Онлайн-школа «SkillFactory»

Итоговый проект
курса «Системный администратор linux»

Выполнил:
студент группы A1008
Слышинский А.В.

2023

Оглавление

1. Инфрастуктура и первоначальная конфигурация ВМ

Для разворачивания стендов использовалась облачная платформа Yandex Cloud.

Для базовой ВМ были выбраны следующие характеристики:

Характеристика	ВМ
Платформа	Intel Cascade Lake
Гарантированная доля vCPU	5%
vCPU	2
RAM	2 ГБ
Объём дискового пространства	15 ГБ

В процессе эксплуатации характеристики менялись в соответствии с нагрузкой и возможностями оплаты за сервисы, т. к. выданный грант был исчерпан.

Итоговые характеристики:

Характеристика	ВМ-1	ВМ-2	ВМ-3
Платформа	Intel Cascade Lake	Intel Cascade Lake	Intel Cascade Lake
Гарантированная доля vCPU	20%	5%	20%
vCPU	2	2	4
RAM	3 ГБ	2 ГБ	6 ГБ
Объём дискового пространства	15 ГБ	15 ГБ	15 ГБ

2. Установка и настройка ПО с использованием ansible

Управление ВМ производилось с рабочего хоста студента(т.к. было очень нестабильное подключение к сети Internet со стороны провайдера) и с хоста ВМ-1.

2.1. Выполненные плейбуки и роли с рабочего хоста студента:

2.1.1. Для всех ВМ:

а. После создания ВМ, с хоста студента, к ним были применен asible-playbook, в котором происходило:

- обновление ПО
- изменение конфигурации ssh-сервера
- изменение настройки даты и времени
- создание нового пользователя, владельцф директории общего доступа, без возможности входа
- создание директории для общего пользования
- создание пользователя mentor с паролем и домашним окружением, с добавлением открытого ssh-ключа, и с добавлением пользователя в группу sudo
- добавление обычных пользователей в группу владельца директории общего доступа
- установка дополнительное ПО: vim, mc

б. Metricbeat: установка и настройка через создание конфигураций из шаблонов, запуск сервиса.

2.1.2. Для ВМ-2 и ВМ-3:

а. Openvpn: установка и настройка клиентов:

- установка Openvpn
- установка и настройка resolvconf:
- скачивание конфигураций клиентов с управляющей ноды
- запуск сервиса для создания VPN-подключения

2.1.3. Для ВМ-1:

а. Openvpn: установка и настройка сервера:

- установка сервера Openvpn
- выпуск сертификтов CA, сервера и клиентов
- создание конфигураций, из шаблонов, для сервера и клиентов
- запуск сервиса для создания VPN-подключения
- скачивание конфигураций клиентов на управляющую ноду

б. Grafana: установка и настройка сервера, создание конфигураций из шаблонов и запуск сервиса.

2.1.4. для ВМ-2 :

а. Bind9, Apache, PHP, Nginx, Filebeat: установка и настройка, создание конфигураций из шаблонов и запуск сервиса.

б. Grafana и Kibana: создание конфигураций для apache(backend) и nginx(frontend), активация конфигурации.

в. Postfix и Dovecot: установка.

г Roundcubemail: установка и настройка, создание конфигураций для apache(backend) и nginx(frontend), активация конфигурации.

2.1.5. Для ВМ-3:

а. Установка и настройка стека ELK:

- Logstash: установка
- Elasticsearch и Kibana: установка и настройка, создание конфигурации из шаблонов и запуск сервиса

2.2. Выполненные плейбуки и роли с ВМ-1(внутри VPN):

2.2.1. Для всех ВМ:

Zabbix-агент: установка и настройка через создание конфигураций из шаблонов, запуск сервиса.

2.2.2. Для ВМ-1(localhost):

Zabbix-сервер: установка и настройка через создание конфигураций из шаблонов, запуск сервиса.

2.2.3. для ВМ-2 :

Веб-доступ к Zabbix-серверу: установка и настройка zabbix-frontend-php, создание конфигураций для apache(backend) и nginx(frontend), активация конфигурации.

2.2.4. Для ВМ-3:

а. Установка и настройка Postgresql:

- установка Postgresql
- создание пользователя и БД из шаблона для zabbix и roundcube, настройка доступов

Для хранения ролей, плейбуков и дополнительных файлов, использовался закрытый репозиторий на GitHub`e, в котором было 2 ветки:

- master - для плейбуков и ролей, выполняемых на всех ВМ и управления с рабочего хоста студента
- srv1 - для плейбуков и ролей, выполняемых на 2-х ВМ и управления с ноды внутри VPN

3. Сетевое взаимодействие

На ВМ-1 был установлен и настроен OpenVPN-сервер, остальные ВМ выступали в роли клиентов.

В конфигурациях использовался виртуальный сетевой интерфейс tap и протокол tcp.

Между собой ВМ взаимодействовали по VPN.

На ВМ-2 был установлен bind9(DNS-сервер).

Для внутреннего домена использовалось имя vpn.srv-test.space. Домен srv-test.space арендуется студентом.

Таблица сетевых идентификаторов ВМ

ВМ	IP внутри VPN	Имя хоста	FQDN
ВМ-1	10.20.30.1	srv1	srv1.vpn.srv-test.space
ВМ-2	10.20.30.2	srv2	srv2.vpn.srv-test.space
ВМ-3	10.20.30.3	srv3	srv3.vpn.srv-test.space

Результаты команды ping, выполненной с ВМ-1 на все ВМ

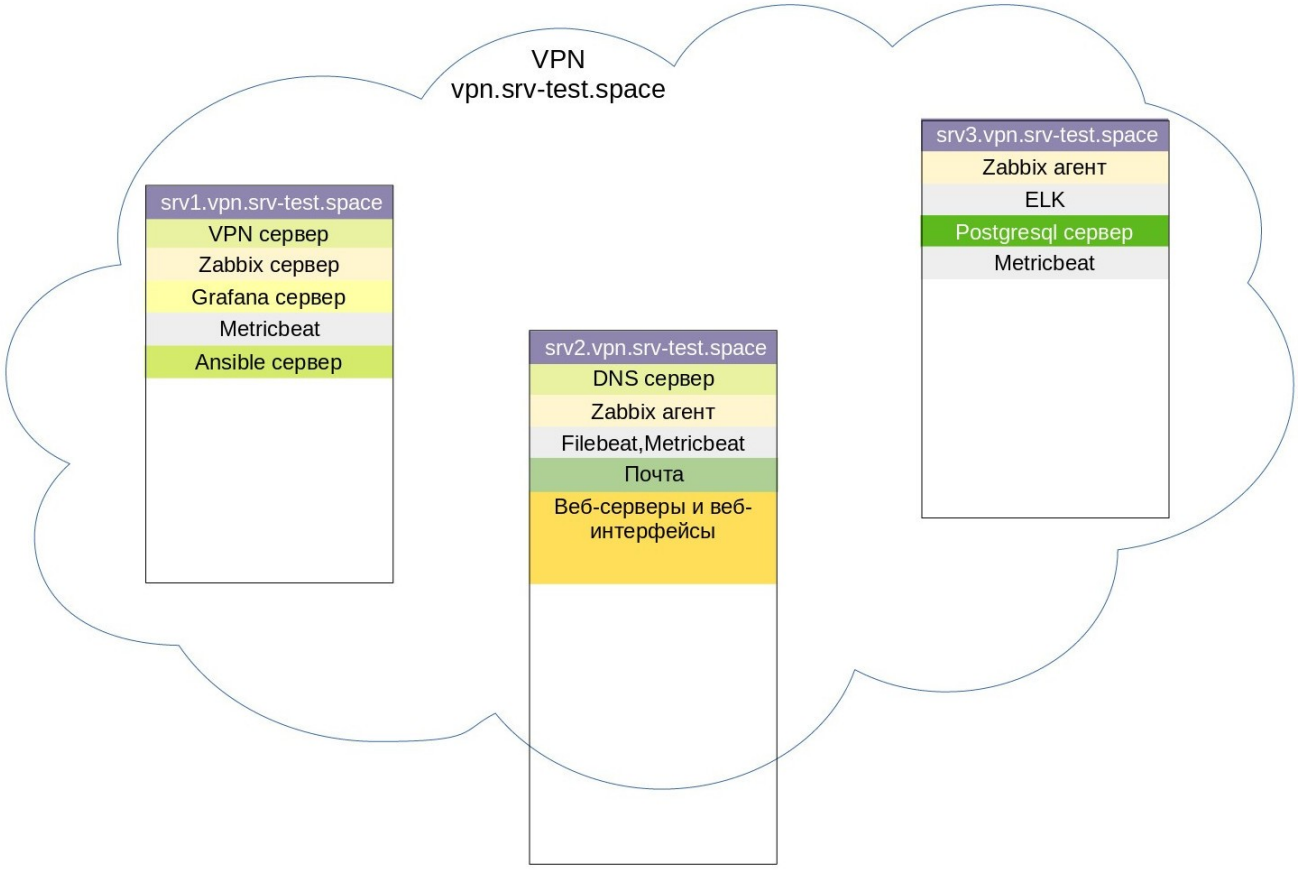
```
qw9_5va@srv1:~$ ping -c 1 srv1.vpn.srv-test.space
PING srv1.vpn.srv-test.space (10.20.30.1) 56(84) bytes of data.
64 bytes from srv1.vpn.srv-test.space (10.20.30.1): icmp_seq=1 ttl=64 time=0.020 ms

--- srv1.vpn.srv-test.space ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.020/0.020/0.020/0.000 ms
```

```
qw9_5va@srv1:~$ ping -c 1 srv2.vpn.srv-test.space
PING ns1.vpn.srv-test.space (10.20.30.2) 56(84) bytes of data.
64 bytes from ns1.vpn.srv-test.space (10.20.30.2): icmp_seq=1 ttl=64 time=0.252 ms

--- ns1.vpn.srv-test.space ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.252/0.252/0.252/0.000 ms
```

```
qw9_5va@srv1:~$ ping -c 1 srv3.vpn.srv-test.space
PING srv3.vpn.srv-test.space (10.20.30.3) 56(84) bytes of data.
64 bytes from srv3.vpn.srv-test.space (10.20.30.3): icmp_seq=1 ttl=64 time=0.692 ms

--- srv3.vpn.srv-test.space ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.692/0.692/0.692/0.000 ms
```

