Cloud Computing and Virtualisation Project Report

201805106 - Yiğit Çakmak

201805107 - Emre Yapıcı

211805003 - Taha Ahmet Ok

Project Topic: Dockerizing Apache Spark in a portable way and making it available to everyone.

How to Use Docker for This Project?

Our project was built using Apache Spark in all 3 ways, using Docker to provide convenience to the users.

Important Note on Using requirements.txt

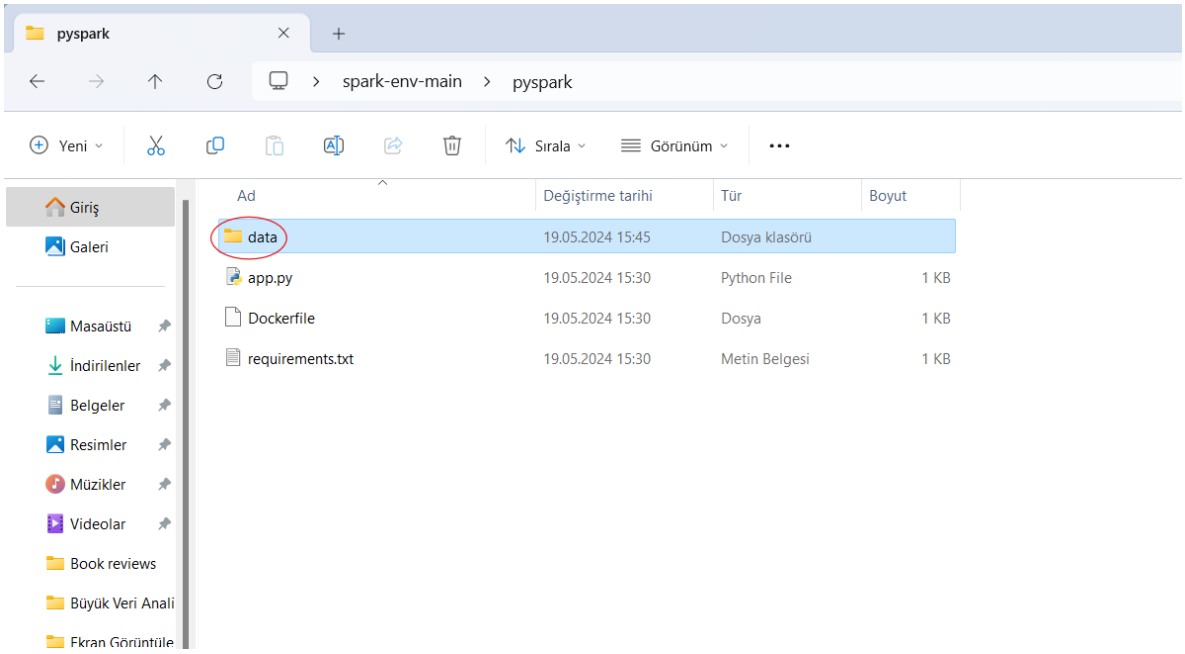
In Python projects, the requirements.txt file is used to list all the dependencies required by the project along with their specific versions. This file allows developers to ensure that the same versions of dependencies are installed in different environments, ensuring consistency and avoiding potential issues caused by version incompatibilities.

Editing requirements.txt for Custom Versions

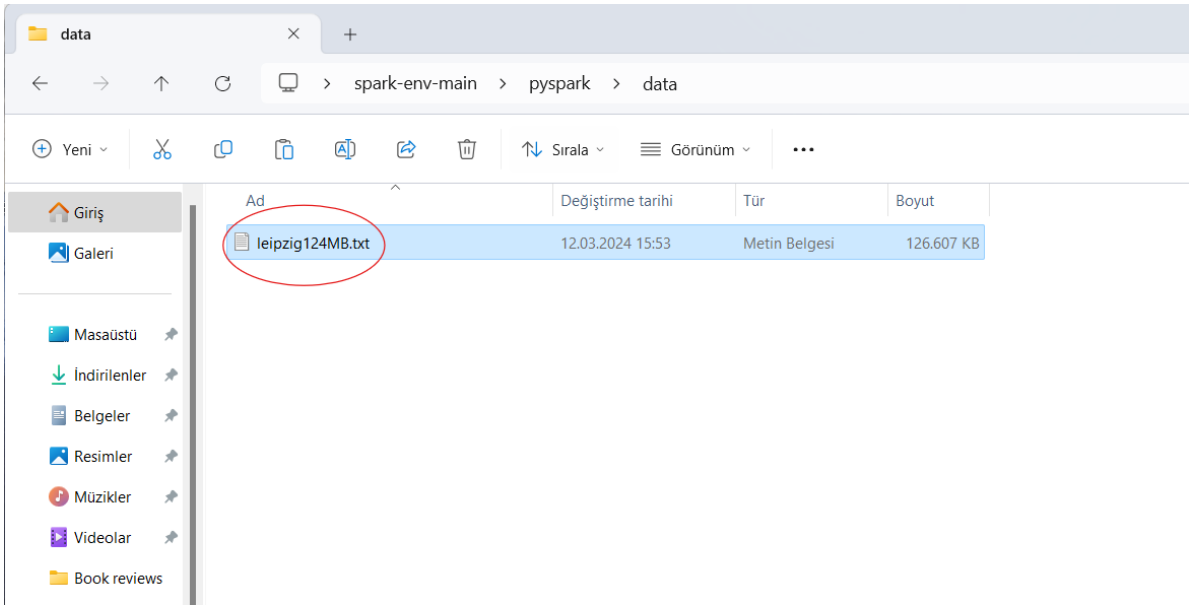
To change the versions of dependencies listed in requirements.txt, you simply need to update the version numbers next to each package name.

First way: Using .py file with Docker

1- Start Docker.
2- Create a data folder within the pyspark folder.

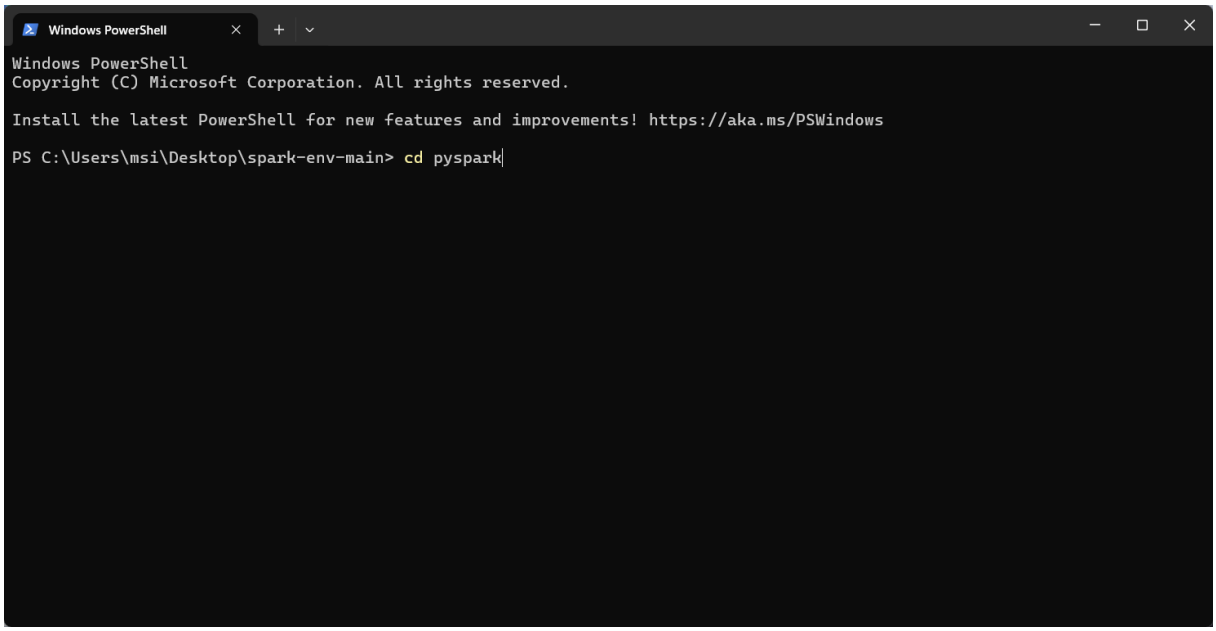


3- Upload the txt you want to use into the pyspark folder.



