PHASE 1

Setting up a webapp in AWS

OPTION A: IF APP IS ALREADY AVAILABLE IN REPO

Create an EC2 instance

Git clone the repo of web app (if already done in local)

Create a venv and install dependencies using requirements.txt

Check if the app is running : configure security group to allow Custom TCP in port 5000. Open run.py and edit as app.run(host="0.0.0.0",port=5000,debug=True)

(Note: Follow the below steps if creating everything in EC2 Instance.)

OPTION B: Setting up the environment in EC2 itself

STEP 1 : Basic Setup

NOTES:

Key Pair:

> A key pair in EC2 is like the keys to your virtual computer. Just like you need a key to unlock and start your car, a key pair lets you securely access your EC2 instance.

> It's made of two halves: a public key that AWS keeps, and a private key that you download.
>
> When you use the private key, it verifies that you're the one allowed to access that specific virtual machine, keeping everything secure and just for you.

Key pair type and RSA:

> The key pair type determines the algorithm used for generating the key pair's cryptographic keys.
>
> We use RSA (Rivest-Shamir-Adleman), which is one of the most common cryptographic algorithms used due to its strength and security. RSA is widely supported and trusted for creating digital signatures and encrypting data.

Private Key and .pem format:

> Just like how documents can be saved in different formats (e.g. PDF, DOCX, or TXT) for different apps or systems, private keys also come in different file formats. Not every system or application can process all these formats, so choosing the right one is crucial.
>
> The **.pem** format stands for **Privacy Enhanced Mail**. It started off as a way to secure emails and has become the go-to format for cryptographic keys because it works with many different types of servers e.g. EC2 instances!

SSH:

> SSH i.e. Secure Shell is a protocol used to make sure only authorized users can access a remote server. When you connect to your EC2 instance later in this project, SSH verifies you have the correct private key that matches the public key on the server.
>
> SSH is also a type of network traffic. Once SSH has authorized you, it'll set up a secure connection between you and the EC2 instance. All data transferred (including your commands and the responses from the instance) gets encrypted. This encryption makes SSH an ideal method for working with virtual servers!

- Launch an EC2 Instance.
- Open VsCode and connect to EC2 instance
- For windows:
  - icacls "nextwork-keypair.pem" /reset
  - icacls "nextwork-keypair.pem" /grant:r "USERNAME:R”
  - For above one replace username by checking in powershell - $env:USERNAME
  - icacls "nextwork-keypair.pem" /inheritance:r
  - Then - ssh -i keypair.pem ec2-user@ec2_public_ip

- Install Apache Maven and Amazon Corretto 8 on EC2 to build Java web apps

NOTE:

Apache Maven:

> Apache Maven is a tool that helps developers build and organize Java software projects. It's also a package manager, which means it automatically download any external pieces of code your project depends on to work.
>
> We're also using Maven today because it's really useful for kick-starting web projects! It uses something called **archetypes**, which are like templates, to lay out the foundations for different types of projects e.g. web apps.
>
> We'll use Maven later on to help us set up all the necessary web files to create a web app structure, so we can jump straight into the fun part of developing the web app sooner.

- Download Apache Maven- wget https://archive.apache.org/dist/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz
- Extract - sudo tar -xzf apache-maven-3.5.2-bin.tar.gz -C /opt
- Add to PATH - echo "export PATH=/opt/apache-maven-3.5.2/bin:$PATH" >> ~/.bashrc
- source ~/.bashrc

Amazon Corretto 8(Java):

> Java is a popular programming language used to build different types of applications, from mobile apps to large enterprise

> systems.
>
> Maven, which we just downloaded, is a tool that NEEDS Java to operate. So if we don't install Java, we won't be able to use Maven to generate/build our web app today.
>
> Amazon Corretto 8 is a version of Java that we're using for this project. It's free, reliable and provided by Amazon.

