

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДИСПЕТЧЕРСКОГО  
УПРАВЛЕНИЯ «ПУЛЬ» (АСДУ «ПУЛЬС»)**

**РУКОВОДСТВО АДМИНИСТРАТОРА**

**Тюмень, 2022**

### Аннотация

«Автоматизированная система диспетчерского управления «ПУЛЬС» (далее по тексту – АСДУ «ПУЛЬС») является программно-аппаратной системой и разработано с целью описания процесса установки АСДУ «ПУЛЬС» на серверах и рабочих станциях.

В документе приведены основные сведения, необходимые для выполнения установки АСДУ «ПУЛЬС», назначение программного обеспечения, конфигурация программно-технического комплекса.

## СОДЕРЖАНИЕ

<b>1. Введение.....</b>	<b>4</b>
<b>2. Область применения.....</b>	<b>5</b>
<b>3. Обозначения и сокращения .....</b>	<b>6</b>
<b>4. Порядок установки .....</b>	<b>7</b>
4.1. Установка и настройка postgresql .....	7
4.2. Установка docker и docker-compose .....	8
4.3. Настройка компонентов АСДУ «ПУЛЬС» .....	9
4.4. Настройка сервера резервных копий базы данных .....	11
4.5. Добавление svg-схемы для новой станции .....	13
4.6. Требования к аппаратному обеспечению.....	13

## 1. ВВЕДЕНИЕ

Полное наименование системы: Автоматизированная система диспетчерского управления «ПУЛЬС»

АСДУ «ПУЛЬС» является программно–аппаратной системой. Основными функциями АСДУ «ПУЛЬС» является сбор, передача, контроль, отображение и накопление информации с устройств автоматики и телемеханики движения поездов.

АСДУ «ПУЛЬС» предназначена для диспетчерского управления устройствами автоматики и телемеханики движения поездов.

АСДУ «ПУЛЬС» позволяет специалистам выполнять мониторинга процесса движения поездов в режиме реального времени, а также своевременно реагировать на изменение состояния устройств и оборудования.

## **2. ОБЛАСТЬ ПРИМЕНЕНИЯ**

АСДУ «ПУЛЬС» предназначено для использования в структурных подразделениях метрополитена.

Пользователи АСДУ «ПУЛЬС» должны владеть начальными навыками использования персонального компьютера и изучить настоящее Руководство.

АСДУ «ПУЛЬС» разворачивается на серверах и рабочих станциях структурных подразделений метрополитена.

### 3. ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

<b>TCP/IP</b>	– набор коммуникационных протоколов, используемых в компьютерных сетях. Базовые протоколы набора – TCP (протокол управления передачей) и IP (интернет-протокол)
<b>FTP</b>	– Стандартный протокол передачи файлов
<b>NGINX</b>	– Веб-сервер
<b>XML</b>	– расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных
<b>АСДУ «ПУЛЬС»</b>	– Автоматизированная система диспетчерского управления «ПУЛЬС».

## 4. ПОРЯДОК УСТАНОВКИ

В данной инструкции расписан процесс установки и настройки системы на базе операционной системы семейства debian. Установка и настройка на других linux системах происходит аналогично.

### 4.1. Установка и настройка postgresql

Для установки и настройки СУБД PostgreSQL необходимо выполнить следующие действия:

#### 1. Установить СУБД.

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

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

```
sudo apt-get update
```

```
sudo apt-get -y install postgresql
```

#### 2. Создать базу данных и пользователя для доступа к ней.

```
sudo -u postgres psql
```

//Дальше после каждой строки нажимать Enter

```
postgres=# create database mydb;
```

```
postgres=# create user myuser with encrypted password 'mypass';
```

```
postgres=# grant all privileges on database mydb to myuser;
```

```
postgres=# alter database mydb set timezone to 'Etc/UTC';
```

```
postgres=# select pg_reload_conf();
```

```
postgres=# \q
```

Перед выполнением команд указать следующие параметры:

- mydb – Имя базы данных.
- myuser – Имя пользователя базы данных.
- 'mypass' – Пароль для пользователя базы данных.

