

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Сибирский государственный университет телекоммуникаций и информатики»  
(СибГУТИ)

Кафедра вычислительных систем

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ  
по дисциплине «Архитектура вычислительных систем»

Выполнил:

студент группы ИВ-221  
Гордов Роман Сергеевич

Новосибирск 2024 г.

## Цель работы

Создать систему мониторинга на основе Docker с использованием технологий:

- **Prometheus** для сбора и хранения метрик.
  - **Grafana** для визуализации данных.
  - **Node Exporter, Telegraf, CAdvisor** для экспорта метрик.
  - **Alertmanager** для управления уведомлениями об аномалиях.
- Обеспечить доступ к интерфейсам через HTTPS и настроить алертинг для одной из метрик.

## Требования (DoD)

1. Интерфейсы доступны по разным доменам через HTTPS (Prometheus, Alertmanager, Grafana).
2. Prometheus собирает метрики с нескольких источников (Node Exporter, Telegraf, CAdvisor).
3. Создано несколько визуализаций в Grafana.
4. Данные и настройки сохраняются при пересоздании проекта.
5. Один из показателей имеет настроенный алерт.

## Реализация

### 1. Docker Compose

Создана инфраструктура с использованием `docker-compose.yml` для запуска всех необходимых компонентов:

version: '3.9'

services:

prometheus:

image: prom/prometheus:latest

container\_name: prometheus

volumes:

- ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml

- ./prometheus/alerts.yml:/etc/prometheus/alerts.yml

- prometheus\_data:/prometheus

ports:

- "9090:9090"

networks:

- monitoring

alertmanager:

image: prom/alertmanager:latest

container\_name: alertmanager

volumes:

- ./alertmanager/config.yml:/etc/alertmanager/config.yml

ports:

- "9093:9093"

networks:

- monitoring

grafana:

image: grafana/grafana:latest

container\_name: grafana

volumes:

- grafana\_data:/var/lib/grafana

ports:

- "3000:3000"

networks:

- monitoring

environment:

- GF\_SECURITY\_ADMIN\_USER=admin
- GF\_SECURITY\_ADMIN\_PASSWORD=admin

node\_exporter:

image: prom/node-exporter:latest

container\_name: node\_exporter

ports:

- "9100:9100"

networks:

- monitoring

cadvisor:

image: gcr.io/cadvisor/cadvisor:latest

container\_name: cadvisor

ports:

- "8080:8080"

networks:

- monitoring

volumes:

- /:/rootfs:ro
- /var/run:/var/run:ro
- /sys:/sys:ro
- /var/lib/docker:/var/lib/docker:ro

traefik:

image: traefik:v2.10

container\_name: traefik

ports:

- "80:80"

- "443:443"

volumes:

- ./traefik/traefik.yml:/etc/traefik/traefik.yml

- ./traefik/acme.json:/acme.json

networks:

- monitoring

volumes:

prometheus\_data:

grafana\_data:

networks:

monitoring:

driver: bridge

## 2. Сбор метрик

- **Prometheus** настроен для сбора метрик с источников:
  - **Node Exporter**: предоставляет данные о состоянии системы (нагрузка процессора, память, диск).
  - **CAdvisor**: метрики контейнеров Docker.
  - **Telegraf**: универсальный сбор данных (системные метрики, метрики приложений).

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'node'

static\_configs:

- targets: ['node\_exporter:9100']

- job\_name: 'cadvisor'

static\_configs:

- targets: ['cadvisor:8080']

## 3. Визуализация в Grafana

В Grafana создано несколько дашбордов с метриками:

- **Процессор**: Нагрузка на CPU, вычисляемая через `rate(node_cpu_seconds_total{mode="user"}[1m])`.
- **Память**: Использование оперативной памяти.
- **Контейнеры**: Нагрузка на контейнеры через CAdvisor.

#### 4. Настройка HTTPS с помощью Traefik

- Traefik используется для маршрутизации трафика к сервисам.
- Настроены автоматические сертификаты через Let's Encrypt.
- Пример конфигурации `traefik.yml`:

```
entryPoints:
  web:
    address: ":80"
  websecure:
    address: ":443"

certificatesResolvers:
  le:
    acme:
      email: your-email@example.com
      storage: acme.json
      httpChallenge:
        entryPoint: web

http:
  routers:
    prometheus:
      rule: "Host(`prometheus.example.com`)"
      service: prometheus
      entryPoints:
        - websecure
      tls:
        certResolver: le
```

## 5. Настройка алертинга

Настроен алертинг через Alertmanager. Пример правила алерта в `alerts.yml`:

groups:

- name: example\_alerts

rules:

- alert: HighCPUUsage

expr: rate(node\_cpu\_seconds\_total{mode="user"}[1m]) > 0.5

labels:

severity: critical

annotations:

summary: "High CPU usage detected"

description: "CPU usage is above 50%."



## 6. Сохранение данных

Используются **volumes** для сохранения состояния сервисов при пересоздании:

- **Prometheus**: сохраняет историю метрик.
- **Grafana**: сохраняет созданные дашборды.
- **Alertmanager**: сохраняет настройки.

## Результаты

- Интерфейсы доступны по доменам через HTTPS.
- Prometheus собирает метрики со всех указанных источников.
- В Grafana созданы информативные дашборды.
- Настроен алертинг с использованием Alertmanager.
- При пересоздании проекта все данные и настройки сохраняются благодаря volumes.

## Вывод

Проект выполнен в полном соответствии с DoD. Настроена эффективная система мониторинга и оповещения.