Инструкция по установке

APACHE HADOOP

в DOCKER

ver 27

(05.12.2023)

Оглавление

[**1.** **Установить  Docker Desktop** 2](#_Toc151588908)

[**2.** **Установка Hadoop ver1** 6](#_Toc151588909)

[**3.** **Установка Hadoop ver2** 7](#_Toc151588910)

[**4.** **Установка cloudera** 14](#_Toc151588911)

[**5.** **Установка Hive** 17](#_Toc151588912)

[**6.** **Пример создания и настройки своего контейнера** 20](#_Toc151588913)

[**7.** **Домашнее задание 1** 21](#_Toc151588914)

[**8.** **Домашнее задание 2** 22](#_Toc151588915)

[**9.** **Домашнее задание 3 part1** 25](#_Toc151588916)

Видеообзор инструкции:

<https://youtu.be/DmqFljyi5r0>

# **Установить  Docker Desktop**

<https://www.docker.com/products/docker-desktop/>

Обратите внимание владельцы MacBook с процессором M1: будут проблемы с устаревшими образами виртуалок



Mac M1.

Перед первой командой необходимо ввести в терминале команду:  
export DOCKER\_DEFAULT\_PLATFORM=linux/amd64

Далее идти по инструкции

Через Safari логин и пароль не вводились, заходит только через Chrome.

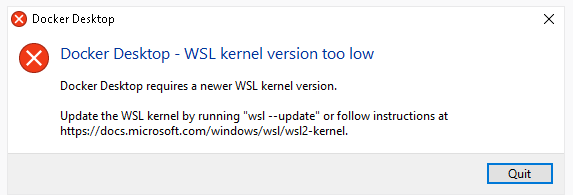
**Для Windows:**

https://docs.docker.com/desktop/install/windows-install/

https://www.docker.com/products/docker-desktop/

В Windows HOME - wsl работать не будет, нужно обновить до PRO

Обновить WSL если потребуется:

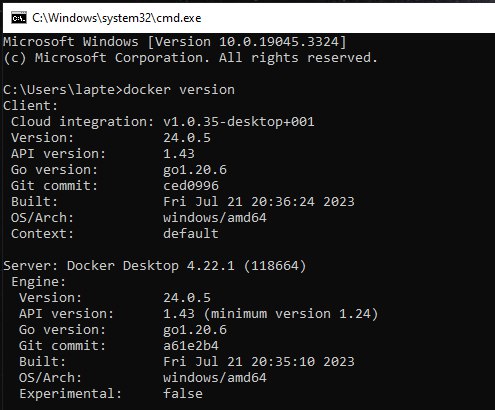


Включить виртуализацию в БИОС, если выдает подобное предупреждение:

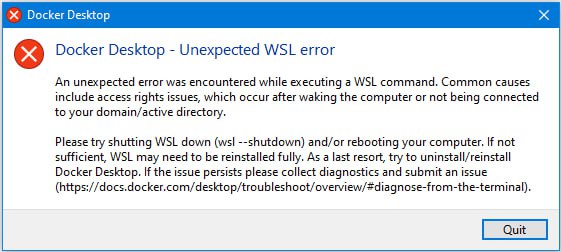


При установке Докера в windows отказаться от WSL по умолчанию (снять/поставить галку)

Проверить версию:

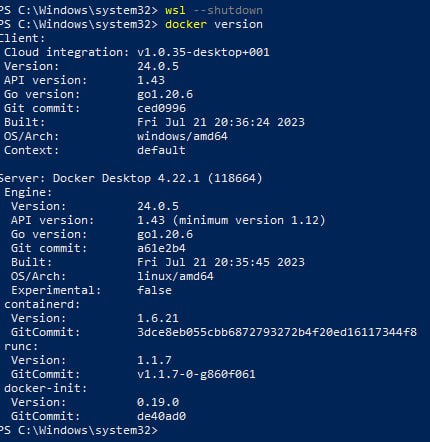


Если такое сообщение:



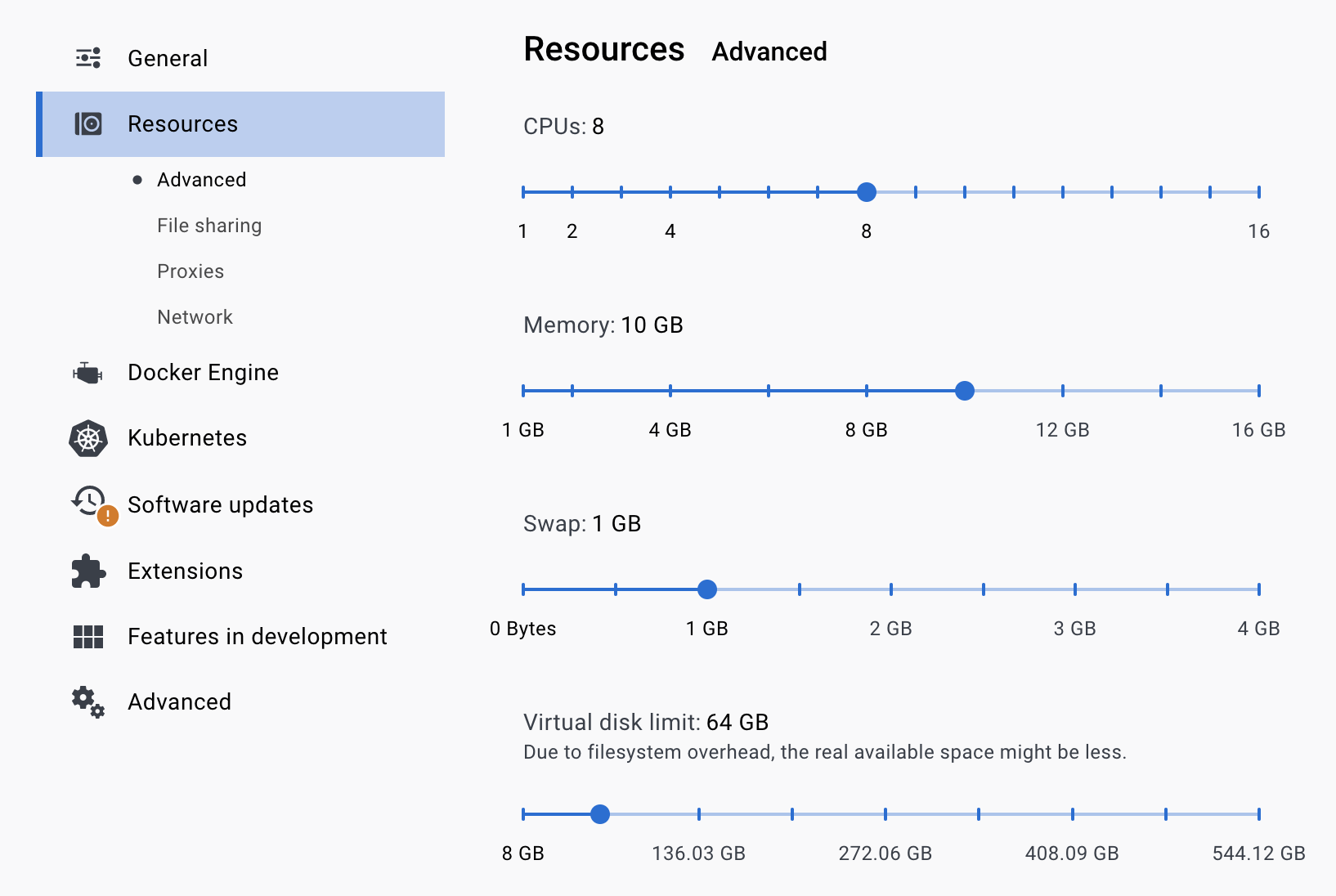
То выполнить

wsl —shutdown

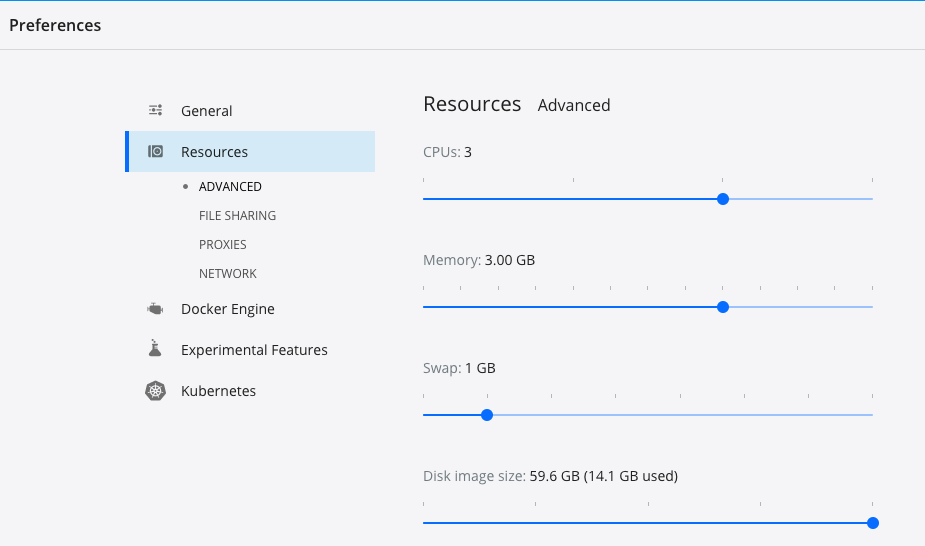


**Добавить докеру больше ресурсов памяти и ядер.**

Зайти в настройки докера и увеличить:



Рабочий вариант:



После работы с docker его желательно выключать, так как он использует дополнительные ресурсы.

# **Установка Hadoop ver1**

* **Скопируйте репозиторий:** [**https://github.com/dertty/apache-spark-docker**](https://github.com/dertty/apache-spark-docker)
* **Перейдите в папку с файлом (docker-compose.jml)**
* **cd /Volumes/Data/docker/apchsprk/hdfs**
* **Выполните команду:**
* **docker compose up -d**

# **Установка Hadoop ver2**

Релиз скачивает около 600 мб данных занимает около 1 Гб на жестком диске

Основная инструкция по установке:

<https://registry.hub.docker.com/r/apache/hadoop>

Тут посмотреть версию последнего релиза Hadoop:

<https://hadoop.apache.org/releases.html>

**Создать папку на локальном компьютере (где установлен докер) (на вашем компьюте создать новую папку, "желательно"с именем hadoop и "желательно", чтобы в имени пути не было пробелов и русских букв)**

Например путь созданной вами структуры папок:

для MacOS **/Volumes/Data/docker/hadoop**

Для WINDOWS **C:\Data\docker\hadoop**

#### В созданной выше папке создать файл docker-compose.yaml (или вставить файлы из архива)

#### Для этого можно воспользоваться блокнотом, в нем создать новый документ, вставить в него текст ниже:

#### Проверить, что расширение файла изменилось (не .txt)

#### Красным выделена последняя версия, которую посмотрели по ссылке ранее, необходимо ее указать

version: "2"

services:

namenode:

image: apache/hadoop:**3.3.6**

hostname: namenode

command: ["hdfs", "namenode"]

ports:

- 9870:9870

env\_file:

- ./config

environment:

ENSURE\_NAMENODE\_DIR: "/tmp/hadoop-root/dfs/name"

datanode:

image: apache/hadoop:**3.3.6**

command: ["hdfs", "datanode"]

env\_file:

- ./config

resourcemanager:

image: apache/hadoop:**3.3.6**

hostname: resourcemanager

command: ["yarn", "resourcemanager"]

ports:

- 8088:8088

env\_file:

- ./config

volumes:

- ./test.sh:/opt/test.sh

nodemanager:

image: apache/hadoop:**3.3.6**

command: ["yarn", "nodemanager"]

env\_file:

- ./config

#### В папке создать файл config содержанием ниже:

#### Проверить, что расширение файла изменилось (не .txt)

HADOOP\_HOME=/opt/hadoop

CORE-SITE.XML\_fs.default.name=hdfs://namenode

CORE-SITE.XML\_fs.defaultFS=hdfs://namenode

HDFS-SITE.XML\_dfs.namenode.rpc-address=namenode:8020

HDFS-SITE.XML\_dfs.replication=1

MAPRED-SITE.XML\_mapreduce.framework.name=yarn

MAPRED-SITE.XML\_yarn.app.mapreduce.am.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

MAPRED-SITE.XML\_mapreduce.map.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

MAPRED-SITE.XML\_mapreduce.reduce.env=HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

YARN-SITE.XML\_yarn.resourcemanager.hostname=resourcemanager

YARN-SITE.XML\_yarn.nodemanager.pmem-check-enabled=false

YARN-SITE.XML\_yarn.nodemanager.delete.debug-delay-sec=600

YARN-SITE.XML\_yarn.nodemanager.vmem-check-enabled=false

YARN-SITE.XML\_yarn.nodemanager.aux-services=mapreduce\_shuffle

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.maximum-applications=10000

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.maximum-am-resource-percent=0.1

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.resource-calculator=org.apache.hadoop.yarn.util.resource.DefaultResourceCalculator

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.queues=default

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.capacity=100

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.user-limit-factor=1

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.maximum-capacity=100

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.state=RUNNING

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.acl\_submit\_applications=\*

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.root.default.acl\_administer\_queue=\*

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.node-locality-delay=40

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.queue-mappings=

CAPACITY-SCHEDULER.XML\_yarn.scheduler.capacity.queue-mappings-override.enable=false

#### Открыть ТЕРМИНАЛ:

#### Для windows (Win+R, затем ввести cmd затем Enter)

#### 

#### Если в windows не выполняются команды в терминале, то использовать powershell <https://dzen.ru/a/XzVQXM0dchqol_jh>

#### Так же часть проблем можно решить запуском коммандера от имени администратора:

#### Поиск-командная строка-правая кнопка мыши-запуск от имени администратора

#### Перейти в папку, которую создавали ранее:

#### cd /Volumes/Data/docker/hadoop

Либо

#### cd "C:\Data\docker\hadoop"

Убедиться, что файлы присутствуют:

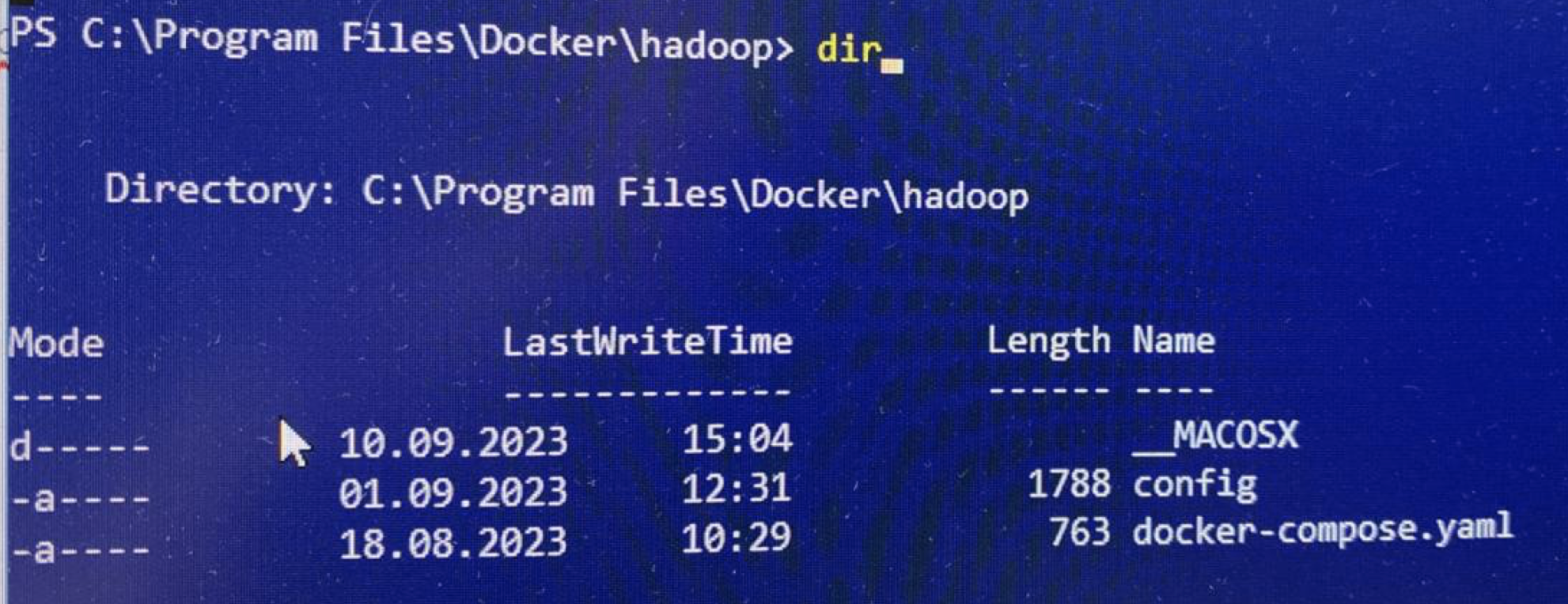
**ls -l**

-rw-r--r-- 1 hadoop apache 2547 Jun 23 15:53 config

-rw-r--r-- 1 hadoop apache 1533 Jun 23 16:07 docker-compose.yaml

Для windows:

**dir**



Выполнить команду:

docker-compose up -d

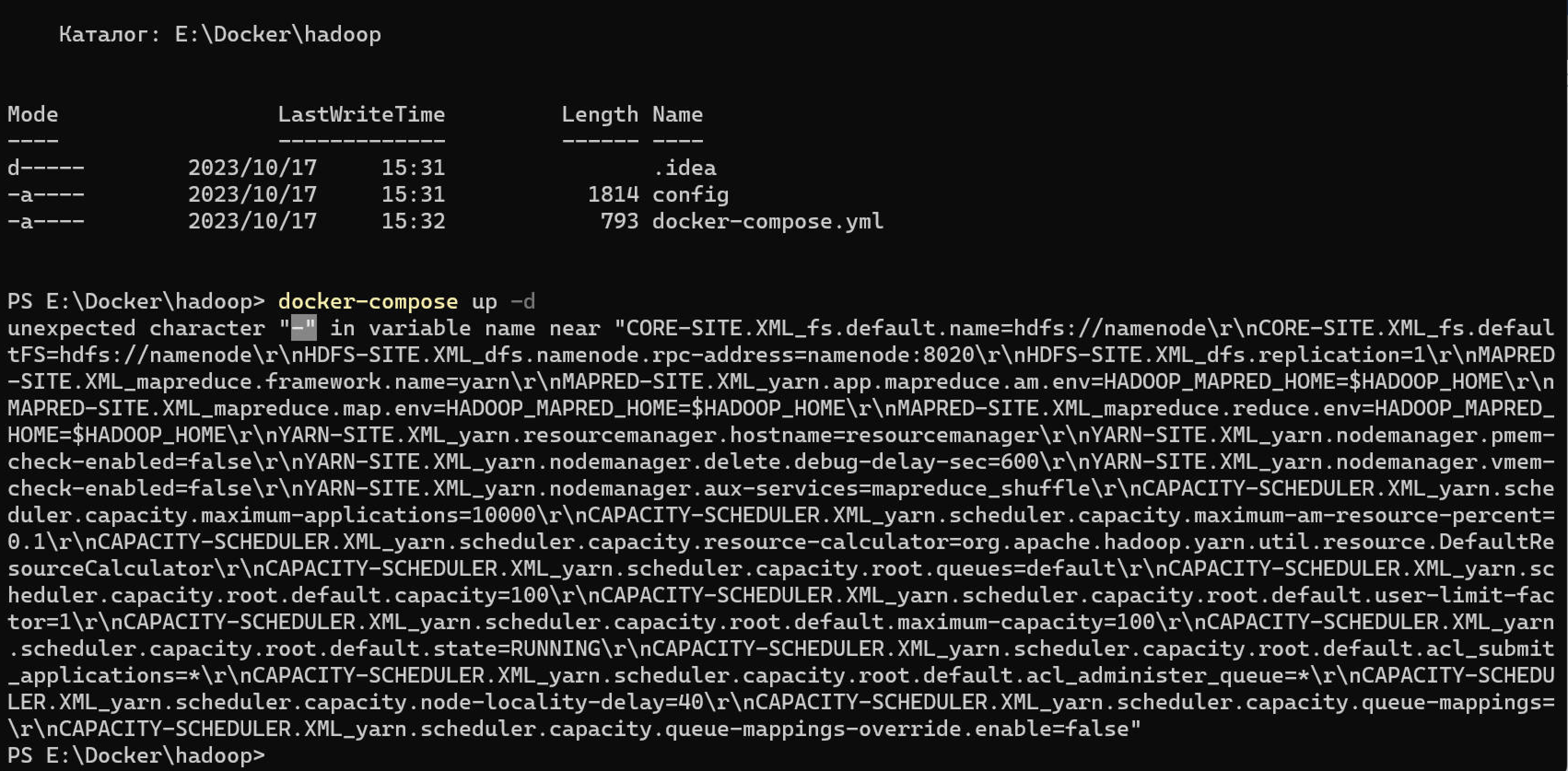
#### docker-compose up -d - эта команда посмотрит файлы (config и docker-compose.yaml) в локальной папке (/Volumes/Data/docker/hadoop) и по инструкциям в них начнет скачивать и устанавливать кластер hadoop

!!!!

Для корректного включения и выключения контейнеров нужно использовать команды docker-compose up -d и docker-compose down

!!!!

Если ошибка:



То скачать репозиторий из гитхаб.

