Настройка мониторинга за 15 минут с помощью Grafana и Prometheus

После настройки получим даш боард:



Клонируем репозиторий c yaml файлами для докер:

user@Ubuntu:~$ **git clone** [**https://git.digitalstudium.com/digitalstudium/grafana-docker-stack.git**](https://git.digitalstudium.com/digitalstudium/grafana-docker-stack.git)

После чего у нас появится директория grafana-docker-stack

user@Ubuntu:~$ **ls -lha grafana-docker-stack/**

total 60K

drwxrwxr-x 3 user user 4.0K Nov 1 17:27 .

drwxr-x--- 15 user user 4.0K Nov 1 17:27 ..

-rw-rw-r-- 1 user user 893 Nov 1 17:27 docker-compose.yml

drwxrwxr-x 8 user user 4.0K Nov 1 17:27 .git

-rw-rw-r-- 1 user user 35K Nov 1 17:27 LICENSE

-rw-rw-r-- 1 user user 490 Nov 1 17:27 node-exporter.yml

-rw-rw-r-- 1 user user 1.5K Nov 1 17:27 README.md

Устанавливаем docker:

user@Ubuntu:~$ **sudo apt update && apt install docker.io**

**Рассмотрим файл** [**grafana-docker-stack**](https://git.digitalstudium.com/digitalstudium/grafana-docker-stack/src/branch/main)**/docker-compose.yml**

* **Он содержит описание 3 контейнеров, это графана, прометеус и ноуд экспортер**
* **1 контейнер Графана — это визуализатор метрик, то, что мы видим в браузере**
* **2 контейнер Прометеус — это база данных, в которой хранятся метрики**
* **3 контейнер Ноуд экспортер — это тот контейнер, который будет собирать метрики с сервера линукс и будет их отдавать на порту 9100**
* **Схема работы такая -> прометеус забирает метрики с ноуд экспортера -> графана забирает метрики с протемеуса и показывает в веб браузере**

Следующий шаг нужно набрать команду:

user@Ubuntu:~$ **docker swarm init**

Swarm initialized: current node (cobxqmqw03094szllzs2b1n4u) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-0qx610as4yehz8hpii5kikpmoek2couseeedsspbp6465d24qq-1p7iy4qn1s54urjpke6sm0m7m 10.0.10.200:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

user@Ubuntu:~$ **docker stack deploy -c grafana-docker-stack/docker-compose.yml monitoring**

Since --detach=false was not specified, tasks will be created in the background.

In a future release, --detach=false will become the default.

Creating network monitoring\_default

Creating service monitoring\_grafana

Creating service monitoring\_prometheus

Creating service monitoring\_node-exporter

**Рассмотрим файл** [**grafana-docker-stack**](https://git.digitalstudium.com/digitalstudium/grafana-docker-stack/src/branch/main)**/docker-compose.yml**

* **У каждого контейнера есть раздел порт. И справа от двоеточия это порт самого контейнера, а слева от двоеточия это тот порт, который будет слушать ваш хост. Таким образорм если вы хотите зайти в вашу графану не по порту 3000, а по порту 80. То меняем порт слева. То есть ports:- "3000:3000" меняем на ports:- "80:3000"**
* **У каждого контейнер есть раздел volume. Для тех кто не знает, если Docker контейнер рестартовал то все данные из него потеряются. То есть при перезагрузке сервера все данные теряются. То есть дашборд графаны будет отображать пустые значения. Поэтому нужно директории в контейнере в которых сохраняются дашборд или конфиги, объявить Volume (волюмами)**
* **Поэтому для Графаны и Прометеуса объявлено по 2 volume.** 
  + **Графана – 1 volume для данных (наш дашбоард) и 1 volume для конфигов**
  + **Прометеус – похожие volume как с графаной**

Пример для Графаны:

volumes:

- grafana-data:/var/lib/grafana

- grafana-configs:/etc/grafana

Пример для Прометеуса:

volumes:

- prom-data:/prometheus

- prom-configs:/etc/Prometheus

И вот у нас должны подняться все контейнеры, проверим командой docker ps:

user@Ubuntu:~/grafana-docker-stack**$ docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6424100f7a34 prom/prometheus:v2.36.0 "/bin/prometheus --c…" 24 minutes ago Up 24 minutes 9090/tcp monitoring\_prometheus.1.zoi2gbq0r5dvczmfqm6jjox2n