4- Right click on the folder named spark-env-main. Then press open in the terminal.

5- Write “cd pyspark” at terminal. Then press the enter.



6- Write " docker build -t pyspark-app.” at terminal. Then press enter. Wait to end of build.

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\msi\Desktop\spark-env-main> cd pyspark
PS C:\Users\msi\Desktop\spark-env-main\pyspark> docker build -t pyspark-app .
```

7- Write “docker run pyspark-app” at terminal.

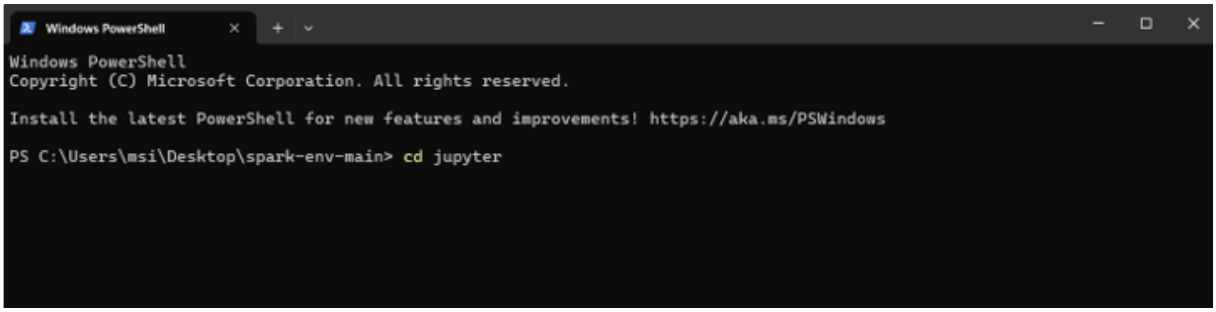
```
Windows PowerShell
 => [1/6] FROM docker.io/library/python:3.12-slim@sha256:afc139a0a640942491ec481ad8dda10f2c5b753f5c969393b1248015  8.2s
 => => resolve docker.io/library/python:3.12-slim@sha256:afc139a0a640942491ec481ad8dda10f2c5b753f5c969393b1248015  0.0s
 => => sha256:afc139a0a640942491ec481ad8dda10f2c5b753f5c969393b12480155fe15a63 1.65kB / 1.65kB                    0.0s
 => => sha256:fd3817f3a855f6c2ada16ac9468e5ee93e361005bd226fd5a5ee1a504e038c84 1.37kB / 1.37kB                    0.0s
 => => sha256:cf001c2f8af7214144935ae5b37c9e626ccf789117c10c1f691766d4658f1b1e 6.69kB / 6.69kB                    0.0s
 => => sha256:09f376ebb190216b0459f470e71bec7b5dfa611d66bf008492b40dcc5f1d8eae 29.15MB / 29.15MB                  5.3s
 => => sha256:276709cbedc1f168290ee408fca2af2aacfeb4f922ddca125e9e8047f9841479 3.51MB / 3.51MB                    0.8s
 => => sha256:2e133733af76c2a11aee79b7cfc733cc8065bc538d74b80ad65846d336a0904e 12.00MB / 12.00MB                  2.7s
 => => sha256:ded8879d9a790c3944bc22c2bed1606a9a2d4d293f2a32399fb183d577c0190a 243B / 243B                        1.1s
 => => sha256:3cf9507408dcb24084c3a20ccd068986cc37d1b13629d357f402581af5177013 3.05MB / 3.05MB                    2.2s
