

Вариант установки Telegraf+Influx+Grafana на win10 с помощью cmd.exe
(localhost; с расположением telegraf.conf на сервере)

1) скачиваем InfluxDB v2xx и Telegraf (тут: <https://portal.influxdata.com/downloads/>)

- для каждой из программ выбираем справа в окошке "Platform" соответственно Windows
- далее снизу на фиолетовом фоне после - "wget" - появляются ссылки для скачивания zip архива (над надписью Documentation) (кстати на следующей строке сразу команды для установки через командную строку если мало ли)

2) Grafana (тут - <https://grafana.com/grafana/download?platform=windows>)

- аналогично жмём - [Download the zip file](#)

3) Создаём три папки в Program Files (да пишут что для винды важно поместить сюда их - мой путь C:\Program Files) - Telegraf, InfluxData, Grafana

- распаковываем в них соответственно скачанные архивы
- если создались дополнительные папки после разархивирования можно безболезненно перенести те файлы выше, в наши папки - так будет проще обращаться к ним из командной строки

4) Запускаем по очереди в обратном порядке - Telegraf будет последним (*не критично*)(проверяем):

Grafana

- открываем нашу папку в Program Files - далее папку - "bin"
- запускаем просто как программу - grafana-server.exe (не закрываем потом появившееся окошко, а сворачиваем)
- затем так же запускал grafana-cli.exe
- проверяем как запустилась Grafana - набираем в браузере такое - <http://localhost:3000>
- открывается Grafana и просит ввести логин пароль - по умолчанию это admin/admin
- далее сразу появиться окно для смены логина\пароля - если нужно делаем
- затем приложение открывается и оставляем его пока в покое

InfluxDB

- запускаем от имени администратора (отныне и далее(!)) - cmd.exe (да в некоторых руководствах пишут про PowerShell, но лично у меня он капризничает)
- пишем - chdir c:\program files\InfluxData
- затем - influxd.exe
- немного ждём и смотрим на то что в командной строке появилась магия - сворачиваем

- проверяем как запустился influx - набираем в браузере такое - <http://localhost:8086>
- открывается Influx UI, которые попросит пройти регистрацию (лучше записать те данные которые будут введены при регистрации, тк программа требовательна к безопасности)
- оставляем в покое пока что :)

Telegraf

- запускаем еще одну cmd.exe
- пишем - chdir c:\program files\telegraf
- затем - telegraf.exe --service install
- скорее всего если вы уже с ним возились может выдать что то вроде, что телеграф уже добавлен в списки сервисов (можно поискать его в списках процессов - команда - tasklist) или запуститься с выдачей ошибок при попытке соединиться с базой - это окей
- закрываем эту cmd.exe просто

- 5) Подружим их между собой теперь (объединим в одну систему - ведь по сути Телеграф собирает метрики - передает в базу, - а оттуда Графана их успешно визуализирует)

