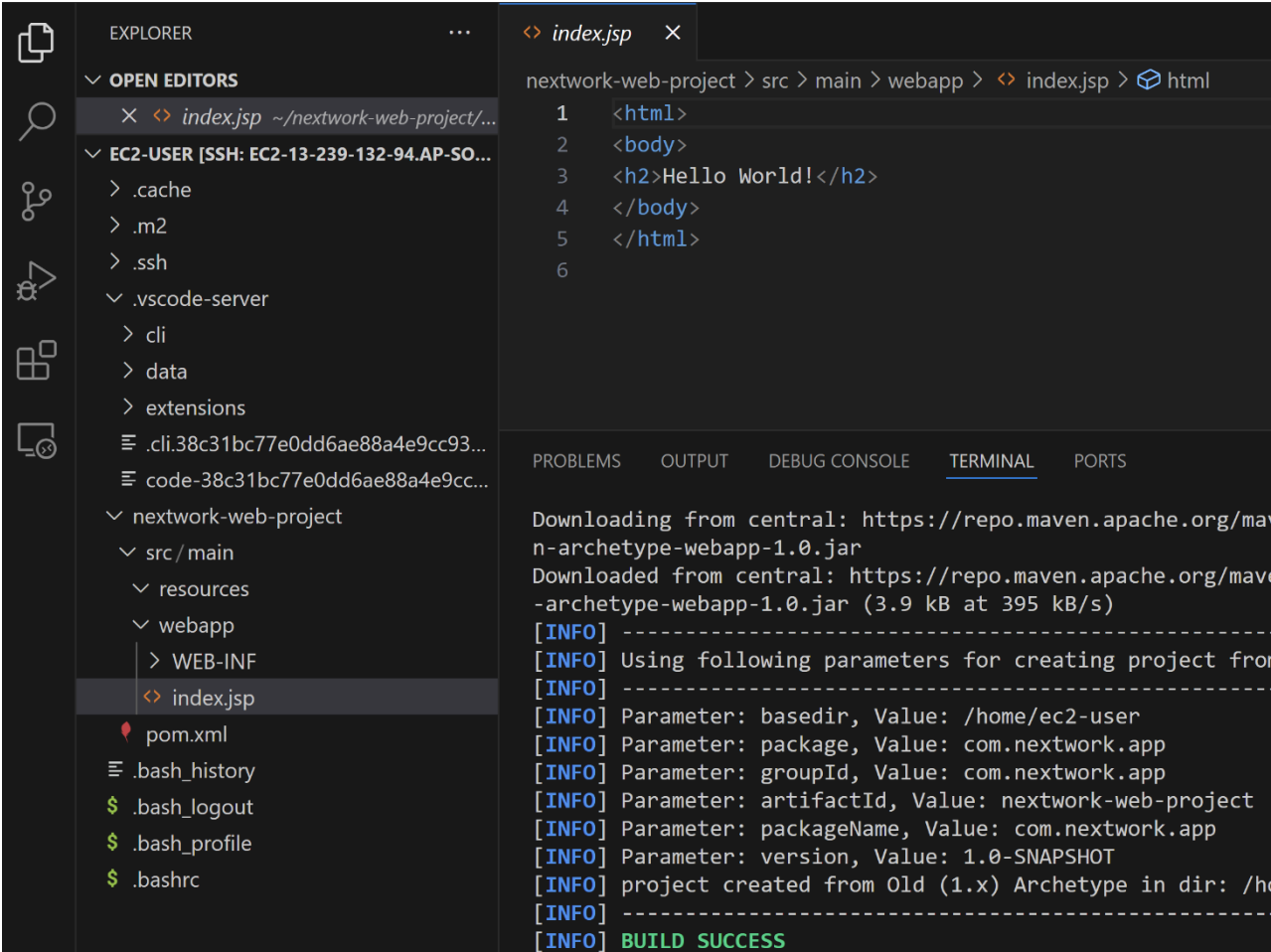




Build a Web App and IDE in AWS

 Prajit Venkatachalam

<https://www.linkedin.com/in/prajit-venkatachalam-435b2a150/>



The screenshot displays an IDE interface with a dark theme. On the left, the Explorer panel shows a project structure for 'nextwork-web-project' located on an EC2 instance. The project contains a 'src/main/webapp' directory with an 'index.jsp' file. The main editor window shows the content of 'index.jsp', which is a simple HTML page with a 'Hello World!' message. Below the editor, the Terminal panel shows the output of a Maven command, indicating a successful build of the web application. The terminal output includes messages about downloading the 'maven-archetype-webapp-1.0.jar' and the parameters used for the project creation.

```
nextwork-web-project > src > main > webapp > index.jsp > html
1 <html>
2 <body>
3 <h2>Hello World!</h2>
4 </body>
5 </html>
6

[INFO] -----
[INFO] Using following parameters for creating project from
[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: /h
[INFO] -----
[INFO] BUILD SUCCESS
```



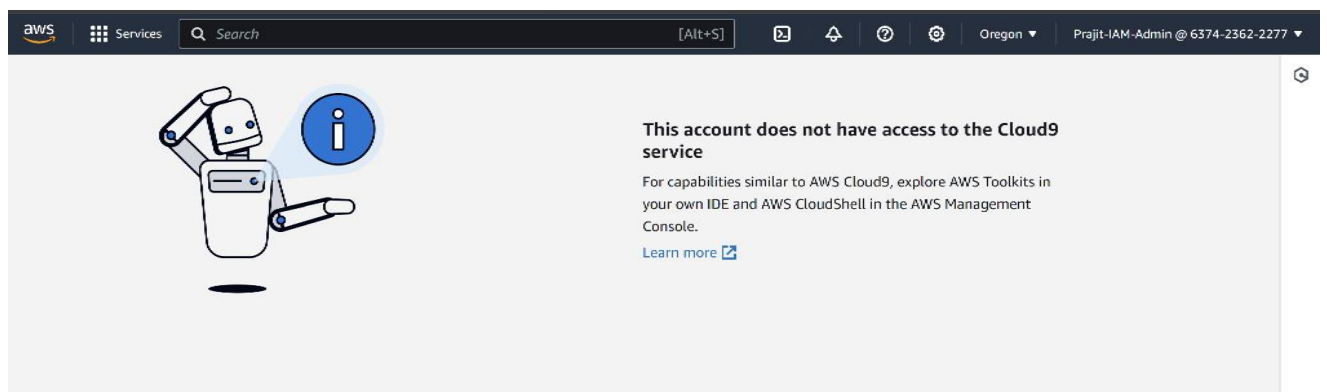
Introducing Today's Project!

What is AWS Cloud9 and why is it useful?

AWS Cloud9 is a cloud-hosted Integrated Development Environment (IDE) offered by AWS. It serves as a tool that developers use to efficiently write, debug, and manage code for their applications. Cloud9 simplifies the development process by providing a seamless environment accessible directly from a web browser.

How I'm using Cloud9 in this project

I encountered an accessibility issue with my account that prevented me from using AWS Cloud9. As an alternative, I used GitBash on my local machine to connect to my EC2 instance via SSH, and Visual code for customizing the application. Attached is a screenshot of the error I received when attempting to access Cloud9 from the AWS console.



