

Инструкция по установке Программы для ЭВМ «АСУ ГТК»

Оглавление

1. Требования к серверной платформе, общесистемному ПО и СУБД.....	3
1.1. Требования к серверной платформе	3
1.2. Требования к сетевым подключениям	3
1.3. Требования к операционной системе	3
1.4. Требования к СУБД.....	4
2. Развертывание программного обеспечения.....	5
2.1. Настройка специфических параметров ОС и / или СУБД.....	5
2.2. Настройка рабочей среды.....	7
2.3. Установка прикладного ПО.....	7
3. Порядок проверки работоспособности.....	12
4. Эксплуатация, техническое обслуживание, ремонт и хранение компонентов системы.....	13
5. Действия при возникновении ошибок и неполадок	14

Перечень сокращений

Термин	Определение
ОС	Операционная система
СУБД	Система управления базами данных
БД	База данных
АРМ	Автоматизированное рабочее место
ПО	Программное обеспечение
ПК	Персональный компьютер
БНО	Бортовое навигационное оборудование

1. Требования к серверной платформе, общесистемному ПО и СУБД

1.1. Требования к серверной платформе

ПО АСУ ГТК может разворачиваться на основе серверов на стандартной архитектуре Intel x86 и совместимых.

Рекомендуется использовать современные процессоры семейств Intel Xeon или AMD EPYC.

Требования к дисковой подсистеме: необходимо настроить RAID 1 с общим объемом памяти от 1 Тб.

1.2. Требования к сетевым подключениям

Для работы системы необходимо подключение сервера системы к сети Интернет с рекомендуемой пропускной способностью от 50 Мбит/с и выделенный IP-адрес.

На ПК АРМ пользователя необходимо наличие подключения к Интернет с пропускной способностью не менее 10 Мбит/с.

1.3. Требования к операционной системе

Серверная часть ПО АСУ ГТК разворачивается на базе операционной системы РЕД ОС 8 Сервер.

В случае отсутствия развернутой операционной системы на сервере, перед началом установки ПО АСУ ГТК необходимо развернуть и настроить операционную систему РЕД ОС 8 Сервер на серверной платформе.

Примечание:

Рекомендуется установка серверной операционной системы с графической оболочкой KDE Plasma.

Выполнение всех команд по установке производится в консоли.

После установки операционной системы необходимо настроить сеть, язык и обновить системные пакеты.

В настройках ОС и СУБД необходимо установить локальное время и часовой пояс.

1.4. Требования к СУБД

База данных программного обеспечения АСУ ГТК разворачивается на основе СУБД PostgreSQL - свободной объектно-реляционной системы управления базами данных (<https://www.postgresql.org/>).

В случае отсутствия развернутой операционной системы на сервере, перед началом установки ПО АСУ ГТК необходимо развернуть и настроить СУБД PostgreSQL в соответствии с руководством СУБД.

Для установки СУБД необходимо войти в систему под профилем root.

Выполнить в консоли управления команду:

```
su -
```

Установить СУБД PostgreSQL:

```
dnf install postgresql18-server
```

2. Развертывание программного обеспечения

2.1. Настройка специфических параметров ОС и / или СУБД

Произвести инициализацию базы данных postgresql:

```
postgresql-18-setup initdb
```

Запустить службу и добавить её в автозагрузку:

```
systemctl enable postgresql-18.service --now
```

Проверить статус службы:

```
systemctl status postgresql-18.service
```

Выйти из профиля root:

```
exit
```

При установке СУБД, необходимо установить дополнительные расширения.

