Информационная система
Мониторинг планово-экономической деятельности 2.0
(ИС МПЭД 2.0)

ИНСТРУКЦИЯ АДМИНИСТРАТОРА (ДЛЯ УСТАНОВКИ ПО)

Санкт-Петербург
2024

Оглавление

Принятые термины и сокращения

Термин	Определение
ИС МПЭД 2.0 / Система	Информационная система Мониторинг планово-экономической деятельности 2.0
ПО	Программное обеспечение
СУБД	Система управления базами данных

1. Общие положения

Настоящая Инструкция системного администратора разработана с целью:

- описания архитектуры системы и взаимосвязи её компонентов;
- описания процесса установки и настройки системы;
- определения порядка диагностирования проблем функционирования системы;
- упорядочения работы должностных лиц, связанной с диагностированием проблем функционирования системы.

В настоящем документе регламентируются действия при выполнении следующих мероприятий:

- установка и настройка системы;
- диагностирование проблем функционирования системы.

2. Описание архитектуры ИС МПЭД 2.0

ИС МПЭД 2.0 представляет собой программно-аппаратный комплекс, состоящий из серверов приложений, сервера базы данных и установленного на них ПО.
Схема развертывания ИС МПЭД 2.0 в инфраструктуре Заказчика представлена на Рис.1

