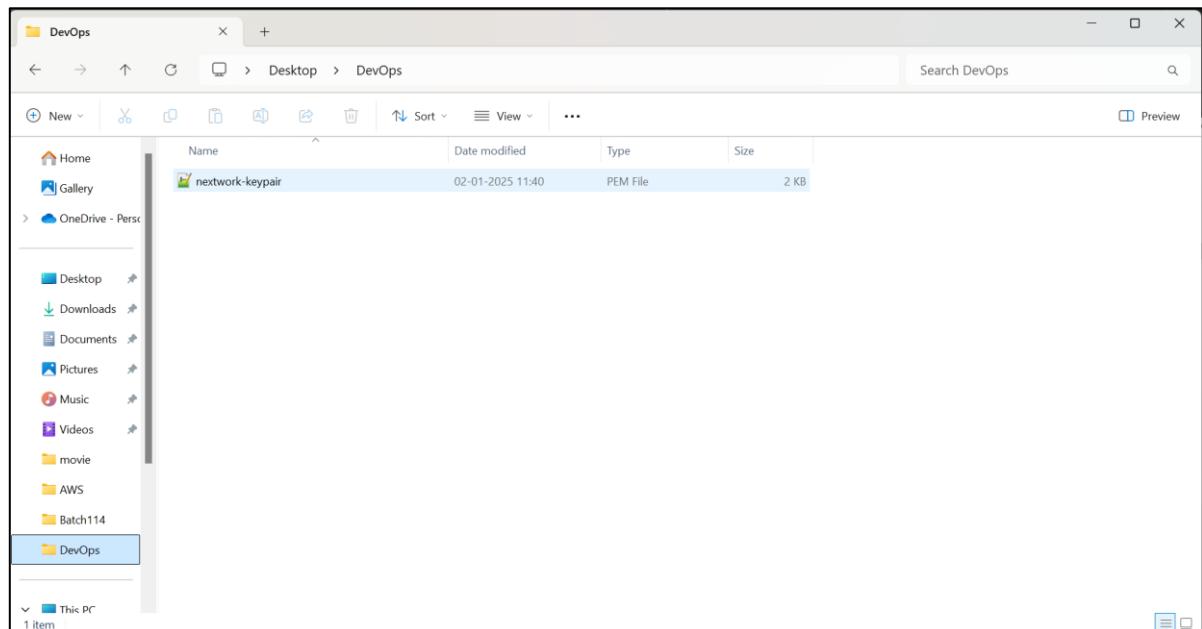


Set Up a Web App in the Cloud

- Setup IAM User

The screenshot shows the AWS IAM User configuration page for 'Vivek-IAM-Admin'. The 'Permissions' tab is selected. The user has an ARN: arn:aws:iam::245712304097:user/Vivek-IAM-Admin. Console access is enabled without MFA. Last console sign-in was today. There is one attached policy named 'AdministratorAccess'. The policy is an AWS managed job function.

- Log In with your IAM Admin User.
- Launch an EC2 Instance
 - Enter the Name of the Instance, eg: nextwork-devops-vivek
 - Choose Amazon Linux 2023 AMI under **Amazon Machine Image(AMI)**
 - Choose **t2.micro** under **Instance type**.
 - Under **Key pair (login)**, choose **Create a new key pair**.
 - Use nextwork-keypair as your key pair's name.
 - Keep the **Key pair type** as **RSA**, and the **Private key file format** as **.pem**
 - A new file will automatically download to your local computer. This is your private key. Before we lose track of our **.pem** file, let's organise it in our computer.
 - Head to your local computer's desktop.
 - Create a new folder in your desktop called **DevOps**.
 - Move your **.pem** file from your **Downloads** folder into your **DevOps** folder



- Back to our EC2 instance setup, head to the **Network settings** section
- For **Allow SSH traffic from**, select the dropdown and choose **My IP**. This makes sure only you can access your EC2 instance.
- If your IP address is different from what's under **My IP**, select **Custom** from the dropdown instead. Enter your IP and make sure to add a /32 to the end e.g. 012.345.678.9/32
- Leave the default values for the remaining sections

[EC2](#) > [Instances](#) > [Launch an instance](#)

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

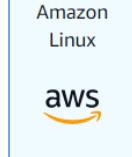
[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

[Recents](#) [Quick Start](#)



Amazon Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Linux

...

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI	Free tier eligible
<small>ami-0fd05997b4dff7aac (64-bit (x86), uefi-preferred) / ami-013b2876e77b2db31 (64-bit (Arm), uefi) Virtualization: hvm ENA enabled: true Root device type: ebs</small>	

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20241212.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Username
64-bit (x86)	uefi-preferred	ami-0fd05997b4dff7aac	ec2-user
Verified provider			

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

[Edit](#)
[Create new key pair](#)

▼ Network settings [Info](#)

[Network](#) | [Info](#)
vpc-096d0213e8f5795c

[Subnet](#) | [Info](#)
No preference (Default subnet in any availability zone)

[Auto-assign public IP](#) | [Info](#)
Enable
Additional charges apply when outside of free tier allowance

[Firewall \(security group\)](#) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[Create security group](#)
 [Select existing security group](#)

We'll create a new security group called 'launch-wizard-6' with the following rules:

Allow SSH traffic from
Helps you connect to your instance

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

▼ Summary

Number of instances | [Info](#)
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2... [read more](#)
ami-0fd05997b4dff7aac