4. Общее описание стенда

Распределение сервисов по ВМ



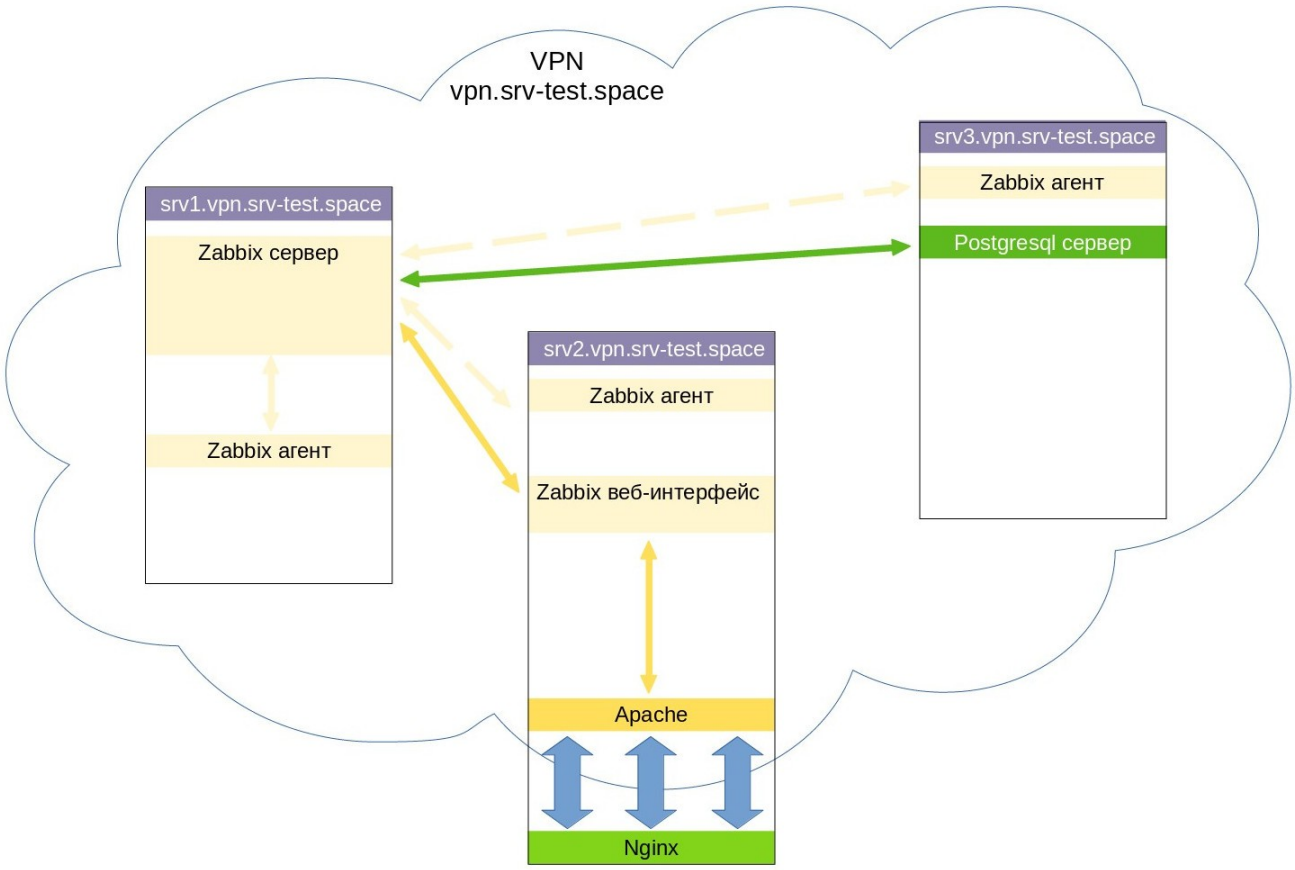
ВМ-1: OpenVPN-server, Zabbix-сервер и Zabbix-агент, Grafana-сервер, Ansible-сервер, Metricbeat.

ВМ-2: Bind9(DNS-сервер), веб-серверы Apache(бэкенд) и Nginx(фронтенд), Почта, Zabbix-агент, Filebeat и Metricbeat.

ВМ-3: PostgreSQL, ELK, Zabbix-агент, Filebeat и Metricbeat.

5. Zabbix

Схема «Zabbix»



Общее описание:

Сервер Zabbix, был запущен на ВМ-1 и получал данные от агентов запущенных на всех ВМ.

Так же сервер был связан с базой данных, находящейся под управлением СУБД PostgreSQL и расположенной на ВМ-3.

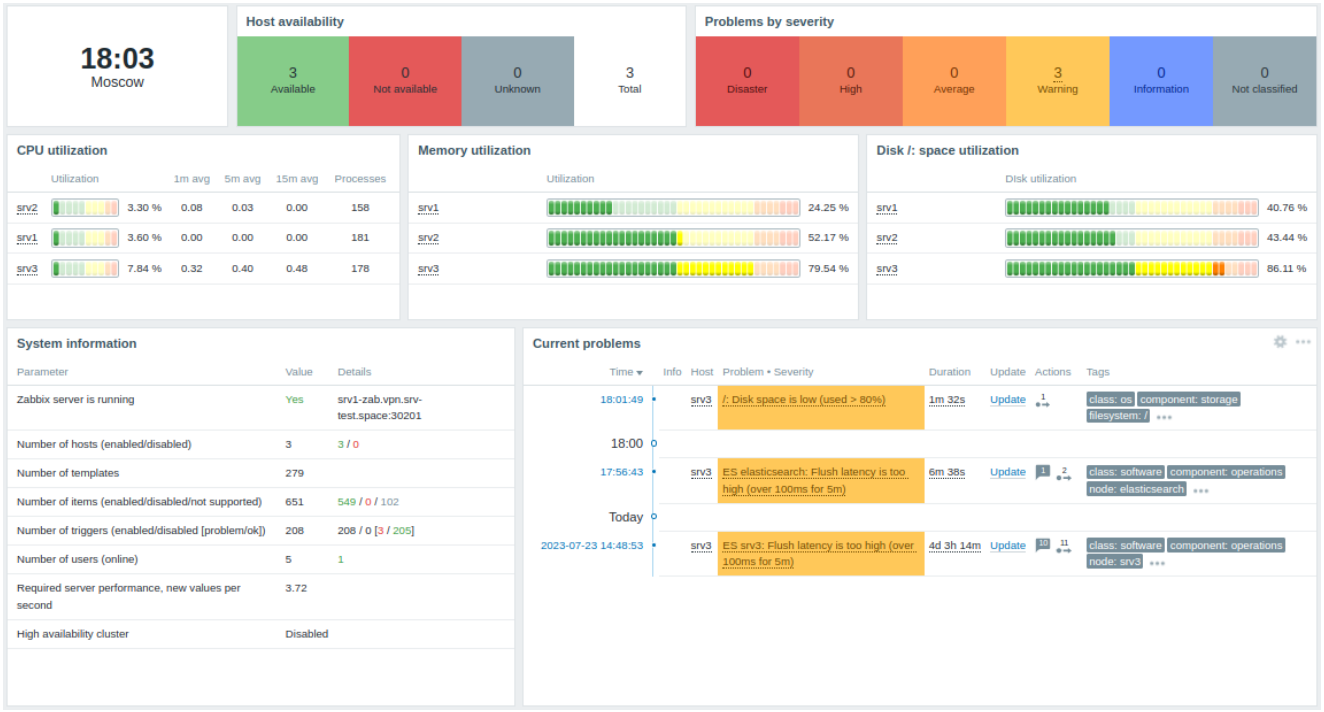
В случае возникновения проблем, было настроено оповещение в Telegram.

Доступ к веб-интерфейсу обеспечивался ВМ-2 через настройку конфигураций для Zabbix(zabbix-frontend-php), Apache и Nginx.

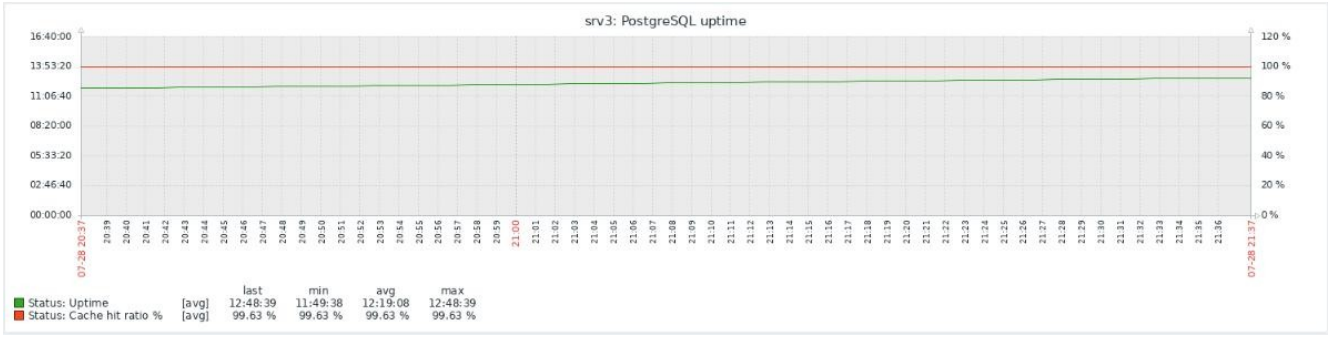
Сервер Zabbix был установлен 2 способами:

