

**ПРОГРАММНЫЙ ПРОДУКТ
СИСТЕМА УПРАВЛЯЮЩИХ ВОЗДЕЙСТВИЙ
INNOSTAGE ORCHESTRATOR**

Руководство по развертыванию

2022

Программный продукт
Система управляющих воздействий
Innostage Orchestrator
Руководство по развертыванию

АННОТАЦИЯ

Данный документ предназначен для специалистов, выполняющих администрирование программного продукта «Система управляющих воздействий **Innostage Orchestrator**» (далее - **Система**), и включает описание действий по установке и настройке **Системы** в операционных системах Astra Linux, РЕД ОС и Ubuntu.

СОДЕРЖАНИЕ

| | | |
|----------|---|----------|
| 1 | Требования к программно-аппаратной части и персоналу | 6 |
| 1.1 | Требования к серверной части | 6 |
| 1.2 | Требования к рабочим станциям..... | 6 |
| 1.3 | Требования к квалификации персонала | 6 |
| 2 | Развертывание Системы в ОС Ubuntu | 7 |
| 2.1 | Установка окружения..... | 7 |
| 2.1.1 | Создание служебной учетной записи ОС | 7 |
| 2.1.2 | Установка СУБД PostgreSQL | 7 |
| 2.1.3 | Установка службы очереди сообщений Redis | 7 |
| 2.1.4 | Установка системных библиотек ОС | 8 |
| 2.1.5 | Установка ассоциаций ОС | 8 |
| 2.1.6 | Настройка каталога приложений | 9 |
| 2.2 | Развертывание базы данных..... | 10 |
| 2.2.1 | Настройка конфигурационного файла | 10 |
| 2.2.2 | Настройка доступа пользователей | 10 |
| 2.2.3 | Создание пользователя и базы данных | 10 |
| 2.3 | Установка Системы..... | 11 |
| 2.3.1 | Установка переменных окружения..... | 11 |
| 2.3.2 | Установка дистрибутива..... | 11 |
| 2.3.3 | Настройка Системы..... | 12 |
| 2.3.4 | Инициализация базы данных | 12 |
| 2.4 | Настройка сервисов и логирования | 13 |
| 2.4.1 | Настройка сервисов | 13 |
| 2.4.2 | Настройка логирования сервисов | 15 |
| 2.5 | Запуск Системы | 15 |

| | | |
|----------|--|-----------|
| 3 | Развертывание Системы в ОС Astra Linux..... | 16 |
| 3.1 | Установка окружения..... | 16 |
| 3.1.1 | Создание служебной учетной записи ОС | 16 |
| 3.1.2 | Установка СУБД PostgreSQL | 16 |
| 3.1.3 | Установка службы очереди сообщений Redis | 16 |
| 3.1.4 | Установка системных библиотек ОС | 17 |
| 3.1.5 | Установка ассоциаций ОС | 17 |
| 3.1.6 | Настройка каталога приложений | 18 |
| 3.2 | Развертывание базы данных..... | 19 |
| 3.2.1 | Настройка конфигурационного файла | 19 |
| 3.2.2 | Настройка доступа пользователей | 19 |
| 3.2.3 | Создание пользователя и базы данных | 19 |
| 3.3 | Установка Системы..... | 20 |
| 3.3.1 | Установка переменных окружения..... | 20 |
| 3.3.2 | Установка дистрибутива..... | 20 |
| 3.3.3 | Настройка Системы..... | 21 |
| 3.3.4 | Инициализация базы данных | 21 |
| 3.4 | Настройка сервисов и логирования | 22 |
| 3.4.1 | Настройка сервисов..... | 22 |
| 3.4.2 | Настройка логирования сервисов | 24 |
| 3.5 | Запуск Системы | 24 |
| 4 | Развертывание Системы в РЕД ОС..... | 25 |
| 4.1 | Установка окружения..... | 25 |
| 4.1.1 | Создание служебной учетной записи ОС | 25 |
| 4.1.2 | Установка СУБД PostgreSQL | 25 |
| 4.1.3 | Установка службы очереди сообщений Redis | 25 |
| 4.1.4 | Установка системных библиотек ОС | 26 |
| 4.1.5 | Установка ассоциаций ОС | 26 |
| 4.1.6 | Настройка каталога приложений | 27 |