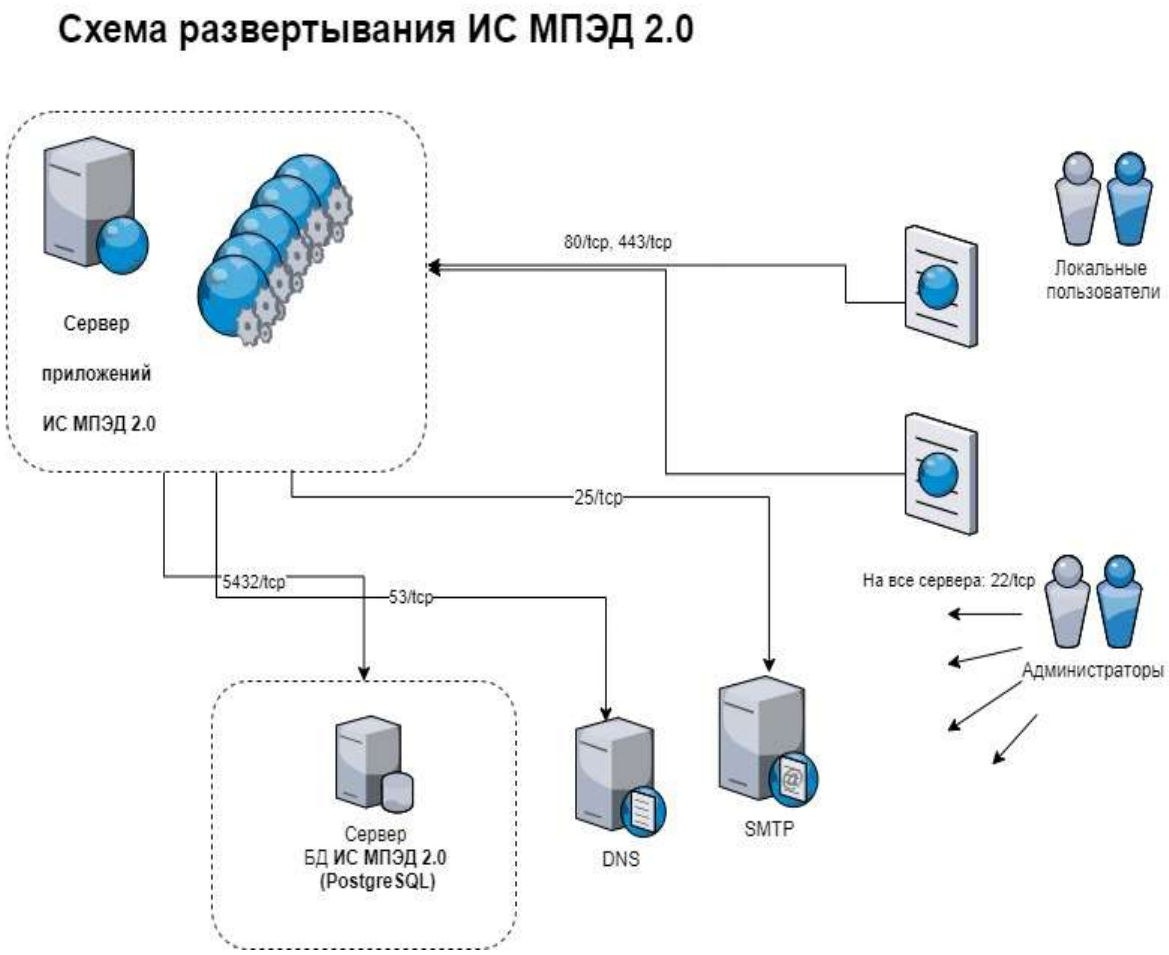

Рис 1. Схема развертывания ИС МПЭД 2.0

ИС МПЭД 2.0 включает в себя следующие серверы:

- сервер баз данных;
- сервер приложений.

На всех серверах системы установлена операционная система РЕД ОС версии 8.0.

Сервер приложений Node.JS версии 20.18.2.

На сервере приложений установлен менеджер процессов pm2 версии 6.0.9.

На сервере приложений установлен брокер сообщений RabbitMQ версии 3.13.7.

На сервере баз данных установлена СУБД PostgreSQL версии 15.13.

3. Установка и настройка системы

3.1 Установка и настройка PostgreSQL

Установка СУБД PostgreSQL производится на сервере баз данных.

3.1.1 Проверить доступность PostgreSQL соответствующей версии в репозитории:
sudo dnf list postgresql15 --showduplicates

3.1.2 Установить PostgreSQL версии 15, а также необходимые расширения из репозитория:
sudo dnf install postgresql15 postgresql15-contrib plpgsql_check-15 orafce-15

3.1.3 Открыть порт для подключения к PostgreSQL на брандмауэре брандмауэре (если установлен в ОС), по-умолчанию это порт 5432/tcp:

- добавить порт
  sudo firewall-cmd --permanent --zone=public --add-port=5432/tcp
- перезапустить firewall
  sudo firewall-cmd --reload
- Проверить включение правила
  sudo firewall-cmd --list-all

3.1.4 Инициализировать базу данных:
sudo /usr/pgsql-15/bin/postgresql-15-setup initdb
3.1.5 Запустить и включить в автозапуск службу
sudo systemctl enable --now postgresql-15

3.1.6 Проверить доступность и установленную версию СУБД:
sudo -u postgres psql -c "SELECT version();"

3.1.7 Настроить парольную аутентификацию в файле конфигурации ***pg_hba.conf*** (расположение по-умолчанию ***/var/lib/pgsql/15/data/pg_hba.conf***):

```
# Remote connection
host   all          all          all                scram-sha-256
```

В разделе LAN connections можно ограничить доступ только из определенных сетей, указав соответствующую маску.

3.1.8 Настроить следующие параметры в файле конфигурации ***postgresql.conf*** (расположение по-умолчанию ***/var/lib/pgsql/15/data/postgresql.conf***):

```
#Включаем прослушивание на внешних интерфейсах системы
listen_addresses = '*'
#Устанавливаем количество соединений
max_connections = 200
#Включаем уровень записи в WAL, необходимый для восстановления из резервной копии
wal_level = replica
#Включаем режим архивации
archive_mode = on
#Устанавливаем команду архивации, вместо /var/lib/pgsql/15/data/wal_backup/ должен быть каталог, в котором будут размещаться резервные копии WAL
```

```
archive_command = 'test ! -f /var/lib/pgsql/15/data/wal_backup/%f.gz && /usr/bin/gzip -c %p > /opt/ postgresql /wal_backup/%f.gz'
#Настраиваем политику удержания WAL
wal_keep_segments = 60
#Параметры локализации
lc_messages = 'en_US.UTF-8'
lc_monetary = 'ru_RU.UTF-8'
lc_numeric = 'ru_RU.UTF-8'
lc_time = 'ru_RU.UTF-8'
#Параметры отображения даты
datestyle = 'iso, mdy'
# Дополнительные параметры
site.user = 'asuse'
site.action = '00000000-0000-0000-0000-000000000000'
```