 => => extracting sha256:09f376ebb190216b0459f470e71bec7b5dfa611d66bf008492b40dcc5f1d8eae                         1.4s
 => => extracting sha256:276709cbedc1f168290ee408fca2af2aacfeb4f922ddca125e9e8047f9841479                         0.2s
 => => extracting sha256:2e133733af76c2a11aee79b7cfc733cc8065bc538d74b80ad65846d336a0904e                         0.5s
 => => extracting sha256:ded8879d9a790c3944bc22c2bed1606a9a2d4d293f2a32399fb183d577c0190a                         0.0s
 => => extracting sha256:3cf9507408dcb24084c3a20ccd068986cc37d1b13629d357f402581af5177013                         0.3s
 => [internal] load build context                                                                                  0.0s
 => => transferring context: 1.04kB                                                                                0.0s
 => [2/6] WORKDIR /app                                                                                             0.2s
 => [3/6] RUN apt-get update && apt-get install -y openjdk-17-jre-headless procps                                 21.6s
 => [4/6] COPY requirements.txt /app/                                                                              0.0s
 => [5/6] RUN pip install --no-cache-dir -r requirements.txt                                                      45.9s
 => [6/6] COPY . /app                                                                                              0.0s
 => exporting to image                                                                                             1.4s
 => => exporting layers                                                                                            1.4s
 => => writing image sha256:63daa24af2d486a5e1a56c68636caeb9524068bec28394d6547b9aab30ee2153                       0.0s
 => => naming to docker.io/library/pyspark-app                                                                     0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\msi\Desktop\spark-env-main\pyspark> docker run pyspark-app
```

8- Finish!! You can see the result at terminal.

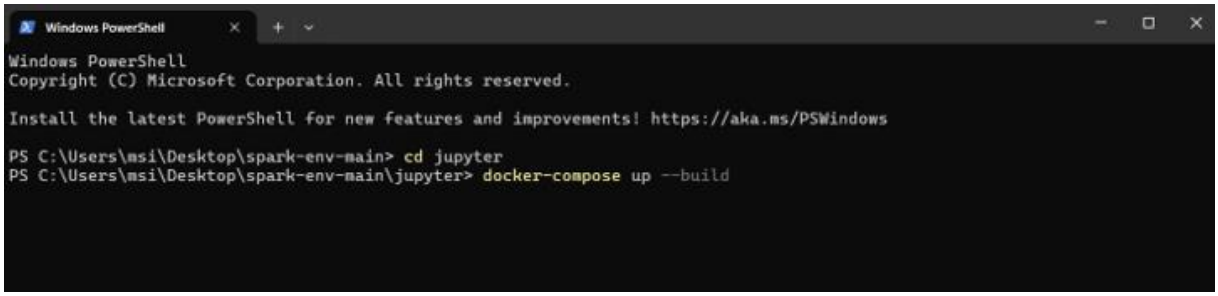
```
Windows PowerShell
PS C:\Users\msi\Desktop\spark-env-main\pyspark> docker run pyspark-app
/app/app.py:7: SyntaxWarning: invalid escape sequence '\s'
  return lower(regexp_replace(c, '[^a-zA-Z\s]', '')).alias('cleaned')
/app/app.py:13: SyntaxWarning: invalid escape sequence '\s'
  words = df.select(explode(split(col("cleaned"), "\s+")).alias("word"))
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/05/19 13:13:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
+----+-------+
|word|  count|
+----+-------+
| the|1370263|
|  of| 595835|
|  to| 566298|
|   a| 507768|
|  in| 473567|
| and| 448469|
|said| 215897|
| for| 214759|
|that| 202017|
|  is| 175755|
+----+-------+
```

Second way: Using Jupyter with Docker

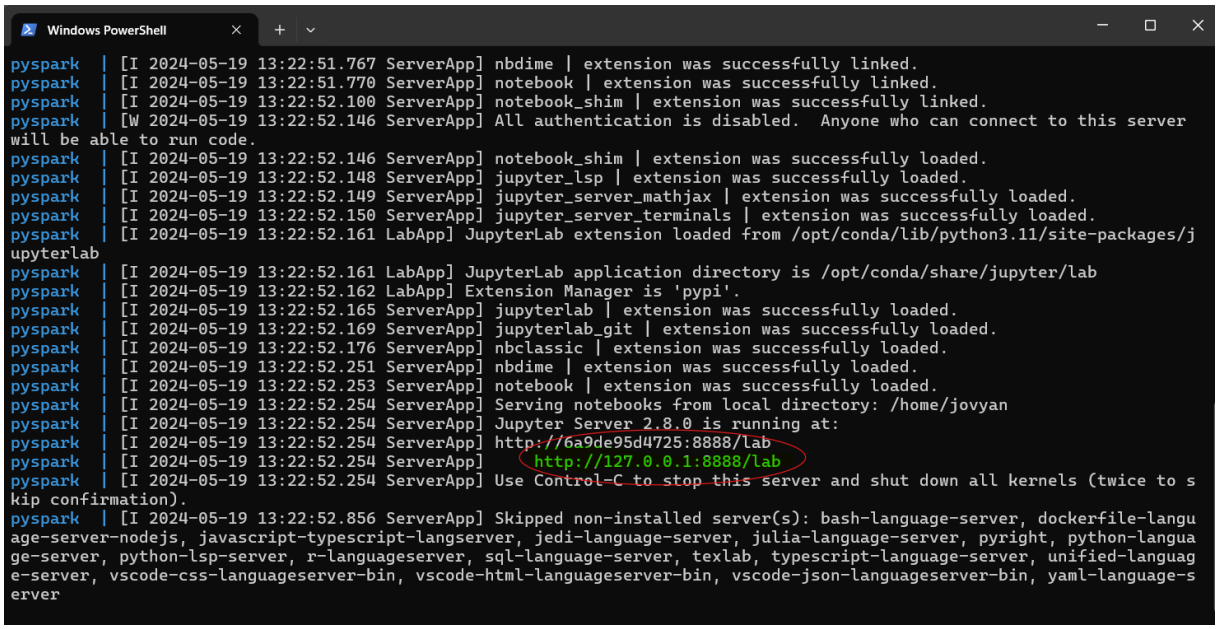
1- Start Docker.
2- Right click on the folder named spark-env-main. Then press open in the terminal.
3- Write “cd pyspark” at terminal. Then press the enter.



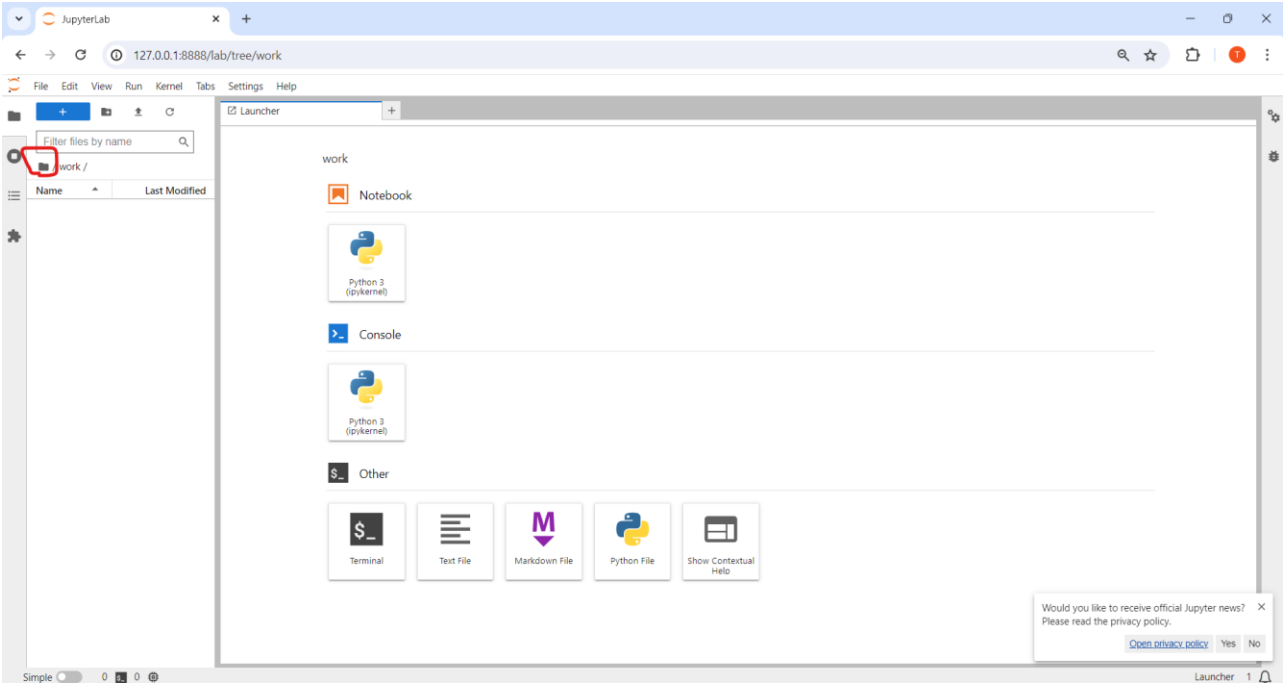
4- Write “docker-compose up –build” at terminal. Then press the enter. Wait to end of build.



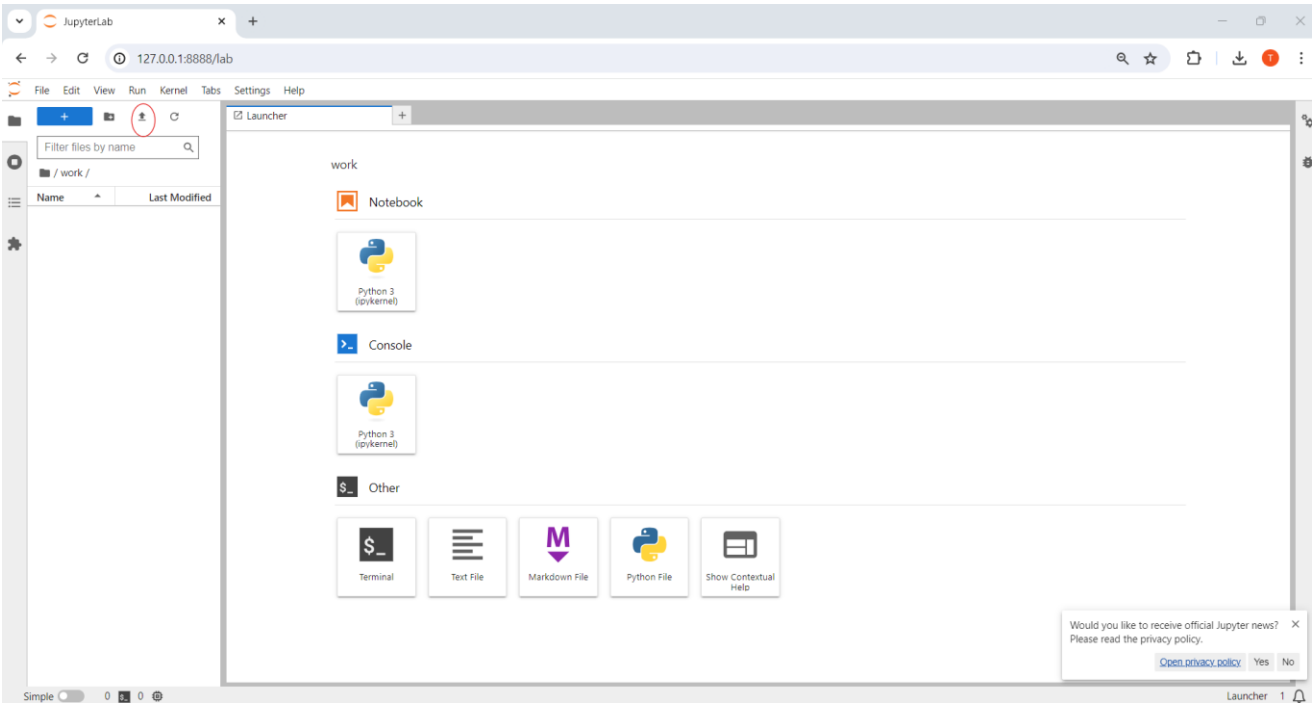
5- Go to this website http://127.0.0.1:8888/lab



