ASSIGNMENT 03

Sabaragamuwa University of Sri Lanka

Faculty of Computing

Software Engineering

SE6103 – Parallel and Distributed Systems

Name:	Disara Mapalagama
Reg. No:	19APSE4302
Academic Period:	3rd Year 2nd Semester
Degree Program:	BSc (Hons) in Software Engineering
Due Date:	06.01.2024

Assignment 03

Question 1

1. Explain the differences between Docker Containers and Virtual machines.

Docker Containers and Virtual Machines (VMs) differ primarily in their architecture. Containers share the host OS kernel and run as isolated processes in user space, making them lightweight and fast to start, with minimal resource overhead. In contrast, VMs run a full operating system with its own kernel on top of a hypervisor, making them more resource-intensive, slower to start, and requiring more memory and storage.

In terms of portability and use cases, containers are more portable as they encapsulate everything needed to run an application, making them ideal for microservices and rapid deployment. They are well-suited for environments where fast scaling is required. VMs, however, provide stronger isolation and can run different OS types, making them suitable for running legacy applications or when higher security and complete isolation are needed.

2. What does the-d flag in the docker run command do? Why is it important when running services like Nginx?

The `-d` flag in the `docker run` command runs the container in **detached mode**, meaning it runs in the background. This is important for services like Nginx, as it allows the container to run continuously without blocking the terminal, enabling other tasks. You can still view logs using the `docker logs` command.

3. Explain the difference between the following commands: docker run-d nginx:latest and docker run nginx:latest. Which one would you use for long-running services?

The difference between the two commands is:

- docker run -d nginx:latest: Runs the Nginx container in detached mode (background). It starts the container and immediately returns control to the terminal, allowing you to run other commands.
- docker run nginx:latest: Runs the Nginx container in the foreground, meaning it will occupy the terminal and display logs and output, blocking further interaction with the terminal until stopped.

For long-running services like Nginx, you would use docker run -d nginx:latest to ensure the service runs in the background without blocking your terminal.

4. When running the command docker run-d-p 8080:80 nginx, what does-p 8080:80 accomplish? Why is port mapping necessary?

In the command docker run -d -p 8080:80 nginx, the -p 8080:80 option is port mapping. It maps port 80 inside the Docker container (where Nginx is running) to port 8080 on the host machine.

- Container Port (80): This is the port where Nginx listens inside the container.
- Host Port (8080): This is the port on the host machine that allows access to the Nginx service running in the container.

Why Port Mapping is Necessary:

Port mapping is necessary because, by default, containers have their own isolated network and aren't directly accessible from the host machine. The -p flag enables communication between the container and the host machine by forwarding traffic from a host port (8080) to a container port (80), allowing users to access the Nginx service by visiting http://localhost:8080 on the host.

5. What is Hadoop and what is it used for?

Hadoop is an open-source framework used for storing and processing large datasets in a distributed computing environment. It is designed to handle massive amounts of data across many computers by distributing the data and computation.

Key Components:

- **Hadoop Distributed File System (HDFS):** A distributed storage system that splits data into blocks and stores them across multiple machines.
- **MapReduce:** A programming model for processing and generating large datasets in parallel by breaking the job into smaller tasks.
- **YARN (Yet Another Resource Negotiator):** A resource management layer that manages and schedules resources across the cluster.

What is it Used For:

- Efficiently stores large volumes of data in a distributed way.
- Processes vast amounts of data quickly using parallel computing across a cluster of machines.
- Used for data mining, machine learning, and large-scale analytics tasks.

Hadoop is commonly used in industries like finance, healthcare, and social media to handle tasks involving large-scale data processing and analytics.

6. What are the advantages of using Apache Spark over Hadoop? Explain.

Apache Spark offers several advantages over Hadoop, including much faster processing due to in-memory computation, while Hadoop relies on slower disk-based operations. Spark supports real-time stream processing, whereas Hadoop is mainly focused on batch processing. Additionally, Spark provides higher-level APIs in languages like Python, Java, and Scala, making it easier to use, while Hadoop's MapReduce requires more complex coding. Spark also supports a unified approach to handle batch, streaming, and iterative workloads, and includes built-in libraries for machine learning and graph processing, unlike Hadoop, which needs separate tools for advanced analytics.

