Command to run docker file:

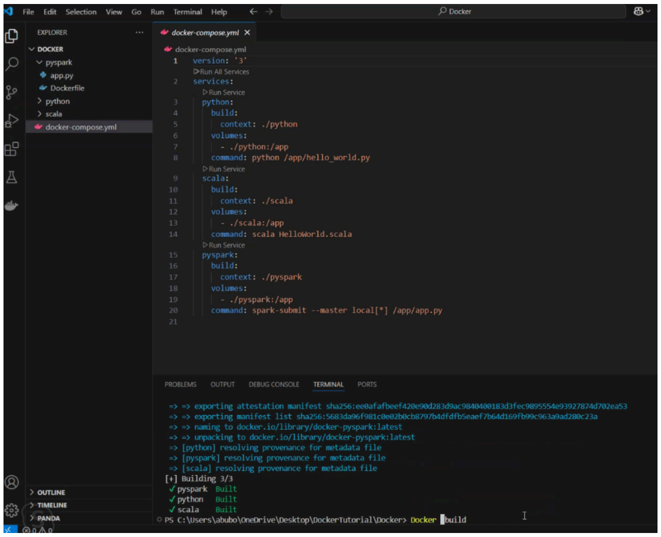
Got the docker file from github and worked on the file via vs Code.

To begin you need to build the image using the build command (shown below) you can name it as you like. Please ensure you are in the actual **directory** before you build the docker image. Building the image allows you to set up all the dependencies, before you can run your code on it.

Depending on your desired code, being pyspark, python or scala, you go in that directory. For now I will be using the example as python. The python folder will have the python '.py' file and its own docker file. The python file name is hello_world.py

I have created the hello_world.py file and now I will go inside the docker file where I have to specify the name of the file and then i have to go to the terminal to run the following commands

1. **docker build -t *name of the file* .** (This command is the first step and it will build the image. The *dot* at the end means to run from the current directory you are in. You can name the Image whatever you want since you are in the python directory it knows what files to run since you have already mentioned it inside the docker file.)
2. **Docker run hello_world** (this will run the image and will show us the outcome)
3. **Docker ps -a** (lists all the containers that are running or were running)



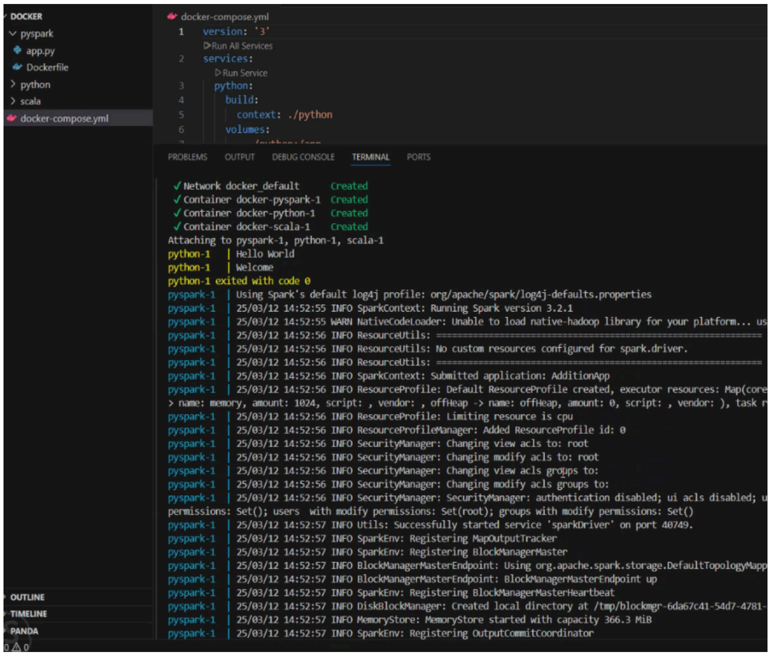
```
version: '3'
services:
  python:
    build:
      context: ./python
    volumes:
      - ./python:/app
    command: python /app/hello_world.py
  scala:
    build:
      context: ./scala
    volumes:
      - ./scala:/app
    command: scala HelloWorld.scala
  pyspark:
    build:
      context: ./pyspark
    volumes:
      - ./pyspark:/app
    command: spark-submit --master local[*] /app/app.py
```

```
=> => exporting attestation manifest sha256:ee0afafbeef420e90d283d9ac9840400183d3fec9895554e93927874d702ea53
=> => exporting manifest list sha256:5683da96f981c0e02b0cb8797b4dfdfb5eaef7b64d169fb99c963a9ad280c23a
=> => naming to docker.io/library/docker-pyspark:latest
=> => unpacking to docker.io/library/docker-pyspark:latest
=> [python] resolving provenance for metadata file
=> [pyspark] resolving provenance for metadata file
=> [scala] resolving provenance for metadata file
[+] Building 3/3
 ✓ pyspark  Built
 ✓ python   Built
 ✓ scala    Built
PS C:\Users\abubo\OneDrive\Desktop\DockerTutorial\Docker> Docker build
```

