



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения  
(ИиППО)

## **ПРАКТИЧЕСКАЯ РАБОТА №4**

по дисциплине «Технологии виртуализации клиент-серверных приложений»

Студент группы ИКБО-10-19

Дараган Федор Алексеевич

Руководитель практической работы

\_\_\_\_\_  
(подпись студента)  
ассистент Мельников Д.А.

\_\_\_\_\_  
(подпись руководителя)

Работа представлена

«\_\_» \_\_\_\_\_ 2022 г.

Допущен к работе

«\_\_» \_\_\_\_\_ 2022 г.

Москва 2022

## Оглавление

Практическая работа № 4: «Мониторинг приложения посредством Prometheus, Grafana, Zabbix, GrayLog в Docker-контейнерах».....	3
Цель работы.....	3
Ход работы.....	3
Вывод.....	14
Ответы на вопросы к практической работе.....	14
1. Назовите основные различия между Prometheus и Zabbix.....	14
2. Как можно запустить две базы PostgreSQL в одном docker-compose файле чтобы они работали одновременно и таблицы внутри не пересекались?.....	14
3. Назовите виды мониторинга систем.....	14
4. При помощи чего можно передавать конфигурационные переменные в контейнер?.....	15
5. Назовите основные различия Docker Swarm и Docker Compose.....	15
6. Назовите пример задачи, которую невозможно выполнить, используя один лишь docker-compose без Dockerfile.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	16

## **Практическая работа № 4: «Мониторинг приложения посредством Prometheus, Grafana, Zabbix, GrayLog в Docker-контейнерах»**

### **Цель работы**

Изучить и применить на практике технологию Docker-compose, Prometheus, Grafana, Zabbix, GrayLog.

### **Ход работы**

Был составлен docker-compose файл, содержащий следующие сервисы:

- БД приложения;
- Приложение;
- БД Zabbix;
- Zabbix-гейтвей для джавы — позволяет считывать статистику Приложения;
- Zabbix-сервер, обрабатывающий все данные;
- Zabbix-агент, для тестирования Zabbix;
- Zabbix веб-интерфейс;
- Prometheus;
- Prometheus exporter для считывания данных с БД приложения;
- БД Grafana;
- Grafana для визуализации данных из Prometheus;
- БД Graylog;
- Elasticsearch для Graylog, используемая для работы с данными;
- Filebeat для считывания данных с БД приложения для Graylog;
- Graylog;

На листинге 1 показан описанный файл.

Листинг 1 — docker-compose файл

```
version: '3.8'
```

```

services:

  app-db:
    container_name: app-db
    image: postgres:alpine
    volumes:
      -
      - ./app_db/tableCreation.sql:/docker-entrypoint-initdb.d/tableCreation.sql
      - app-db-data:/var/lib/postgresql/data
      - ./app_db/logs:/logs
    environment:
      POSTGRES_PASSWORD: admin
    command: >
      -c logging_collector=on
      -c log_directory=/logs
      -c log_filename=postgresql.log
      -c log_destination=csvlog
      -c log_statement=all

  app:
    container_name: app
    build:
      context: ./app
      target: prod
    volumes:
      - app-external:/app/external
    environment:
      JAVA_TOOL_OPTIONS: >
        -Dcom.sun.management.jmxremote.ssl=false
        -Dcom.sun.management.jmxremote.authenticate=false
        -Dcom.sun.management.jmxremote.port=5000
        -Dcom.sun.management.jmxremote.rmi.port=5000
        -Dcom.sun.management.jmxremote.host=0.0.0.0
        -Djava.rmi.server.hostname=app
      POSTGRES_HOSTNAME: app-db
      POSTGRES_PORT: 5432
      POSTGRES_DB: postgres
      POSTGRES_USER: app

```

```

    POSTGRES_PASSWORD: app
depends_on:
  - app-db
ports:
  - 8080:8080

zabbix-db:
  container_name: zabbix-db
  image: postgres:alpine
  volumes:
    - zabbix-db-data:/var/lib/postgresql/data
  environment:
    POSTGRES_USER: admin
    POSTGRES_PASSWORD: admin

zabbix-java-gateway:
  container_name: zabbix-java-gateway
  image: zabbix/zabbix-java-gateway:alpine-6.2-latest

zabbix-server:
  container_name: zabbix-server
  image: zabbix/zabbix-server-pgsql:alpine-6.2-latest
  environment:
    ZBX_HOSTNAME: zabbix-server
    DB_SERVER_HOST: zabbix-db
    POSTGRES_USER: admin
    POSTGRES_PASSWORD: admin
    ZBX_JAVAGATEWAY_ENABLE: true
    ZBX_JAVAGATEWAY: zabbix-java-gateway
  depends_on:
    - zabbix-db
    - zabbix-java-gateway
  restart: unless-stopped

zabbix-agent:
  container_name: zabbix-agent
  image: zabbix/zabbix-agent:alpine-6.2-latest

