Docker x Grafana p.1

**Установка Docker**

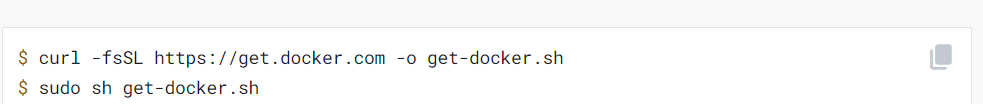
Первым делом установим curl:

**apt update**

**apt install curl -y**

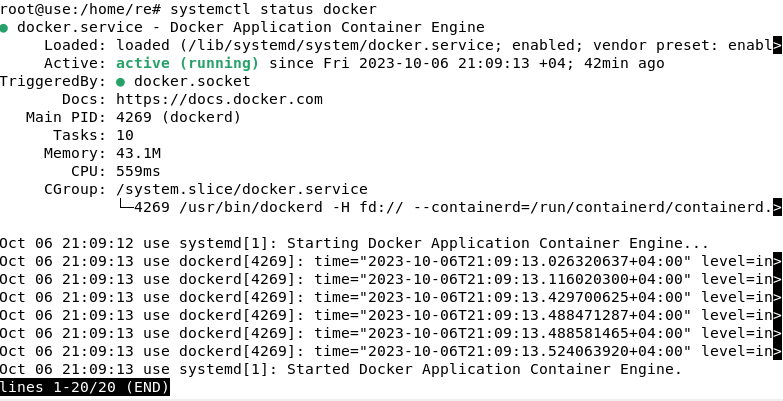
curl позволит нам выполнять http-запросы не выходя из командой строки

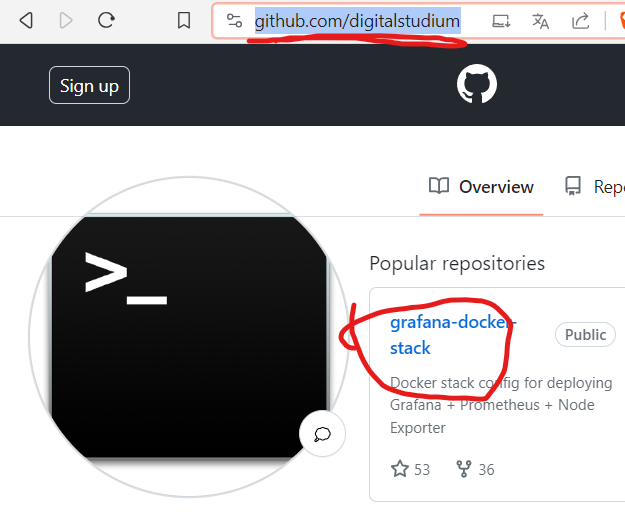
Установим Docker:



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

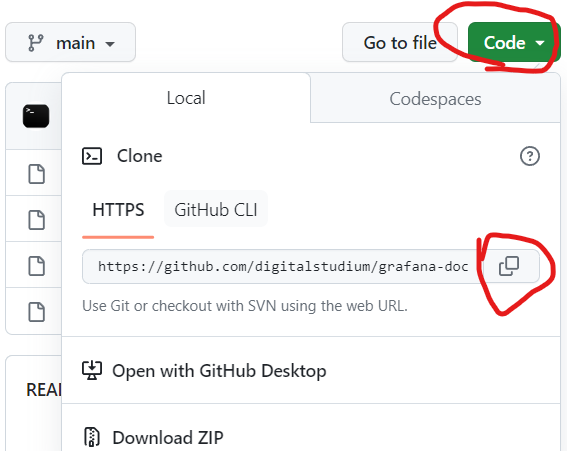
Командой **systemctl status docker** проверяем его работу:





Первым делом заходим в браузер и ищем там **github.com/digitalstudium**

Переходим в репозиторий **grafana-docker-stack**

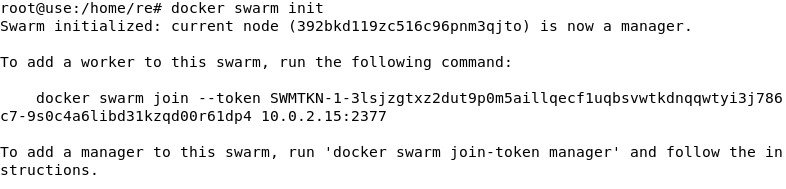


Копируем ссылку на репозиторий

Заходим на машину и пишем команду **git clone + адрес репозитория**:



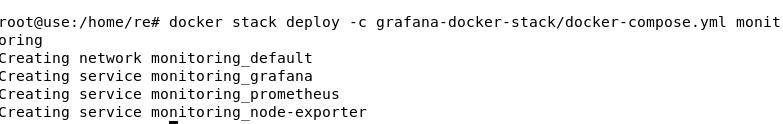
Далее пишем команду **docker swarm init**:



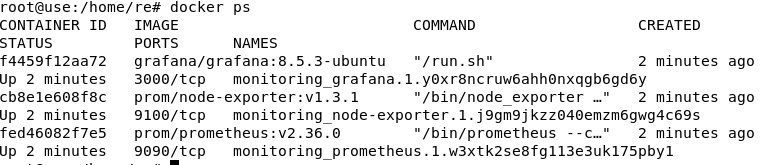
(Делает наш сервер менеджером узла, Docker Swarm – стандартный оркестратор для контейнеров)

Далее создаем стек:

**docker stack deploy -c grafana-docker-stack/docker-compose.yml monitoring**



С помощью команды **docker ps** проверяем работу – если все правильно выполнено видим что все три контейнера поднялись:



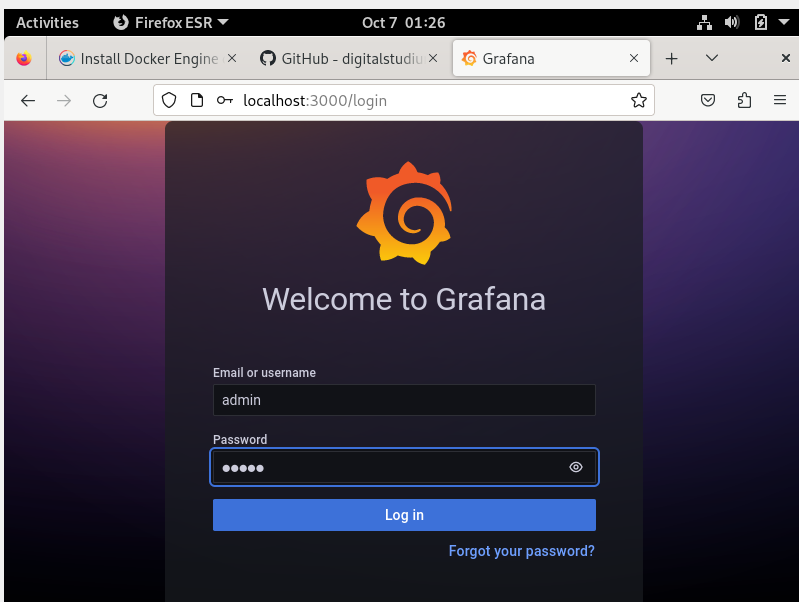
Grafana

Prometheus

Node-explorer

Давайте зайдем в браузер и проверим что все контейнеры исправно работают, на скрине выше мы видим столбик **PORTS –** ориентируемся на него

[**http://localhost:3000**](http://localhost:3000)

