*Порядок запуска: 1. InfluxDB 2. Telegraf 3. Grafana или 1. Prometheus 2. Grafana*

**Установка Telegraf, InfluxDB и Grafana на Windows**

**1. Установка Telegraf**

Telegraf – это агент для сбора метрик и отправки их в базы данных (например, InfluxDB). Вот шаги по его установке на Windows:

1. **Скачивание Telegraf**:
   * Перейдите на сайт [Telegraf](https://www.influxdata.com/time-series-platform/telegraf/) (прямая ссылка: https://dl.influxdata.com/telegraf/releases/telegraf-1.32.0\_windows\_amd64.zip) и найдите раздел загрузки для Windows.
   * Выберите версию для Windows (обычно это .zip файл).
   * Скачайте и распакуйте архив в удобную директорию, например, C:\Telegraf.
2. **Настройка конфигурации Telegraf** 
   1. **для InfluxDB v1** :
   * После распаковки в папке будет файл telegraf.conf. Это основной конфигурационный файл Telegraf, где настраиваются плагины для сбора данных и отправки метрик в базы данных (например, InfluxDB).
   * Откройте файл в текстовом редакторе (например, Notepad++) и отредактируйте его. Для отправки данных в InfluxDB v1, найдите секцию [[outputs.influxdb]] и внесите соответствующие данные:

toml

Копировать код

[[outputs.influxdb]]

urls = ["http://localhost:8086"] # Адрес сервера InfluxDB

database = "telegraf" # Имя базы данных

### 2.2 Настройка конфигурации Telegraf для InfluxDB v2:

* После распаковки в папке будет файл telegraf.conf. Это основной конфигурационный файл Telegraf, где настраиваются плагины для сбора данных и отправки метрик в базы данных (например, InfluxDB).
* Откройте файл в текстовом редакторе (например, Notepad++) и отредактируйте его. Для отправки данных в InfluxDB v2, найдите секцию [[outputs.influxdb\_v2]] и внесите соответствующие данные:

[[outputs.influxdb\_v2]]

urls = ["http://localhost:8086"] # Адрес сервера InfluxDB

token = "your-token" # Ваш токен авторизации

organization = "your-org" # Имя вашей организации в InfluxDB

bucket = "telegraf" # Имя вашего bucket в InfluxDB v2

**Пояснения:**

* **urls**: адрес вашего сервера InfluxDB, по умолчанию это http://localhost:8086.
* **token**: это токен авторизации, который вы получите в интерфейсе InfluxDB при создании пользователя или проекта. Он заменяет логин и пароль.
* **organization**: имя вашей организации в InfluxDB v2, которую вы задали при установке.
* **bucket**: в InfluxDB v2 вместо базы данных используется **bucket**, указывайте здесь соответствующий bucket, куда будут отправляться метрики.

1. **Запуск Telegraf**:
   * Откройте командную строку и перейдите в директорию, где находится Telegraf.

cd C:\Telegraf

* + Запустите Telegraf с конфигурационным файлом:

telegraf.exe --config telegraf.conf

* + Убедитесь, что Telegraf успешно запущен и отправляет данные в базу данных.
  + Каталог плагинов для телеграфа <https://docs.influxdata.com/telegraf/v1/plugins/>

**2. Установка InfluxDB**

InfluxDB – это СУБД для временных рядов, которая часто используется для хранения метрик, собранных с помощью Telegraf.

1. **Скачивание InfluxDB**:
   * Перейдите на сайт [InfluxDB Downloads](https://www.influxdata.com/downloads/) (прямая ссылка: <https://download.influxdata.com/influxdb/releases/influxdb2-client-2.7.5-windows-amd64.zip>) и выберите версию для Windows.
   * Скачайте .zip файл и распакуйте его, например, в папку C:\InfluxDB.
2. **Запуск InfluxDB**:
   * Откройте командную строку и перейдите в директорию, куда вы распаковали InfluxDB.

cd C:\InfluxDB

* + Запустите сервер InfluxDB с помощью команды:

influxd.exe

* + InfluxDB будет работать на порту 8086 по умолчанию.