3.1.9 Перезапустить сервис PostgreSQL для применения новых параметров:
sudo systemctl restart postgresql-15

3.1.10 Запустить импорт начальной базы данных
su - postgres
gunzip -c mped.dump.gz | psql postgres --echo-errors &> import.log
3.1.11 Задать пароли суперпользователю СУБД ***postgres*** и мастер пользователю системы ***asuse***:
sudo -u postgres psql
\password postgres
\password asuse
\q

3.1.12 Настройка резервного копирования

Создать скрипт резервного копирования. В данном примере каталог запуска PostgreSQL - ***/var/lib/pgsql/15/data***, на сервере хранятся 2 последних базовых резервных копии (в каталоге ***db_backup***) и WAL за последние 2-е суток (в каталоге ***wal_backup***):

```
#!/bin/bash

PG_HOME=/var/lib/pgsql/15/data
export PG_HOME
mkdir $PG_HOME/pg_backup
/usr/bin/pg_basebackup -U postgres -D $PG_HOME/pg_backup -Ft -z -Xf

INDEX=$(date +"%u")

test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && rm $PG_HOME/db_backup/base.${INDEX}.tar.gz
cp $PG_HOME/pg_backup/base.tar.gz $PG_HOME/db_backup/base.${INDEX}.tar.gz

test -e $PG_HOME/db_backup/base.last.tar.gz && rm $PG_HOME/db_backup/base.last.tar.gz
ln $PG_HOME/db_backup/base.${INDEX}.tar.gz $PG_HOME/db_backup/base.last.tar.gz

rm -r $PG_HOME/pg_backup
test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && test -e $PG_HOME/db_backup/base.$(date --date="-1 day" +"%u").tar.gz &&  find $PG_HOME/db_backup -type f -mtime +1 -exec rm {} \;
find $PG_HOME/wal_backup -type f -mtime +2 -exec rm {} \;
```

Настроить регулярный запуск скрипта резервного копирования по необходимому расписанию.

3.2 Установка и настройка брокера сообщений RabbitMQ

Установка брокера сообщений RabbitMQ производится на сервере приложений.

3.2.1 Установить RabbitMQ
sudo dnf install rabbitmq-server

3.2.2 Запустить и включить в автозапуск службу
sudo systemctl enable --now rabbitmq-server

3.2.3 Включить плагины для управления
sudo rabbitmq-plugins enable rabbitmq_management

3.2.3 Настроить RabbitMQ

создать пользователя с правами администратора
sudo rabbitmqctl add_user 'rabbit' 'DMGfskPT4TqMH4PE'
sudo rabbitmqctl set_user_tags rabbit administrator
sudo rabbitmqctl set_permissions --vhost '/' 'rabbit' ".*" ".*" ".*"
удалить гостевого пользователя по-умолчанию
sudo rabbitmqctl delete_user guest
создать пользователя asuse для работы сервисов
sudo rabbitmqctl add_user 'asuse' 'raTu2JMsbDtRAJ6t'
создать vhost asuse_notify для работы сервисов
sudo rabbitmqctl add_vhost asuse_notify
выдать на него права администратору (rabbit) и служебному пользователю (asuse)
sudo rabbitmqctl set_permissions --vhost 'asuse_notify' 'asuse' ".*" ".*" ".*"
sudo rabbitmqctl set_permissions --vhost 'asuse_notify' 'rabbit' ".*" ".*" ".*"

3.2.4 Открыть необходимые порты для подключения к интерфейсу администрирования на брандмауэре (если установлен в ОС), по-умолчанию это порт 15672/tcp:

- добавить порт
  sudo firewall-cmd --permanent --zone=public --add-port=15672/tcp
- перезапустить firewall
  sudo firewall-cmd --reload
- проверить включение правила
  sudo firewall-cmd --list-all

3.3 Установка и настройка сервисов под управлением Node.JS

3.3.1 Установка и настройка Node.JS

Установка Node.JS производится на сервере приложений.