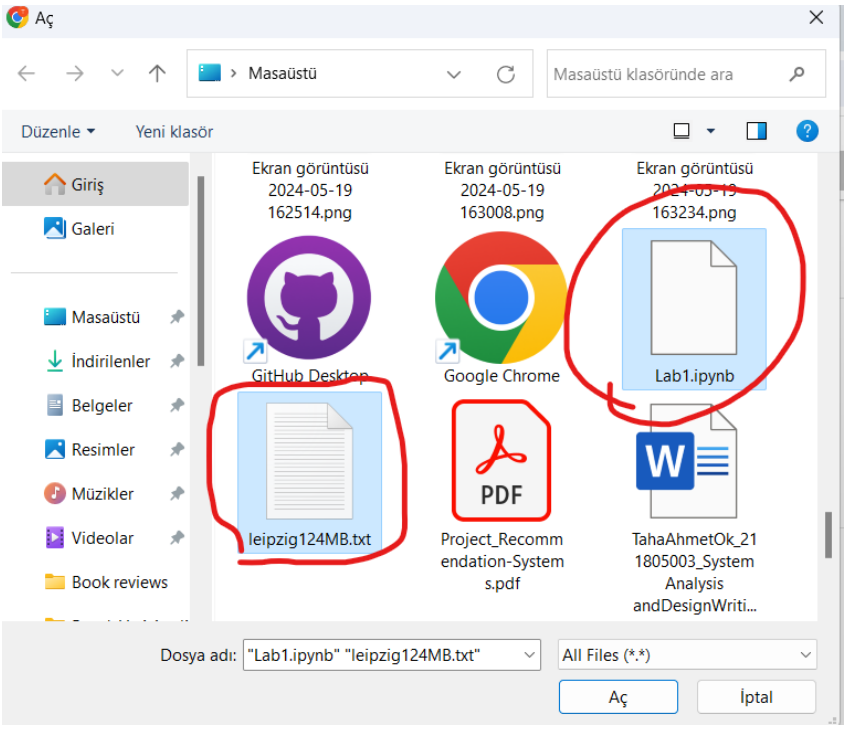
6- Click this icon.



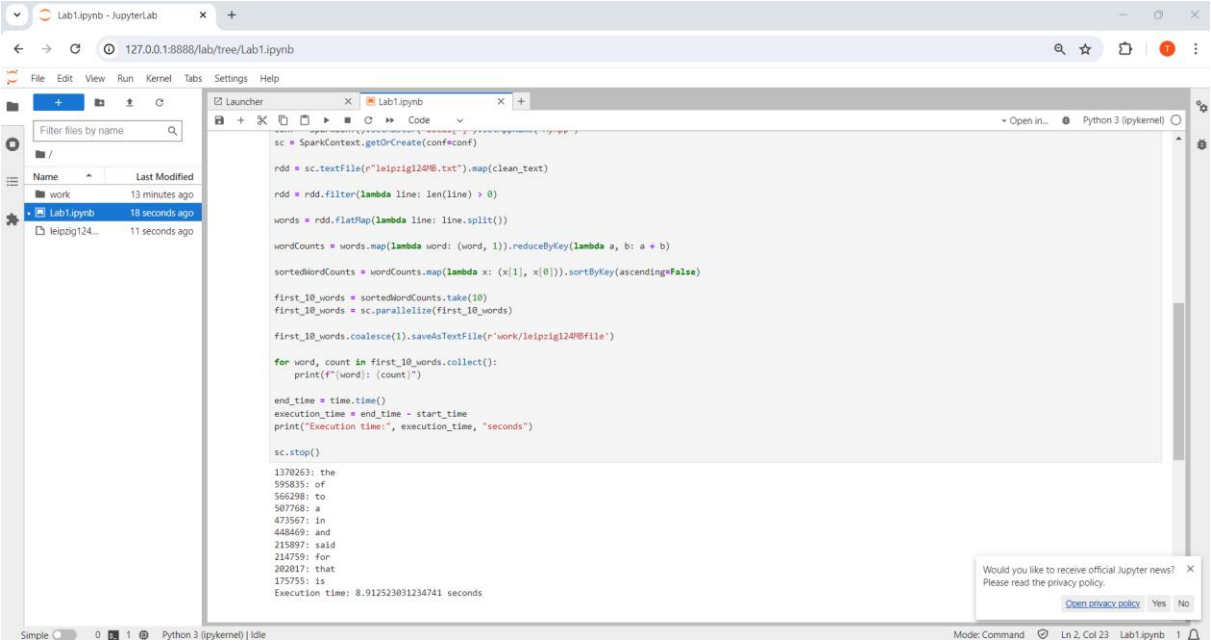
7- Click this icon.



8- Select your Ipynb and txt file.



9- Finish!! You can see the result.



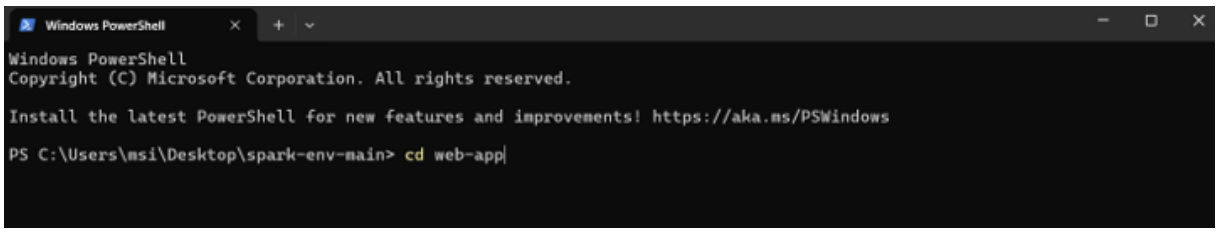
Third way: Using web-app with Docker

We use Aws S3 to store the txt file and the web page shows the number of words in the loaded text.

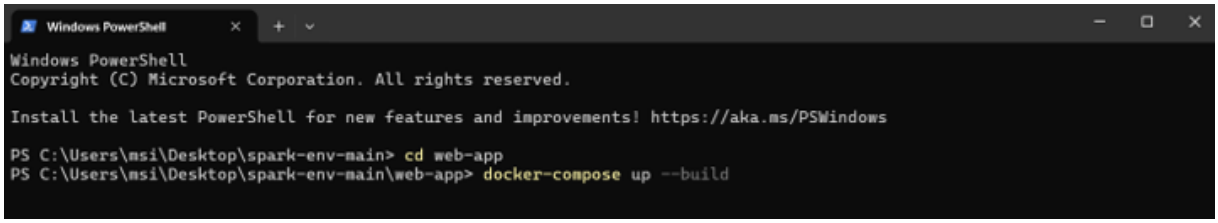
1-Start Docker.

2- Right click on the folder named spark-env-main. Then press open in the terminal.

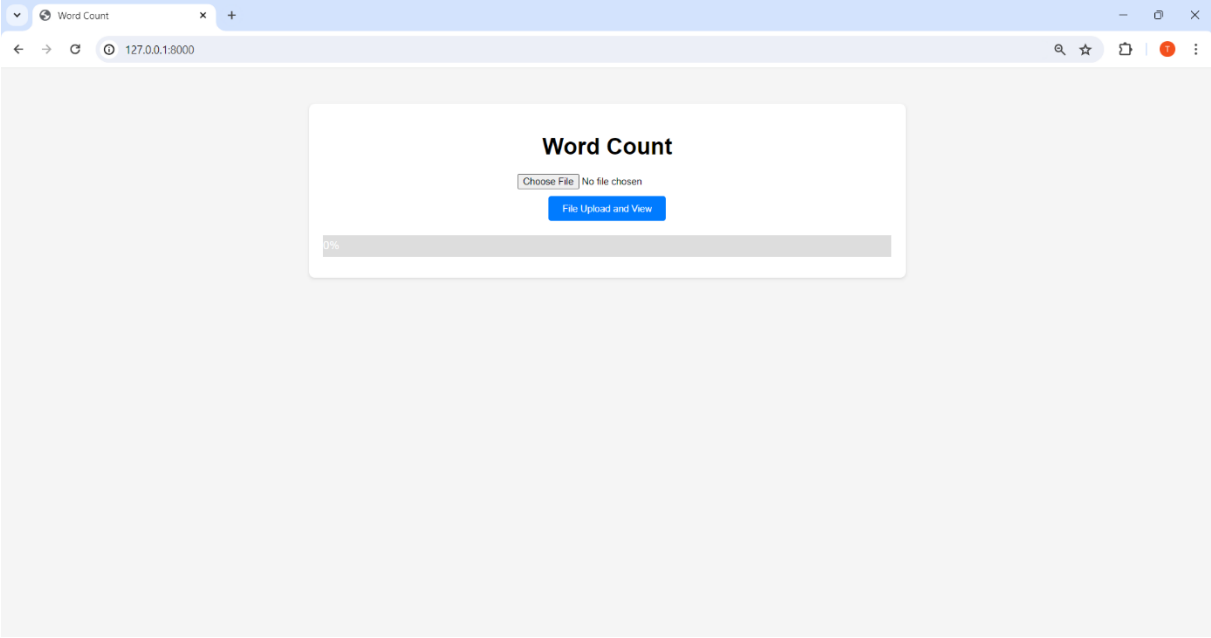
3- Write “cd web-app” at terminal. Then press the enter.



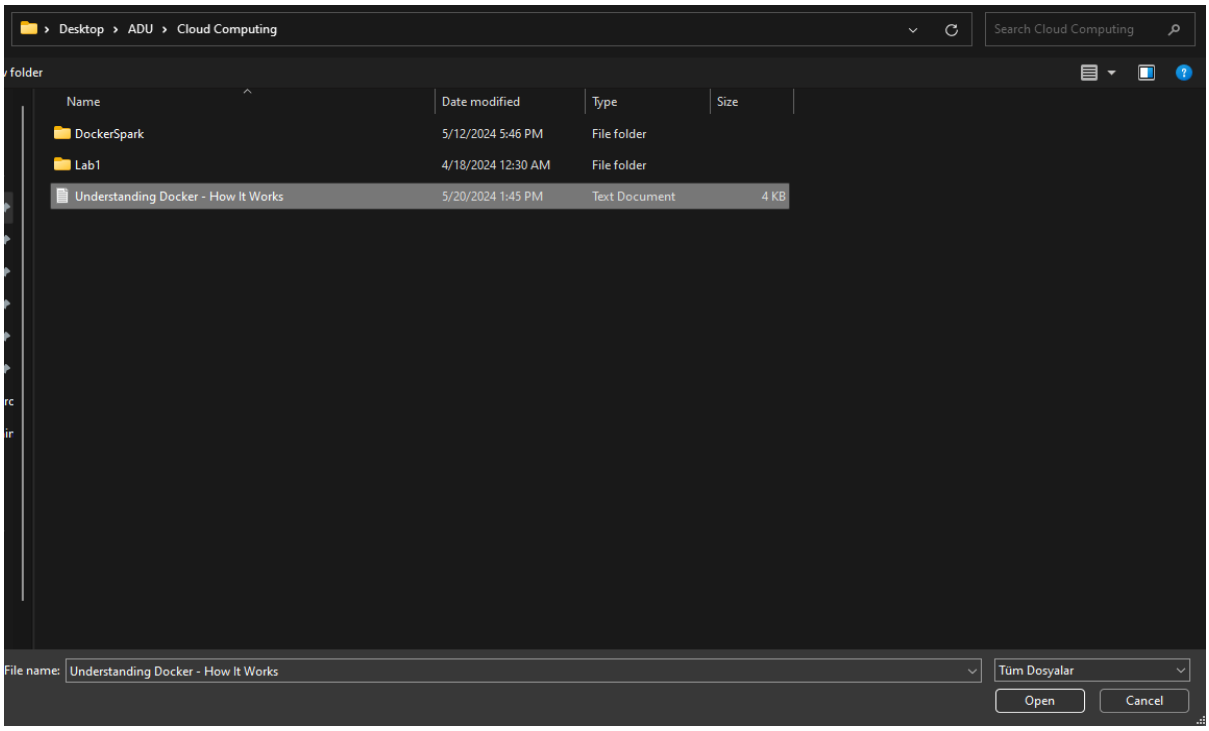
4-Write “docker-compose up –build” at terminal. Then press the enter. Wait to end of build.



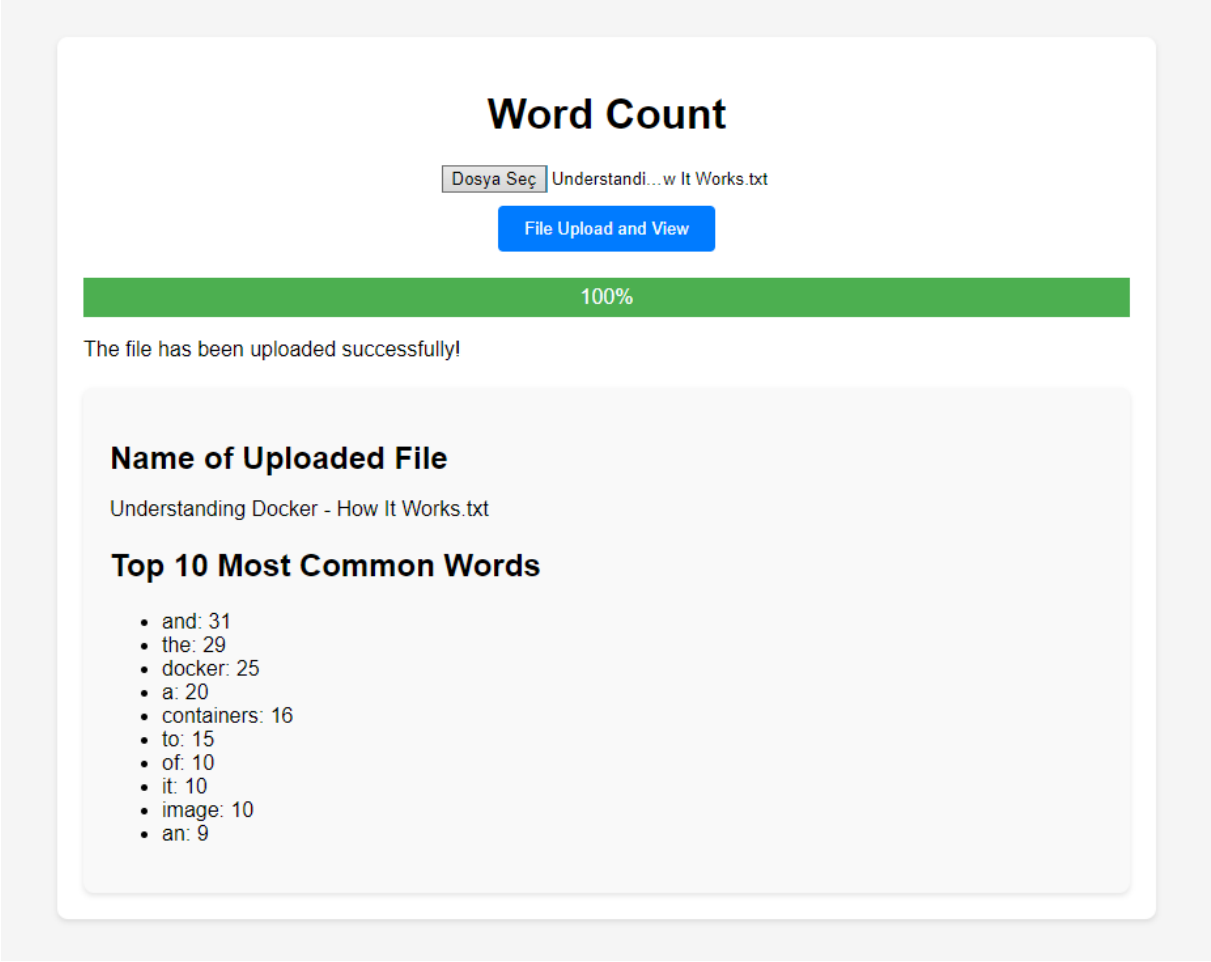
5- Go to this website http://127.0.0.1:8000/



6- Press the “Choose File” button. Then Choose txt for counting word.



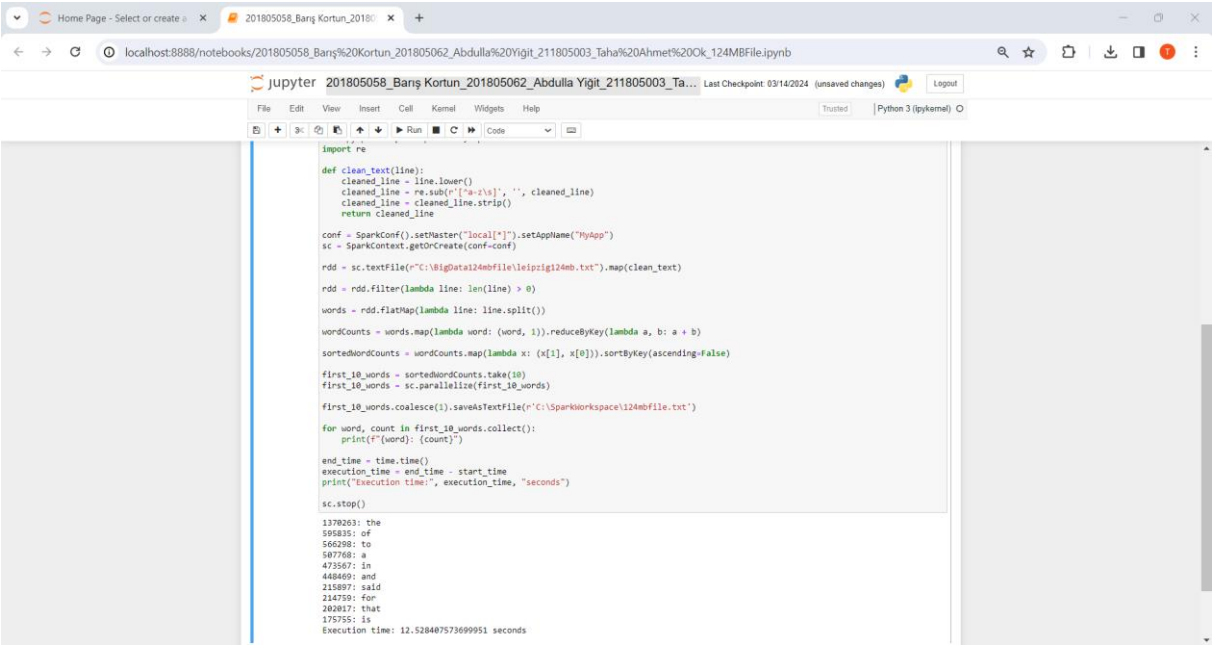
7- After selecting the file, click the “File Upload and View” button and here is the result!



Comparison Using Docker and Without Using Docker

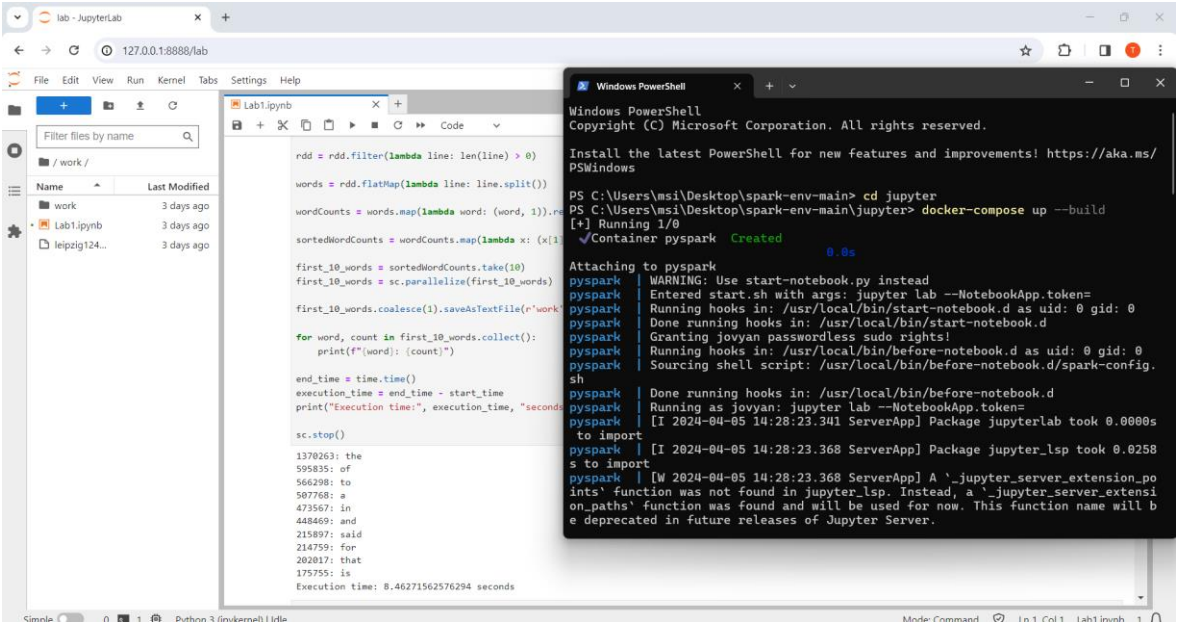
The report includes screenshots and explanations for the lab of the Big Data course conducted using Docker and without Docker. Our goal is to show that the work done using Docker is correct. We made 3 way of making word count application with Docker. Therefore, there are 3 way of making word count application; working in terminal with .py file, working in jupyter notebook and showing in internet tab.

Jupyter Notebook Without Using Docker for 124 Mb



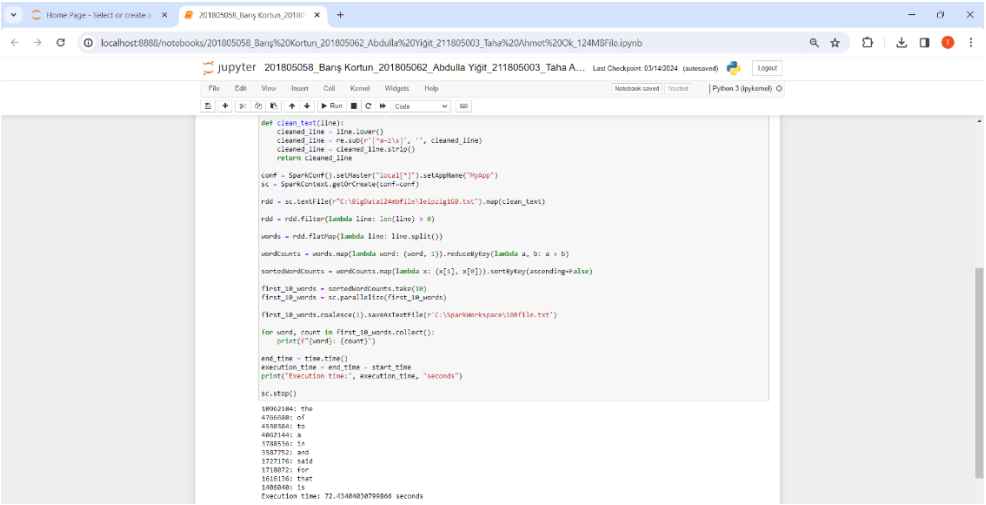

