Руководство по установке и настройке Grafana   
под CentOS 8

## Общие положения

Grafana используется для построения интерактивных отчетов по временным рядам журналов Оркестратора и Робота с визуализацией результата в виде различных диаграмм. Grafana – отдельное приложение с веб-интерфейсом, работающее как служба Windows. По умолчанию служба слушает 3000 порт. Во вне этот порт не открывается, обращение к Grafana осуществляется через Front (nginx или IIS). Опционально Front может устанавливать для Grafana заголовок авторизации, чтобы не требовалось вводить логин и пароль.

Схема интеграции Grafana с Оркестратором приведена на рисунке 1:



Рисунок 1 – Схема интеграция Grafana с Оркестратором

Согласно рисунку 1 интеграция с Grafana осуществляется следующим образом:

1. Устанавливается приложение Grafana. Например, на том же сервере, где и WebApi.
2. Средствами интерфейса Grafana настраивается источник данных для отчетов – БД ltoolslogs с журналами Оркестратора и Робота.
3. Настраивается конфигурационный файл Grafana (для возможности настроить проксирование через Front, который является общим с Оркестратором).
4. Создается ApiKey, который прописывается далее в конфигурационном файле nginx (не обязательный шаг).
5. Настраивается проксирование с установкой заголовка авторизации в Front. Как было сказано выше, настройка заголовка авторизации не обязательна[[1]](#footnote-1).
6. Средствами интерфейса Grafana создаются или импортируются отчеты[[2]](#footnote-2), отчеты публикуются. При публикации отчетов Grafana формирует внешние ссылки на них. Возможно, некоторые отчеты потребуют создания view в БД.
7. Ссылки на опубликованные отчеты добавляются в конфигурационный файл WebApi, чтобы они открывались через интерфейс Оркестратора. Больше никакой связи Оркестратор с Grafana не имеет.

## 1. Установка Grafana

Далее предполагается что Grafana устанавливается на сервере Оркестратора.

Установочный файл Grafana идет в комплекте поставки. Также может быть скачан с официального сайта Grafana, командой:

$ wget https://dl.grafana.com/oss/release/grafana-7.3.4-1.x86\_64.rpm

* 1. Запускаем установку:

$ sudo dnf -y localinstall grafana-7.3.4-1.x86\_64.rpm

$ sudo systemctl daemon-reload

$ sudo systemctl enable --now grafana-server

* 1. Проверяем, что служба Grafana работает.

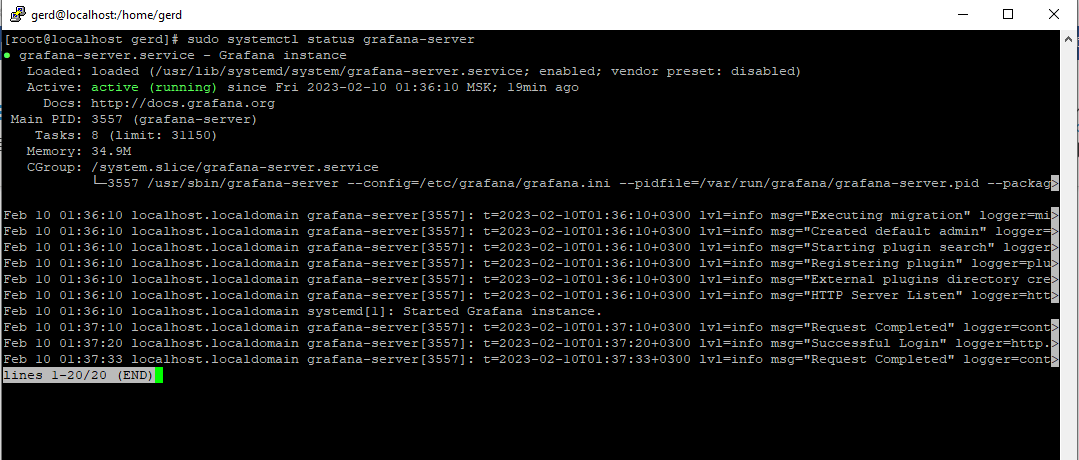
$ sudo systemctl status grafana-server

* 1. Открываем порт 3000

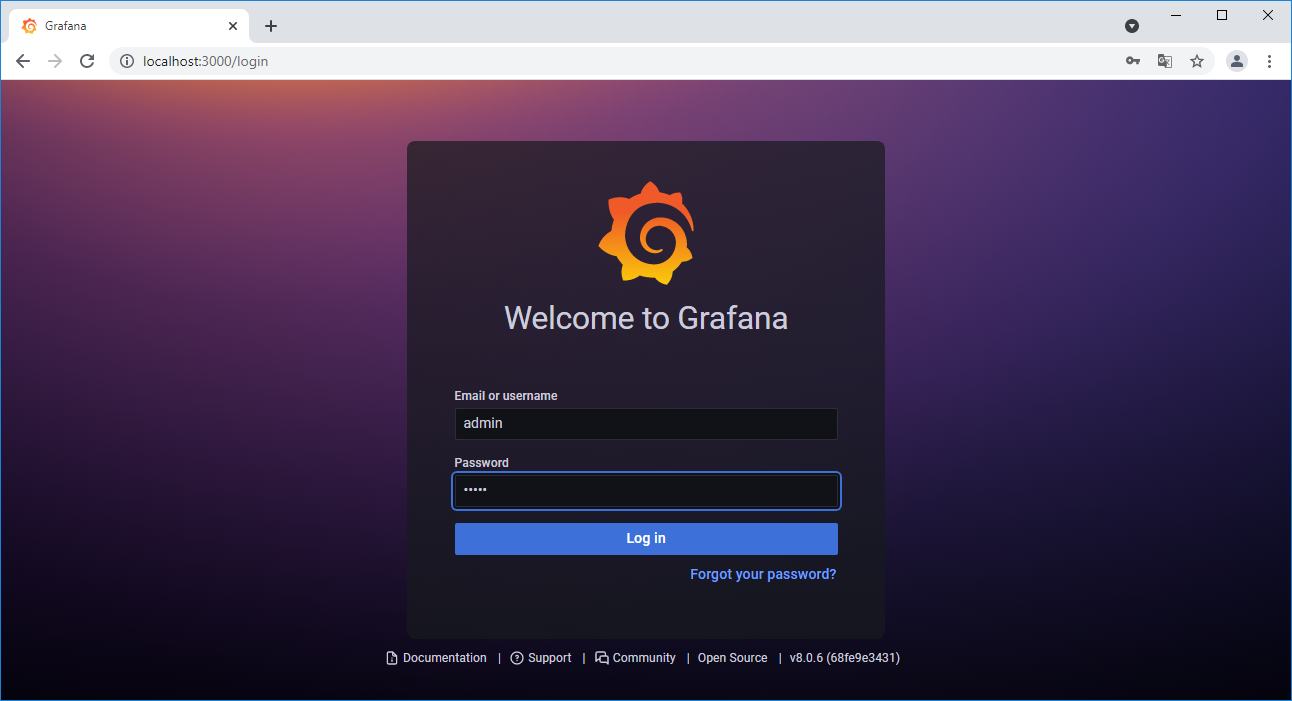
$ sudo firewall-cmd --add-port=3000/tcp --permanent