```

```

privileged: true
environment:
  ZBX_HOSTNAME: zabbix-agent
  ZBX_SERVER_HOST: zabbix-server
depends_on:
  - zabbix-server
restart: unless-stopped

zabbix-web-interface:
  container_name: zabbix-web-interface
  image: zabbix/zabbix-web-nginx-pgsql:alpine-6.2-latest
  environment:
    ZBX_HOSTNAME: zabbix-web-interface
    DB_SERVER_HOST: zabbix-db
    POSTGRES_USER: admin
    POSTGRES_PASSWORD: admin
    ZBX_SERVER_HOST: zabbix-server
  depends_on:
    - zabbix-server
  ports:
    - 80:8080
  restart: unless-stopped

prometheus:
  container_name: prometheus
  image: prom/prometheus:latest
  volumes:
    - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
  ports:
    - 9090:9090

postgres-exporter:
  container_name: postgres-exporter
  image: prometheuscommunity/postgres-exporter:latest
  environment:
    DATA_SOURCE_NAME: "postgresql://app:app@app-db:5432/postgres?sslmode=disable"
  depends_on:

```

- app-db
- prometheus

grafana:

```
container_name: grafana
image: grafana/grafana:latest
volumes:
  - ./grafana/data:/var/lib/grafana/
ports:
  - 3000:3000
depends_on:
  - postgres-exporter
```

graylog-mongo:

```
container_name: graylog-mongo
image: bitnami/mongodb:latest
volumes:
  - graylog-mongo-data:/bitnami/mongodb
environment:
  MONGODB_ROOT_PASSWORD: 123
  MONGODB_USERNAME: graylog
  MONGODB_PASSWORD: graylog
  MONGODB_DATABASE: graylog
```

graylog-elasticsearch:

```
container_name: graylog-elasticsearch
image: bitnami/elasticsearch:7.17.7
environment:
  - http.host=0.0.0.0
  - transport.host=localhost
  - network.host=0.0.0.0
  - xpack.security.enabled=false
  - node.max_local_storage_nodes=3
  - "ES_JAVA_OPTS=-Dlog4j2.formatMsgNoLookups=true -Xms512m -Xmx512m"
  - bootstrap.memory_lock=true
  - discovery.type=single-node
ulimits:
```

```

    memlock:
      soft: -1
      hard: -1
  deploy:
    resources:
      limits:
        memory: 1g

graylog-filebeat:
  container_name: graylog-filebeat
  image: elastic/filebeat:8.5.1
  entrypoint: filebeat -e -strict.perms=false
  user: root
  volumes:
    - ./filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml
    - ./app_db/logs:/var/app/log

graylog:
  container_name: graylog
  image: graylog/graylog:4.3
  environment:
    GRAYLOG_PASSWORD_SECRET: somepasswordpepper
    # Password: admin
    GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
    GRAYLOG_HTTP_EXTERNAL_URI: http://127.0.0.1:9000/
    GRAYLOG_WEB_ENDPOINT_URI: http://127.0.0.1:9000/api
    GRAYLOG_ELASTICSEARCH_HOSTS: http://graylog-elasticsearch:9200
    GRAYLOG_MONGODB_URI:
mongodb://graylog:graylog@graylog-mongo:27017/graylog
    GRAYLOG_MESSAGE_JOURNAL_MAX_SIZE: 128mb
    # Content pack options
    GRAYLOG_CONTENT_PACKS_AUTO_INSTALL: content-pack-a11181bd-ab8e-4a6b-
8153-99452fe8a521-1.json
    GRAYLOG_CONTENT_PACKS_DIR: data/contentpacks
    GRAYLOG_CONTENT_PACKS_LOADER_ENABLED: true
  entrypoint: /usr/bin/tini -- wait-for-it graylog-elasticsearch:9200 --
/docker-entrypoint.sh
  volumes:

```



```

    - ./graylog/contentpacks:/usr/share/graylog/data/contentpacks
restart: always
depends_on:
  - graylog-mongo
  - graylog-elasticsearch
  - graylog-filebeat
ports:
  # Graylog web interface and REST API
  - 9000:9000

volumes:
  app-db-data:
  app-external:
  zabbix-db-data:
  graylog-mongo-data:

```

Приложение имеет двухступенчатую сборку в Dockerfile, аналогичную практическому заданию 3. При помощи `java gmi` оно передает свои характеристики Zabbix. Приложение представляет собой интернет-магазин, являющийся прошлой курсовой работой автора.

