

Развертывание стека мониторинга prometheus / grafana

--

Для начала работ вам понадобится:

- *docker*
- *docker-compose*

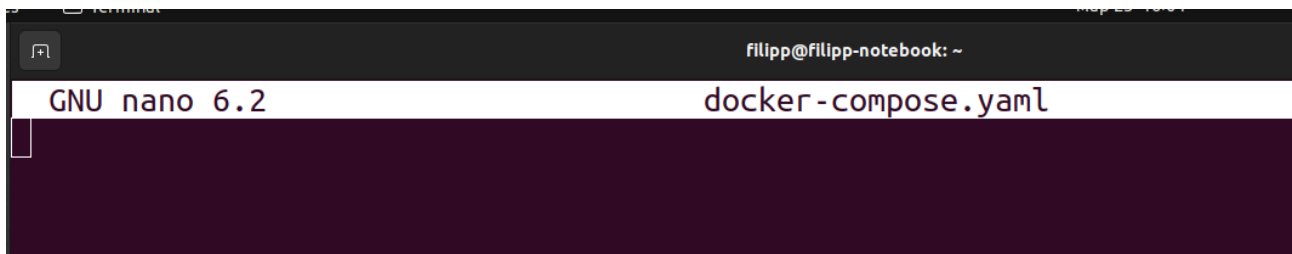
Подробная установка всех необходимых компонентов рассмотрена в предыдущих уроках

--

Часть 1. Развертывание стека prometheus / grafana

Создадим файл `docker-compose.yml`:

```
$ nano docker-compose.yml
```



Добавим в файл следующее содержимое и сохраним изменения:

```
version: "3.9"
services:

  grafana:
    image: grafana/grafana:8.5.3-ubuntu
    ports:
      - "3000:3000"
    volumes:
```

- grafana-data:/var/lib/grafana
- grafana-configs:/etc/grafana

prometheus:

image: prom/prometheus:v2.36.0

ports:

- "9090:9090"

volumes:

- prom-data:/prometheus
- prom-configs:/etc/prometheus

node-exporter:

image: prom/node-exporter:v1.3.1

ports:

- "9100:9100"

volumes:

- /proc:/host/proc:ro
- /sys:/host/sys:ro
- /:/rootfs:ro

command:

- '--path.procfs=/host/proc'
- '--path.sysfs=/host/sys'
- '--collector.filesystem.mount-points-exclude'
-

'^(/sys|/proc|/dev|/host|/etc|/rootfs/var/lib/docker/containers|/rootfs/var/lib/docker/overlay2|/rootfs/run/docker/netns|/rootfs/var/lib/docker/aufs)(\$\$|/)'

volumes:

grafana-data:

grafana-configs:

prom-data:

prom-configs:

```
GNU nano 6.2 docker-compose.yaml *
version: "3.9"
services:

  grafana:
    image: grafana/grafana:8.5.3-ubuntu
    ports:
      - "3000:3000"
    volumes:
      - grafana-data:/var/lib/grafana
      - grafana-configs:/etc/grafana

  prometheus:
    image: prom/prometheus:v2.36.0
    ports:
      - "9090:9090"
    volumes:
      - prom-data:/prometheus
      - prom-configs:/etc/prometheus

  node-exporter:
    image: prom/node-exporter:v1.3.1
    ports:
      - "9100:9100"
    volumes:
      - /proc:/host/proc:ro
```

Запускаем стек приложений в docker-compose:

```
$ docker-compose up
```

```

filipp@filipp-notebook:~$ docker-compose up
Creating network "filipp_default" with the default driver
Creating volume "filipp_grafana-data" with default driver
Creating volume "filipp_grafana-configs" with default driver
Creating volume "filipp_prom-data" with default driver
Creating volume "filipp_prom-configs" with default driver
Creating filipp_node-exporter_1 ... done
Creating filipp_prometheus_1 ... done
Creating filipp_grafana_1 ... done
Attaching to filipp_prometheus_1, filipp_node-exporter_1, filipp_grafana_1
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:491 level=info
prometheus_1 | r size retention was set so using the default time retention" duration=15d
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:535 level=info
Prometheus Server" mode=server version="(version=2.36.0, branch=HEAD, revisi
68c83283ce5233844807dfdc5ba5)"
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:540 level=info
"(go=go1.18.2, user=root@b3126bd1c115, date=20220530-13:56:56)"
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:541 level=info
(Linux 5.19.0-35-generic #36~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Feb 17 1
86_64 15b855b6edf8 (none))"
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:542 level=info
ft=1048576, hard=1048576)"
prometheus_1 | ts=2023-03-23T13:06:20.684Z caller=main.go:543 level=info
ft=unlimited, hard=unlimited)"
prometheus_1 | ts=2023-03-23T13:06:20.687Z caller=web.go:553 level=info

```

Выполним команду просмотра запущенных контейнеров, чтобы убедиться в успешном запуске следующих компонентов стека:

- *prometheus*
- *grafana*
- *node-exporter*

```
$ docker ps
```

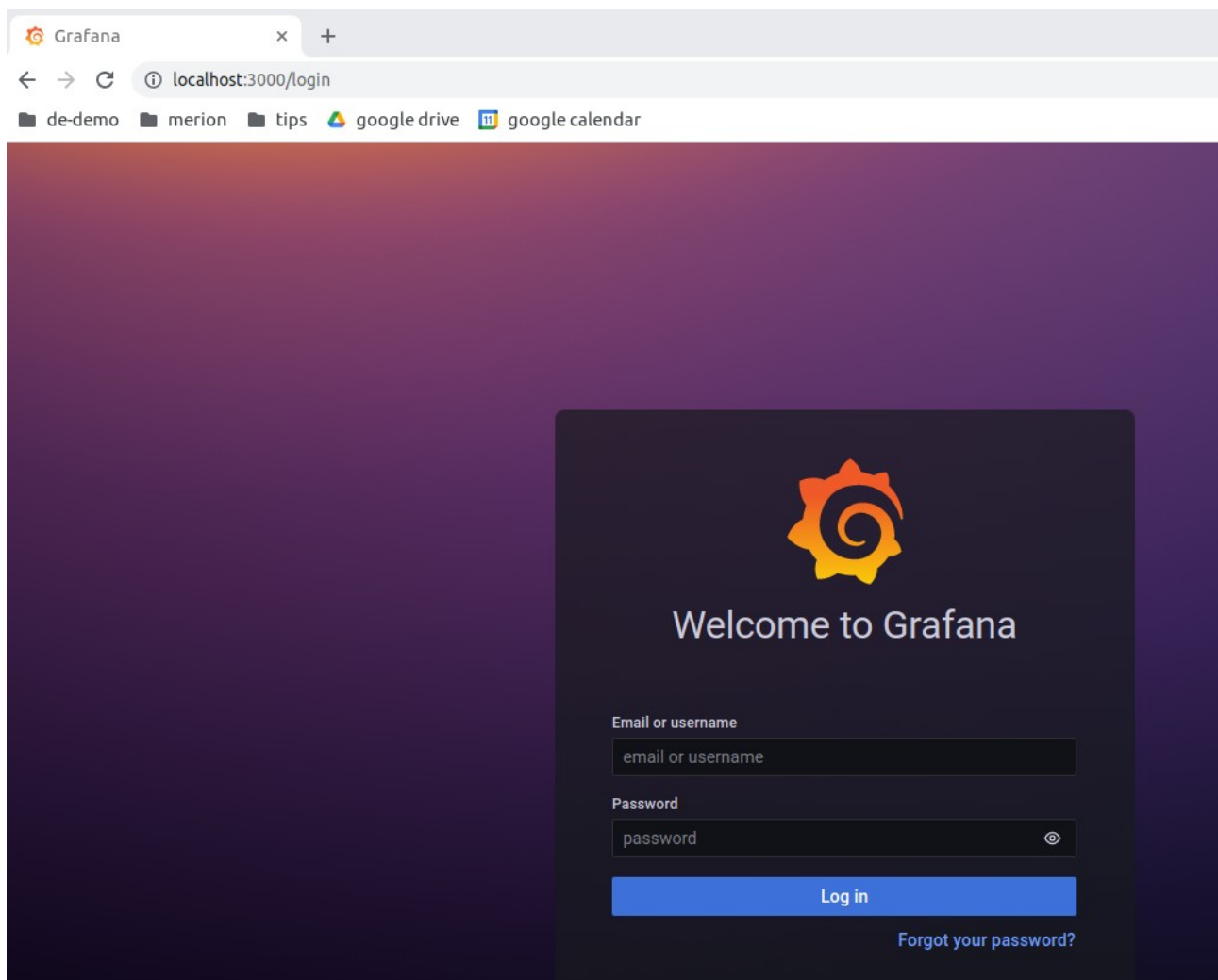
```

filipp@filipp-notebook:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS                               NAMES
55675aff2bea   grafana/grafana:8.5.3-ubuntu        "/run.sh"               About a minute ago
Up About a minute   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   filipp_grafana_1
15b855b6edf8   prom/prometheus:v2.36.0             "/bin/prometheus --c..." About a minute ago
Up About a minute   0.0.0.0:9090->9090/tcp, :::9090->9090/tcp   filipp_prometheus_1
6f76f720a9c6   prom/node-exporter:v1.3.1           "/bin/node_exporter ..." About a minute ago
Up About a minute   0.0.0.0:9100->9100/tcp, :::9100->9100/tcp   filipp_node-exporter_1

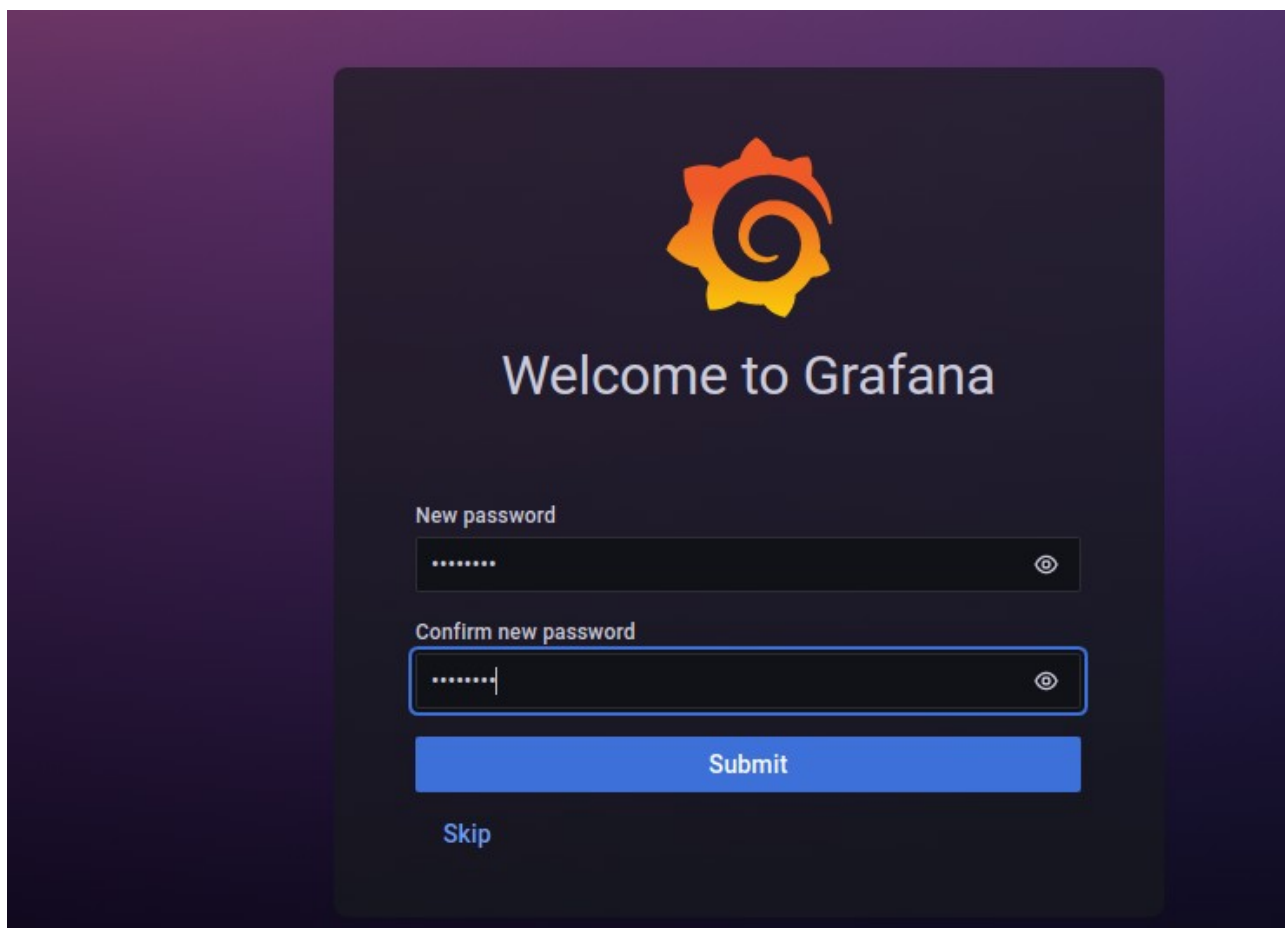
```

Обратите внимание, что веб-интерфейс grafana использует порт 3000 локального хоста. Переместимся в браузер и перейдем по следующему адресу:

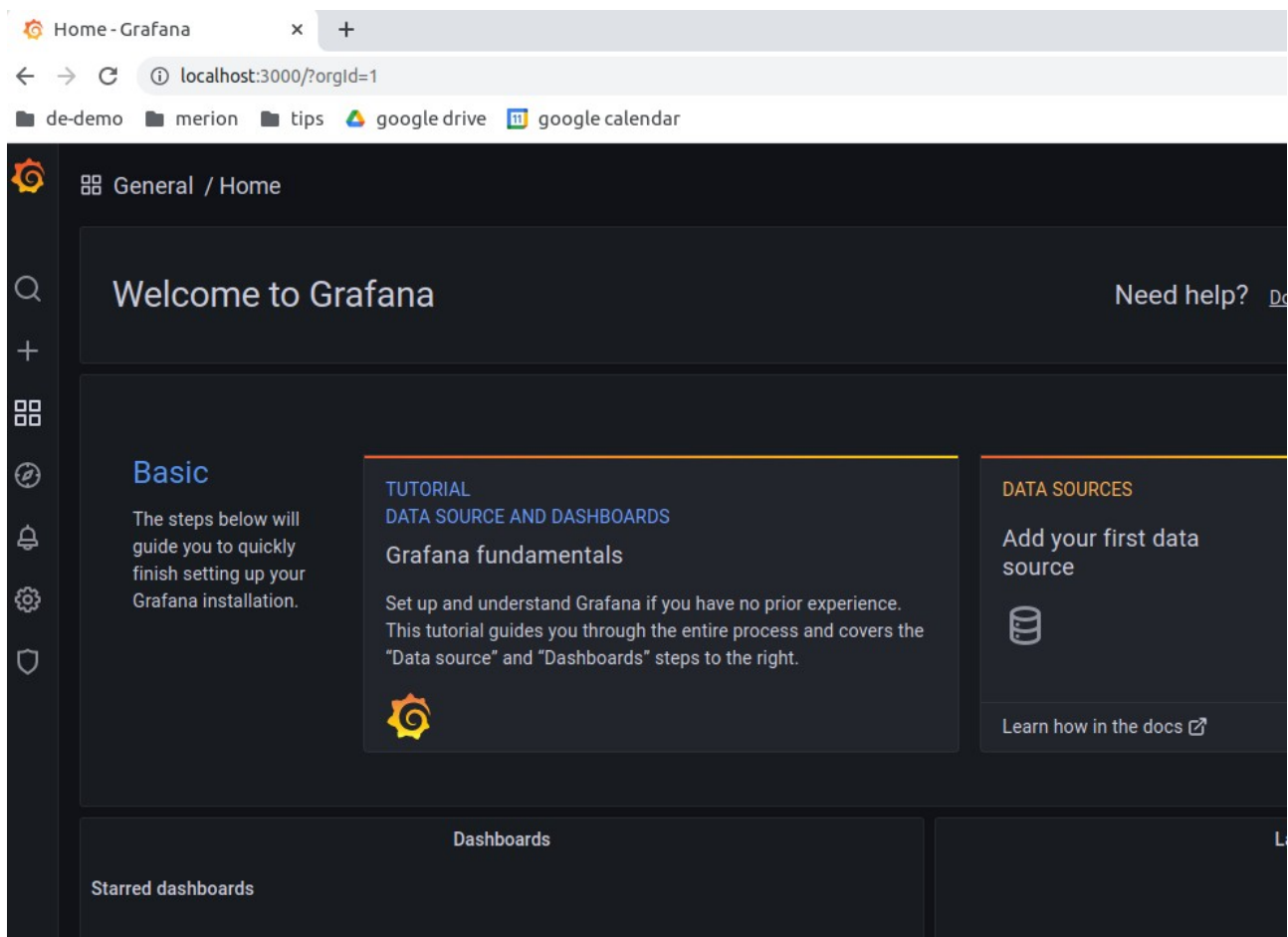
browser: <http://localhost:3000/>



Введем учетные данные `admin:admin` и выполним смену пароля для пользователя `admin`:



Авторизация успешно завершена:

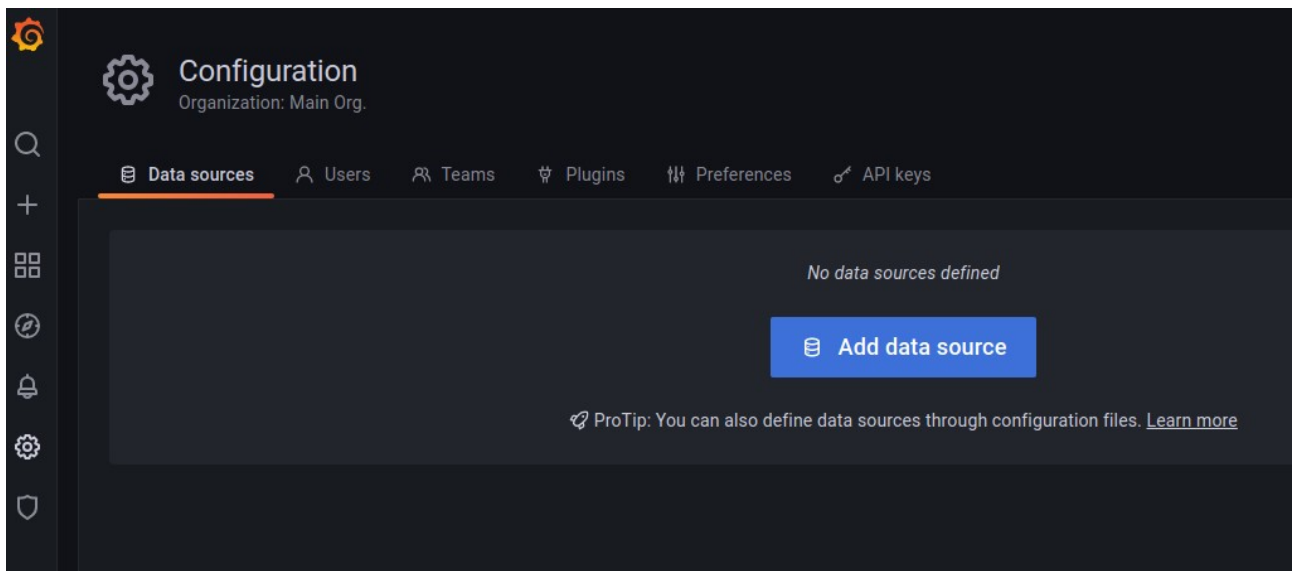


Веб-интерфейс в рабочем состоянии

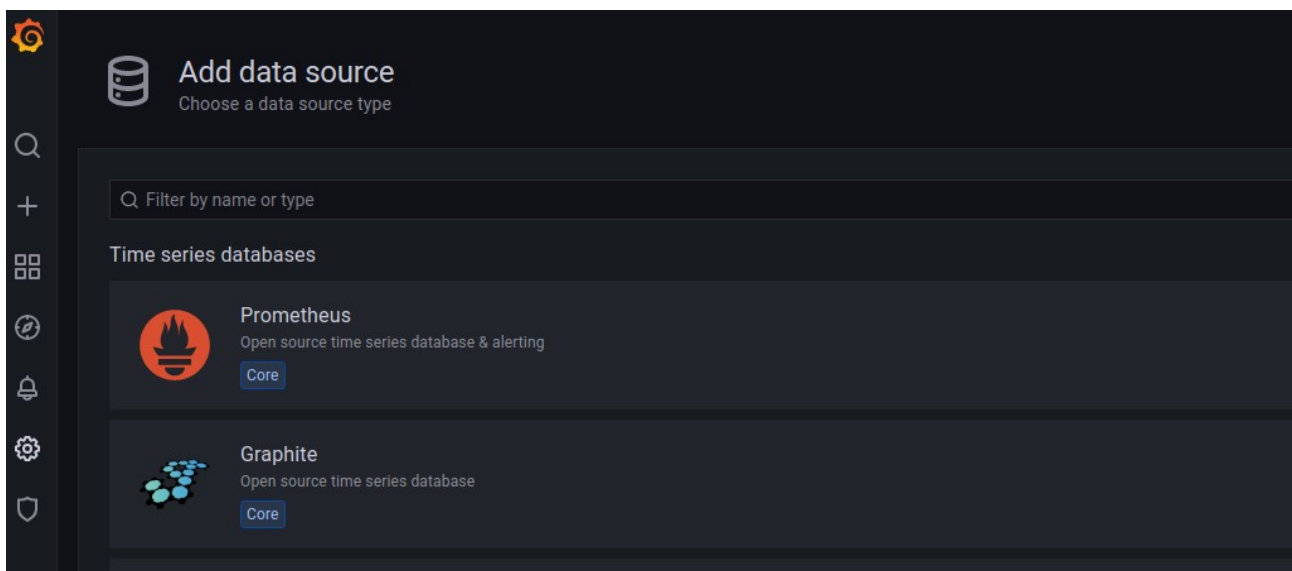
--

Часть 2. Настройка подключения prometheus и grafana для осуществления сбора метрик

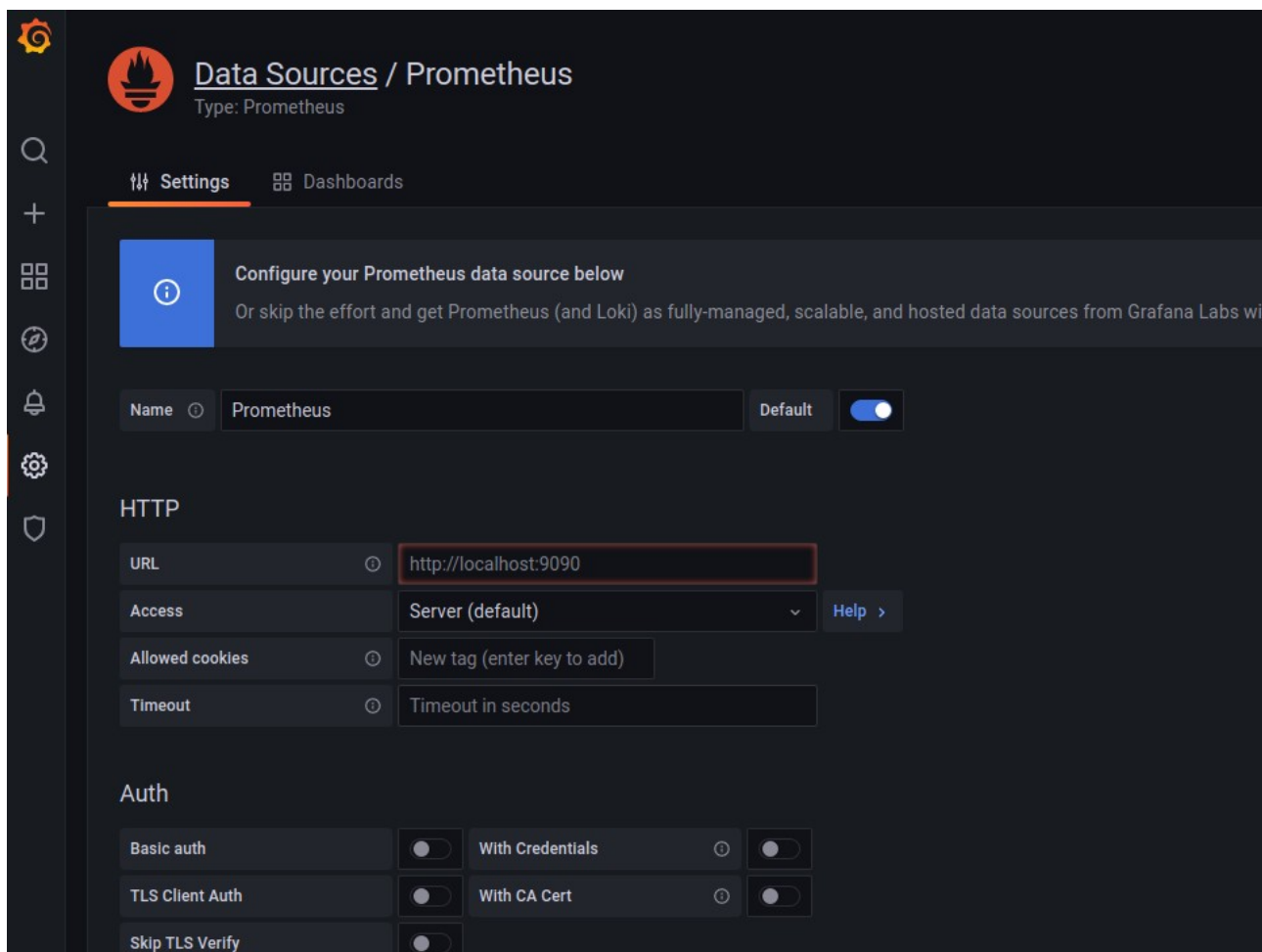
Находясь в веб-интерфейсе grafana, переходим в раздел "Configuration -> Data Sources":



Добавим новый источник данных, нажатием кнопки "Add data source":



Выбираем тип источника "Prometheus"



Чтобы обеспечить сетевую связанность между prometheus и grafana, воспользуемся командой инспектирования сети нашего стека приложений в docker. Сперва запросим список имеющихся сетей:

```
$ docker network ls
```

```
filipp@filipp-notebook:~$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
3271e21b707b        bridge              bridge              local
b311038ddd47        filipp_default      bridge              local
34a7c0b58f04        host                host                local
4ed5072f636c        none                null                local
```

