«Автоматизированная система идентификации клиентов и блокировки интернет-ботов»
«Humanometr»

Инструкция по установке экземпляра программного обеспечения,
предоставленного для проведения экспертной проверки**.**

Данные для распаковки архив-файла:
Пароль: Human0metr!
(где 0 – ноль)

Севастополь, 2022 г.

Настройки на стороне сервера

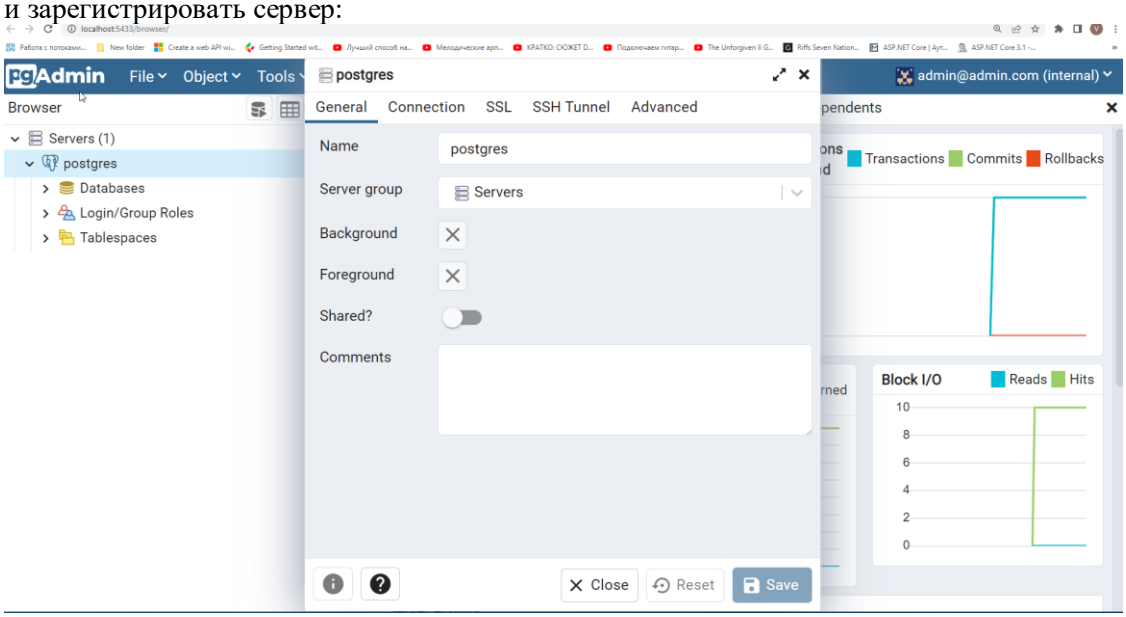
Предусловия: необходимо иметь установленный Docker.
Требования для установки Doker на ОС Windows: версия 10 64-бит +.
Доступ в интернет к репозиторию контейнеров Doker.

