**Сопроводительная документация по развертыванию**

**Программы для ЭВМ**

**«Система автоматизации товарно-транспортной логистики 2.0»**

На 10 листах

2022

**СОДЕРЖАНИЕ**

[1. Общие положения 3](#_Toc122434599)

[1.1. Полное наименование Программы для ЭВМ, обозначение 3](#_Toc122434600)

[1.2. Разработчик Программы для ЭВМ 3](#_Toc122434601)

[1.3. Назначение документа 3](#_Toc122434602)

[2. Описание требований 3](#_Toc122434603)

[2.1. Минимальные аппаратные требования 3](#_Toc122434604)

[2.2. Требования к рабочим станциям 4](#_Toc122434605)

[3. Развертывание 4](#_Toc122434606)

[3.1. Подготовка сервера 4](#_Toc122434607)

[3.2. Развертывание Postgres 14.6 4](#_Toc122434608)

[3.3. Развертывание Minio на Debian 11 5](#_Toc122434609)

[3.4. Развертывание RabbitMQ 3.10.7 7](#_Toc122434610)

[3.5. Развертывание проекта САТТЛ 2.0 10](#_Toc122434611)

# **Общие положения**

## **Полное наименование Программы для ЭВМ, обозначение**

Полное наименование Программы для ЭВМ: Система автоматизации товарно-транспортной логистики 2.0.

Краткое наименование (обозначение) Программы для ЭВМ: «САТТЛ 2.0».

## **Разработчик Программы для ЭВМ**

Полное наименование: Общество с ограниченной ответственностью «Философия.ИТ».

Сокращенное наименование: ООО «Философия.ИТ».

## **Назначение документа**

Настоящий документ входит в комплект эксплуатационной документации по Системе САТТЛ 2.0 и предназначен для описания развертывания Системы.

# **Описание требований**

## **Минимальные аппаратные требования**

Минимальные аппаратные требования указаны в таблице Таблица 1.

Таблица 1 – Минимальные аппаратные требования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Роль сервера** | **Количество ядер процессора** | **Объем оперативной памяти** | **Объем физической памяти** | **Операционная система** |
| 1 | Сервер САТТЛ 2.0 | 10 | 32 ГБ | 500 ГБ | Debian 11 |
| 2 | Postgres 14.6 | 8 | 32 ГБ | 1 ТБ | Debian 11 |
| 3 | Minio хранение файлов | 8 | 32 ГБ | 1 ТБ | Debian 11 |

## **Требования к рабочим станциям**

Автоматизированное рабочее место (АРМ) пользователей системы может быть развернуто на любых станциях, операционная система которых поддерживает работу браузера.

# **Развертывание**

## **Подготовка сервера**

* Установить корректный часовой пояс (например, MSK): sudo timedatectl set-timezone Europe/Moscow
* Обновить список репозиториев и текущие версии установленного ПО: sudo apt-get update && apt-get upgrade
* В настройках межсетевого экрана разрешить входящие подключения к порту TCP/5432.

## **Развертывание Postgres 14.6**

* Необходимо добавить репозиторий, предоставляющий пакет Postgress 14. sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
* Импортировать ключ GPG для добавленного репозитория.

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add –

* Затем обновить индекс пакетов APT.

sudo apt -y update

* Установить Postgres 14.6

sudo apt install postgresql-14

* Настроить прослушиваемые интерфейсы. Для этого в файле

/etc/postgresql/14/main/postgresql.conf необходимо внести изменения в параметре

«listen\_addresses»:

listen\_addresses = "\*"

* Настроить доступ к базам данных. Для этого в файле

/etc/postgresql/13/main/pg\_hba.conf необходимо добавить строку:

host all all 0.0.0.0/0 md5

Данная директива разрешает доступ любым пользователям к любым базам данных (к которым у пользователя есть доступ) с любых хостов с аутентификацией по паролю. При необходимости можно ужесточить правила доступа, для этого в файле pg\_hba.conf есть примеры.

* Создать необходимых пользователей и базы данных:

sudo su – postgres

createuser -l -P newuser

createdb -E UTF8 -O glogdev sattledb

exit

* Перезапустить сервис PostgreSQL:

sudo systemctl restart postgresql

Развертывание сервера PostgreSQL завершено.

## **Развертывание Minio на Debian 11**

* Загрузить и установить бинарный файл сервера Minio с официального веб-сайта:

wget https://dl.min.io/server/minio/release/linux-amd64/minio

* Сделать файл исполняемым

chmod +x minio

* Поместить файл в каталог **/usr/local/bin**, где его ищет стартовая программа Minio systemd:

sudo mv minio /usr/local/bin

* Создать нового пользователя MinIO и вызвать группу **minio**

sudo useradd -r minio -s /sbin/nologin

* После этого обновить владельца бинарного файла Minio до minio:

sudo chown minio:minio /usr/local/bin/minio

* Создать папку где будут находится объекты хранилища

sudo mkdir /usr/local/share/minio

* Создать папку с конфигурационным файлом

sudo mkdir /etc/minio

sudo nano /etc/default/minio

* В открывшемся редакторе следует описать переменные. Пример:

MINIO\_ACCESS\_KEY="ключ доступа"

MINIO\_VOLUMES="/usr/local/share/minio/" # директория куда будут сохранятся объекты

MINIO\_OPTS="-C /etc/minio --address {ip-адресс машины}:9000"

MINIO\_SECRET\_KEY="закрытый ключ"

* Далее необходимо пользователю minio дать необходимые права на файлы/директории

sudo chown minio-user:minio-user /usr/local/bin/minio

sudo chown minio-user:minio-user /usr/local/share/minio

sudo chown minio-user:minio-user /etc/minio

* Чтобы управлять сервером Minio через systemd, нужно скачать файл дескриптора службы Minio

curl -O https://raw.githubusercontent.com/minio/minio-service/master/linux-systemd/minio.service

* Если имя пользователя или расположение файла конфигурации были заданы отличающимися от указанных в инструкции, следует изменить значение параметров (User, Group и/или EnvironmentFile) в файле службы, иначе пропускаем этот пункт:

nano minio.service

* Переместить файл в каталог конфигураций

sudo mv minio.service /etc/systemd/system

* Далее нужно перезагрузить все юниты systemd

sudo systemctl daemon-reload

* Для активации Minio при авто запуске ввести

sudo systemctl enable minio

* Запуск сервера Minio производится командой

sudo systemctl start minio

## **Развертывание RabbitMQ 3.10.7**

* Добавить в систему все ключи, которые используются для подписи репозиториев с Erlang и RabbitMQ

curl -1sLf https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-erlang/gpg.E495BB49CC4BBE5B.key | sudo gpg --dearmor | sudo tee /usr/share/keyrings/io.cloudsmith.rabbitmq.E495BB49CC4BBE5B.gpg > /dev/null

curl -1sLf https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-server/gpg.9F4587F226208342.key | sudo gpg --dearmor | sudo tee /usr/share/keyrings/io.cloudsmith.rabbitmq.9F4587F226208342.gpg > /dev/null

curl -1sLf "https://keys.openpgp.org/vks/v1/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null

* Для работы с репозиториями нужно установить пакет apt-transport-https

sudo apt install apt-transport-https

* Далее нужно создать файл /etc/apt/sources.list.d/rabbitmq.list и записать в него репозитории Erlang и RabbitMQ

deb [signed-by=/usr/share/keyrings/io.cloudsmith.rabbitmq.E495BB49CC4BBE5B.gpg] https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-erlang/deb/ubuntu bionic main

deb-src [signed-by=/usr/share/keyrings/io.cloudsmith.rabbitmq.E495BB49CC4BBE5B.gpg] https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-erlang/deb/ubuntu bionic main

deb [signed-by=/usr/share/keyrings/io.cloudsmith.rabbitmq.9F4587F226208342.gpg] https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-server/deb/ubuntu bionic main

deb-src [signed-by=/usr/share/keyrings/io.cloudsmith.rabbitmq.9F4587F226208342.gpg] https://dl.cloudsmith.io/public/rabbitmq/rabbitmq-server/deb/ubuntu bionic main

* Установка Erlang

sudo apt install -y erlang-base \

erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \

erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key \

erlang-runtime-tools erlang-snmp erlang-ssl \

erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl

* Для установки RabbitMQ необходимо выполнить команду

sudo apt-get install rabbitmq-server -y --fix-missing

* Чтобы проверить запустился ли и работает сервер используется команда

sudo systemctl status rabbitmq-server

* Если необходимо можно добавить RabbitMQ в автозагрузку командой

sudo systemctl enable rabbitmq-server

* Если нужен доступ к веб-интерфейсу, то его необходимо включить

sudo rabbitmq-plugins enable rabbitmq\_management

* Доступ к веб-сервису можно получить http://адрес\_сервера:15672 для localhost это будет http://localhost:15672. По умолчанию логин и пароль для доступа - guest:guest. Но эти данные доступа работают только для localhost.
* Чтобы подключатся удаленно необходимо создать пользователя. Вместо <user\_name> поставить имя пользователя, которого необходимо создать, а <password> заменить на пароль.

sudo rabbitmqctl add\_user <user\_name> <password>

* Для того чтобы сделать только что созданного пользователя администратором выполнить такую команду:

sudo rabbitmqctl set\_user\_tags <user\_name> administrator

* Так же нужно дать данному пользователю доступ в RabbitMQ командой

sudo rabbitmqctl set\_permissions -p / admin ".\*" ".\*" ".\*"

* Через веб-интерфейс создать все необходимые очереди

## **Развертывание проекта САТТЛ 2.0**

* Установка JDK Liberica 11

wget https://download.bell-sw.com/java/11.amd64/bellsoft-jdk11-linux-amd64.deb

sudo apt install ./bellsoft-jdk11-linux-amd64.deb

* Сконфигурировать файл application.properties исходя из ранее развернутых сервисов (Postgress 14, Minio)
* В папке с исходным кодом проекта выполнить команду

gradle build

* После выполнения команды в директории проекта по пути /build/libs будет находиться jar файл его нужно запустить командой

java –jar {jar\_file.jar}

Развертывание завершено.