Big Data Analytics

Isara Anantavrasilp

Lecture 11: YARN and Cloudera Services

MapReduce Extensions

- MapReduce provides fundamental big data operations in Hadoop
- There are many projects that work on top of MapReduce, e.g.
  - **Apache Hive**: Data warehouse software that allows manipulating big data using SQL
  - **Apache Pig**: Platform for analyzing big data via high-level programming language
- However, MapReduce runs as a long linear process
  - Good for large dataset
  - Bad for small low-latency or interactive applications
  - Also bad for parallel jobs

YARN

- YARN (Yet Another Resource Negotiator): Job scheduling application in Hadoop
  - YARN does not concern how the jobs are run
  - It concerns only on job scheduling
- YARN architecture consists of two components
  - **Resource Manager (RM):** Manages resources across the Hadoop cluster
  - **Node Manager (NM):** Runs on each host and manages individual machine

Containers

- RM and NM manage resources of a container
- **Container: A** logical representation of actual computational resources e.g. CPU, memory, etc.
  - When MapReduce runs on top of YARN, it will run on a dedicated container
- YARN handles only task scheduling. It does not concern any application-level process, monitoring or error handling
- This makes YARN simple and compatible with all kinds of application including shell scripts and compiled apps

YARN Application

- Application that runs on YARN is called YARN Application
- YARN Application consists of two components:
  - **Application Master** (**AM**): Coordinates the overall process flows
  - Application codes that will run on each node (more precisely, the ***specification*** describing the code to be executed)
- In our previous MapReduce exercises, JobTracker implements the Application Master and tasks are the application codes

YARN Process

- The client submits YARN Application to the system
- When the app is started the client calls the Resource Manager and requests a container to execute the Application Master
- RM register a container to run the AM
- The AM starts in the provided container and registers itself with the RM
- It then begins the process of negotiating its required resources
  - Containers
  - Concrete resources (e.g. amount of CPU or memory)

YARN Process (2)

- The RM provides the AM with the details of the containers it has been allocated
- AM then communicates with the Node Managers to start the application-specific task for each container
- This is done by providing the NM with the specification of the application to be executed
  - JAR file
  - A script, a path to a local executable
  - Anything else that the NM can invoke.
- Lastly, NM instantiates the container for the application code and starts the application based on the provided specification

Implementing YARN Application

- Writing YARN Application is not easy
- YARN API is complicated and the developers must handle errors and process monitoring themselves
- Normally, we do not implement YARN app directly, especially in simple big data project
- We implement our programs on top of existing frameworks e.g. MapReduce, Apache Twill or Clourdera Kitten

Example of Projects that Run on YARN

- **Apache TEZ**: Application framework which allows for pipelining data. It models MapReduce jobs as directed-acyclic-graph DAG of tasks.
  - Map/Reduce tasks are run through DAG pipeline without storing on HDFS.
  - This reduces time and I/O overhead
- **Apache Hive:** An engine for querying data stored on HDFS through standard SQL syntax
- **Apache Spark**: A cluster computing framework. It provides large-scale data processing engine and tools such as SQL processing, streaming data, machine learning, and graph processing

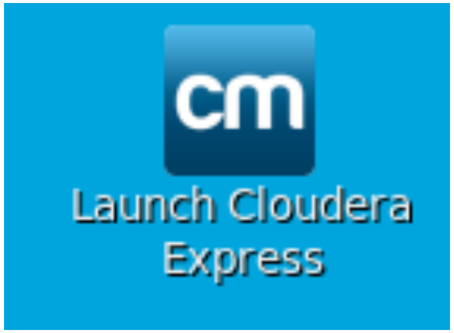
Exploring Cloudera QuickStart

- Activate Cloudera Express to manage our cluster (of one)
- Start Cloudera Manager
- Start Cloudera services via Cloudera Manager
- Run MapReduce jobs with Cloudera Manager
  - Monitor cluster performances