Дождаться установки дистрибутива

**docker-compose up -d**

Creating network "docker-3\_default" with the default driver

Creating docker-3\_namenode\_1 ... done

Creating docker-3\_datanode\_1 ... done

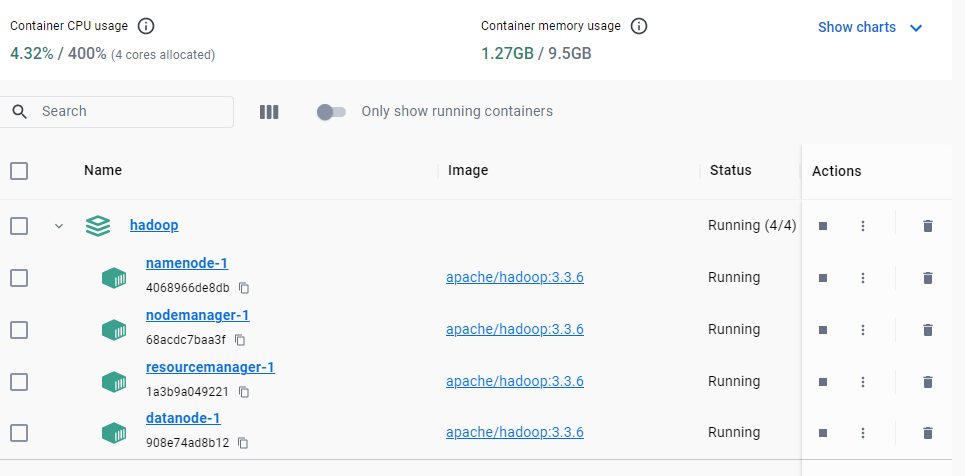
Creating docker-3\_nodemanager\_1 ... done

Creating docker-3\_resourcemanager\_1 ... done

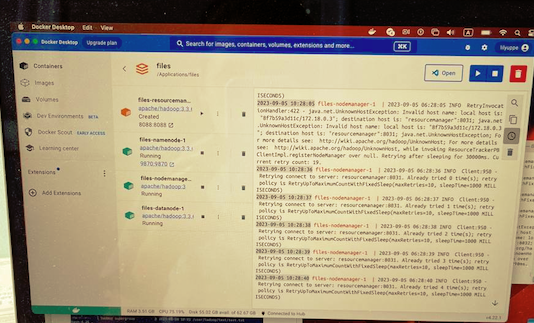
или так:



Убедиться, что у вас все ноды поднялись:



На этом скриншоте не работает ресурсменеджер, значит надо добавить памяти, по инструкции выше.

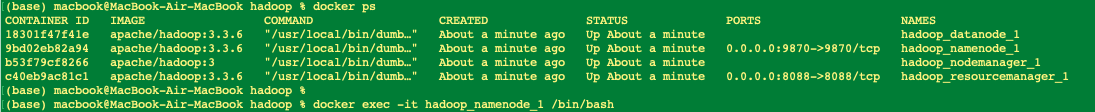


Выполнить команду

docker ps

Посмотреть название контейнера (**hadoop\_namenode\_1**) – может быть другим

Но именно NAMENODE



Выполнить команду

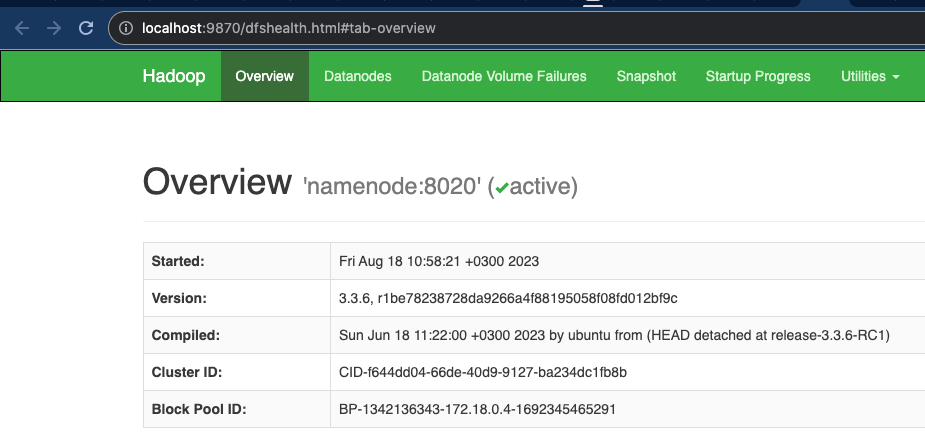
docker exec -it **hadoop-namenode-1** /bin/bash

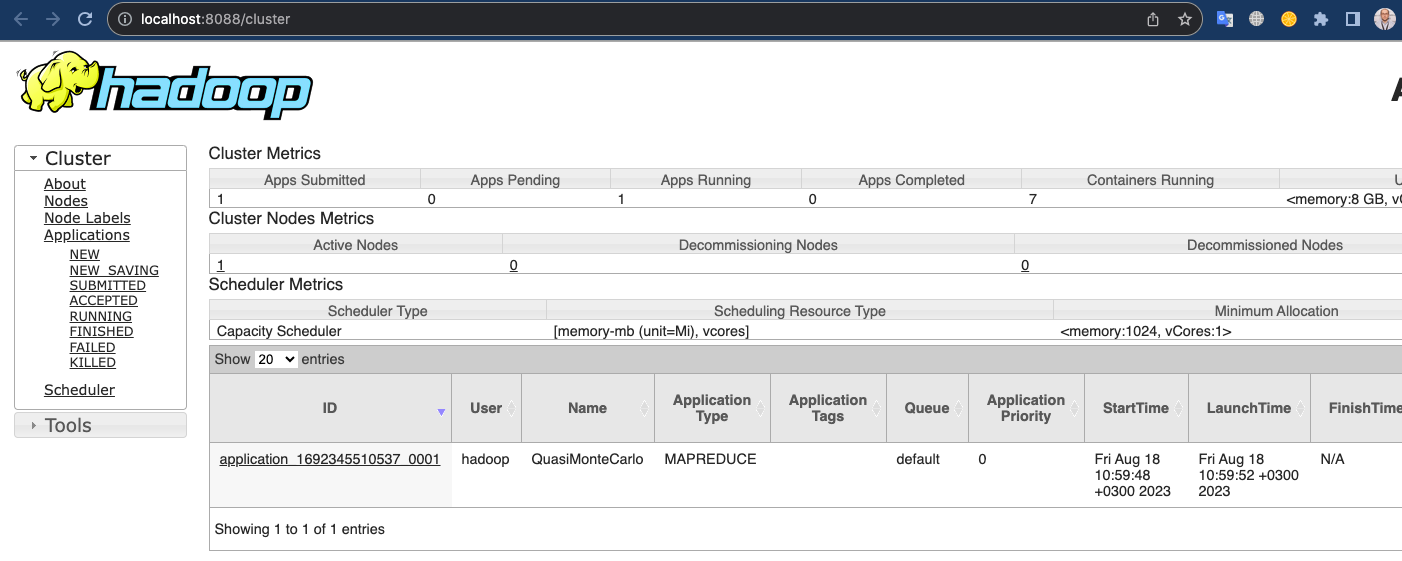
После этого у вас запустится виртуальный контейнер с Hadoop

Затем работу контейнера проверить по ссылкам (на уровне операционной системы):

Namenode UI <http://localhost:9870/>

ResourceManager UI <http://localhost:8088/>





whoami

В контейнере выполнить команды для проверки:

hdfs dfs -mkdir -p /user/root/

hdfs dfs -ls /

ИЛИ решилась запуском и докера и командной строки от имени администратора, потом docker-compose up -d , затем docker exec -it hadoop-namenode-1 /bin/bash

**Остановка контейнера**

Выйти из контейнера (на уровне контейнера)

CTRL+D

Выполнить в основной операционной системе:

docker-compose down

Обязательно, чтобы остановить контейнеры, иначе при повторном поднятии они будут глючить

Выключить Docker

ЕСЛИ какие то команды не выполняются, проверить, что все 4 ноды подняты:

docker ps

# **Установка cloudera**

**Так же можно установить cloudera, по желанию.**

Релиз скачивает около 1 Гб данных занимает около 4 Гб на жестком диске

Основная инструкция:

<https://github.com/emirkorkmaz/cloudera-quickstart-docker-compose>

**Создать папку на уровне операционной системы где установлен докер**

Например путь созданной вами структуры папок:

для MacOS **/Volumes/Data/docker/Cloudera\_Manager**

Для WINDOWS **C:\Data\docker\Cloudera\_Manager**

Скачать файл **docker-compose.yaml** по ссылке <https://github.com/emirkorkmaz/cloudera-quickstart-docker-compose/blob/master/cloudera-quickstart/docker-compose.yaml>

и поместить его в папку выше.