Выбираем нужную сеть (в моем случае это filipp_default, ваше название сети будет отличаться):

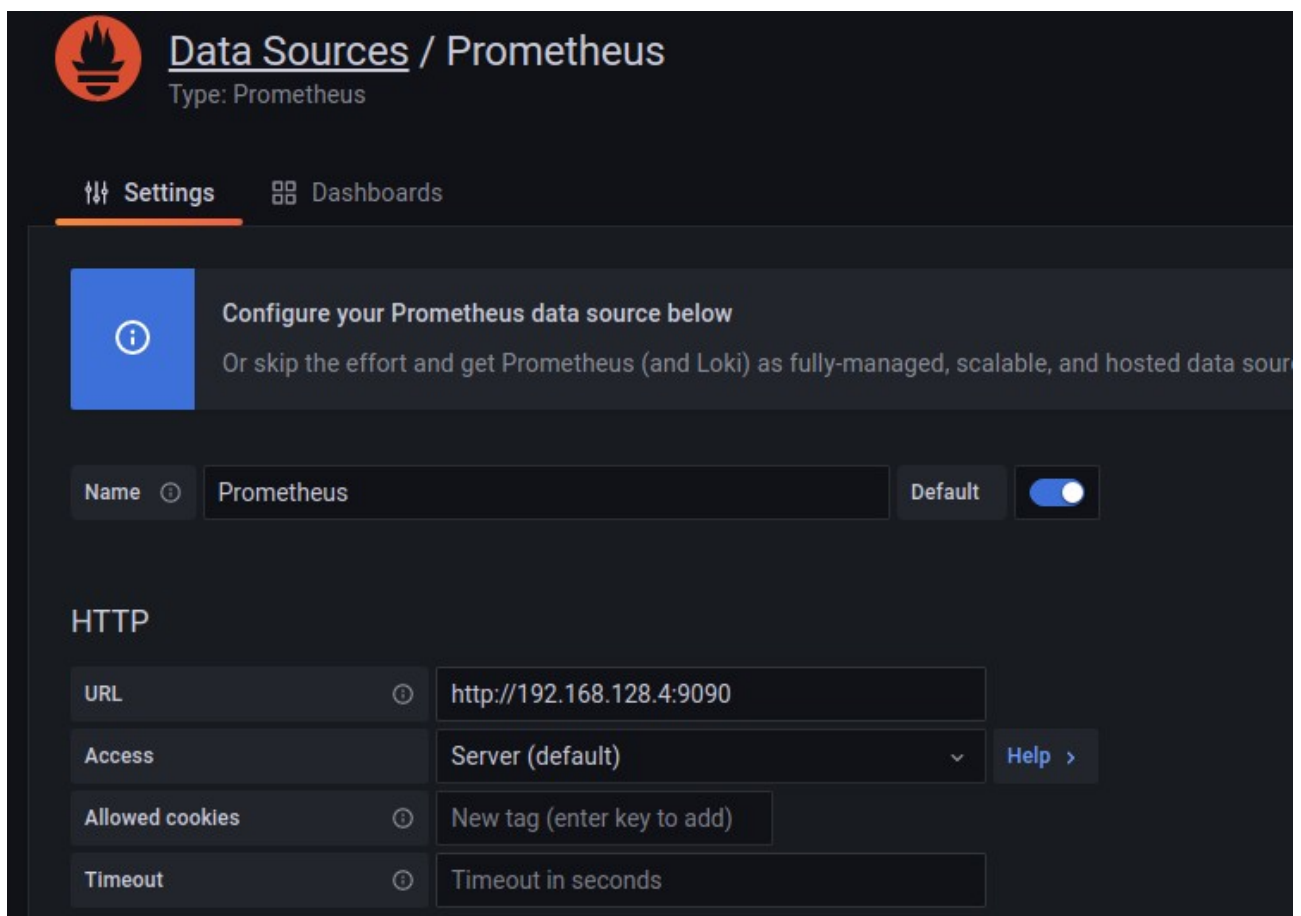
```
$ docker network inspect filipp_default
```


```

    "Containers": {
      "15b855b6edf8fb06caef781d5f2167606bb1fcfee79dc21f42f7fb27aa3bb68d": {
        "Name": "filipp_prometheus_1",
        "EndpointID": "67de21b5f5d26e2a7cde60b24906db2dee14c65a41005c8ad1ecdc6c8b499ac5",
        "MacAddress": "02:42:c0:a8:80:04",
        "IPv4Address": "192.168.128.4/20",
        "IPv6Address": ""
      },
      "55675aff2bea701a79ec8903e61a36a3b461aa43cbc308381880156d1d1f022a": {
        "Name": "filipp_grafana_1",
        "EndpointID": "971b470a33ce5e24b2a32bbba2e76fc9c3232cdd2c62dc3ed918f8795c281062",
        "MacAddress": "02:42:c0:a8:80:03",
        "IPv4Address": "192.168.128.3/20",
        "IPv6Address": ""
      },
      "6f76f720a9c62e9c48f31871df8768b643b6628c1b84fc4f08ab78908cd3168c": {
        "Name": "filipp_node-exporter_1",
        "EndpointID": "60cc1cac510aa621cfd0610fd6f15e5d9063d76469e63852bca47248252f1368",
        "MacAddress": "02:42:c0:a8:80:02",
        "IPv4Address": "192.168.128.2/20",
        "IPv6Address": ""
      }
    }
  }
}

```

Извлекаем `IPv4Address` контейнера с `prometheus` и подставляем это значение в параметр настройки `data source` в `grafana`, добавив в `url` порт `9090`:



 **Data Sources / Prometheus**
Type: Prometheus

Settings | Dashboards

Configure your Prometheus data source below
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data source

Name ⓘ Prometheus Default ☒

HTTP

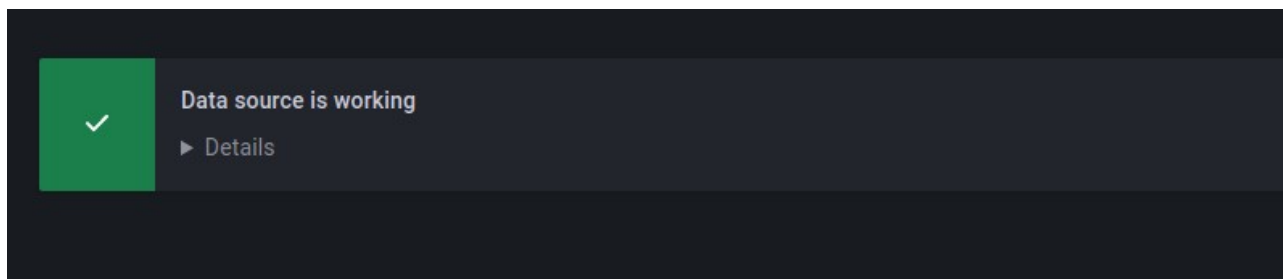
URL ⓘ http://192.168.128.4:9090

Access Server (default) Help >

Allowed cookies ⓘ New tag (enter key to add)