1. **Создание базы данных для Telegraf**:

* перейти в <http://127.0.0.1:8086> и настроить пользователя, пароль, организацию, бакет **ЗАПИСАТЬ данные себе, они нужны для связки с телеграфом и графаной**

**У меня admin – admin123 – PerformanceLab – telegraf (пример)**

* Скопировать ТОКЕН для связки себе в файл
* Мой токен(пример):

8\_qtkykkxXJFgfnJHItu0s3--p0UIkH2SBTLgNiNpe7o0I1oUO5NbVO1Dz2LfYv2itTMX9YTG7iihNCnt8PC\_w==

* Идем в директорию телеграфа и меняем конфиг outputs.influxdb\_v2, прописать указанные выше: urls:порт, пароль, организацию, бакет
* Остановить, если запущен телегрф старый через админа повер шел : C:\Program Files\InfluxData\telegraf> net stop telegraf

**3. Установка Grafana (все ресурсы графаны доступны только с ВПН)**

Grafana – это платформа для визуализации данных, которая поддерживает подключение к таким источникам, как InfluxDB.

1. **Скачивание Grafana**:
   * Перейдите на [страницу загрузки Grafana](https://grafana.com/grafana/download?platform=windows) и выберите версию для Windows.
   * Распаковать архив, у меня 20 мин заняло
2. **Установка Grafana**:
   * Перейти в директорию графаны и запустить через **PowerShell с правами админа:**

& "C:\Program Files\Test\_Telegraf\_Influx\_Grafana\grafana-enterprise-11.2.0.windows-amd64\grafana-v11.2.0\bin\grafana.exe" server --homepath "C:\Program Files\Test\_Telegraf\_Influx\_Grafana\grafana-enterprise-11.2.0.windows-amd64\grafana-v11.2.0"

Либо самим перейти в директорию и выполнить .\grafana.exe server c правами **АДМИНА**!

C:\Program Files\Test\_Telegraf\_Influx\_Grafana\grafana-enterprise-11.2.0.windows-amd64\grafana-v11.2.0\bin> .\grafana.exe server

1. **Запуск Grafana**:
   * По завершении установки откройте браузер и перейдите по адресу http://localhost:3000. Это интерфейс Grafana.
   * Введите логин и пароль по умолчанию:
     + Логин: admin
     + Пароль: admin
   * После первого входа вам будет предложено сменить пароль.

В defaults.ini в папке config можно посмотреть основные конфиги графаны, в том числе логин и пароль:

admin\_user = admin

admin\_password = admin

1. **Добавление InfluxDB в Grafana**:
   * В интерфейсе Grafana перейдите в раздел **Home-Connections-Data sources-Add data source**
   * Нажмите кнопку **Add Data Source** и выберите InfluxDB.
   * Заполните следующие поля:
   * Query language выбирать для инфлакса 2 версии+ Flux для 1 версии InfluxQL
     + **URL**: http://localhost:8086
     + **Database**: telegraf
     + **User**: если вы настраивали пользователя InfluxDB, укажите его. Если нет, оставьте пустым.
   * Нажмите **Save & Test**, чтобы проверить подключение к базе данных.
   * **ЧАСТО Ошибки** возникают при неправильной вставке токина или пароля, попробуйте вставить токен в ячейку рядом, что бы увидеть токен, а затем уже в нужную.
2. **Создание панели визуализации**:
   * Перейдите в раздел **Create -> Dashboard**.
   * Нажмите **Add Panel**, выберите нужные метрики из Telegraf и создайте графики.
   * Или нажать импорт и найти необходимый id\json на сайте <https://grafana.com/grafana/dashboards/>
   * Id для Jmetr и Influx 2+версии – 17506
   * Id для Windows и Influx 2.\*+версии – 18081
   * Id Linux – 17191 для Influx 2.\*+версии
   * Id Linux – 14731 для Prometheus+NoudExporter
   * Id Linux – 10180 для Prometheus+NoudExporter
   * Id Linux – 7675 для Prometheus+NoudExporter

УСТАНОВКА **Telegraf** на **Linux**

Чтобы установить Telegraf на Ubuntu 24, следуйте шагам ниже:

**1. Добавление репозитория InfluxData**

Чтобы установить Telegraf через официальный репозиторий InfluxData, вам нужно сначала добавить его.