```
1370263: the
595835: of
566298: to
507768: a
473567: in
448469: and
215897: said
214759: for
202017: that
175755: is
Execution time: 12.528407573699951 seconds
```

Jupyter Notebook With Using Docker for 124 Mb



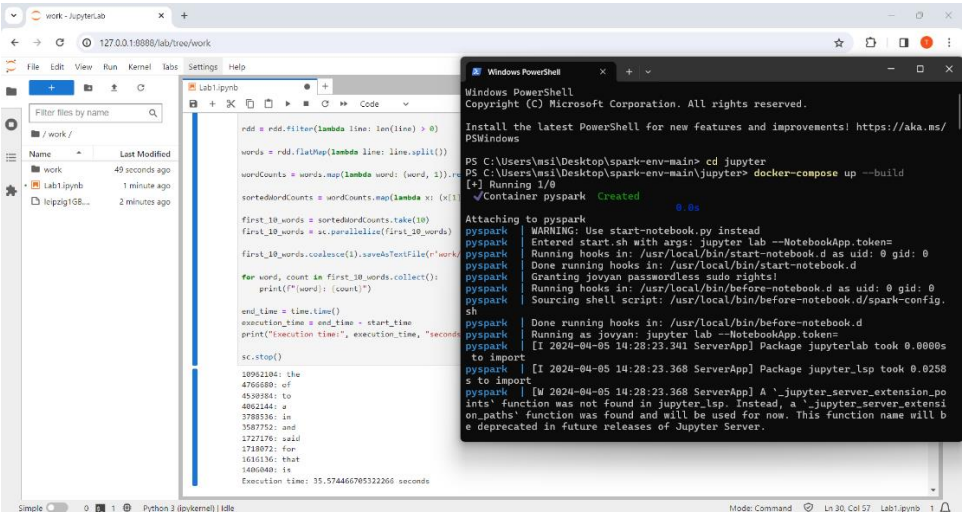