Timeout ⓘ Timeout in seconds

Сохраняем изменения и проверяем подключение нажатием кнопки "Save & test":

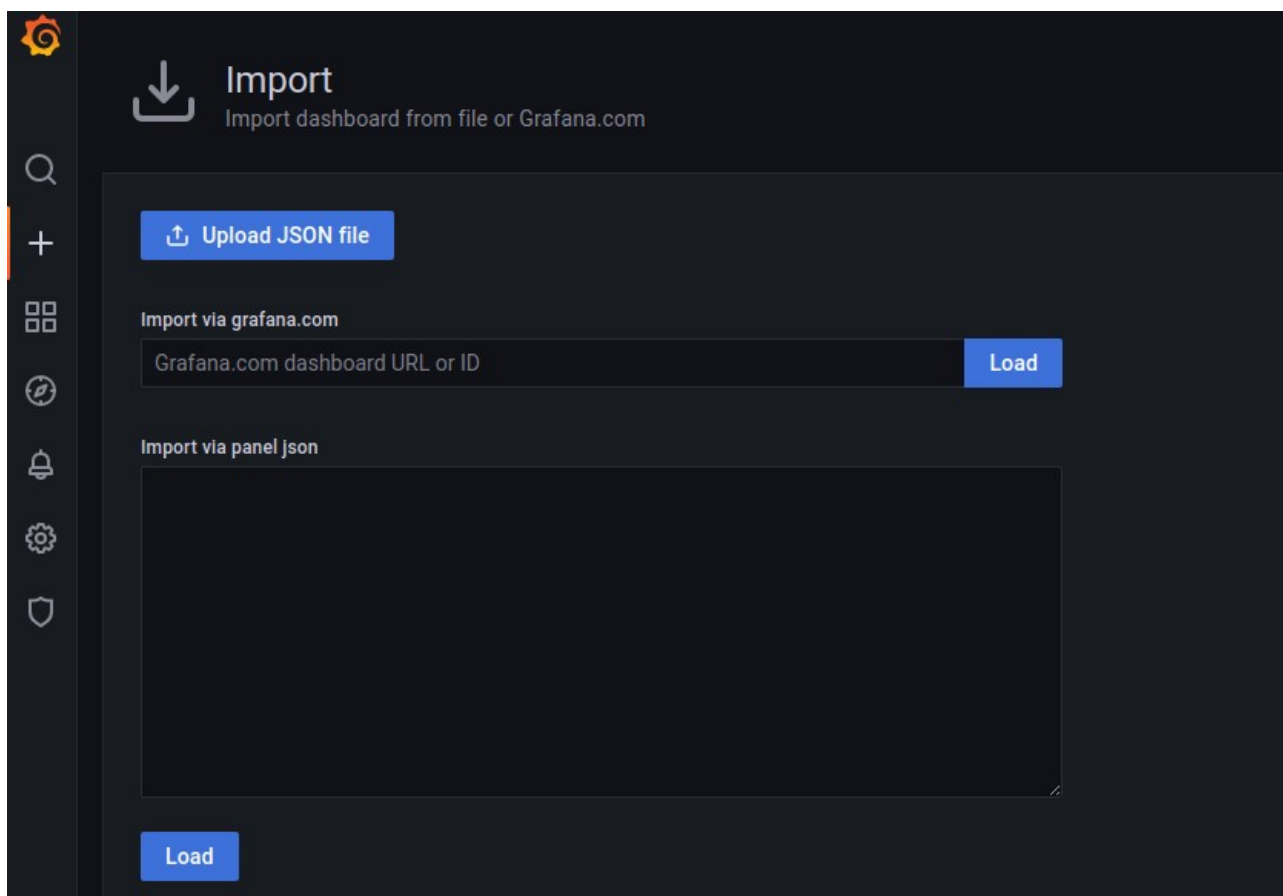


Подключение успешно настроено

--

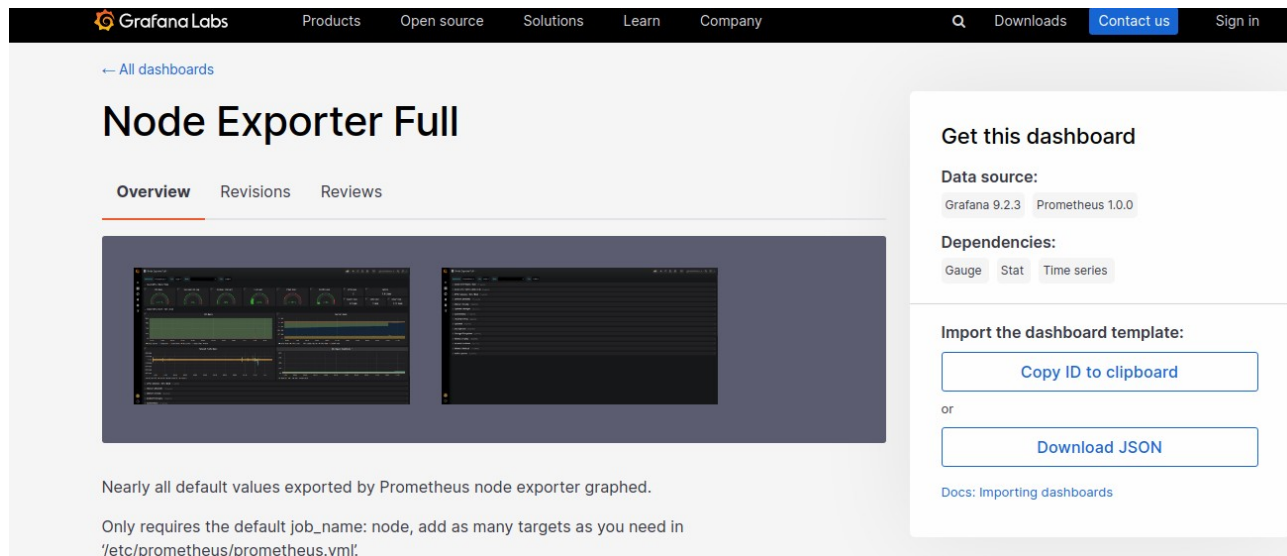
Часть 3. Конфигурация dashboard в grafana

В веб-интерфейсе grafana переходим в раздел "Create -> Import"



За основу мы возьмем json дашборда, подготовленный самими разработчиками grafana. перейдем по следующему адресу в браузере:

browser: `https://grafana.com/grafana/dashboards/1860-node-exporter-full/`



Нажмем кнопку "Download JSON":