### 3. Настроить доступа к базе данных.

1. В конфигурационный файл `pg_hba.conf` добавить в конце файла новую строку:

```
host mydb myuser 0.0.0.0/0 md5
```

2. Отредактировать строку конфигурационного файла `postgresql.conf` следующим образом:

```
listen_adresses = '*'
```

### 4. Настроить параметры сервера баз данных.

Отредактировать конфигурационный файл `postgresql.conf` следующим образом:

```
max_connections = 36
shared_buffers = 1GB
work_mem = 512MB
maintenance_work_mem = 1GB
autovacuum_work_mem = 8MB
effective_io_concurrency = 200
wal_compression = on
wal_log_hints = on
wal_buffers = 64MB
checkpoint_timeout = 10min
checkpoint_completion_target = 0.9
max_wal_size = 200GB
effective_cache_size = 10GB
autovacuum_max_workers = 5
autovacuum_vacuum_cost_limit = 3000
```

После сохранения конфига перезапустить сервис `service postgresql restart`

### 5. Открыть внешний доступ к порту 5432

## 4.2. Установка `docker` и `docker-compose`

Установка данного ПО заключается в развертывании на сервере следующих компонентов:

- Контейнеризатор приложений `Docker`



- Инструмент для запуска и управления мультиконтейнерными приложениями docker-compose

Порядок установки:

```
sudo apt-get install docker.io curl
```

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

### 4.3. Настройка компонентов АСДУ «ПУЛЬС»

1. Распаковать архив с кодом в отдельную папку. Папка должна быть заранее создана. Для этого необходимо выполнить следующие команды:

```
mkdir <folder_name>
```

```
tar -C <folder_path> -xvf <archive_name>
```

2. В файле metro\_backend/quark\_config/docker/backend/config.ini указать параметры подключения к БД в секции [DATABASE] и сохранить файл.  
name = значение mydb из пункта 5.1.  
user = значение myuser из пункта 5.1.  
password = значение mypass из пункта 5.1.
3. Настройка файла docker-compose.yml
  1. Для сервиса frontend аргумент QUARK\_API\_HOST должен указывать на ip-адрес, по которому веб-приложение будет доступно для подключения пользователей.
  2. При необходимости подключения к контроллеру, добавить для сервиса producer переменные окружения.  
environment:
    - IP=IP адрес контроллера
    - PORT=Сетевой порт для подключения контроллера
4. Запустить сборку через docker-composer из корня папки созданный в пункте 2.

```
sudo docker-compose -f docker-compose.yml up --build -d
```

5. После первой сборки нужно создать компанию, администратора и запустить миграцию данных.

- 5.1. Запустить команду для подключения к контейнеру backend.

```
sudo docker exec -it metro_backend_1 bash
```

- 5.2. Запустить команду для создания компании.

```
quark-admin create_data_area
```

- 5.3. Ввести значение 'осн' (без кавычек) в поле DataArea: и нажать Enter.

- 5.4. Нажать у и Enter.

- 5.5. Запустить команду для создания администратора.

```
quark-admin create_superuser
```

- 5.6. Ввести значение 'admin' (без кавычек) в поле Username: и нажать Enter.

- 5.7. Ввести значение 'admin@admin.com' в поле Email address: и нажать Enter.

- 5.8. В данном шаге ничего вводить не нужно и нажать Enter.

- 5.9. В данном шаге нужно ввести пароль для администратора и нажать Enter.

- 5.10. В данном шаге нужно ввести пароль для администратора повторно и нажать Enter.

- 5.11. Запустить команду для загрузки начальных настроек.

```
quark-admin migrate_data
```

- 5.12. Запустить команду для выхода из докера контейнера.

```
exit
```

6. Если IP адрес не менялся, то запуск в браузере будет доступен по адресу <http://127.0.0.1>

Для того, чтобы посмотреть список запущенных контейнеров необходимо

выполнить команду:  
`docker ps - a`

Для корректной работы АСДУ «ПУЛЬС» должны быть запущены следующие контейнеры:

```
metro_backend_1
metro_consumer_1
metro_envoy_1
metro_frontend_1
metro_kafka_1
metro_producer_1
metro_zookeeper_1
```