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

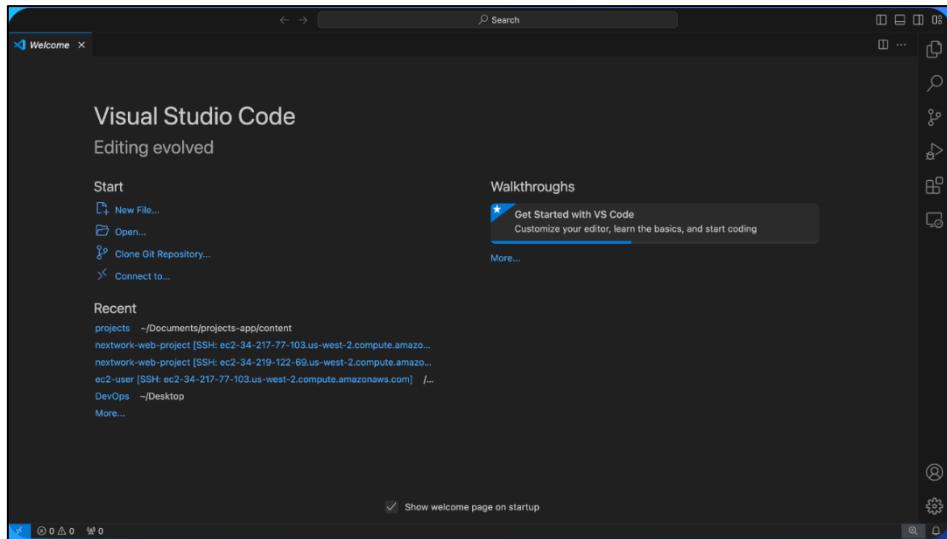
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IP-v4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#)
[Launch instance](#)
[Preview code](#)

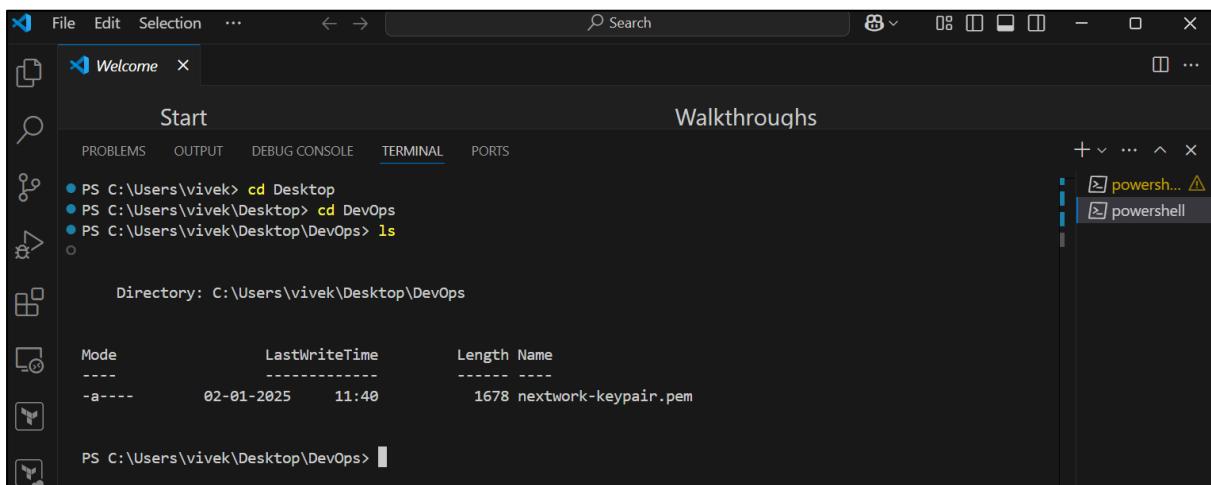
○ choose Launch instance

The screenshot shows the AWS EC2 Instances page with the URL [EC2 > Instances > Launch an instance](#). A green success banner at the bottom states: "Success Successfully initiated launch of instance (i-0fa042c101e71afb7)". The top navigation bar includes the AWS logo, search bar, and user information: Mumbai | Vivek-IAM-Admin @ 2457-1230-4097 ▾.

- Open VS Code



- Select **Terminal** from the top menu bar.
- Select **New Terminal** from the dropdown
- Navigate your terminal to the DevOps folder. You'll do this by entering this command in the terminal: `cd Desktop` and `cd DevOps`
- Once you're in the DevOps folder, you might want to check if your .pem file is there. Use this command: `ls`



- **Change the permissions of your .pem file**
- In the terminal, run the following command to allow access to your .pem file.

`ssh -i [PATH TO YOUR .PEM FILE] ec2-user@[YOUR PUBLIC IPV4 ADDRESS]`

eg: `ssh -i C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem ec2-user@ 13.235.86.185`

- Are you sure you want to continue connecting (yes/no/[fingerprint]) will flash on the screen
 - Type yes

The screenshot shows the Visual Studio Code interface with the terminal tab selected. The terminal window displays a PowerShell session:

```
PS C:\Users\vivek> cd Desktop
PS C:\Users\vivek\Desktop> cd DevOps
PS C:\Users\vivek\Desktop\DevOps> ls

Directory: C:\Users\vivek\Desktop\DevOps

Mode                LastWriteTime         Length Name
----                ——————              ——— —
-a---       02-01-2025     11:40          1678 nextwork-keypair.pem

PS C:\Users\vivek\Desktop\DevOps> ssh -i C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem ec2-user@15.207.248.121
ssh: connect to host 15.207.248.121 port 22: Connection timed out
PS C:\Users\vivek\Desktop\DevOps> ssh -i C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem ec2-user@13.235.86.185
The authenticity of host '13.235.86.185 (13.235.86.185)' can't be established.
ED25519 key fingerprint is SHA256:chu/HUBKKM5I7t/Yu9vBM/T1dhQMjcX/UUnbxltfaLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 
```