3.3.1.1 Проверить доступность Node.JS соответствующей версии в репозитории:
dnf list nodejs.x86_64 –showduplicates

3.3.1.2 Установить Node.JS версии 20 из репозитория:
sudo dnf install nodejs-20.18.2

3.3.1.3 Проверить установленную версию Node.JS и npm:
node -v
npm –v

3.3.1.4 В случае необходимости запуска приложения на портах <1024, предоставить привилегии на привязку к этим портам:
sudo setcap 'cap_net_bind_service=+ep' $(readlink -f $(which node))

3.3.1.5 Открыть необходимые порты для подключения к приложениям на брандмауэре (если установлен в ОС), в данном случае приложение запущено на порту 3100/tcp:

- добавить порт
  sudo firewall-cmd --permanent --zone=public --add-port=3100/tcp
- перезапустить firewall
  sudo firewall-cmd --reload
- проверить включение правила
  sudo firewall-cmd --list-all

3.3.1.6 Создать пользователя и при необходимости задать пароль пользователю ***webuser***, от имени которого будут выполняться web-приложения:
sudo useradd webuser
sudo passwd webuser

3.3.1.7 Если доступ сервера в интернет осуществляется через прокси-сервер, установить параметры доступа для npm через него:
npm config set https-proxy http://x.x.x.x:xxxx
npm config set proxy http:// x.x.x.x:xxxx

3.3.1.8 Установить менеджер процессов pm2
sudo npm install -g pm2

3.3.1.9 Если доступ сервера в интернет осуществляется через прокси-сервер, установить параметры доступа для npm от имени пользователя ***webuser*** через него:
sudo su - webuser
npm config set https-proxy http://x.x.x.x:xxxx
npm config set proxy http:// x.x.x.x:xxxx

3.3.1.10 Установить модуль управления ротацией логов для pm2 для пользователя webuser.
sudo su - webuser
pm2 install pm2-logrotate

3.3.1.11 Настроить автоматический перезапуск приложений при перезагрузке сервера.
sudo su - webuser
pm2 startup

3.3.2 Запуск и мониторинг сервисов

3.3.2.1 Создать необходимую структуру каталогов для работы сервисов

mkdir -p /home/webuser/asuse-ai/{node,config,files,logs}
chown -R webuser:webuser /home/webuser/asuse-ai

