Общество с ограниченной ответственностью «КРИПЛАТ»
ИНН/КПП 7743461050 / 774301001
125130, г. Москва, вн.тер.г. Муниципальный Округ Коптево,
ул. Зои и Александра Космодемьянских, д. 31, к. 1, помещ. 26/2
Тел.+7 (495) 19-88
e-mail: info@kriplat.net
https://www.kriplat.net

ИНФОРМАЦИОННАЯ СИСТЕМА ЭЛЕКТРОННОЙ ОЧЕРЕДИ ТРАНСПОРТНЫХ СРЕДСТВ (ИС ЭОТС, ИС «Электронная очередь»)

Инструкция по установке программного обеспечения

Листов 12

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Обозначение/ сокращение	Определение
БД	База данных
ВМ	Виртуальная машина
ОС	Операционная система
Система	Информационная система электронной очереди транспортных средств
СУБД	Система управления базами данных
ТЗ	Техническое задание
HTTP	Протокол передачи данных
SSL	Криптографический протокол, обеспечивающий защищенную передачу данных в компьютерной сети
SQL	Язык структурированных запросов

1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ

Система предназначена для постановки ТС и других участников движения в электронную очередь, автоматического управления ходом движения очереди, индикации факта наступления очереди участников, приема от различных типов СКУД или из программных интерфейсов (API) данных о попадании и покидании зон ожидания, для контроля нахождения участников очереди в зонах ожидания.

Выполнение функций пользователями Системы осуществляется через веб-интерфейс.

В Системе реализовано разграничение прав пользователей путем назначения ролей. Доступны следующие роли:

Пользователь – лицо, которое использует Систему, обладает стандартным набором прав.

Администратор – пользователь с расширенным набором прав. Кроме стандартной функциональности, администратор имеет возможность управлять доступом пользователей.

Предоставление доступа к Системе обеспечивается с помощью логина и пароля.

2. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И ОБОРУДОВАНИЮ

2.1. Архитектурные решения

Архитектура Системы соответствует следующей структуре:

- сервер БД;
- сервер приложений;
- интеграционный сервер;
- Web-клиент.

2.2. Требования к общему программному обеспечению

Для работы специального программного обеспечения Системы на выделенные серверные ресурсы необходимо установить общее программное обеспечение, включая:

1. Ubuntu 18.04 - операционная система;
2. PostgreSQL – реляционная система управления базами данных;
3. PostGIS - открытое программное обеспечение, реализующее поддержку географических объектов в реляционной базе данных PostgreSQL, выполненное в виде расширения к PostgreSQL;
4. Docker –инструмент виртуализации с открытым исходным кодом, работающий на уровне операционной системы, автоматизирующий развертывание приложений в контейнерах Linux;
5. Portainer – графический интерфейс для управления Docker контейнерами из браузера;
6. Nginx - веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах;
7. Grafana – система визуализации, ориентированная на данные систем ИТ-мониторинга;
8. Prometheus - приложение, используемое для мониторинга событий и оповещения. Оно записывает показатели в базу данных временных рядов, построенную с использованием модели HTTP pull, с гибкими запросами и оповещением в режиме реального времени.

Для клиентской части программного комплекса необходимо наличие персонального компьютера с установленным программным обеспечением:

- операционной системы: Microsoft Windows 7 или выше, MacOS 11 или выше, Linux десктопные версии;
- веб-браузер Chrome (версия не ниже 83.0.4103.116) или Яндекс (версия не ниже 20.6.2).

3. ПОРЯДОК УСТАНОВКИ СЕРВИСОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Перед началом установки системы необходимо выделить доменное имя с настроенным SSL-сертификатом и локальную подсеть с маской 255.255.255.0.

3.1. Установка Ubuntu Server 18.04

1. Загрузить с установочной флешки или DVD-диска Ubuntu Server 18.04 на выделенных серверах.

2. Выбрать язык отображения меню приветствия.
Выбираем, например, “English” и нажимаем кнопку “Enter”.

3. Выбрать нужную раскладку клавиатуры и нажимаем на кнопку “Done”.

4. Выбрать вариант установки Ubuntu Server.
Выбираем “Install Ubuntu” и нажимаем кнопку “Enter”.

5. Установщик пытается автоматически настроить сетевое подключение, используя DHCP.
Можно задать IP-адрес вручную или настроить подключение к сети после установки.
Нажать кнопку “Done”.

6. Установщик предлагает указать информацию о прокси-сервере.
Указать, если он используется. Нажать кнопку “Done”.

7. Выбрать, на какой диск будет установлена новая операционная система, и выделить место для установки. Под систему будет занято все выделенное место на диске.
Выбираем “Use An Entire Disk” и нажимаем на кнопку “Enter”.

8. Выбрать диск, на который необходимо установить Ubuntu Server и нажать кнопку “Enter”.

9. Можно увидеть, какие разделы будут созданы на диске.
Нажать кнопку “Done”.

