

## 4.9 Централизованный сбор метрик Prometheus

1. Скачиваем с гитхаба ласт версию прометеуса

2. wget

`https://github.com/prometheus/prometheus/releases/download/v2.36.1/prometheus-2.36.1.linux-amd64.tar.gz`

```
xokage@ubuntu: ~/prometheus/prometheus-2.36.1.linux-amd64
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:9100"]
```

3.

Меняю конфиг prometheus.yml, а именно блок в job prometheus, target на localhost:9100, поскольку node\_exporter по дефолту работает на 9100 порту

### Installation and Usage

If you are new to Prometheus and `node_exporter` there is a [simple step-by-step guide](#).

The `node_exporter` listens on HTTP port 9100 by default. See the `--help` output for more options.

### Ansible

For automated installs with [Ansible](#), there is the [Cloud Alchemy role](#).

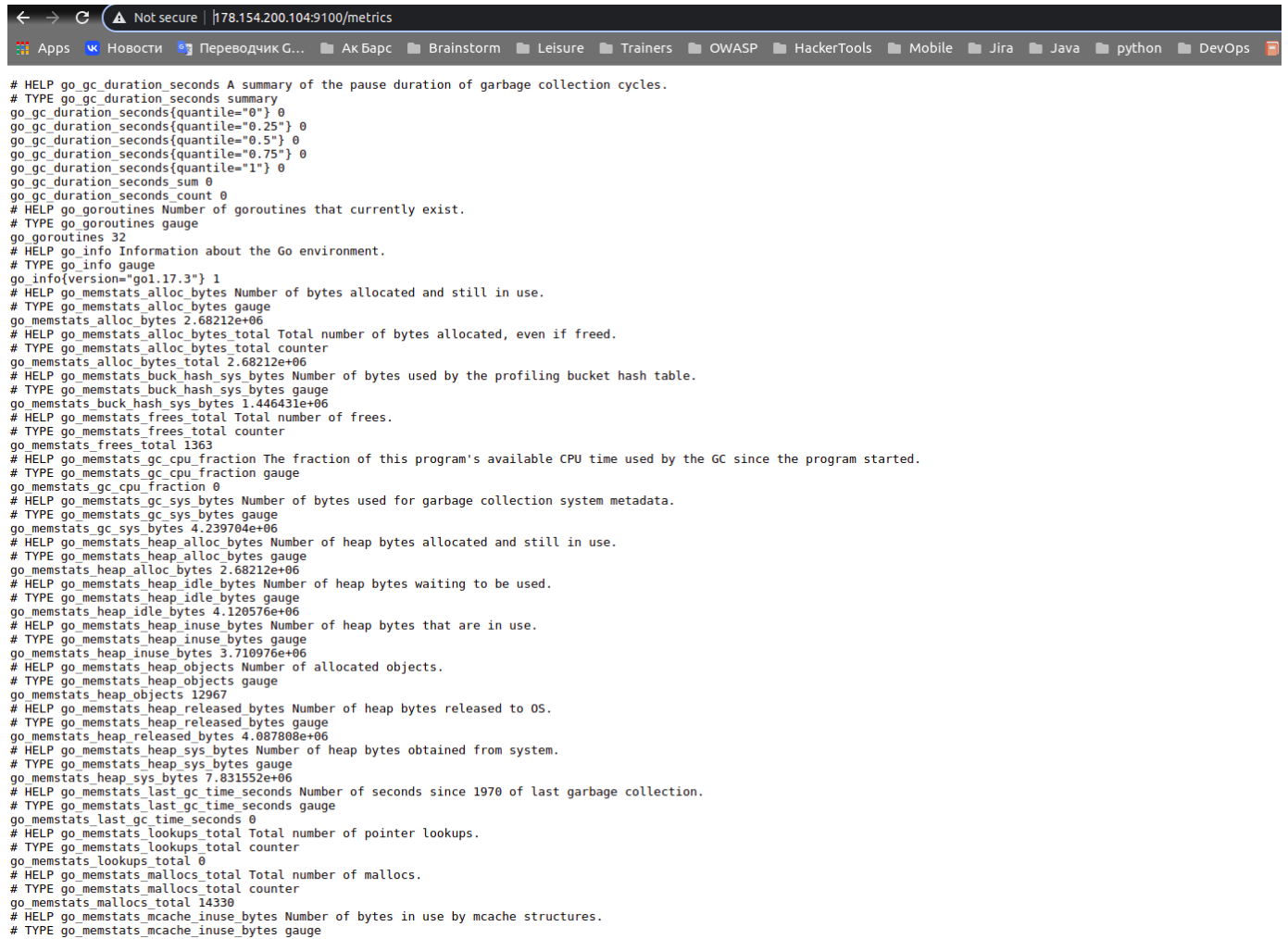
4. Качаем node\_exporter

`wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_`



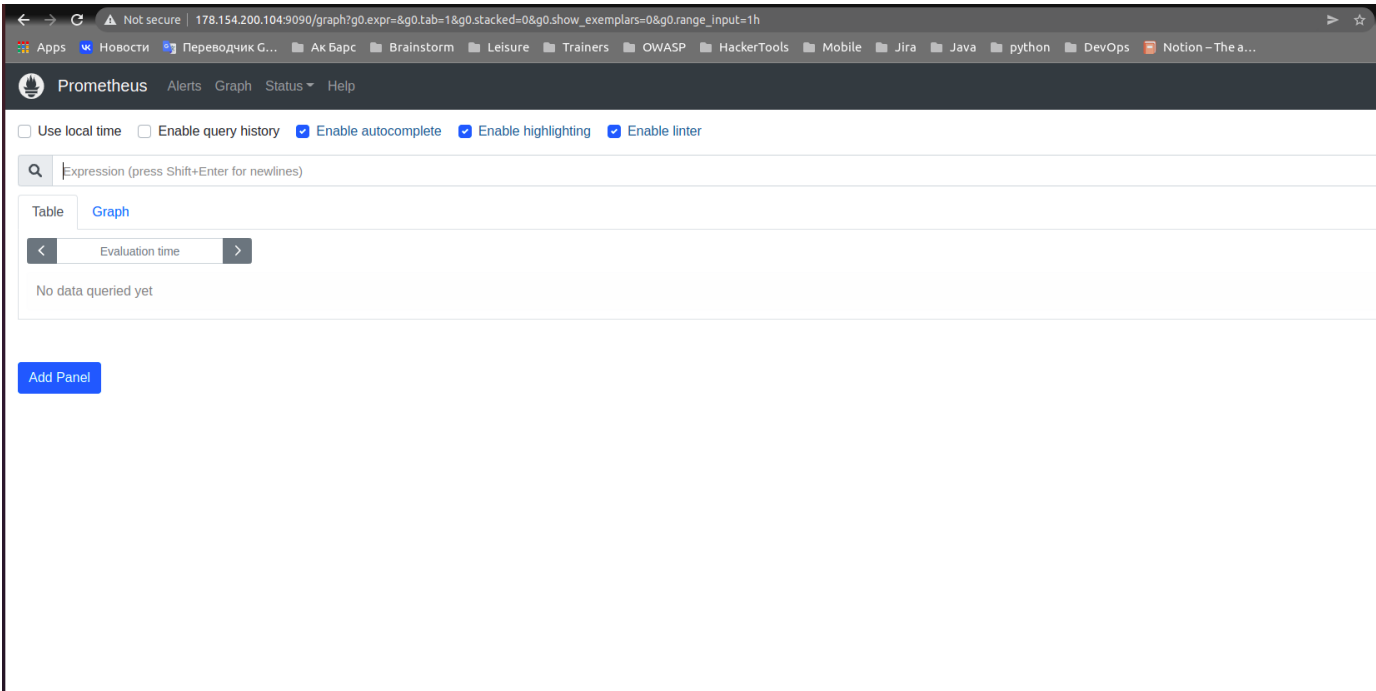
и запускаем ./node\_exporter

## 5. Запустив, через браузер можно увидеть метрики

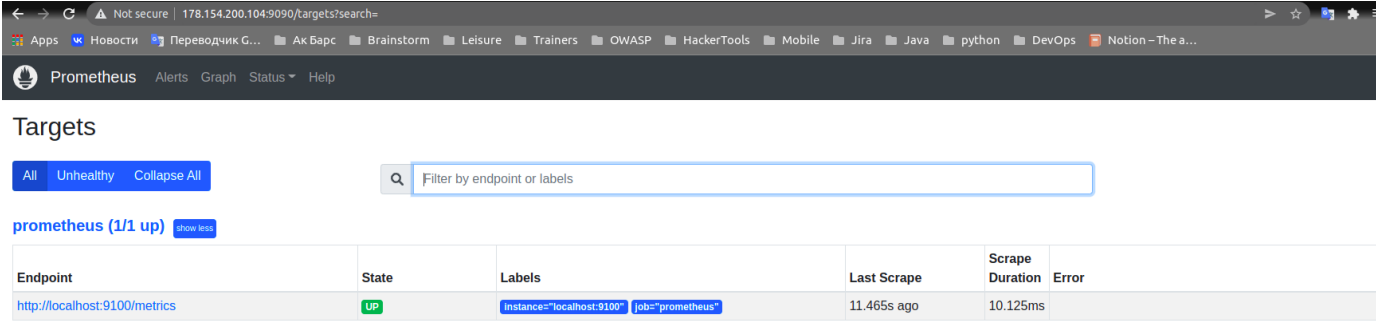


```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 32
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.68212e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 2.68212e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.446431e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1363
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 0
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 4.239704e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.68212e+06
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 4.120576e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 3.710976e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 12967
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 4.087808e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 7.831552e+06
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 0
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 14330
# HELP go_memstats_mcache_inuse_bytes Number of bytes in use by mcache structures.
# TYPE go_memstats_mcache_inuse_bytes gauge
```

## 6. Далее запускаем прометеус



7.