3.3.2.2 Создать файл конфигурации

Для поддержания непрерывной работы сервисов под управлением Node.JS при помощи менеджера процессов pm2 создается файл ***ecosystem.config.js*** в каталоге ***/home/webuser/asuse-ai/config***, например:

```
'use strict';
module.exports = {
   apps: [
      {
      name: 'asuse-ai-config',
      script: 'npm start',
      cwd: '/home/webuser/asuse-ai/node/asuse-ai-config/',
      watch: false,
      ignore_watch: ["public"],
      exp_backoff_restart_delay: 100,
      env: {
         'NODE_ENV': 'production',
         'ASUSE_CONF_URL': 'http://mped.domain.tld:3000',
         'ASUSE_CONF_DB': 'postgresql://asuse:password@mped.domain.tld:5432/config?pool_max=20'
         }
      },
      {
      name: 'asuse-ai-admin',
      script: 'npm start',
      cwd: '/home/webuser/asuse-ai/node/asuse-ai-admin/',
      watch: false,
      ignore_watch: ["public"],
      exp_backoff_restart_delay: 100,
      env: {
         'NODE_ENV': 'production',
         'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
         }
      },
      {
            name: 'asuse-ai-api-gateway',
      script: 'npm start',
      cwd: '/home/webuser/asuse-ai/node/asuse-ai-api-gateway/',
      watch: false,
      ignore_watch: ["public"],
      exp_backoff_restart_delay: 100,
      env: {
         'NODE_ENV': 'production',
         'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
         }
      },
      {
            name: 'asuse-ai-auth',
      script: 'npm start',
      cwd: '/home/webuser/asuse-ai/node/asuse-ai-auth/',
      watch: false,
      ignore_watch: ["public"],
      exp_backoff_restart_delay: 100,
      env: {
         'NODE_ENV': 'production',
```

```
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-databases',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-databases/',
watch: false,
ignore_watch: ["public"],
exp_backoff_restart_delay: 100,
env: {
    'NODE_ENV': 'production',
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-file-storage',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-file-storage/',
watch: false,
ignore_watch: ["public"],
exp_backoff_restart_delay: 100,
env: {
    'NODE_ENV': 'production',
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-meta',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-meta/',
watch: false,
ignore_watch: ["public"],
exp_backoff_restart_delay: 100,
env: {
    'NODE_ENV': 'production',
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-mped',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-mped/',
watch: false,
ignore_watch: ["public"],
exp_backoff_restart_delay: 100,
env: {
    'NODE_ENV': 'production',
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-notify-service',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-notify-service/',
watch: false,
ignore_watch: ["public"],
exp_backoff_restart_delay: 100,
env: {
    'NODE_ENV': 'production',
    'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
    }
},
{
        name: 'asuse-ai-user-data',
script: 'npm start',
cwd: '/home/webuser/asuse-ai/node/asuse-ai-user-data/',
watch: false,
ignore_watch: ["public"],
```

```
      exp_backoff_restart_delay: 100,
      env: {
         'NODE_ENV': 'production',
         'ASUSE_CONF_URL': 'http://mped.domain.tld:3000'
         }
      }
   ]
};
```

3.3.2.3 Настройка конфигурации сервисов

Настройка конфигурации сервисов производится при помощи внесения необходимых параметров (подключение к БД, адрес и порт сервиса) для соответствующего сервиса в столбец *config_json* таблицы *config.t_config* базы данных *config* .

3.3.2.4 Управление сервисами

Для запуска и мониторинга сервисов используется менеджер процессов pm2.
Запуск сервисов: **pm2 start ecosystem.config.js –update-env**
Список сервисов: **pm2 list**
Перезапуск сервиса: **pm2 restart asuse-ai-service** (перезапуск сервиса с именем asuse-ai-service) или **pm2 restart 2** (перезапуск сервиса с id=2)
Остановка сервиса: **pm2 stop asuse-ai-service** (остановка сервиса с именем asuse-ai-service) или **pm2 stop all** для остановки всех сервисов
Просмотр журналов сервиса: **pm2 logs 2 --lines 200** (отображает последние 200 строк лога приложения с id=2)

4. Диагностика неисправностей системы

4.1. Диагностика сервисов под управлением Node.JS

4.1.1. Для запуска мониторинга состояния сервисов под управлением Node.JS можно выполнить команду:
pm2 monit
4.1.2. Для просмотра состояния сервисов под управлением Node.JS можно выполнить команду:
pm2 list

4.1.3. Для просмотра журналов выполняется команда:
pm2 logs <название или id сервиса> --lines <количество последних строк лога>

Журналы сервисов (потоки stdout и stderr) так же доступны в каталоге ***/home/webuser/.pm2/logs***

Собственные журналы сервисов доступны в каталоге ***/home/webuser/asuse-ai/node/logs***

4.2. Диагностика состояния СУБД
Журналы базы данных хранятся в течение 7 дней и располагаются в директории ***/var/lib/pgsql/15/data*** на сервере БД.