Docker x Apache p.2

**Установка Docker**

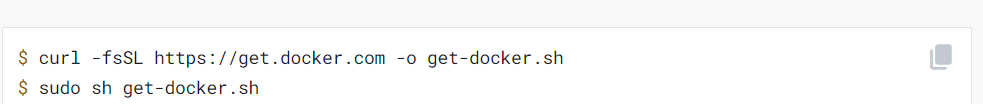
Первым делом установим curl:

**apt update**

**apt install curl -y**

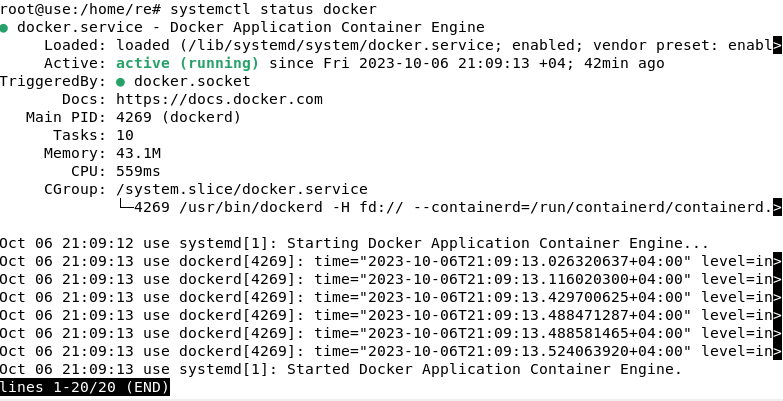
curl позволит нам выполнять http-запросы не выходя из командой строки

Установим Docker:



По адресу get.docker.com находится скрипт, который облегчит нам задачу и поможет установить его в 2 команды

Командой **systemctl status docker** проверяем его работу:



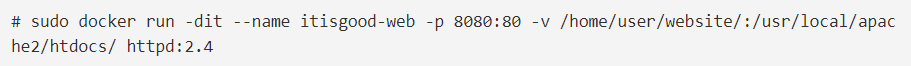
Настройка контейнера **Apache**

Мы будем использовать образ под названием httpd:2.4 из Docker Hub.

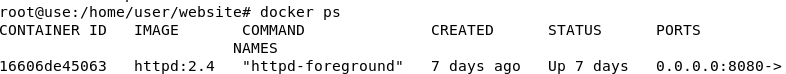
Наш план состоит в том, чтобы запросы, сделанные на наш публичный IP-адрес на порт 8080, перенаправлялись на порт 80 в контейнере.

Кроме того, вместо того, чтобы обслуживать контент из самого контейнера, мы будем обслуживать простую веб-страницу из **/home/user/website.**

Мы сделаем это путем отображения **/home/user/website/** на **/usr/local/apache2/htdocs/** на контейнере.



На этом этапе наш контейнер должен быть запущен:



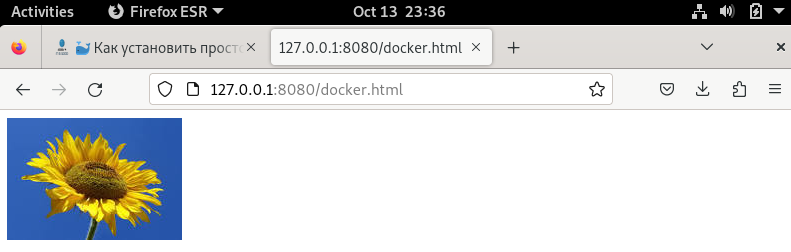
Теперь давайте создадим простую веб-страницу с именем docker.html в каталоге /home/user/website.

Затем откройте браузер на 127.0.0.1:8080/docker.html

Перед вами должна появиться страница, которую мы создали ранее.

Зайдите в файл веб-страницы и добавьте туда фотографию(любую из интернета):

<img src="ссылка на фотографию" />



Поднимаем **Apache Superset**

Устанавливаем докер-образ Superset

docker pull apache/superset



Командой **docker images** проверяем наличие контейнеров в системе

**Проверяем работу Superset**

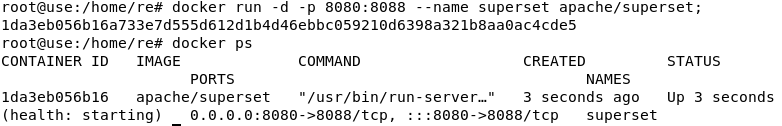
Контейнер надо запустить. Запущенный контейнер в дальнейшем можно будет остановить. И важно не путать это действие с, скажем так, выключением контейнера и с удалением образа.