10. Подтвердить внесенные изменения.
Выбрать “Continue” и нажать кнопку “Enter”.

11. Указать полное имя пользователя для учетной записи администратора, имя сервера, затем логин и пароль для новой учетной записи.
Нажать кнопку “Done”.

12. Чтобы подключаться к серверу по SSH, необходимо выбрать “Install OpenSSH server”.
Нажать кнопку “Done”.

13. Возможно выбрать дополнительные компоненты для установки.
Нажать кнопку “Done”.

14. Автоматически выполняется процесс установки Ubuntu Server 18.04.

15. После завершения установки Ubuntu Server 18.04 нажать кнопку “Reboot Now”.

16. Вынуть установочный диск Ubuntu Server 18.04 из CD/DVD привода.
Нажать кнопку “Enter”.

17. Указать имя пользователя и пароль для авторизации в Ubuntu, который был указан ранее в процессе установки системы.

3.2. Установка и настройка БД PostgreSQL

3.2.1. Установка PostgreSQL

Предполагается установка на ВМ.

1. Подключаем официальный репозиторий postgresql
   yum -y install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

2. Устанавливаем postgresql12 с расширением postgis25
   yum -y install postgresql12-server postgresql12-contrib postgis25_12

3. Следующий шаг - создание базы для хранения данных:
   postgresql-12-setup initdb

4. Далее - разрешить и осуществить запуск Postgres:
   systemctl enable postgresql-12
   systemctl start postgresql-12

3.2.2. Настройка PostgreSQL

Необходимо выстроить путь расположения баз.

1. Выполняем команду:
   systemctl status postgresql-12

2. Ищем в выводе команды строку, похожую на:
   /usr/bin/postgres -D /var/lib/pgsql/12/data/ -p 5432

Путь размещения баз и конфигурационных файлов - /var/lib/pgsql/data

3. Далее открыть конфигурационный файл postgresql.conf:
   vi /var/lib/pgsql/12/data/postgresql.conf

4. Найти строку:
   #listen_addresses = 'localhost'

5. Заменить ее на:
   listen_addresses = '*'

6. Далее открыть файл:
   vi /var/lib/pgsql/12/data/pg_hba.conf

7. Добавить (например IP):
   host all all 192.168.1.1/8 password
   host all all 192.168.1.1/24 password

Последняя строка позволяет подключаться к серверу с любого компьютера в сети 192.168.1.1/24 и 192.168.1.1/8; первая и вторая all дает возможность подключаться всем учетным записям и ко всем базам; password — для подключения необходима аутентификация.

8. Для применения настроек необходимо осуществить перезапуск сервера баз данных:
   systemctl restart postgresql-12

9. Открыть сетевой порт в брандмауэре:
   firewall-cmd --permanent --add-port=5432/tcp
   firewall-cmd --reload

3.2.3. Создание базы данных и пользователя

1. Для подключения к СУБД необходимо авторизоваться в БД под пользователем postgres:
   su - postgres

2. Подключиться к PostgreSQL:
   psql

3. В целях безопасности необходимо задать пароль для учетной записи postgres:
   =# ALTER USER postgres PASSWORD 'password';
   * где password — создаваемый пароль.

4. Создать отдельного пользователя:

   CREATE USER tr_user ENCRYPTED PASSWORD 'db_password'

3.2.3.1.Создание базы данных PostgreSQL

1. Создание БД, например, electronicqueueo существляется командой:
   =# CREATE DATABASE Transitapp OWNER Transitapp user;

2. Подключение к созданной БД:
   =# \connect Transitapp

3. Создание необходимых расширений в БД:
   create extension btree_gist;

```
create extension dblink;
create extension pg_buffercache;
create extension pg_trgm;
create extension postgis;
create extension postgis_topology;
```

4. Сформировать в БД необходимые таблицы.

3.3. Установка Nginx

Предполагается установка на ВМ.

1. Устанавливаем Nginx командой:
   yum -y install nginx

2. Запуск сервиса командой:
   systemctl start nginx

3. Проверка состояния выполняется командой:
   systemctl status nginx

4. Включение автозапуска при старте ОС:
   systemctl enable nginx

5. Открыть сетевые порты в брандмауэре:
   firewall-cmd --permanent --add-port=80/tcp
   firewall-cmd --permanent --add-port=443/tcp
   firewall-cmd –reload

6. Проверка установки nginx:

в браузере набрать адрес: http://ip_адрес_вм_с_установленным_nginx
Должна отобразиться страница с приглашением к работе

Конфигурационные файлы nginx находятся в директории:
/etc/nginx/

7. После изменений в конфигурации nginx проверка конфигурации осуществляется командой:
   nginx –t

В случае отсутствия ошибок в конфигурационных файлах, вывод команды должен быть:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

8. Перезагрузка nginx для применения конфигурации:
   nginx -s reload

9. Далее необходимо выбрать имя домена, на котором будут разворачиваться сервисы и от которого имеется в наличии SSL-сертификат.

