CSCI461: Introduction to Big Data (Big Data Analytics)

Lab #8: Apache Hadoop & MapReduce

Agenda

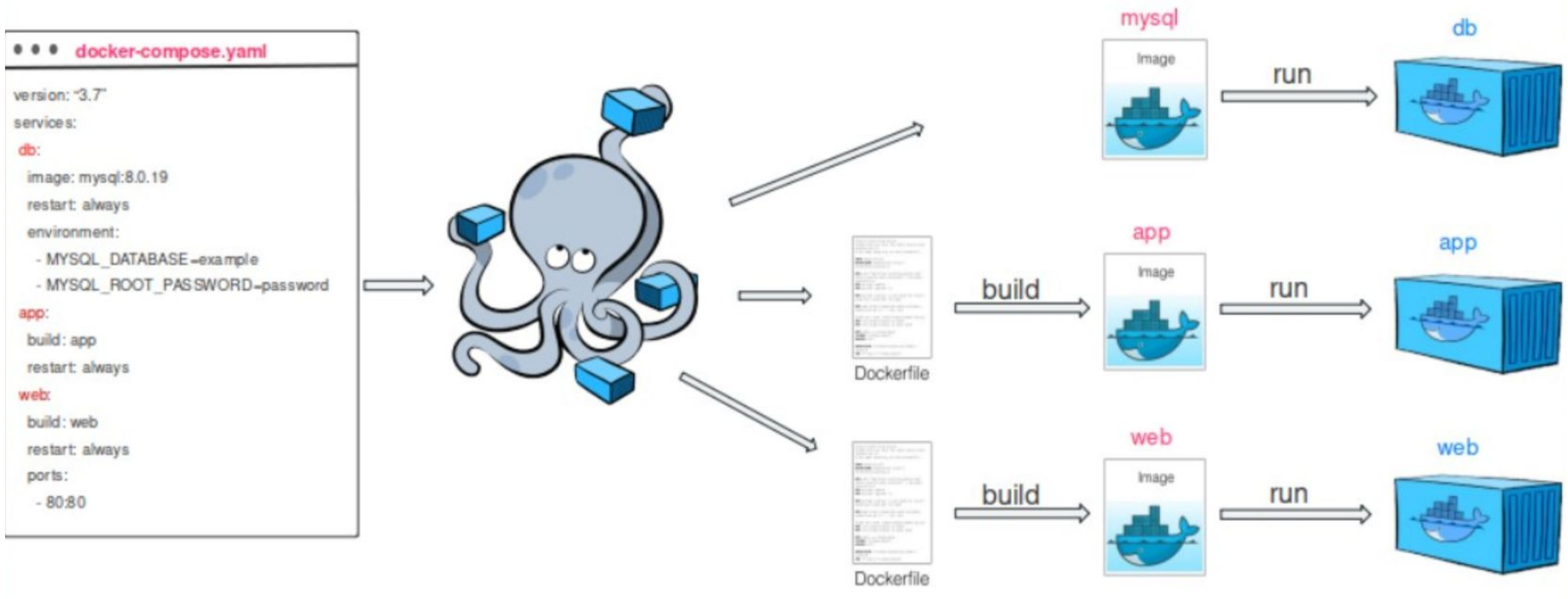
- What is Hadoop? [Lecture Slides]
- Setup the Environment
- Apache Hadoop on Docker using Docker Compose
- HDFS Commands
- Hadoop UI
- MapReduce Example
- Today’s Task

Setup the Environment

- You need to have the following java version:
  - openjdk version "1.8.0_232" or any similar version [Check this]
- Install IntelliJ IDEA Community Edition. [Check this]

Apache Hadoop on Docker using Docker Compose

- Docker Compose is a tool for defining and running multi-container Docker applications.
- It allows you to define an entire application stack, including services, networks, and volumes, in a single file called docker-compose.yml.
- This file provides a concise and human-readable way to specify how different Docker containers should interact with each other.



docker-compose.yaml

```yaml
version: "3.7"
services:
 db:
  image: mysql:8.0.19
  restart: always
  environment:
   - MYSQL_DATABASE=example
   - MYSQL_ROOT_PASSWORD=password
 app:
  build: app
  restart: always
 web:
  build: web
  restart: always
  ports:
   - 80:80
```

Apache Hadoop on Docker using Docker Compose

- Download the folder from the following link.
- Go to the directory where the docker-compose.yml exists:
  - Run the following command: `docker-compose up`
- **NOTE**: Make sure you don’t have any other related hadoop containers.
- **NOTE**: Make sure you have installed the docker-compose utility if required.

