Set Up a Web App in the Cloud

In this step, you're going to:

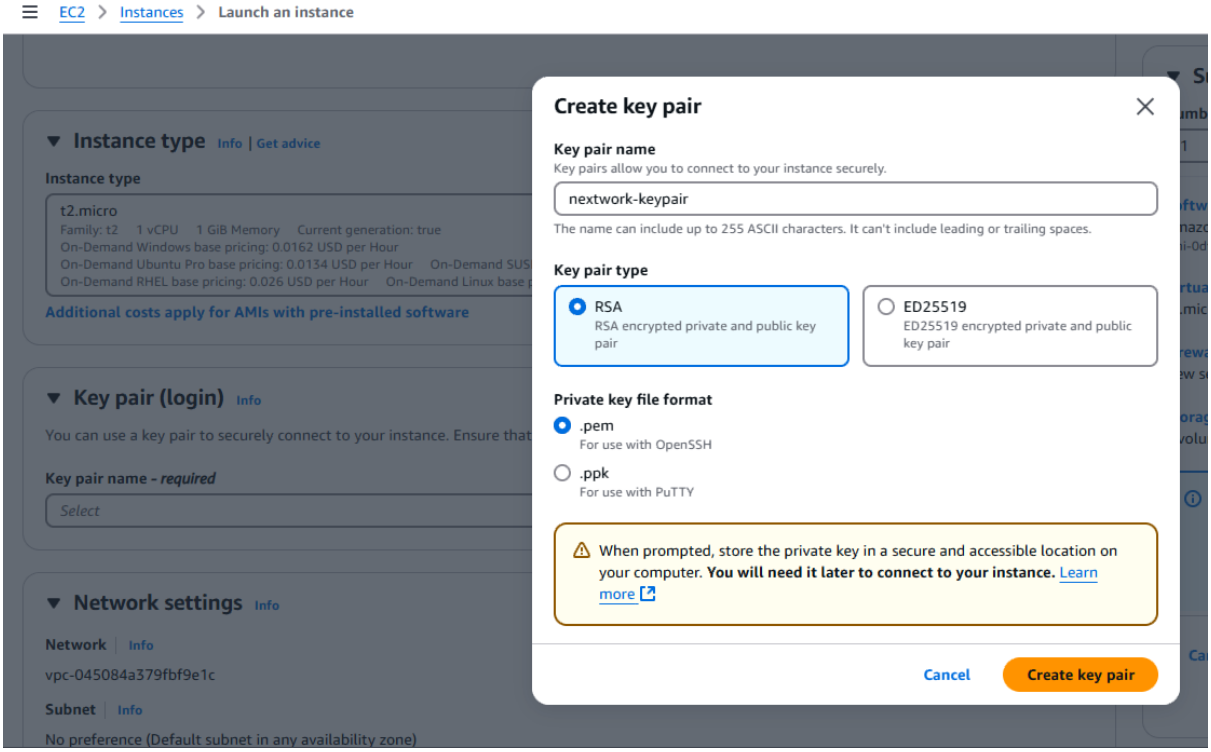
1. Launch a new EC2 instance.
2. Set up a key pair for secure access.
3. Set up network settings for your instance.

Log In with your IAM Admin User.

Head to **Amazon EC2** in your AWS Management Console.