- You will connect to your EC-2 Instance

```
PS C:\Users\vivek\Desktop\DevOps> ssh -i C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem ec2-user@13.235.86.185
,      #
~\_ #####      Amazon Linux 2023
~~ \#####\
~~ \###|
~~   \#/ __  https://aws.amazon.com/linux/amazon-linux-2023
~~   \~'-'>
~~   /
~~ _._ _/
~~ /_/
~~ /m/
[ec2-user@ip-172-31-4-219 ~]$ 
```

- **Install Apache Maven and Amazon Corretto 8**

Now that the Connection DONE, This means your terminal has now entered into your EC2 instance and can use it like a computer that's right in front of you!

Our goal today is to set up a web app inside this instance, so let's install two tools that are going to help us build Java web apps.

Apache Maven and Amazon Corretto 8

- **In this step, you're going to:**
 1. Install Apache Maven on your EC2 instance.
 2. Install Amazon Corretto 8, a version of Java.
 3. Verify the installations.
- Install Apache Maven using the commands below. You can copy and paste **all** of these lines into the terminal together, no need to run them line by line.

```
wget https://archive.apache.org/dist/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz
```

```
sudo tar -xzf apache-maven-3.5.2-bin.tar.gz -C /opt
```

```
echo "export PATH=/opt/apache-maven-3.5.2/bin:$PATH" >> ~/.bashrc
```

```
source ~/.bashrc
```

```
--2025-01-02 10:12:23-- https://archive.apache.org/dist/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8738691 (8.3M) [application/x-gzip]
Saving to: 'apache-maven-3.5.2-bin.tar.gz'

apache-maven-3.5.2-bin.tar.gz 100%[=====] 8.33M 4.42MB/s in 1.9s
2025-01-02 10:12:25 (4.42 MB/s) - 'apache-maven-3.5.2-bin.tar.gz' saved [8738691/8738691]

[ec2-user@ip-172-31-4-219 ~]$
```

Apache Maven Installed

- Now we're going to install Java 8, or more specifically, Amazon Corretto 8.
- Run these commands:

```
sudo dnf 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
```

- To verify that Maven is installed correctly, run the following command next:

```
mvn -v
```

```
[ec2-user@ip-172-31-4-219 ~]$ mvn -v
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)
Maven home: /opt/apache-maven-3.5.2
Java version: 1.8.0_432, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.119-129.201.amzn2023.x86_64", arch: "amd64", family: "unix"
```

- To verify that you've installed Java 8 correctly, run this next:

```
java -version
```

```
[ec2-user@ip-172-31-4-219 ~]$ java -version
openjdk version "1.8.0_432"
OpenJDK Runtime Environment Corretto-8.432.06.1 (build 1.8.0_432-b06)
OpenJDK 64-Bit Server VM Corretto-8.432.06.1 (build 25.432-b06, mixed mode)
```

Create the Application

We've assembled both Maven and Java into our EC2 instance. Now let's cut straight to generating the web app!

In this step, you're going to:

- Run **Maven** commands in your terminal to generate a Java web app.
- Use **mvn** to generate a Java web app. To do this, use these commands:

```
mvn archetype:generate \
```

```
    -DgroupId=com.nextwork.app \
```

```
    -DartifactId=nextwork-web-project \
```

```
    -DarchetypeArtifactId=maven-archetype-webapp \
```

```
    -DinteractiveMode=false
```

- Watch out for a **BUILD SUCCESS** message in your terminal once your application is all set up.

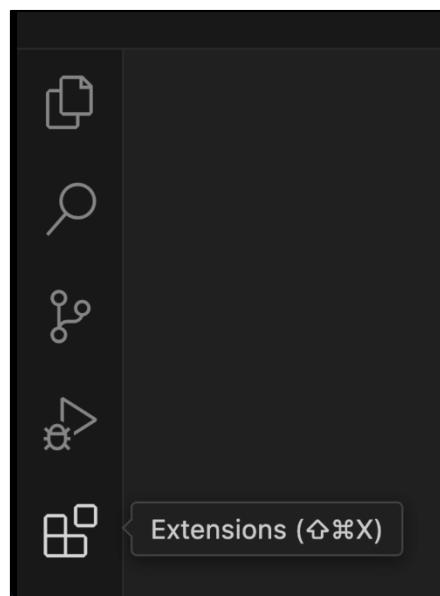
```
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/ec2-user
[INFO] Parameter: package, Value: com.nextwork.app
[INFO] Parameter: groupId, Value: com.nextwork.app
[INFO] Parameter: 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: 8.536 s
[INFO] Finished at: 2025-01-02T10:19:40Z
[INFO] Final Memory: 16M/92M
[INFO] -----
[ec2-user@ip-172-31-4-219 ~]$ ]
```