Telegraf+Influx

- возвращаемся к Influx UI - выбираем слева - Data - а затем уже сверху Telegraf
- далее справа вверху - +Create Configuration, затем выбираем System и continue
- вводим имя, клацаем - Create and Verify
- копируем и сохраняем данные из полей "API token" и "Start Telegraf"
- нажимаем Listen for Data - всё ок - Finish
- но не финиш никакой на самом деле
- двойной клик на имя созданного конфига и видим окно Edit Telegraf Configuration
- всё что после bucket = "*****" (у меня 70 строка), меняем смело с 71 строки на представленные ниже (просто ctrl c - ctrl v) - (по умолчанию там вроде ubuntu метрики, которые Инфлюкс снимает в принципе без проблем с винды, но всё же Телеграф сердится и пишет, что не поддерживаю):

```
[[inputs.win_perf_counters]]
[[inputs.win_perf_counters.object]]
# Processor usage, alternative to native, reports on a per core.
ObjectName = "Processor"
Instances = ["*"]
Counters = [
    "% Idle Time",
    "% Interrupt Time",
    "% Privileged Time",
    "% User Time",
    "% Processor Time"
]
Measurement = "win_cpu"
# Set to true to include _Total instance when querying for all (*).
#IncludeTotal=false
```

```
[[inputs.win_perf_counters.object]]
# Disk times and queues
ObjectName = "LogicalDisk"
Instances = ["*"]
Counters = [
    "% Idle Time",
    "% Disk Time",
    "% Disk Read Time",
    "% Disk Write Time",
    "% User Time",
    "% Free Space",
    "Current Disk Queue Length",
    "Free Megabytes",
    "Disk Read Bytes/sec",
    "Disk Write Bytes/sec"
]
Measurement = "win_disk"
# Set to true to include _Total instance when querying for all (*).
#IncludeTotal=false
```

```
[[inputs.win_perf_counters.object]]
ObjectName = "System"
Counters = [
    "Context Switches/sec",
    "System Calls/sec",
    "Processor Queue Length",
    "Threads",
    "System Up Time",
    "Processes"
]
Instances = ["-----"]
Measurement = "win_system"
# Set to true to include _Total instance when querying for all (*).
#IncludeTotal=false
```

```
[[inputs.win_perf_counters.object]]
# Example query where the Instance portion must be removed to get data
back,
# such as from the Memory object.
ObjectName = "Memory"
Counters = [
    "Available Bytes",
    "Cache Faults/sec",
    "Demand Zero Faults/sec",
    "Page Faults/sec",
    "Pages/sec",
    "Transition Faults/sec",
    "Pool Nonpaged Bytes",
    "Pool Paged Bytes"
]
# Use 6 x - to remove the Instance bit from the query.
```

```

Instances = ["-----"]
Measurement = "win_mem"
# Set to true to include _Total instance when querying for all (*).
#IncludeTotal=false

[[inputs.win_perf_counters.object]]
# more counters for the Network Interface Object can be found at
# https://msdn.microsoft.com/en-us/library/ms803962.aspx
ObjectName = "Network Interface"
Counters = [
    "Bytes Received/sec",
    "Bytes Sent/sec",
    "Packets Received/sec",
    "Packets Sent/sec"
]
Instances = ["*"] # Use * - to remove the Instance bit from the query.
Measurement = "win_net"
#IncludeTotal=false #Set to true to include _Total instance when querying
for all (*).

[[inputs.win_perf_counters.object]]
# Process metrics
ObjectName = "Process"
Counters = [
    "% Processor Time",
    "Handle Count",
    "Private Bytes",
    "Thread Count",
    "Virtual Bytes",
    "Working Set"
]
Instances = ["*"]
Measurement = "win_proc"
#IncludeTotal=false #Set to true to include _Total instance when querying
for all (*).

```

- далее Save Changes и yes yes

немного действий сразу для Grafana

- (на всякий случай создаём Scraper) вверху открываем вкладку Scrapers
→ Create Scraper
- вводим имя, выбираем наш именной bucket - Create
- далее следующая вверху вкладка Tokens
- тут можно управлять ими - сделаем сразу токен дающий право на всё
- +Generate Token
- All Access Token
- набираем имя и Save
- двойной клик по нему (по имени) - видим и сохраняем этот “супер токен” отдельно
- а плюс к нему нам понадобится вот ещё что - снова в одной из верхних вкладок выбираем - Buckets

- видим, что напротив нашего именного bucket есть номер её (id) - наводим мышь туда - появится copy to clipboard - копируем и сохраняем рядом с супер токеном (а заодно и с самим именем)