81cf31c7276d prom/node-exporter:v1.3.1 "/bin/node\_exporter …" 24 minutes ago Up 24 minutes 9100/tcp monitoring\_node-exporter.1.jtn0gskpcujmhmx5gt90gbfzf

4c04af3ab032 grafana/grafana:8.5.3-ubuntu "/run.sh" 24 minutes ago Up 24 minutes 3000/tcp monitoring\_grafana.1.iiyzc52tcxms4y1bz9raamp1i

Видим, что у нас поднялось 3 контейнера:

* Графана
* Прометеус
* Ноуд экспортер

Заходим в Графану на адрес нашей машины в браузере <http://10.0.10.200:3000/login>

По умолчанию логин/пароль для графаны это admin/admin

При первом входе нужно сменить пароль.

Далее настраиваем источник данных, чтобы Графана знала откуда ей брать данные для графиков. В нашем случае нужно забирать их у Прометеуса. Вообщем добавляем источник данных Data Sources -> Prometheus -> <http://prometheus:9090>

Нажимаем копку Save and Test

Должна появится надпись Data source is working – она означает что Прометеус находится в рабочем режиме.

Мы можем проверить работает ли Прометеус, открыв в браузере <http://10.0.10.200:9090/>

Мы можем проверить работает ли Ноуд экспортер, открыв в браузере <http://10.0.10.200:9100>

Если все открывается значить все работает.

Этого мало чтобы заработал мониторинг

Теперь скачиваем файл json по запросу **grafana dashboard node exporter** <https://grafana.com/api/dashboards/1860/revisions/37/download>

Вставляем его в графану – 4 квадратика - > Browse (Manage) -> Upload Jsone -> Load -> Выбираем в новом окне Prometheus и нажимаем Импорт

После появляется настроенный даш боард, но в нем ничего нет, потому-что в наш ноуд экпортер нужно добавить прометеус.

Это делается след. образом – открываем файл:

root@Ubuntu:/home/user/grafana-docker-stack**# vim /var/lib/docker/volumes/monitoring\_prom-configs/\_data/prometheus.yml**

**Добавляем в него 3 строчки:**

**- job\_name: "node-exporter"**

**# metrics\_path defaults to '/metrics'**

**# scheme defaults to 'http'.**

**static\_configs:**

**- targets: ["node-exporter:9100"]**

Далее заходим в Прометеусоткрыв в браузере <http://10.0.10.200:9090/>

Далее заходи в раздел Status -> Targets

Здесь у нас должен появится наш ноуд экпортер. Но этого не будет потому-что прометеус не знает об этом, чтобы он узнал что перезагрузить конфиги прометеуса. Чтобы он узнал нужно послать сигнал sighub контейнеру Прометеуса.

root@Ubuntu:/home/user/grafana-docker-stack# **docker ps | grep promеtheus**

**6424100f7a34** prom/prometheus:v2.36.0 "/bin/prometheus --c…" 54 minutes ago Up 54 minutes 9090/tcp monitoring\_prometheus.1.zoi2gbq0r5dvczmfqm6jjox2n

root@Ubuntu:/home/user/grafana-docker-stack# **docker kill -s SIGHUP 6424100f7a34**

Теперь заходим обратно в прометеус , перезагружаем страничку и видем что у нас добавился источник данных Node exporter.

Если зайти в графану, мы увидим данные.

Чтобы удалить все три ноды Docker (контейнеры)

user@Ubuntu:~$ **docker stack rm monitoring**

#Также нужно удалить их локальные хранилища

docker volume ls

docker volume rm monitoring\_grafana-configs

docker volume rm monitoring\_grafana-data

docker volume rm monitoring\_prom-configs

docker volume rm monitoring\_prom-data

Чтобы обновить запускаемые версии, вводим в гугле - docker hub Grafana

Заходим по первой ссылке, щелкаем Tags. И видим разные версии с 8 по 11. Например берем 11.03.0, открываем yaml файл **docker-compose.yml** и меняем там версию в строчке **image: grafana/grafana:8.5.3-ubuntu**