

## **#Скачиваем Prometheus**

```
curl -LO https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz
```

```
mkdir prometheus
```

```
mv prometheus-2.26.0.linux-amd64.tar.gz prometheus
```

## **#Разархивируем**

```
tar -xvf prometheus-2.26.0.linux-amd64.tar.gz
```

Удаляем архив

```
rm -i prometheus-2.26.0.linux-amd64.tar.gz
```

## **#Создадим пользователя из под которого будет работать прометейс**

```
Useradd --no-create-home --shell /usr/sbin/nologin Prometheus
```

## **#Создадим пользователя из под которого будет работать node-exporter**

```
Useradd --no-create-home --shell /bin/false node_exporter
```

## **#Создадим директории для Prometheus**

```
mkdir {/etc/,/var/lib/}Prometheus
```

```
ls {/etc/,/var/lib/}Prometheus
```

```
[root@MiWiFi-R4A-srv prometheus]# mkdir {/etc/,/var/lib/}prometheus
[root@MiWiFi-R4A-srv prometheus]# ls {/etc/,/var/lib/}prometheus
/etc/prometheus:

/var/lib/prometheus:
[root@MiWiFi-R4A-srv prometheus]# █
```

## **#Скопируем утилиты prometheus и promtool в /usr/local/bin/**

```
cp -iv prometheus-2.26.0.linux-amd64/prom{etheus,tool} /usr/local/bin/
```

```
[root@MiWiFi-R4A-srv prometheus]# ls prometheus-2.26.0.linux-amd64/
console libraries consoles LICENSE NOTICE prometheus prometheus.yml promtool
[root@MiWiFi-R4A-srv prometheus]# cp -iv prometheus-2.26.0.linux-amd64/prom{etheus,tool} /usr/local/bin/
/
«prometheus-2.26.0.linux-amd64/prometheus» -> «/usr/local/bin/prometheus»
«prometheus-2.26.0.linux-amd64/promtool» -> «/usr/local/bin/promtool»
[root@MiWiFi-R4A-srv prometheus]# █
```

## **#Скопируем директории для того чтобы передать права пользователя**

### **Prometheus и настроить его запуск**

```
[root@MiWiFi-R4A-srv prometheus]# cp -riv prometheus-2.26.0.linux-amd64/{console{s,_libraries},prometheus.yml} /etc/prometheus/█
```

```

«prometheus-2.26.0.linux-amd64/prometheus» -> «/usr/local/bin/prometheus»
«prometheus-2.26.0.linux-amd64/promtool» -> «/usr/local/bin/promtool»
[root@MiWiFi-R4A-srv prometheus]# cp -riv prometheus-2.26.0.linux-amd64/{console{s,_libraries},prometheus.yml} /etc/prometheus/
«prometheus-2.26.0.linux-amd64/consoles» -> «/etc/prometheus/consoles»
«prometheus-2.26.0.linux-amd64/consoles/index.html.example» -> «/etc/prometheus/consoles/index.html.example»
«prometheus-2.26.0.linux-amd64/consoles/node-cpu.html» -> «/etc/prometheus/consoles/node-cpu.html»
«prometheus-2.26.0.linux-amd64/consoles/node-disk.html» -> «/etc/prometheus/consoles/node-disk.html»
«prometheus-2.26.0.linux-amd64/consoles/node-overview.html» -> «/etc/prometheus/consoles/node-overview.html»
«prometheus-2.26.0.linux-amd64/consoles/node.html» -> «/etc/prometheus/consoles/node.html»
«prometheus-2.26.0.linux-amd64/consoles/prometheus-overview.html» -> «/etc/prometheus/consoles/prometheus-overview.html»
«prometheus-2.26.0.linux-amd64/consoles/prometheus.html» -> «/etc/prometheus/consoles/prometheus.html»
«prometheus-2.26.0.linux-amd64/console Libraries» -> «/etc/prometheus/console Libraries»
«prometheus-2.26.0.linux-amd64/console Libraries/menu.lib» -> «/etc/prometheus/console Libraries/menu.lib»
«prometheus-2.26.0.linux-amd64/console Libraries/prom.lib» -> «/etc/prometheus/console Libraries/prom.lib»
«prometheus-2.26.0.linux-amd64/prometheus.yml» -> «/etc/prometheus/prometheus.yml»
[root@MiWiFi-R4A-srv prometheus]# █