Контент из скачанного json-файла копируем и вставляем в интерфейс импорта дашборда в grafana:



Import

Import dashboard from file or Grafana.com

Upload JSON file

Import via grafana.com

Grafana.com dashboard URL or ID


Load

Import via panel json

```
"1h",
"2h",
"1d"
],
"time_options": [
  "5m",
  "15m",
  "1h",
  "6h",
  "12h",
  "24h",
  "2d",
  "7d",
  "30d"
]
},
"timezone": "browser",
"title": "Node Exporter Full",
"uid": "rYdddIPWk",
"version": 9,
"weekStart": ""
}
```

Load

Сохраняем изменения нажатием клавиши "Load". В опциях настройки импорта указываем добавленный нами ранее источник данных Prometheus и нажимаем клавишу "Import":

 **Import**
Import dashboard from file or Grafana.com

Options

Name

Folder


General

Unique Identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

rYdddIPWk

Change uid

prometheus

 Prometheus

Import Cancel

--

Часть 4. Конфигурация prometheus для сбора метрик с node-exporter

Снова воспользуемся командой инспектирования сети, используемой стеком, и извлекаем `IPv4Address` контейнера с `node-exporter`:

```
$ docker network inspect filipp_default
```

```

},
"ConfigOnly": false,
"Containers": {
  "15b855b6edf8fb06caef781d5f2167606bb1fcfee79dc21f42f7fb27aa3bb68d": {
    "Name": "filipp_prometheus_1",
    "EndpointID": "67de21b5f5d26e2a7cde60b24906db2dee14c65a41005c8ad1ecdc6c8b499ac5",
    "MacAddress": "02:42:c0:a8:80:04",
    "IPv4Address": "192.168.128.4/20",
    "IPv6Address": ""
  },
  "55675aff2bea701a79ec8903e61a36a3b461aa43cbc308381880156d1d1f022a": {
    "Name": "filipp_grafana_1",
    "EndpointID": "971b470a33ce5e24b2a32bbba2e76fc9c3232cdd2c62dc3ed918f8795c281062",
    "MacAddress": "02:42:c0:a8:80:03",
    "IPv4Address": "192.168.128.3/20",
    "IPv6Address": ""
  },
  "6f76f720a9c62e9c48f31871df8768b643b6628c1b84fc4f08ab78908cd3168c": {
    "Name": "filipp_node-exporter_1",
    "EndpointID": "60cc1cac510aa621cfd0610fd6f15e5d9063d76469e63852bca47248252f1368",
    "MacAddress": "02:42:c0:a8:80:02",
    "IPv4Address": "192.168.128.2/20",
    "IPv6Address": ""
  }
}

```

Изменим файл конфигурации prometheus на хосте, где он запущен (указывайте тот volume контейнера, который используется у вас, наименование проекта docker-compose у вас будет отличаться, что ведет к изменению наименования volume для контейнера):

```

$ sudo nano
/var/lib/docker/volumes/filipp_prom-configs/_data/prometheus.yml

```



```

GNU nano 6.2 /var/lib/docker/volumes/filipp_prom-configs/_data/prometheus.yml
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

```

Изменяем секцию `scrap_configs` и сохраняем изменения:

```

job_name: node_exporter
target: <nodeexporter-ip>:9100

```

```

scrape_configs:
  # The job name is added as a label `job=<job_name>` to c
  - job_name: "node_exporter"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["192.168.128.2:9100"]

```

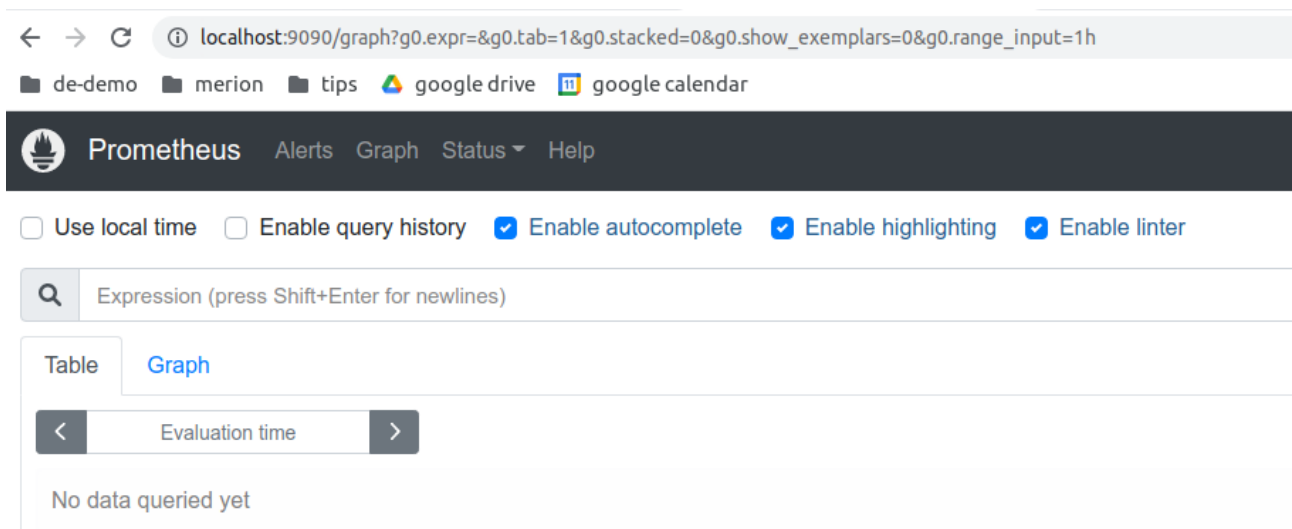
Посылаем контейнеру сигнал выполнить перезагрузку (reload) изменившегося файла конфигурации (SIGHUP от сочетания слов SIGNAL - сигнал, и HANG UP - обрыв связи). При этом фактически перезагрузка контейнера prometheus произведена не будет:


```
docker kill --signal=SIGHUP filipp_prometheus_1
```

```
filipp@filipp-notebook:~$ docker kill --signal=SIGHUP filipp_prometheus_1
filipp_prometheus_1
```