Connect VSCode with your EC2 Instance

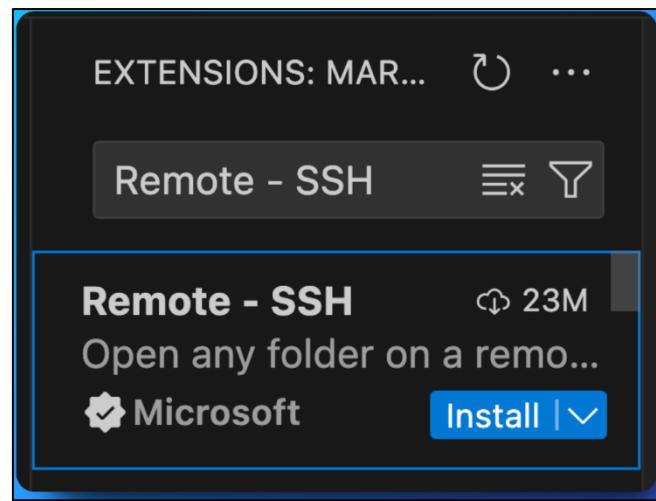
In this step, you'll connect VSCode to your EC2 instance so you can see and edit the web app you've just created.

In this step, you're going to:

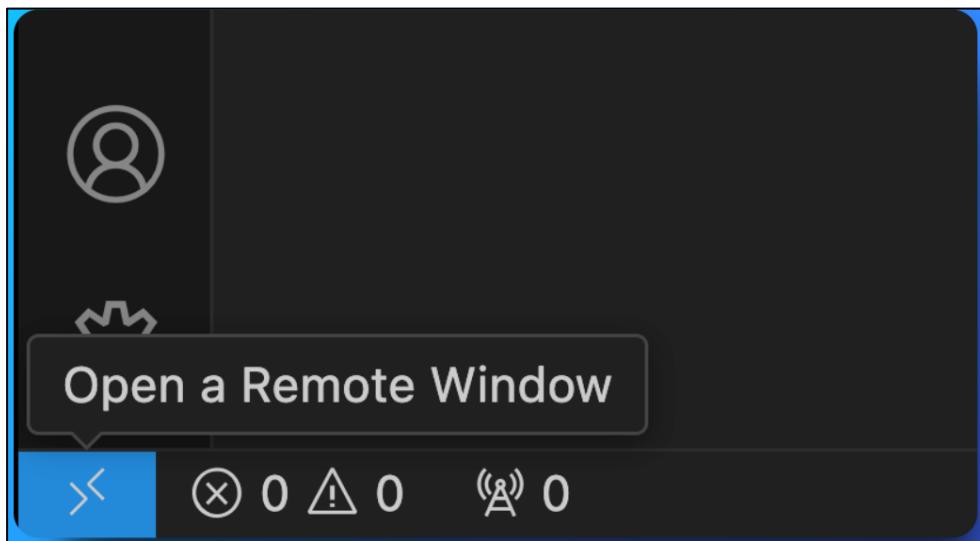
- Install an extension in VSCode.
 - Use the extension to set up a connection between VSCode and your EC2 instance.
 - Explore and edit your Java web app's files using VSCode.
-
- Clicking on the **Extensions** icon at the side of your VSCode window.



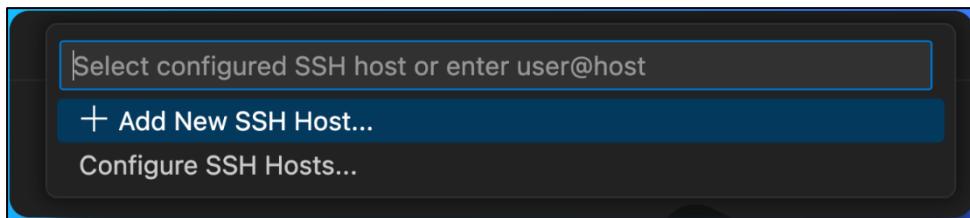
- In the search bar, type Remote - SSH and click **Install** for the extension.



- Click on the double arrow icon at the bottom left corner of your VSCode window. This button is a shortcut to use Remote - SSH.



- Select **Remote-SSH: Connect to Host...**
- Select **+ Add New SSH Host...**

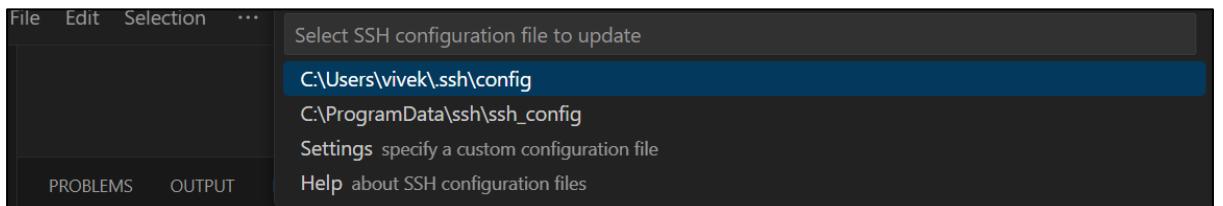


- Enter the SSH command you used to connect to your EC2 instance:
`ssh -i [PATH TO YOUR .PEM FILE] ec2-user@[YOUR PUBLIC IPV4 DNS]`
- Replace **[PATH TO YOUR .PEM FILE]** with the actual path to your private key file (e.g., ~/Desktop/DevOps/nextwork-keypair.pem). Delete the square brackets!

- Replace [YOUR PUBLIC IPV4 DNS] with the Public DNS you just found. Delete the square brackets!