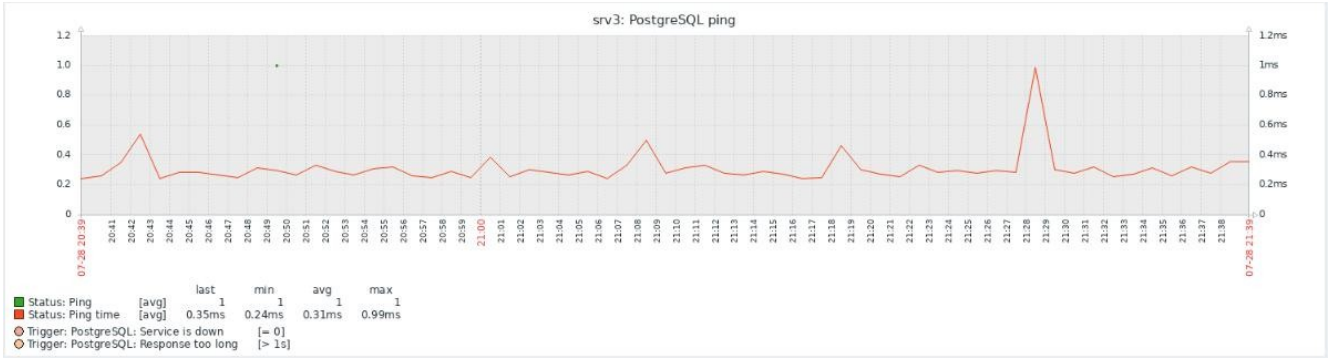
- через роль ansible
- с использованием технологии «docker»

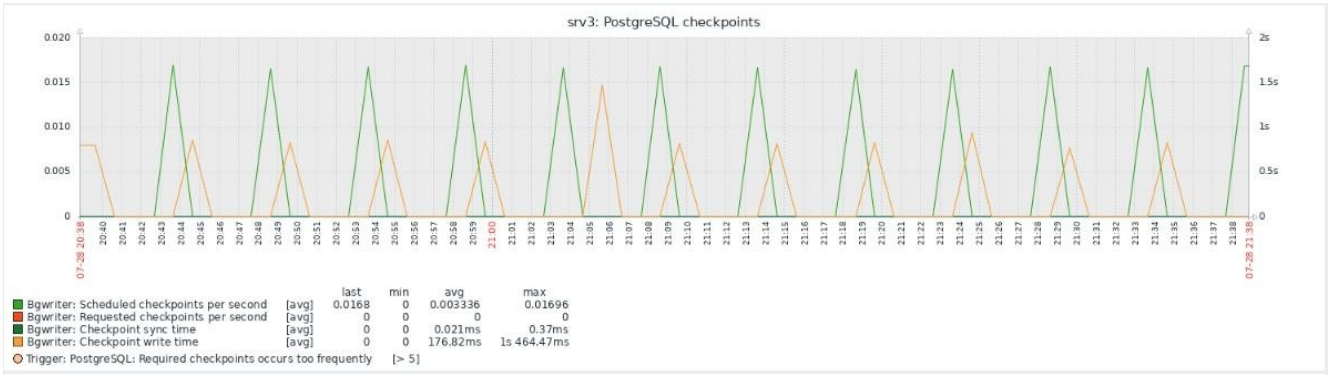
Мониторинг основных всех ВМ в Zabbix



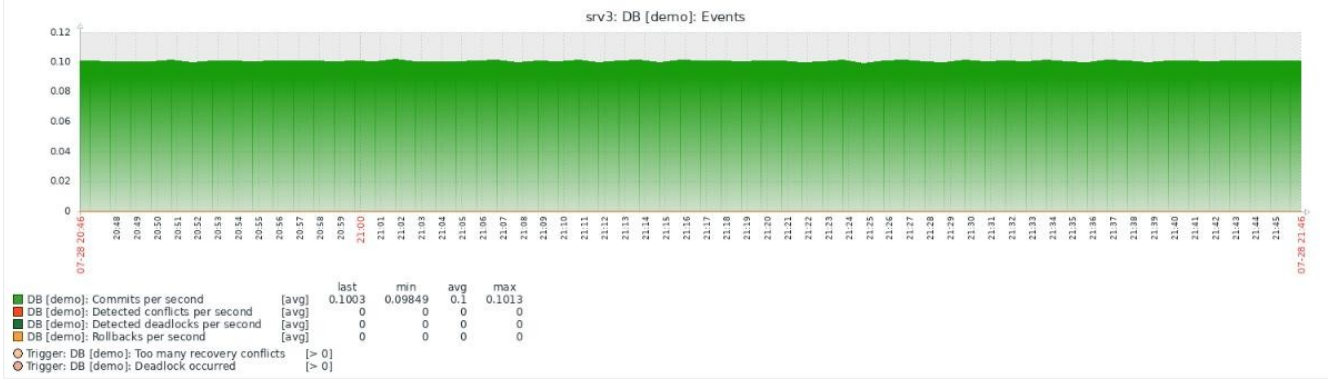
Мониторинг состояния PostgreSQL в Zabbix

