

# Лабораторная работа №5. Мониторинг.

---

## Что потребуется перед началом:

- ПК, способный запустить систему виртуализации с виртуальной машиной GNU/Linux.
- Минимум 6 GiB свободного места на жестком диске (под систему и снапшоты).
- Пакет или установщик системы виртуализации (рекомендуется VirtualBox).
- Загруженный образ дистрибутива (рекомендуется Ubuntu 20.04).
- Результаты работы прошлых лабораторных.

## План и задачи лабораторной:

1. Часть 1. Установка и настройка VictoriaMetrics + Grafana.
  1. Подготовка рабочего окружения
  2. Установка VictoriaMetrics
  3. Установка Grafana
2. Часть 2. Установка и настройка AlertManager
  1. Установка AlertManager
  2. Установка vmaalert
  3. Настройка AlertManager-bot
  4. Проверим работу оповещений
  5. Мониторим свое приложение

**Отчет** - в любом читаемом формате (pdf, md, doc, docx, pages).

Обязательное содержимое отчета:

0. Фамилия и инициалы студента, номер группы, номер варианта
1. План и задачи лабораторной работы
2. Краткое описание хода выполнения работы
3. Приложить очищенный вывод `history` выполненных команд

## Что нужно сделать, чтобы сдать лабораторную?

1. Выполнить все действия, представленные в методических указаниях и ознакомиться с материалом
2. Продемонстрировать результаты выполнения преподавателю, быть готовым повторить выполнение части задач из лабораторной по требованию
3. Ответить на контрольные вопросы

# Вступление

---

Для полного понимания рекомендуется ознакомиться с [документацией](#).

## Часть 1. Установка и настройка VictoriaMetrics + Grafana.

---

### 1.1. Подготовка рабочего окружения

В предыдущих лабораторных мы уже настроили рабочее окружение в ОС Ubuntu.

```
# Заведите пользователя ansible
sudo adduser ansible
sudo usermod -aG wheel ansible

# Скопируйте authorized_keys если вы ходили под другим пользователем (например
mak):
sudo cp -r /home/mak/.ssh /home/ansible/ && chown -R ansible:ansible
/home/ansible/

# Установите ansible
sudo apt install ansible
```

Теперь катнем ansible - поставим node\_exporter и другое ПО:

```
git clone git@bmstu.codes:iu5/infrastructure/ansible-monitoring-test.git
cd ansible-monitoring-test

ansible-playbook -i inventory_vms/ playbooks/all_hosts.yml
cd -
```

Проверим:

```
sudo systemctl status node_exporter
```

Попробуем поставить версию новее:

```
wget
https://github.com/prometheus/node_exporter/releases/download/v1.4.0/node_exporter-1.4.0.linux-amd64.tar.gz
sudo tar -xvf node_exporter-1.4.0.linux-amd64.tar.gz -C /usr/local/bin/

sudo chown -R root:root /usr/local/bin/node_exporter-1.4.0.linux-amd64

sudo systemctl edit --full node_exporter.service
```

Добавим к пути до исполняемого файла каталог с новой версией: `node_exporter-1.4.0.linux-amd64`

```
sudo systemctl daemon-reload && sudo systemctl restart node_exporter
```

## 1.2. Установка VictoriaMetrics

Теперь установим VictoriaMetrics из готовых сборок. Для этого идем в [releases](#) на GitHub и качаем нужную версию и архитектуру, распакуем сразу в каталог с бинарями и напишем systemd-unit:

```
wget
https://github.com/VictoriaMetrics/VictoriaMetrics/releases/download/v1.83.0/victoria-metrics-linux-amd64-v1.83.0.tar.gz

sudo tar -xvf victoria-metrics-linux-amd64-v1.83.0.tar.gz -C /usr/local/bin/
sudo chown root:root /usr/local/bin/victoria-metrics-prod

sudo systemctl edit --force --full victoria-metrics.service
```

```
[Unit]
Description=victoria-metrics
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Restart=always
RestartSec=2
StartLimitBurst=3
StartLimitInterval=60
Type=simple
ExecStart=/usr/local/bin/victoria-metrics-prod \
  -promscrape.config=/etc/prometheus/prometheus.conf \
  -storageDataPath=/mnt/data/victoria-metrics/ \
```

```
-retentionPeriod=3 \  
-search.latencyOffset=15s \  
-httpListenAddr=0.0.0.0:8428
```

```
ExecReload=/bin/kill -HUP $MAINPID
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Теперь нужно создать требуемые каталоги и конфиги:

```
sudo mkdir -p /etc/prometheus  
sudo mkdir -p /mnt/data/victoria-metrics  
sudo nano /etc/prometheus/prometheus.conf  
sudo chown -R root:prometheus /etc/prometheus  
sudo chown -R prometheus:prometheus /mnt/data/victoria-metrics
```

Файл конфигурации для VictoriaMetrics:

```
global:  
  scrape_interval:      15s # Set the scrape interval to every X seconds.  
  # Default is every 1 minute.  
  
# A scrape configuration containing exactly one endpoint to scrape:  
# Here it's Victoria Metrics itself.  
scrape_configs:  
  - job_name: 'victoria-metrics'  
    static_configs:  
      - targets: ['localhost:8428']  
  
  - job_name: 'node'  
    static_configs:  
      - targets:  
        - 'localhost:9100'
```

Запускаем:

```
sudo systemctl start victoria-metrics
```

Проверим слушает ли порт:

```
sudo netstat -nlpt | grep 8428
```

Теперь пробросим порт 8428 на хостовую систему и зайдем в браузер в [интерфейс](#).

Проверим [список целей](#).

Почему URL node\_exporter недоступен?

Попробуем составить запрос в [vmui](#):

```
rate(node_network_receive_bytes_total[1m])
```

## 1.3. Установка Grafana

# В связи с санкциями стандартный способ не подойдет:

```
# sudo apt-get install -y apt-transport-https
# sudo apt-get install -y software-properties-common wget
# wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
# echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list

# apt update
# apt install libfontconfig1 grafana

# Качаем пакет через VPN/ЯД и ставим:
wget
https://gist.githubusercontent.com/Yegorov/dc61c42aa4e89e139cd8248f59af6b3e/ra
w/20ac954e202fe6a038c2b4bb476703c02fe0df87/ya.py

chmod 750 ya.py
./ya.py https://disk.yandex.ru/d/OmpS6D4ZKWPYWg ./

sudo apt install libfontconfig1
sudo dpkg -i grafana-enterprise_9.2.3_amd64.deb

sudo systemctl status grafana-server
sudo netstat -nlpt | grep graf
```

Пробрасываем нужный порт (default - 3000), заходим в [интерфейс Grafana](#), user: `admin`, password: `admin`.

Устанавливаем новый пароль для `admin`.

Настраиваем [datasource](#).

Пробуем подключиться к другому серверу: [grafana.argobay.ml](#).

## Часть 2. Установка и настройка AlertManager, rsyslog и logrotate.

### 2.1. Установка AlertManager

```
cd ansible-monitoring-test
ansible-playbook -i inventory_vms/ playbooks/monitoring.yml --tags
alertmanager
```

## 2.2. Установка vmaalert

```
wget
https://github.com/VictoriaMetrics/VictoriaMetrics/releases/download/v1.83.0/v
mutils-linux-amd64-v1.83.0.tar.gz

sudo tar -xvf vmutils-linux-amd64-v1.83.0.tar.gz -C /usr/local/bin/
sudo chown root:root /usr/local/bin/vm*

sudo systemctl edit --force --full vmaalert.service
```

vmaalert.service:

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

[Service]
User=prometheus
Group=prometheus
Restart=always
RestartSec=2
StartLimitBurst=3
StartLimitInterval=30
Type=simple
ExecStart=/usr/local/bin/vmaalert-prod \
  -rule="/etc/prometheus/*-rules.yml" \
  -evaluationInterval=15s \
  -datasource.url=http://127.0.0.1:8428 \
  -notifier.url=http://127.0.0.1:9093 \
  -remoteWrite.url=http://127.0.0.1:8428 \
  -remoteRead.url=http://127.0.0.1:8428 \
  -httpListenAddr=127.0.0.1:8880
ExecReload=/bin/kill -HUP $MAINPID

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

Конфиг оповещений:

```
sudo nano /etc/systemd/system/vmalert.service
```

```
groups:
- name: main
  rules:

- alert: HostSystemdServiceCrashed
  expr: node_systemd_unit_state{state="failed"} == 1
  for: 0m
  labels:
    severity: critical
    module: infra
  annotations:
    summary: "Host systemd service crashed (instance {{ $labels.instance
}})"
    description: "{{ $labels.name }} at {{ $labels.instance }} - state
failed."

- alert: HostOOMKillDetected
  expr: increase(node_vmstat_oom_kill[1m]) > 0
  for: 0m
  labels:
    severity: critical
    module: infra
  annotations:
    summary: "Host OOM kill detected (instance {{ $labels.instance }})"
    description: "OOM kill detected at {{ $labels.instance }}, VALUE =
{{ $value }}, LABELS: {{ $labels }}."

- alert: HostOutOfMemory
  expr: node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes <
0.2s
  for: 1m
  labels:
    severity: warning
    module: infra
  annotations:
    summary: "Host {{ $labels.instance }} {{ $labels.env }} hight memory
usage"
    description: "{{ $labels.instance }} has more than 70% of its memory
used, VALUE = {{ $value }}, LABELS: {{ $labels }}."

- alert: HostHighCpuLoad
  expr: 100 - (avg by(instance)
(rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 70
  for: 0m
  labels:
    severity: warning
```

```

    module: infra
    annotations:
        summary: "Host high CPU load (instance {{ $labels.instance }})"
        description: "{{ $labels.instance }} CPU load is > 70%, VALUE = {{
$value }}, LABELS: {{ $labels }}."

- alert: HostOutOfDiskSpace
    expr: (node_filesystem_avail_bytes * 100) / node_filesystem_size_bytes
< 20 and ON (instance, device, mountpoint) node_filesystem_readonly == 0
    for: 2m
    labels:
        severity: warning
        module: infra
    annotations:
        summary: "Host out of disk space (instance {{ $labels.instance }})"
        description: "{{ $labels.instance }} disk is almost full (< 20%
left), VALUE = {{ $value }}, LABELS: {{ $labels }}."

- alert: HostNetworkRxErrors
    expr: rate(node_network_receive_errs_total[2m]) /
rate(node_network_receive_packets_total[2m]) > 0.01
    for: 2m
    labels:
        severity: warning
        module: infra
    annotations:
        summary: "Host Network Receive Errors (instance {{ $labels.instance
}})"
        description: '{{ $labels.instance }} interface {{ $labels.device }}
has encountered {{ printf "%.0f" $value }} receive errors in the last five
minutes. VALUE = {{ $value }}.'

- alert: HostNetworkTxErrors
    expr: rate(node_network_transmit_errs_total[2m]) /
rate(node_network_transmit_packets_total[2m]) > 0.01
    for: 2m
    labels:
        severity: warning
        module: infra
    annotations:
        summary: "Host Network Transmit Errors (instance {{ $labels.instance
}})"
        description: '{{ $labels.instance }} interface {{ $labels.device }}
has encountered {{ printf "%.0f" $value }} transmit errors in the last five
minutes. VALUE = {{ $value }}.'

- alert: HostClockSkew

```



```

    expr: (node_timex_offset_seconds > 0.05 and
deriv(node_timex_offset_seconds[5m]) >= 0) or (node_timex_offset_seconds <
-0.05 and deriv(node_timex_offset_seconds[5m]) <= 0)
    for: 2m
    labels:
        severity: warning
        module: infra
    annotations:
        summary: "Host clock skew (instance {{ $labels.instance }})"
        description: "Clock skew detected. Clock is out of sync. VALUE = {{
$value }}. LABELS: {{ $labels }}."