Eg: ssh -i C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem ec2-user@ ec2-13-235-86-185.ap-south-1.compute.amazonaws.com

- Select the configuration file at the top of your window. It should look similar to /Users/username/.ssh/config



- A **Host added!** popup will confirm that you've set up your SSH Host - yay!
- Select the blue **Open Config** button on that popup.
- Confirm that all the details in your configuration file look correct:
 - **Host** should match up with your EC2 instance's IPv4 DNS.
 - **IdentityFile** should match up to nextwork-keypair.pem's location in your local computer.
 - **User** should say ec2-user

```
C: > Users > vivek > .ssh > config
1 Host ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
2 HostName ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
3 IdentityFile C:UsersvivekDesktopDevOpsnextwork-keypair.pem
4 User ec2-user
5
```

The screenshot shows a code editor window with a dark theme. The title bar says 'config'. The content of the file is as follows:

```
C: > Users > vivek > .ssh > config
1 Host ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
2 HostName ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
3 IdentityFile C:UsersvivekDesktopDevOpsnextwork-keypair.pem
4 User ec2-user
5
```

- As you can see the Identity File Path is not correct, edit the config Identity File Path

From

C:UsersvivekDesktopDevOpsnextwork-keypair.pem

to

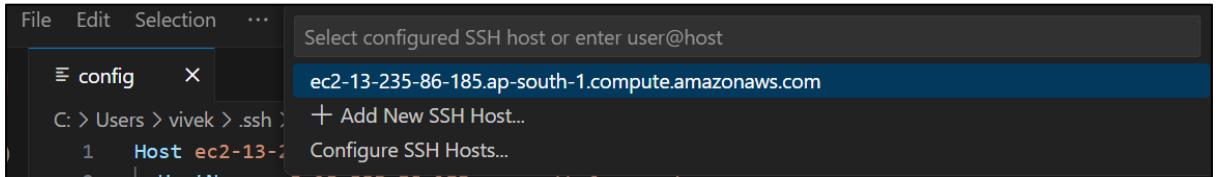
C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem

```
C: > Users > vivek > .ssh > config
1 Host ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
2 HostName ec2-13-235-86-185.ap-south-1.compute.amazonaws.com
3 IdentityFile C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem
4 User ec2-user
5
```

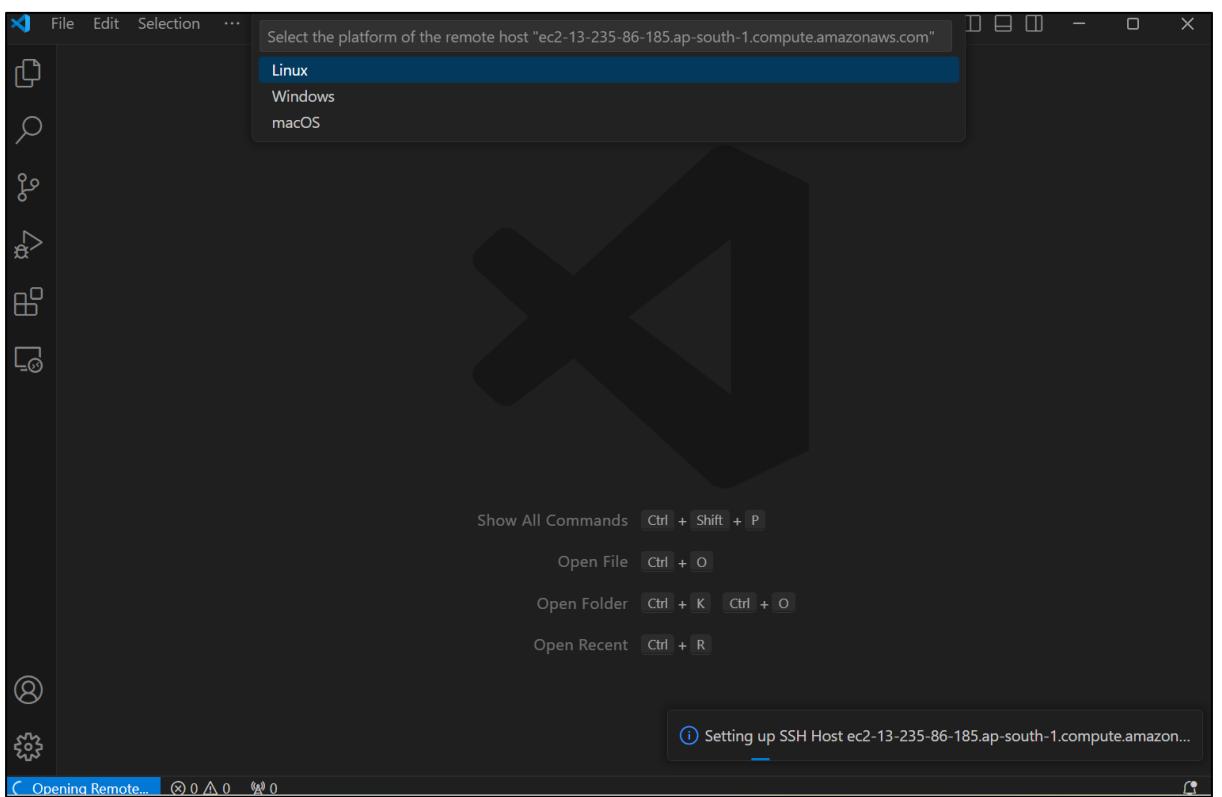
The screenshot shows the same code editor window as before, but the 'IdentityFile' line has been edited. The 'IdentityFile' line now contains the correct path: 'C:\Users\vivek\Desktop\DevOps\nextwork-keypair.pem'. The rest of the configuration file remains the same.