По сути `filebeat` является не самым элегантным решением, поскольку он просто копирует файл с логами из PostgreSQL в GrayLog при помощи общего примонтированного хранилища. На листинге 2 показан конфигурационный файл для `filebeat`.

## Листинг 2 — `filebeat.yml`

```

filebeat.config:
  modules:
    path: ${path.config}/modules.d/*.yaml
    reload.enabled: false

filebeat.modules:
  - module: postgresql
    log:
      enabled: true
      var.paths: ["/var/app/log/*.csv*"]

output.logstash:

```

```
hosts: ["graylog:5555"]
```

Для сохранения настроенной конфигурации Grafana в папку с проектом сохраняется дамп ее БД. К сожалению, функции импорта и экспорта настроек в данном контейнере не предусмотрено его создателем.

В случае с GrayLog сохранение конфигурации возможно при помощи строчек конфигурации, показанных на листинге 3. Однако экспорт настроек производится вручную в отдельный файл в папку contentpacks. Файл достаточно большой и малоинформативный, поэтому представлен не будет.

### Листинг 3 — Отрывок из docker-compose файла

```
# ...
  environment:
    # ...
    GRAYLOG_CONTENT_PACKS_AUTO_INSTALL: content-pack-a11181bd-ab8e-4a6b-
8153-99452fe8a521-1.json
    GRAYLOG_CONTENT_PACKS_DIR: data/contentpacks
    GRAYLOG_CONTENT_PACKS_LOADER_ENABLED: true
  volumes:
    - ./graylog/contentpacks:/usr/share/graylog/data/contentpacks
# ...
```

На листинге 4 показан файл конфигурации для Prometheus.

### Листинг 4 — Файл конфигурации Prometheus

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: prometheus
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: postgres-exporter
    static_configs:
      - targets: ["postgres-exporter:9187"]
```

Для Zabbix импорт конфигурации аналогичен GrayLog, однако здесь файл в разы меньше и представлен на листинге 5.

### Листинг 5 — Файл конфигурации Zabbix

```
zabbix_export:
  version: '6.2'
  date: '2022-11-17T08:27:28Z'
  host_groups:
    -
      uuid: a571c0d144b14fd4a87a9d9b2aa9fcd6
      name: Applications
    -
      uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix servers'
  hosts:
    -
      host: app
      name: app
      templates:
        -
          name: 'Generic Java JMX'
      groups:
        -
          name: Applications
      interfaces:
        -
          type: JMX
          useip: 'NO'
          dns: app
          port: '5000'
          interface_ref: if1
      inventory_mode: DISABLED
    -
      host: zabbix-agent
      name: zabbix-agent
      templates:
        -
          name: 'Linux by Zabbix agent'
```

