Steps to Set Up Hadoop with Docker

Jithendrian Sundaravaradan [*]

January 26, 2025

1 Install Docker Desktop on Windows

1. Go to `https://www.docker.com/products/docker-desktop` and download Docker Desktop for Windows (make sure to choose the x64 version).
2. Install Docker Desktop and follow the on-screen instructions.

2 Pull the Required Docker Images

You will use the `bde2020/hadoop` image, which is widely used for setting up Hadoop clusters in Docker. Run the following commands to pull the appropriate images for NameNode and DataNode:

```
docker pull bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
docker pull bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
```

3 Create a `docker-compose.yml` File

In the folder where you want to store your Hadoop configuration, create a `docker-compose.yml` file. Here's an example configuration:

```
version: '3'
services:
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    environment:
      - CLUSTER_NAME=test
    ports:
      - "9870:9870"
      - "9000:9000"
```

[*] With the help of prompt engineering

```
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
    networks:
      - hadoop

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode
    environment:
      - CLUSTER_NAME=test
    ports:
      - "9864:9864"
    volumes:
      - hadoop_datanode:/hadoop/dfs/data
    networks:
      - hadoop

volumes:
  hadoop_namenode:
  hadoop_datanode:

networks:
  hadoop:
```

This configuration will set up two services: `namenode` and `datanode` with the appropriate ports and environment variables.

docker compose file

4 Run Docker Compose

Once you've created your `docker-compose.yml` file, navigate to the folder where it is located and run the following commands:

1. Stop any running containers (if applicable):

   ```
   docker-compose down
   ```

2. Start the containers in detached mode:

   ```
   docker-compose up -d
   ```

5 Verify Container Status

Check if the containers are up and running using the following command:

```
docker ps -a
```

You should see the `namenode` and `datanode` containers listed with the status as `Up`.

6 Access the Containers for Debugging (if needed)

If you need to interact with the containers for debugging or further configuration, you can access the container shell:

1. For `NameNode`:

   ```
   docker exec -it namenode bash
   ```

2. For `DataNode`:

   ```
   docker exec -it datanode bash
   ```

7 Check Logs for Errors (if containers are exiting)

If the containers are still exiting, check the logs for any error messages:

```
docker logs namenode
docker logs datanode
```

This removes all unused containers, images, and volumes.

8 Adjust Docker Resources

Make sure Docker has enough resources allocated to run Hadoop:

1. Open Docker Desktop.
2. Go to **Settings ¿ Resources**.
3. Adjust CPU, Memory, and Disk allocation as needed.

9 Summary

By following these steps, you can set up Hadoop with Docker on your Windows machine using the bde2020/hadoop images for the NameNode and DataNode. The key steps include pulling the appropriate Docker images, creating a proper docker-compose.yml file, and using Docker commands to start and monitor the containers.

10 Clean Up (if needed)

If there are any issues, you may need to clean up your Docker environment:

10.1 All containers

```
docker system prune -a
```

10.2 Specific containers

10.2.1 List All Docker Containers (including stopped ones)

Run this command to see the list of all containers:

```
docker ps -a
```

Identify the containers related to Hadoop (such as namenode and datanode).

10.2.2 Remove Specific Containers

To remove specific containers (such as namenode and datanode), use the following command:

```
docker rm <container_id>
```

For example:

```
docker rm namenode datanode
```

10.2.3 Remove Specific Docker Images

To remove the Hadoop images from your system (such as bde2020/hadoop-namenode and bde2020/hadoop-datanode), use the following command:

```
docker rmi <image_name>
```

For example:

```
docker rmi bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8 bde2020/hadoop-datanode:2.0.0-had
```

10.2.4 Optionally Clean Unused Volumes

If you want to remove unused volumes related to Hadoop (e.g., `hadoop_namenode` and `hadoop_datanode`), run:

```
docker volume prune --filter "label=hadoop"
```

10.2.5 Remove Only Unused Containers, Networks, and Volumes (Optional)

If you want to remove only the unused containers, networks, and volumes, but not everything else, use:

```
docker system prune --volumes
```

This will only clean up the unused resources related to Hadoop (containers, networks, volumes) without removing everything in your Docker setup.