Prometheus

Приложение для мониторинга событий и оповещений — **Prometheus**;

Оно собирает метрики от exporter.

Пример:

https://www.youtube.com/watch?v=X\_g-eJqiiLo

https://git.digitalstudium.com/digitalstudium/grafana-docker-stack

docker-compose.yml

version: "3.9"

services:

**grafana**:

image: grafana/grafana:8.5.3-ubuntu

# image: grafana/grafana:**latest**

ports:

- "**3000:3000**"

volumes:

- grafana-data:/var/lib/grafana

- grafana-configs:/etc/grafana

**prometheus**:

image: prom/prometheus:v2.36.0

# image: prom/prometheus:**latest**

ports:

- "**9090:9090**"

volumes:

- prom-data:/prometheus

- prom-configs:/etc/prometheus

**node-exporter**:

image: prom/node-exporter:v1.3.1

# image: prom/node-exporter:**latest**

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:

docker-compose up -d

Открываем **prometheus:**

<http://192.168.18.5:9090/>

## Добавить node-exporter в prometheus

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

Каталог **monitoring\_prom-configs может называться иначе**

добавить IP наблюдаемых хостов. Внимательно с отступами

- job\_name: "pc2"

static\_configs:

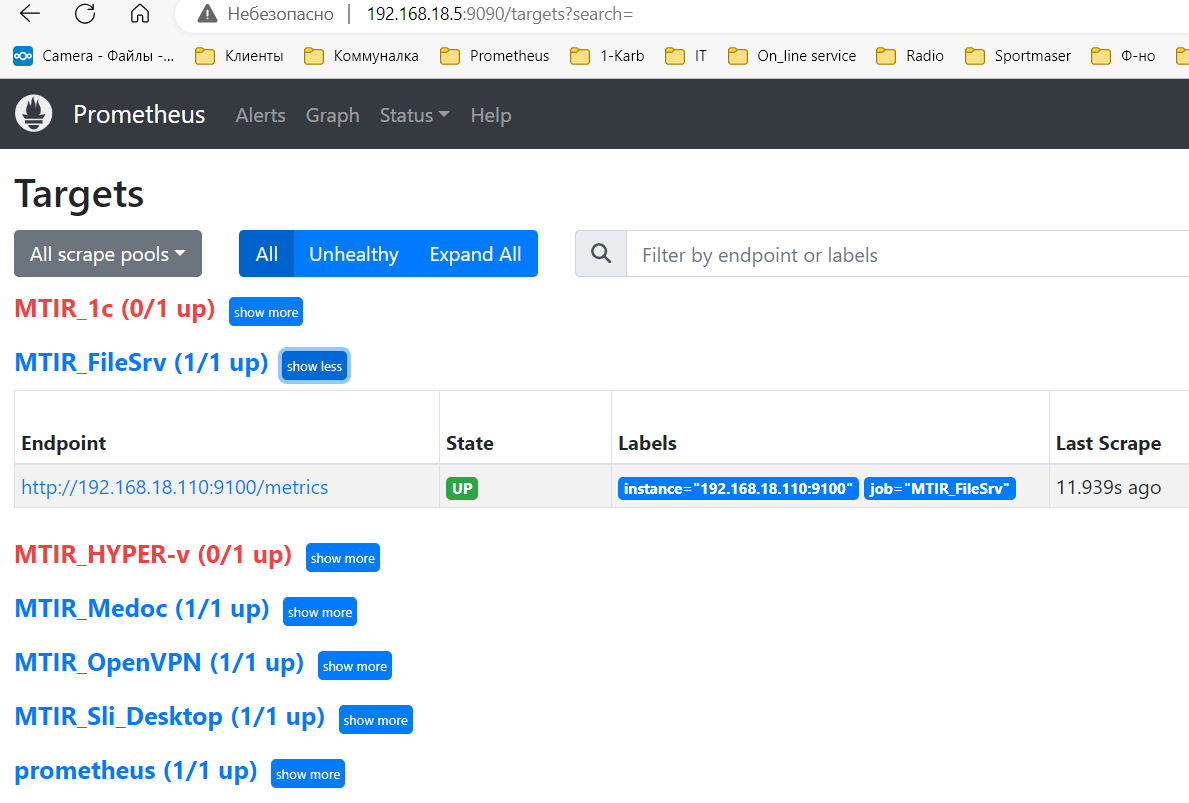
- targets: ["172.16.181.221:9100"]

- job\_name: "pc1"

static\_configs:

- targets: ["172.16.181.220:9100"]

Кликнув по вкладке Status -> Targets, можно увидеть, с каких нод сейчас идёт сбор метрик



# Пример запроса для Windows (Windows exporter)

Определить свободное место на диске C: (GB)

windows\_logical\_disk\_free\_bytes{volume="C:"}/1024/1024/1024

windows\_logical\_disk\_free\_bytes{volume="D:"}/1024/1024/1024

Количество ОЗУ в системе (GB)

windows\_cs\_physical\_memory\_bytes/1024/1024/1024

Сколько ОЗУ свободно

windows\_os\_physical\_memory\_free\_bytes/1024/1024/1024

Версия

windows\_os\_info

# Пример запроса для Linux (**node-exporter)**

### кол-во оперативной памяти

Для всех хостов, которіе монитоятся

node\_memory\_MemFree\_bytes / (1024 \* 1024)

Для конкретного хоста

node\_memory\_**MemTotal**\_bytes{instance="172.16.181.6:9100"}

node\_memory\_**MemFree**\_bytes{instance="172.16.181.6:9100"}

node\_memory\_**Cached**\_bytes{instance="172.16.181.6:9100"}

Если хотим не в байтах, а в мегабайтах

node\_memory\_**MemFree**\_bytes{instance="172.16.181.6:9100"}/1024/1024

### Процент свободной памяти на сервере

#Узнать остаток свободной памяти

round((node\_memory\_MemFree\_bytes / node\_memory\_MemTotal\_bytes) \* 100)

создать на базе этого запроса сообщение, при критическом уровне свободной памяти менее 5%

round((node\_memory\_MemFree\_bytes / node\_memory\_MemTotal\_bytes) \* 100) < 5

### Свобдное место на диске

node\_filesystem\_avail\_bytes{instance="172.16.181.6:9100",fstype="ext4",mountpoint="/"}

Если вы хотите узнать более подробную информацию об использовании дискового пространства на этой ноде, вы можете использовать другие метрики, такие как **node\_filesystem\_size\_bytes**, чтобы узнать общий размер файловой системы, или **node\_filesystem\_used\_bytes**, чтобы узнать, сколько пространства уже занято.

### Загруженность процессора

100 - (avg by (instance) (irate(node\_cpu\_seconds\_total{mode="idle", instance="172.16.181.6:9100"}[5m])) \* 100)