| | | |
|-------|---|-----------|
| 4.2 | Развертывание базы данных | 28 |
| 4.2.1 | Настройка конфигурационного файла | 28 |
| 4.2.2 | Настройка доступа пользователей | 28 |
| 4.2.3 | Создание пользователя и базы данных | 28 |
| 4.3 | Установка Системы | 29 |
| 4.3.1 | Установка переменных окружения | 29 |
| 4.3.2 | Установка дистрибутива | 29 |
| 4.3.3 | Настройка Системы | 30 |
| 4.3.4 | Инициализация базы данных | 30 |
| 4.4 | Настройка сервисов и логирования | 31 |
| 4.4.1 | Настройка сервисов | 31 |
| 4.4.2 | Настройка логирования сервисов | 33 |
| 4.5 | Запуск Системы | 33 |
| | Перечень используемых сокращений | 34 |

1 ТРЕБОВАНИЯ К ПРОГРАММНО-АППАРАТНОЙ ЧАСТИ И ПЕРСОНАЛУ

1.1 Требования к серверной части

Все компоненты **Системы** устанавливаются на один виртуальный либо физический сервер под управлением операционной системы Astra Linux 2.12, РЕД ОС 7.3, Ubuntu 20.04.

Минимальные требования к аппаратной части:

- Процессор: не менее 2 ГГц, 4 ядра.
- Оперативная память: не менее 6 Гб.
- Дисковое пространство: не менее 100 Гб свободного дискового пространства.

Требования к программной части:

- ОС (64-разрядная): Astra Linux 2.12, РЕД ОС 7.3, Ubuntu 20.04.
- СУБД: Postgres Pro 11 и выше, PostgreSQL 11 и выше.
- Python 3.8.

1.2 Требования к рабочим станциям

Для работы с **Системой** рекомендуется использовать автоматизированное рабочее место со следующими аппаратными требованиями:

- Монитор с разрешением 1280x1024 и выше.
- Масштабирование дисплея 100%.

В качестве веб-обозревателя рекомендуется использовать актуальные версии браузера Google Chrome, Яндекс.Браузер, FireFox или Microsoft Edge.

1.3 Требования к квалификации персонала

Администратор **Системы** должен обладать квалификацией, обеспечивающей:

- базовые навыки администрирования ОС семейства Linux (настройка репозитория, системные настройки и т. д.);
- базовые навыки работы с СУБД PostgreSQL.

2 РАЗВЕРТЫВАНИЕ СИСТЕМЫ В ОС UBUNTU

2.1 Установка окружения

2.1.1 Создание служебной учетной записи ОС

Создать пользователя orchestrator, добавить пользователя в группу sudo:

```
# Создание пользователя orchestrator
sudo adduser orchestrator

# Добавление пользователя в группу sudo
sudo usermod -aG sudo orchestrator

# Вход под пользователем orchestrator
su orchestrator
```

2.1.2 Установка СУБД PostgreSQL

Установить СУБД PostgreSQL и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:

```
# Установка PostgreSQL и необходимых компонентов
sudo apt install postgresql postgresql-contrib

# Добавление сервиса в автозагрузку
sudo systemctl enable postgresql

# Запуск сервиса
sudo service postgresql start
```

2.1.3 Установка службы очереди сообщений Redis

Установить службу очереди сообщений Redis и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:

```
# Установка Redis и необходимых компонентов
```

```
sudo apt install redis-server
```

```
# Добавление сервиса в автозагрузку и запуск
```

```
sudo systemctl enable redis-server
```

```
sudo service redis-server start
```

