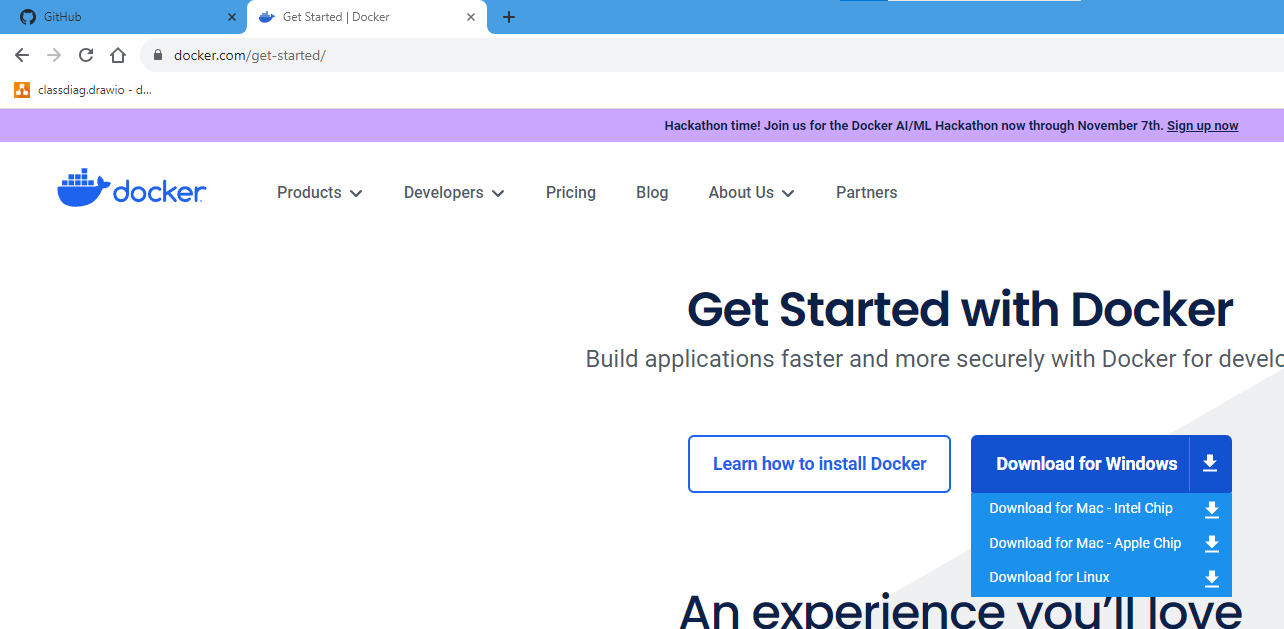
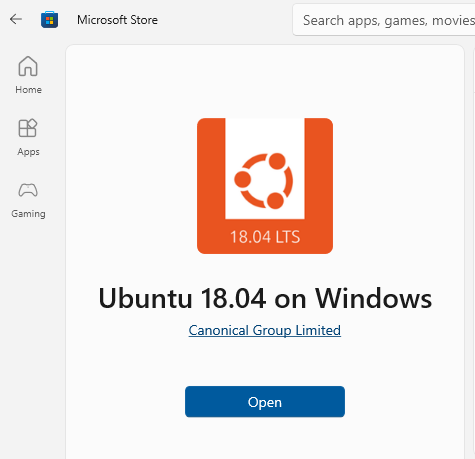
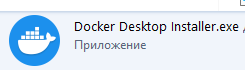
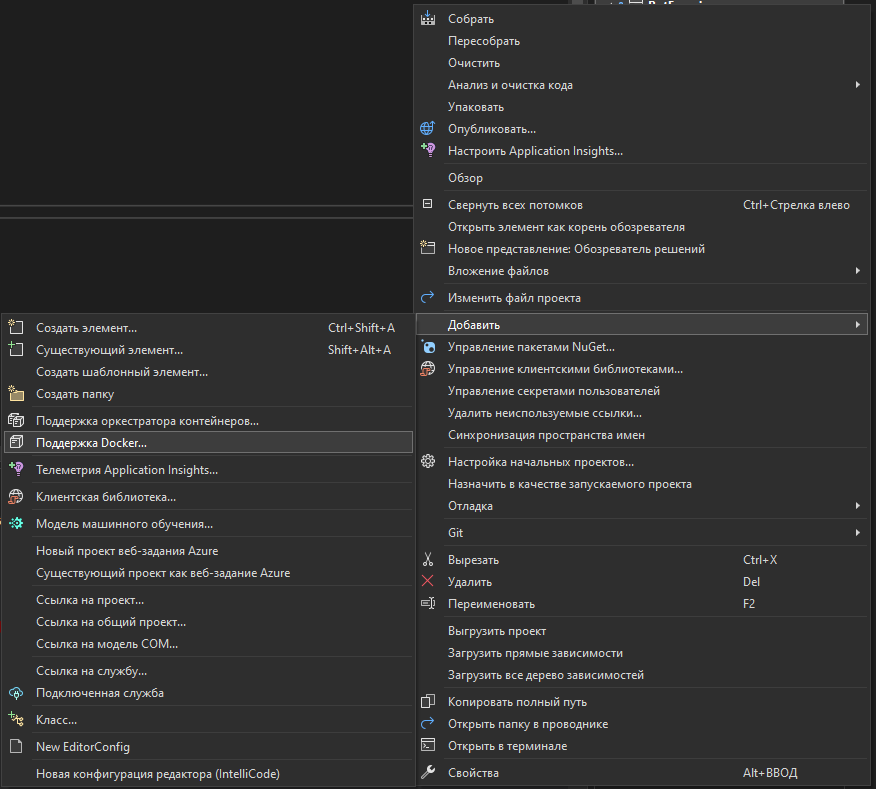
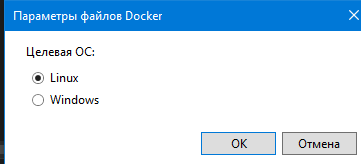
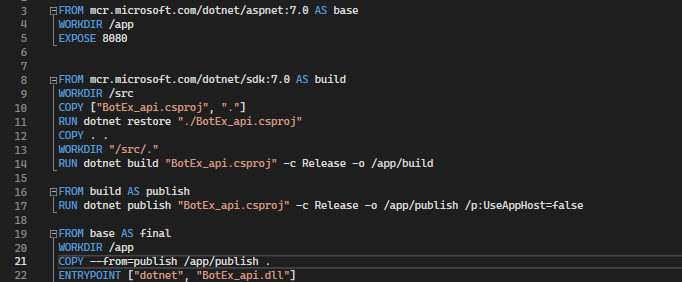
Оборачивание API и БД в Docker

