**BIG DATA**

**JOBSHEET 7 – SPARK DOCKER**



**RIO BAGAS HERMAWAN / 2241720193**

**TEKNIK INFORMATIKA – 3D**

**TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

# Interaksi dengan Spark di Lingkungan Windows Menggunakan Docker

Dalam praktikum ini kita akan menjalankan Apache Spark di Windows menggunakan Docker dan mencoba membuat job sederhana dengan berbagai macam alternatif cara.

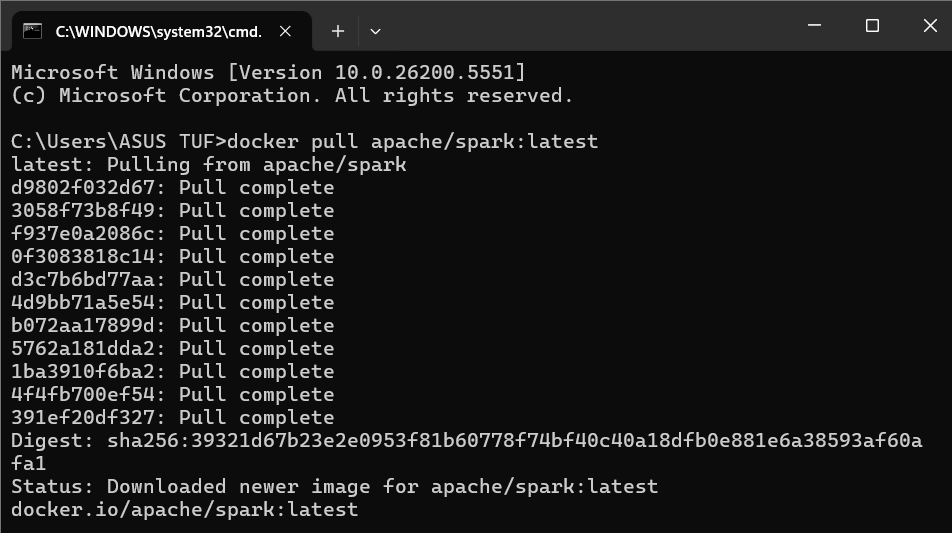
Prasyarat

1. Windows 10/11 (64-bit) dengan versi Pro, Enterprise, atau Education
2. Docker Desktop untuk Windows diinstal dan berjalan
3. WSL 2 (Windows Subsystem for Linux versi 2) diaktifkan

Langkah-langkah

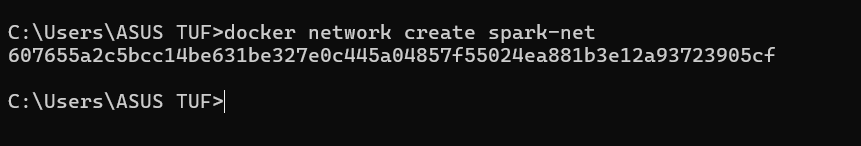
1. Pull Image Spark Resmi

docker pull apache/spark:latest

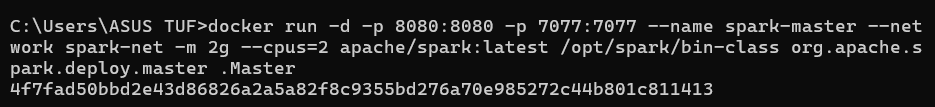


1. Menjalankan Spark Master

Sebelumnya buat docker network sebagai berikut

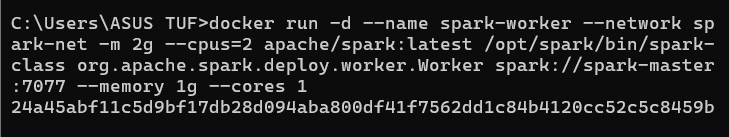


Kemudian jalankan spark-master dalam network tersebut



Kita alokasikan resource untuk memastikan tidak kekurangan resource dalam menjalankan job.