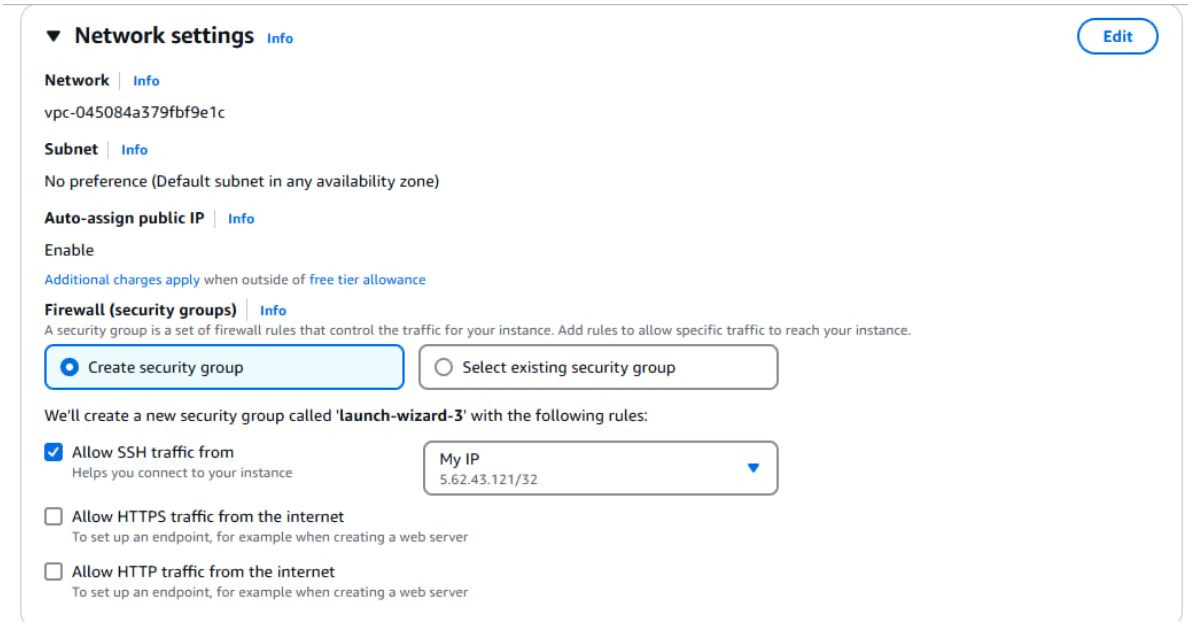
Switch your **Region** to the one closest to you.

- In your EC2 console, select **Instances** from the left-hand navigation panel.
- Choose **Launch instances**.
- Let's set up your EC2 instance.
- In **Name**, enter the value nextwork-devops-yourname.
    - Don't forget to replace **yourname** with your name!
- Choose **Amazon Linux 2023 AMI** under **Amazon Machine Image (AMI)**.
- Leave **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**
- Select **Create key pair.**

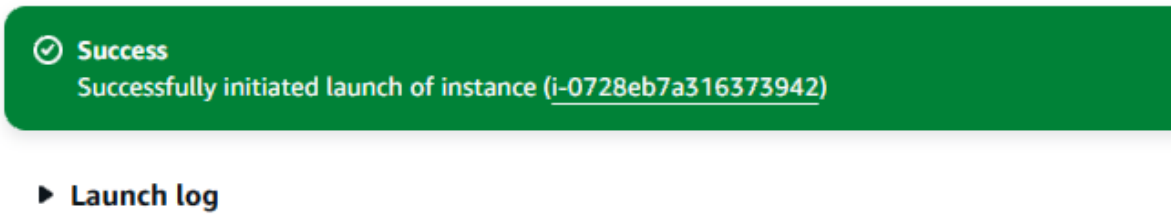


- A new file will automatically download to your local computer. This is your private key.
- Before we lose track of our .pem file, let's organise it in our computer.

- Head to your local computer's desktop.
- Create a new folder 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.
- Double check that the IP address under **My IP** is correct.
- 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



- When you're ready, choose **Launch instance**.



Open VSCode and navigate your terminal to the DevOps folder. You'll do this by entering this command in the terminal:

cd ~/Downloads/DevOps

Change the permissions of your .pem file:

In the terminal, run the following command to allow access to your .pem file.

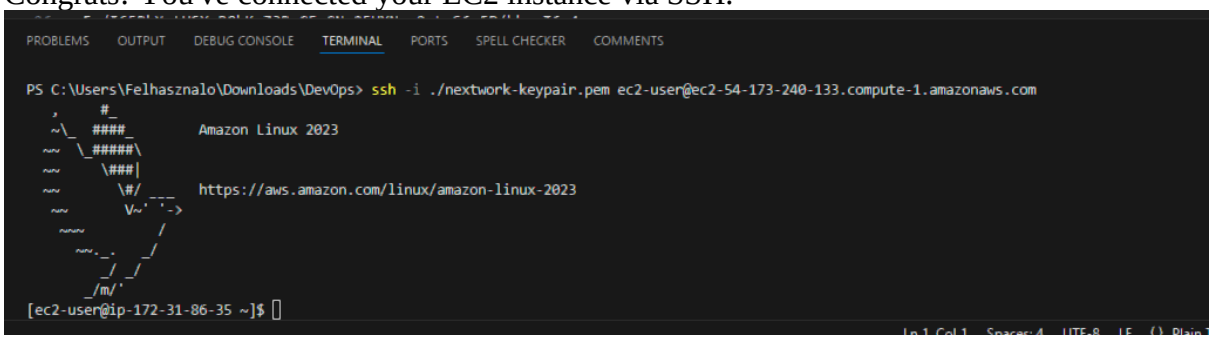
icacls "nextwork-keypair.pem" /reset

icacls "nextwork-keypair.pem" /grant:r "USERNAME:R"