2.1.4 Установка системных библиотек ОС

Установить необходимые системные библиотеки ОС – открыть терминал и выполнить следующие команды:

```
# Установка пакетов компиляции gcc
```

```
sudo apt install gcc
```

```
# Установка дополнительных системных библиотек Python
```

```
sudo apt install python3-dev
```

```
sudo apt install python3-pip
```

```
sudo apt install virtualenv
```

```
# Установка библиотек программных интерфейсов
```

```
sudo apt install libpq-dev
```

```
sudo apt install libsasl2-dev libldap2-dev libssl-dev
```

```
# Установка библиотек Kerberos
```

```
sudo apt install krb5-config krb5-user libkrb5-dev
```

2.1.5 Установка ассоциаций ОС

Создать ассоциации ОС для python и python-config на версию Python 3.8:


```
# Проверка доступных версий python и python-config
ls /usr/bin/python*

# Проверка доступных ассоциаций python и python-config
sudo update-alternatives --list python
sudo update-alternatives --list python-config

# Добавление новых ассоциаций
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.8 1
sudo update-alternatives --install /usr/bin/python-config python-config /usr/bin/python3.8-config 1

# Выбор ассоциаций
sudo update-alternatives --config python
sudo update-alternatives --config python-config

# Проверка примененных изменений
python -V
python-config --includes
```

2.1.6 Настройка каталога приложений

Создать и настроить папки для размещения **Системы**. Создать папку /opt/orchestrator, предоставить служебной учетной записи доступ к папке:

```
# Создание каталога orchestrator в opt
sudo mkdir /opt/orchestrator

# Назначение владельца каталога (рекурсивно)
sudo chown -R orchestrator:orchestrator /opt/orchestrator

# Предоставление прав (рекурсивно)
sudo chmod -R 775 /opt/orchestrator
```

2.2 Развертывание базы данных

2.2.1 Настройка конфигурационного файла

Открыть конфигурационный файл `postgresql.conf` в редакторе `nano` и указать доверенные адреса, с которых будет доступна база данных **Системы** (параметр `listen_addresses`), перезапустить сервис:

```
# Открытие конфигурационного файла postgresql.conf
sudo nano /etc/postgresql/12/main/postgresql.conf

# Установить значение параметра listen_addresses
# listen_addresses = 'localhost'

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

2.2.2 Настройка доступа пользователей

Открыть конфигурационный файл `pg_hba.conf` в редакторе `nano` и настроить доступ пользователей к базе данных **Системы** (секция `local`), перезапустить сервис:

```
# Открытие конфигурационного файла pg_hba.conf
sudo nano /etc/postgresql/12/main/pg_hba.conf

# Установить значение секции local
# local all all md5

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

2.2.3 Создание пользователя и базы данных

Открыть терминал и подключиться к консоли управления СУБД (`psql`) под пользователем `postgres`, создать базу данных и пользователя с необходимым паролем, проверить подключение:

```
# Подключение к консоли управления СУБД
sudo -u postgres psql

# Создание пользователя БД: <password> заменить на необходимое значение
psql => CREATE USER orchestrator WITH PASSWORD '<password>';

# Создание БД
psql => CREATE DATABASE orchestrator OWNER=orchestrator ENCODING=utf8;

# Добавление прав на БД
psql => GRANT ALL PRIVILEGES ON DATABASE orchestrator TO orchestrator;

# Выход из консоли
psql => \q

# Проверка подключения к созданной БД под созданным пользователем
psql -U orchestrator orchestrator
--/Ввести пароль/--
orchestrator => \q
```

2.3 Установка Системы

2.3.1 Установка переменных окружения

Добавить переменные окружения и осуществить перезагрузку ОС после добавления:

```
# Добавление переменной окружения AIRFLOW_HOME
echo "AIRFLOW_HOME=/opt/orchestrator" | sudo tee -a /etc/environment > /dev/null

# Добавление переменной окружения CONFIG_PROCESSOR_MANAGER_LOGGER
echo "CONFIG_PROCESSOR_MANAGER_LOGGER=True" | sudo tee -a /etc/environment > /dev/null

# Перезагрузка ОС
sudo reboot
```

2.3.2 Установка дистрибутива

Из распакованного архива с поставкой дистрибутивов **Системы** скопировать дистрибутивы на сервер, где должна быть развернута **Система**, в папку /opt/orchestrator.