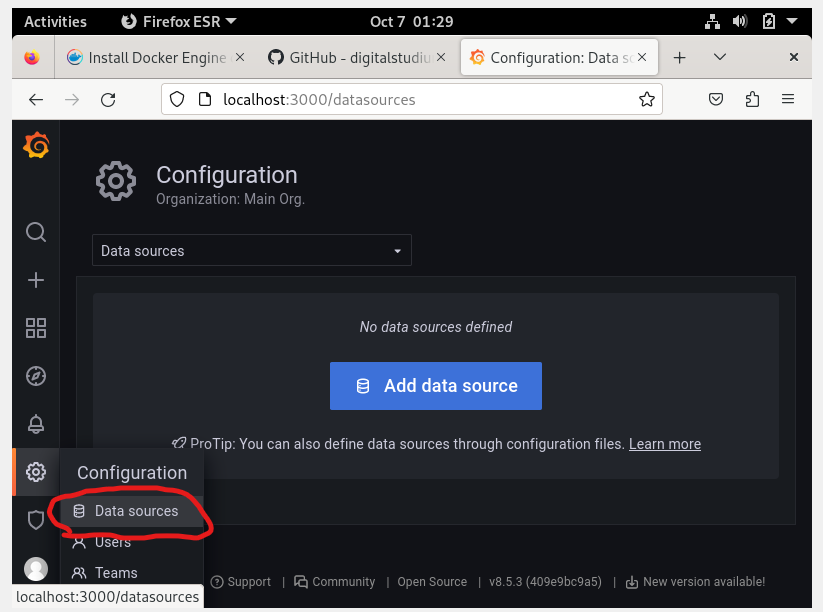
****

Дефолтный пароль для входа в Grafana **– admin**

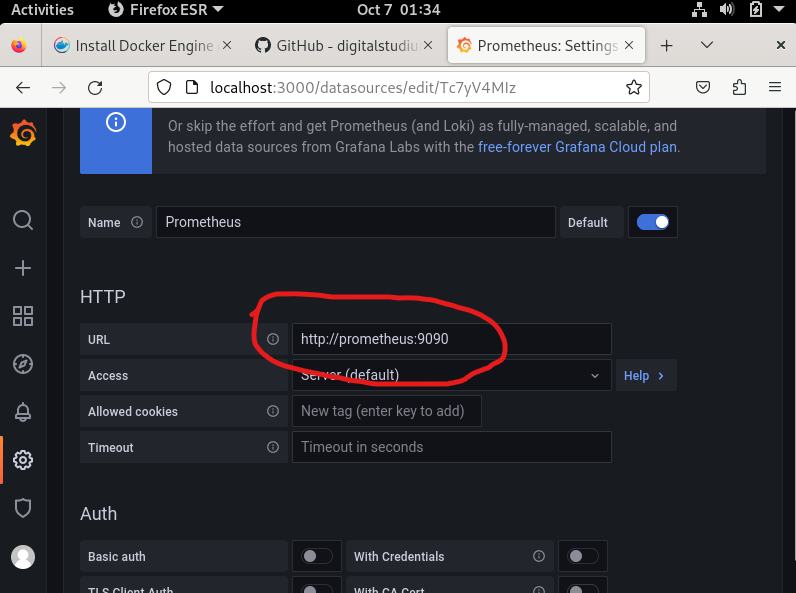
(его нужно изменить, для удобства поставьте **toortoor**)

По такому же принципу проверяем работу остальных контейнеров

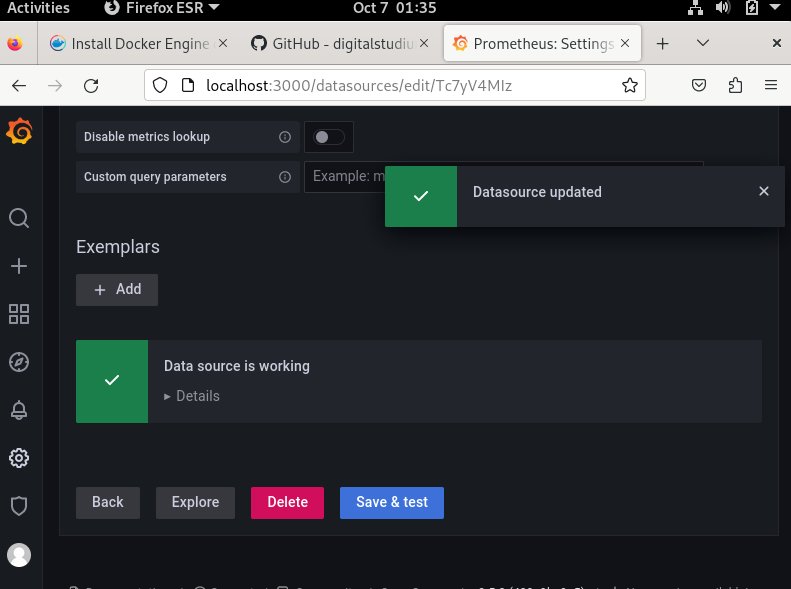
Заходим в Grafana, нам нужно добавить Prometheus, для этого нажимаем на шестеренку и переходим в Data Sources