- Download - sudo dnf install -y java-1.8.0-amazon-corretto-devel
- Tells EC2 instance where java is located- export JAVA_HOME=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64
- To run java commands from any location of EC2- export PATH=/usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre/bin/:$PATH

- To see if Maven is installed correctly- mvn -v

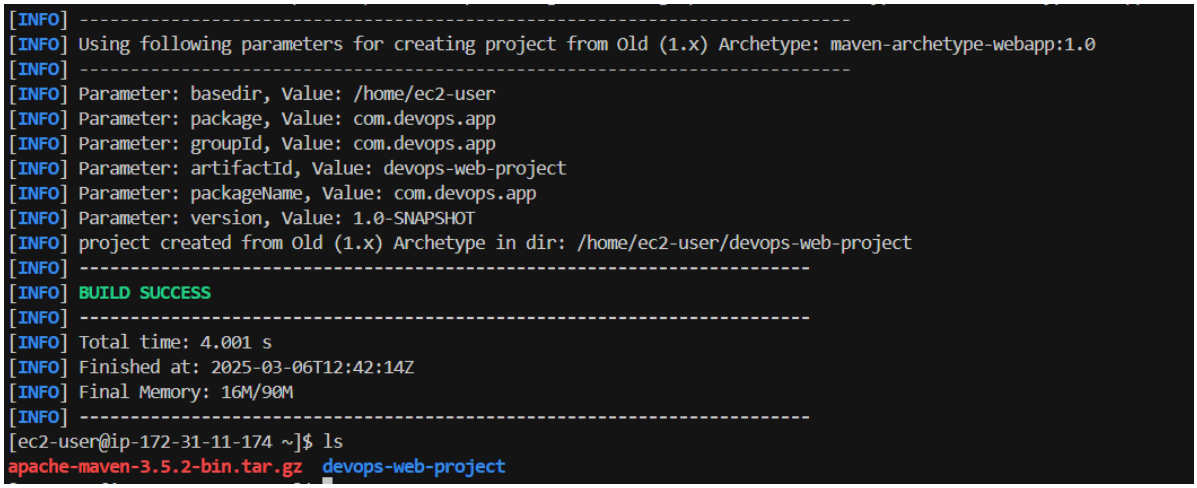
- To change java version - sudo alternatives --config java

STEP 2 : Application Creation

- Run Maven commands in terminal :

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

- **mvn archetype:generate -** Create new Maven project from a template called archetype . It sets up a basic structure for the project.
- **-DartifactId=devops-web-project - Name of project**
- **-DarchetypeArtifactId=maven-archetype-webapp - Specifies that you are creating the web application.**
- **-DinteractiveMode=false -** runs the command without pausing for user input, so Maven will go ahead and install everything without waiting for your confirmation.

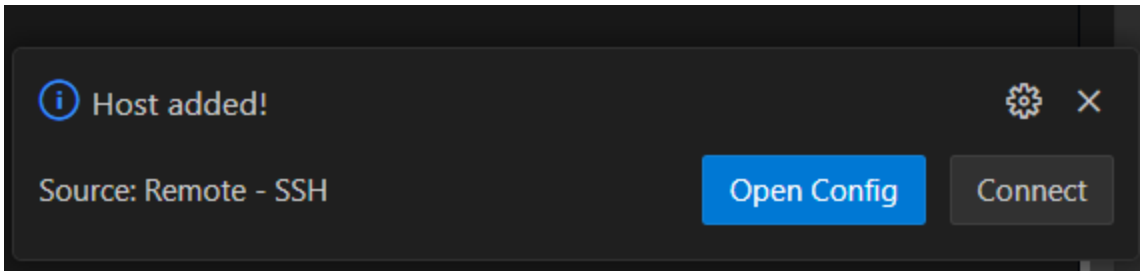


STEP 3 : Connect VSCode with EC2 instance

NOTE:

> Connecting with SSH in VSCode's terminal lets you send text commands to the EC2 instance, but to get all the benefits of VSCode (like file navigation and code editing directly on EC2 instance). Makes it easier to edit and manage the web app.