Создать изолированное виртуальное окружение, активировать и осуществить установку **Системы**:

```
# Создание изолированного виртуального окружения
virtualenv /opt/orchestrator/venv

# Активация созданного виртуального окружения
source /opt/orchestrator/venv/bin/activate

# Установка дистрибутива
pip install -U /opt/orchestrator/nextstage-<version>-py3-none-any.whl
```

2.3.3 Настройка Системы

Произвести настройку **Системы**: проверить установленную версию, установить необходимые адреса БД, службы очереди сообщений.

```
# Проверка установленной версии системы
nextstage version

# Конфигурация подключения к БД
nextstage config db

# Конфигурация подключения к службе очереди сообщений:
# Redis, localhost, 6379
nextstage config broker
```

Произвести настройку параметров работы **Системы**.

Примечание: при использовании HTTPS, предварительно необходимо выпустить сертификат с расширением файла .pem и ключ с расширением файла .key, и разместить их на сервере в папке /opt/orchestrator.

```
# Конфигурация параметров работы системы:
# IP или HostName, HTTP, 8080, auth_db
nextstage config webserver

# Предоставление прав на бинд порта 443 в случае работы по HTTPS
sudo setcap CAP_NET_BIND_SERVICE=+eip /usr/bin/python3.8
```

2.3.4 Инициализация базы данных

Выполнить команду для инициализации базы данных **Системы**, создать пользователя системы с ролью «Администратор»:

```
# Инициализация БД
nextstage db init

# Создание системного пользователя с ролью "Администратор"
nextstage config create_admin
```

2.4 Настройка сервисов и логирования

2.4.1 Настройка сервисов

Создать сервис `webserver` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса webserver
sudo nano /etc/systemd/system/webserver.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Webserver

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow webserver'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=webserver
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис `scheduler` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса scheduler
sudo nano /etc/systemd/system/scheduler.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Scheduler

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow scheduler'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=scheduler
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис worker и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса worker
sudo nano /etc/systemd/system/worker.service

# Настройки запуска сервиса
[Unit]
Description = Orchestrator Worker

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow celery worker'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=worker
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Обновить данные ОС о наличии сервисов:

```
# Обновление данных о сервисах  
sudo systemctl daemon-reload
```

2.4.2 Настройка логирования сервисов

Добавить конфигурационный файл настроек логирования службы rsyslog для созданных сервисов и перезапустить службу:

```
# Создание конфигурационного файла для rsyslog  
sudo nano /etc/rsyslog.d/orchestrator.conf  
  
# Добавить в конфигурационный файл следующее содержимое  
if $programname == 'webserver' then /var/log/orchestrator/webserver.log  
if $programname == 'scheduler' then /var/log/orchestrator/scheduler.log  
if $programname == 'worker' then /var/log/orchestrator/worker.log  
  
# Перезапуск службы rsyslog  
sudo systemctl restart rsyslog
```

2.5 Запуск Системы

Управление запуском/остановкой/перезапуском **Системы** осуществляется с помощью созданных служб ОС: webserver, scheduler, worker.

Осуществить добавление данных сервисов в автозагрузку и запуск:

```
# Добавление сервисов в автозагрузку  
sudo systemctl enable worker  
sudo systemctl enable scheduler  
sudo systemctl enable webserver  
  
# Запуск сервисов  
sudo service worker start  
sudo service scheduler start  
sudo service webserver start
```

3 РАЗВЕРТЫВАНИЕ СИСТЕМЫ В ОС ASTRA LINUX

3.1 Установка окружения

3.1.1 Создание служебной учетной записи ОС

Создать пользователя orchestrator, добавить пользователя в группу sudo:

```
# Создание пользователя orchestrator
sudo adduser orchestrator

# Добавление пользователя в группу sudo
sudo usermod -aG sudo orchestrator

# Вход под пользователем orchestrator
su orchestrator
```

3.1.2 Установка СУБД PostgreSQL

Установить СУБД PostgreSQL и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:

```
# Установка PostgreSQL и необходимых компонентов
sudo apt install postgresql postgresql-contrib

# Добавление сервиса в автозагрузку
sudo systemctl enable postgresql

# Запуск сервиса
sudo service postgresql start
```