Launching Cloudera Manager

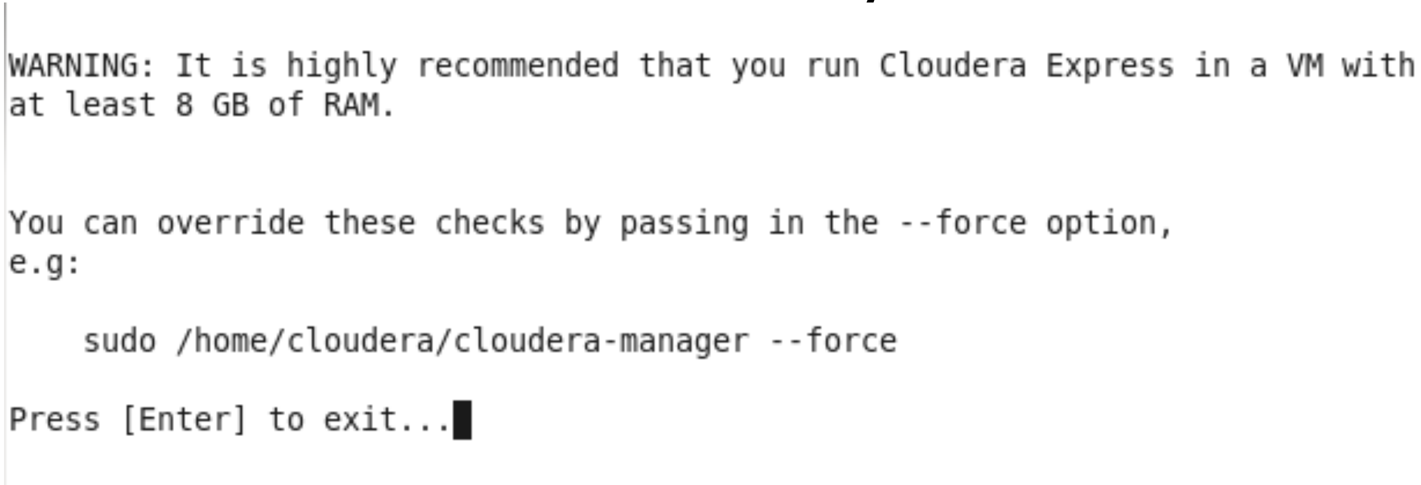
- Before starting the VM, set the RAM to be more than 8GB if you could

- Cloudera provides two versions of the manager, **Express** (free) and **Enterprise** one
- We will go for the free version
- Click Launch Cloudera Express



Launching Cloudera Manager

- If you have low memory, the system may inform you to increase VM memory



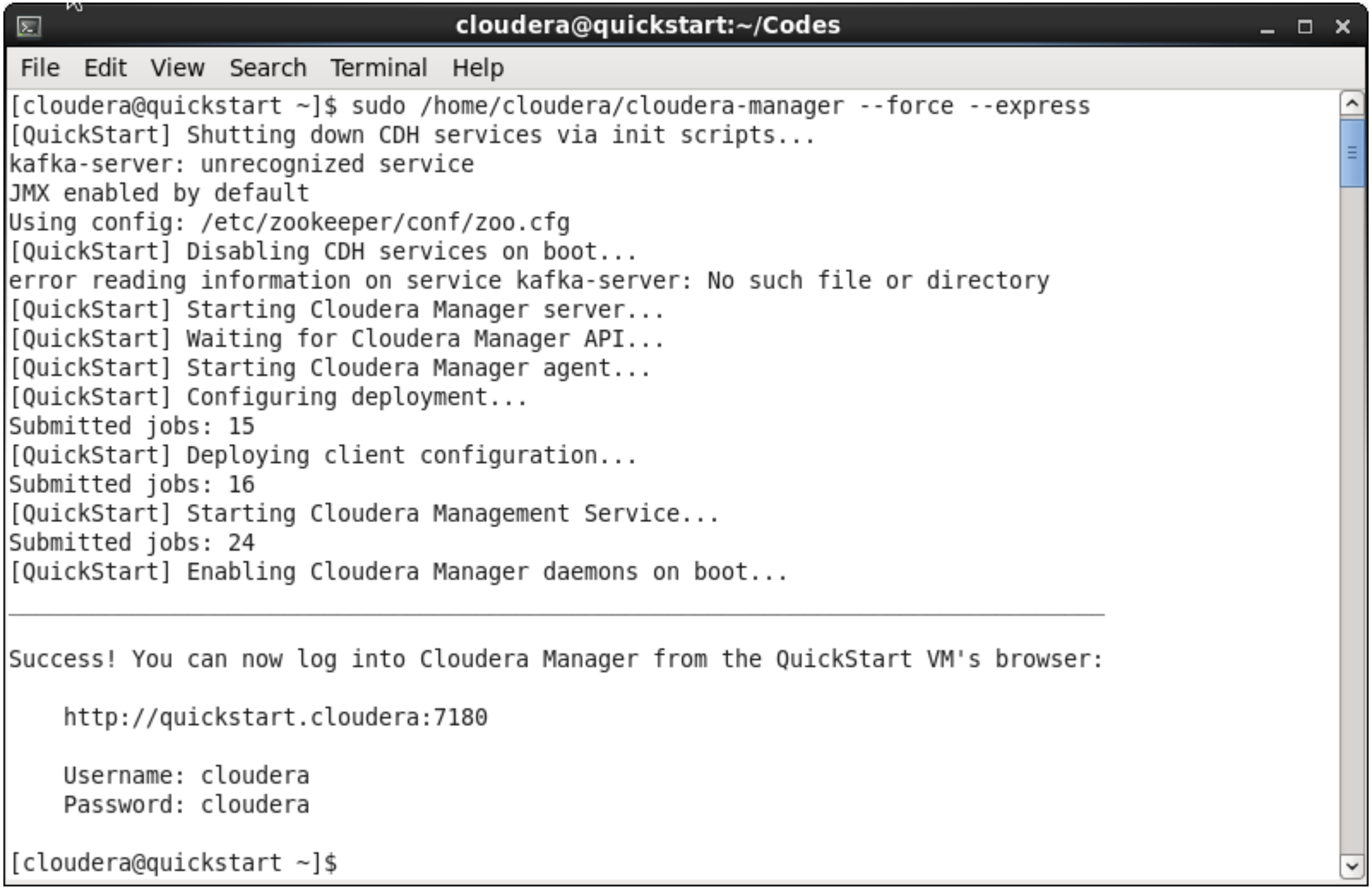
```
WARNING: It is highly recommended that you run Cloudera Express in a VM with
at least 8 GB of RAM.


You can override these checks by passing in the --force option,
e.g:

    sudo /home/cloudera/cloudera-manager --force

Press [Enter] to exit...
```