Установка pg_partman

Установить библиотеки разработки для PostgreSQL:

```
sudo dnf install postgresql18-devel.x86_64
```

Перейти в домашний каталог:

```
cd ~
```

Клонировать репозиторий:

```
git clone https://github.com/pgpartman/pg_partman.git
```

Перейти в каталог pg_partman:

```
cd pg_partman
```

Установить путь к pg_config

```
export PATH=/usr/pgsql-18/bin:$PATH
```

Установить pg_partman:

```
sudo PATH=$PATH make NO_BGW=1 install
```

Зайти под профилем postgres:

```
su - postgres
```

Запустить postgres shell:

```
psql
```

Создать расширение:

```
CREATE EXTENSION pg_partman;
```

Проверить установку расширения:

```
SELECT * FROM pg_extension WHERE extname = 'pg_partman';
```

Выйти из postgres shell:

```
\q
```

Выйти из профиля postgres:

```
exit
```

Установка pg_cron

Установить pg_cron:

```
sudo dnf install pg_cron-18.x86_64
```

Зайти под профилем postgres:

```
su - postgres
```

Запустить postgres shell:

```
psql
```

Создать расширение:

```
CREATE EXTENSION pg_cron;
```

Проверить установку расширения:

```
SELECT * FROM pg_extension WHERE extname = 'pg_cron';
```

Выйти из postgres shell:

```
\q
```

Выйти из профиля postgres:

```
exit
```

Установка postgis

Установить postgis:

```
sudo dnf install postgis-pgsql-18.x86_64
```

Зайти под профилем postgres:

```
su - postgres
```

Запустить postgres shell:

```
psql
```

Создать расширение:

```
CREATE EXTENSION postgis;
```

Проверить установку расширения:

```
SELECT * FROM pg_extension WHERE extname = 'postgis';
```

Выйти из postgres shell:

```
\q
```

Выйти из профиля postgres:

```
exit
```

Установка Docker

Зайти под профилем root:

```
su -
```

Установить Docker:

```
dnf install docker-ce docker-ce-cli
```

Запустить службу и добавить её в автозагрузку:

```
systemctl enable docker --now
```

Проверить статус службы:

```
systemctl status docker
```

Выйти из профиля root:

```
exit
```

Установка NGINX

Зайти под профилем root:

```
su -
```

Установить NGINX:

```
dnf install nginx
```

Запустить службу и добавить её в автозагрузку:

```
systemctl enable nginx --now
```

Проверить статус службы:

```
systemctl status nginx
```

Выйти из профиля root:

```
exit
```

Установка Node.js

Установить Node.js

```
sudo dnf install nodejs
```

2.2. Настройка рабочей среды

После настройки серверного ПО и создания рабочей базы данных, перейти в домашний каталог:

```
cd ~
```

Перейти в домашний каталог и создать каталоги для сборки проектов и хранения данных:

```
mkdir -p {code,data}
```

2.3. Установка прикладного ПО Сервис считывания данных

Создать каталог для данных проекта:

```
mkdir -p data/pipeserver2/{config,logs}
```

Поместить файл конфигурации Config.xml в каталог data/pipeserver2/config .

Перейти в каталог:

```
cd code
```

Скачать код проекта.

Перейти в каталог:

```
cd pipeserver2
```

Собрать образ:

```
docker build -t pipeserver2:latest -f /home/master/code/pipeserver2/PipeServer2/Dockerfile .
```

Создать и запустить контейнер:

```
docker run --name pipeserver2 -d -v /home/master/data/pipeserver2/config:/app/config -v /home/master/data/pipeserver2/logs:/app/logs pipeserver2:latest
```

Примечание:

Контейнер сервиса считывания данных не запущен, так как демонстрационная среда изолирована и не имеет подключения к источнику данных.

Сервис фильтрации данных

Перейти в каталог

```
cd code
```

Скачать код проекта.

Перейти в каталог

```
cd migrate_to_postgres
```

Собрать образ

```
docker build -t filter.webapi:latest -f /home/master/code/migrate_to_postgres/Filter.WebApi/Dockerfile .
```

Создать и запустить контейнер

```
docker run --name filter.webapi -e ConnectionStrings__migrate="Host=172.17.0.1;Port=5432;Database=gtk;Include Error Detail=true;CommandTimeout=90;Username=postgres;Password=*****" -d filter.webapi:latest
```

Примечание:

Контейнер сервиса фильтрации данных не запущен так как демонстрационная среда изолирована и в неё не поступают новые данные из источника данных.