3.1.3 Установка службы очереди сообщений Redis

Установить службу очереди сообщений Redis и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:


```
# Установка Redis и необходимых компонентов
```

```
sudo apt install redis-server
```

```
# Добавление сервиса в автозагрузку и запуск
```

```
sudo systemctl enable redis-server
```

```
sudo service redis-server start
```

3.1.4 Установка системных библиотек ОС

Установить необходимые системные библиотеки ОС – открыть терминал и выполнить следующие команды:

```
# Установка пакетов компиляции gcc
```

```
sudo apt install gcc
```

```
# Установка дополнительных системных библиотек Python
```

```
sudo apt install python3-dev
```

```
sudo python3 -m pip install --upgrade pip
```

```
sudo apt install virtualenv
```

```
# Установка библиотек программных интерфейсов
```

```
sudo apt install libpq-dev
```

```
sudo apt install libsasl2-dev libldap2-dev libssl-dev
```

```
# Установка библиотек Kerberos
```

```
sudo apt install krb5-config krb5-user libkrb5-dev
```

3.1.5 Установка ассоциаций ОС

Создать ассоциации ОС для python и python-config на версию Python 3.8:

```
# Проверка доступных версий python и python-config
ls /usr/bin/python*

# Проверка доступных ассоциаций python и python-config
sudo update-alternatives --list python
sudo update-alternatives --list python-config

# Добавление новых ассоциаций
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.8 1
sudo update-alternatives --install /usr/bin/python-config python-config /usr/bin/python3.8-config 1

# Выбор ассоциаций
sudo update-alternatives --config python
sudo update-alternatives --config python-config

# Проверка примененных изменений
python -V
python-config --includes
```

3.1.6 Настройка каталога приложений

Создать и настроить папки для размещения **Системы**. Создать папку /opt/orchestrator, предоставить служебной учетной записи доступ к папке:

```
# Создание каталога orchestrator в opt
sudo mkdir /opt/orchestrator

# Назначение владельца каталога (рекурсивно)
sudo chown -R orchestrator:orchestrator /opt/orchestrator

# Предоставление прав (рекурсивно)
sudo chmod -R 775 /opt/orchestrator
```

3.2 Развертывание базы данных

3.2.1 Настройка конфигурационного файла

Открыть конфигурационный файл `postgresql.conf` в редакторе `nano` и указать доверенные адреса, с которых будет доступна база данных **Системы** (параметр `listen_addresses`), перезапустить сервис:

```
# Открытие конфигурационного файла postgresql.conf
sudo nano /etc/postgresql/12/main/postgresql.conf

# Установить значение параметра listen_addresses
# listen_addresses = 'localhost'

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

3.2.2 Настройка доступа пользователей

Открыть конфигурационный файл `pg_hba.conf` в редакторе `nano` и настроить доступ пользователей к базе данных **Системы** (секция `local`), перезапустить сервис:

```
# Открытие конфигурационного файла pg_hba.conf
sudo nano /etc/postgresql/12/main/pg_hba.conf

# Установить значение секции local
# local all all md5

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

3.2.3 Создание пользователя и базы данных

Открыть терминал и подключиться к консоли управления СУБД (`psql`) под пользователем `postgres`, создать базу данных и пользователя с необходимым паролем, проверить подключение:

```
# Подключение к консоли управления СУБД
sudo -u postgres psql

# Создание пользователя БД: <password> заменить на необходимое значение
psql => CREATE USER orchestrator WITH PASSWORD '<password>';

# Создание БД
psql => CREATE DATABASE orchestrator OWNER=orchestrator ENCODING=utf8;

# Добавление прав на БД
psql => GRANT ALL PRIVILEGES ON DATABASE orchestrator TO orchestrator;

# Выход из консоли
psql => \q

# Проверка подключения к созданной БД под созданным пользователем
psql -U orchestrator orchestrator
--/Ввести пароль/--
orchestrator => \q
```