#### **4.4. Настройка сервера резервных копий базы данных**

Для установки и настройки СУБД PostgreSQL необходимо выполнить следующие действия:

1. Создать каталог для хранения резервных копий.

```
sudo mkdir /pg_backup
```

2. Создать файл скрипта.

```
sudo touch /usr/local/bin/pgsql_backup.sh
```

Добавить в файл следующее содержимое:

```
#!/bin/bash
```

```
#
```

```
# Backup a Postgresql database into a daily file.
```

```
#
```

```
BACKUP_DIR=/pg_backup
```

```
DAYS_TO_KEEP=14
```

```
FILE_SUFFIX=_pg_backup.sql
```

```
HOST=localhost
```

```
PORT=5432
```

```
DATABASE=mydb
```

```
USER=myuser
PASSWORD=mypass
```

```
FILE=`date +"%Y%m%d%H%M"`${FILE_SUFFIX}
```

```
OUTPUT_FILE=${BACKUP_DIR}/${FILE}
```

```
# do the database backup (dump)
```

```
# use this command for a database server on localhost. add other options if need
be.
```

```
pg_dump --
dbname=postgresql://${USER}:${PASSWORD}@${HOST}:${PORT}/${DATABASE} -F p -f ${OUTPUT_FILE}
```

```
# gzip the mysql database dump file
```

```
gzip $OUTPUT_FILE
```

```
# show the user the result
```

```
echo "${OUTPUT_FILE}.gz was created:"
```

```
ls -l ${OUTPUT_FILE}.gz
```

```
# prune old backups
```

```
find $BACKUP_DIR -maxdepth 1 -mtime +$DAYS_TO_KEEP -name
"*${FILE_SUFFIX}.gz" -exec rm -rf '{}' ';' 
```

Для параметров:

```
DATABASE=mydb
```

```
USER=myuser
```

```
PASSWORD=mypass
```

Указать значения из пункта 5.1.

3. Добавить права на запуск скрипта.

```
sudo chmod 777 /usr/local/bin/pgsql_backup.sh
```

4. Настроить периода запуска скрипта.

В файл /etc/crontab добавить новую строку:

```
0 0 * * * root /usr/local/bin/pgsql_backup.sh
```

#### 4.5. Добавление svg-схемы для новой станции

Для добавления svg-схемы для новой станции необходимо выполнить следующие действия:

1. Добавить файл схемы в каталог с исходным кодом  
metro-frontend/src/app/modules/elements/components/
2. Выполнить команду

```
npm run build
```

#### 4.6. Требования к аппаратному обеспечению

Для штатной работы всех модулей АСДУ «ПУЛЬС» необходима минимальная конфигурация серверов, которая представлена в таблице 1.

Таблица 1 – Минимальная конфигурация сервера

<b>АРМ</b>		
<b>Характеристика</b>	<b>Единица измерения</b>	<b>Значение</b>
ЦПУ	Модель	Intel Core i5 или аналог
ОЗУ DDR4	Гбайт	8 или более
Жесткий диск SSD	Гбайт	100 или более
Жесткий диск HDD	Гбайт	100 или более
Операционная система	Версия	Microsoft Windows 8 или более новая
<b>Единый сервер для всех подсистем</b>		
<b>Характеристика</b>	<b>Единица измерения</b>	<b>Значение</b>
ЦПУ	Модель	Не менее 8 доступных вычислительных ядер
ОЗУ DDR4	Гбайт	16 или более
Жесткий диск SSD	Гбайт	100 или более
Жесткий диск HDD (RAID 1 или 0+1)	Гбайт	250 или более
Операционная система	Версия	Linux
<b>Резервный сервер для всех подсистем</b>		

<b>Характеристика</b>	<b>Единица измерения</b>	<b>Значение</b>
ЦПУ	Модель	Не менее 8 доступных вычислительных ядер
ОЗУ DDR4	Гбайт	16 или более
Жесткий диск SSD	Гбайт	100 или более
Жесткий диск HDD (RAID 1 или 0+1)	Гбайт	250 или более
Операционная система	Версия	Linux