```
1370263: the
595835: of
566298: to
507768: a
473567: in
448469: and
215897: said
214759: for
202017: that
175755: is
Execution time: 8.46271562576294 seconds
```

Jupyter Notebook Without Using Docker for 1 GB



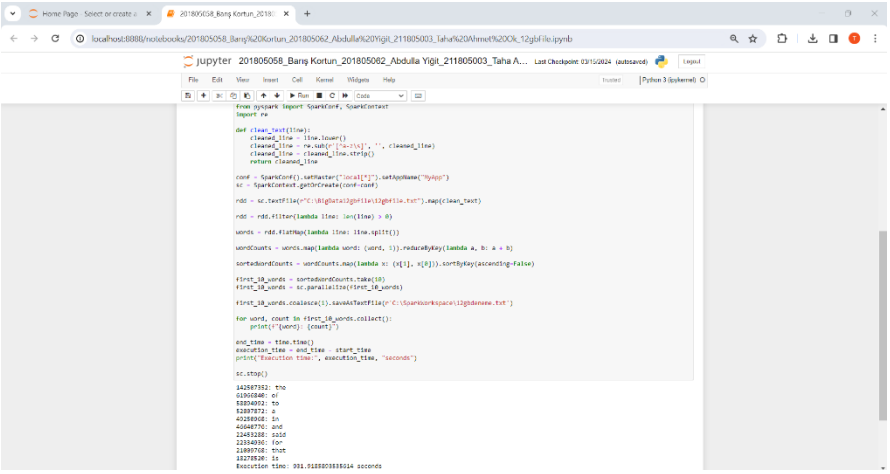
```
10962104: the
4766680: of
4530384: to
4062144: a
3788536: in
3587752: and
1727176: said
1718072: for
1616136: that
1406040: is
Execution time: 72.43404030799866 seconds
```

Jupyter Notebook With Using Docker for 1 GB



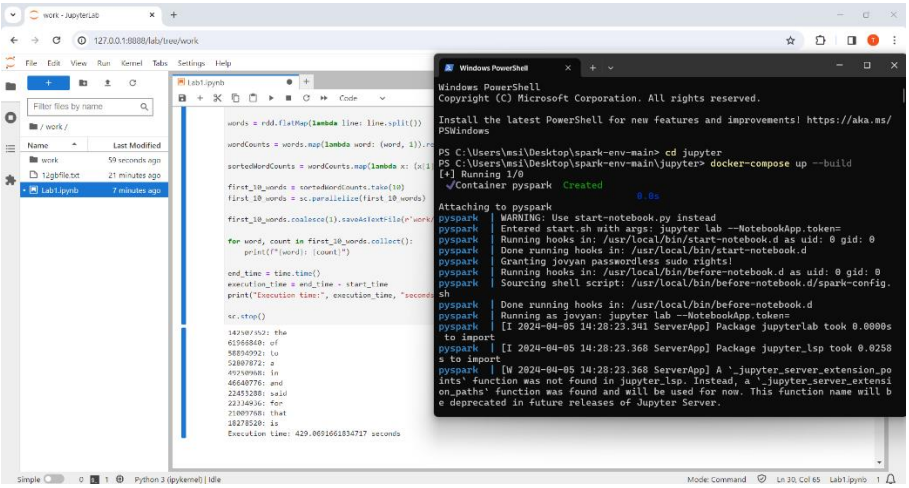
```
10962104: the
4766680: of
4530384: to
4062144: a
3788536: in
3587752: and
1727176: said
1718072: for
1616136: that
1406040: is
Execution time: 35.574466705322266 seconds
```

Jupyter Notebook Without Using Docker for 12 GB