Настройка location's:

https://api.имя_домена.ru/electronicqueue проксирует в контейнер electronicqueue.service

https://api.имя_домена.ru/electronicqueue-api проксирует в контейнер electronicqueue-api.service

https://api.имя_домена.ru/electronicqueue/loader смотрит проксирует в контейнер electronicqueue-loader

https://api.имя_домена.ru/electronicqueue/report/common проксирует в контейнер electronicqueue-report-service

Пример конфига nginx:

```
server {
      listen 80;
      server_name доменное_имя_сервера.ru;

  location = /robots.txt {
        add_header  Content-Type  text/plain;
        return 200 "User-agent: *\nDisallow: /\n";
    }

      location = / {
      return 301 https://$server_name$request_uri;
    }
}

server {
      listen 80;
      server_name api.доменное_имя_сервера.ru;

  location = /robots.txt {
        add_header  Content-Type  text/plain;
        return 200 "User-agent: *\nDisallow: /\n";
    }

      location = / {
      return 301 https://$server_name$request_uri;
    }
}

server {
      listen 80;
      server_name electronicqueue.доменное_имя_сервера.ru;

  location = /robots.txt {
        add_header  Content-Type  text/plain;
        return 200 "User-agent: *\nDisallow: /\n";
    }
```

```
        location = / {
        return 301 https://$server_name$request_uri;
    }
}

server {
    listen 443 ssl;
    server_name доменное_имя_сервера.ru;
    ssl_certificate ssl/fullchain.доменное_имя_сервера.ru.pem;
    ssl_certificate_key ssl/privkey.доменное_имя_сервера.ru.key;

    location = /robots.txt {
      add_header  Content-Type  text/plain;
      return 200 "User-agent: *\nDisallow: /\n";
    }
}

server {
    listen 443 ssl;
    server_name electronicqueue.доменное_имя_сервера.ru;
    ssl_certificate ssl/fullchain.доменное_имя_сервера.ru.pem;
    ssl_certificate_key ssl/privkey.доменное_имя_сервера.ru.key;

    location = /robots.txt {
      add_header  Content-Type  text/plain;
      return 200 "User-agent: *\nDisallow: /\n";
    }

    location / {
        proxy_pass http:// 192.168.1.1:8001/;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection  $http_connection;
    }
}

server {
    listen 443 ssl;
    server_name api.имя_сервера.ru;
    ssl_certificate ssl/fullchain.доменное_имя_сервера.ru.pem;
    ssl_certificate_key ssl/privkey.доменное_имя_сервера.ru.key;

    client_max_body_size 1G;

    location = /robots.txt {
      add_header  Content-Type  text/plain;
      return 200 "User-agent: *\nDisallow: /\n";
```

```
        }
        }

        location /electronicqueue/loader/ {
            proxy_pass http:// 192.168.1.1:8004/;
            proxy_set_header Host $host;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection  $http_connection;
        }

        location /electronicqueue /report/common/ {
            proxy_pass http:// 192.168.1.1:8005/;
            proxy_set_header Host $host;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection  $http_connection;
        }
    }
```

3.4. Установка Docker

Предполагается установка на все виртуальные машины.

1. Выполнить:
   curl -fsSL https://get.docker.com/ | sh

2. Запуск сервиса командой:
   systemctl start docker

3. Проверка состояния:
   systemctl status docker

4. Включение автозапуска при старте ОС:
   systemctl enable docker

5. Проверка работоспособности docker:
   docker run hello-world

6. Вывод команды при успешном выполнении должен начинаться со строк:
   Hello from Docker!
   This message shows that your installation appears to be working correctly.

Далее осуществить импорт docker-образов с программным обеспечением Системы на все виртуальные машины.

3.5. Установка Portainer

Предполагается установка на ВМ.

Для начала установки Portainer необходимо создать volume, где будут размещены данные:

root@dedicated:~# docker volume create portainer_data

Далее следует запустить контейнер следующей командой:
root@dedicated:~# docker run --name portainer --restart always -d -p 9001:9000 -v portainer_data:/data -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer

По окончании установки, web-интерфейс Portainer будет доступен по ссылке: https://ваш_ip_адрес:9001.

Отобразиться окно регистрации с предложением ввести пароль администратора. Рекомендуется ввести сложный пароль и подтвердить его.

Далее появится окно с выбором окружения, Local или Remote. Если Docker установлен локально, необходимо выбрать Local и подключиться нажатием Connect. После чего отобразиться рабочее окружение Portainer.

Открываем на хосте порты, необходимые для Portainer:

firewall-cmd --permanent –add-port=9001/tcp
firewall-cmd –reload

4. РАЗВЕРТЫВАНИЕ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

После установки и запуска на выделенных виртуальных машинах компонент общего программного обеспечения (nginx, docker, postgres и др.), необходимо установить на эти же ресурсы специальное программное обеспечение.