3.3 Установка Системы

3.3.1 Установка переменных окружения

Добавить переменные окружения и осуществить перезагрузку ОС после добавления:

```
# Добавление переменной окружения AIRFLOW_HOME
echo "AIRFLOW_HOME=/opt/orchestrator" | sudo tee -a /etc/environment > /dev/null

# Добавление переменной окружения CONFIG_PROCESSOR_MANAGER_LOGGER
echo "CONFIG_PROCESSOR_MANAGER_LOGGER=True" | sudo tee -a /etc/environment > /dev/null

# Перезагрузка ОС
sudo reboot
```

3.3.2 Установка дистрибутива

Из распакованного архива с поставкой дистрибутивов **Системы** скопировать дистрибутивы на сервер, где должна быть развернута **Система**, в папку /opt/orchestrator.

Создать изолированное виртуальное окружение, активировать и осуществить установку **Системы**:

```
# Создание изолированного виртуального окружения
virtualenv /opt/orchestrator/venv

# Активация созданного виртуального окружения
source /opt/orchestrator/venv/bin/activate

# Установка дистрибутива
pip install -U /opt/orchestrator/nextstage-<version>-py3-none-any.whl
```

3.3.3 Настройка Системы

Произвести настройку **Системы**: проверить установленную версию, установить необходимые адреса БД, службы очереди сообщений.

```
# Проверка установленной версии системы
nextstage version

# Конфигурация подключения к БД
nextstage config db

# Конфигурация подключения к службе очереди сообщений:
# Redis, localhost, 6379
nextstage config broker
```

Произвести настройку параметров работы **Системы**.

Примечание: при использовании HTTPS, предварительно необходимо выпустить сертификат с расширением файла .pem и ключ с расширением файла .key, и разместить их на сервере в папке /opt/orchestrator.

```
# Конфигурация параметров работы системы:
# IP или HostName, HTTP, 8080, auth_db
nextstage config webserver

# Предоставление прав на бинд порта 443 в случае работы по HTTPS
sudo setcap CAP_NET_BIND_SERVICE=+eip /usr/bin/python3.8
```

3.3.4 Инициализация базы данных

Выполнить команду для инициализации базы данных **Системы**, создать пользователя системы с ролью «Администратор»:

```
# Инициализация БД
nextstage db init

# Создание системного пользователя с ролью "Администратор"
nextstage config create_admin
```

3.4 Настройка сервисов и логирования

3.4.1 Настройка сервисов

Создать сервис `webserver` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса webserver
sudo nano /etc/systemd/system/webserver.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Webserver

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow webserver'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=webserver
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис `scheduler` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса scheduler
sudo nano /etc/systemd/system/scheduler.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Scheduler

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow scheduler'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=scheduler
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис worker и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса worker
sudo nano /etc/systemd/system/worker.service

# Настройки запуска сервиса
[Unit]
Description = Orchestrator Worker

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow celery worker'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=worker
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Обновить данные ОС о наличии сервисов:

```
# Обновление данных о сервисах
sudo systemctl daemon-reload
```

3.4.2 Настройка логирования сервисов

Добавить конфигурационный файл настроек логирования службы rsyslog для созданных сервисов и перезапустить службу:

```
# Создание конфигурационного файла для rsyslog
sudo nano /etc/rsyslog.d/orchestrator.conf

# Добавить в конфигурационный файл следующее содержимое
if $programname == 'webserver' then /var/log/orchestrator/webserver.log
if $programname == 'scheduler' then /var/log/orchestrator/scheduler.log
if $programname == 'worker' then /var/log/orchestrator/worker.log

# Перезапуск службы rsyslog
sudo systemctl restart rsyslog
```

3.5 Запуск Системы

Управление запуском/остановкой/перезапуском **Системы** осуществляется с помощью созданных служб ОС: webserver, scheduler, worker.