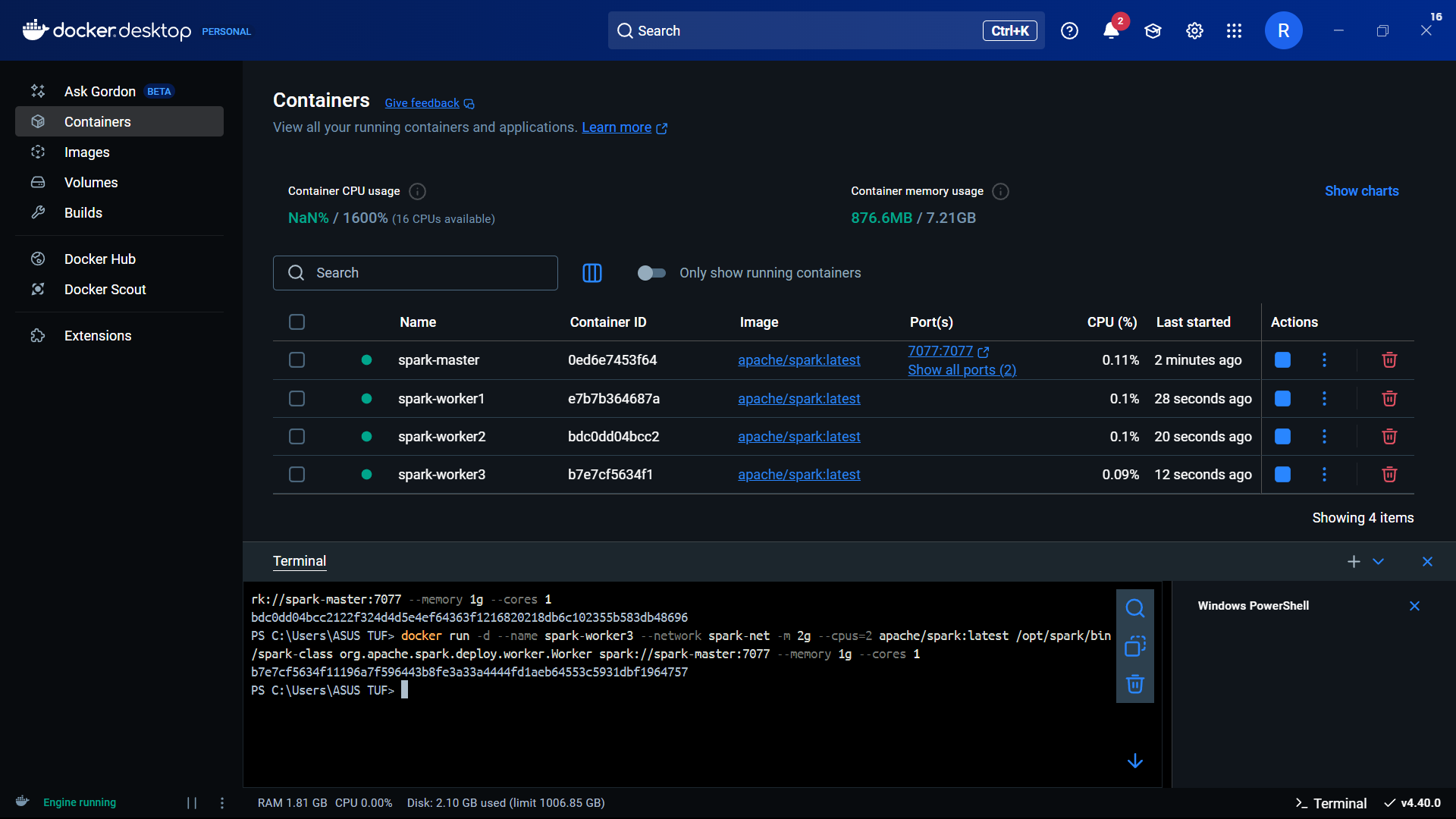
1. Menjalankan Spark Worker



Kita perlu alokasikan resource misalnya 2G memori dan 2 core CPU

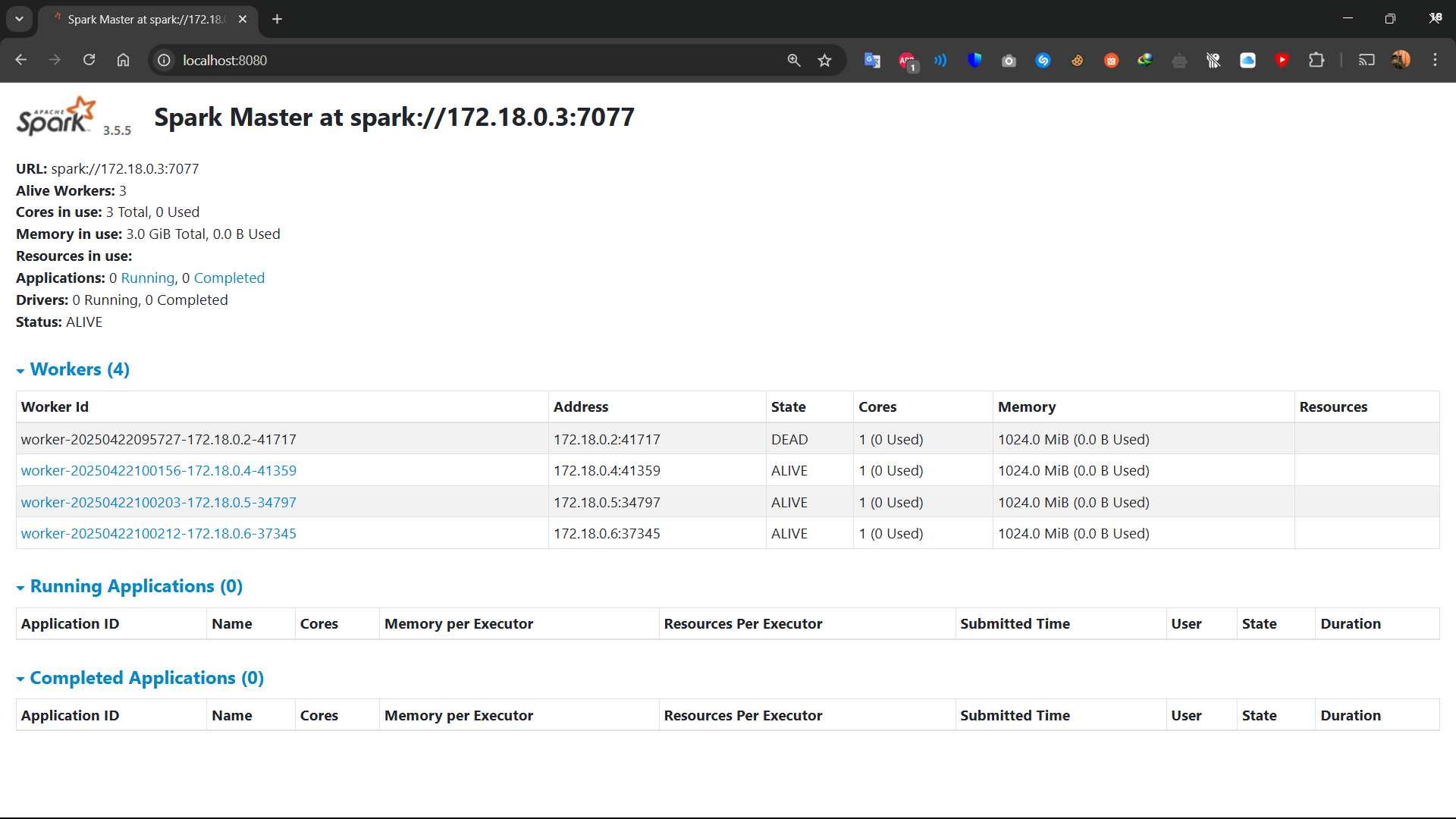
jalankan perintah di atas beberapa kali dengan nama yang berbeda untuk membuat beberapa worker. Misalnya spark-worker1, spark-worker2, dan seterusnya

Contoh menggunakan 3 worker.