Для начала сгенерируем секретный ключ, сделать это можно командой :

**openssl rand -base64 42**

****

Команда запуска docker-контейнера с Superset:

**docker run -d -p 8080:8088 -e "SUPERSET\_SECRET\_KEY=your\_secret\_key\_here" --name superset apache/superset;**

Флаг -d (detach) означает отключение контейнера от консоли. Флаг -p отвечает за проброс трафика между портами контейнера и указанными портами основной машины.

Командой **docker ps** проверяем работу контейнера

Хорошим инструментом для мониторинга сети является утилита netstat.

Установим ее командой **apt-get install net-tools**

Теперь проверим, запустился ли контейнер.

**netstat -tupln**

Эта команда покажет состояние портов. Описание флагов отразит суть отображаемой информации.

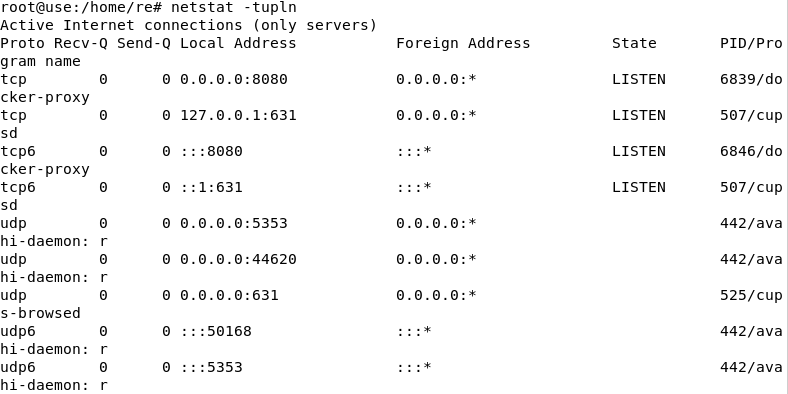
-t - (или --tcp) показать TCP порты;

-u - (или --udp) показать UDP порты;

-l - (или --listening) посмотреть только прослушиваемые порты;

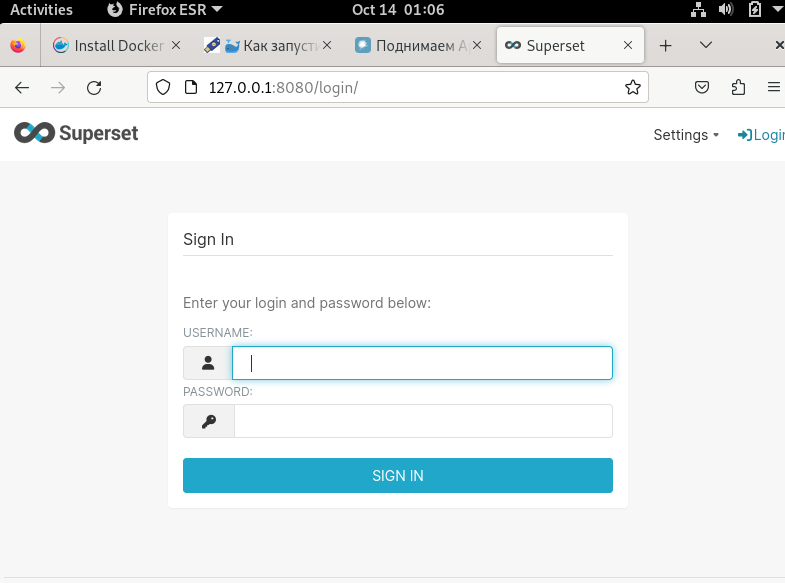
-p - (или --program) показать имя программы и ее PID;

-n - (или --numeric) показывать ip адреса в числовом виде.

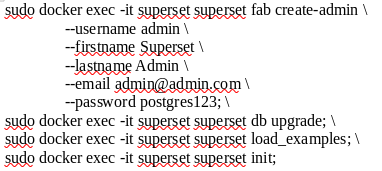


По итогу необходимо удостовериться, что порт **8080** занят процессом **docker-proxy** и прослушивается, а также в том, что присутствует активный контейнер с названием **apache/superset.**

Проверяем:



И дальше пройти не получится, ведь пользователей мы никаих не создавали, к тому же в самом Superset никаких данных нет. Чтобы это исправить, нам надо все это настроить и создать.