В таргетах можно увидеть наш инстанс, где висит экспортер

8. Далее нужно установить экспортер на второй тачке, и добавить в таргеты прометеуса еще одну тачку

9. Метрики со второй тачки

```

← → ↻ ⚠ Not secure | 51.250.77.238:9100/metrics
Apps Новости Переводчик G... Ак Басп Brainstorm Leisure Trainers OWASP HackerTools Mobile Jira Java

# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.38496e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.38496e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.446159e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 748
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 0
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 4.178024e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.38496e+06
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 1.2288e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 2.605056e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 9463
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 1.2288e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 3.833856e+06
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 0
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 10711

```

## 10. Поменял конфиг и запустил снова прометеус

```

xokage@ubuntu: ~/prometheus/prometheus-2.36.1-linux-amd64

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:9100"]
      - targets: ["51.250.77.238:9100"]

```

← → ↻ 🔒 Not secure | 178.154.200.104:9090/targets?search=

Apps Новости Переводчик G... Ак Бэпс Brainstorm Leisure Trainers OWASP HackerTools Mobile Jira Java python DevOps Notion – The a...

Prometheus Alerts Graph Status Help

Targets

All Unhealthy Collapse All

Filter by endpoint or labels

prometheus (2/2 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="prometheus"	1.415s ago	9.211ms	
http://51.250.77.238:9100/metrics	UP	instance="51.250.77.238:9100" job="prometheus"	89.000ms ago	11.374ms	

11. node\_filesystem\_avail\_bytes /1024/1024 Просмотр загруженности диска.

Prometheus Alerts Graph Status Help

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

node\_filesystem\_avail\_bytes /1024/1024

Execute

Load time: 64ms Resolution: 14s Result series: 11

Table Graph

Evaluation time

{device="devmapper/vol_g1-logical_vol1",fstype="ext4",instance="localhost:9100",job="prometheus",mountpoint="/var/log"}	892.37890625
{device="devmapper/vol_g1-logical_vol2",fstype="xfs",instance="localhost:9100",job="prometheus",mountpoint="/var/log/db"}	974.60546875
{device="devvda2",fstype="ext4",instance="51.250.77.238:9100",job="prometheus",mountpoint="/"}	26268.37890625
{device="devvda2",fstype="ext4",instance="localhost:9100",job="prometheus",mountpoint="/"}	8256.13671875
{device="tmpfs",fstype="tmpfs",instance="51.250.77.238:9100",job="prometheus",mountpoint="/run"}	392.4140625
{device="tmpfs",fstype="tmpfs",instance="51.250.77.238:9100",job="prometheus",mountpoint="/run/lock"}	5
{device="tmpfs",fstype="tmpfs",instance="51.250.77.238:9100",job="prometheus",mountpoint="/run/user/1000"}	393.16796875
{device="tmpfs",fstype="tmpfs",instance="localhost:9100",job="prometheus",mountpoint="/run"}	177.64453125
{device="tmpfs",fstype="tmpfs",instance="localhost:9100",job="prometheus",mountpoint="/run/lock"}	5
{device="tmpfs",fstype="tmpfs",instance="localhost:9100",job="prometheus",mountpoint="/run/user/1000"}	198.30859375

Remove Panel

Add Panel

Метрики с локалхоста:

```
xokage@ubuntu:~/prometheus/prometheus-2.36.1.linux-amd64$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            961M     0  961M   0% /dev
tmpfs           199M   21M  178M  11% /run
/dev/vda2       15G   6.1G   8.1G  43% /
tmpfs           992M     0  992M   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
tmpfs           992M     0  992M   0% /sys/fs/cgroup
/dev/mapper/vol_g1-logical_vol1  976M   17M  893M   2% /var/log
/dev/mapper/vol_g1-logical_vol2 1014M   40M  975M   4% /var/log/db
tmpfs           199M     0  199M   0% /run/user/1000
xokage@ubuntu:~/prometheus/prometheus-2.36.1.linux-amd64$
```

Метрики с 51.250.77.238:9100:

```
ubuntu@ubuntu2:~/prometheus2/node_exporter-1.3.1.linux-amd64$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           394M   776K  393M   1% /run
/dev/vda2       30G   2.6G   26G  10% /
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           5.0M     0   5.0M   0% /run/lock
tmpfs           2.0G     0  2.0G   0% /sys/fs/cgroup
tmpfs           394M     0  394M   0% /run/user/1000
ubuntu@ubuntu2:~/prometheus2/node_exporter-1.3.1.linux-amd64$
```

В первом случае в корне свободно 8.1G, во втором 26G. Прометеус показывает те

же показатели

26268.37890625

8256.13671875

Все сходится