This image shows the directory with all the different files. At the bottom we have the *docker-compose.yml* this file allows you to run all the files all together rather than building all of them individually. The first command that will build all the images all together is:

docker-compose build

You can see in the image below in the terminal that it has built all the files (*green colour built mentioned at the bottom*)



```yaml
version: '3'
services:
  python:
    build:
      context: ./python
    volumes:
```

```
 ✓Network docker_default      Created
 ✓Container docker-pyspark-1  Created
 ✓Container docker-python-1   Created
 ✓Container docker-scala-1    Created
Attaching to pyspark-1, python-1, scala-1
python-1   | Hello World
python-1   | Welcome
python-1 exited with code 0
pyspark-1  | Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
pyspark-1  | 25/03/12 14:52:55 INFO SparkContext: Running Spark version 3.2.1
pyspark-1  | 25/03/12 14:52:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
pyspark-1  | 25/03/12 14:52:56 INFO ResourceUtils: ==============================================================
pyspark-1  | 25/03/12 14:52:56 INFO ResourceUtils: No custom resources configured for spark.driver.
pyspark-1  | 25/03/12 14:52:56 INFO ResourceUtils: ==============================================================
pyspark-1  | 25/03/12 14:52:56 INFO SparkContext: Submitted application: AdditionApp
pyspark-1  | 25/03/12 14:52:56 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(core
> name: memory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task r
pyspark-1  | 25/03/12 14:52:56 INFO ResourceProfile: Limiting resource is cpu
pyspark-1  | 25/03/12 14:52:56 INFO ResourceProfileManager: Added ResourceProfile id: 0
pyspark-1  | 25/03/12 14:52:56 INFO SecurityManager: Changing view acls to: root
pyspark-1  | 25/03/12 14:52:56 INFO SecurityManager: Changing modify acls to: root
pyspark-1  | 25/03/12 14:52:56 INFO SecurityManager: Changing view acls groups to:
pyspark-1  | 25/03/12 14:52:56 INFO SecurityManager: Changing modify acls groups to:
pyspark-1  | 25/03/12 14:52:56 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; u
permissions: Set(); users  with modify permissions: Set(root); groups with modify permissions: Set()
pyspark-1  | 25/03/12 14:52:57 INFO Utils: Successfully started service 'sparkDriver' on port 40749.
pyspark-1  | 25/03/12 14:52:57 INFO SparkEnv: Registering MapOutputTracker
pyspark-1  | 25/03/12 14:52:57 INFO SparkEnv: Registering BlockManagerMaster
pyspark-1  | 25/03/12 14:52:57 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapp
pyspark-1  | 25/03/12 14:52:57 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
pyspark-1  | 25/03/12 14:52:57 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
pyspark-1  | 25/03/12 14:52:57 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-6da67c41-54d7-4781-
pyspark-1  | 25/03/12 14:52:57 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
pyspark-1  | 25/03/12 14:52:57 INFO SparkEnv: Registering OutputCommitCoordinator
```

Once the image has been built we need to run the image to get the results. To run the docker compose file the next command is:

docker-compose up

This will build all the files and as you can see in the image below it has created the python scripts and if we were able to scroll more down we would have been able to see the scala and pyspark results too.