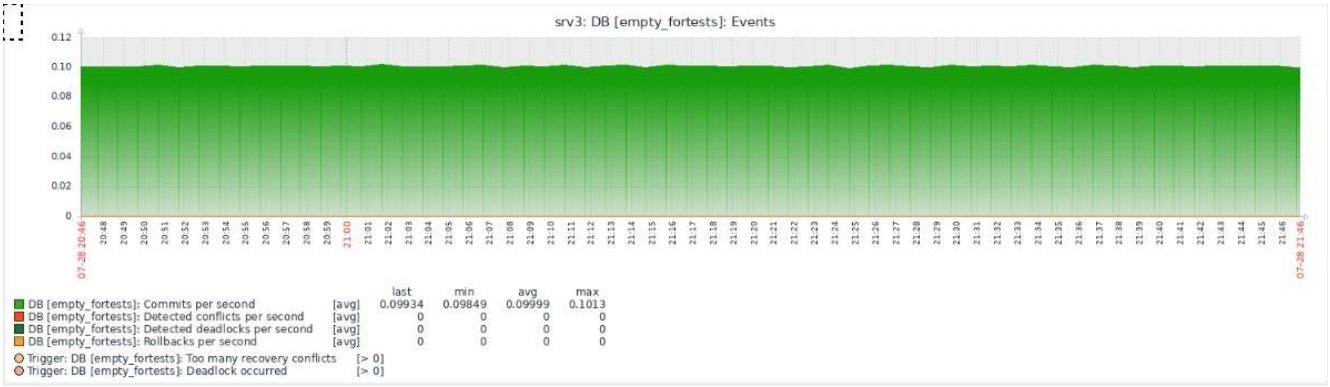


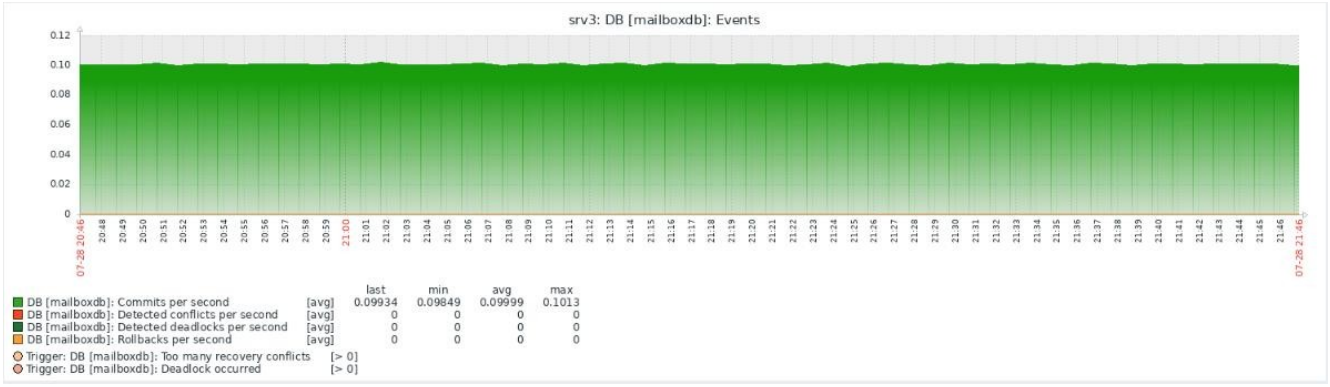


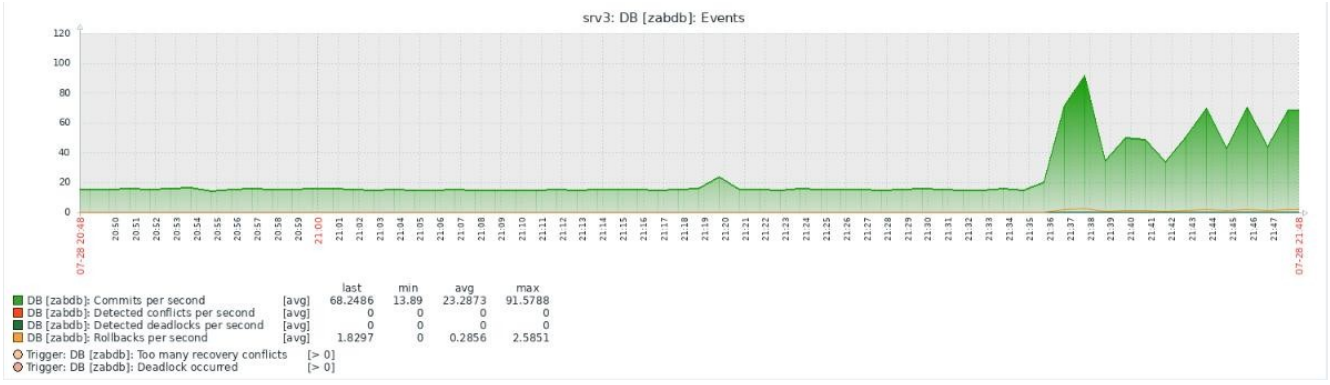


Мониторинг состояния баз данных PostgreSQL в Zabbix

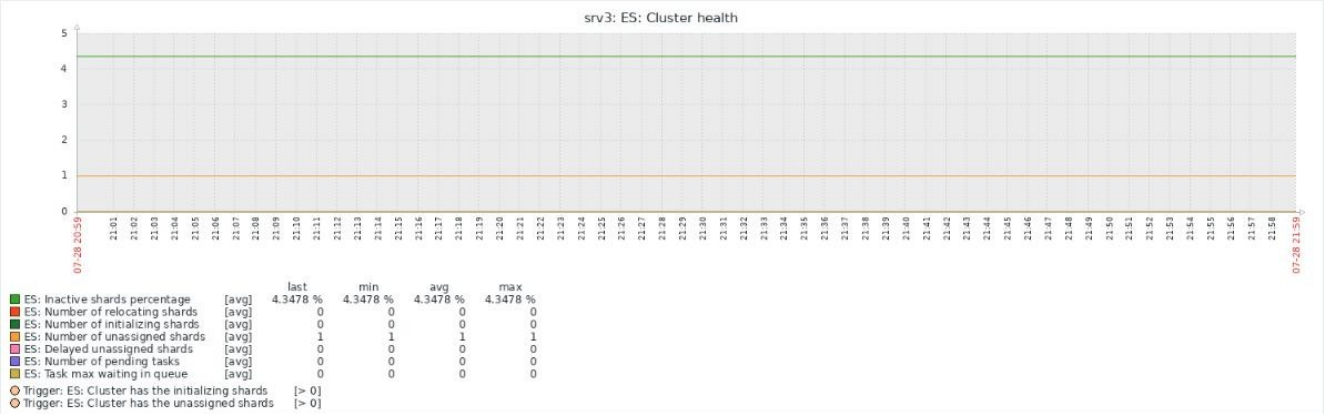


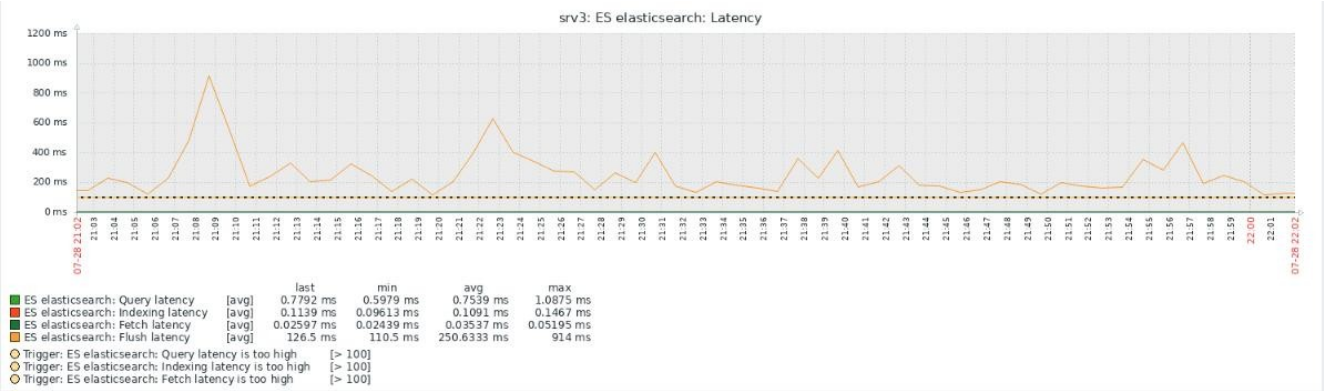






Мониторинг состояния ELK в Zabbix



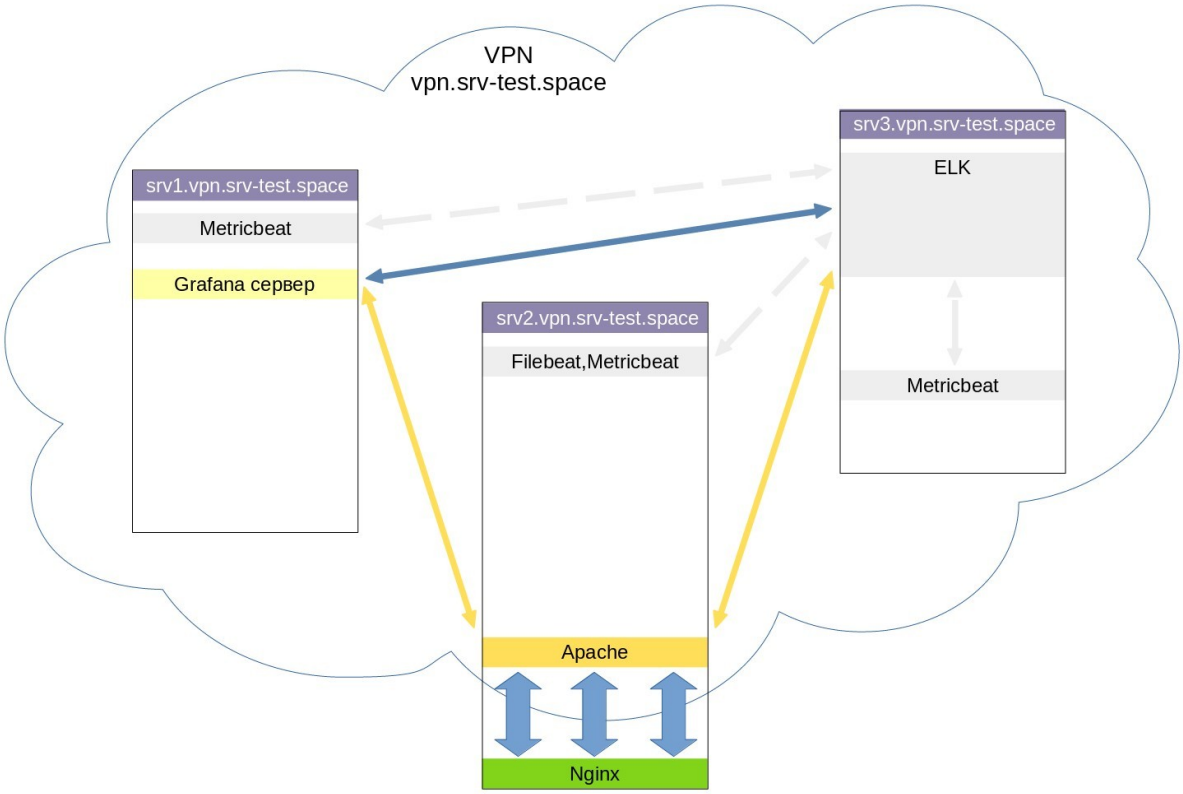


Создание контейнера для Zabbix-сервера

```
/home/share_local/git/SFadmin_Fi-Pr_sav/docker/zabbix/docker-compose.yaml
version: "3.3"
services:
  zabbix-server:
    image: zabbix/zabbix-server-pgsql:ubuntu-6.4-latest
    container_name: zabbix-server
    hostname: zabbix-server
    restart: unless-stopped
    environment:
      - TZ=Europe/Moscow
      - ZBX_SERVER_HOST=zab-srv
      - ZBX_LISTENPORT=30201
      - DB_SERVER_HOST=10.20.30.3
      - POSTGRES_DB=zabdb
      - POSTGRES_USER=lk534
      - POSTGRES_PASSWORD=iruKnmlxddOwNDkKtnw
    ports:
      - 30201:30201
```

6. ELK

Схема «ELK»



Общее описание:

Сервер Elasaticsearch был запущен на ВМ-3 и получал данные от:

- Metricbeat запущенном на всех ВМ
- Filebeat запущенном на ВМ-2

Для визуализации данных использовалась:

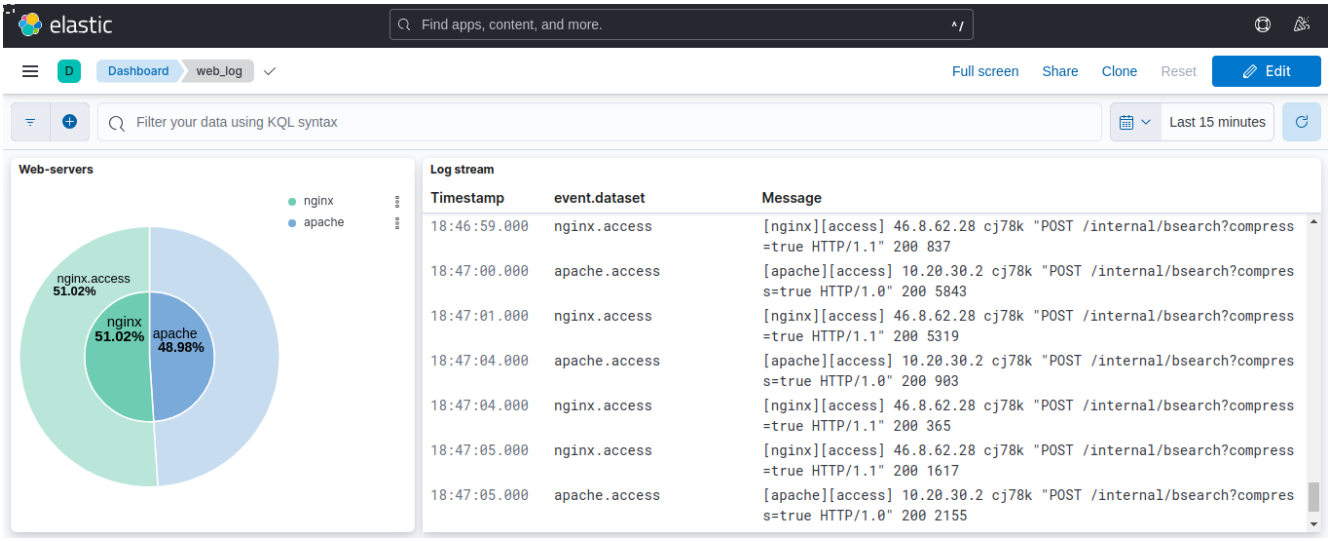
- Kibana, запущеная на ВМ-3
- Grafana, запущеная на ВМ-1

Доступ к веб-интерфейсу обеспечивался ВМ-2 через настройку конфигураций для Apache и Nginx.