One thing I didn't expect

One thing that I did not expect in this project was an error with accessing Cloud9 service from AWS console.

This project took me...

This project took me 1.5 hours to complete include report writing and finding the alternative solution (GitBash and Visual code).



Set up an IAM User

An IAM user is an additional user that gets access to my AWS account's resources. When creating a user, I can specify in detail the level of access they have to my account's resources and services.

The importance of IAM users

It is important to create IAM users because the root user is vulnerable to security breaches that could result in my billing information being accessed without my permission.

I created an IAM user with Administrator access. This means my IAM user is allowed to perform all possible actions on all the resources in my account.

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Email sign-in instructions

Console sign-in URL
 <https://nextwork-alias-prajit.signin.aws.amazon.com/console>

User name
 Prajit-IAM-Admin

Console password
 ***** [Show](#)

Cancel

Download .csv file

Return to users list



Environments & IDEs

What is an environment?

Environment is set of requirements/resources required to run and build your software application. It can be also defined as a software environment which provides all the necessary tools, libraries, and settings that developers need to build and run an application. This environment can be created in Cloud9.

What is an IDE?

An IDE, or Integrated Development Environment, refers to software designed to assist developers in writing, debugging, and managing code more efficiently. It functions like a comprehensive toolkit, providing essential components such as a text editor, tools to execute and test code, and often additional features to streamline the software development process.