icacls "nextwork-keypair.pem" /inheritance:r

Make sure to double check that the file name in your command i.e. **nextwork-keypair.pem** matches the file in your DevOps folder.

Connect to your EC2 Instance

- Head back to your AWS Management Console.
- Click on **Instances** from the left-hand navigation panel.
- Click on the checkbox next to your EC2 instance to view its details.
- Under the **Details** tab, look for **Public IPv4 DNS**.
- Now we'll connect to our instance via SSH.
- Head back to VSCode and open your terminal again.
- Use the following command 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
- Your terminal will ask if you want to continue connecting to this EC2 instance. This is SSH's way of asking if you trust this server.
- Enter yes to continue connecting.
- Congrats! You've connected your EC2 instance via SSH.
- 

Install Apache Maven and Amazon Corretto 8

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

- Now we're going to install Java 8, or more specifically, Amazon Correto 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

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

- java -version

Create the Application

In this step, you're going to:

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

mvn archetype:generate \

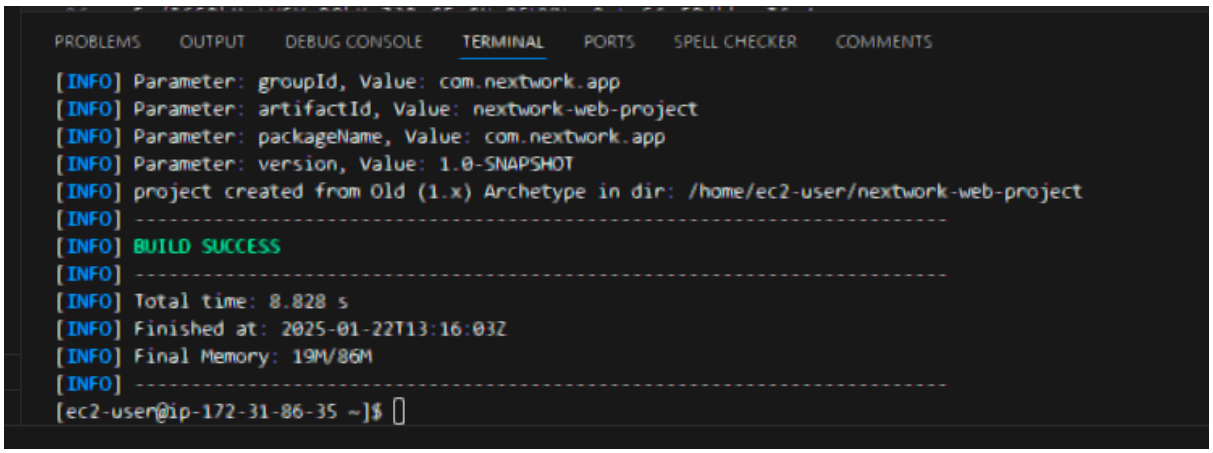
-DgroupId=com.nextwork.app \

-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.



Clicking on the **Extensions** icon at the side of your VSCode window.