1. Откройте терминал.
2. Обновите индекс пакетов:

sudo apt update

1. Установите пакеты для добавления репозиториев:

sudo apt install -y wget gnupg2 curl

1. Добавьте ключ репозитория:

curl -s https://repos.influxdata.com/influxdata-archive\_compat.key | sudo apt-key add -

1. Добавьте репозиторий для Ubuntu 24:

echo "deb https://repos.influxdata.com/ubuntu focal stable" | sudo tee /etc/apt/sources.list.d/influxdata.list

**2. Установка Telegraf**

Теперь можно установить Telegraf.

1. Обновите индекс пакетов снова:

sudo apt update

1. Установите Telegraf:

sudo apt install -y telegraf

**3. Настройка и запуск Telegraf**

1. После установки Telegraf, вы можете настроить его, отредактировав конфигурационный файл:

sudo nano /etc/telegraf/telegraf.conf

1. Чтобы запустить Telegraf и добавить его в автозагрузку, используйте systemctl:

sudo systemctl start telegraf

sudo systemctl enable telegraf

1. Для проверки статуса службы:

sudo systemctl status telegraf

**4. Тестирование Telegraf**

Чтобы проверить работоспособность Telegraf, вы можете запустить его в режиме тестирования:

telegraf --config /etc/telegraf/telegraf.conf --test

**5. Обновление Telegraf**

Если в будущем вам потребуется обновить Telegraf, просто выполните:

sudo apt update

sudo apt upgrade telegraf

Теперь Telegraf должен быть установлен и готов к использованию на Ubuntu 24.

*При запуске телеграфа на виртуалке (убунту) не видел ip основной машины (винды), помогло:*

*Выключить антивирус*

Установка **Prometheus** на Linux

**Обновление системы**

Как всегда, начнем с обновления системы:

sudo apt update

sudo apt upgrade -y

**Загрузка и установка Prometheus**

Сначала загрузите и распакуйте Prometheus:

cd /tmp

curl -LO https://github.com/prometheus/prometheus/releases/download/v2.46.0/prometheus-2.46.0.linux-amd64.tar.gz

tar -xvf prometheus-\*.tar.gz

cd prometheus-2.46.0.linux-amd64/

выполнить команду напрямую из папки для ручного запуска:

./prometheus --config.file prometheus.yml \

--storage.tsdb.path ./data \

--web.console.templates=./consoles \

--web.console.libraries=./console\_libraries

После запуска Prometheus будет доступен по адресу http://<IP\_адрес\_вашей\_машины>:9090.

(вот здесь будет прометеус /tmp/prometheus-2.46.0.linux-amd64)

**Остановка Prometheus**

Чтобы остановить Prometheus, просто прервите выполнение программы, нажав Ctrl+C в терминале, где был запущен Prometheus.

Таким образом, Prometheus не будет запускаться автоматически при перезагрузке, и вы сможете запускать его вручную по мере необходимости.

Установка **Node Exporter** на **сервер** (нужен для более полного сбора метрик в **Prometheus**)

Перейти в директорию \tmp

* **Загрузите Node Exporter**:

https://github.com/prometheus/node\_exporter/releases/download/v1.6.1/node\_exporter-1.6.1.linux-amd64.tar.gz

* **Распакуйте архив**:

tar xvfz node\_exporter-1.6.1.linux-amd64.tar.gz

* **Запустите Node Exporter**:

cd node\_exporter-1.6.1.linux-amd64

./node\_exporter

Node Exporter будет работать на порту 9100 по умолчанию пока не закроете консоль, либо же сделать службу для автозапуска..

### Настройка Prometheus для сбора метрик с Node Exporter

1. Откройте конфигурационный файл Prometheus (prometheus.yml). Обычно он находится в папке, где вы установили Prometheus, например: /etc/prometheus/prometheus.yml.
2. Добавьте в секцию scrape\_configs новый блок для Node Exporter:

scrape\_configs:

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['localhost:9100']

Если Node Exporter установлен на другом сервере, замените localhost:9100 на IP-адрес этого сервера, например: 192.168.0.105:9100.