Then Press Ctrl+S to save the config.

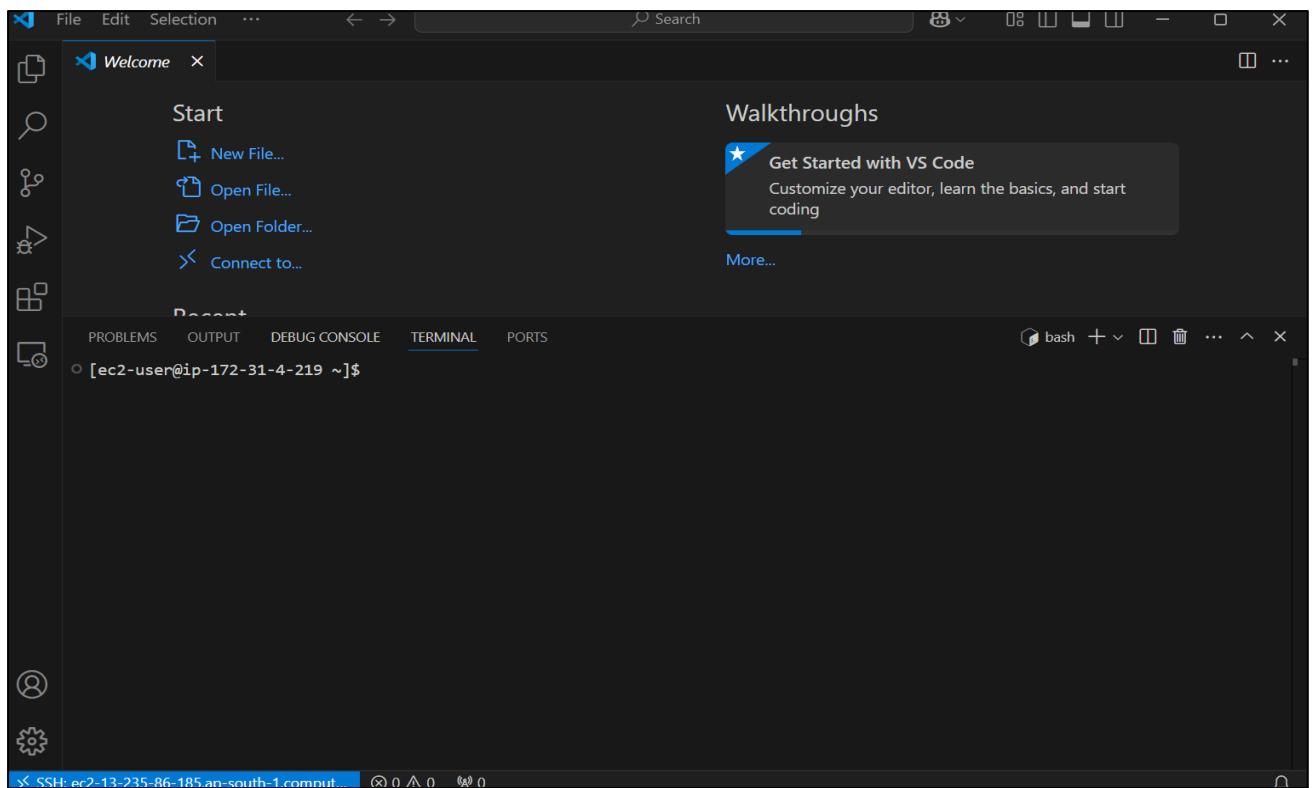
- Now you're ready to connect VSCode with your EC2 instance!
- Click on the double arrow button on the bottom left corner and select **Connect to Host** again.
- You should now see your EC2 instance listed at the top.



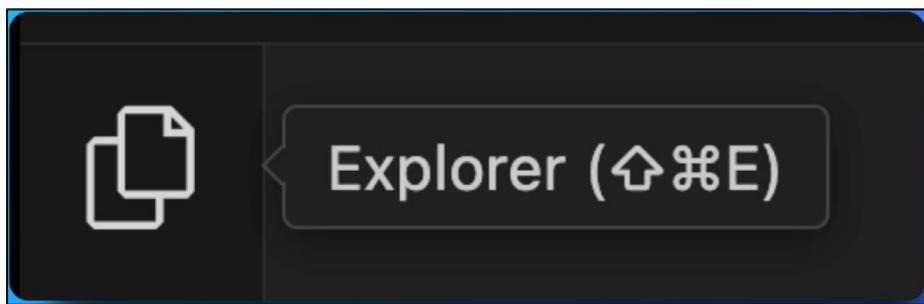
- Select the EC2 instance and off we go to a new VSCode window
- Check the bottom right hand corner of your new VSCode window - it should show your EC2 instance's IPV4 DNS.



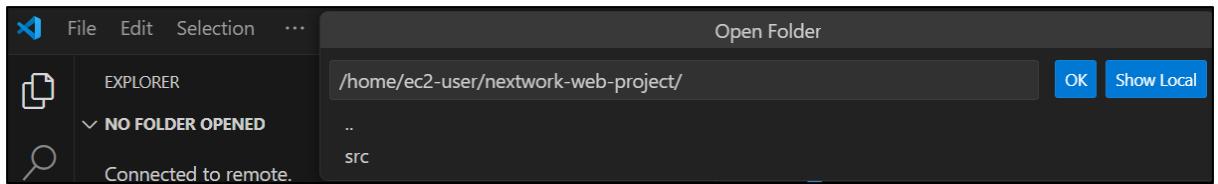
- Select **Linux** and then press **Continue**
- you've connected VSCode with your EC2 instance!