Сервер Elasaticsearch и Kibana были установлены 2 способами:

- через роль ansible
- с использованием технологии «docker»

Отображение статистики и логов Apache и Nginx в Kibana



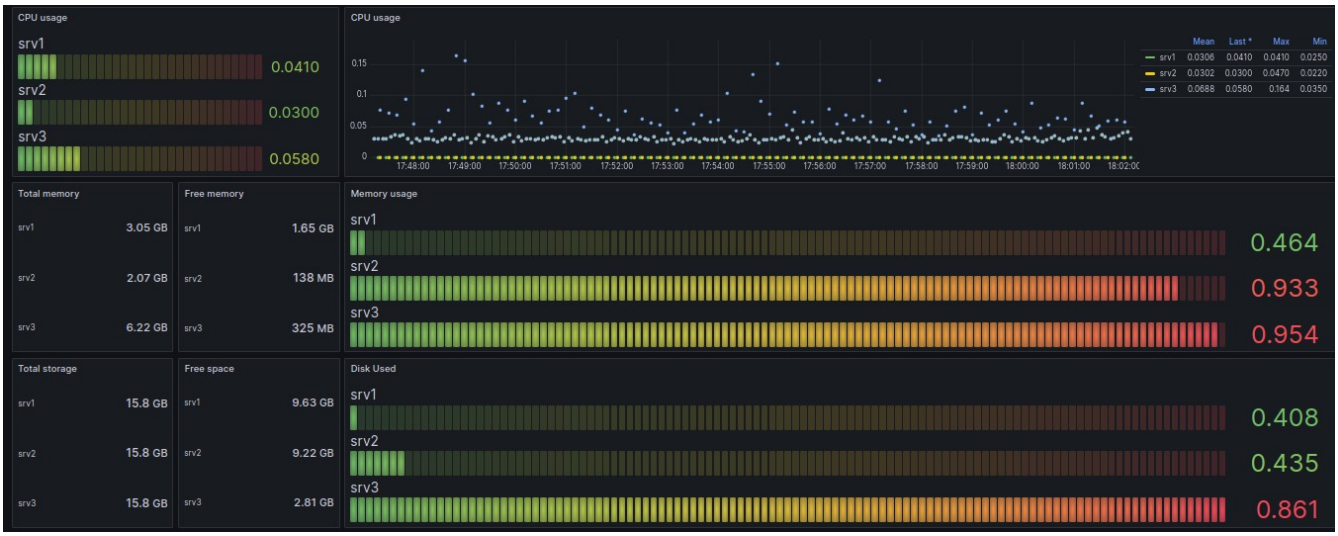
Создание контейнеров для Elasticsearch и Kibana

/home/share_local/docker/elk/docker-compose.yaml

```
version: "3.3"
services:
  elasticsearch:
    image: elasticsearch:8.8.1
    container_name: elasticsearch
    hostname: elasticsearch
    restart: unless-stopped
    environment:
      - TZ=Europe/Moscow
    ports:
      - 9200:9200
    volumes:
      - ./config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml

  kibana:
    image: kibana:8.8.1
    container_name: kibana
    hostname: kibana
    restart: unless-stopped
    environment:
      - TZ=Europe/Moscow
    ports:
      - 5601:5601
    volumes:
      - ./config/kibana.yml:/etc/kibana/kibana.yml
```

7. Grafana

Дашбоард с мониторингом CPU, mem, disk usage



Общее описание:

Grafana была установлена 2 способами:

- через роль ansible
- с использованием технологии «docker»

В случае возникновения проблем было настроено оповещение в Telegram.

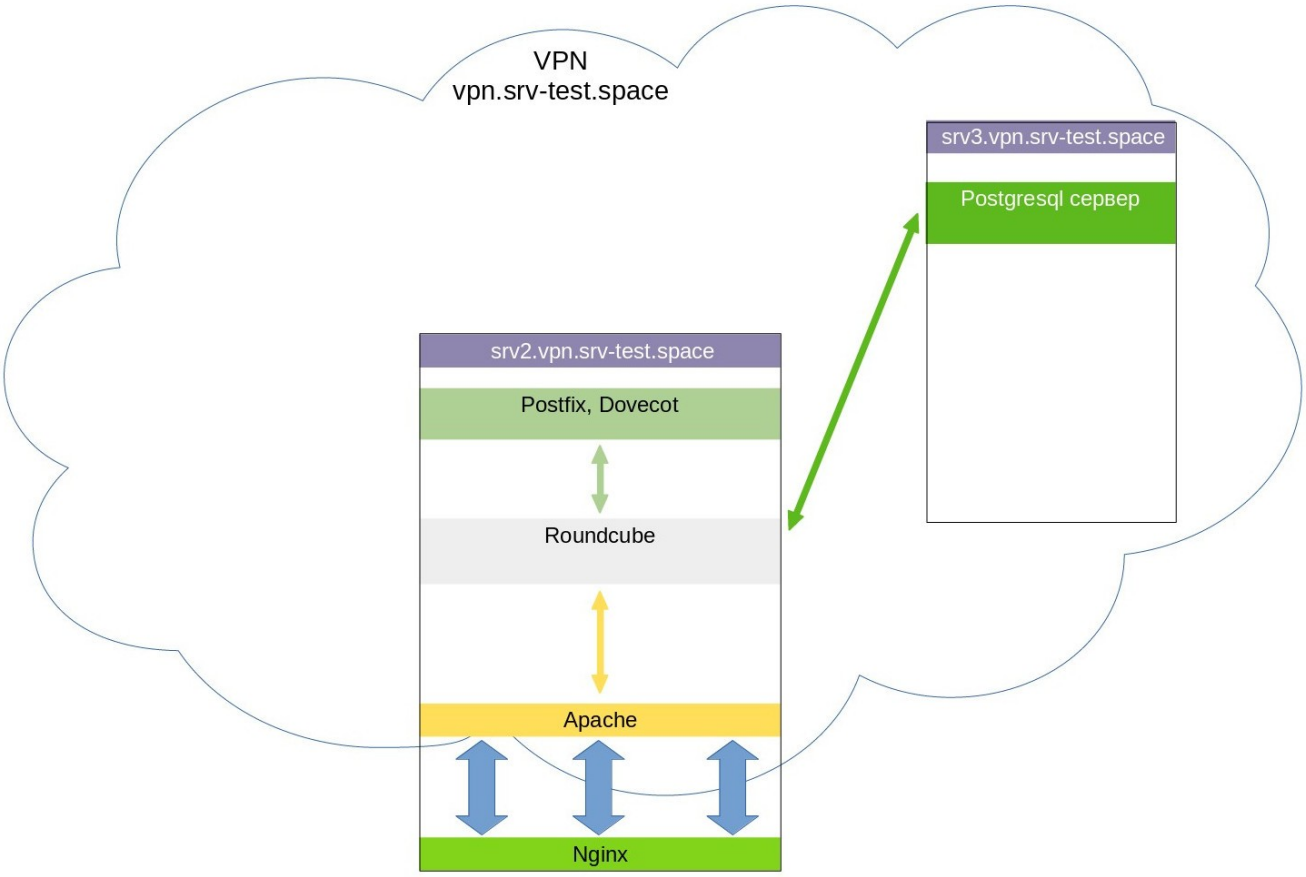
Доступ к веб-интерфейсу обеспечивался ВМ-2 через настройку конфигураций для Apache и Nginx.

Создание контейнера для Grafana, с переносом datasource и dashboard. Настройка алертингов проводилась через веб-интерфейс.

```
/home/share_local/git/SFadmin_Fi-Pr_sav/docker/grafana/docker-compose.yaml
version: '3.3'
services:
  grafana:
    image: grafana/grafana-enterprise:10.0.1-ubuntu
    container_name: grafana
    restart: unless-stopped
    environment:
      - TZ=Europe/Moscow
      - GF_SECURITY_ADMIN_USER=gho856t
      - GF_SECURITY_ADMIN_PASSWORD=YVtW1SwH+KIghEtgAun1U
      - GF_SERVER_ROOT_URL=http://srv1-graf.vpn.srv-test.space/
      - GF_USERS_ALLOW_SIGN_UP=false
      #- GF_INSTALL_PLUGINS=alexanderzobnin-zabbix-app
    ports:
      - 3000:3000
    volumes:
      - ./config/datasources/datasources_elk.yaml:/etc/grafana/provisioning/datasources/elk.yaml
      - ./config/dashboard.yaml:/etc/grafana/provisioning/dashboards/main.yaml
      - ./config/dashboards:/var/lib/grafana/dashboards
```

8. Почта

Схема «Почта».



Общее описание:

Для обеспечения доставки и получения электронных почтовых сообщений использовались Postfix и Dovecot установленные на ВМ-2.