1. Сохраните изменения и перезапустите Prometheus для применения новой конфигурации:

sudo systemctl restart prometheus

### Проверка работы

1. Откройте веб-интерфейс Prometheus в браузере:

http://<IP-адрес сервера с Prometheus>:9090

1. Перейдите в раздел **"Status"** → **"Targets"**. Вы должны увидеть, что Node Exporter отображается как активный таргет.
2. Перейдите в раздел **"Graph"** и выполните запросы на метрики, которые предоставляет Node Exporter. Например:
   * node\_cpu\_seconds\_total
   * node\_memory\_MemAvailable\_bytes
   * node\_filesystem\_size\_bytes

### Дополнительно:

* **Автозапуск Node Exporter**: Если вы хотите, чтобы Node Exporter автоматически запускался при перезагрузке, создайте systemd unit:
  1. Создайте файл /etc/systemd/system/node\_exporter.service:

sudo nano /etc/systemd/system/node\_exporter.service

* 1. Добавьте в него следующее содержимое:

[Unit]

Description=Node Exporter

After=network.target

[Service]

User=root

ExecStart=/path/to/node\_exporter

[Install]

WantedBy=default.target

* 1. Сохраните файл и активируйте сервис:

sudo systemctl daemon-reload

sudo systemctl enable node\_exporter

sudo systemctl start node\_exporter

Теперь Prometheus будет автоматически собирать метрики с вашего сервера через Node Exporter.

Основные команды для работы с **Telegraf**, **InfluxDB** и **Grafana** в ТЕРМИНАЛЕ (**Linux**).

**Telegraf**

1. **Запуск Telegraf:**

telegraf --config /etc/telegraf/telegraf.conf

Запуск Telegraf с указанным конфигурационным файлом.

1. **Проверка конфигурации Telegraf:**

telegraf --config /etc/telegraf/telegraf.conf --test

Проверка конфигурации Telegraf, выводит данные в терминал для тестирования.

1. **Запуск как службы (systemd):**

sudo systemctl start telegraf

1. **Остановка Telegraf:**

sudo systemctl stop telegraf

1. **Просмотр статуса Telegraf:**

sudo systemctl status telegraf

**InfluxDB (InfluxDB 2.x)**

1. **Запуск InfluxDB:**

sudo systemctl start influxdb

1. **Остановка InfluxDB:**

sudo systemctl stop influxdb

1. **Просмотр статуса InfluxDB:**

sudo systemctl status influxdb

1. **CLI InfluxDB:**

influx

Запуск командной строки для работы с InfluxDB.

1. **Показать базы данных:**

influx bucket list

1. **Запись данных в базу:**

influx write --bucket 'bucket\_name' --org 'org\_name' --file data.txt

1. **Запрос данных:**

influx query 'from(bucket: "bucket\_name") |> range(start: -1h)'

**Grafana**

1. **Запуск Grafana:**

sudo systemctl start grafana-server

1. **Остановка Grafana:**

sudo systemctl stop grafana-server

1. **Просмотр статуса Grafana:**

sudo systemctl status grafana-server

1. **Перезапуск Grafana:**

sudo systemctl restart grafana-server

1. **Обновление конфигурации Grafana (например, если внесли изменения в конфиг):**

sudo systemctl reload grafana-server

Эти команды помогут управлять Telegraf, InfluxDB и Grafana в терминале для мониторинга и сбора метрик.

**Prometheus** в терминале:

**1. Запуск Prometheus**

./prometheus --config.file=prometheus.yml

Основная команда для запуска Prometheus с указанием файла конфигурации.

**2. Просмотр помощи и доступных флагов**

./prometheus --help

Выводит справку с доступными флагами и опциями.

**3. Флаги запуска Prometheus**

Вот некоторые важные флаги:

* **: Указание пути к файлу конфигурации (по умолчанию**--config.fileprometheus.yml).

./prometheus --config.file=/path/to/config.yml

* **--storage.tsdb.path**: указание пути для хранения данных.

./prometheus --storage.tsdb.path=/path/to/data

* **--web.listen-address**: указание адреса и порта для HTTP-сервера Prometheus.

./prometheus --web.listen-address=":9090"

* **--storage.tsdb.retention.time**: время хранения данных (по умолчанию 15 дней).