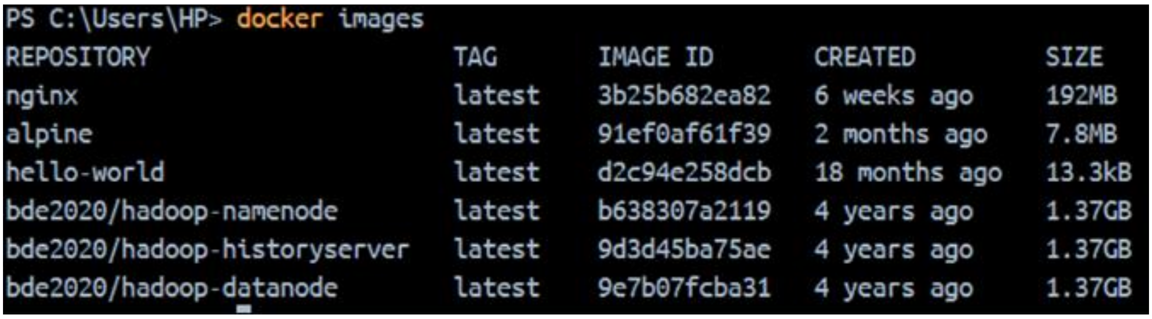
Question 2

1. Use the knowledge you gained during the Hadoop lab sessions to analyze the following example of a text and find the number of occurrences of each word in the text.

Hadoop is an open-source framework that stores and processes large amounts of data across a network of computers. Hadoop is a very interesting framework to learn about networking.

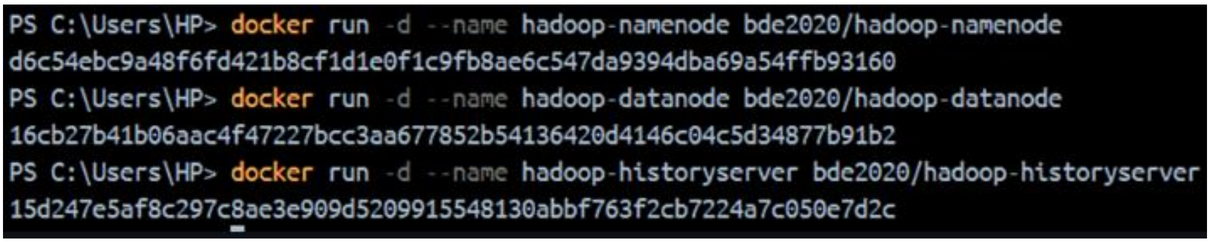
Answer:

Confirming Pre-requisites Command: Docker images Output:



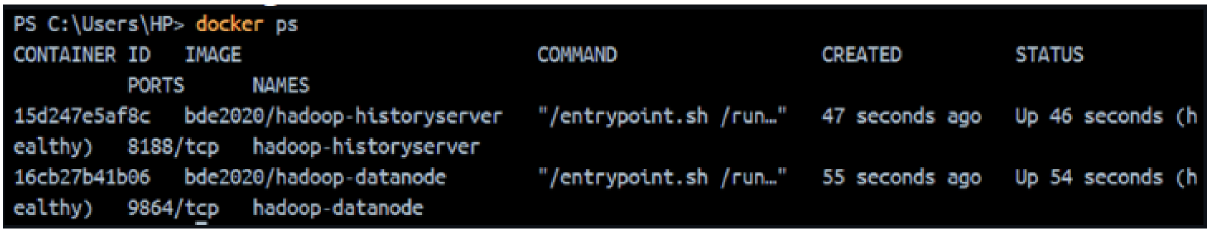
```
PS C:\Users\HP> docker images
REPOSITORY                      TAG       IMAGE ID       CREATED          SIZE
nginx                           latest    3b25b682ea82   6 weeks ago      192MB
alpine                          latest    91ef0af61f39   2 months ago     7.8MB
hello-world                     latest    d2c94e258dcb   18 months ago    13.3kB
bde2020/hadoop-namenode         latest    b638307a2119   4 years ago      1.37GB
bde2020/hadoop-historyserver    latest    9d3d45ba75ae   4 years ago      1.37GB
bde2020/hadoop-datanode         latest    9e7b07fcba31   4 years ago      1.37GB
```

Start containers:



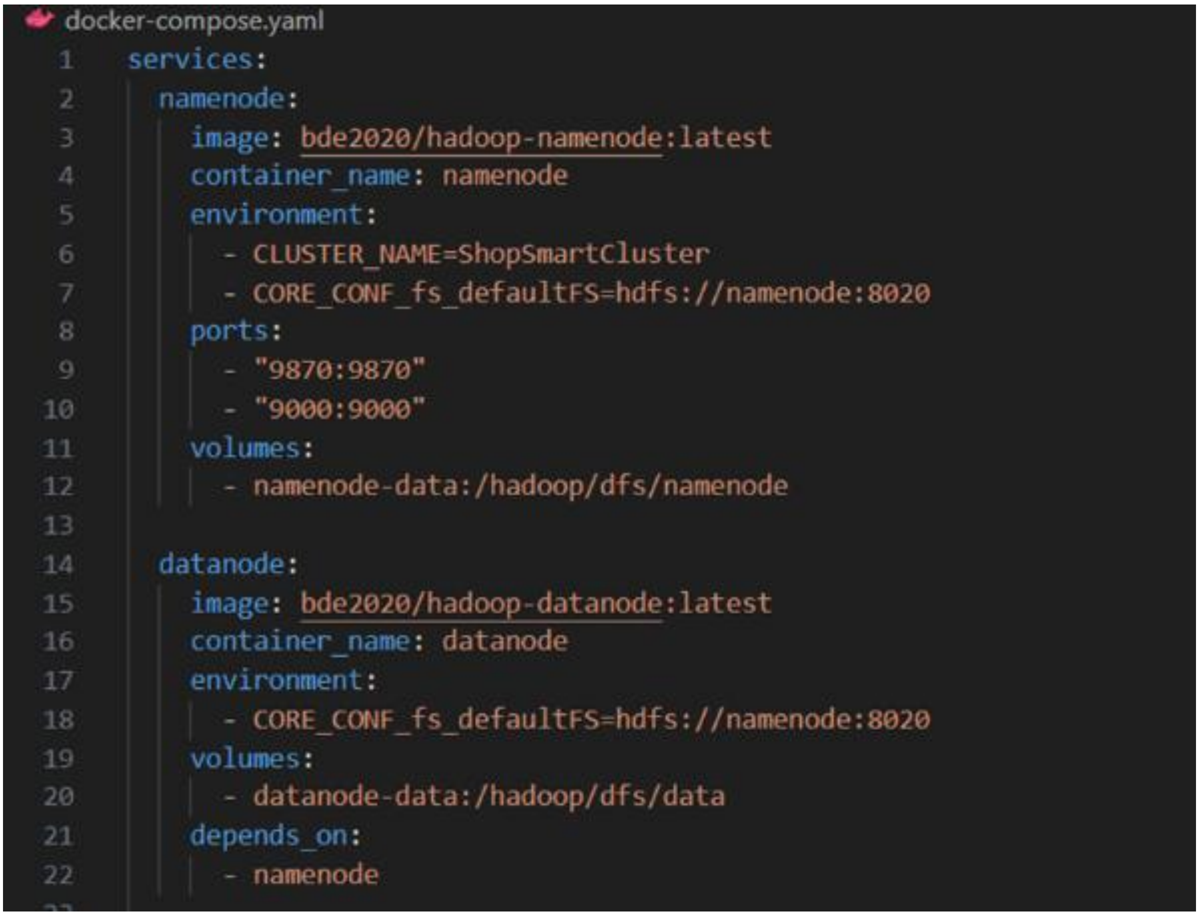
```
PS C:\Users\HP> docker run -d --name hadoop-namenode bde2020/hadoop-namenode
d6c54ebc9a48f6fd421b8cf1d1e0f1c9fb8ae6c547da9394dba69a54ffb93160
PS C:\Users\HP> docker run -d --name hadoop-datanode bde2020/hadoop-datanode
16cb27b41b06aac4f47227bcc3aa677852b54136420d4146c04c5d34877b91b2
PS C:\Users\HP> docker run -d --name hadoop-historyserver bde2020/hadoop-historyserver
15d247e5af8c297c8ae3e909d5209915548130abbf763f2cb7224a7c050e7d2c
```

Confirmed running:



```
PS C:\Users\HP> docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS
          PORTS      NAMES
15d247e5af8c   bde2020/hadoop-historyserver   "/entrypoint.sh /run…"   47 seconds ago   Up 46 seconds (h
ealthy)   8188/tcp   hadoop-historyserver
16cb27b41b06   bde2020/hadoop-datanode        "/entrypoint.sh /run…"   55 seconds ago   Up 54 seconds (h
ealthy)   9864/tcp   hadoop-datanode
```