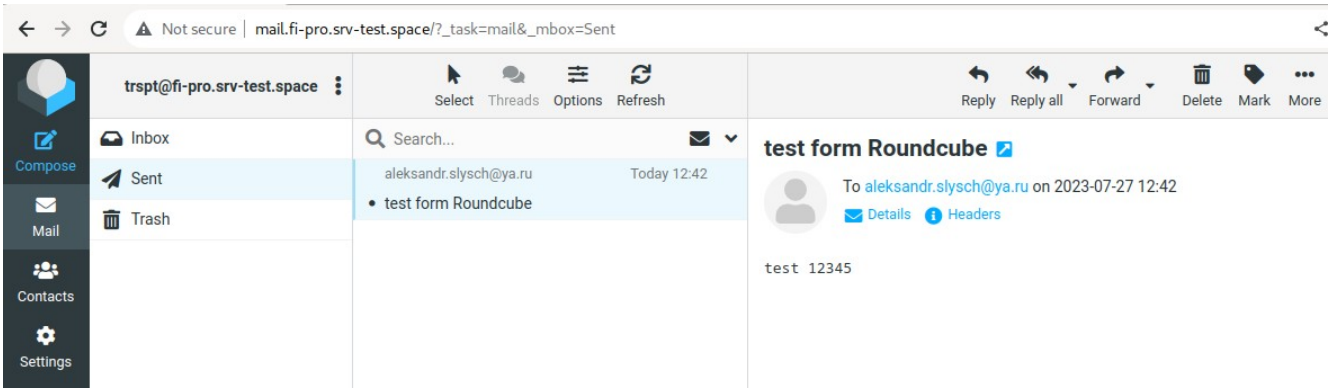
Используемые протоколы и порты:

Назначение	Протокол	Порт
отправка сообщений	smtp	25
приемка сообщений	imap	143

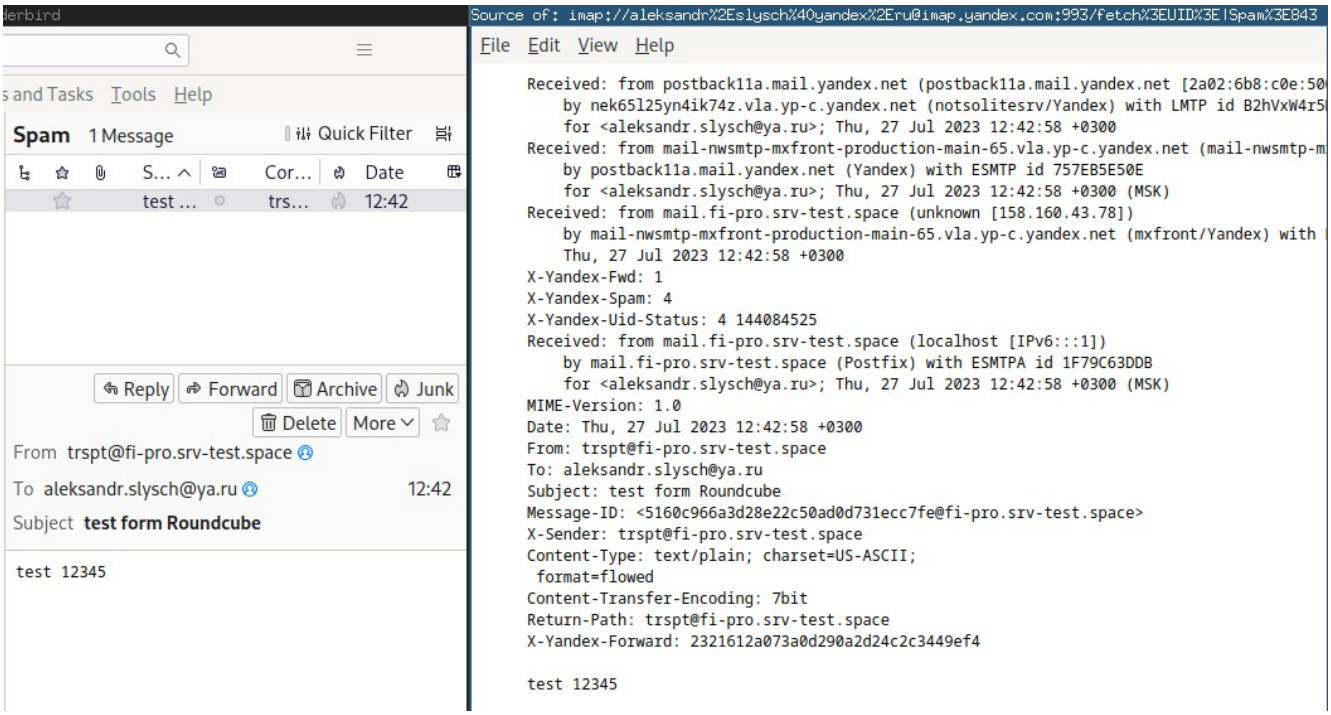
Для обеспечения веб-доступа к электронным почтовым ящикам использовался RoundCube установленный на ВМ-2 и связаный с базой данных, находящейся под управлением СУБД PostgreSQL и расположенной на ВМ-3.

Доступ к веб-интерфейсу обеспечивался ВМ-2 через настройку конфигураций для Apache и Nginx.

Отправленное сообщение на личный почтовый ящик

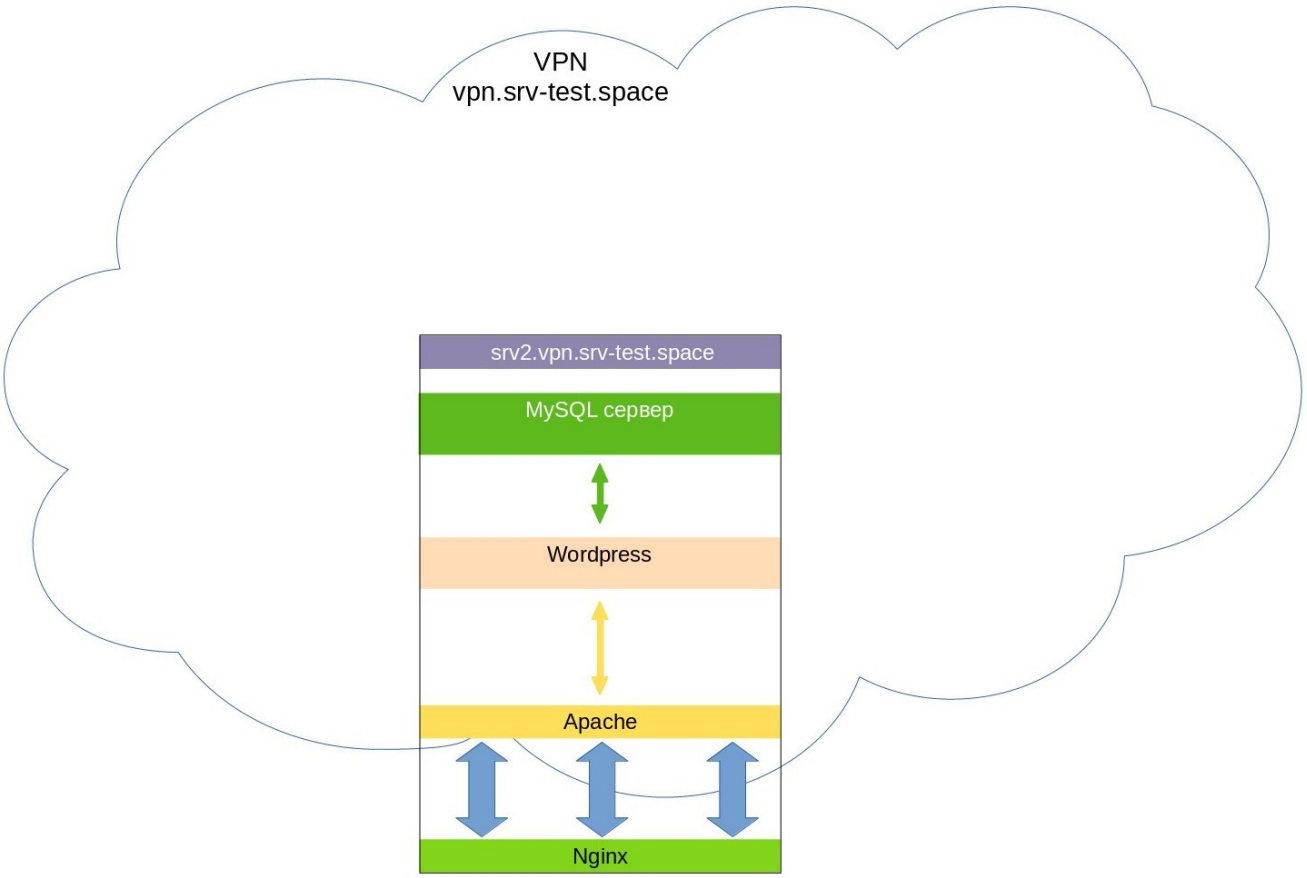


Полученное сообщение в личном почтовом ящике



9. WordPress

Схема «WordPress».



Общее описание:

Был установлен на ВМ-2.

Был связан с базой данных, находящейся под управлением СУБД MySQL и также расположенной на ВМ-2.

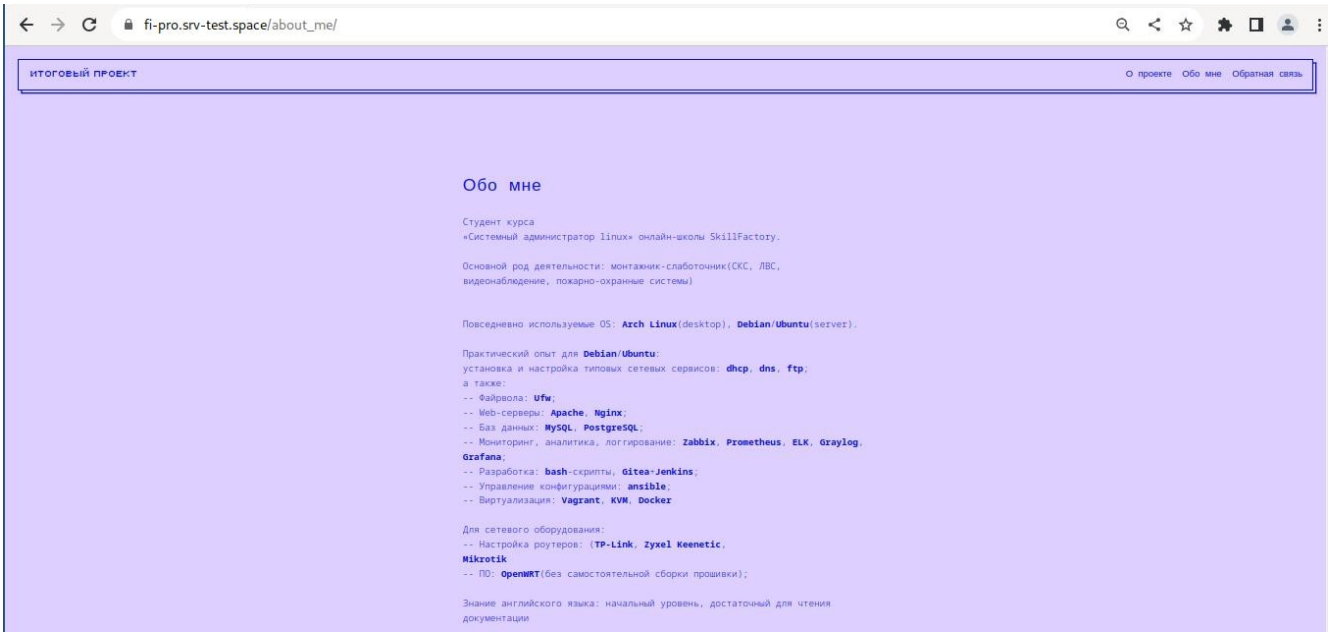
В WordPress было создано 3 страницы:

- с описанием проекта
- краткой информацией об авторе
- с формой обратной связи

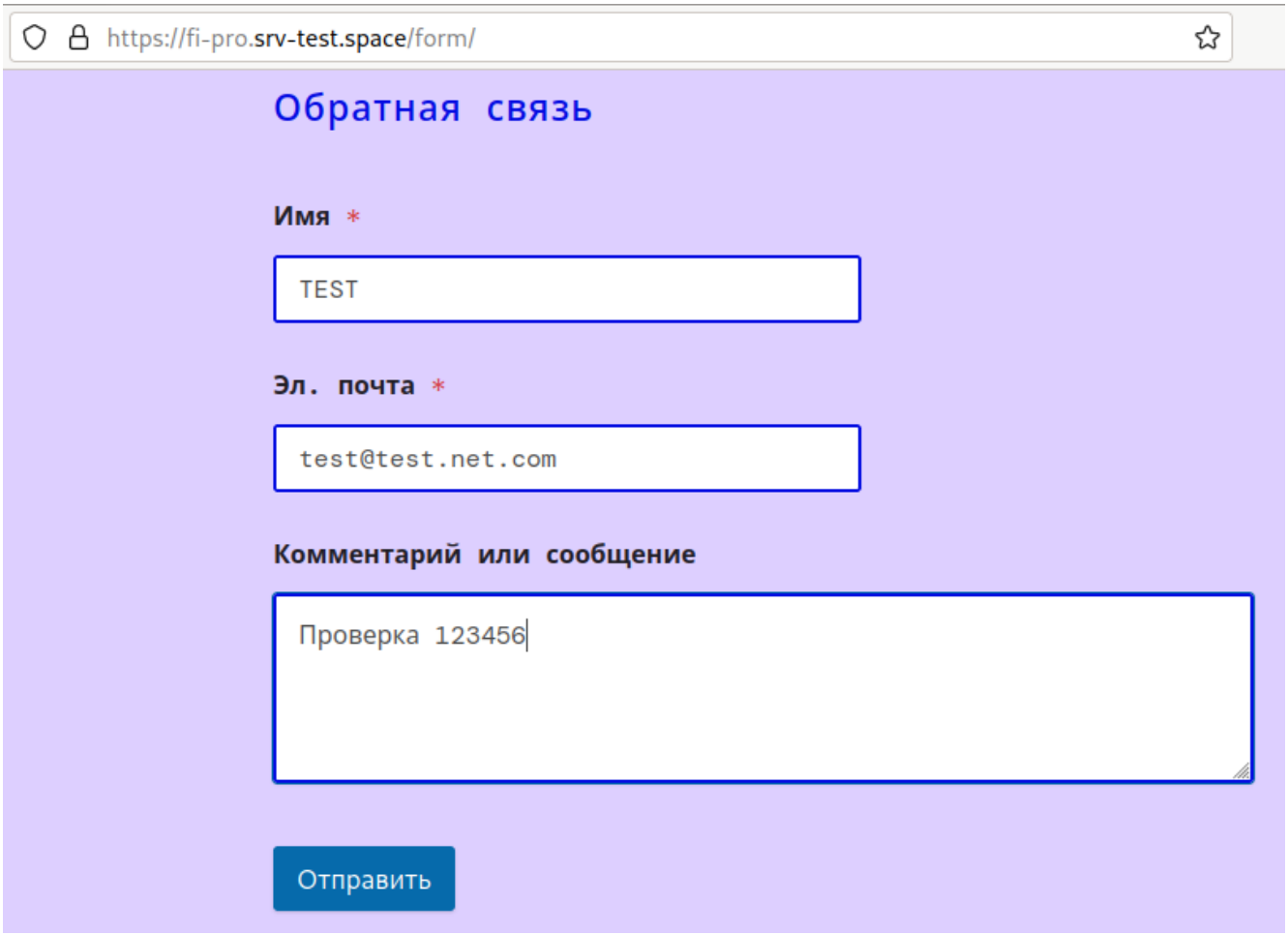
Доступ к веб-интерфейсу обеспечивался ВМ-2 через настройку конфигурации для Apache и Nginx.

Страница с описанием проекта

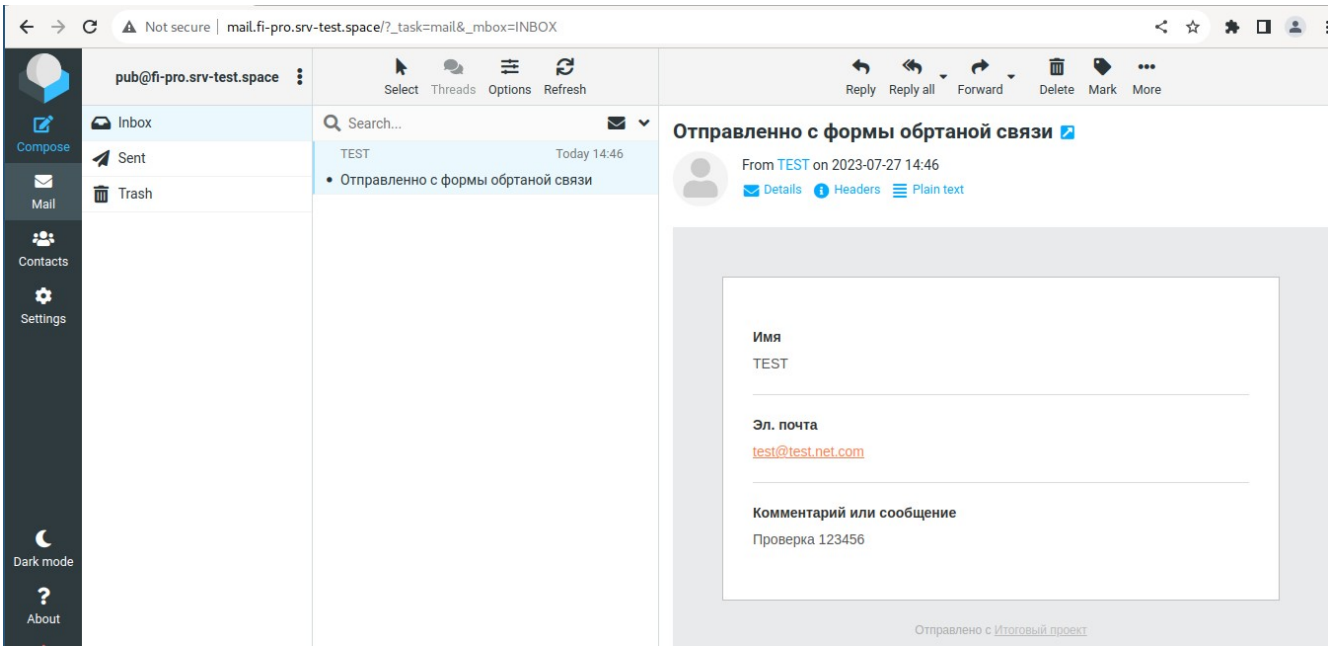
Страница с краткой информацией об авторе



Подготовленное, для отправки, сообщение со страницы с формой обратной связи

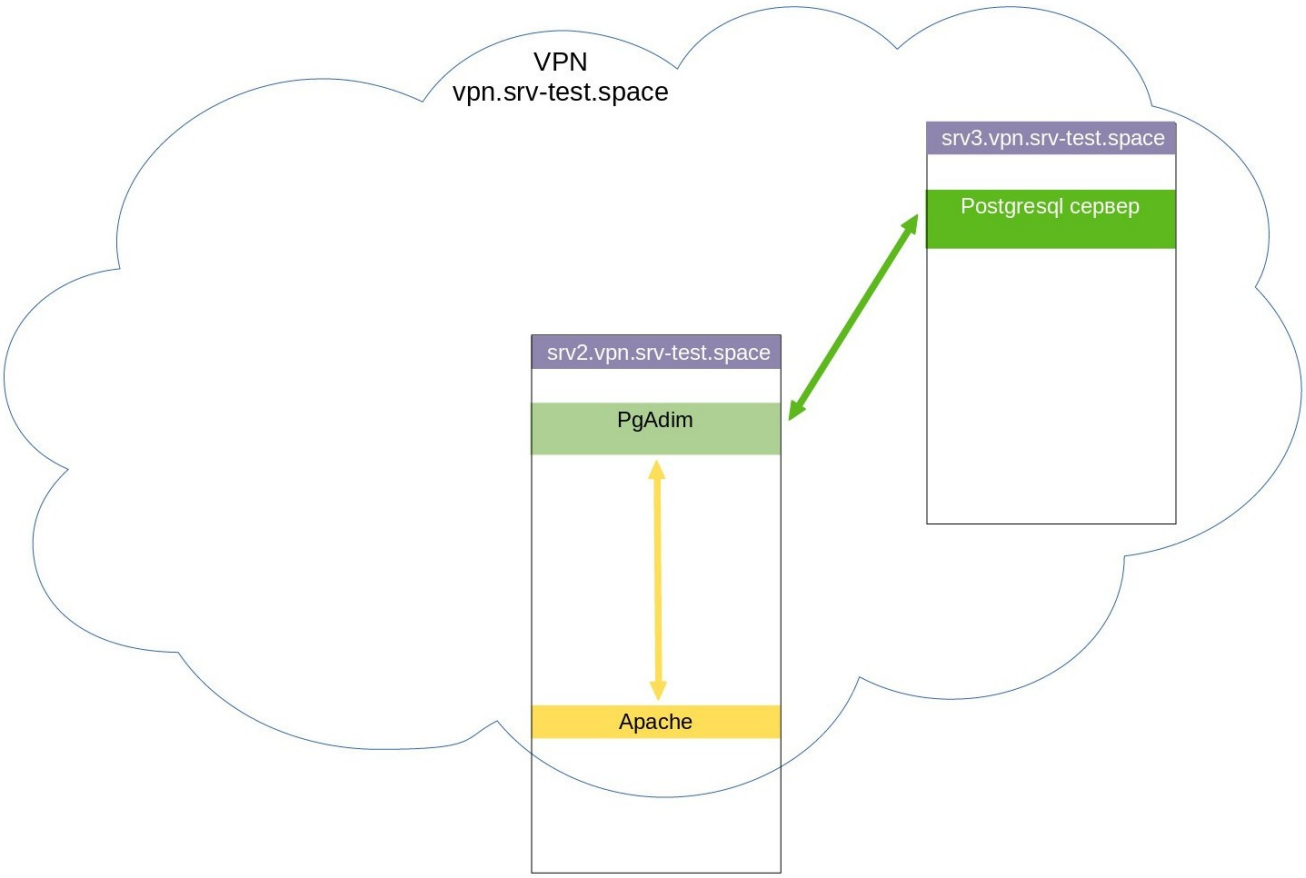


Полученное сообщение со страницы с формой обратной связи



10. PgAdmin

Схема «PgAdmin».