ПО АСУ ГТК**Серверная часть**

Перейти в каталог

```
cd code
```

Скачать код проекта.

Перейти в каталог:

```
cd ais-dotnet-api
```

Собрать образ:

```
docker build -t ais-dotnet-api:latest .
```

Создать и запустить контейнер:

```
docker run --name ais-dotnet-api --restart=always -d -p 5212:5212 -e  
ConnectionStrings__Postgres="Host=172.17.0.1;Port=5432;Database=gtk;Include Error  
Detail=true;CommandTimeout=90;Username=postgres;Password=*****" ais-dotnet-api:latest
```

Клиентская часть

Перейти в каталог:

```
cd code
```

Скачать код проекта.

Перейти в каталог:

```
cd ais-react-app-v2
```

Поместить файл конфигурации .env в каталог code/ais-react-app-v2.

Установить зависимости:

```
npm ci
```

Выполнить сборку:

```
npm run build
```

Публикация клиентской части

Перейти в домашний каталог:

```
cd ~
```

Зайти под профилем root:

```
su -
```

Создать каталог для публикации клиентской части:

```
mkdir -p /var/www/gtk
```

Скопировать содержимое каталога build:

```
cp -r /home/master/code/ais-react-app-v2/build/* /var/www/gtk/
```

Скопировать содержимое каталога assets в каталог /var/www/gtk/assets

Назначить права доступа:

```
chmod -R 755 /var/www/gtk/
```

Выйти из профиля root:

```
exit
```

Настройка NGINX

Зайти под профилем root:

```
su -
```

Открыть файл конфигурации:

```
nano /etc/nginx/nginx.conf
```

Закомментировать блок кода:

```
# server {  
#     listen      80;  
#     listen      [::]:80;  
#     server_name _;  
#     root        /usr/share/nginx/html;  
  
# Load configuration files for the default server block.  
#     include /etc/nginx/default.d/*.conf;  
# }
```

Внести блок кода непосредственно над ранее закомментированным:

```
server {  
    listen 80;  
    server_name _;  
    root /var/www/gtk;  
    index index.html;  
    access_log /var/log/nginx/gtk_access.log;  
    error_log /var/log/nginx/gtk_error.log;  
}
```

Сохранить изменения Ctrl+O .

Закреть Ctrl+X .

Проверить конфигурацию на наличие ошибок:

```
nginx -t
```

Перезаписать правила Selinux:

```
restorecon -Rv /var/www/gtk/
```

Перезапустить NGINX:

```
systemctl restart nginx
```

Выйти из профиля root:

```
exit
```

3. Порядок проверки работоспособности

Для входа в ПО АСУ ГТК необходимо открыть браузер и пройти по ссылке.

`http://localhost/index.html`

При успешной установке и настройке должен отобразиться экран авторизации ПО.

На экране авторизации необходимо ввести логин и пароль пользователя, нажать кнопку «ВОЙТИ» и войти в ПО.

При успешной авторизации отобразится интерфейс ПО АСУ ГТК.

В случае неуспешной авторизации будет выведено сообщение о неверных реквизитах доступа.

4. Эксплуатация, техническое обслуживание, ремонт и хранение компонентов системы

Специального регламентного обслуживания Система не требует.

Контроль над используемым дисковым пространством БД осуществляется штатными средствами администрирования системы и СУБД.

Рекомендуется проводить периодические профилактические работы на серверной платформе, физическую чистку от пыли, проверку системы охлаждения и электропитания, а также регулярное обновление программного обеспечения и резервное копирование данных.

5. Действия при возникновении ошибок и неполадок

В случае возникновения любой критической ошибки при использовании системы, требуется зафиксировать условия возникновения ошибки и связаться с разработчиком.

Критической ошибкой в Системе является любая невозможность использования основного функционала.