What is the benefit of using Cloud9

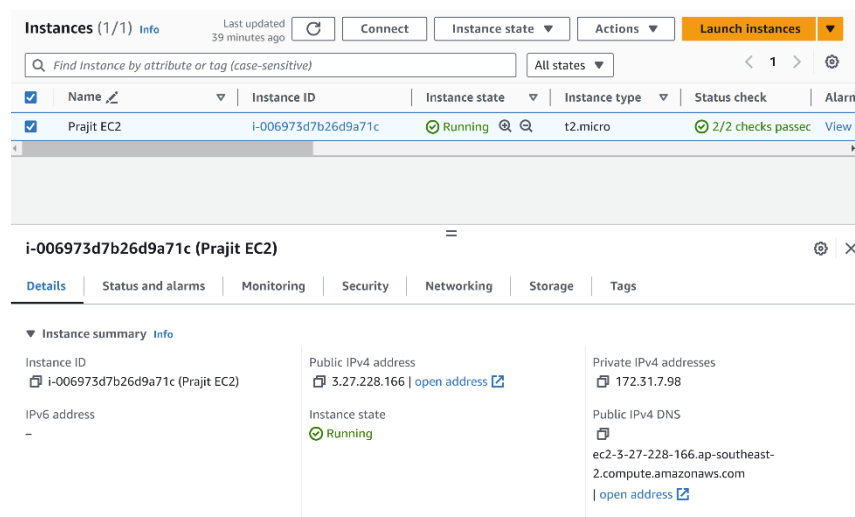
One advantage of using Cloud9 is that there's no need to download or install any external software to use the IDE. It can be accessed directly through a web browser via the AWS console.



Alternative way

Step 1: Creating an EC2 instance

I created an EC2 instance from the AWS Management Console (Important: Choose only Amazon Linux 2 AMI- free tier, allow HTTP & HTTPS traffic from network settings, generate key pair). I also generated a key pair during the instance creation process, as I will be connecting to my EC2 instance via SSH from my local machine.




Step 2: Connecting to my EC2 instance

I used Git Bash to connect to my EC2 instance via SSH. Before that, I changed the permissions of my key pair by navigating to the directory where the key pair was downloaded, using the command `cd /path/to/your/directory` (this is just an example; you need to replace it with the actual path of your key pair's folder by right-clicking on the folder and selecting "copy as path"). Then, I used the command `chmod 400 /path/to/your-key-pair.pem` (right-click on the key pair file and select "copy as path").

I connected to my EC2 instance using the following command:
`ssh -i /path/to/your-key-pair.pem ec2-user@your-ec2-instance-public-dns` (The Public DNS can be copied from the EC2 instance dashboard).

```
vpraj@ip-172-31-79-80 ~  
vpraj@DESKTOP~TCNBJQM MINGW64 ~  
$ cd "C:\Users\vpraj\OneDrive\Desktop\AWS\Projects\I0. Setting up a Web app+IDE in the cloud"  
  
vpraj@DESKTOP~TCNBJQM MINGW64 ~/OneDrive/Desktop/AWS/Projects/I0. Setting up a Web app+IDE in the cloud  
$ chmod 400 "C:\Users\vpraj\OneDrive\Desktop\AWS\Projects\I0. Setting up a Web app+IDE in the cloud\Devops-Proj  
ect-EC2.pem"  
  
vpraj@DESKTOP~TCNBJQM MINGW64 ~/OneDrive/Desktop/AWS/Projects/I0. Setting up a Web app+IDE in the cloud  
$ ssh -i "C:\Users\vpraj\OneDrive\Desktop\AWS\Projects\I0. Setting up a Web app+IDE in the cloud\Devops-Proje  
ct-EC2.pem" ec2-user@ec2-3-27-228-166.ap-southeast-2.compute.amazonaws.com  
The authenticity of host 'ec2-3-27-228-166.ap-southeast-2.compute.amazonaws.com (3.27.228.166)' can't be establ  
ished.  
ED25519 key fingerprint is SHA256:Dl+dqLH58u+ogf9csKDL34hsccdmjbZo6LBHTzJF1g.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-3-27-228-166.ap-southeast-2.compute.amazonaws.com' (ED25519) to the list of kno  
wn hosts.
```



```
#_#  
#####  
~~~~~|  
####  
##|  
#/  
V~^ ^r->  
~~~~~  
_____  
_/m/_
```

```
Amazon Linux 2  
  
AL2 End of Life is 2025-06-30.  
  
A newer version of Amazon Linux is available!  
  
Amazon Linux 2023, GA and supported until 2028-03-15.  
https://aws.amazon.com/linux/amazon-linux-2023/
```

```
[ec2-user@ip-172-31-79-80 ~]$
```