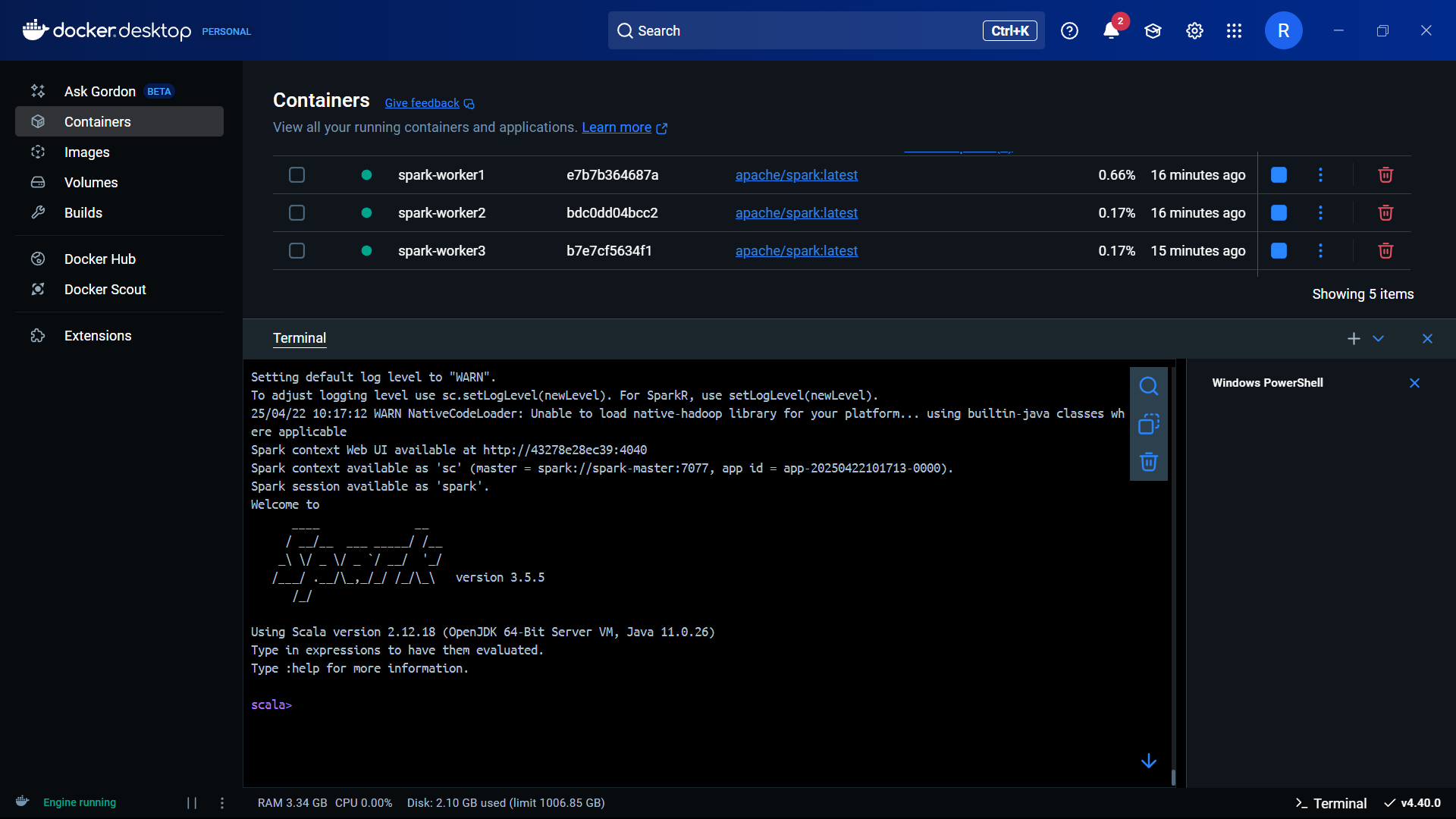


1. Mengakses Spark Web UI

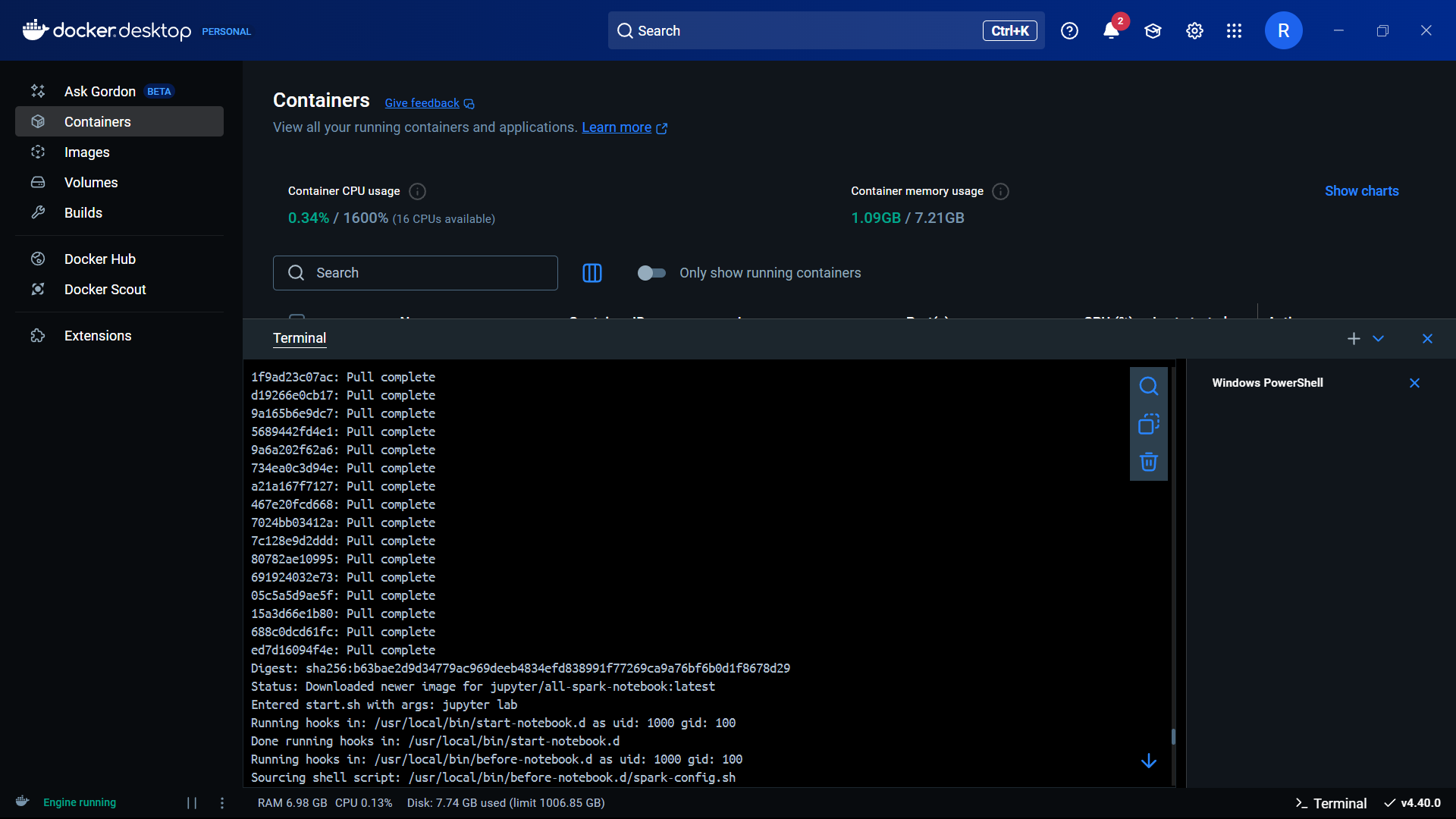
<http://localhost:8080>



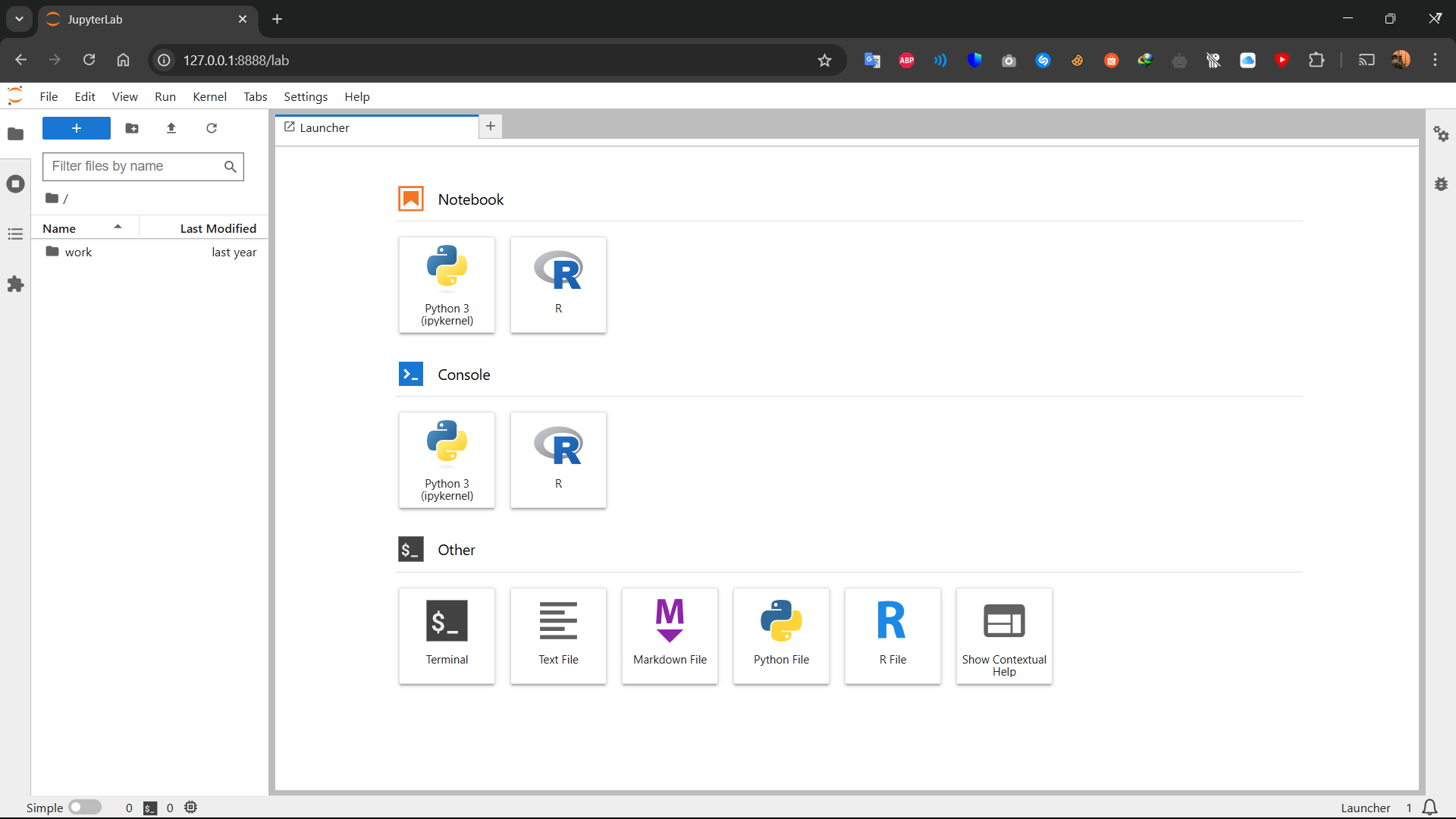