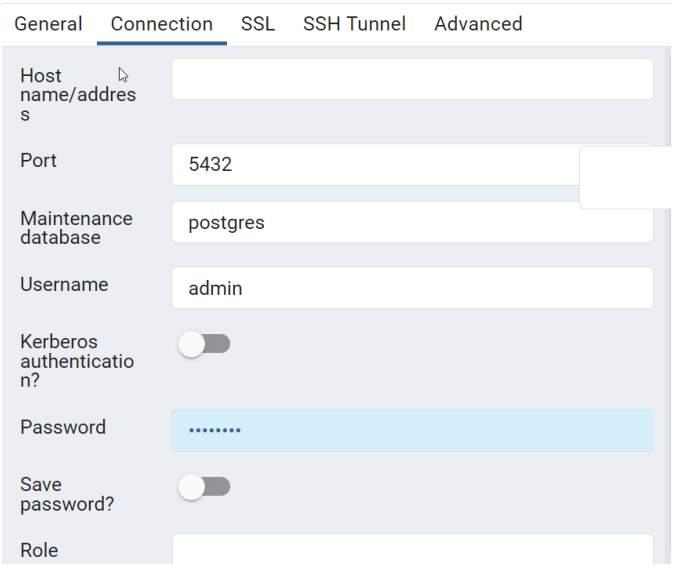
1. Создаем инфраструктуру.
   1.1. Заходим в папку Humanometr_infrastructure.
   1.2. Выполняем команду “docker-compose -f docker-compose.yml up” (либо файл infrastructureGenerator.bat)
   1.3. Выполняем команду “docker ps -a” в консоли, должны получить что-то похожее на

```
c6ab660254a2   edoburu/pgbouncer      "/entrypoint.sh /usr…"   7 minutes ago   Exited (1) 3 minutes ago   rucaptchafilerepo_pgbouncer_1
d050c4586d53   postgres               "docker-entrypoint.s…"   7 minutes ago   Exited (0) 3 minutes ago   rucaptchafilerepo_postgres_1
f01dd18b3349   rabbitmq:3-management  "docker-entrypoint.s…"   7 minutes ago   Exited (0) 3 minutes ago   rucaptchafilerepo_rabbitmq_1
1c494c6e2edc   dpage/pgadmin4         "/entrypoint.sh"         7 minutes ago   Exited (0) 3 minutes ago   rucaptchafilerepo_pg-admin_1
b3c831aebba4   redis                  "docker-entrypoint.s…"   7 minutes ago   Exited (0) 3 minutes ago   rucaptchafilerepo_redis_1
```

2. Создаем и запускаем приложение.
   2.1.Заходим в папку Humanometr-api
   2.2. Последовательно выполняем команды (либо apiGenerator.bat):
      2.2.1. docker load -i .\humanometrapi.tar
      2.2.2. docker compose -f docker-compose.yml -f docker-compose.override.yml up
      2.2.3. проверяем, что приложение запустилось. Если вы запускали локально, то приложение должно быть доступно по адресу “http://localhost:5001/client.html”, если на своем хостинге, то по тому адресу, который вы определите сами (в дальнейшем по тексту назовем его Humanometr-host)

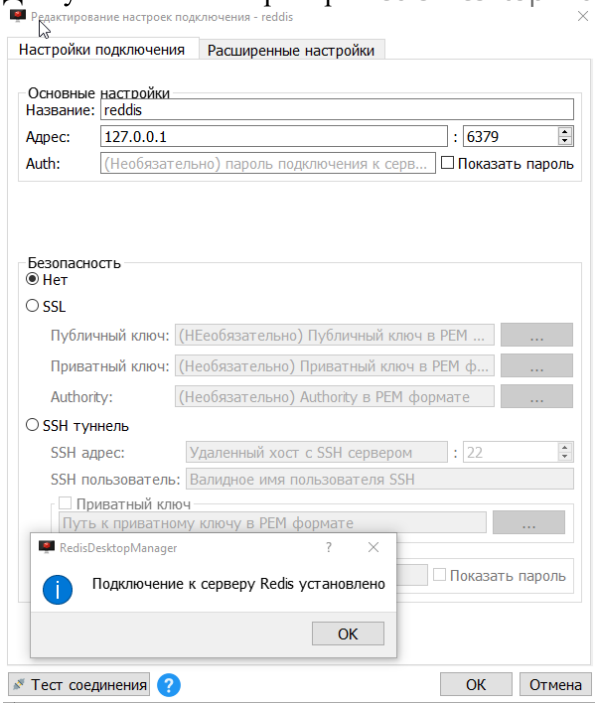
Для подключения к БД можно воспользоваться любым инструментом, умеющим работать с Postgres, например https://dbeaver.io/download/. Данные для подключения по умолчанию: .... В том числе в контейнерах по умолчанию присутствует PGAdmin по адресу
http://localhost:5433/login?next=%2F
login: admin@admin.com
password : password
и зарегистрировать сервер:





Имя хоста: postgres
Пароль: password

Доступ к Reddis на примере Redis Desktop Manager



Доступ к Rabbit
http://localhost:15672/
Username: guest
Password: guest

Добавление виджета Humanometr на веб-страницу

Добавление виджета Humanometr на веб-страницу производится аналогично установке популярных виджетов «Google recaptcha» и «hcaptcha».