Step 3: Set Up the Development Environment on EC2

- ## 1. Install Apache Maven using the command

```
sudo wget https://repos.fedorapeople.org/repos/dchen/apache-
maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-
maven.repo
sudo sed -i s/$releasever/6/g /etc/yum.repos.d/epel-apache-
maven.repo
sudo yum install -y apache-maven
```

- ## 2. Install Amazon Corretto 8 using the command

```
sudo amazon-linux-extras enable corretto8
sudo yum install -y java-1.8.0-amazon-corretto-devel
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64
export PATH=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre/bin/:$PATH
```

- ### 3. Verify the installation using the command

```
java -version
mvn -version
```



Maven & Java

Maven is a tool that helps developers build their software by automating the steps required to produce the final product that computers can run. The steps includes compiling, linking, packaging and testing.

Maven is required for this project because I am building a web app in a specific programming language that cannot be built on its own without the help of Maven.

Java is a programming language that I am using to create my web application for this project. It is used to build different types of applications, from mobile apps to large enterprise systems.

Java is required for this project because it is a versatile tool for creating various applications, including web apps. The Java version that I am using in this project is Amazon Corretto 8.

```
Complete!  
[ec2-user@ip-172-31-7-98 ~]$ export JAVA_HOME=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64  
[ec2-user@ip-172-31-7-98 ~]$ export PATH=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre/bin/:$PATH  
[ec2-user@ip-172-31-7-98 ~]$  
[ec2-user@ip-172-31-7-98 ~]$ java -version  
java -version  
openjdk version "1.8.0_422"  
OpenJDK Runtime Environment Corretto-8.422.05.1 (build 1.8.0_422-b05)  
OpenJDK 64-Bit Server VM Corretto-8.422.05.1 (build 25.422-b05, mixed mode)  
[ec2-user@ip-172-31-7-98 ~]$ mvn -version  
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)  
Maven home: /usr/share/apache-maven  
Java version: 1.8.0_422, vendor: Amazon.com Inc.  
Java home: /usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "5.10.224-212.876.amzn2.x86_64", arch: "amd64", family: "unix"  
[ec2-user@ip-172-31-7-98 ~]$
```



Step 4: Install Visual Studio Code and Remote - SSH Extension

- Download and install Visual Studio Code from [here](#) if you haven't already.
- Install the Remote - SSH extension in Visual Studio Code:
 - Open Visual Studio Code.
 - Go to the Extensions view by clicking on the Extensions icon in the Activity Bar or pressing Ctrl+Shift+X.
 - Search for "Remote - SSH" and install it.