Проверяем наш Телеграф

- открываем снова cmd.exe (от имени администратора конечно)
- пишем - `chdir c:\program files\telegraf`
- смотрим на наш первый простой токен - что нам записалось - что то вроде - `export INFLUX_TOKEN=*****ваш токен***`
- меняем слово `export` на `set` (особенности синтаксиса cmd)
- и важно - в конце токена добавляем пробел
- пишем в командную строку - `set INFLUX_TOKEN=*****ваш простой токен***` (с пробелом в конце) (пример: `set INFLUX_TOKEN=axlxxdL1hD0ca_efmu-VApM-_VF5Fc87PLbY8CTpEc5lhVMGnmTYTFyFZLf cWptAvKGpiDAGcVnAtIQ_7lx9nw==`) - Enter
- далее в командную строку - вставляем команду которую записали для API token и нажимаем Enter
- Телеграф запускается и не бухтит
- сворачиваем это окно cmd
- в идеале в Influx (на вкладке слева Explore) можно будет наблюдать появление данных

А теперь Grafana+Influx :)

- возвращаемся на нашу вкладку в браузере на порте 3000
- слева в меню значок шестерёнки - Configuration
- выбираем Data sources потом Add Data Source -> InfluxDB -> Select
- заполняем нужные поля (остальные оставляем как есть):
- можно задать имя
- далее Query Language - InfluxQL (в принципе аналогично и на Flux но данные мне не льёт)
- в разделе "HTTP" - URL -> <http://localhost:8086/>
- в разделе "Custom HTTP Headers" жмём +Add Header
- в поле Header заносим слово - Authorization
- в поле Value - слово Token потом пробел и номер/значение вашего токена (если не запускаться можно супер токен)
- в разделе "InfluxDB Details"
- в поле Database вносим имя базы (на данном этапе у меня оно совпадает с именем Bucket - единственного именного)
- Http Method - Get
- Min time interval - 10s
- жмем Save & Test
- видим такое - Data source is working

Идём далее - ещё немного

- Добавим снятие метрик с уже настроенными графиками
- слева в меню есть плюс (+) - нажимаем и выбираем сразу - Import

- в поле Import via Grafana.com добавляем - <https://grafana.com/grafana/dashboards/1902> - далее Load
- открывается панель с метриками, но их вероятно нет, хотя вроде всё ок
- а дело тут ещё видимо вот в чём - Телеграф льёт данные в вёдра(те самые bucket) Инфлюкса, который Графана нашла, но нашла с Базой Данных, которой нет :)
- Нам надо смаппить (to map) наш Bucket в БД (подробнее [тут](#))
- открываем ещё одну cmd.exe
- вводим - chdir c:\program files\InfluxData
- затем - influx.exe
- выйдет список из команд
- и нам надо ввести будет такую длинную строку (понадобятся параметры:
 - *db* - имя нашей БД(которое было указано при настройке графаны (в разделе "InfluxDB Details"), как и писал в моём случае совпадает с именем bucket)
 - *rp* - что то для Influx (название связанного файла - оставил просто example)
 - *bucket-id* - тот самый id который мы записали вместе с супер токеном
 - *org* - имя организации которое было указано при регистрации в Influx
 - *token* - супер токен)

Вид запроса в общем виде:

```
influx v1 dbrp create --db **имяБД** --rp **имяфайларп** --bucket-id **idВедра** --default --org
**название организации** --token **супер токен**
```

Вид запроса в моём случае:

```
influx v1 dbrp create --db adminfoxdb --rp exemplerp --bucket-id 99622feeb20b9c4e --default --org
adminfoxgroup --token
axlxxdL1hD0ca_efmu-VApM-_VF3Fc95PLbY9CTpEc4lhVMGnmTYTFyFZLfcWptAvKGpiDAGcVnAt
IQ_7lx9nw==
```

В ответ пришло:

ID	Database	Bucket ID	Retention Policy	Default Organization	ID
07dc402fb1672000	adminfoxdb	99622feeb20b9c4e	exemplerp	true	119d4e4d3bed2189

- сворачиваем командную строку
- разворачиваем Графану
- смотрим как оживают графики
- пьём чай :)