Общее описание:

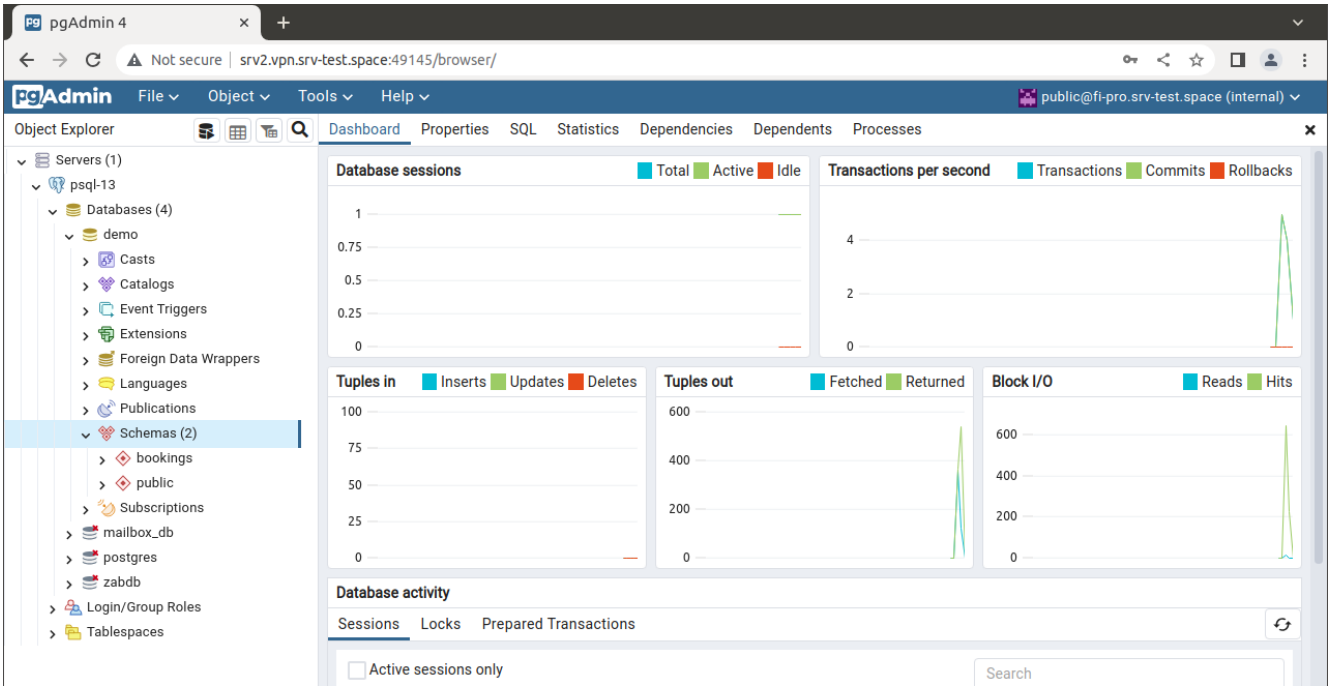
Был установлен на ВМ-2.

Связан с базами данных, находящихся под управлением СУБД PostgreSQL и расположенных на ВМ-3.

Доступ к веб-интерфейсу PgAdmin обеспечивался ВМ-2 через настройку конфигурации для Apache.

Для доступа было необходимо подключение к VPN.

Подключение к PgAdmin внутри VPN



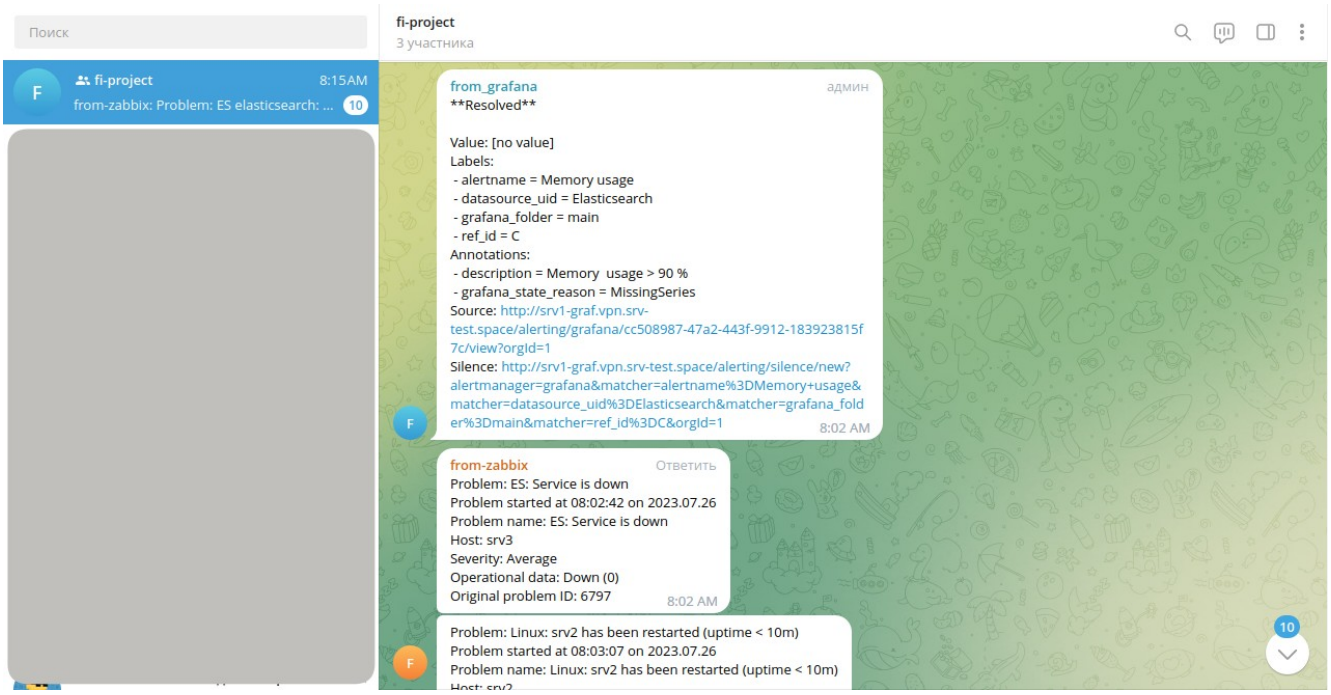
11. Прочее

11.1 Используемые версии ПО:

Операционная система	Ubuntu 20.04
Apache	Apache/2.4.41 (Ubuntu)
Nginx	1.18.0 (Ubuntu)
PHP	7.4.3-4ubuntu2.19
Zabbix (сервер и агенты)	6.4.4
Grafana	grafana-enterprise_10.0.1
ELK	8.8.2
PostgreSQL	13.11
Roundcube	1.6.2
WordPress	6.6.2
Docker	24.0.4, build 3713ee1

11.2 Страница с нейтральной картинкой

11.3 Для получения алертингов от сервера Zabbix и Grafana была создана группа в Telegram



11.4. Настройка файрвола(ufw)

На ВМ-1

```
Status: active

To                         Action      From
--                         ------      ----
48701/tcp                  ALLOW       Anywhere
Anywhere                   ALLOW       10.20.30.0/28
49101                      ALLOW       158.160.43.78
49101                      ALLOW       158.160.43.186
48701/tcp (v6)             ALLOW       Anywhere (v6)
```

На ВМ-2

```
Status: active

To                         Action      From
--                         ------      ----
48701/tcp                  ALLOW       Anywhere
Anywhere                   ALLOW       10.20.30.0/28
Nginx HTTP                 ALLOW       Anywhere
Nginx HTTPS                ALLOW       Anywhere
Dovecot Secure IMAP        ALLOW       Anywhere
Postfix Submission         ALLOW       Anywhere
Postfix                    ALLOW       Anywhere
Dovecot IMAP               ALLOW       Anywhere
48701/tcp (v6)             ALLOW       Anywhere (v6)
Nginx HTTP (v6)            ALLOW       Anywhere (v6)
Nginx HTTPS (v6)           ALLOW       Anywhere (v6)
Dovecot Secure IMAP (v6)   ALLOW       Anywhere (v6)
Postfix Submission (v6)    ALLOW       Anywhere (v6)
Postfix (v6)               ALLOW       Anywhere (v6)
Dovecot IMAP (v6)          ALLOW       Anywhere (v6)
```

На ВМ-3

```
Status: active

To                         Action      From
--                         ------      ----
48701/tcp                  ALLOW       Anywhere
Anywhere                   ALLOW       10.20.30.0/28
48777                      ALLOW       46.8.62.28
48701/tcp (v6)             ALLOW       Anywhere (v6)
```