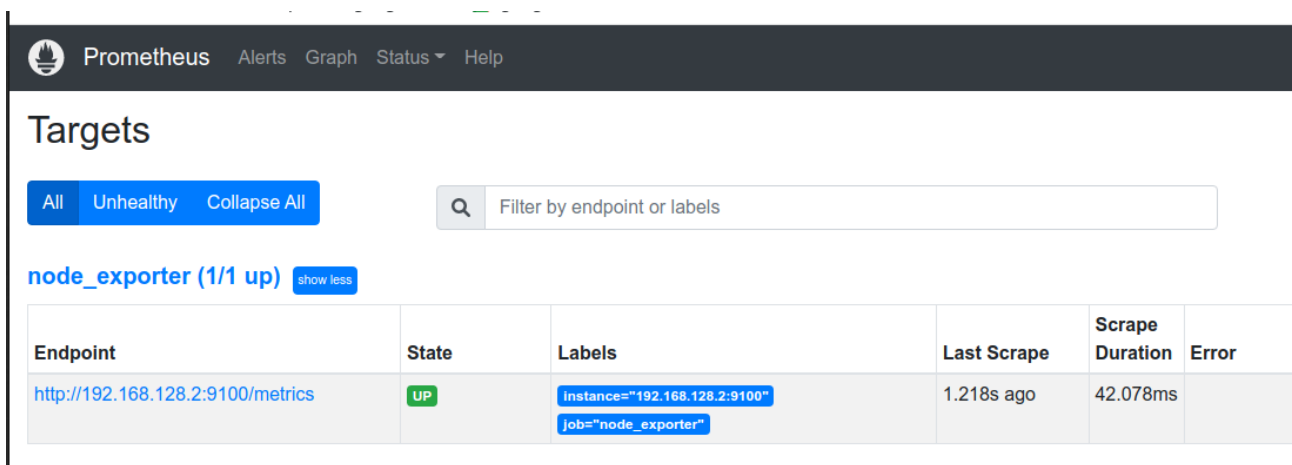
Перейдем в веб-интерфейс prometheus и убедимся в корректности добавленного target (сборщика метрик, в нашем случае - node_exporter):

browser: <http://localhost:9090/>



The screenshot shows the Prometheus web interface at localhost:9090/. The top navigation bar includes 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below the navigation bar, there are checkboxes for 'Use local time', 'Enable query history', 'Enable autocomplete', 'Enable highlighting', and 'Enable linter'. A search bar is present with the placeholder text 'Expression (press Shift+Enter for newlines)'. The 'Graph' tab is selected, and the 'Evaluation time' is shown. The main content area displays 'No data queried yet'.

Переходим в раздел "Status -> Targets" и убеждаемся, что target node_exporter успешно добавлен и обладает статусом "UP":

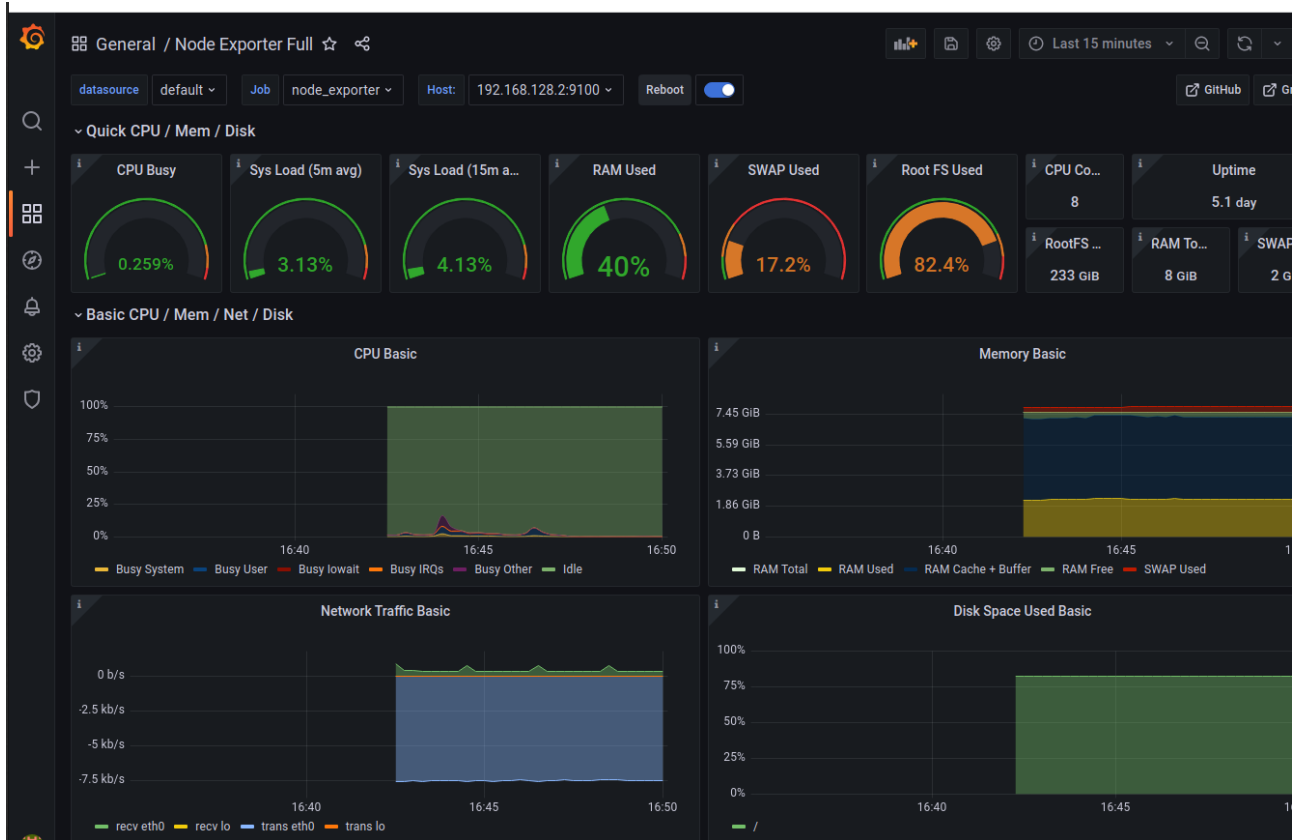


The screenshot shows the Prometheus web interface at the 'Targets' page. The top navigation bar includes 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. The 'Targets' section has a filter bar with 'All', 'Unhealthy', and 'Collapse All' buttons, and a search bar with the placeholder text 'Filter by endpoint or labels'. Below the filter bar, the text 'node_exporter (1/1 up)' is displayed. A table shows the target details:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.128.2:9100/metrics	UP	instance="192.168.128.2:9100" job="node_exporter"	1.218s ago	42.078ms	

Часть 5. Визуализация метрик

Возвращаемся в веб-интерфейс grafana, перезагружаем дашборд и изменяем интервал показываемых метрик на "Last 15 minutes":



Мониторинг настроен успешно

--

Для завершения работы в окне терминала с запущенным docker-compose проектом выполните нажатие клавиш CTRL+C и команду "docker-compose down":

```
^CGracefully stopping... (press Ctrl+C again to force)
Stopping filipp_grafana_1      ... done
Stopping filipp_prometheus_1  ... done
Stopping filipp_node-exporter_1 ... done
filipp@filipp-notebook:~$ docker-compose down
Removing filipp_grafana_1      ... done
Removing filipp_prometheus_1  ... done
Removing filipp_node-exporter_1 ... done
Removing network filipp_default
```

--