```
groups:
-
  name: 'Zabbix servers'
interfaces:
-
  useip: 'NO'
  dns: zabbix-agent
  interface_ref: if1
inventory_mode: DISABLED
```

На рисунке 1 показано приложение.

[Главная](#) [Магазины](#) [Товары](#) [Войти](#) [Регистрация](#)

Данный сайт предоставляет возможность для управления комплексом магазинов.

Есть возможность для управления магазинами, если зайти на сайт под администраторским аккаунтом.

Администратор может:

- Добавлять новые магазины
- Удалять существующие
- Добавлять новые товары
- Добавлять товары в магазины и редактировать их количество
- Принимать заказы от пользователей

Зарегистрированный пользователь может:

- Создавать новые заказы
- Удалять свои заказы
- Переименовывать свои заказы
- Добавлять товары из магазинов в свой заказ
- Отправлять готовый заказ администратору
- Возвращать непроверенный заказ обратно, для дальнейшего редактирования

Любой посетитель может:

- Зарегистрировать новый аккаунт
- Просматривать списки магазинов
- Просматривать содержимое магазинов
- Просматривать список всех товаров
- Просматривать подробную информацию о товарах
- Заходить в аккаунт с помощью логина и пароля

Количество товаров в заказе связано с количеством в магазине.

Рисунок 1. Скриншот приложения

На рисунке 2 показан интерфейс Graylog.

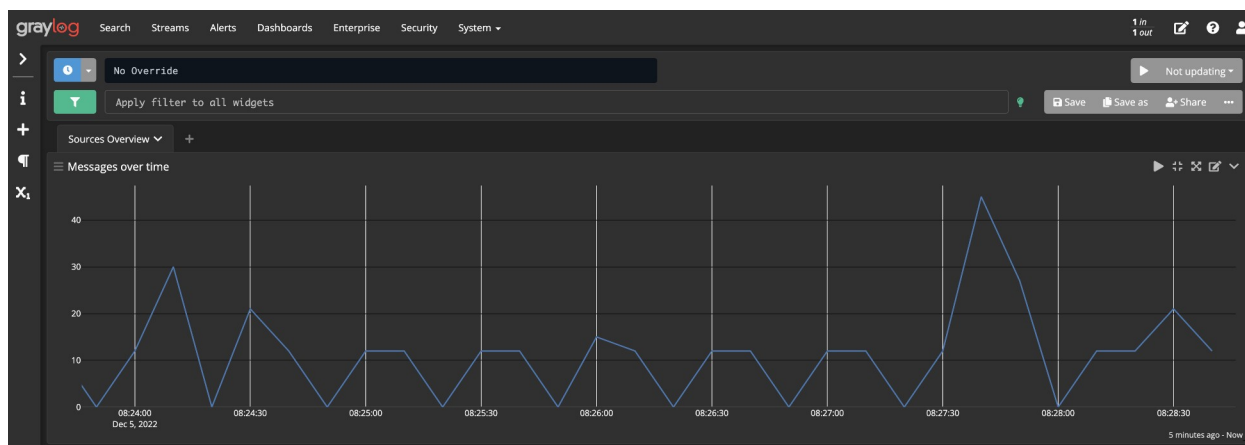


Рисунок 2. Скриншот интерфейса GrayLog

На рисунке 3 показан интерфейс Zabbix.

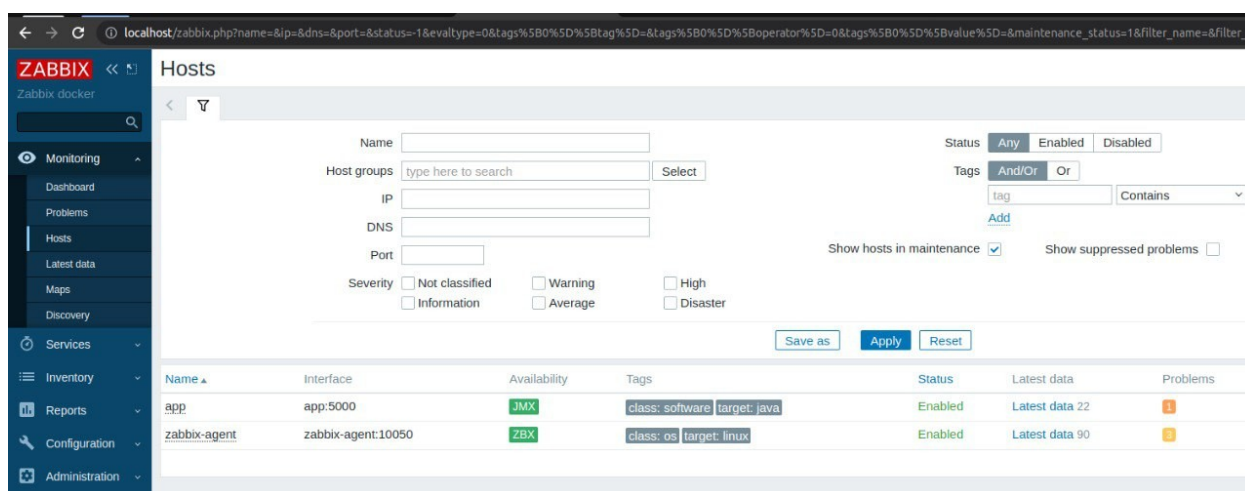


Рисунок 3. Скриншот интерфейса Zabbix

На рисунке 4 показан интерфейс Grafana.

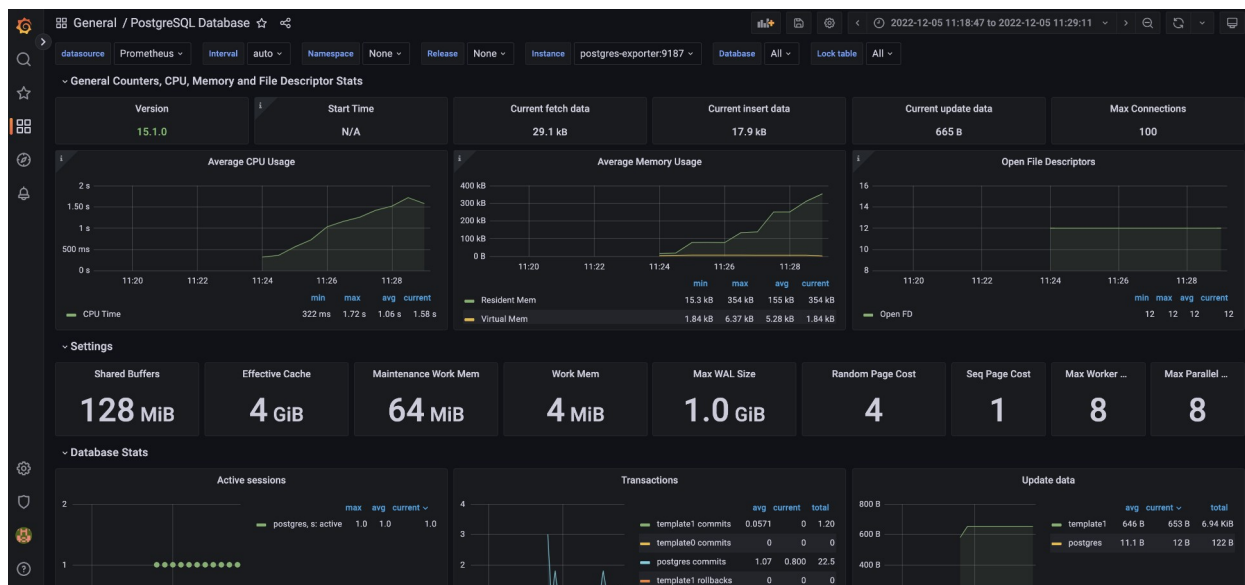


Рисунок 4. Скриншот интерфейса Grafana

## Вывод

В ходе выполнения практической работы №4 была изучена и применена на практике технология Docker-compose. Были использованы Prometheus, Grafana, Zabbix, GrayLog.

## Ответы на вопросы к практической работе

### 1. Назовите основные различия между Prometheus и Zabbix.

Zabbix имеет встроенную систему аутентификации и авторизации, Zabbix включает в себя встроенную панель мониторинга, когда в Prometheus необходима интеграция с Grafana.