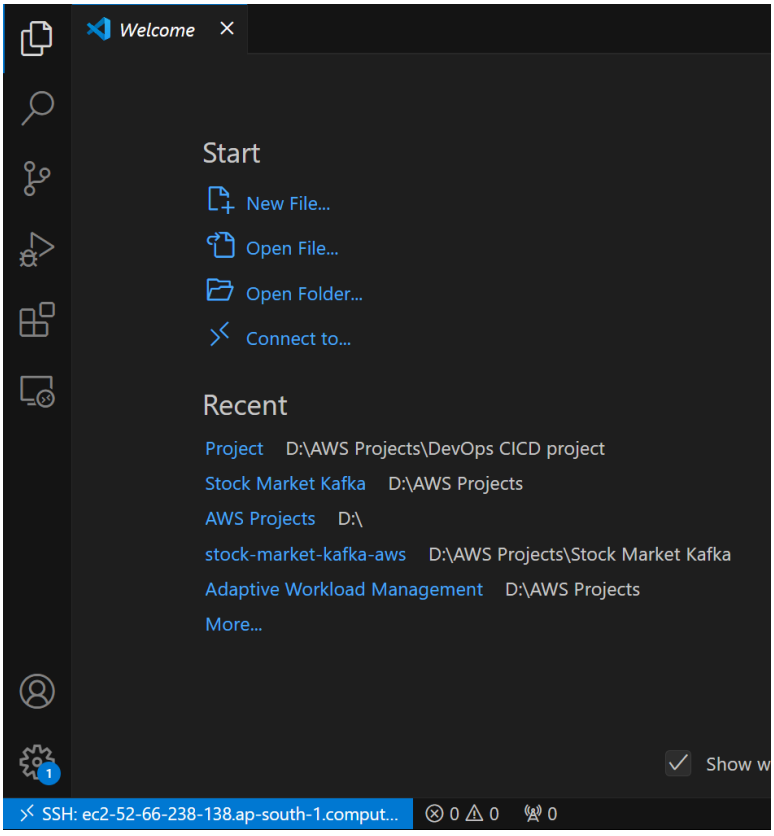
- Go to Extensions in VSCode
- Select Remote-SSH extension and install
    - The **Remote - SSH extension** in VSCode lets you connect directly via SSH to another computer securely over the internet. This lets you use VSCode to work on files or run programs on that server as if you were doing it on your own computer.
- Click on double arrow icon on bottom left of VSCode to open a remote window
- Select **Remote-SSH: Connect to Host...**
- Select **+ Add New SSH Host...**
- Enter the ssh command to connect to EC2 in the text field and press enter
- Choose the config file to get this:



- Click on Open Config to get:

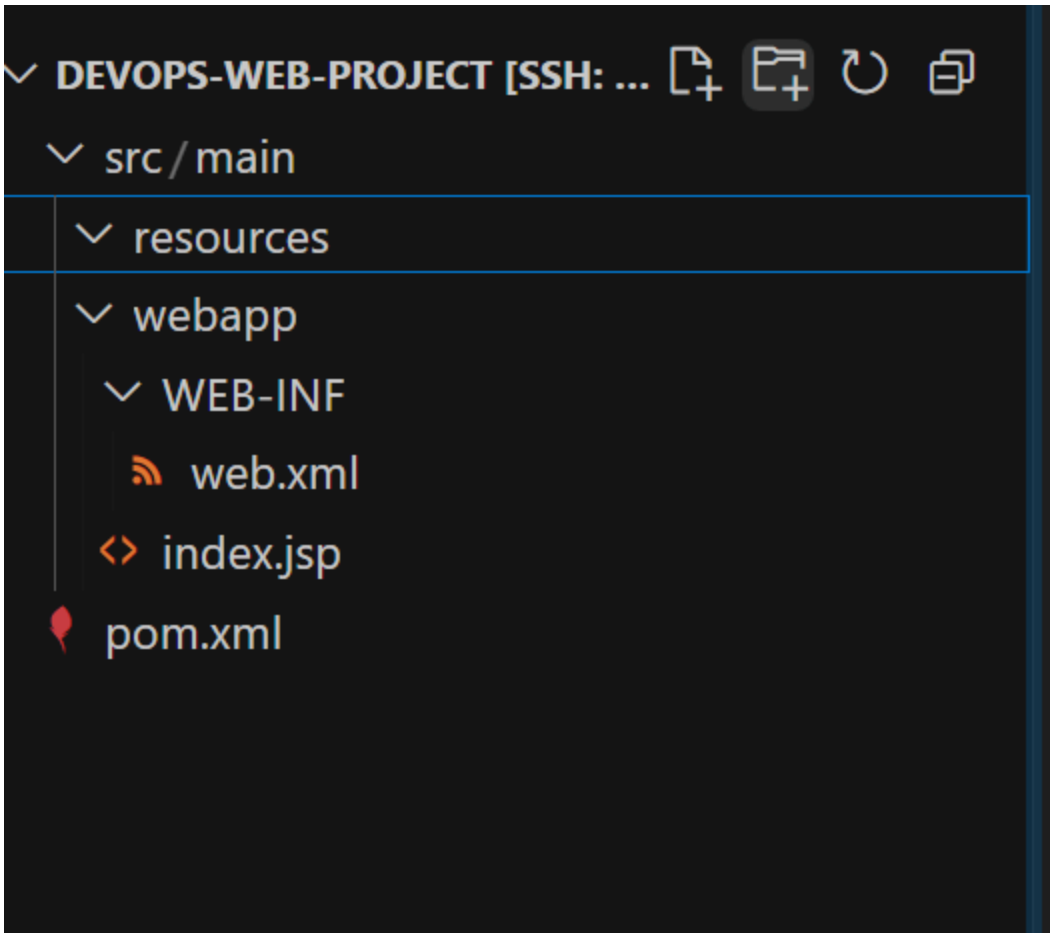
```
Host ec2-52-66-238-138.ap-south-1.compute.amazonaws.com

  HostName ec2-52-66-238-138.ap-south-1.compute.amazonaws.com
  IdentityFile "D:/AWS Projects/DevOps CICD project/rsa-keypair.pem"
  User ec2-user
```