```

## **#Передадим права пользователя для Prometheus**

```

[root@MiWiFi-R4A-srv prometheus]# chown -Rv prometheus: /usr/local/bin/prom{etheus,tool} /etc/prometheus /var/lib/prometheus/
владелец «/usr/local/bin/prometheus» оставлен как prometheus:prometheus
владелец «/usr/local/bin/promtool» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/index.html.example» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/node-cpu.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/node-disk.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/node-overview.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/node.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/prometheus-overview.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles/prometheus.html» оставлен как prometheus:prometheus
владелец «/etc/prometheus/consoles» оставлен как prometheus:prometheus
владелец «/etc/prometheus/console Libraries/menu.lib» оставлен как prometheus:prometheus
владелец «/etc/prometheus/console Libraries/prom.lib» оставлен как prometheus:prometheus
владелец «/etc/prometheus/console Libraries» оставлен как prometheus:prometheus
владелец «/etc/prometheus/prometheus.yml» оставлен как prometheus:prometheus
владелец «/etc/prometheus» оставлен как prometheus:prometheus
изменён владелец «/var/lib/prometheus/» с root:root на prometheus:prometheus
[root@MiWiFi-R4A-srv prometheus]# █

```

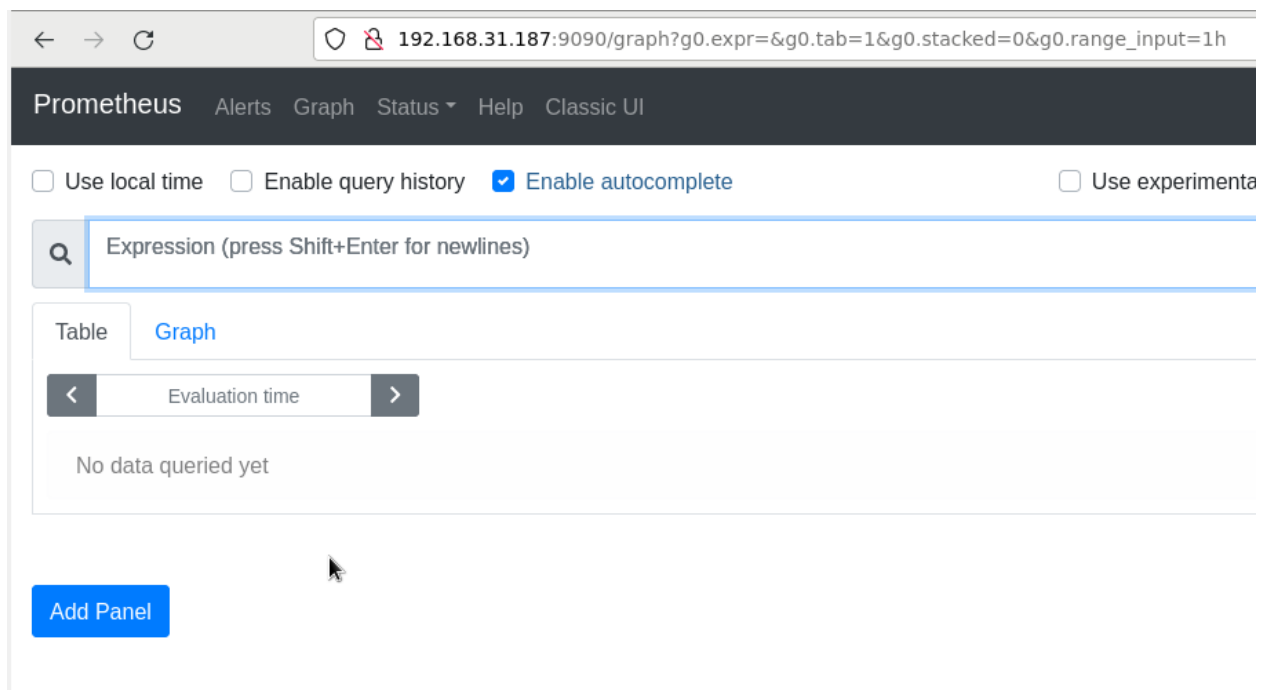
**#Проверим запуск Prometheus** (расшифровка команды: нужно выполнить команду от имени какого-то пользователя, пользователя Prometheus, используя путь до утилиты /usr/local/bin/Prometheus, которая будет выполняться, далее идут ключи конфигурационного файла и где он находится --config.file /etc/prometheus/prometheus.yml, где будет храниться база данных --storage.tsdb.path /var/lib/prometheus/, далее где находятся шаблоны в консоли --web.console.templates=/etc/prometheus/consoles, и где находятся библиотека этих самых консолей --web.console.libraries=/etc/prometheus/console\_libraries)

```

sudo -u prometheus /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /var/lib/prometheus/ --web.console.templates=/etc/prometheus/consoles --web.console.libraries=/etc/prometheus/console_libraries