1. Menjalankan Spark Shell



1. Menggunakan Jupyter Notebook dengan Spark



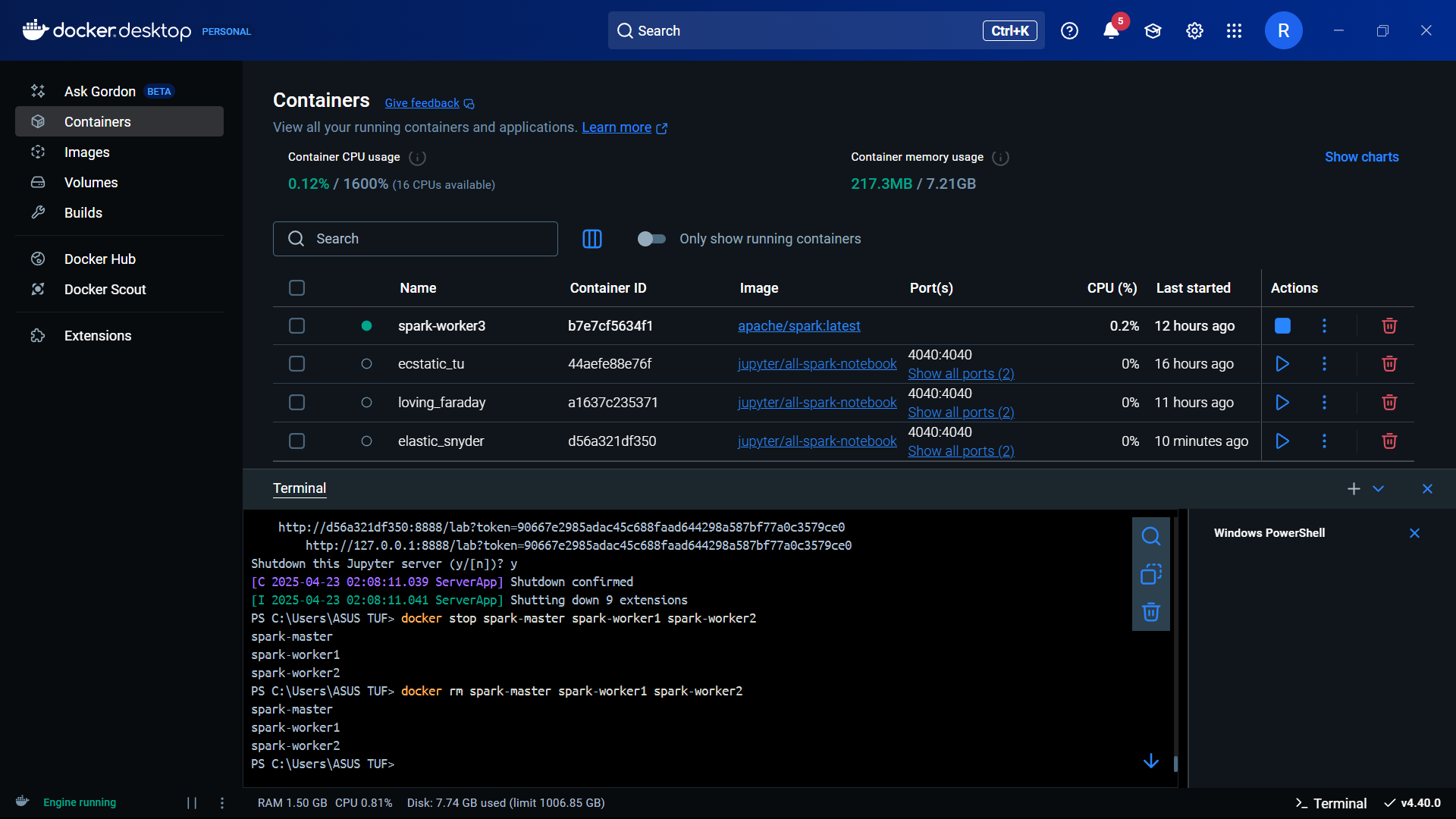
Setelah itu, akses Jupyter Notebook di: http://localhost:8888