#### Либо в созданной выше папке создать файл docker-compose.yaml

version: '2'

services:

cloudera:

image: cloudera/quickstart:latest

privileged: true

hostname: quickstart.cloudera

command: /usr/bin/docker-quickstart

ports:

- "8020:8020" # HDFS

- "8022:22" # SSH

- "7180:7180" # Cloudera Manager

- "8888:8888" # Hue

- "11000:11000" # Oozie

- "50070:50070" # HDFS Rest Namenode

- "50075:50075" # HDFS Rest Datanode

- "2181:2181" # Zookeeper

- "8088:8088" # YARN Resource Manager

- "19888:19888" # MapReduce Job History

- "50030:50030" # MapReduce Job Tracker

- "8983:8983" # Solr

- "16000:16000" # Sqoop Metastore

- "8042:8042" # YARN Node Manager

- "60010:60010" # HBase Master

- "60030:60030" # HBase Region

- "9090:9090" # HBase Thrift

- "8080:8080" # HBase Rest

- "7077:7077" # Spark Master

tty: true

stdin\_open: true

volumes:

- /var/shared\_cloudera\_quickstart:/media/shared\_from\_local

#### Открыть ТЕРМИНАЛ:

#### Для windows (Win+R, затем ввести cmd затем Enter)

#### 

#### Перейти в папку, которую создавали ранее:

#### cd /Volumes/Data/docker/Cloudera\_Manager

Либо

#### cd C:\Data\docker\Cloudera\_Manager

Выполнить команду:

docker-compose up -d

Пример выполнения:

✔ cloudera 1 layers [⣿] 0B/0B Pulled

✔ 1d00652ce734 Pull complete

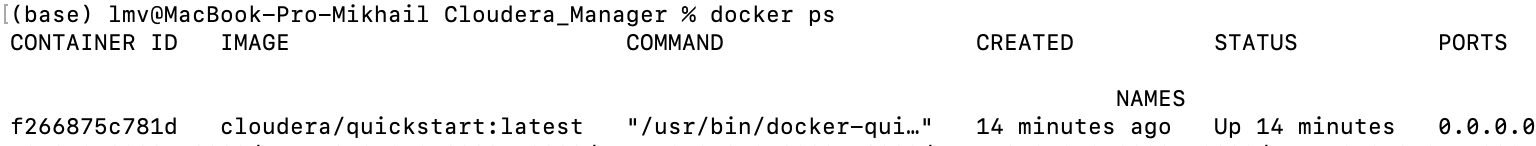
✔ Network cloudera\_manager\_default Created

✔ Container cloudera\_manager-cloudera-1 Started

Выполнить:

docker ps

Посмотреть CONTAINER ID



Выполнить:

docker attach f266875c781d

или

docker exec -it cloudera\_manager-cloudera-1 /bin/bash

где f266875c781d - наименование вашего контейнера

консоль перейдет внутрь контейнера, выполнить:

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

Результат выполнения:

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

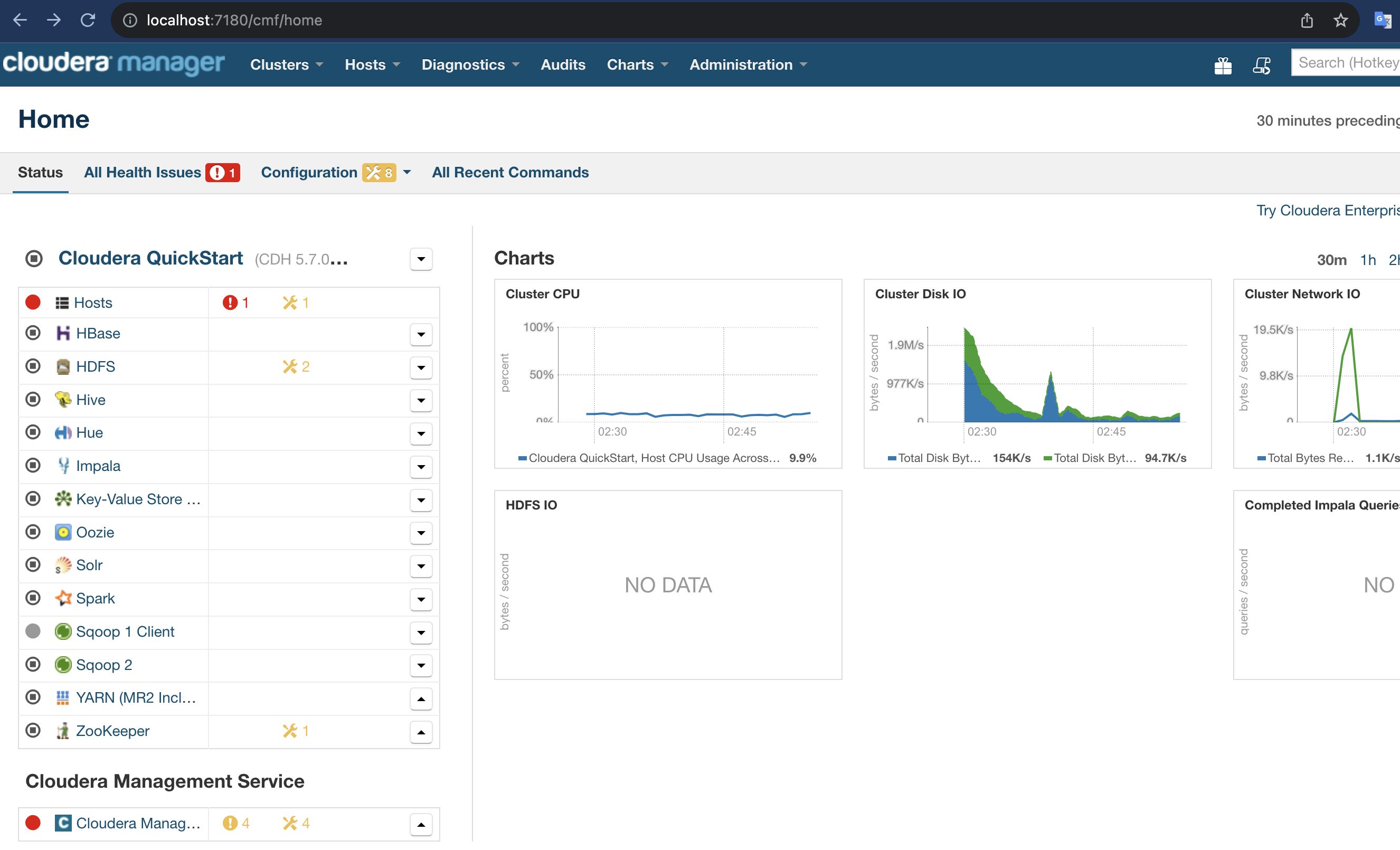
http://quickstart.cloudera:7180

Username: cloudera

Password: cloudera

Можно перейти по адресу:

<http://localhost:7180/>



Посмотреть IP адрес контейнера

docker inspect \

-f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' container\_name\_or\_id

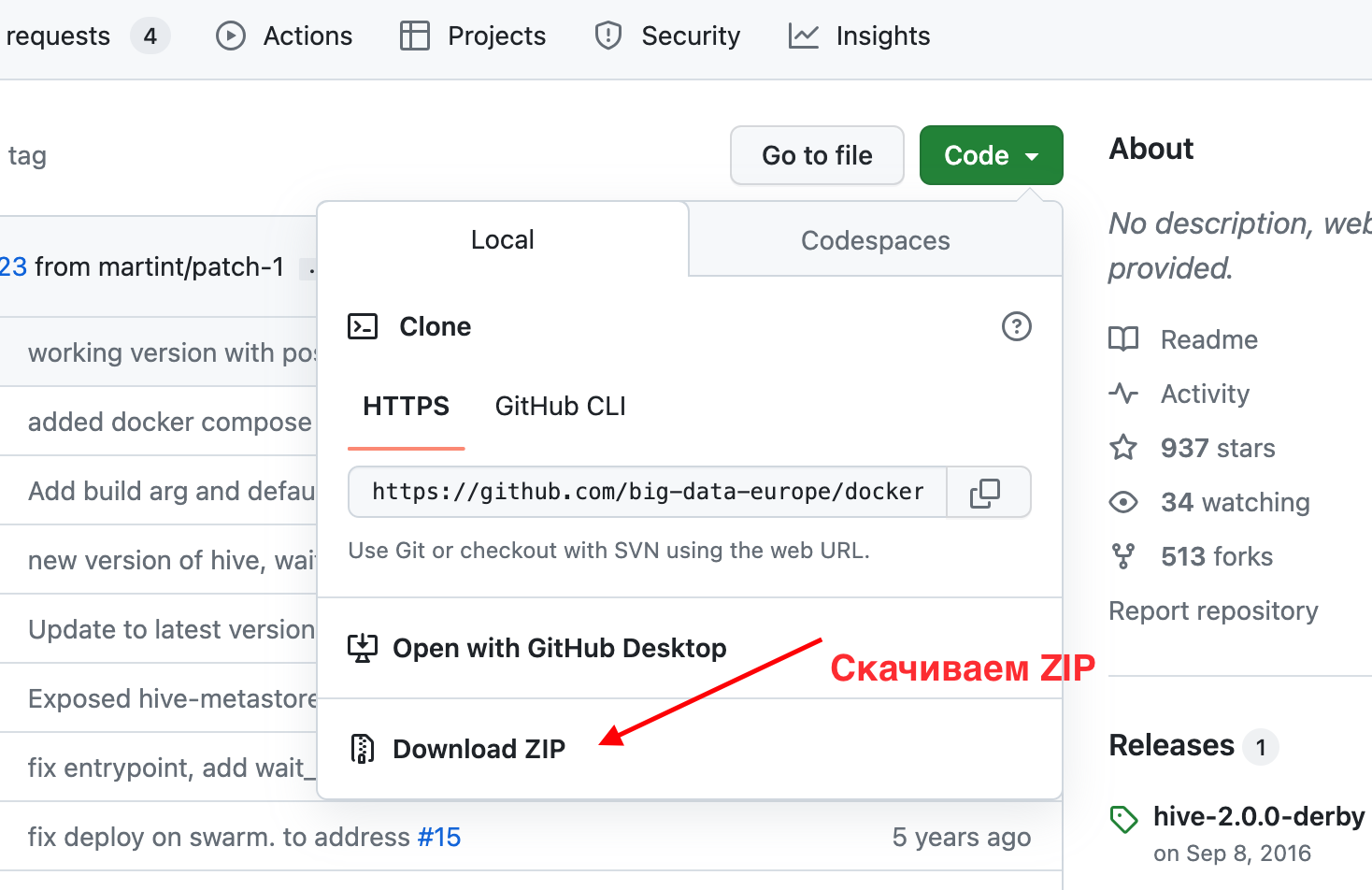
# **Установка Hive**

**Так же можно установить Hive, по желанию.**

1. скачал Docker файл отсюда:

https://github.com/big-data-europe/docker-hive

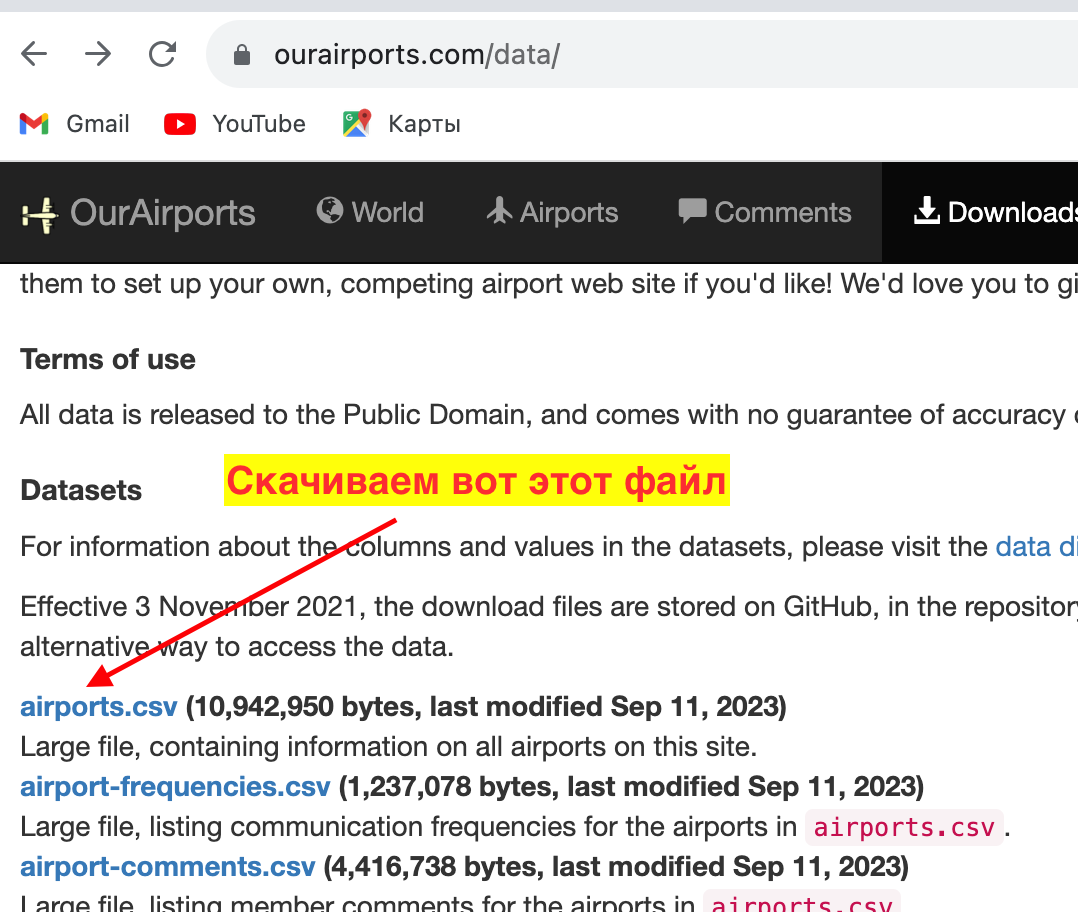
Скачиваем ZIP c гита:



2. А csv-файл отсюда:

https://ourairports.com/data/

Скачиваем файл [**airports.csv**](https://davidmegginson.github.io/ourairports-data/airports.csv)**:**



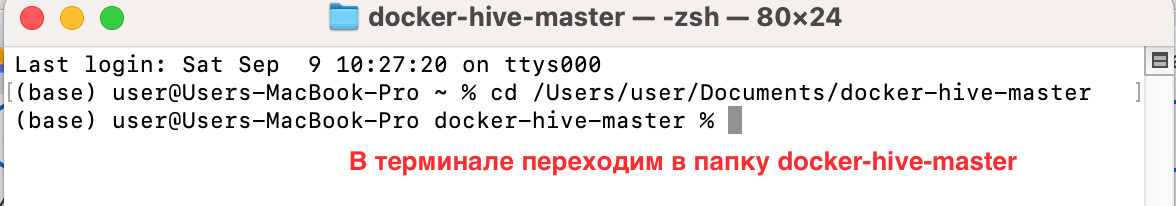
3. Развернул hive так:

C:\Users\docker-hive-master> docker-compose up -d

ZIP открываем и оставляем папку как есть, ничего здесь не запускаем:



4. Переходим в терминале в папку docker-hive-master:



5. Находясь в этой папке выполняем команду docker-compose up -d



6. Запустил контейнер:

PS C:\Users\docker-hive-master> docker-compose exec hive-server bash

7. Положил в него csv-файл в папку root

PS C:\Users> docker cp "C:\Users\airports.csv" docker-hive-master-hive-server-1:./root

6. в HDFS надо было создать директорию пользователя. Моё имя root

whoami

поэтому и папку назвал root

hdfs dfs -mkdir -p /user/root

Как проверить – команда hdfs dfs -ls после создания папки находит её и отрабатывать без ошибок

hdfs dfs -ls

**ИНСТРУКЦИЯ ПО DOCKER**

Команда

docker ps -a

Покажет все контейнеры, даже выключенные

Команда

docker rm uy6hy6ug

Удалит нужный контейнер (имя контейнера выводится в первой команде)

# **Пример создания и настройки своего контейнера**

1. Установить Docker

2. Скачать необходимый образ (зависит от версии CPU)

\* docker pull floydhub/tensorflow:1.9.0-py3\_aws.35

3. Запустить контейнер

\* docker run -it -p 9999:9999 floydhub/tensorflow:1.9.0-py3\_aws.35 /bin/bash

4. Настроить контейнер

\* apt update && apt upgrade -y

\* apt install curl wget git mc htop -y

\* cd /tmp

\* git clone https://github.com/matterport/Mask\_RCNN.git

\* cd Mask\_RCNN

\* pip3 install --upgrade pip

\* which jupyter

\* pip3 install -r requirements.txt

\* pip3 install --upgrade Cython pycocotools

\* python -m pip uninstall keras --yes

\* python -m pip uninstall tensorflow --yes

\* pip install tensorflow==1.5.0

\* pip install keras==2.0.8

\* python3 setup.py install

5. Проверить правильность версий

\* python3 -c "import tensorflow as tf; print(tf.\_\_version\_\_);import keras; print(keras.\_\_version\_\_)"

6. Сохранить изменения контейнера в образ

\* docker ps

Посмотреть ID контейнера, вписать ниже:

\* docker stop db55669ce797

\* docker commit db55669ce797 floydhub/tensorflow:1.9.0-py3\_aws.35\_4

\* docker rm db55669ce797

7. Запустить контейнер с новыми настройками

\* docker run -it -v /Volumes/Data/docker/:/mounted -p 9999:9999 -p 6006:6006 floydhub/tensorflow:1.9.0-py3\_aws.35\_4 /bin/bash -c "/usr/local/bin/jupyter notebook --notebook-dir=/tmp --ip='\*' --port=9999 --no-browser --allow-root"

8. Подключиться к ноутбуку

\* http://localhost:9999/?token=1437be79b3001ec63ff6e6bdf66b88a534e587456501b335

9. Подключиться к контейнеру

\* docker ps

\* docker exec -it 35b81eab0302 bash

# **Домашнее задание 1**

**Выполнить “hdfs dfs -ls” и “hdfs dfs -ls /” и сказать, в чём разница между этими командами.**

Команда

**hdfs dfs -ls**

- смотрим содержимое «домашней (контейнера) директории»

Команда

**hdfs dfs -ls /**

- а так мы заглянули в корень HDFS-системы и получили список папок, которые почти в случайном порядке расположены во всех Нодах внутри HDFS

# **Домашнее задание 2**

Литература:

<https://medium.com/@artem_gogin/how-exactly-hadoop-stores-the-data-23da0679d173>

<https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html>

<https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

https://hadoop.apache.org/docs/r2.6.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/DistCp.html

https://www.oreilly.com/library/view/hadoop-with-python/9781492048435/ch01.html

Задания:

1. Опробовать консольные утилиты для работы с кластером

* Создать/скопировать/удалить папку
* Положить в HDFS любой файл
* Скопировать/удалить этот файл
* Просмотреть размер любой папки
* Посмотреть как файл хранится на файловой системе (см. команду fsck)
* \*Установить нестандартный фактор репликации (см. команду setrep)

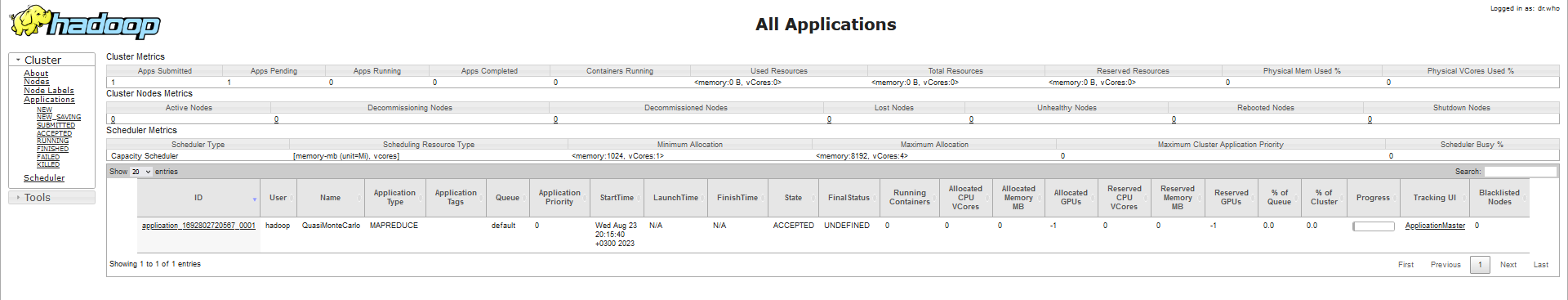
\*2. Опробовать доступ для работы с кластером

