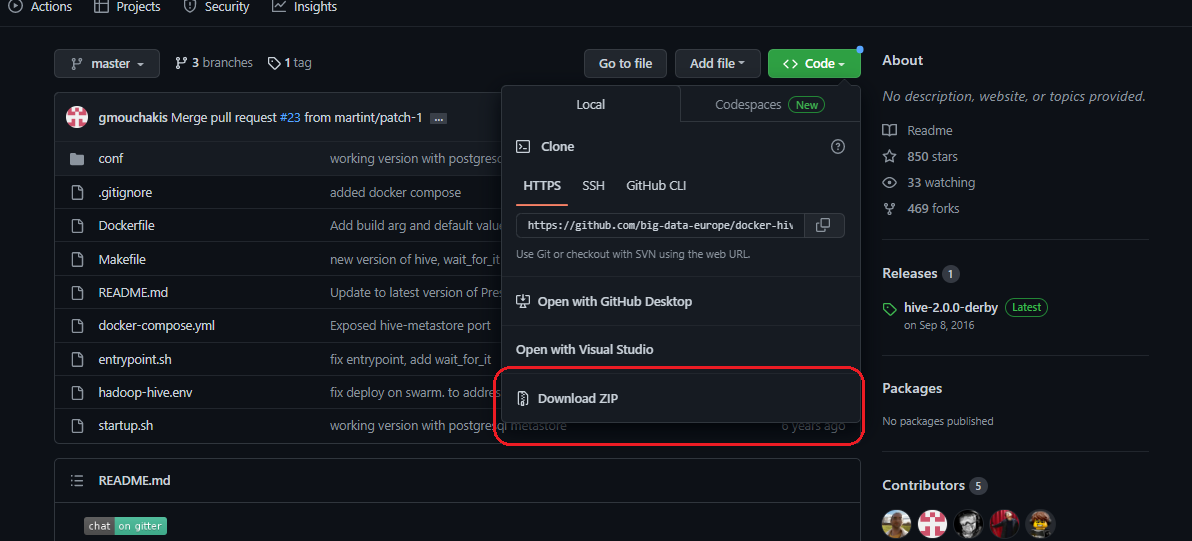
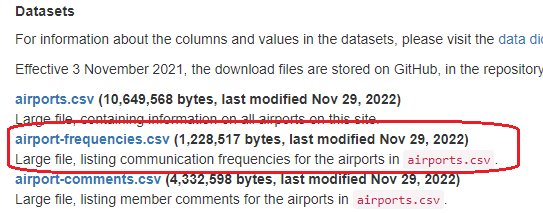
2) Сам докер качаем и ставим себе с его родного сайта: <https://www.docker.com/products/docker-desktop/>

1) Качаем сборку с Hive отсюда: <https://github.com/big-data-europe/docker-hive>



её нужно будет распаковать из архива и куда-то там себе положить

2) А csv-файл airports я погуглил и нашёл здесь: <https://ourairports.com/data/>



Я в нём даже изменений никаких не увидел, значит нам подойдёт

3) После установки докера сборку нужно развернуть. Запускаем саму прогу, она должна фоном работать. А далее делаем так: в терминале идём в папку, в которой разархивирована сборки и пишем команду

docker-compose up -d

всё немного покрутится-повертится и станет норм

4) в развернувшийся контейнер нужно будет положить файл airports.csv. Сделать это можно тоже через терминал, синтаксис такой: **команда «что» имя\_контейнера:«куда»** пример:

docker cp "C:\Users\Mysha\Desktop\путь\до\файла\airports.csv" docker-hive-master-hive-server-1:./root

и файл ляжет в контейнер в папку root

5) теперь есть смысл в контейнер зайти. Опять переходим в папку, в которой разархивирована сборка и пишем команду:

docker-compose exec hive-server bash

это мы как бы просим в hive-server запустить bash

6) готово, мы в контейнере. Здесь можно поиграться по всякому, побегать по папкам, насоздавать своих файлов. Помним, что в root у нас лежит файл с аэропортами, плюс на GitHub, откуда мы качали сборку, тоже написали пару примеров работы, да и вообще в этой сборке есть даже отдельная директория со всякими учебными файлами.

7) Теперь про HDFS. С HDFS сразу нормально работать не получится, так как там нет пользовательской директории, которую он при каждом запросе будет искать и не находить, так что её там надо отдельно создать. Сначала проверим как нас зовут

whoami

Это имя и используем для HDFS. Пишем команду для создания папки

hdfs dfs -mkdir /user/root

папка готова

8) перабрасываем в HDFS файл с аэропортами. Сначала создадим для неё папку

hdfs dfs -mkdir lalala

а потом бросаем в эту папку файл. Делается это командой -put

hdfs dfs -put /root/airports.csv lalala/

9) всё, файл с аэропортами в HDFS, даём ему права на исполнение

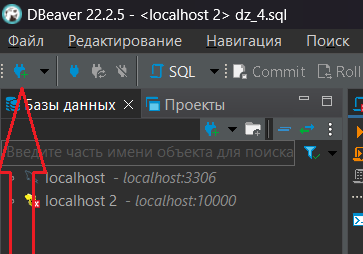
hdfs dfs -chmod 777 lalala/airports.csv

10) и можем подключаться к Hive. В этой сборке с Hive можно работать прям в терминале, достаточно запустить команду:

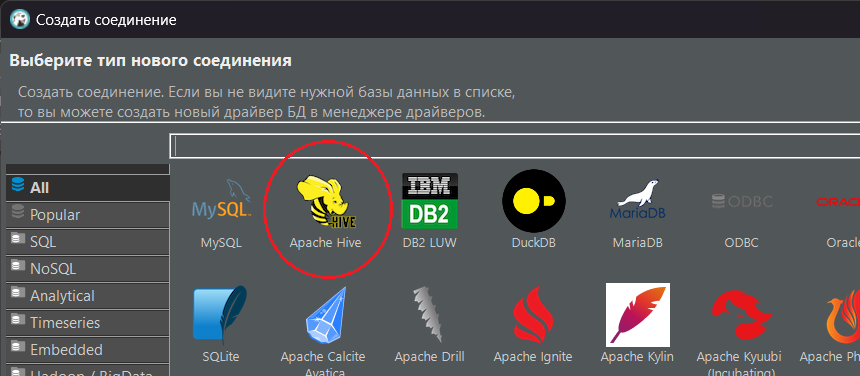
/opt/hive/bin/beeline -u jdbc:hive2://localhost:10000

Но это не тру, так как Hive всё же не MySQL и широкую таблицу в терминале прилично вывести не получится, так как он не поддерживает многие плюшки, что есть у MySQL. Так что лучше сразу настроить себе подключение в dBeaver и работать через него. Делается это проще простого:

- идём в dBeaver, давим на розетку



- в списке выбираем Hive



- и тут даже париться не придётся, так как если у нас контейнер запущен, то для соединения с ним достаточно указать Хост: localhost и Порт: 10000

Можно сделать тест соединения, но вообще проблем быть не должно. Тыкаем на «Готово» и работаем 😊 