$ sudo firewall-cmd --reload

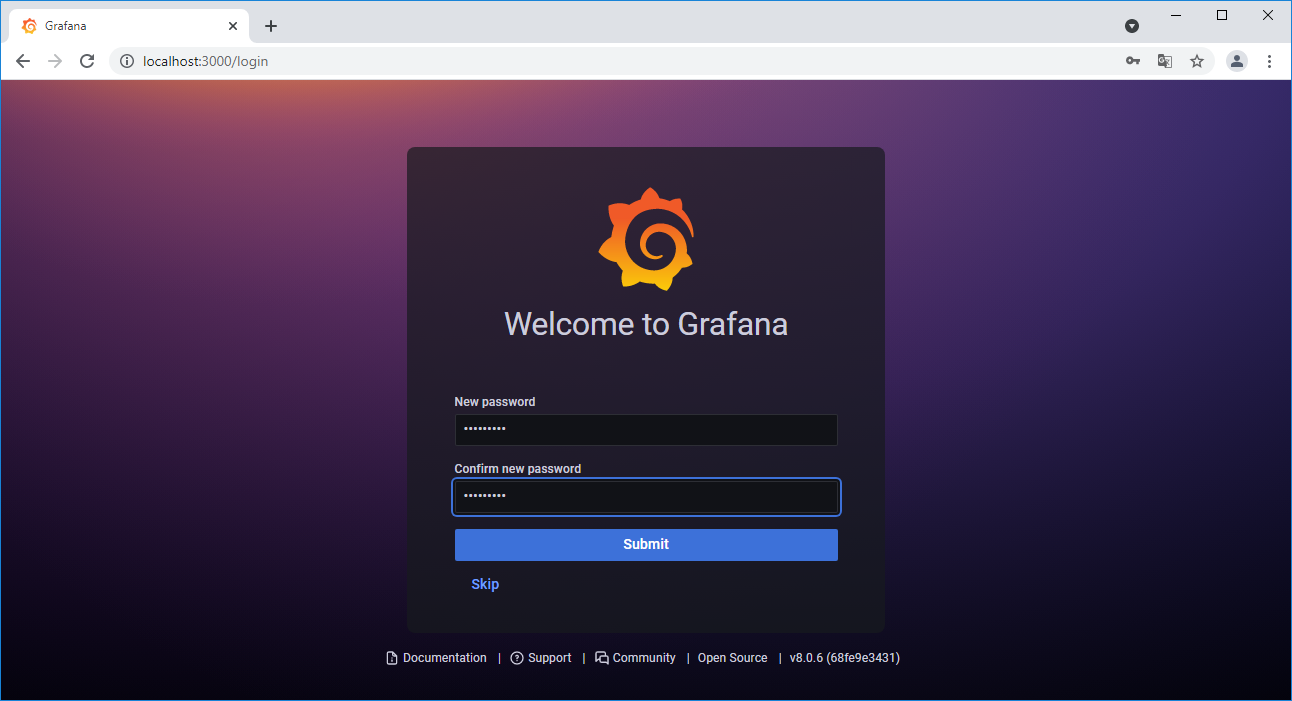
* 1. Никакого специального интерфейса для управления этой службой у Grafana нет.



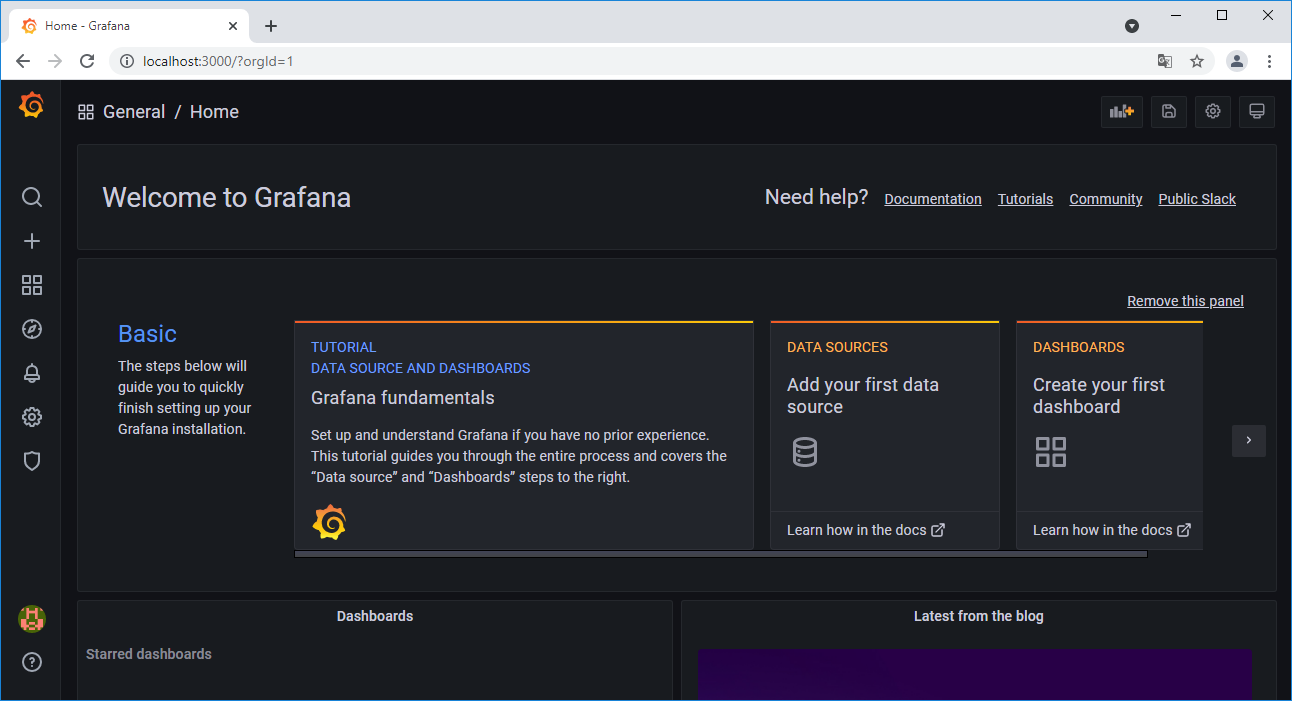
* 1. Заходим в веб-интерфейс Grafana по адресу <http://localhost:3000> и вводим логин. со встроенной учетной записью admin/admin:



* 1. Меняем пароль по умолчанию на новый (Grafana сама предложит это сделать):

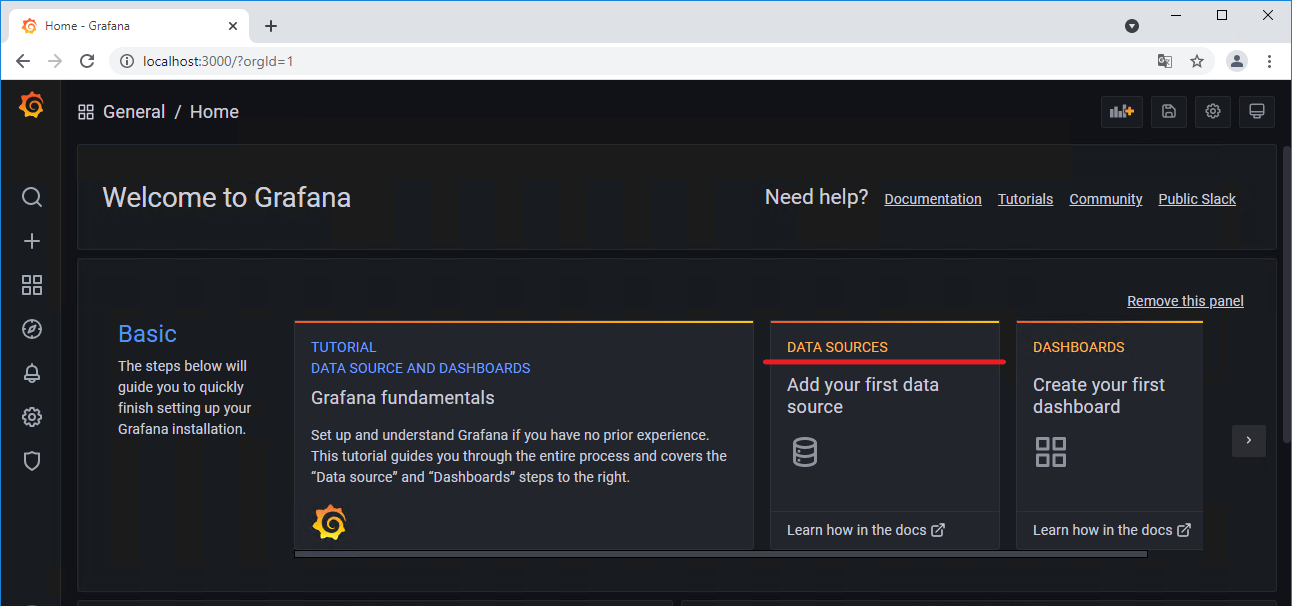


* 1. Откроется панель управления Grafana. Установка Grafana завершена:

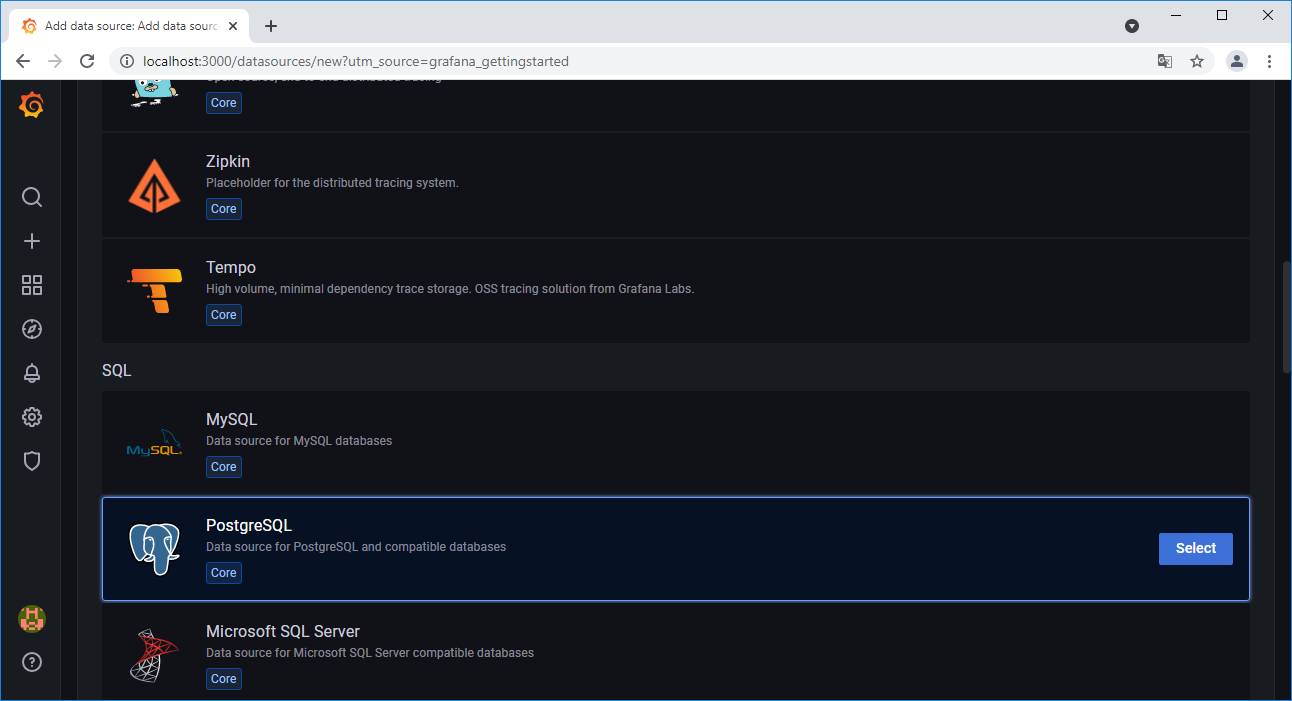


## 2. Настройка источника данных для отчетов

Переходим в раздел Configuration/Data Sources. Для только что установленной Grafana можно перейти из раздела General/Home. Можно воспользоваться левым боковым меню «шестеренка»:



Выбираем поставщика данных PostgreSQL или Microsoft SQL Server (в зависимости от вендора БД Оркестратора):

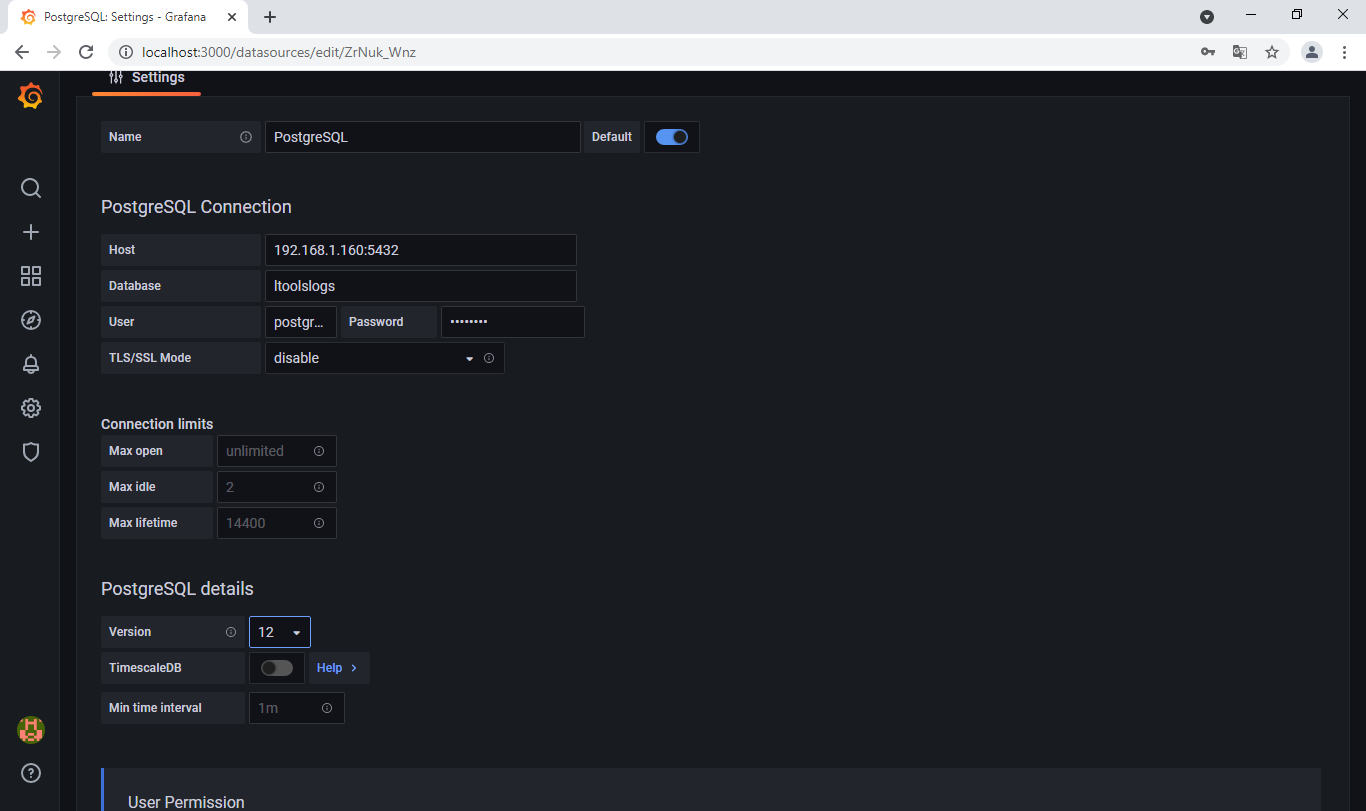




Настраиваем подключение к БД ltoolslogs:

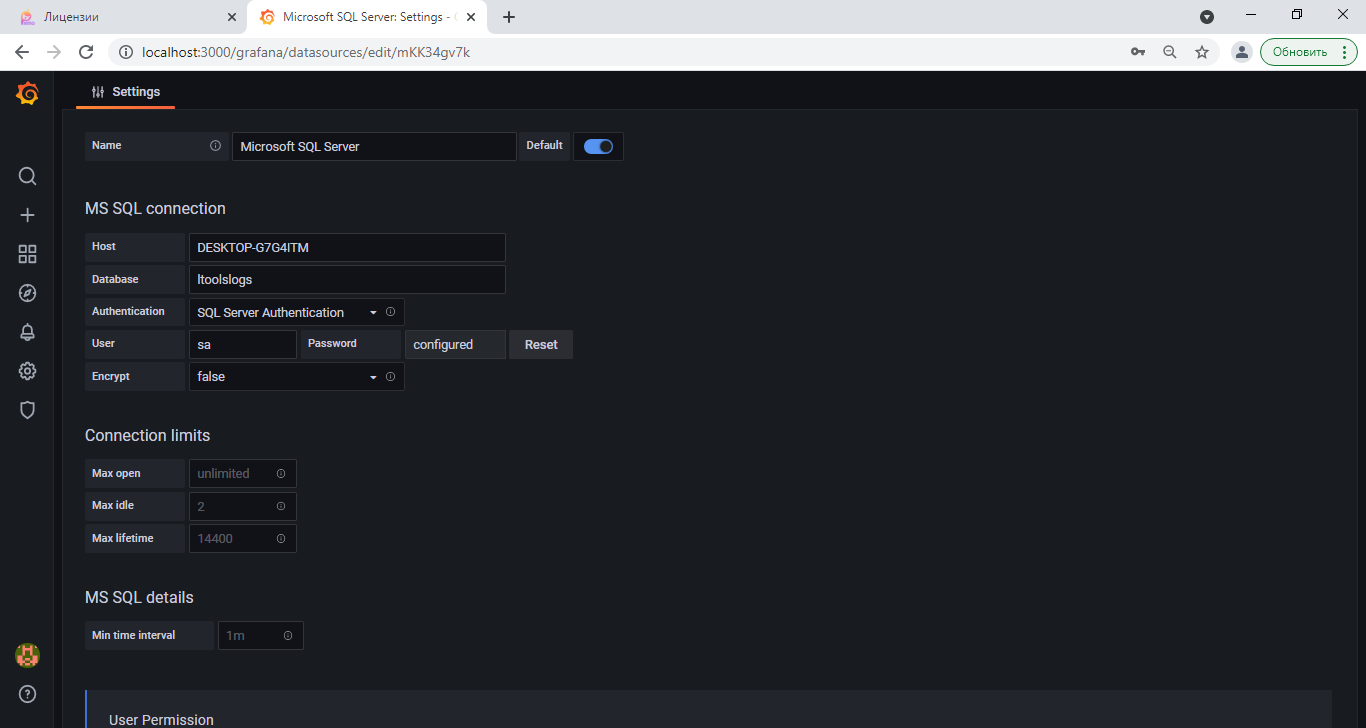
Для PostgreSQL:

* Name – оставляем по умолчанию PostgreSQL (можно выбрать произвольное);
* Host – IP-адрес и порт сервера БД (уточнить в конфигурационном файле WebApi);
* Database – ltoolslogs;
* User/Password – пользователь/пароль БД (уточнить в конфигурационном файле WebApi);
* TLS/SSL Mode – disable;
* Connection limits – все параметры этого блока оставляем по умолчанию;
* Version – 12;
* TimescaleDB[[3]](#footnote-3) – false;
* Min time interval – оставляем по умолчанию 1m;

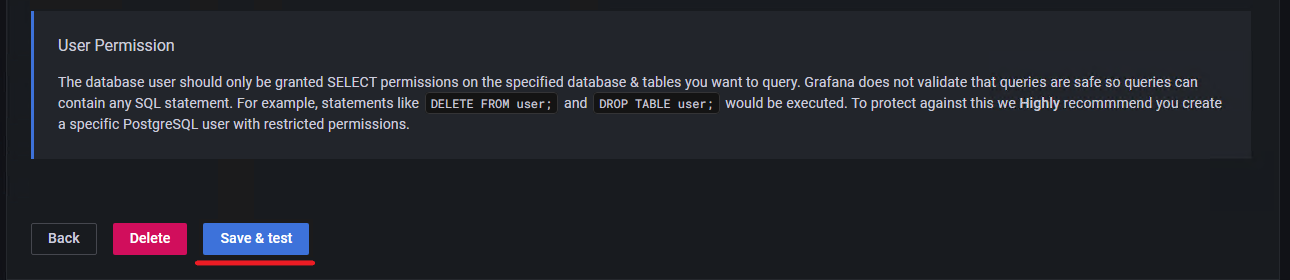


Для Microsoft SQL Server:

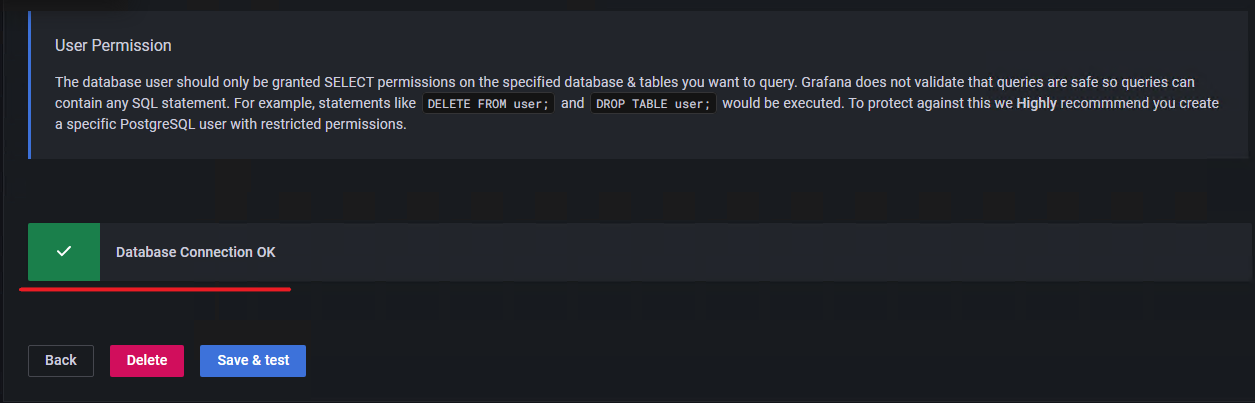
* Name – оставляем по умолчанию Microsoft SQL Server (можно выбрать произвольное);
* Host – IP-адрес (или имя хоста) и порт сервера БД (уточнить в конфигурационном файле WebApi);
* Database – ltoolslogs;
* Authentication – SQL Server Authentication
* User/Password – пользователь/пароль БД (уточнить в конфигурационном файле WebApi);
* Encrypt – false;
* Connection limits – все параметры этого блока оставляем по умолчанию;
* Min time interval – оставляем по умолчанию 1m;



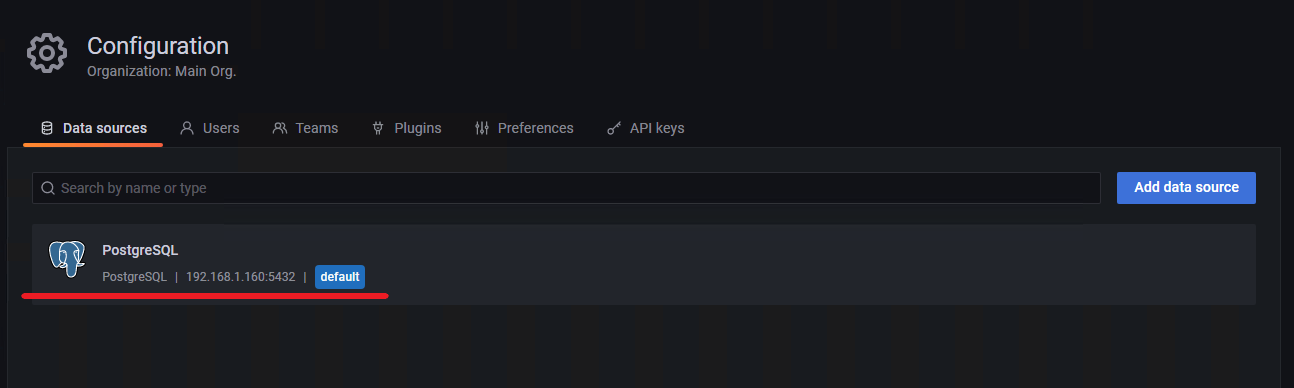
Внизу формы настройки подключения нажимаем кнопку «Save & test»:

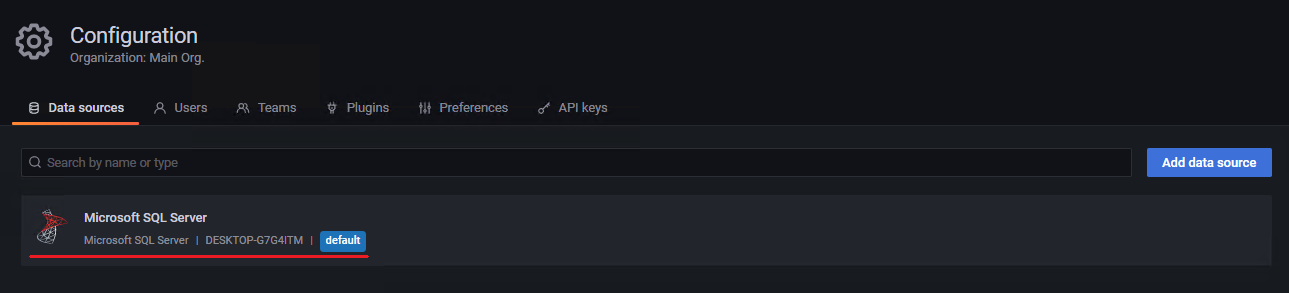


Если все сделано верна и БД ltoolslogs доступна, отобразится сообщение об удачном подключении к БД:



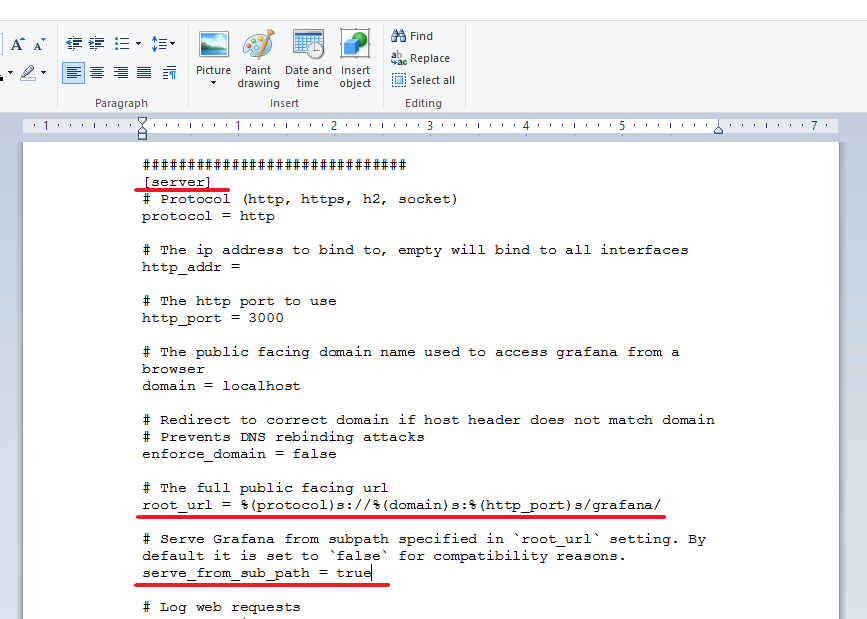
В разделе Configuration/Data Sources будет отображаться созданное подключение с наименованием PostgreSQL (Microsoft SQL Server):





## 3. Настройка конфигурационного файла Grafana

В WordPad (или аналогичной программе, не Notepad!!!) открываем конфигурационный файл Grafana C:\Program Files\GrafanaLabs\grafana\conf\defaults.ini



В секции [server] (можно найти писком по файлу, Ctrl + F) меняем дефолтные значения параметров на:

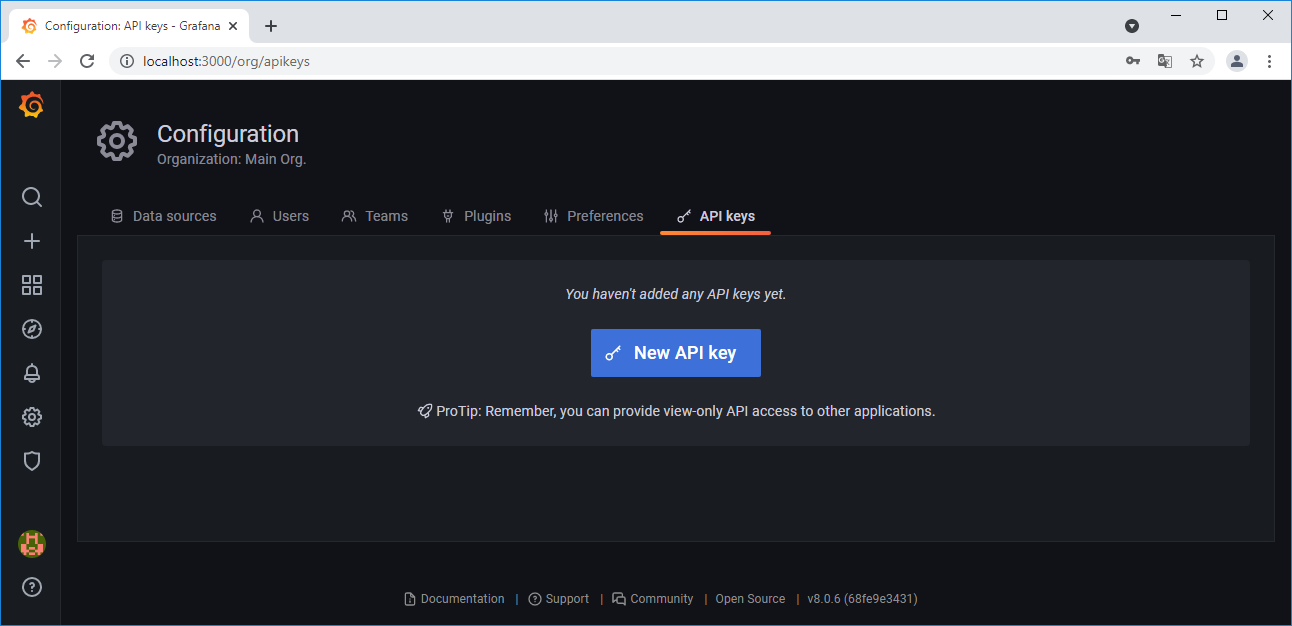
* root\_url = %(protocol)s://%(domain)s:%(http\_port)s/grafana/
* serve\_from\_sub\_path = true

Сохраняем файл и перезапускаем службу Grafana:

$ sudo systemctl restart grafana-server

## 4. Создание Api Key (не обязательно)

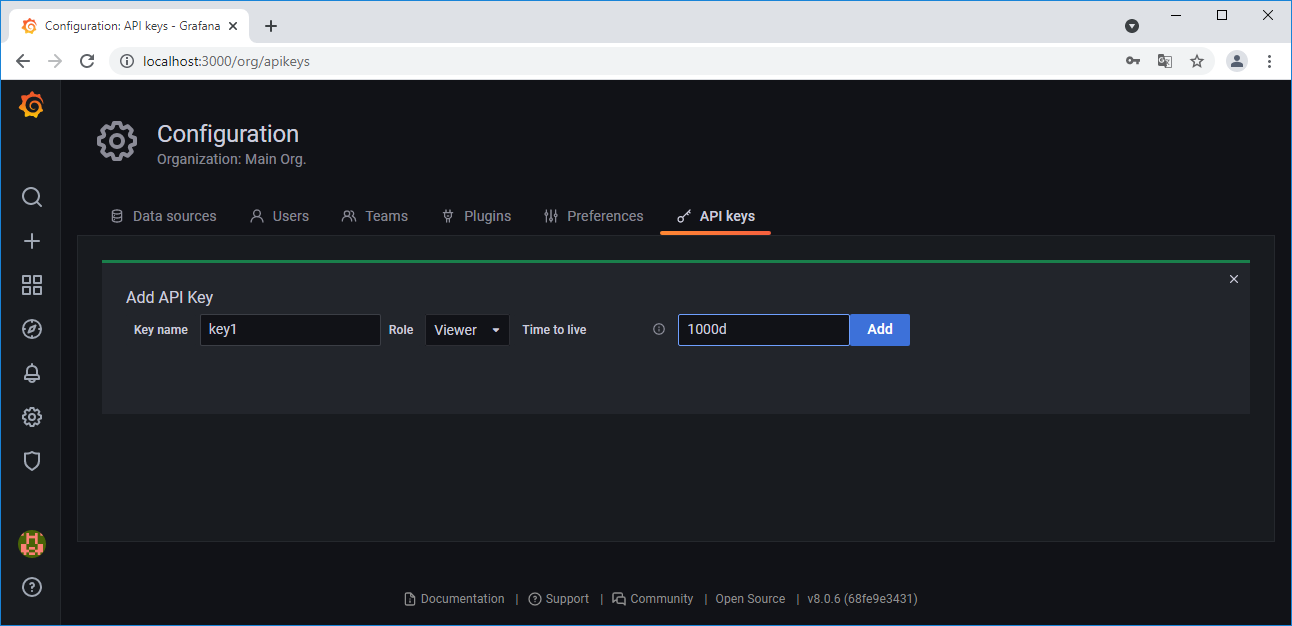
ApiKey создается в интерфейсе Grafana в разделе Configuration/Api keys (боковое левое меню «шестеренка»). Кликаем по кнопке «New API key»:



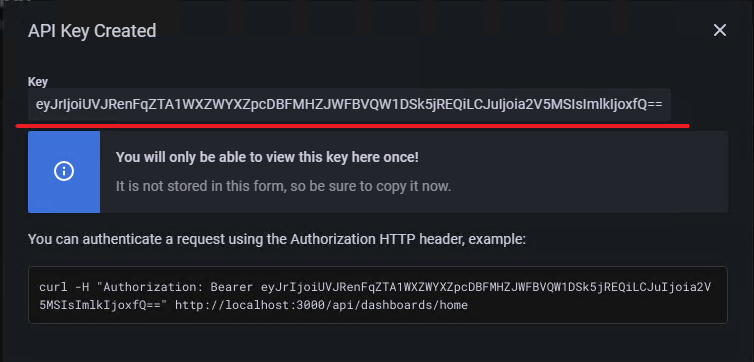
Задаем параметры нового ApiKey:

* Key name – key1 (произвольное наименование на латинице);
* Role – оставляем по умолчанию Viewer (только просмотр);
* Time to life – 1000d. Время в днях, через которое ApiKey станет не активным и авторизация посредством него работать перестанет. Выбирать следует большим.

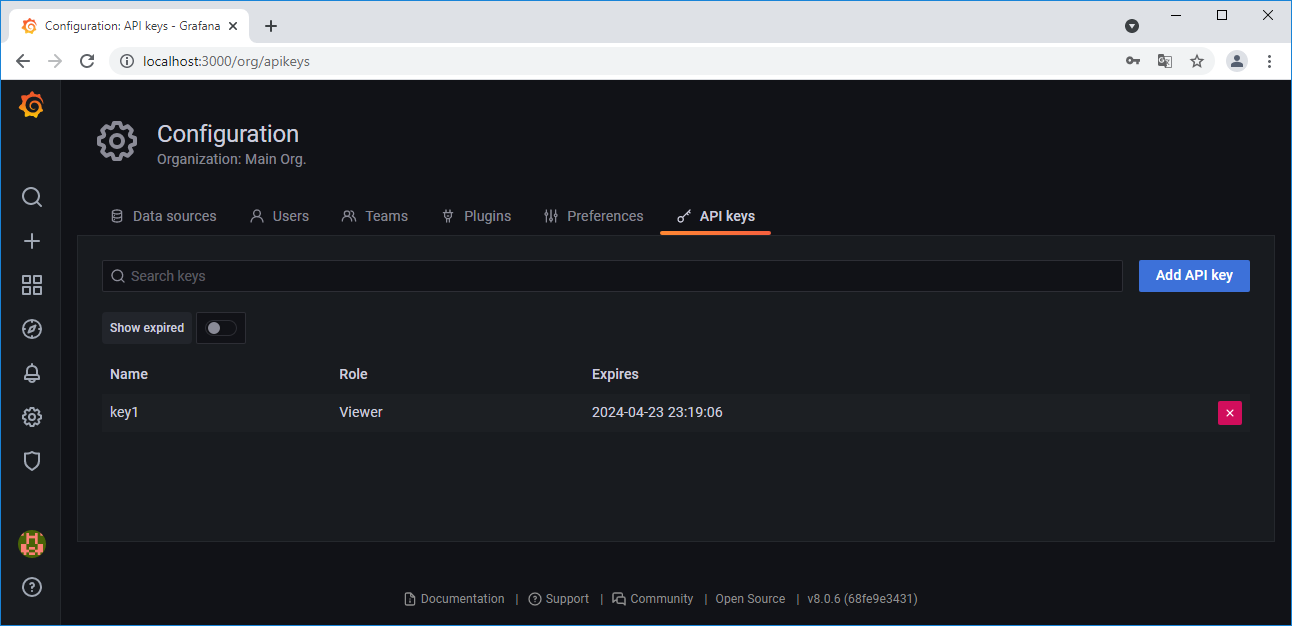
Нажимаем кнопку «Add»:



Сохраняем где-то отдельно полученный ApiKey, так как через интерфейс Grafana увидеть его больше будет не возможно:



Закрываем модальное окно с новым ApiKey. Этот ApiKey под наименованием key1, которое ему дали ранее, будет отображаться в списке всех ApiKey:



## 5. Настройка Front

### 5.1. nginx

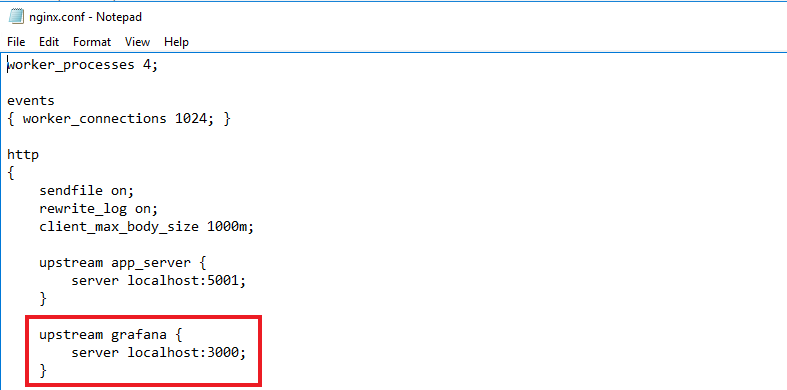
Открываем файл C:\Primo\nginx-1.21.1\conf\nginx.conf и добавляем следующие правила:

После секции upstream app\_server добавляем секцию

upstream grafana {

server localhost:3000;

}



После правила перенаправления location/api/ добавляем правило

location /grafana/ {

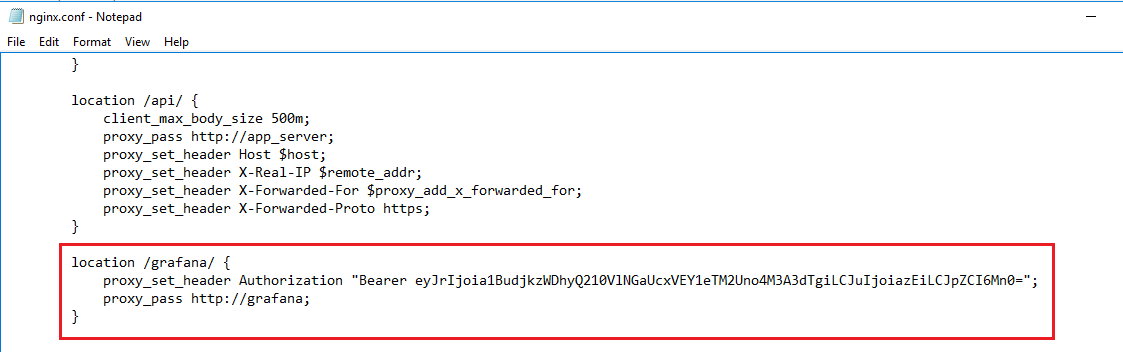
proxy\_set\_header Authorization "Bearer {ApiKey}";

proxy\_pass http://grafana;

}

ApiKey (без фигурных скобок) подставляем созданный в п.4.