./prometheus --storage.tsdb.retention.time=30d

* **--query.max-concurrency**: устанавливает максимальное количество параллельных запросов.

./prometheus --query.max-concurrency=20

**4. Перезагрузка конфигурации**

Prometheus поддерживает перезагрузку конфигурации без перезапуска. Для этого отправляется HTTP-запрос на эндпоинт /-/reload:

curl -X POST http://localhost:9090/-/reload

Это полезно, если вы внесли изменения в конфигурацию и хотите применить их без остановки сервиса.

**5. Просмотр метрик Prometheus**

Чтобы просмотреть метрики самого Prometheus, можно отправить запрос через браузер или curl на эндпоинт /metrics:

curl http://localhost:9090/metrics

**6. Остановка Prometheus**

Чтобы остановить сервер Prometheus, выполните команду:

kill <PID>

Где <PID> - это идентификатор процесса (можно получить с помощью ps aux | grep prometheus).

**7. Просмотр версии Prometheus**

Для того чтобы узнать версию Prometheus, выполните:

./prometheus --version

**8. Запуск консольного интерфейса**

После того как Prometheus запущен, доступ к его веб-интерфейсу осуществляется через браузер по умолчанию по адресу:

http://localhost:9090

**Примеры создания своего дажборда под Windows (вместо «telegraf» указывать свой бакет):**

1. Простой запрос для получения данных из определённого измерения

from(bucket: "telegraf")

  |> range(start: -1h)  // за последний час

  |> filter(fn: (r) => r.\_measurement == "cpu")

  |> filter(fn: (r) => r.\_field == "usage\_user")

Этот запрос извлекает данные из измерения cpu за последний час, фильтруя по полю usage\_user.

2. Агрегация данных (сумма, среднее)

Для вычисления средней нагрузки на CPU за последний час (после написания запроса нужно выбрать график, не таблицу):

from(bucket: "telegraf")

  |> range(start: -1h)

  |> filter(fn: (r) => r.\_measurement == "cpu" and r.\_field == "usage\_user")

  |> mean()

Этот запрос извлекает среднее значение поля usage\_user за последний час.

3. Использование диска (Disk Usage)

Запрос для получения информации о проценте использования файловой системы:

from(bucket: "telegraf")

  |> range(start: -1h)

  |> filter(fn: (r) => r.\_measurement == "disk" and r.\_field == "used\_percent")

  |> group(columns: ["device"])

  |> mean()

Этот запрос возвращает средний процент использования диска для каждого устройства за последний час.

3. Общая загрузка системы

Запрос для отображения общей загрузки системы по ключевым метрикам: процессор, память и диск:

cpu = from(bucket: "telegraf")

  |> range(start: -1h)

  |> filter(fn: (r) => r.\_measurement == "cpu" and r.\_field == "usage\_system")

  |> mean()

mem = from(bucket: "telegraf")

  |> range(start: -1h)

  |> filter(fn: (r) => r.\_measurement == "mem" and r.\_field == "used\_percent")

  |> mean()

disk = from(bucket: "telegraf")

  |> range(start: -1h)

  |> filter(fn: (r) => r.\_measurement == "disk" and r.\_field == "used\_percent")

  |> mean()

union(tables: [cpu, mem, disk])

Этот запрос объединяет метрики процессора, памяти и использования диска для отображения общей загрузки системы за последний час.

Дополнительно

Чтобы понять, как создавать информационные панели в системах мониторинга, таких как **Grafana**, необходимо выполнить несколько шагов.

### 1. ****Понимание источников данных****

* **Шаг 1:** Определите, с какой системой сбора данных вы работаете. Например, это может быть **InfluxDB**, **Prometheus**, **Elasticsearch** или другая база данных, поддерживаемая Grafana.
* **Шаг 2:** Изучите структуру данных в источнике данных. В каждой системе есть свои показатели и таблицы:
  + **InfluxDB:** Метрики организованы в «измерения» (**measurement**), поля (**fields**) и теги (**tags**).
  + **Prometheus:** Используются метрики, которые имеют метки (**labels**), фильтруемые в запросах.
* **Шаг 3:** Изучите данные, которые собирает ваш источник данных. Например, для систем мониторинга это могут быть:
  + Метрики CPU, памяти, сети, I/O.
  + Метрики приложений (Java, PostgreSQL, Kafka).
  + Логи и события (например, с Nginx, Apache).