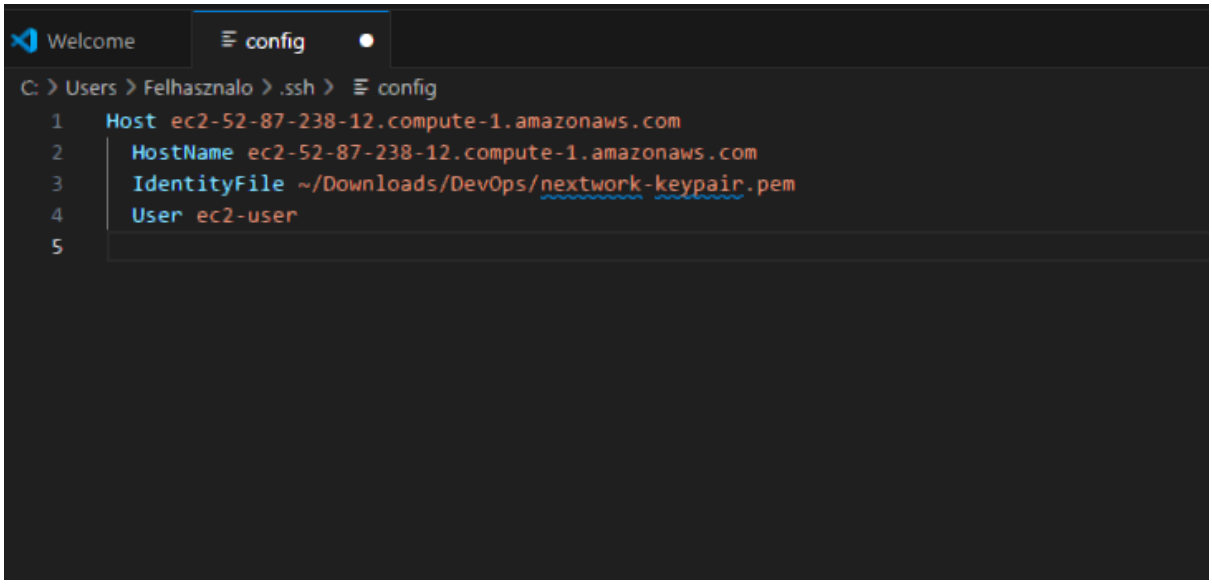
In the search bar, type Remote - SSH and click **Install** for the extension.

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...**

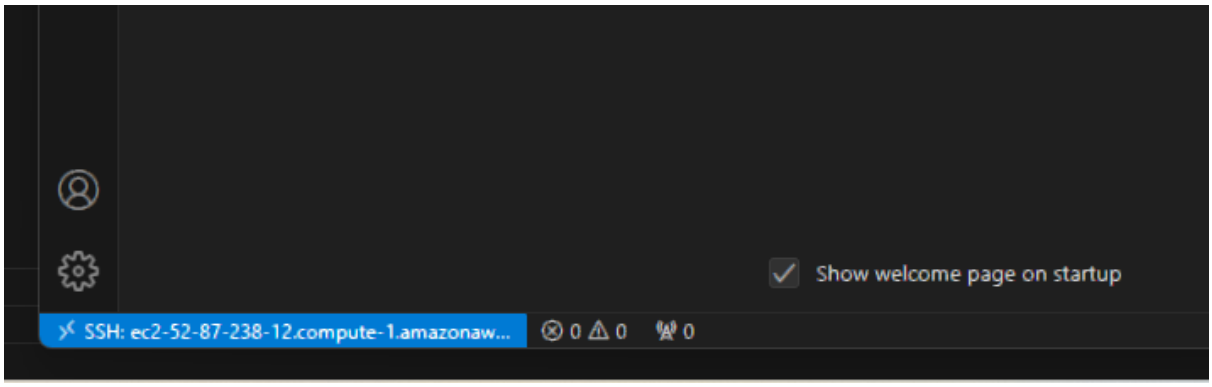
Enter the SSH command you used to connect to your EC2 instance: ssh -i [PATH TO YOUR .PEM FILE] ec2-user@[YOUR PUBLIC IPV4 DNS]

- 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
- 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
- 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