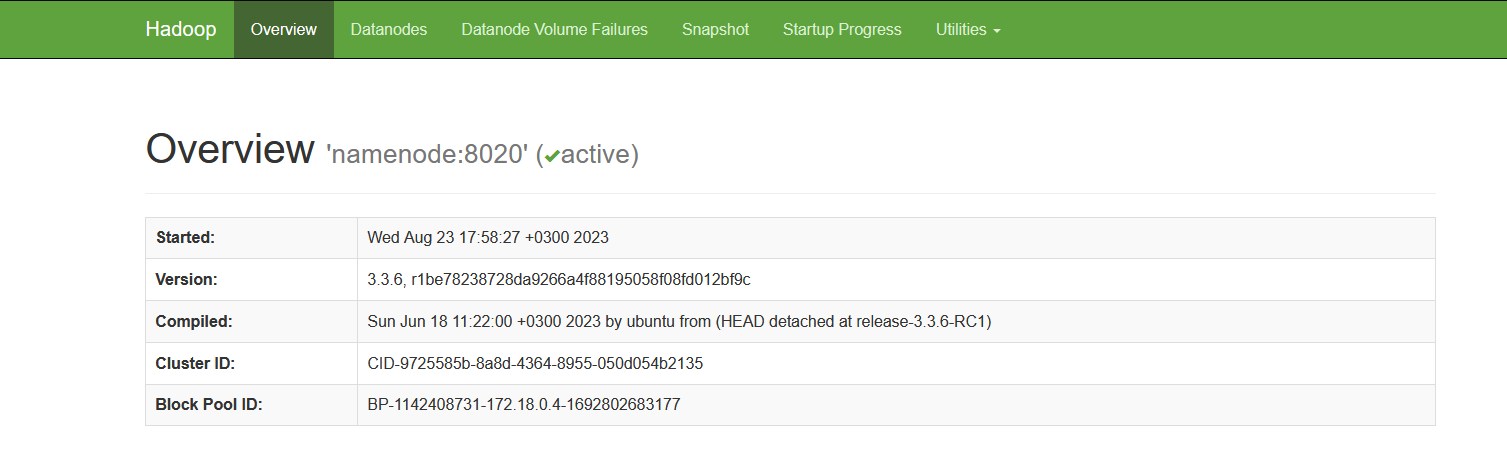
* Используя утилиту CURL
* Используя python3
* Используя libhdfs3
* Используя snakebite

1. Запуск
2. **docker-compose up -d**









1. Создание папки Lection2

hdfs dfs -mkdir -p /user/root

hdfs dfs -ls /

hdfs dfs -mkdir /lection2

hdfs dfs -ls /

1. Создаем и удаляем папку test

hdfs dfs -mkdir -p /user/test

hdfs dfs -ls /user

hdfs dfs -rm -r /user/test

hdfs dfs -ls /user

1. Создаем файл test.txt

hdfs dfs -touchz /user/lesson2/test.txt

hdfs dfs -ls /user/lesson2

1. Копируем файл HomeTask2 с локального ресурса

(base) root@MacBook-Pro-Mikhail hadoop #

docker cp /Volumes/Data/docker/hadoop/y\_test.csv hadoop-namenode-1:/tmp

**#docker cp /Volumes/Data/data4/docker2/hdp/docker/IMDB.csv namenode:/tmp**

bash-4.2$

**hdfs dfs -copyFromLocal /tmp/IMDB.csv /user/root/IMDB.csv**

hdfs dfs -copyFromLocal /tmp/y\_test.csv /user/lesson2/copy.csv

hdfs dfs -ls /user/lesson2

1. Уточняем размер папки

hdfs dfs -du /user

1. Удаляем файл copy.docx

hdfs dfs -rm -r /user/lesson2/copy.csv

hdfs dfs -ls /user/lesson2

hdfs dfs -du /user

1. Проверка файловой системы

hdfs fsck /user/lesson2

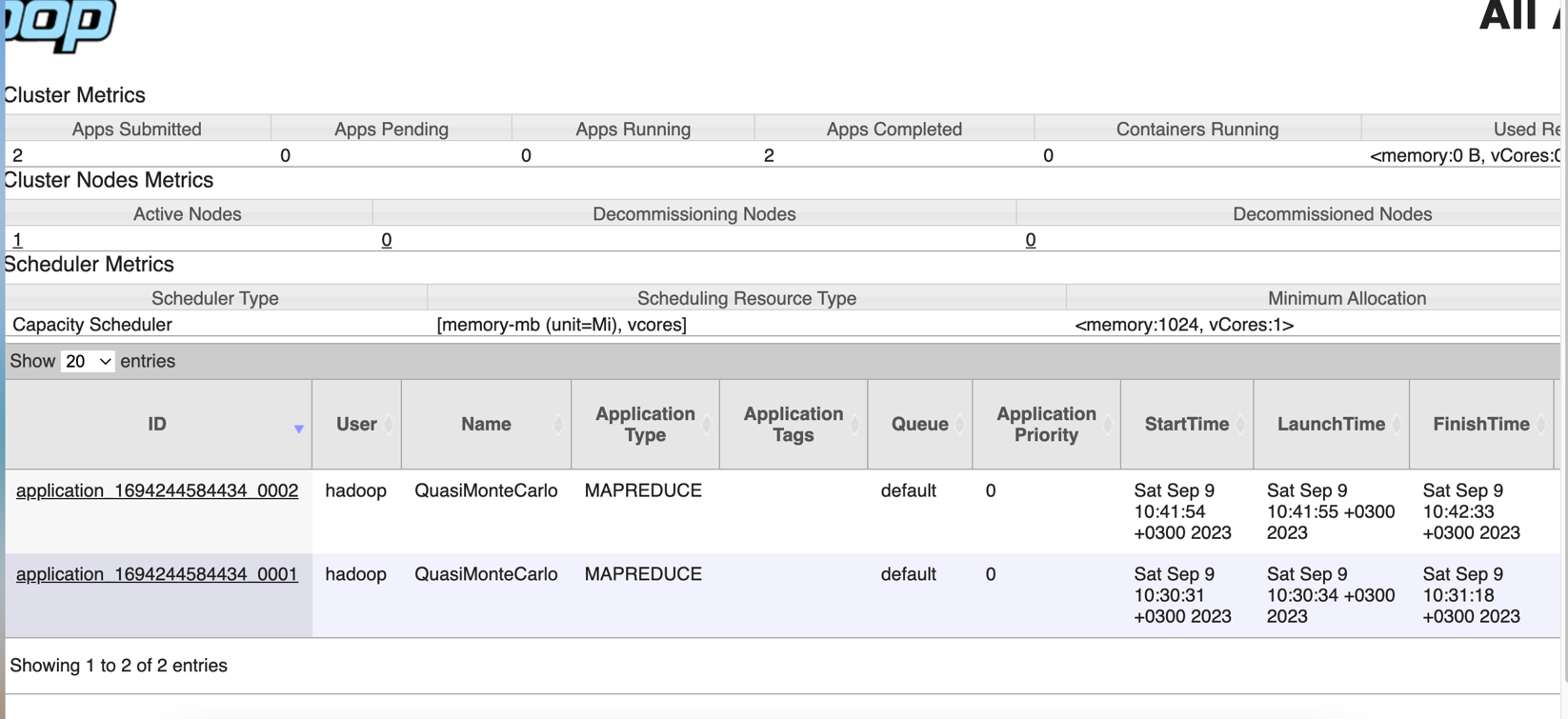
10. Скопировать файл из HDFS на локальный сервер

hdfs dfs -copyToLocal /user/root/copy.csv

# **Домашнее задание 3 part1**

Выполнить любую задачу включенную в этот JAR

Найти свою задачи в интерфейсе



**pyspark cassandra docker**

https://github.com/anguenot/pyspark-cassandra

https://hub.docker.com/r/dawidnowak/spark-cassandra

https://github.com/TurbineAnalytics/docker-spark-cassandra

**docker exec -it hive-server beeline -u 'jdbc:hive2://localhost:10000/'**