Humanometr требует добавления двух небольших фрагментов кода на стороне клиента для отображения виджета с капчей на HTML-странице. Для этого ресурс javascript Humanometr необходимо включить где-либо на своей HTML-странице. <script> должен быть загружен через HTTPS и может быть размещен в любом месте на странице, а также должен иметь атрибут

id="humanometr-script"

<script id="humanometr-script" src="https://<Humanometr-host>/index.js" async defer></script>
(где Humanometr-host – известный адрес, по которому развернуто приложение Humanometr, см. п. 2.2.3 выше)

Также вы можете указать в ссылке скрипта параметр **onload** с именем функции, которую реализуете вы, и которая будет вызвана после инициализации виджета. В данную функцию будут переданы 2 параметра – secret и response, которые понадобятся далее. Используйте эту функцию, чтобы сохранить эти параметры так, чтобы вам потом удобно было отправить их в API при сабмите формы *(если не указать параметр **onload,** то в дальнейшем нужно будет указать атрибут **data-callback** для контейнера с виджетом для Humanometr)*.

<script id="humanometr-script" src="https://<Humanometr-host>/index.js?onload=mySaveTokenFunction" async defer></script>
(где Humanometr-host – известный адрес, по которому развернуто приложение Humanometr, см. п. 2.2.3 выше)

Теперь необходимо добавить пустой DOM-контейнер, в который виджет Humanometr будет вставлен автоматически. Контейнер обычно представляет собой (но может быть любым элементом) и должен иметь класс humanometer, атрибут data-sitekey, являющийся вашим общедоступным ключом сайта и атрибут data-callback, являющийся именем функции, которую реализуете вы, для получения идентификатора пользователя.

<div class="humanometr" data-sitekey="<ключ_сайта>" data-callback="<имя_функции>">
Где «ключ_сайта» – ключ, необходимый для записи и подсчета статистики обращений. GUID, соответствующий конкретному защищаемому сайту. По умолчанию в системе зарегистрирован ключ 10000000-ffff-ffff-ffff-000000000001, но можно добавить новые ключи, например
insert into public."SiteConfigs" ("Key", "Expiration", "Name") values
(<uuid>, <годен до>, <статус>);

где

<uuid> - gen_random_uuid(), либо другой произвольный/уникальный контексте таблицы GUID
<годен до> - '2023-03-24 03:00:00.000 +0300'
<статус> - 'Actual' или 'Expired' ;

«имя_функции» – имя функции, которую реализуете вы и которая будет вызвана после инициализации виджета. В данную функцию будут переданы 2 параметра – **secret** и **response**, которые понадобятся далее. Используйте эту функцию, чтобы сохранить эти параметры так, чтобы вам потом удобно было в нужный момент отправить **secret** и **response** в API вашего сайта, например, при сабмите формы на вашем сайте. Далее серверная часть вашего сайта должна сделать запрос в API сервиса капчи

```
curl -X 'POST' \
  'http://localhost:5001/siteverify' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
```

```
  "secret": "<secret>",
  "response": "<response>",
  "remoteip": "<remoteip>",
  "sitekey": "<sitekey>"
}',
```

где

<secret> - secret,
<response> - response,
<remoteip> - адрес вашего сервера (опционально),
<sitekey> - «ключ_сайта»

, чтобы получить информацию о том, действительно ли пользователь успешно прошел капчу. *(если ранее вы указали параметр **onload** в адресе скрипта, то атрибут **data-callback** будет проигнорирован, **onload работает аналогичным образом с data-callback**).*

Пример
виджет:

```
<div class="humanometr" data-sitekey="10000000-ffff-ffff-ffff-000000000001" data-
callback="saveTokenCallback">
```

скрипт сохранения secret и response в сессионное хранилище sessionStorage, чтобы после отправить

```
<script>
    function saveTokenCallback(secret, response) {
        sessionStorage.setItem('secret', secret);
        sessionStorage.setItem('response', response);
        //  далее вы сможете отправить secret, response на серверную часть вашего сайта, чтобы
получить информацию об успешности прохождения капчи пользователем (как описано выше)
    }
</script>
```