```

## **#Далее в браузер ip через порт 9090**



## #Создадим файл для запуска демона system prometheus.service

vim /etc/systemd/system/prometheus.service

```
evgeniy : vim — Konsol
Файл  Правка  Вид  Закладки  Настройка  Справка
[Unit]
Description=Prometheus Monitoring
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

## #Запустим демон prometheus.service

```
[root@MiWiFi-R4A-srv evgeniy]# systemctl daemon-reload && systemctl start prometheus.service && systemctl status prometheus
Unit prometheus.service could not be found.
[root@MiWiFi-R4A-srv evgeniy]# systemctl daemon-reload && systemctl start prometheus.service && systemctl status prometheus
● prometheus.service - Prometheus Monitoring
   Loaded: loaded (/etc/systemd/system/prometheus.service; disabled; vendor preset: disabled)
   Active: active (running) since Бс 2022-01-30 19:03:09 MSK; 6s ago
 Main PID: 7066 (prometheus)
   CGroup: /system.slice/prometheus.service
           └─7066 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.t...

янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.652Z caller...t=4
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.655Z caller...t=4
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.658Z caller...t=4
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.658Z caller...515ms
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.661Z caller...GIC
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.661Z caller...ed"
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.661Z caller...yml
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.664Z caller=ma...us
янв 30 19:03:09 MiWiFi-R4A-srv prometheus[7066]: level=info ts=2022-01-30T16:03:09.664Z caller...s."
Hint: Some lines were ellipsized, use -l to show in full.
[root@MiWiFi-R4A-srv evgeniy]#
```

#Скачиваем **node\_exporter** (модуль для prometheus, который позволяет собирать информацию о состоянии машин, агент программа для отслеживания состояния машин)

curl -LO [https://github.com/prometheus/node\\_exporter/releases/download/v1.1.2/node\\_exporter-1.1.2.linux-amd64.tar.gz](https://github.com/prometheus/node_exporter/releases/download/v1.1.2/node_exporter-1.1.2.linux-amd64.tar.gz)

## #Разархивируем

tar -xvf node\_exporter-1.1.2.linux-amd64.tar.gz

Удаляем архив

rm -i prometheus-2.26.0.linux-amd64.tar.

#Допишем в конфиг /etc/prometheus/prometheus.yml

```
- job_name: 'node_exporter'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:9100']
```

```
[root@MiWiFi-R4A-srv home]# systemctl restart prometheus.service
```

```
[root@MiWiFi-R4A-srv home]# systemctl status prometheus.service
● prometheus.service - Prometheus Monitoring
   Loaded: loaded (/etc/systemd/system/prometheus.service; disabled; vendor preset: disabled)
   Active: active (running) since Бс 2022-01-30 19:29:07 MSK; 1min 21s ago
 Main PID: 7299 (prometheus)
   CGroup: /system.slice/prometheus.service
           └─7299 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --stc
```

## # Устанавливаем node\_exporter