- alert: InstanceDown
  expr: up == 0
  for: 1m
  labels:
      severity: critical
      module: infra
  annotations:
      summary: "Instance {{ $labels.instance }} down"
      description: "{{ $labels.instance }} of job {{ $labels.job }} has
been down for more than 1 minute."

- alert: HostEdacUncorrectableErrorsDetected
  expr: node_edac_uncorrectable_errors_total > 0
  for: 0m
  labels:
      severity: warning
      module: infra
  annotations:
      summary: "Host EDAC Uncorrectable Errors detected (instance {{
$labels.instance }})"
      description: '{{ $labels.instance }} has had {{ printf "%.0f" $value
}} uncorrectable memory errors reported by EDAC. VALUE = {{ $value }}. LABELS:
{{ $labels }}.'

- alert: HostPhysicalComponentTooHot
  expr: node_hwmon_temp_celsius > 70
  for: 3m
  labels:
      severity: critical
      module: infra
  annotations:
      summary: "Host physical component too hot (instance {{
$labels.instance }})"
      description: "Physical hardware component {{ $labels.instance }} too
hot. VALUE = {{ $value }}."

```

```
sudo systemctl start vmalert
```

## 2.3. Настройка AlertManger-bot

Идем в telegram к боту [@BotFather](#) и создаем нового бота командой `/newbot`. Когда пройдем квест по созданию получим токен, его нужно будет вписать в конфиг alertmanage-bot-у, но т.к. мы настраиваем его с помощью ansible, то в файл

`inventory_vms/group_vars/all/secrets.yml`, после чего катим:

```
ansible-playbook -i inventory_vms/ playbooks/monitoring.yml --tags  
alertmanager-bot
```

Если все хорошо - идем к своему боту (в моем примере [https://t.me/yuki\\_iu5\\_bot](https://t.me/yuki_iu5_bot)) и пишем `/start`. Бот должен ответить.

## 2.4. Проверим работу оповещений

```
sudo systemctl stop node_exporter
```

## 2.5. Мониторим свое приложение

```
sudo apt install python3-pip  
pip3 install flask  
pip3 install prometheus_flask_exporter
```

Напишем тестовый сервис:

```
#!/usr/bin/env python3  
  
from flask import Flask, request  
from prometheus_flask_exporter import PrometheusMetrics  
  
app = Flask(__name__)  
metrics = PrometheusMetrics(app)  
  
endpoints = ("one", "two", "error")  
  
@app.route('/one')  
def endpoint_one():
```

```

    return 'ONE'

@app.route('/two')
def endpoint_two():
    return 'TWO'

@app.route('/error')
def endpoint_err():
    return ":( ", 500

if __name__ == "__main__":
    app.run("0.0.0.0", 5000, threaded=True)

```

Запустим:

```

export FLASK_APP=test
export DEBUG_METRICS=false
./test.py

```

Ставим дашборд: [https://raw.githubusercontent.com/rycus86/prometheus\\_flask\\_exporter/master/examples/sample-signals/grafana/dashboards/example.json](https://raw.githubusercontent.com/rycus86/prometheus_flask_exporter/master/examples/sample-signals/grafana/dashboards/example.json)

```

# Проверяем
for i in {1..50}; do curl 127.0.0.1:5000/one; done

```

Задача: добавить кастомные метрики в свое или тестовое приложение, построить график

Доп. задача: направить логи в rsyslog, а оттуда в файл, который ротится logrotate

## Контрольные вопросы

1. Что такое мониторинг, зачем нужен, какие основные компоненты входят?
2. Как настроить рассылку оповещений?
3. Возможные ограничения при рассылке оповещений?
4. Как добавить метрику в свой код?
5. Как построить график в Grafana?
6. Что такое PromQL? Как обнаружить "всплески" в метрике?