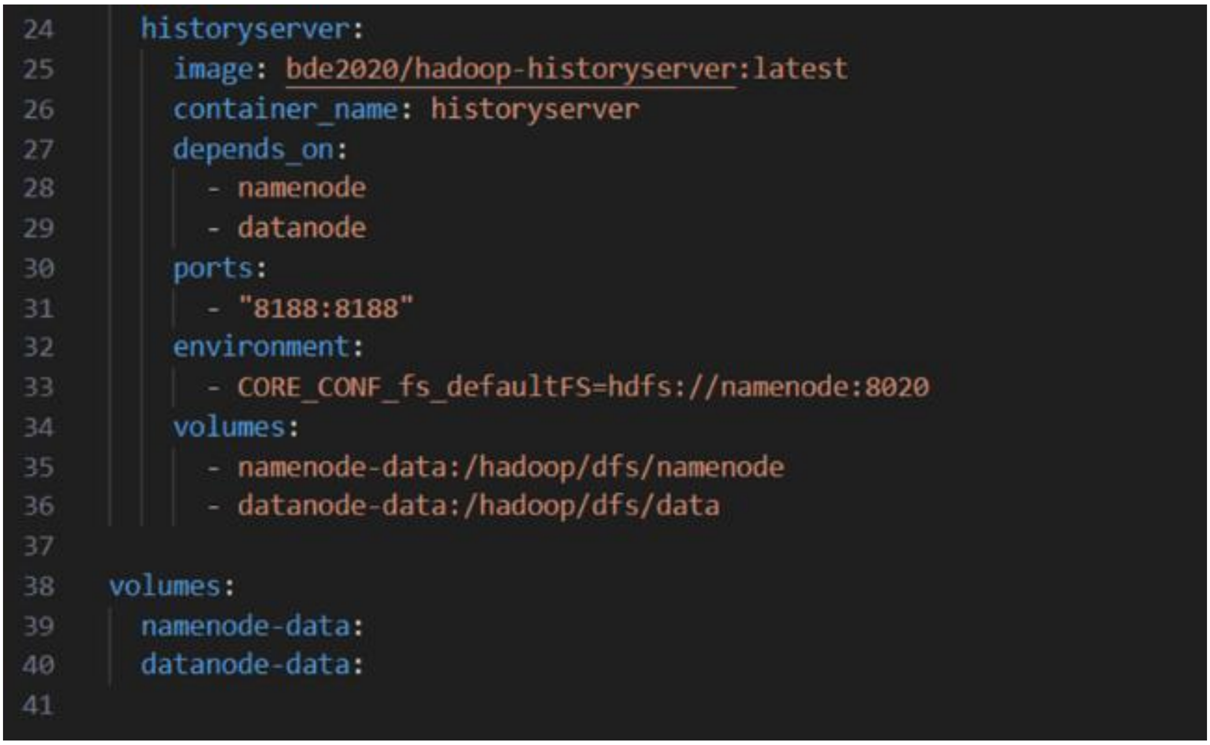
Created docker-compose.yaml and the text file (ex01.txt) in ex01 directory:



docker-compose.yaml

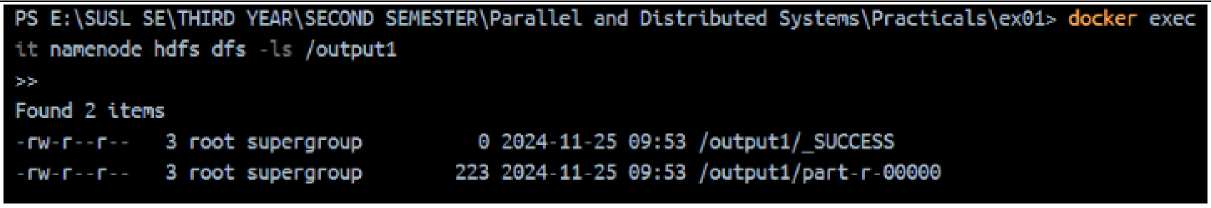
```yaml
services:
  namenode:
    image: bde2020/hadoop-namenode:latest
    container_name: namenode
    environment:
      - CLUSTER_NAME=ShopSmartCluster
      - CORE_CONF_fs_defaultFS=hdfs://namenode:8020
    ports:
      - "9870:9870"
      - "9000:9000"
    volumes:
      - namenode-data:/hadoop/dfs/namenode

  datanode:
    image: bde2020/hadoop-datanode:latest
    container_name: datanode
    environment:
      - CORE_CONF_fs_defaultFS=hdfs://namenode:8020
    volumes:
      - datanode-data:/hadoop/dfs/data
    depends_on:
      - namenode
```