Apache Hadoop on Docker using Docker Compose

Service	**NameNode**	**DataNode**	**ResourceManager**	**NodeManager**	**HistoryServer**
Port	9870	9864	8088	8042	8188

HDFS Commands

- To verify that hadoop and all daemons all running successfully use:
  - jps
- To list all files and directories in HDFS use:
  - hadoop fs -ls / or hdfs dfs -ls /
- To create a new directory in HDFS use:
  - hadoop fs -mkdir <dir/> or hdfs dfs -mkdir <dir/>
- To copy data from local to HDFS use:
  - hadoop fs -put <local-file-path> <hdfs-file-path> or hdfs dfs -put <local-file-path> <hdfs-file-path>
- To copy data from HDFS to local use:
  - hadoop fs -get <local-file-path> <hdfs-file-path> or hdfs dfs -get <local-file-path> <hdfs-file-path>

HDFS Commands (CONT.)

- To view data of a file in HDFS use:
  - hadoop fs -cat <file-path-in-hdfs> or hdfs dfs -cat <file-path-in-hdfs>
- To move a file from one location to another in HDFS use:
  - hadoop fs -mv <source-path-in-hdfs> <dest-path-in-hdfs> or hdfs dfs -mv <source-path-in-hdfs> <dest-path-in-hdfs>
- To copy a file from one location to another in HDFS use:
  - hadoop fs -cp <source-path-in-hdfs> <dest-path-in-hdfs> or hdfs dfs -cp <source-path-in-hdfs> <dest-path-in-hdfs>
- To copy data from local to HDFS use:
  - hadoop fs -copyFromLocal <local-file-path> <hdfs-file-path> or hdfs dfs -copyFromLocal <local-file-path> <hdfs-file-path>
- To copy data from HDFS to local use:
  - hadoop fs -copyToLocal <local-file-path> <hdfs-file-path> or hdfs dfs -copyToLocal <local-file-path> <hdfs-file-path>

HDFS Commands (CONT.)

- To move data from local to HDFS use:
  - hadoop fs -moveFromLocal <local-file-path> <hdfs-file-path> or hdfs dfs -moveFromLocal <local-file-path> <hdfs-file-path>
- To move data from HDFS to local use:
  - hadoop fs -moveToLocal <local-file-path> <hdfs-file-path> or hdfs dfs -moveToLocal <local-file-path> <hdfs-file-path>
- To remove a file in HDFS use:
  - hadoop fs -rm <file-path-in-hdfs> or hdfs dfs -rm <file-path-in-hdfs>
- To create a file in a specific location in HDFS use:
  - hadoop fs -touchz <file-path-and-name-in-hdfs> or hdfs dfs -touchz <file-path-and-name-in-hdfs>

NOTE: Almost all normal terminal commands are used in HDFS.

Hands-on Example on MapReduce

- We’re going to implement a trivial example on Hadoop utilizing MapReduce for Word Count.
- Assuming we have a cluster of 100 nodes, totaling 100 TB disk space, 10 TB RAM, etc.
- We will read a text file, and we want to produce the number of occurrences of each word.
- Attached with lab slides is the Java code used and how to run the example in addition to the data file used.
- **Let’s open IntelliJ.**

Hands-on Example on MapReduce (CONT.)

- MapReduce jobs consists mainly of 3 classes:
  - **The Driver**: the main class of the job, contains the configurations of the job, submitting the job to the cluster, configuring Mapper and Reducer
  - **The Mapper**: The map method to specify the map behavior, each map deals with one split at a time.
  - **The Reducer**: The reduce method to specify the reduce behavior.
- We will explain the code in intelliJ.
- Add the following jars to the libraries in intelliJ: [Jars]
  - hadoop-common-3.3.1.jar
  - hadoop-mapreduce-client-common-3.3.1.jar
  - hadoop-mapreduce-client-core-3.3.1.jar
  - hadoop-client-api-3.3.1.jar
  - hadoop-client-runtime-3.3.1.jar
  - hadoop-hdfs-3.3.1.jar
- To know more about how to write a MapReduce Job.

Hands-on Example on MapReduce (CONT.)

- After writing your MapReduce job, we need to have the jar file of the job. [build]
- Then move it to the container. [Same for the data file]
  - docker cp <file> container-name:<path-in-the-container>
- Move the data file from the container to the HDFS:
  - hdfs dfs -copyFromLocal <path-of-data-file> <hdfs-path>
- Finally run the following commands to submit the job:
  - hadoop jar <jar-path> <DriverClassNameWithout.java> <input-path-on-hdfs> <output-path-on-hdfs>

NOTE: Output directory shouldn't be created before running the job.

Today’s Task

- You’re required to implement the **Inverted Index** algorithm using **MapReduce.**
- Refer to the following link to understand what is inverted index is?
- On Moodle you’re required to submit **ONE** zip file contains:
    - Driver Class (.java)
    - Mapper Class (.java)
    - Reducer Class (.java)
    - Screenshot from your ResourceManager UI. (showing the job)
    - Screenshot from HDFS in terminal/bash. (showing the input and output directories)
    - The output file of running your job.
- **CHEATING = ZERO.**
- DEADLINE: ***Saturday, April 20th 2024 at 11:59 PM***. (No extension will be made)

Thanks