Если не нужна автоматическая авторизация в Grafana, заголовок proxy\_set\_header Authorization можно не устанавливать (удалить всю строку)

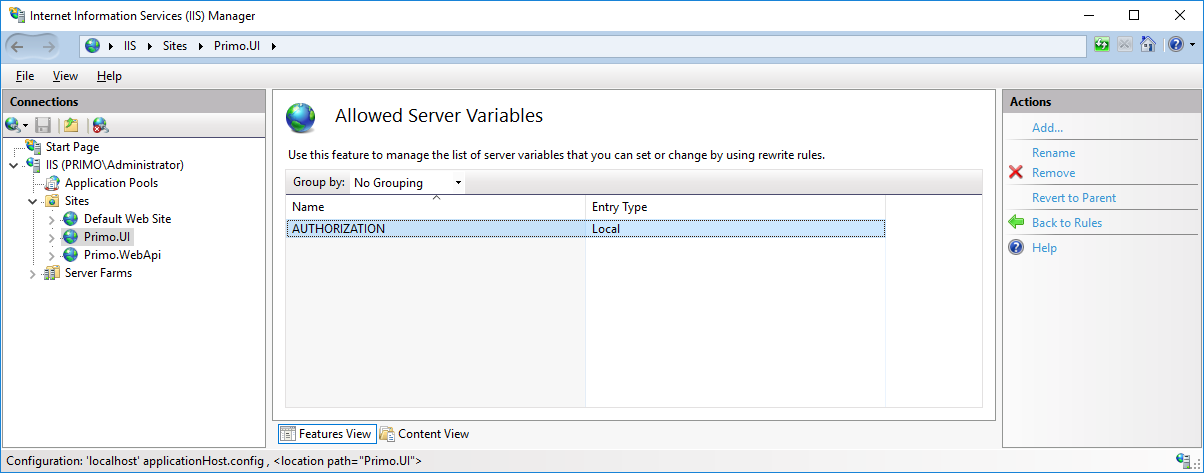


Из cmd перезапускаем nginx:

C:\Primo\nginx-1.21.1>nginx -s reload

### 5.2. IIS

Добавляем серверную переменную AUTHORIZATION:



Открываем файл C:\Primo\UI\web.config и добавляем следующее правило после правила «Reverse Proxy to API»:

<rule name="Grafana" stopProcessing="false">

<match url="^grafana(.\*)" />

<action type="Rewrite" url="http://localhost:3000/grafana{R:1}" />

<serverVariables>

<set name="AUTHORIZATION" value="Bearer eyJrIjoickxTcUFDTWVvajlGNVVBT3pDUUxtQXBhMHFRbmVrOEEiLCJuIjoia2V5MSIsImlkIjoxfQ==" />

</serverVariables>

</rule>



## 6. Создание или импорт отчетов

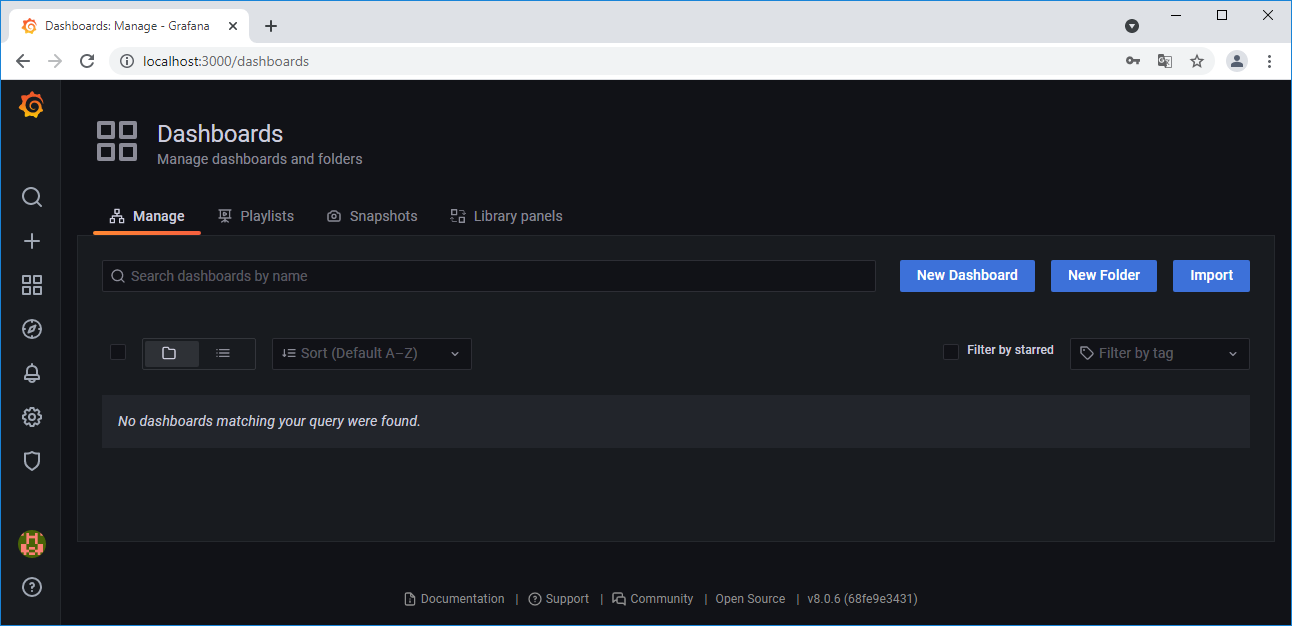
Создание отчетов – отдельная большая тема. Для этого требуется, как минимум:

* владение средствами Grafana;
* владение SQL, в частности, его диалектом для PostgreSQL или Microsoft SQL Server и   
  SQL-образным DSL Grafana;
* понимание структуры и семантики данных БД ltoolslogs.

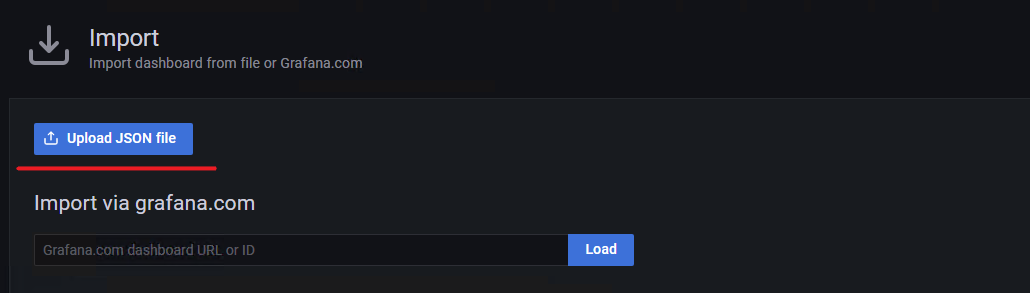
Поэтому для примера рассмотрим импорт готового отчета в Grafana.

Вообще говоря, наличие view в БД для отчетов в Grafana не обязательно. Но, для этого примера потребуется создать view – v\_AllWorked-postgres (или v\_AllWorked-mssql). Скрипт создания v\_AllWorked-postgres.sql (v\_AllWorked-mssql.sql) включен в поставку. Этот скрипт требуется выполнить в БД ltoolslogs.