Для этого нам нужно ввести адрес контейнера Prometheus:



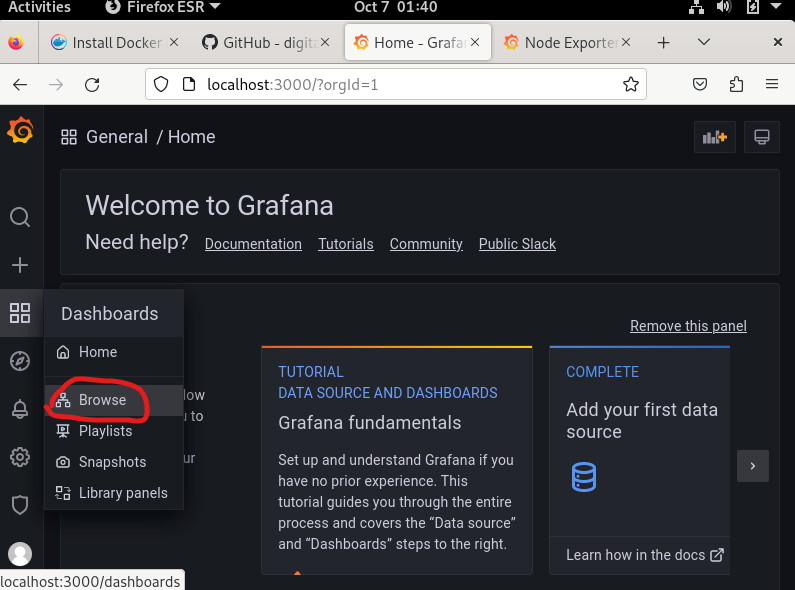
Листаем вниз, нажимаем на сохранение и если все выполнено правильно Grafana выдаст что Prometheus активен:



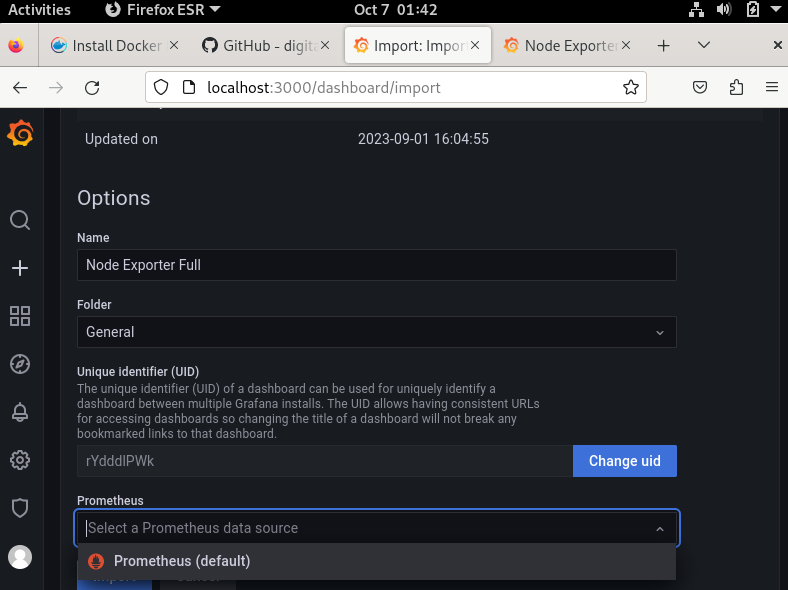
Но опять же, этих шагов недостаточно для полноценного мониторинга системы. Теперь нам нужно импортировать DashBoard графаны

Для этого нам нужно зайти в браузер и набрать **Grafana dashboard node exporter**, переходим по первой ссылке и нажимаем кнопку **Copy ID**