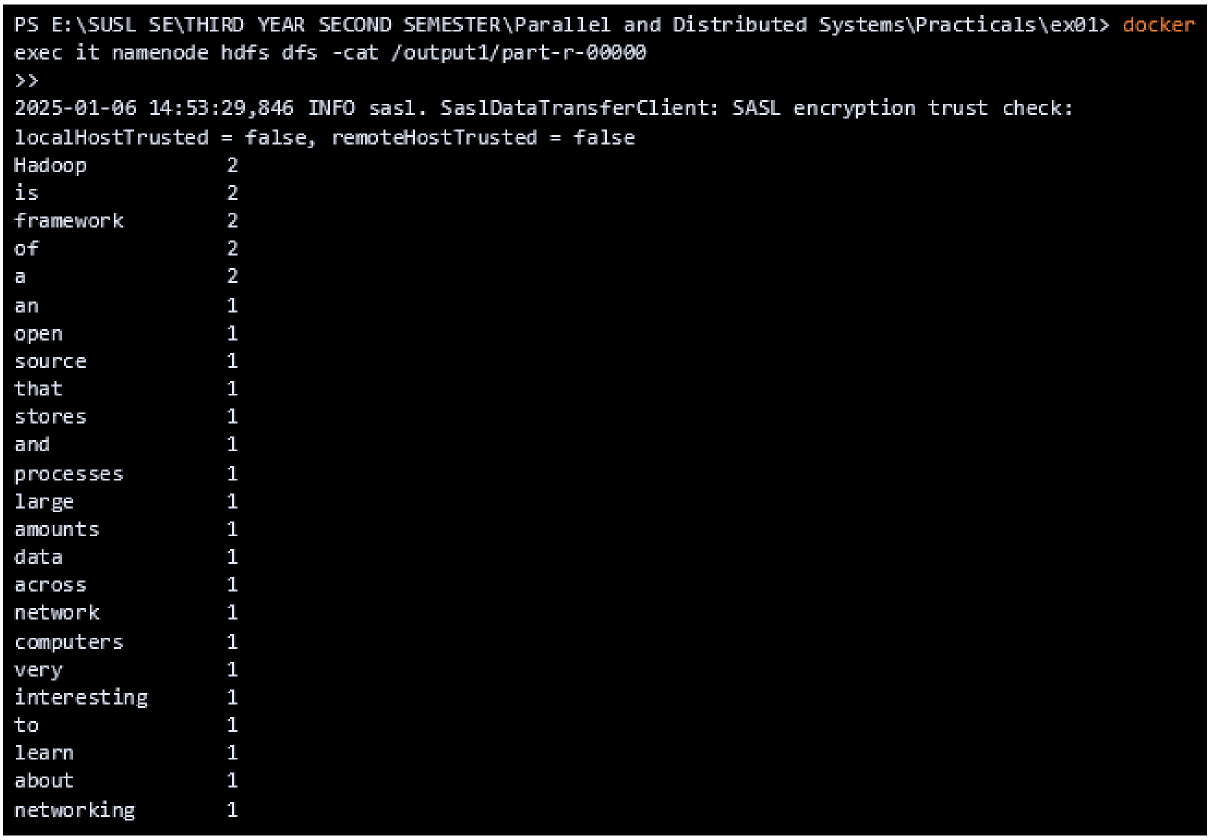
```yaml
  historyserver:
    image: bde2020/hadoop-historyserver:latest
    container_name: historyserver
    depends_on:
      - namenode
      - datanode
    ports:
      - "8188:8188"
    environment:
      - CORE_CONF_fs_defaultFS=hdfs://namenode:8020
    volumes:
      - namenode-data:/hadoop/dfs/namenode
      - datanode-data:/hadoop/dfs/data

volumes:
  namenode-data:
  datanode-data:
```



```
PS E:\SUSL SE\THIRD YEAR\SECOND SEMESTER\Parallel and Distributed Systems\Practicals\ex01> docker exec
it namenode hdfs dfs -ls /output1
>>
Found 2 items
-rw-r--r--   3 root supergroup          0 2024-11-25 09:53 /output1/_SUCCESS
-rw-r--r--   3 root supergroup        223 2024-11-25 09:53 /output1/part-r-00000
```

Deployed the cluster and uploaded files:



```
PS E:\SUSL SE\THIRD YEAR SECOND SEMESTER\Parallel and Distributed Systems\Practicals\ex01> docker
exec it namenode hdfs dfs -cat /output1/part-r-00000
>>
2025-01-06 14:53:29,846 INFO sasl. SaslDataTransferClient: SASL encryption trust check:
localHostTrusted = false, remoteHostTrusted = false
Hadoop          2
is              2
framework       2
of              2
a               2
an              1
open            1
source          1
that            1
stores          1
and             1
processes       1
large           1
amounts         1
data            1
across          1
network         1
computers       1
very            1
interesting     1
to              1
learn           1
about           1
networking      1
```