Untuk menghentikan container:

docker stop spark-master spark-worker

docker rm spark-master spark-worker



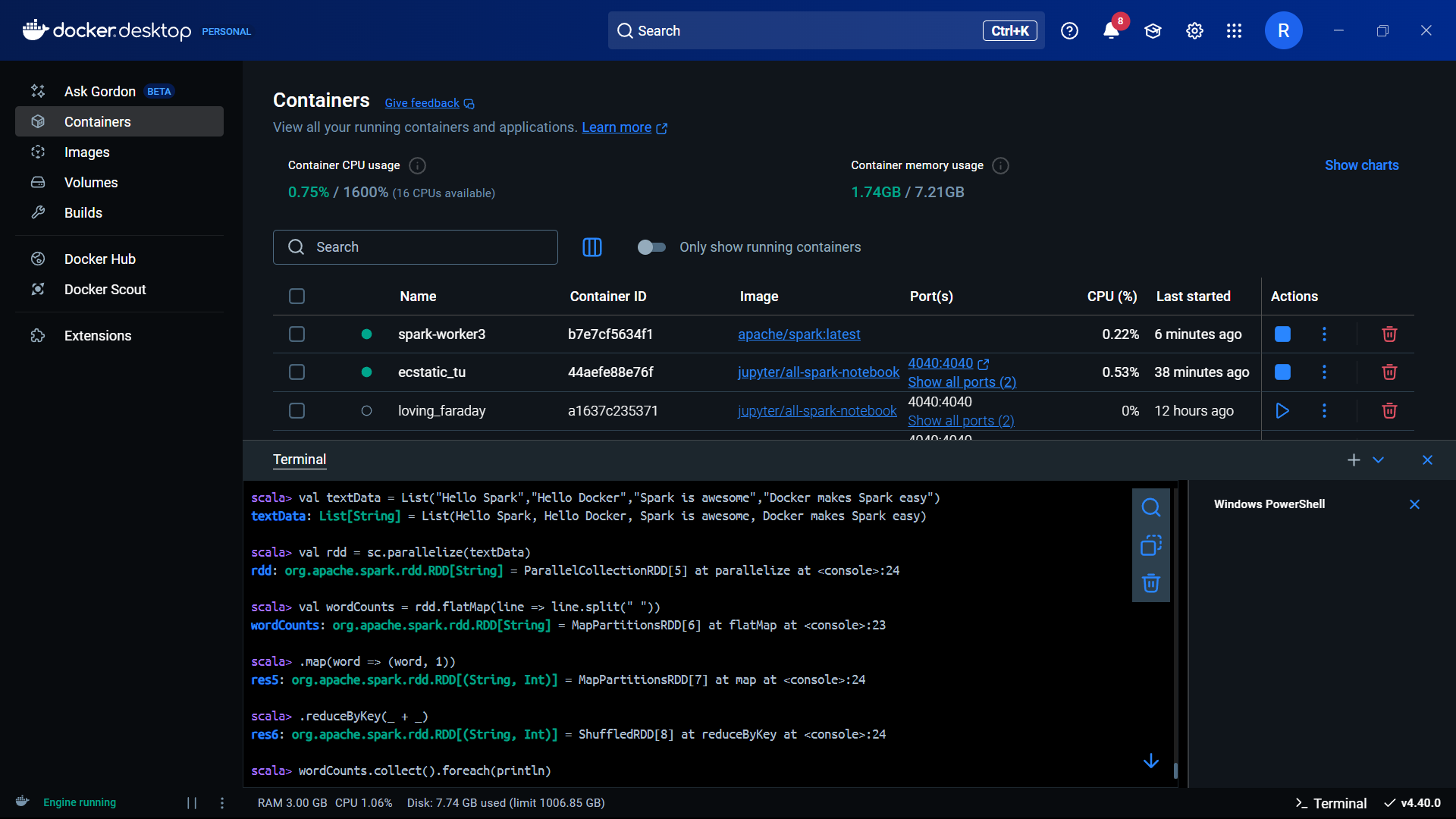
Contoh Program Word Count dengan Spark di Docker