```
142507352: the
61966840: of
58894992: to
52807872: a
49250968: in
46640776: and
22453288: said
22334936: for
21009768: that
18278520: is
Execution time: 931.9185893535614 seconds
```

Jupyter Notebook With Using Docker for 12 GB



```
142507352: the
61966840: of
58894992: to
52807872: a
49250968: in
46640776: and
22453288: said
22334936: for
21009768: that
18278520: is
Execution time: 429.0691661834717 seconds
```

Showing Results at Terminal With Using Docker For 124 Mb

```
Windows PowerShell

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build
PS C:\Users\msi\Desktop\spark-env-main\pyspark> docker run pyspark-app
/app/app.py:7: SyntaxWarning: invalid escape sequence '\s'
  return lower(regexp_replace(c, '[^a-zA-Z\s]', '')).alias('cleaned')
/app/app.py:13: SyntaxWarning: invalid escape sequence '\s'
  words = df.select(explode(split(col("cleaned"), "\s+")).alias("word"))
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/04/05 15:21:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
+----+-------+
|word|  count|
+----+-------+
| the|1370263|
|  of| 595835|
|  to| 566298|
|   a| 507768|
|  in| 473567|
| and| 448469|
|said| 215897|
| for| 214759|
|that| 202017|
|  is| 175755|
+----+-------+
```

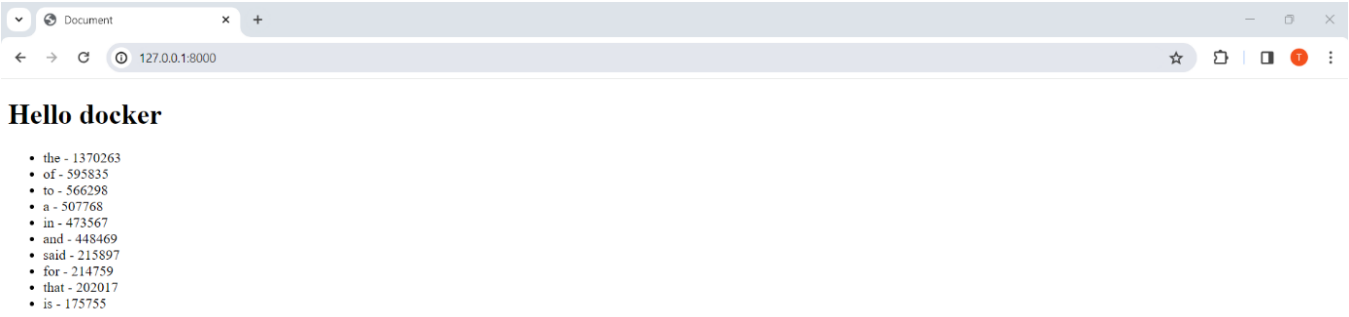
Showing Results at Terminal With Using Docker For 1 GB

```
Windows PowerShell

 => => writing image sha256:e40dce66a1222d1a707a37938a8520856fcf8d829bb95afcfe2f4d53f87c2271                    0.0s
 => => naming to docker.io/library/pyspark-app                                                                  0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\msi\Desktop\spark-env-main\pyspark> docker run pyspark-app
/app/app.py:7: SyntaxWarning: invalid escape sequence '\s'
  return lower(regexp_replace(c, '[^a-zA-Z\s]', '')).alias('cleaned')
/app/app.py:13: SyntaxWarning: invalid escape sequence '\s'
  words = df.select(explode(split(col("cleaned"), "\s+")).alias("word"))
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/04/05 15:32:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
+----+--------+
|word|   count|
+----+--------+
| the|10962104|
|  of| 4766680|
|  to| 4530384|
|   a| 4062144|
|  in| 3788536|
| and| 3587752|
|said| 1727176|
| for| 1718072|
|that| 1616136|
|  is| 1406040|
+----+--------+

PS C:\Users\msi\Desktop\spark-env-main\pyspark>
```

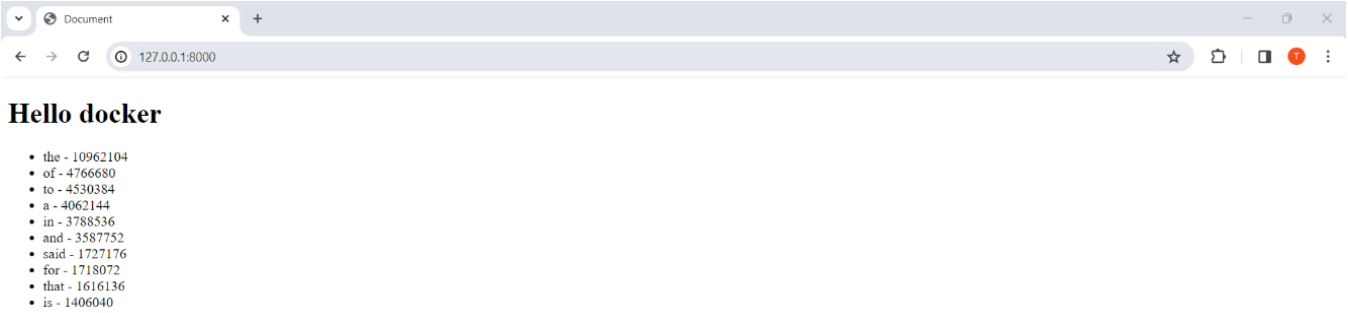
Showing Results at Terminal With Using Docker For 12 GB

```
Windows PowerShell

 => => writing image sha256:2120f6786ef2e04686c19aa4eefadeafe64b2baaee7ac0fd29d754c412695a76                    0.0s
 => => naming to docker.io/library/pyspark-app                                                                  0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\msi\Desktop\spark-env-main> docker run pyspark-app
/app/./app.py:7: SyntaxWarning: invalid escape sequence '\s'
  return lower(regexp_replace(c, '[^a-zA-Z\s]', '')).alias('cleaned')
/app/./app.py:13: SyntaxWarning: invalid escape sequence '\s'
  words = df.select(explode(split(col("cleaned"), "\s+")).alias("word"))
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/29 18:49:04 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
+----+---------+
|word|    count|
+----+---------+
| the|142507352|
|  of| 61966840|
|  to| 58894992|
|   a| 52807872|
|  in| 49250968|
| and| 46640776|
|said| 22453288|
| for| 22334936|
|that| 21009768|
|  is| 18278520|
+----+---------+
```

Showing Results at Web-App With Using Docker For 124 Mb



Showing Results at Web-App With Using Docker For 1 Gb



Showing Results at Web-App With Using Docker For 12 Gb