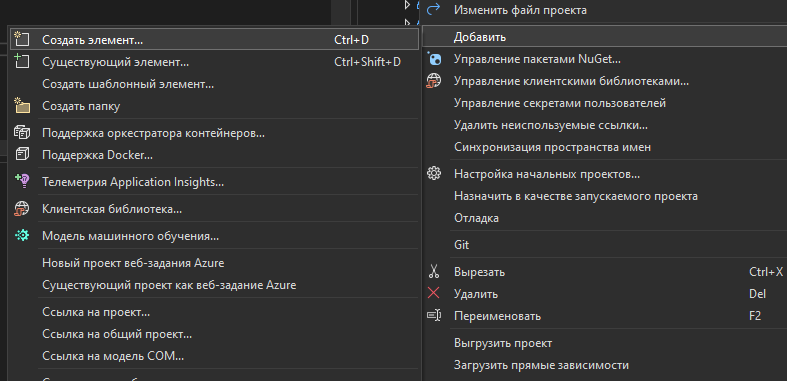
1. Перейти на сайт <https://www.docker.com/get-started/>
2. Скачать Docker

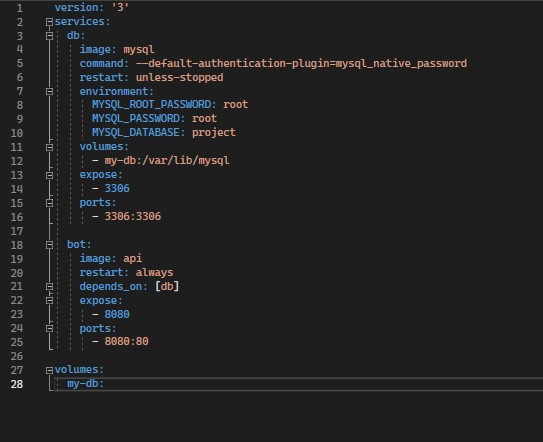
Перед установкой Docker’a, нужно установить WSL(Windows Subsystem for Linux), на котором у нас будет находиться наше API и БД, я выбрал этот: 

После установки WSL, устанавливаем Docker, запустив ранее скачанный файл

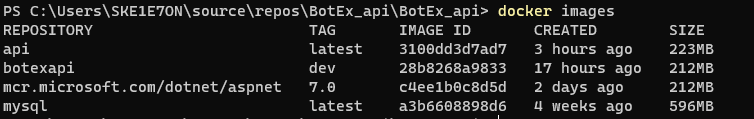
1. Заходим в проект с API, и создаем Dockerfile в корне проекта  