### 2. Как можно запустить две базы PostgreSQL в одном docker-compose файле чтобы они работали одновременно и таблицы внутри не пересекались?

Необходимо запускать их в разных контейнерах с разными портами.

### 3. Назовите виды мониторинга систем.

- Инфраструктурный мониторинг (например, Zabbix, Nagios);
- Мониторинг ошибок программных платформ (Sentry);

- Мониторинг производительности приложений (Prometheus);
- Мониторинг безопасности систем (Nessus, OpenVas);
- Сбор системных журналов (GrayLog);

**4. При помощи чего можно передавать конфигурационные переменные в контейнер?**

При помощи переменных окружения, устанавливаемых ключевым словом `environment`.

**5. Назовите основные различия Docker Swarm и Docker Compose**

Docker Compose предназначен для запуска и управления группой связанных контейнеров. Docker Swarm предназначен для управления группой запущенных Docker Engine как единым целым и оркестровки.

**6. Назовите пример задачи, которую невозможно выполнить, используя один лишь docker-compose без Dockerfile**

В docker-compose без Dockerfile нельзя в несколько стадий выполнить сборку образа контейнера.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Виртуализация — Википедия — URL: <https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- Open Virtualization Format — Википедия — URL: [https://ru.wikipedia.org/wiki/Open\\_Virtualization\\_Format](https://ru.wikipedia.org/wiki/Open_Virtualization_Format) (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- Файл VMX – чем открыть, описание формата — URL: <https://open-file.ru/types/vmx> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- ISO-образ — Википедия — URL: <https://ru.wikipedia.org/wiki/ISO-%D0%BE%D0%B1%D1%80%D0%B0%D0%B7> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.