# Технология хранения и архитектура больших данных

*П. В. Купцов. Факультет КНиИТ СГУ, кафедра ИиП*

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

## Предварительная информация

Все задания в этом курсе выполняются с использованием контейнеров docker, конфигурируемых и запускаемых при помощи docker compose. Убедитесь что Docker установлен.

Для этого в Windows или MacOS нужно скачать и установить Docker Desktop https://docs.docker.com/get-started/get-docker/.

Далее, для работы с контейнерами Docker в Windows или MacOS потребуется запускать приложение  Docker Desktop.

Под Linux Docker можно установить из репозитория. При этому будет установлен и запущен сервис, который обеспечивает работу Docker. В дальнейшем можно просто работать с контейнерами не запуская предварительно никаких дополнительных приложений.

*Справочная информация: некоторые команды управления контейнерами, выполняются в консоли.*

Запустить контейнер в фоновом режиме: в консоли перейти в папку с файлом «docker-compose.yaml» и выполнить

- `docker compose up -d`

Получить список запущенных контейнеров:

- `docker ps`

Остановить контейнеры: находясь в консоли в той же папке, что и при запуске, выполнить

- `docker compose down`

Принудительное завершение контейнера: выполнить команду ниже, вместо <NAME> ввести имя, которое показывает команда docker ps в последнем столбце.

- `docker kill <NAME>`

Выполнить команду в запущенном контейнере (в примере в контейнере mysql_db запускается оболочка bash и мы получаем возможность работать с файловой системой контейнера):

- `docker compose exec mysql_db bash`

Если перед запуском контейнеров выполняется установка дополнительных пакетов при помощи опции build и Dockerfile, то при необходимости пересобрать контейнеры (например, при изменении Dockerfile) можно следующей командой:

- `docker compose build --no-cache`

Логи, которые контейнер пишет в консоль можно посмотреть при помощи следующей команды. Помогает, например, когда контейнер падает.

- `docker logs <NAME>`

Очистка неиспользуемых ресурсов контейнеров и удаление не используемых в данный момент анонимных томов:

- `docker system prune --volumes`

Удаление всех томов, в том числе именованных. Все сделанные в контейнерах изменения будут удалены, за исключением тех, которые сохранялись в подмонтированных каталогах.

- `docker volume prune --all`

Удаление всех контейнеров из локального репозитория (после этого при запуске контейнеров будет автоматически запущено их скачивание):

- `docker system prune --all`

Удаление именованных томов, созданных контейнерами. Если удалить том, то все изменения, сделанные при работе с контейнером буду потеряны (если данные контейнера сохранялись в этом томе).

- `docker volume rm <NAME>`

Внимание! Если при попытке запуска контейнера появляется ошибка «Error response from daemon: failed to set up container networking» то это значит что на локальном компьютере занять порт, на который должен отображаться порт сервера, запускаемого в контейнере. В этой ситуации нужно сначала выполнить полную выгрузку упавшего контейнера docker compose down, а затем поменять настройку порта локального компьютера (хоста) в разделе ports файла docker-compose.yaml.

# Задание 1. SQL

Скачайте и установите DBeaver Community Edition (бесплатную версию) с сайта https://dbeaver.io/.

Скачайте и распакуйте архив bigd_course_mysql_jupyter.zip в своей рабочей папке.

Откройте в простом текстовом редакторе, типа Блокнота, конфигурационный файл docker-compose.yaml. Изучите его структуру и содержание. В этом Вам поможет файл docker-compose-template.yaml и документация по docker compose.

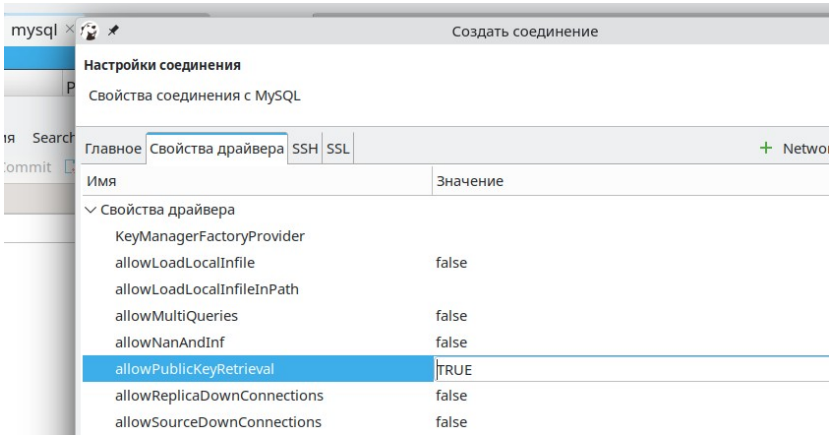
Файл docker-compose.yaml описывает запуск двух контейнеров: сервер базы данных mysql и сервер Jupyter, которые связаны друг с другом сетью.

Откройте консоль в папке bigd_course_mysql_jupyter и запустите контейнеры при помощи команды docker compose up -d.

При помощи консольной команды docker compose ps проверьте, что контейнеры корректно работают. Обратите внимание: после запуска контейнера ему потребуется некоторое время, чтобы восстановить базу из дампа. Поэтому подождите несколько минут перед попыткой соединиться с базой.

Запустите DBeaver и создайте новое соединение (меню База данных, Новое соединение). Выберите тип базы MySQL и укажите параметры подключения, которые можно найти в конфигурационном файле docker-compose.yaml. Укажите адрес хоста localhost.