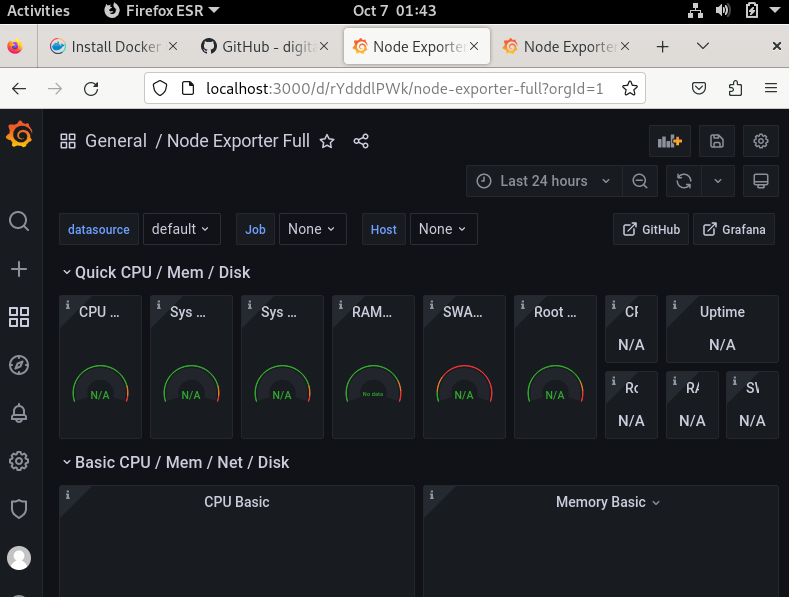
Далее заходим обратно в Grafana и нажимаем на **Browse**:



Далее нажимаем на импорт и вводим скопированный ID, после чего в появившемся окне выбираем Prometheus



На выводе получаем dashboard, но информация на нем не отображается

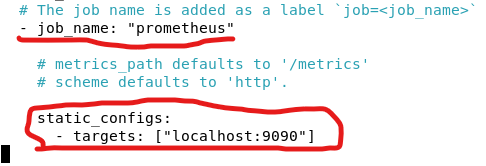


Причина проста – передачей данных с компьютера на Prometheus занимается node-exporter, который мы еще не привязали

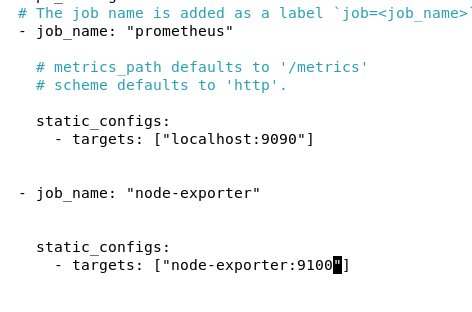
Это делается следующим образом –

**nano /var/lib/docker/volumes/monitoring\_prom-configs/\_data/prometheus.yml**

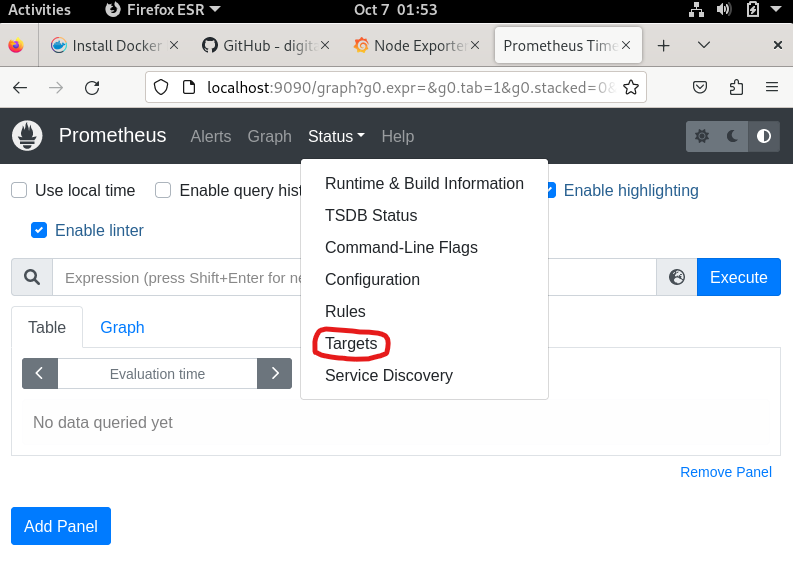
Открываем файл и листаем в самый низ



Нам нужно продублировать эти строчки в файле, только с данным node-exporter



Сохраняем файл и заходим в Prometheus



Prometheus пока не знает о изменениях, поэтому нужно перезагрузить конфинг, идем в терминал и пишем:

**docker ps | grep prometheus**

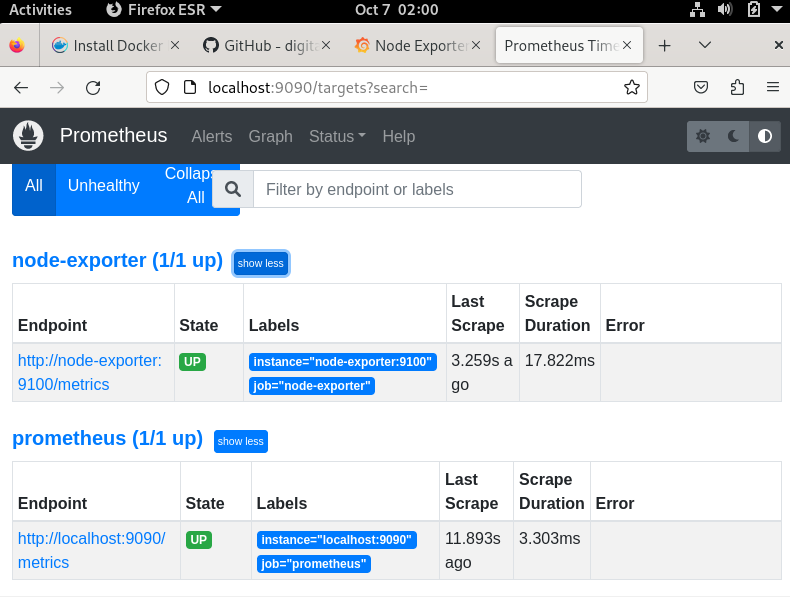
Таким образом мы выясним ID контейнера



Копируем его, и пишем команду **docker kill -s SIGHUP + ID контейнера**

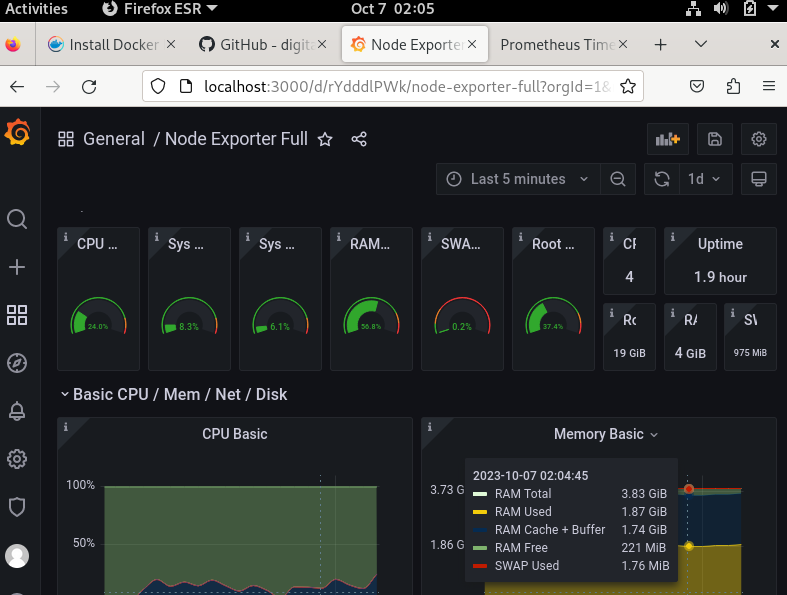


Сигнал послан, идем проверять



Видим что node exporter запущен и работает

**Получили подобный результат – практика выполнена**

****

**К.В.**

**1.?**

**2.?**

**3.?**

**4.?**

**5.?**