После этого у нас появляется сразу заполненный Dockerfile, в котором нужно изменить EXPOSE на 8080: 

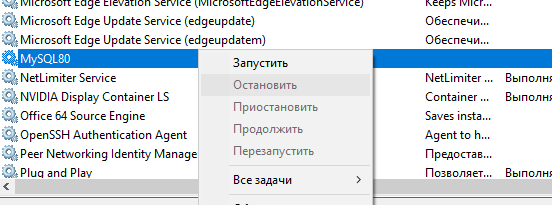
Далее в корне проекта создаем обычный элемент, называем его docker-compose.yml, он будет отвечать за БД

Заполнить его нужно примерно так: 

1. Открываем консоль в VS, прописываем туда следующее: , где “api” – это название image’a, a “Dockerfile” – название файла который был создан ранее

При успешном создании в консоли можно прописать «**docker images**», это выведет наши image’s созданные ранее, в моем случае меня интересует этот: 

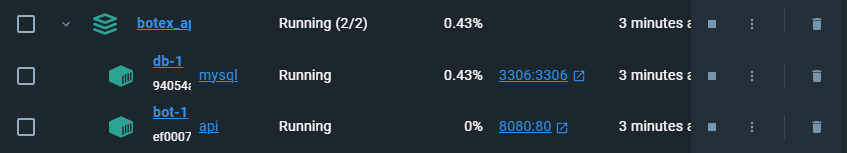
1. Перед запуском контейнера, нужно отключить mysql на ПК, в поисковой строке пишем «Службы», находим MySQL80, нажимаем ПКМ->Остановить



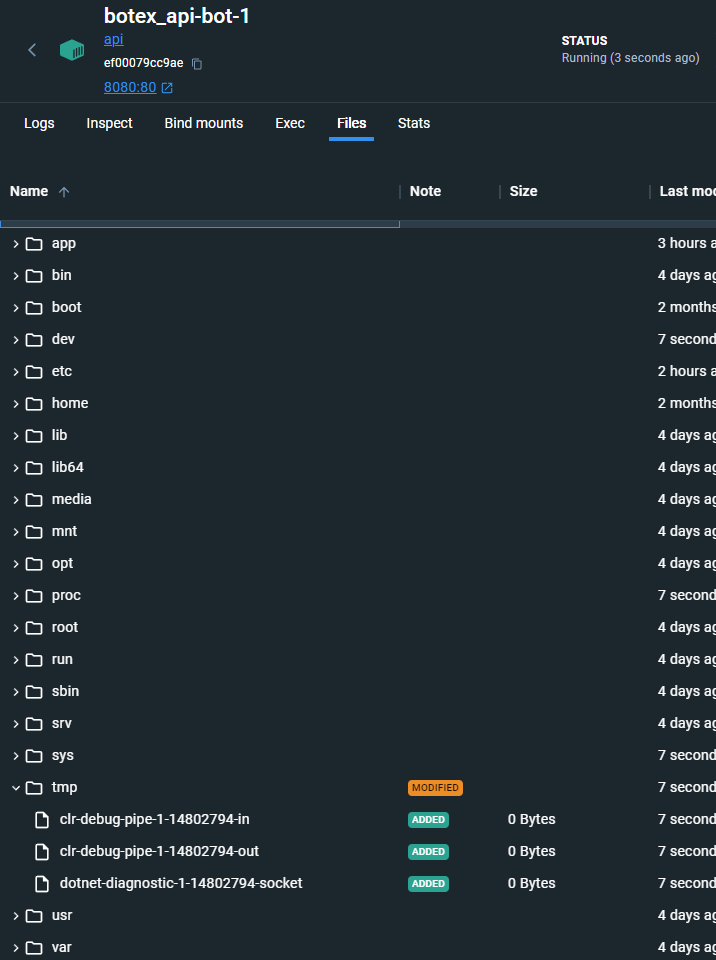
1. Для запуска нашего контейнера пишем в консоль следующее:



После этого в Docker’e будет так:



Далее, нужно перенести дамп БД(self-contained), в контейнер db-1



1. Переходим во вкладку «Exec», прописываем следующие команду:

«mysql -u root -p project< tmp/projectDataBase.sql», где project – название БД в файле docker-compose, а «tmp/projectDataBase.sql» путь к дампу

1. Для проверки пишем в ту же консоль «mysql -u root -p», вводим пароль, далее пишем команду «show tables», в следующей строке нужно вписать «;»