- You can also force the Manager to start using the command:

```
sudo /home/cloudera/cloudera-manager –
force --express
```

Launching Cloudera Manager

-
-



cloudera@quickstart:~/Codes
File Edit View Search Terminal Help

```
[cloudera@quickstart ~]$ sudo /home/cloudera/cloudera-manager --force --express
[QuickStart] Shutting down CDH services via init scripts...
kafka-server: unrecognized service
JMX enabled by default
Using config: /etc/zookeeper/conf/zoo.cfg
[QuickStart] Disabling CDH services on boot...
error reading information on service kafka-server: No such file or directory
[QuickStart] Starting Cloudera Manager server...
[QuickStart] Waiting for Cloudera Manager API...
[QuickStart] Starting Cloudera Manager agent...
[QuickStart] Configuring deployment...
Submitted jobs: 15
[QuickStart] Deploying client configuration...
Submitted jobs: 16
[QuickStart] Starting Cloudera Management Service...
Submitted jobs: 24
[QuickStart] Enabling Cloudera Manager daemons on boot...
________________________________________________________________________________

Success! You can now log into Cloudera Manager from the QuickStart VM's browser:

    http://quickstart.cloudera:7180

    Username: cloudera
    Password: cloudera

[cloudera@quickstart ~]$
```

Let’s try some Hadoop job

- Start some Hadoop job that we did before, e.g.:

```
hadoop jar law.jar LineAndWordCount
t8.shakespeare.txt out
```

- Can you run it?

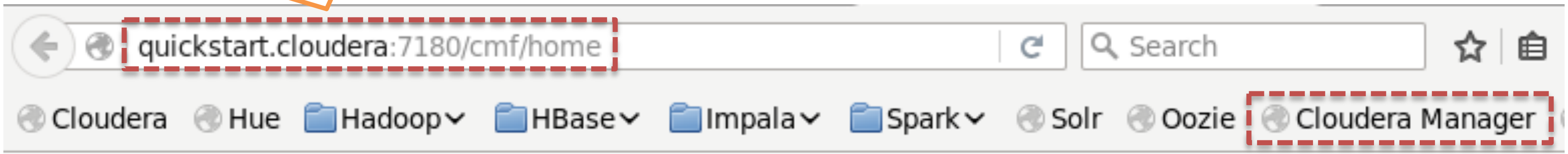
- You might see error like this:

```
18/10/15 03:07:40 INFO client.RMProxy: Connecting to
ResourceManager at quickstart.cloudera/10.0.2.15:8032
18/10/15 03:07:40 WARN ipc.Client: Failed to connect
to server: quickstart.cloudera/10.0.2.15:8020: try
once and fail.
```

Starting HDFS and YARN

- HDFS and YARN are now managed with the Manager
- We have to start them using the web-based UI

Type the following URL to start



quickstart.cloudera:7180/cmf/home

Or simply click on the bookmark