### 2. ****Исследование метрик и структуры данных****

* **Шаг 4:** Используйте консоль или интерфейс вашей базы данных для анализа доступных данных:
  + **InfluxDB**: Воспользуйтесь Flux-запросами, чтобы просмотреть доступные измерения (SHOW MEASUREMENTS), поля (SHOW FIELD KEYS) и теги (SHOW TAG KEYS).
  + **Prometheus**: Используйте PromQL для изучения доступных метрик (выполните запрос типа up или node\_cpu\_seconds\_total).
* **Шаг 5:** Попробуйте выполнить базовые запросы в графической панели Grafana:
  + Для **InfluxDB**: запросы могут выглядеть как:

from(bucket: "telegraf")

|> range(start: -1h)

|> filter(fn: (r) => r.\_measurement == "cpu")

* + Для **Prometheus**: простой запрос:

node\_cpu\_seconds\_total{mode="idle"}

### 3. ****Создание дашборда: практический подход****

* **Шаг 6:** Открой Grafana и перейди в раздел **Dashboards** > **Create Dashboard**.
* **Шаг 7:** Добавьте новую панель (панель может быть графиком, таблицей, датчиком и т. д.).
* **Шаг 8:** Укажите источник данных (InfluxDB, Prometheus или другой).
* **Шаг 9:** Выполни запрос к метрикам:
  + Для **InfluxDB**: выберите «bucket» и настройте параметры запроса (измерение, поле, тег).
  + Для **Prometheus**: напиши PromQL-запрос, отфильтровав метрики, например, по определённым лейблам.

### 4. ****Фильтрация данных и использование переменных****

* **Шаг 10:** Для более гибкого дашборда, можно добавить **переменные**:
  + В разделе **Settings** > **Variables** добавь переменные для фильтрации по времени, хостам, инстансам или другим параметрам (например, имя сервера, регион).
  + Пример переменной для InfluxDB:

import "influxdata/influxdb/v1"

v1.tagValues(bucket: "telegraf", tag: "\_host")

* + Переменные позволят динамически изменять данные на дашборде, не изменяя запросы.

### 5. ****Настройка отображения и визуализации****

* **Шаг 11:** Выбери тип визуализации, который лучше всего подходит для данных:
  + **Графики**: для отображения изменения метрики во времени.
  + **Gauge или Singlestat**: для отображения текущего значения метрики.
  + **Таблицы**: для детального анализа данных.
* **Шаг 12:** Настрой параметры отображения, такие как оси, подписи, форматирование чисел, фильтры и пороги (thresholds), которые помогут выделить критические значения.

### 6. ****Анализ и улучшение****

* **Шаг 13:** После создания дашборда просматривайте данные, анализируйте их поведение и настраивайте пороговые значения для оповещений (alerts).
* **Шаг 14:** Постепенно улучшай панель управления, добавляя больше показателей и переменных, чтобы сделать ее более информативной.

### Примеры запросов для разных систем:

#### Пример для InfluxDB:

from(bucket: "telegraf")

|> range(start: -1h)

|> filter(fn: (r) => r.\_measurement == "mem")

|> filter(fn: (r) => r.\_field == "used\_percent")

#### Пример для Prometheus:

rate(node\_network\_receive\_bytes\_total{device="eth0"}[5m])

#### Пример использования переменной:

* Переменная для выбора хоста:

import "influxdata/influxdb/v1"

v1.tagValues(bucket: "telegraf", tag: "host")

### 7. ****Документация и изучение примеров****

* **Документация Grafana:** изучите официальную документацию Grafana, чтобы понять все доступные опции.
* **Примеры панелей мониторинга:** используйте готовые панели мониторинга, доступные в Grafana (включая **библиотеку панелей мониторинга Grafana**), чтобы понять, как можно создавать запросы и визуализировать данные.

Этот подход даст вам полное представление о том, как самостоятельно создавать информационные панели, начиная с понимания структуры данных и заканчивая их визуализацией.