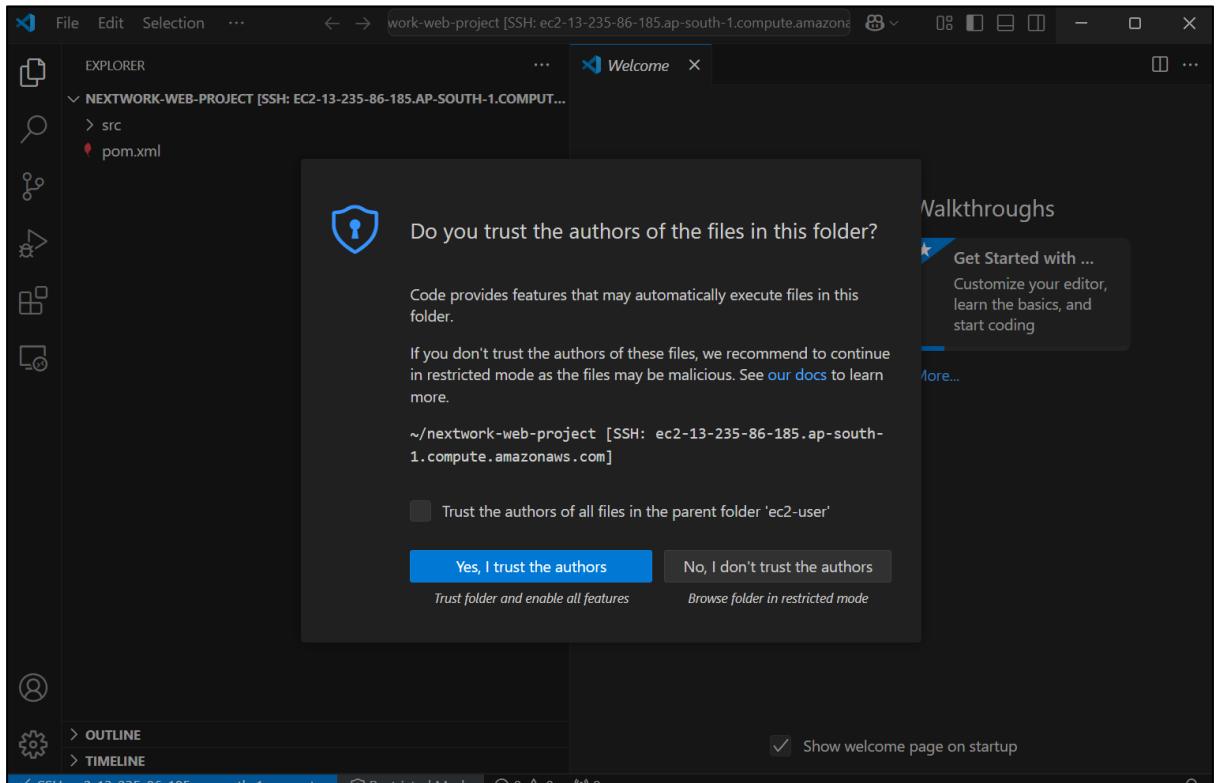
- Now let's open up your web app's files.
- From VSCode's left hand navigation bar, select the **Explorer** icon.



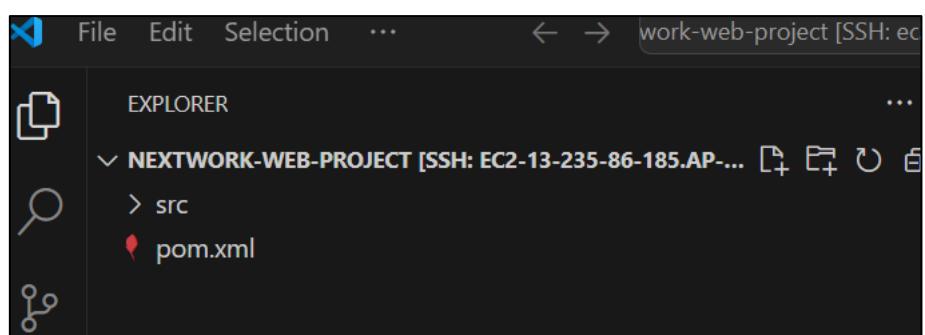
- Select **Open folder**.
- At the top of your VSCode window, you should see a drop down of different file and folder names. Ooooo, this is VSCode asking you which specific file/folder you'd like to open!
- Enter /home/ec2-user/nextwork-web-project.
- Press **OK**.