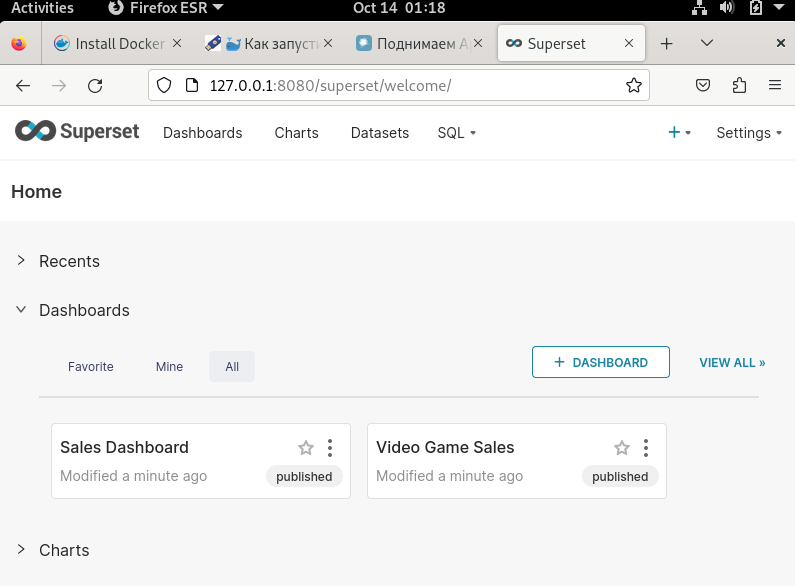


Для удобства исполнения заходим в **LibreOffice** и пишем команды там, после копируем и вставляем в терминал

Все команды собраны в одну так, чтобы для их исполнения можно было ограничиться однократным нажатием на Enter. Обратите внимание, что в команде указано в качестве пароля.

Команда docker exec позволяет нам передавать внутрь контейнера команды, как если бы мы заходили в виртуальную машину и вводили команды в консоль. Предназначение остальных команд, думаю, понятно из текста.

Заходим под админом и видим что то подобное:



**Устанавливаем Postgres и Pgadmin**

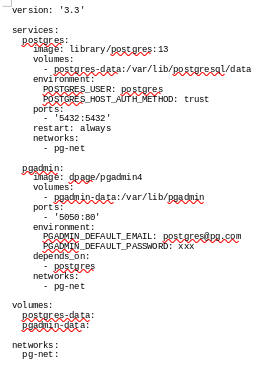
Posgtres - база данных. Сам по себе сервер с СУБД может управляться единственным образом путем подключения приложения к нему. И с одной стороны, мы подключимся нашим сервером Superset внутри docker-контейнера к Postgres, что еще надо то. Но настраивать и следить за БД так не получится. Для этого предусмотрен специальный сервер-интерфейс СУБД. Соответственно, под этот сервер также необходим docker-контейнер. Подробнее PgAdmin будет рассмотрен дальше.

И вот тут мы начинаем говорить о контейнерах, работающих в паре. Здесь отлично подойдет docker-compose. Для запуска композиций наиболее удобно применять так называемые docker-compose-файлы. В них прописываются все настройки запускаемых контейнеров, а дальше все включается одной командой. По сути, мы таким уже пользовались, когда создавали композицию **Grafana x node-exporter x Prometheus**.

Но файл мы скачали из интернета, а сейчас пишем вручную.

Заходим в LibreOffice

Содержание файла:

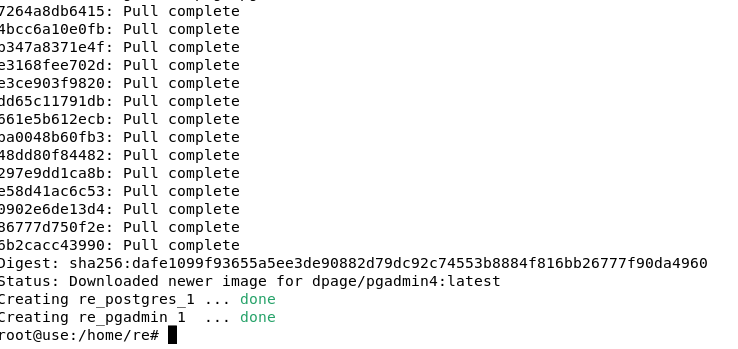


Этот текст копируем, а далее создаем файл docker-compose.yml, написав:

**nano docker-compose.yml**

Далее вставляем текст и выходим нажатием **Ctrl+X -> Y -> Enter**. После этого в директории с созданным файлом поднимаем сервера postgres и pgadmin с помощью команды:

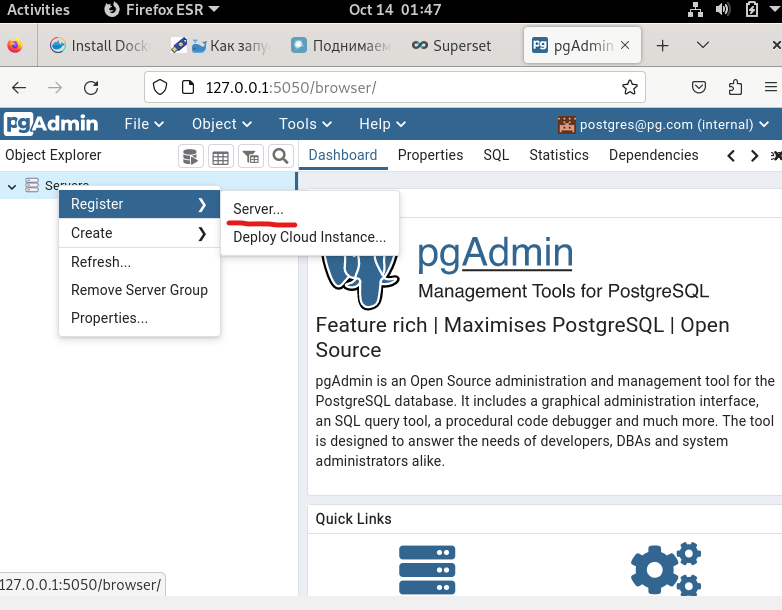
**docker-compose up -d**

****

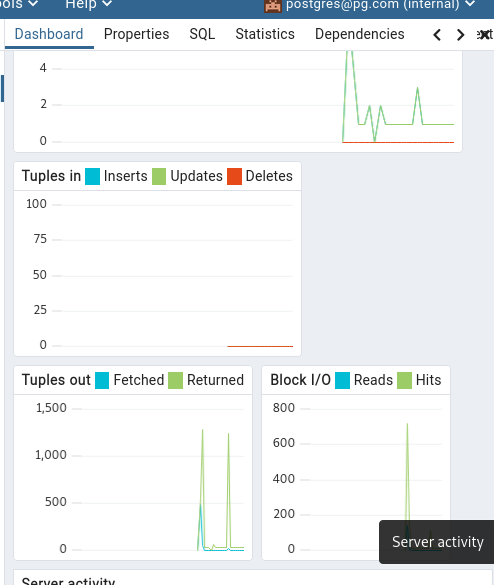
**Проверяем работу Postgres через PgAdmin**

Для начала заходим на порт 5050. На открывшейся странице вводим почту и пароль от pgadmin из yml-файла.

Отлично, мы в панели администратора. Регистрируем новый сервер (подключаемся к нему, иными словами). Нужная кнопка на картинке ниже.



Открыватеся список полей для заполнения. Обязательных два. Первое - name на вкладке General заполняем, как нравится. Исключательно косметическое действие в данный момент. Второй параметр важный - Host name/address на вкладке Connection, здесь надо вписать адрес сервера с СУБД. Вписываем postgres. Проверяем, что port совпадает с тем, что был указан в графе ports: в конфиг-файле docker-compose.yml (то есть 5432). Если да, то жмём Save и любуемся красивыми графиками.

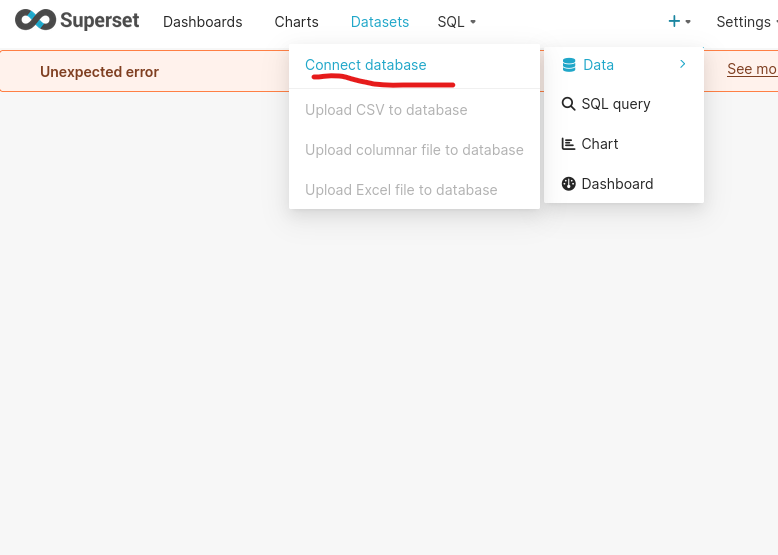


**Подключаем Postgres к Superset**

Отдельно оговорюсь, что созданная и подключенная в качестве примера к Superset БД слишком уж далека от характеристики "гибкая", так как в нее даже нельзя данные свои загрузить. Поэтому о ней мы забудем, а подключим лучше свой Postgres.

Итак, сделать надо следующее.

С помощью Connect database (локация кнопки на скриншоте) мы окрываем меню с настройками нового подключения. В появившемся окне выбираем PostgreSQL.

****

Далее вводим в HOST адрес 172.17.0.1, PORT - 5432, DATABASE NAME - как указали в конфиге, то есть postgres. Если у вас не появилось под вышеперчисенными полями никаких красных подписей - полдела сделано, осталась буквально пара мелочей.

Проверяем, что все есть - для этого захдим в Setting -> Databases и если видим строчку с нашей БД значит все готово, Superset с Postgres работают.