- Make sure the provided details are correct
- Then connect to Host and choose Linux→continue
- You can see the ec2 ip at bottm left corner in the new VSCode window



- Open Explorer and Open folder and paste this - /home/ec2-user/devops-web-project

- Ensure to change to the project name that you have created



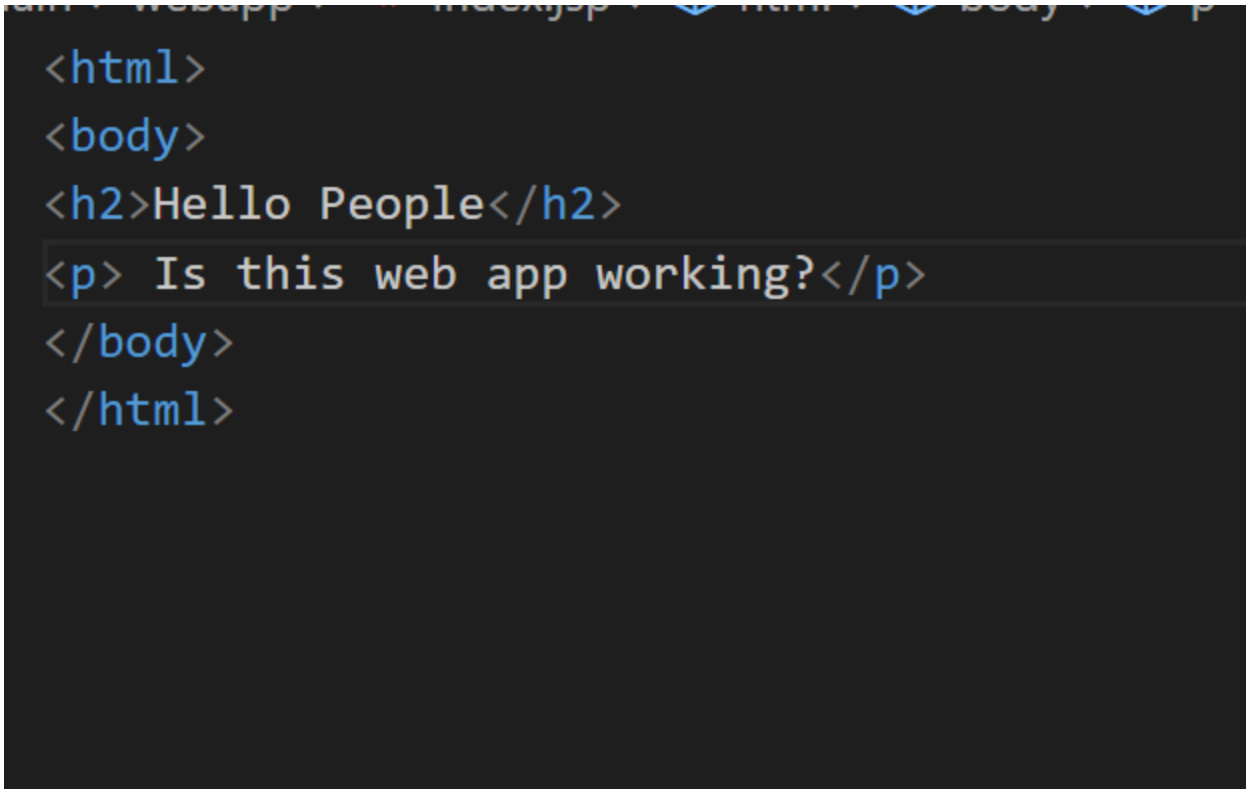
NOTE:

- The src (source) folder holds all the source code files that define how your web app looks and works.
- src is further divided into webapp, which are the web app's files e.g. HTML, CSS, JavaScript, and JSP files, and resources, which are the configuration files a web app might need e.g. connection settings to a database.
- pom.xml is a Maven **Project Object Model** file. It stores information and configuration details that Maven will use to build the project.

- Open index.jsp

NOTE:

> 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!



STEP 4 : Install Git on EC2

- In EC2's terminal enter the below commands:
  - To update to latest versions of installed software(best practice to update before installing new softwares to avoid compatibility issues) - sudo dnf update -y
  - To install git- sudo dnf install git -y
  - To check git installation - git —version
  - Create a repo in github
  - Make sure you are in the ssh connected terminal
  - Connect to it by:

- Create local repo - git init
- Type - git remote add origin [YOUR GITHUB REPO LINK]
- For the above link use ssh to avoid complexities like username and password issues that comes later
- Type - git add .
- Type- git commit -m “Updated index.jsp bro!!!”
- Finally type - git push -u origin master
- To check if git is connected to local repo - git remote -v
- Check if you are authenticated - git ls-remote origin
- To see what changes are staged - git diff --staged
- Error happens when pushing to master

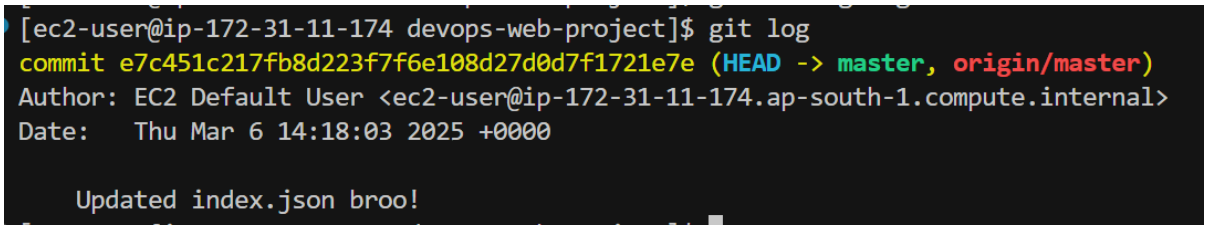
```
[ec2-user@ip-172-31-11-174 devops-web-project]$  git push -u origin master
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

- Follow these steps:
- -la ~/.ssh - if no id_rsa (private ssh key is present follow you will get the above error)
- ssh-keygen -t rsa -b 4096 -C "your-email@example.com" - Enter gmail asoociated with the created github repo
- When asked to enter some file location just keep pressing Enter for default
- cat ~/.ssh/id_rsa.pub - to get public key
- Copy the above output and go to github
    - Go to settings→SSH and GPS keys→New SSH key→Paste the copied key →Click Add SSH key

- Test the SSH Connection - ssh -T git@github.com
- If successful:

  Hi [Your username]! You've successfully authenticated, but GitHub does not provide shell access.
- Now try git push again

- To see commit logs - git log



- **EC2 Default User** isn't really your name, and the EC2 instance's IPv4 DNS is not your email. Let's configure your local Git identity so Git isn't using these default values
- Run these commands in your terminal to manually set your name and email. Don't forget to replace **"Your name"** with your name (keep the quote marks), and **you@mail.com** with your email address.
- git config --global user.name "Your Name"
- git config --global user.email you@mail.com

_____PHASE 1 COMPLETED_____