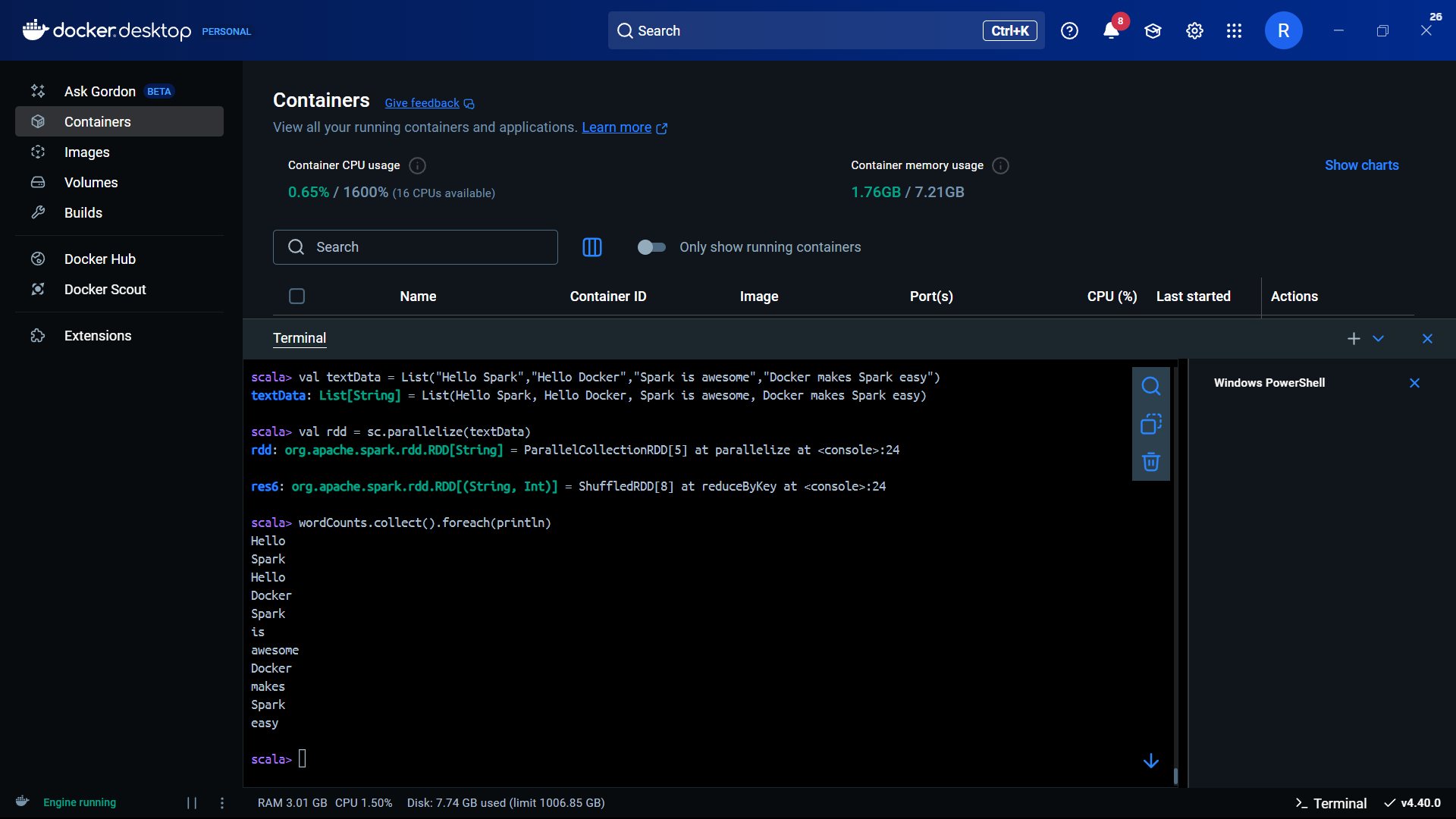
Berikut adalah contoh program Word Count (menghitung kemunculan kata) menggunakan Apache Spark

yang bisa dijalankan di lingkungan Docker:

Cara 1 : Menggunakan Spark Shell

1. Jalankan Spark Shell di Docker seperti contoh di atas
2. Ketikkan kode berikut di Spark Shell:



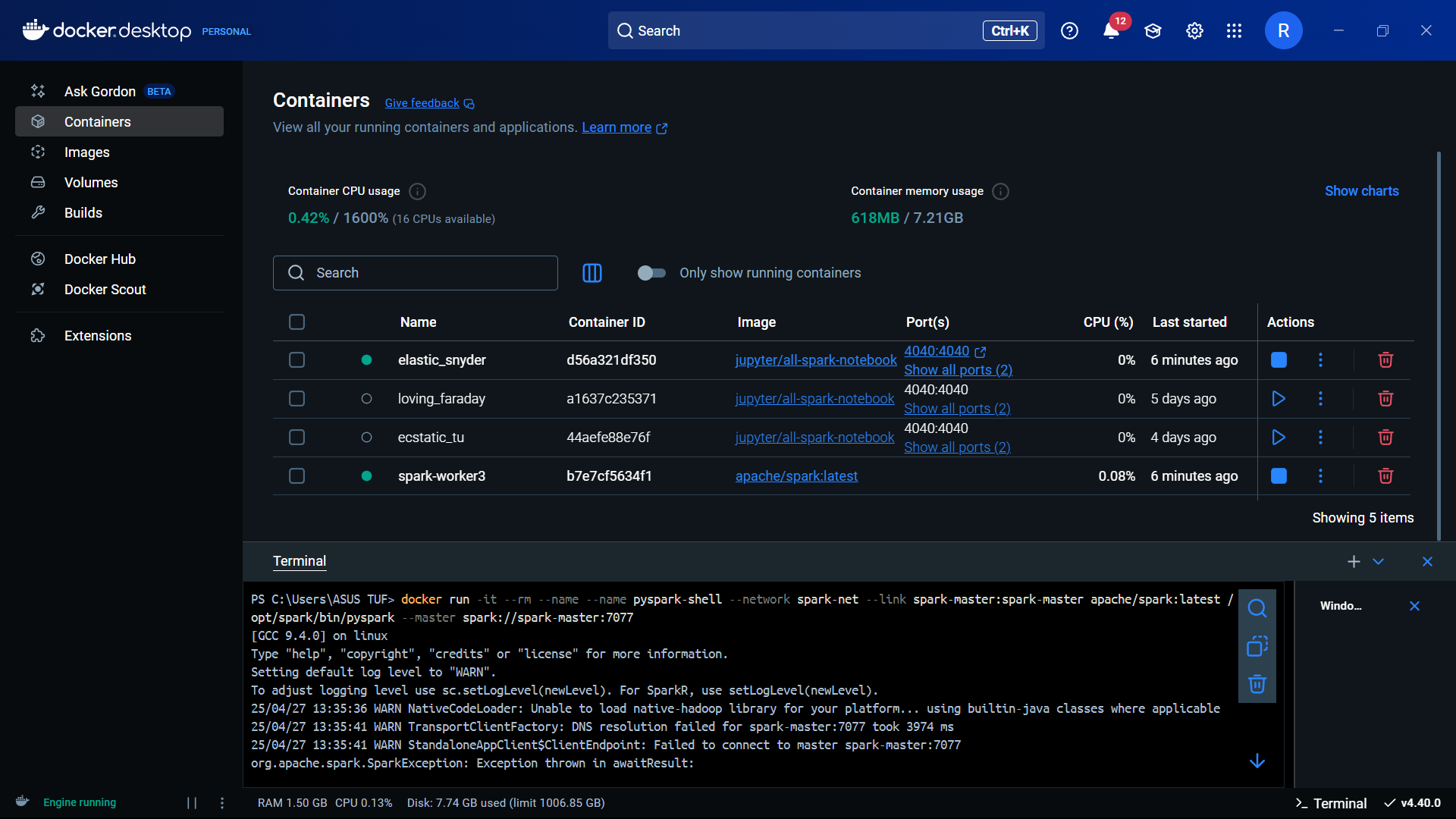


Untuk keluar dari spark-shell gunakan:

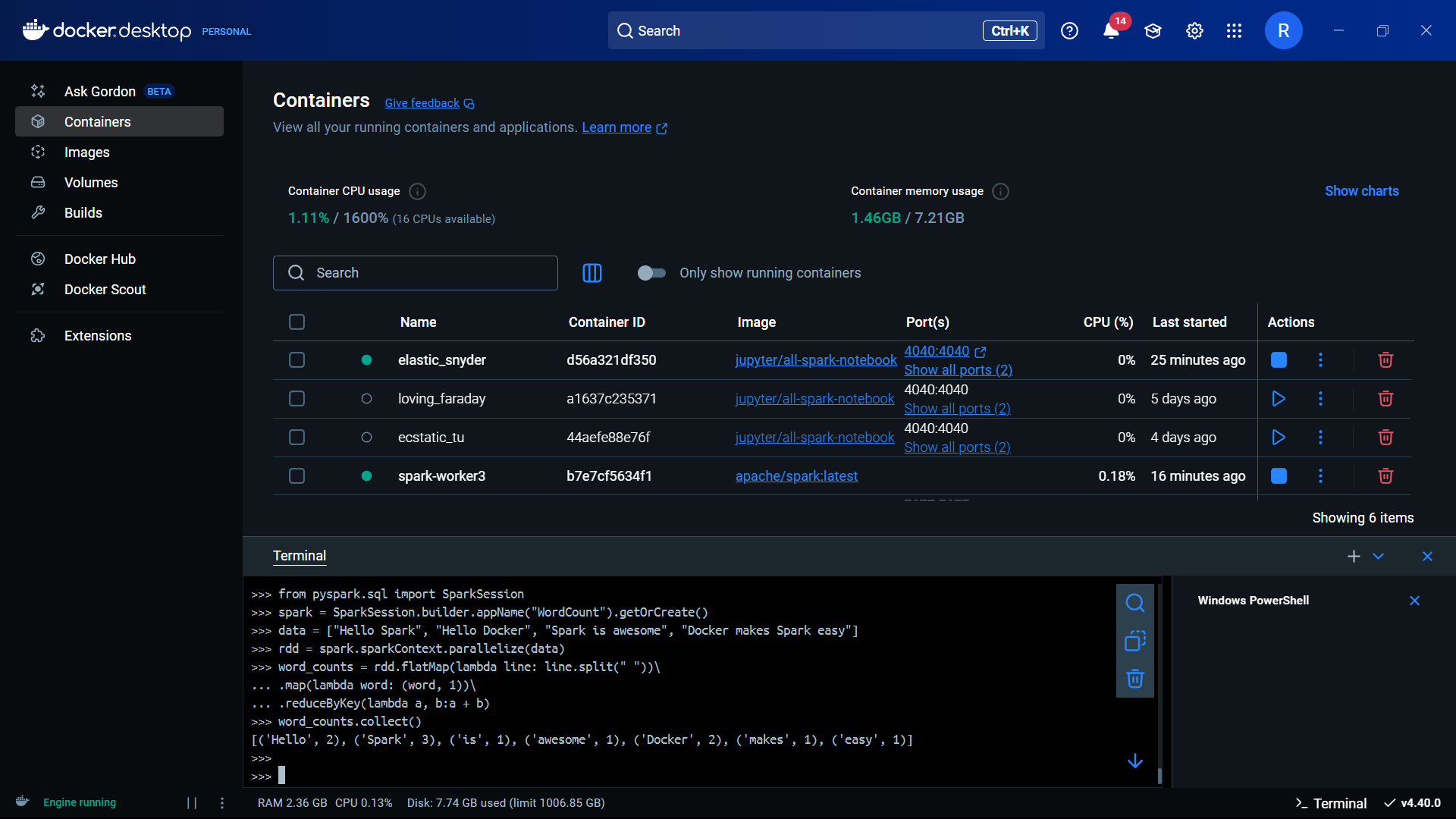
System.exit(0)