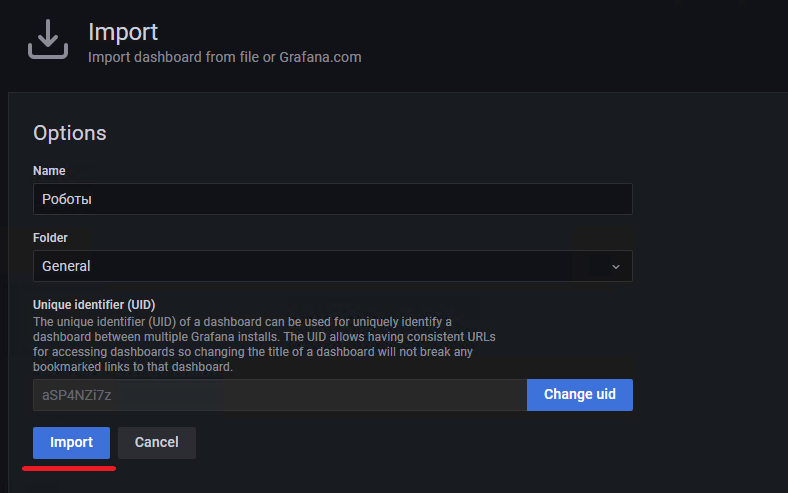
Для импорта отчета переходим в раздел Dashboards/Manage и нажимаем кнопку «Import»:



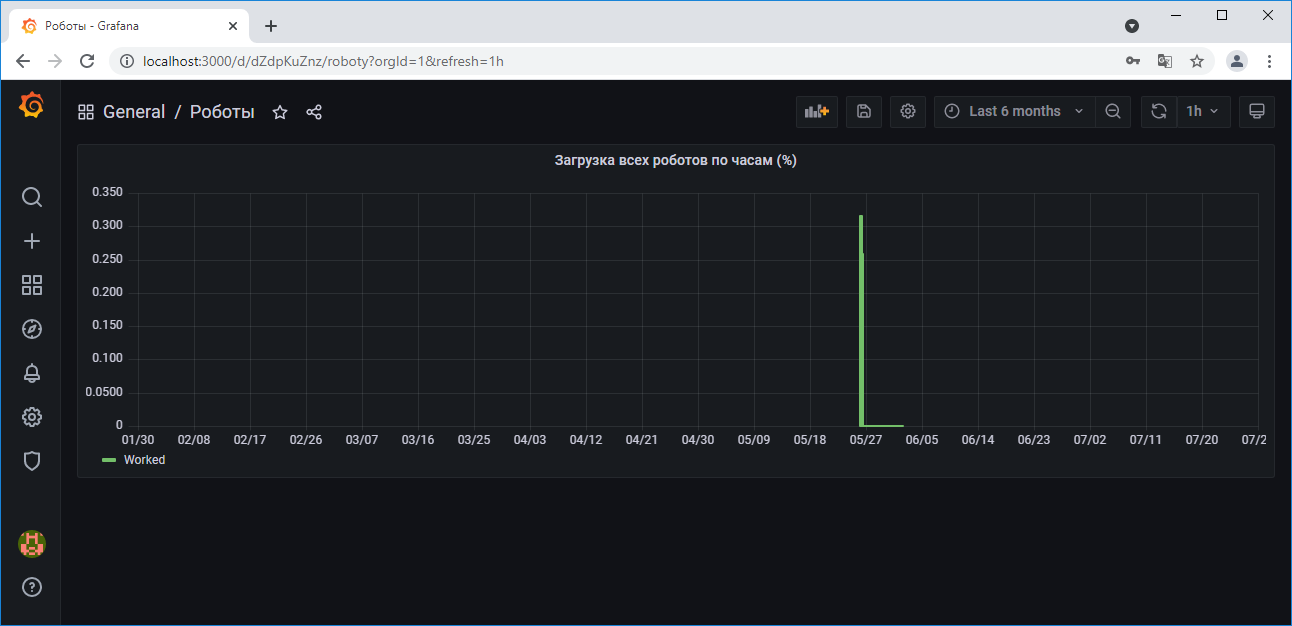
В открывшейся форме нажимаем кнопку «Upload JSON file»:



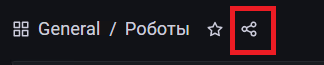
Выбираем идуший в комплекте поставки пример отчета – файл   
Роботы-1627543691525.json – и нажимаем кнопку «Import»:



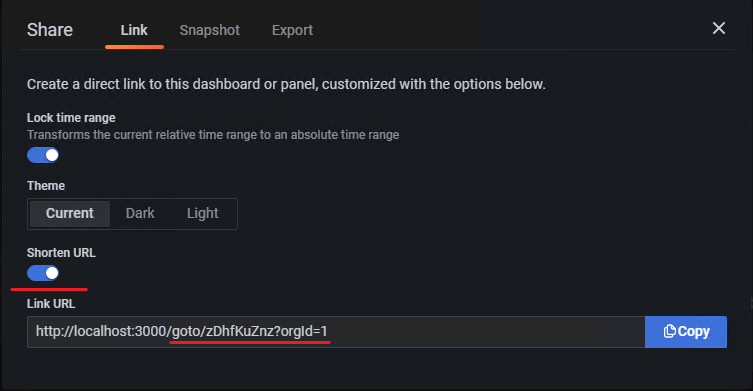
Если все выполнено верно, и файл отчета корректный, сразу откроется сам отчет:



Далее публикуем его, чтобы получить внешнюю ссылку на этот отчет. Нажимаем на кнопку «Share dashboard or panel»:



В открывшейся форме ставим Shorted URL = true и копируем адрес ссылки:

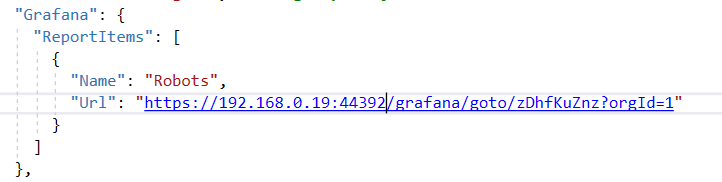


Окончательная ссылка на отчет получается после ручной корректировки как

https://{IP Оркестратора}:44392/grafana/goto/zDhfKuZnz?orgId=1

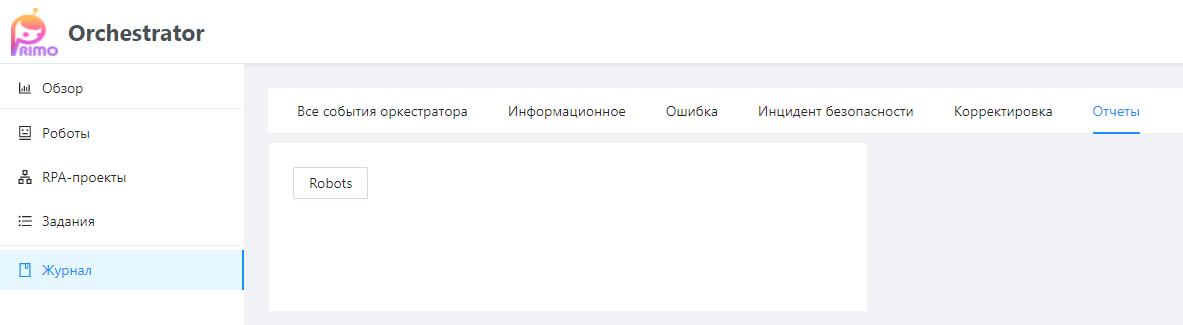
## 7. Добавление в конфигурационный файл WebApi ссылок на опубликованные отчеты

Ссылки на опубликованные отчеты добавляются в конфигурационный файл C:\Primo\WebApi\appsettings.ProdWin.json в секцию Grafana:ReportItems:



Задается Url отчета, полученный в п.6, и произвольное наименование отчета. После этого службу WebApi нужно перезапустить.

Проверить, что все настроено верно, можно через интерфейс Оркестратора в разделе Журнал/Отчеты:



По клику по кнопке Robots («Robots» – наименование отчета из конфигурационного файла) откроется отчет в Grafana.

1. В некоторых случаях не желательна, так как ослабляет безопасность [↑](#footnote-ref-1)
2. Документация по созданию отчетов (дашбордов в терминах Grafana) находится на официальном сайте Grafana <https://grafana.com/>. Там же можно скачать дистрибутив с последней версией Grafana. [↑](#footnote-ref-2)
3. Grafana поддерживает TimescaleDB, что находит свое отражение при создании отчетов. За подробной информацией необходимо обратиться к официальной документации на официальном сайте Grafana. [↑](#footnote-ref-3)