```
[root@MiWiFi-R4A-srv home]# cp -vi node_exporter-1.1.2.linux-amd64/node_exporter /usr/local/bin
«node_exporter-1.1.2.linux-amd64/node_exporter» -> «/usr/local/bin/node_exporter»
```

## #Передаем права node\_exporter одноименному пользователю

```
[root@MiWiFi-R4A-srv home]# chown node_exporter: /usr/local/bin/node_exporter
```

## #Создадим юнит для node\_exporter

```
[root@MiWiFi-R4A-srv home]# nano /etc/systemd/system/node_exporter.service
```

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

## #Перезапускаем все демоны и запускаем node\_exporter

```
[root@MiWiFi-R4A-srv home]# systemctl daemon-reload
[root@MiWiFi-R4A-srv home]# systemctl start node_exporter.service
[root@MiWiFi-R4A-srv home]# systemctl status node_exporter.service
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; disabled; vendor preset: disabled)
   Active: active (running) since Бс 2022-01-30 19:46:10 MSK; 5s ago
     Main PID: 7524 (node_exporter)
        CGroup: /system.slice/node_exporter.service
                └─7524 /usr/local/bin/node_exporter

янв 30 19:46:10 MiWiFi-R4A-srv node_exporter[7524]: level=info ts=2022-01-30T16:46:10.897Z call.
янв 30 19:46:10 MiWiFi-R4A-srv node_exporter[7524]: level=info ts=2022-01-30T16:46:10.897Z call.
```

## #Если зайдём на хост 9100 node\_exporter, то увидим метрики

```
[root@MiWiFi-R4A-srv home]# curl localhost:9100
<html>
    <head><title>Node Exporter</title></head>
    <body>
        <h1>Node Exporter</h1>
        <p><a href="/metrics">Metrics</a></p>
    </body>
</html>[root@MiWiFi-R4A-srv home]#
```



## #Проверим работу node\_exporter и копируем запрос

```
node_filesystem_files_free{device="/dev/mapper/centos_miwifi--r4a--srv-root",fstype="xfs",mountpoint="/" } 1.2717006e+07
node_filesystem_files_free{device="/dev/sda1",fstype="xfs",mountpoint="/boot"} 523954
node_filesystem_files_free{device="tmpfs",fstype="tmpfs",mountpoint="/run"} 544374
node_filesystem_files_free{device="tmpfs",fstype="tmpfs",mountpoint="/run/user/0"} 544973
node_filesystem_files_free{device="tmpfs",fstype="tmpfs",mountpoint="/run/user/1000"} 544962
# HELP node_filesystem_free_bytes Filesystem free space in bytes.
# TYPE node_filesystem_free_bytes gauge
node_filesystem_free_bytes{device="/dev/mapper/centos_miwifi--r4a--srv-root",fstype="xfs",mountpoint="/" } 1.6223715328e+10
node_filesystem_free_bytes{device="/dev/sda1",fstype="xfs",mountpoint="/boot"} 8.23676928e+08
node_filesystem_free_bytes{device="tmpfs",fstype="tmpfs",mountpoint="/run"} 2.222645248e+09
node_filesystem_free_bytes{device="tmpfs",fstype="tmpfs",mountpoint="/run/user/0"} 4.4644352e+08
node_filesystem_free_bytes{device="tmpfs",fstype="tmpfs",mountpoint="/run/user/1000"} 4.46394368e+08
# HELP node_filesystem_readonly Filesystem read-only status.
# TYPE node_filesystem_readonly gauge
node_filesystem_readonly{device="/dev/mapper/centos_miwifi--r4a--srv-root",fstype="xfs",mountpoint="/" } 0
```

## #Вставляем в Prometheus и видим, что код запроса 523954 одинаковый

Prometheus Alerts Graph Status Help Classic UI

☐ Use local time ☐ Enable query history ☒ Enable autocomplete ☐ Use experimental editor ☒ Enable highlighting ☒ Enable linter

node\_filesystem\_files\_free{device="/dev/sda1",fstype="xfs",mountpoint="/boot"}

Execute

Table Graph

node\_filesystem\_files\_free{device="/dev/sda1",fstype="xfs",instance="localhost:9100",job="node\_exporter",mountpoint="/boot"} 523954

## #Сделаем настройку графаны (скачиваем)