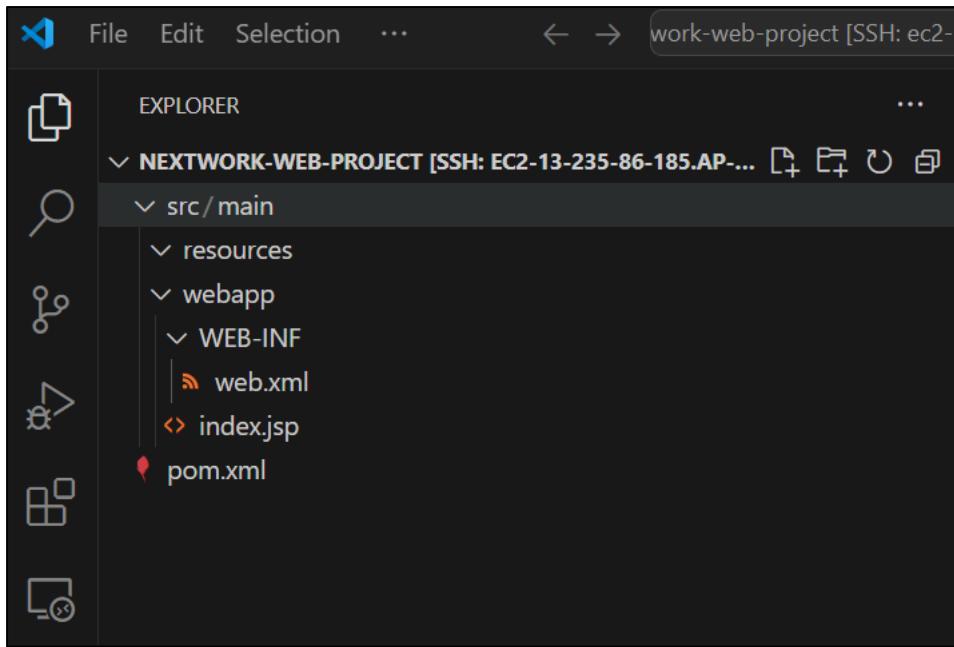
- VSCode might show you a popup asking if you trust the authors of the files in this folder. If you see this popup, select **Yes, I trust the authors**.



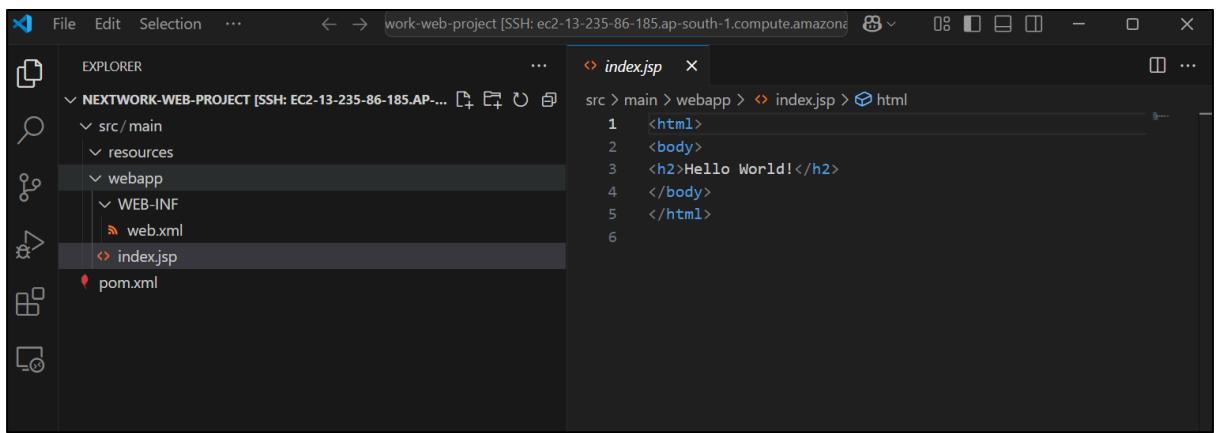
- Check your VSCode window's file explorer again - a folder called **nextwork-web-project** is here!



- Try expanding all the subfolders in the file explorer. All folders have a > icon next to their name.



- From your file explorer, click into **index.jsp**.



- **index.jsp** is a file used in Java web apps. It's similar to an HTML file because it contains markup to display web pages.

However, index.jsp can also include Java code, which lets it generate dynamic content.

This means content can change depending on things like user input or data from a database. Social media apps are great examples of web apps because the content you see is always changing, updating and personalised to you. HTML files are static and can't include Java code. That's why it's so important to install Java in your EC2 instance - so you can run the Java code in your web app!

- Welcome to editor view of index.jsp. Now we're really using VSCode's IDE abilities - editing code is much easier here than in the terminal.
- Let's try modifying **index.jsp** by changing the placeholder code to the code snippet below. Don't forget to replace **{YOUR NAME}** from the following code with your name:

```
<html>
<body>
<h2>Hello {YOUR NAME}!</h2>
<p>This is my NextWork web application working!</p>
</body>
</html>
```

- Save the changes you've made to **index.jsp** by selecting **Command/Ctrl + S** on your keyboard.
- Nice work - you've just set up a web app on an EC2 instance AND connect it with VSCode, one of the most popular and practical IDEs out there.

Delete Your Resources

- In your AWS Management Console, head to **Amazon EC2** to delete your EC2 instance.
- Select **Instances** from the left hand navigation panel. Select the checkbox next to your instance.
- Click on the **Instance state** dropdown and select **Terminate (delete) instance**.
- Choose **Terminate (delete)**.

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows 'Instances' is selected. The main pane displays a table of instances with one row selected. The selected instance is named 'nextwork-dev-ops-vivek' with the ID 'i-0ada190ec4b40e4ce'. Its status is 'Running'. The 'Actions' dropdown menu is open, and the 'Terminate (delete)' option is highlighted.

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows 'Instances' is selected. The main pane displays a table of instances with one row selected. The selected instance is named 'nextwork-dev-ops-vivek' with the ID 'i-0ada190ec4b40e4ce'. Its status is now 'Terminated'. The 'Actions' dropdown menu is closed.