Cara 2: Menggunakan PySpark (Python)

1. Jalankan PySpark Shell di Docker: Dalam command juga terdapat definisi network juga.



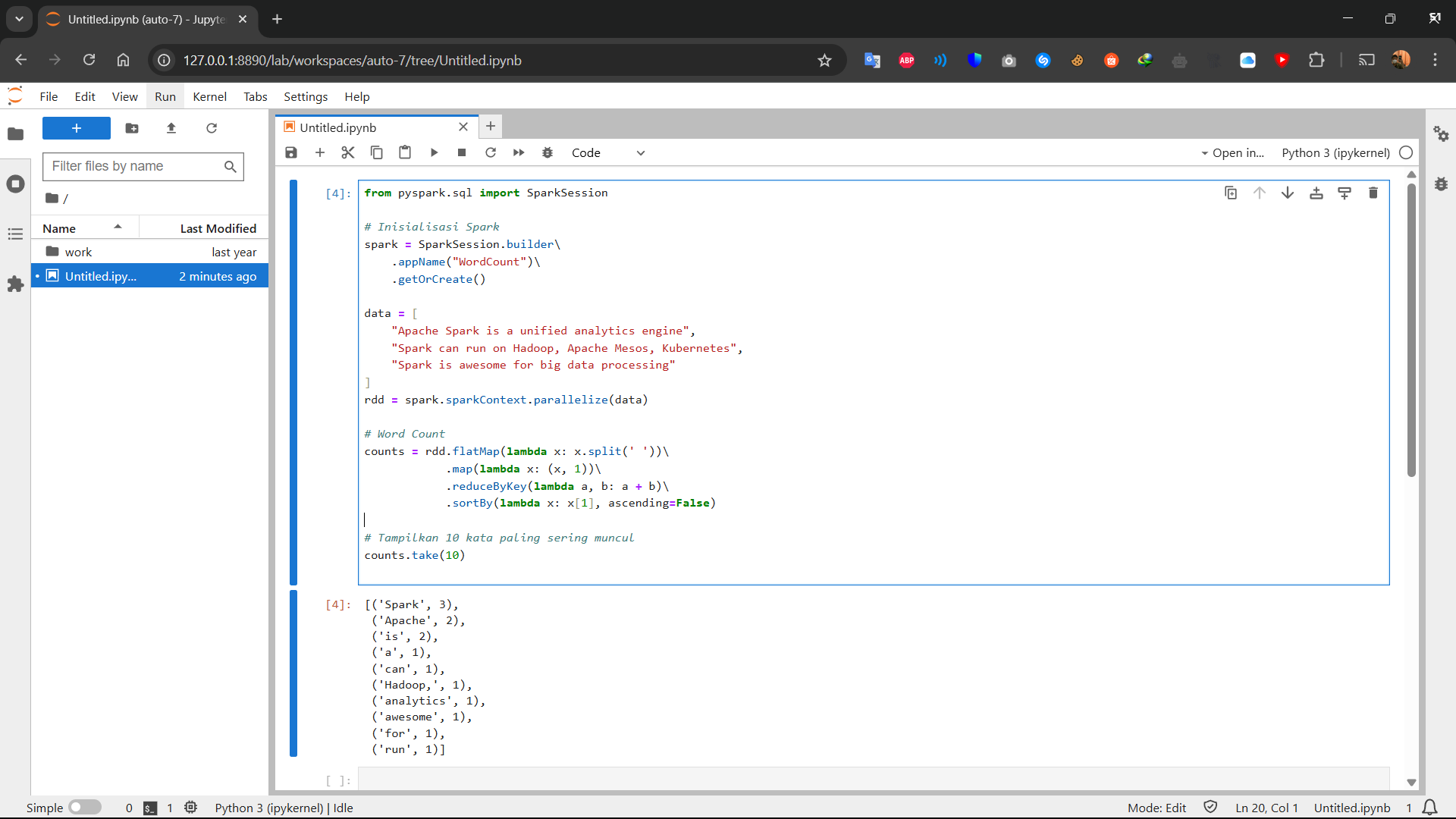
1. Ketikkan kode Python berikut:



Untuk keluar dari pyspark-shel menggunakan: exit()

Cara 3: Menggunakan Jupyter Notebook

Jika Anda menggunakan Jupyter Notebook (seperti di container jupyter/all-spark-notebook):



Menjalankan Program sebagai Script

1. Buat file wordcount.py dengan isi berikut:
2. Jalankan script, jangan lupa juga mendifinisikan network spark-net

Perhatikan dalam command tersebut mendifinisikan akses data ke local.

Program-program di atas akan menghasilkan output seperti:

Hello: 2

Spark: 3

Docker: 2

is: 1

awesome: 1 ...