```
[root@MiWi-Fi-R4A-srv home]# curl -LO https://dl.grafana.com/oss/release/grafana-8.3.4-1.x86_64.rpm
% Total    % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68.9M 100 68.9M 0 0 4415k 0 0:00:15 0:00:15 --:--:-- 4621k
[root@MiWi-Fi-R4A-srv home]#
```

## #Устанавливаем скачанный пакет

```
[root@MiWi-Fi-R4A-srv home]# yum install ./grafana-8.3.4-1.x86_64.rpm
```

## #Перезапускаем все демоны и запускаем grafana-server

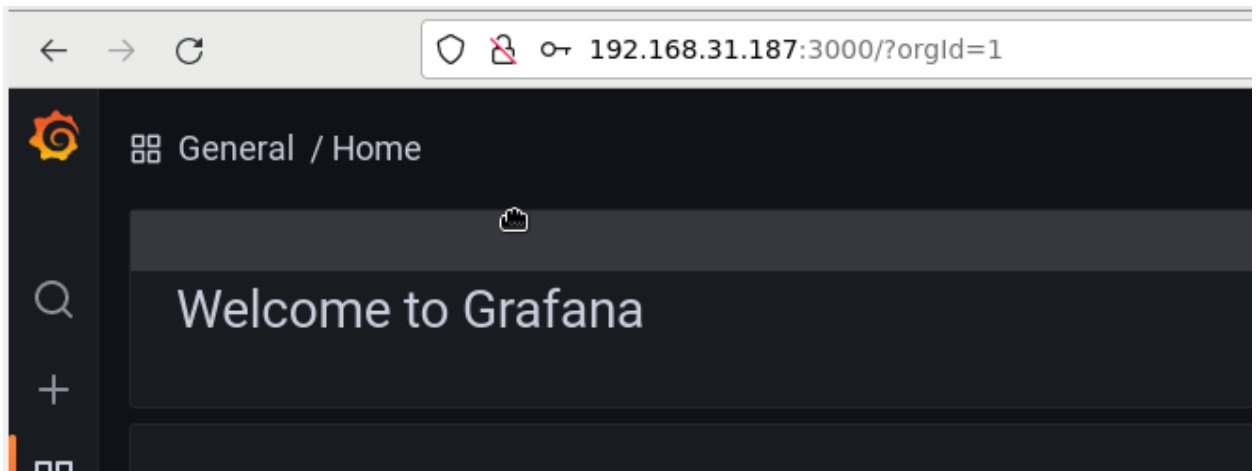
```
[root@MiWi-Fi-R4A-srv home]# systemctl daemon-reload && systemctl start grafana-server && systemctl stat
us grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Bc 2022-01-30 20:25:39 MSK; 44s ago
     Docs: http://docs.grafana.org
    Main PID: 7997 (grafana-server)
      CGroup: /system.slice/grafana-server.service
              └─7997 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --pidfile=/var/run/graf...