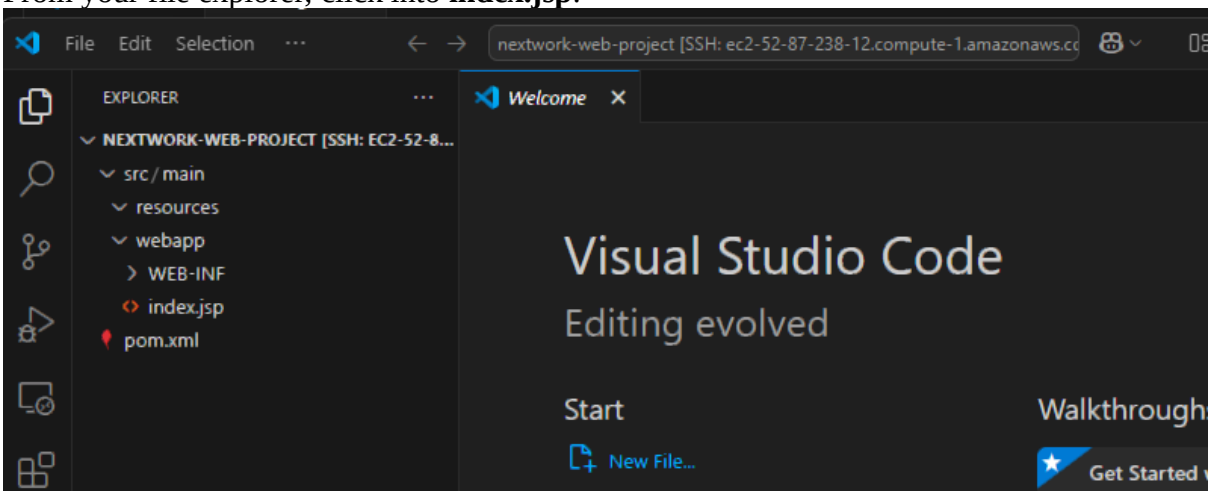
- Now you're ready to connect VSCode with your EC2 instance!
- Click on the double arrow button on the bottom left corner and select **Connect to Host** again.
- You should now see your EC2 instance listed at the top.
- Select the EC2 instance and off we go to a new VSCode window

- Check the bottom right-hand corner of your new VSCode window - it should show your EC2 instance's IPV4 DNS.



Nice work - you've connected VSCode with your EC2 instance.

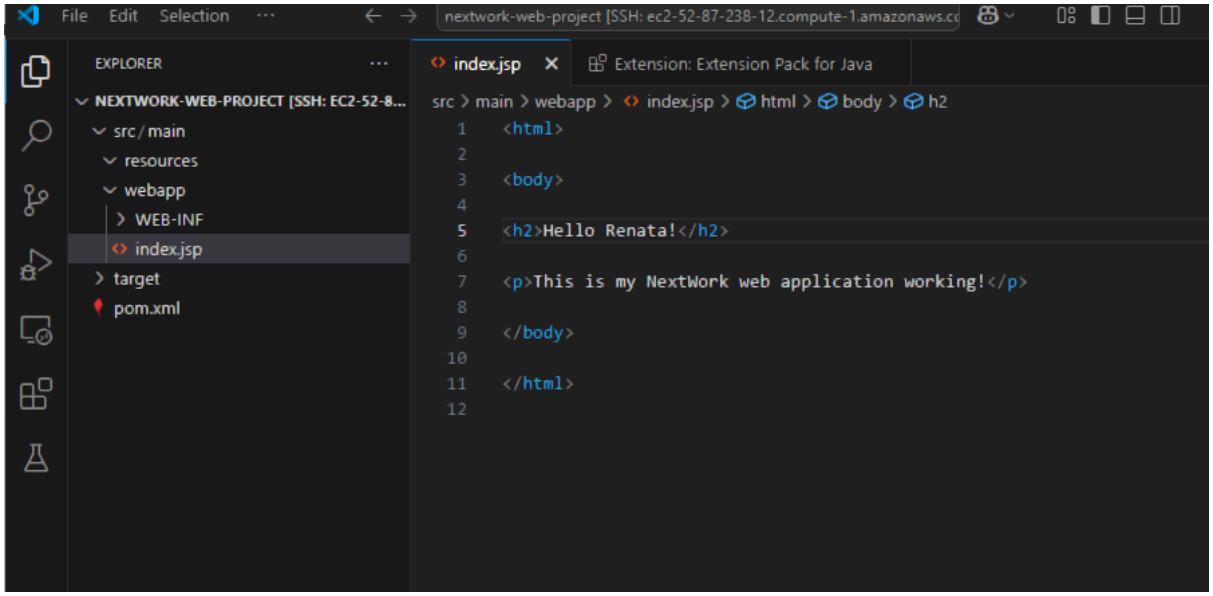
Now let's open up your web app's files.

- 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.
- Exploring done! So how can VSCode help you edit your application files? Let's find out.
- From your file explorer, click into **index.jsp**.
- 

- 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.