Step 5: Configure the SSH Connection in Visual Studio Code

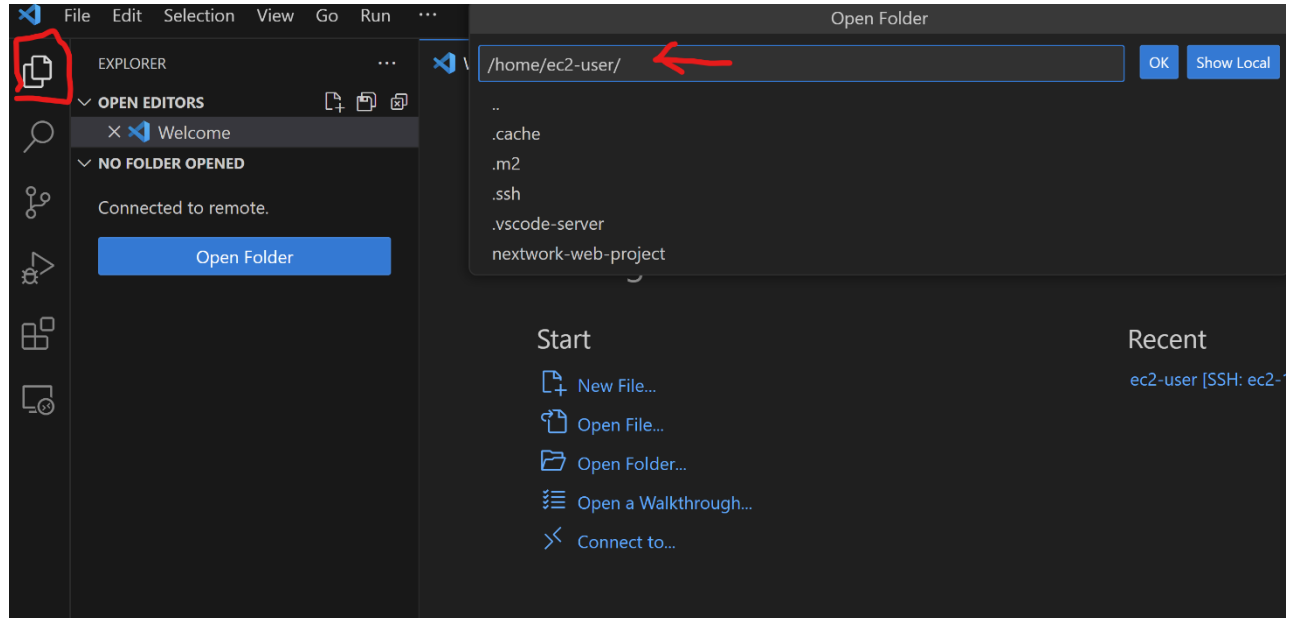
- Open Visual Studio Code.
- Press Ctrl+Shift+P to open the Command Palette.
- Type Remote-SSH: Add New SSH Host.
- Enter the SSH connection string, for example:
 - 'ssh -i /path/to/your-key-pair.pem ec2-user@your-ec2-instance-public-dns' (Same command that was used in step 2 to connect to the EC2 instance)
- Choose a file to save the SSH configuration.

Step 6: Connect to the EC2 Instance

- Press Ctrl+Shift+P to open the Command Palette.
- Type Remote-SSH: Connect to Host.
- Select your configured host.

Step 7: Create a folder in the file explorer

- Once connected to EC2 instance, open a folder in the file explorer from the left top corner. (please refer to the image below)



Step 7: Set Up Your Java Web Application

- Open a new terminal in Visual Studio Code (connected to your EC2 instance).
- Generate a new Maven project using the command

```
mvn archetype:generate \
  -DgroupId=com.nextwork.app \
  -DartifactId=nextwork-web-project \
  -DarchetypeArtifactId=maven-archetype-webapp \
  -DinteractiveMode=false
```

Step 9: Accessing the index.jsp file for customization

- Once the above mentioned code is ran in the terminal, it will scan for the project, and “nextwork-web-project” will be created under EC2-USER folder. (Refer to the picture below)
- From that folder you can open the index.jsp file and customize it.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays the project structure: `nextwork-web-project` with subfolders `src/main/webapp` and `WEB-INF`. The `index.jsp` file is selected. The main editor shows the content of `index.jsp`, which is a simple HTML page:

```
1 <html>
2 <body>
3 <h2>Hello World!</h2>
4 </body>
5 </html>
6
```

 The bottom pane shows the Terminal output, which includes the Maven command used to generate the project and the resulting build success message:

```
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetype/maven-archetype-webapp-1.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetype/maven-archetype-webapp-1.0.jar (3.9 kB at 395 kB/s)
[INFO] Using following parameters for creating project from archetype:
[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] BUILD SUCCESS
```

Create the Application

- To create a simple Java web app, I ran the command

```
mvn archetype:generate \
  -DgroupId=com.nextwork.app \
  -DartifactId=nextwork-web-project \
  -DarchetypeArtifactId=maven-archetype-webapp \
  -DinteractiveMode=false
```

- Once the web app was created, my IDE's (Visual studio code) file explorer was populated with Java web app structure under the EC2 USER folder, which contains the `index.jsp` file.
- To customize the Web app's display, I updated the `index.jsp` file. The difference between `index.jsp` and `index.html` is that `index.jsp` can contain Java code, which is supported by cloud for creating web application.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