янв 30 20:26:23 MiWi-Fi-R4A-srv systemd[1]: [/usr/lib/systemd/system/grafana-server.service:29]...ce'
янв 30 20:26:23 MiWi-Fi-R4A-srv systemd[1]: [/usr/lib/systemd/system/grafana-server.service:31]...ce'
```

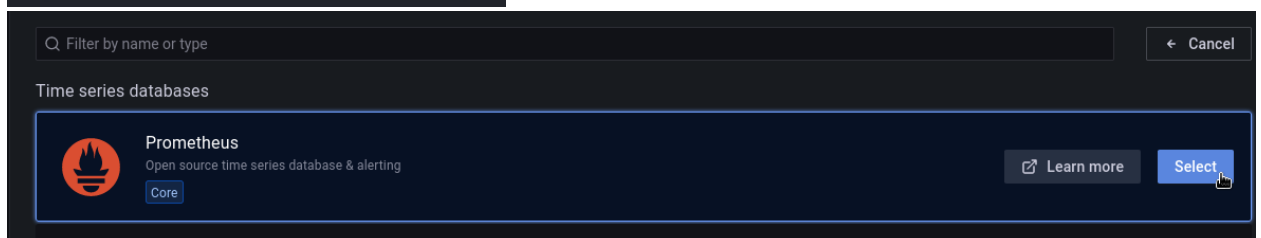
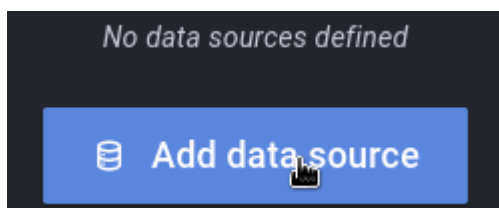
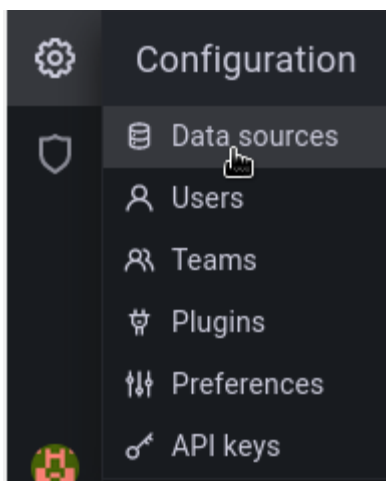
## #Заходим в grafana через браузер порт 3000

Пользователь: admin

Пароль: admin



## #Прописываем в настройках Prometheus localhost:9090



HTTP

URL

Access

[Back](#) [Explore](#) [Delete](#) [Save & test](#)

## #Построим графики на prometheus на основе собранных метрик

[+](#) Create

- [Dashboard](#)
- [Folder](#)
- [Import](#)
- [Alert rule](#)

[Upload JSON file](#)

Import via [grafana.com](#)

[Load](#)

(или 11074)





## Import

Import dashboard from file or Grafana.com

### Importing dashboard from Grafana.com

Published by

rfraile

Updated on

2022-01-10 11:41:23

### Options

Name

Node Exporter Full

Folder

General

#### Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used to 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

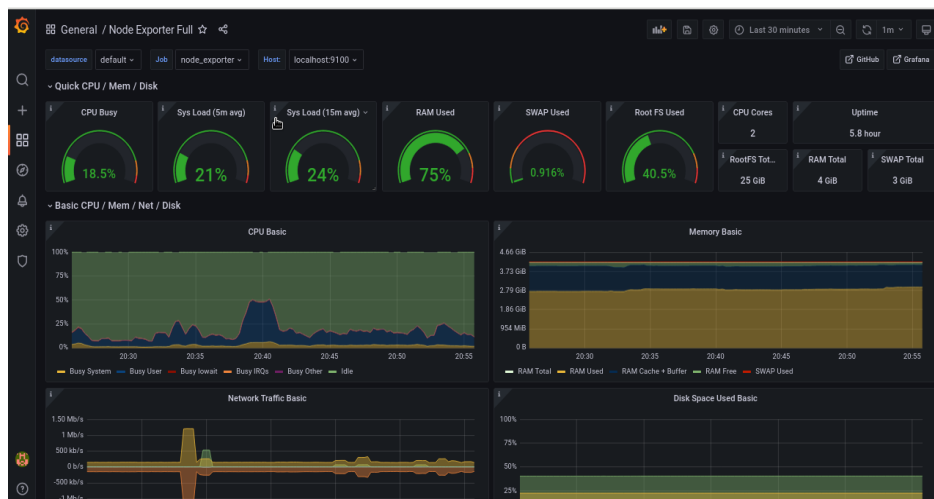
Change uid

Prometheus

Prometheus

Import

Cancel



CPU / Memory / Net / Disk (7 panels)

Memory Meminfo (15 panels)

Memory Vmstat (4 panels)

System Timesync (4 panels)

System Processes (7 panels)

System Misc (7 panels)

Hardware Misc (3 panels)

Systemd (2 panels)

Storage Disk (8 panels)

Storage Filesystem (5 panels)

Network Traffic (17 panels)

Network Sockstat (5 panels)

Network Netstat (11 panels)

Node Exporter (2 panels)