В окне конфигурации соединения нажмите кнопку «Тест соединения». Вероятно, будет выдано сообщение об ошибке «Public key retrival is not allowed». В таком случае перейдите на вкладку «Свойства драйвера» и задайте «allowPublicKeyRetrival=TRUE»




В папке bigd_course_mysql_jupyter найдите папку work. В ней есть файл MySQL.sql. Откройте его в DBeaver: меню «Файл», «Найти файл по имени».

На панели инструментов DBeaver найдите выпадающий список «Текущее соединение (Ctrl+9)» и при необходимости выберите там базу «sakila».

Далее работа будет идти в редакторе SQL программы DBeaver. Для удобства рекомендуется включить перенос строк: меню «Редактирование», «Форматирование», «Вкл/Выкл перенос строк».

Изучите структуру базы sakila, которую можно найти в файле sakila-en.a4.pdf, находящимся в папке bigd_course_mysql_jupyter.

Решите задачи, которые Вы найдёте в открытом sql-файле. Вписывайте запросы и полученные ответы сразу после задач.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 2. Python + SQL

Запустите контейнеры из папки bigd_course_mysql_jupyter.

Перейдите в браузере по адресу 127.0.0.1:8888. Откроется страничка Jupyter.

Откройте в Jupyter файл work/Python+SQL.ipynb и выполните задания в нём.

По окончании работы сохраните файл, закройте Jupyter и после этого остановите контейнеры консольной командой docker compose down.

# Задание 3. Pandas

Запустите контейнеры из папки bigd_course_mysql_jupyter.

Перейдите в браузере по адресу 127.0.0.1:8888. Откроется страничка Jupyter.

Откройте в Jupyter файл work/Pandas.ipynb и выполните задания в нём. В качестве дополнительного справочного материла можно использовать файл Pandas_Cheat_Sheet.pdf, который можно найти в папке bigd_course_mysql_jupyter.

По окончании работы сохраните файл, закройте Jupyter и после этого остановите контейнеры консольной командой docker compose down.

# Задание 4. Pandas+SQL

Запустите контейнеры из папки bigd_course_mysql_jupyter.

Перейдите в браузере по адресу 127.0.0.1:8888. Откроется страничка Jupyter.

Откройте в Jupyter файл work/Pandas+SQL.ipynb и выполните задания в нём.

По окончании работы сохраните файл, закройте Jupyter и после этого остановите контейнеры консольной командой docker compose down.

# Задание 5. ClickHouse

Скачайте и распакуйте архив bigd_course_clickhouse.zip в своей рабочей папке.

Откройте в простом текстовом редакторе, типа Блокнота, конфигурационный файл docker-compose.yaml, который находится в папке bigd_course_clickhouse. Изучите его структуру и содержание.

Запустите контейнер с сервером ClickHouse из папки bigd_course_clickhouse.

Запустите DBeaver и настройте соединение с запущенным сервером базы, Параметры подключения возьмите из конфигурационного файла docker-compose.yaml. Укажите адрес хоста localhost.

База с которой Вы соединитесь содержит единственную таблицу trips. Структура этой таблицы описана в лекциях.

В папке bigd_course_clickhouse/work найдите файл ClickHouseSQL.sql. Откройте его в DBeaver и выполните находящиеся в нём задания. Вписывайте запросы и полученные ответы сразу после задач.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 6. MongoDB

Скачайте и распакуйте архив bigd_course_mongodb.zip в своей рабочей папке.

Найдите в папке bigd_course_mongodb/work задания в файле MongoDB.pdf

Запустите контейнер с сервером MongoDB из папки bigd_course_mongodb.

Запустите оболочку командной строки mongosh внутри контейнера сервера базы данных. Для этого выполните в консоли команду

```
docker exec -it bigd_course_mongodb-mongo-1 mongosh
```

Работа в консоли базы, выполните задания из файла MongoDB.pdf.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 7. HDFS

Скачайте и распакуйте архив bigd_course_hadoop.zip в своей рабочей папке.

Откройте в ней файл с заданиями «1 - Работа с HDFS.pdf» и выполните их.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 8. MapReduce

После выполнения предыдущего задания у Вас в рабочей папке должна быть папка bigd_course_hadoop.

Откройте в ней файл с заданиями «2 - Работа с MapReduce.pdf» и выполните их.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 9. Hive

После выполнения предыдущего задания у Вас в рабочей папке должна быть папка bigd_course_hadoop.

Откройте в ней файл с заданиями «3 - Работа с Hive.pdf» и выполните их.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.

# Задание 10. PySparkLocal

Это задание выполняется на локальном Spark, без развёртывания кластера. Для этого нужно будет запустить единственный контейнер, в котором установлен локальный Spark и Jupyter для исполнения кода PySpark.

Скачайте и распакуйте в своей рабочей папке архив bigd_course_pyspark_local.zip.

Перейдите в браузере по адресу 127.0.0.1:8888. Откроется страничка Jupyter.

Откройте в Jupyter файл work/PySparkLocal.ipynb и выполните задания в нём.

По окончании работы сохраните файл, закройте Jupyter и после этого остановите контейнеры консольной командой docker compose down.

# Задание 11. SparkRDD

Скачайте и распакуйте архив bigd_course_spark.zip в своей рабочей папке.

Откройте в ней файл с заданиями «1 - SparkRDD.pdf» и выполните их.

По окончании работы не забудьте остановить контейнеры консольной командой docker compose down.