Осуществить добавление данных сервисов в автозагрузку и запуск:

```
# Добавление сервисов в автозагрузку
sudo systemctl enable worker
sudo systemctl enable scheduler
sudo systemctl enable webserver

# Запуск сервисов
sudo service worker start
sudo service scheduler start
sudo service webserver start
```


4 РАЗВЕРТЫВАНИЕ СИСТЕМЫ В РЕД ОС

4.1 Установка окружения

4.1.1 Создание служебной учетной записи ОС

Создать пользователя `orchestrator`, добавить пользователя в группу `sudo`:

```
# Создание пользователя orchestrator
sudo adduser orchestrator
sudo passwd orchestrator

# Добавление пользователя в группу sudo
sudo usermod -aG wheel orchestrator

# Вход под пользователем orchestrator
su orchestrator
```

4.1.2 Установка СУБД PostgreSQL

Установить СУБД PostgreSQL и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:

```
# Установка PostgreSQL и необходимых компонентов
sudo yum install postgresql-server postgresql-contrib

# Инициализация
sudo /usr/bin/postgresql-setup --initdb

# Добавление сервиса в автозагрузку
sudo systemctl enable postgresql

# Запуск сервиса
sudo service postgresql start
```

4.1.3 Установка службы очереди сообщений Redis

Установить службу очереди сообщений Redis и добавить сервис в автозагрузку – открыть терминал и выполнить следующие команды:

```
# Установка Redis и необходимых компонентов
```

```
sudo yum install redis
```

```
# Добавление сервиса в автозагрузку и запуск
```

```
sudo systemctl enable redis
```

```
sudo service redis start
```

4.1.4 Установка системных библиотек ОС

Установить необходимые системные библиотеки ОС – открыть терминал и выполнить следующие команды:

```
# Установка пакетов компиляции gcc
```

```
sudo yum install gcc
```

```
# Установка дополнительных системных библиотек Python
```

```
sudo yum install python3-devel
```

```
sudo python -m pip install --upgrade pip
```

```
pip install virtualenv
```

```
# Установка библиотек программных интерфейсов
```

```
sudo yum install libpq-devel
```

```
sudo yum install openldap-devel openssl-devel
```

```
sudo yum install ldap2pg
```

```
# Установка библиотек Kerberos
```

```
sudo yum install krb5-workstation krb5-devel krb5-libs libkadm5
```

4.1.5 Установка ассоциаций ОС

Создать ассоциации ОС для python и python-config на версию Python 3.8:

```
# Проверка доступных версий python и python-config
ls /usr/bin/python*

# Проверка доступных ассоциаций python и python-config
sudo alternatives --display python
sudo alternatives --display python-config

# Добавление новых ассоциаций
sudo alternatives --install /usr/bin/python python /usr/bin/python3.8 1
sudo alternatives --install /usr/bin/python-config python-config /usr/bin/python3.8-config 1

# Выбор ассоциаций
sudo alternatives --config python
sudo alternatives --config python-config

# Проверка примененных изменений
python -V
python-config --includes
```

4.1.6 Настройка каталога приложений

Создать и настроить папки для размещения **Системы**. Создать папку /opt/orchestrator, предоставить служебной учетной записи доступ к папке:

```
# Создание каталога orchestrator в opt
sudo mkdir /opt/orchestrator

# Назначение владельца каталога (рекурсивно)
sudo chown -R orchestrator:orchestrator /opt/orchestrator

# Предоставление прав (рекурсивно)
sudo chmod -R 775 /opt/orchestrator
```

4.2 Развертывание базы данных

4.2.1 Настройка конфигурационного файла

Открыть конфигурационный файл `postgresql.conf` в редакторе `nano` и указать доверенные адреса, с которых будет доступна база данных **Системы** (параметр `listen_addresses`), перезапустить сервис:

```
# Открытие конфигурационного файла postgresql.conf
sudo nano /var/lib/pgsql/data/postgresql.conf

# Установить значение параметра listen_addresses
# listen_addresses = 'localhost'

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

4.2.2 Настройка доступа пользователей

Открыть конфигурационный файл `pg_hba.conf` в редакторе `nano` и настроить доступ пользователей к базе данных **Системы** (секция `local`), перезапустить сервис:

```
# Открытие конфигурационного файла pg_hba.conf
sudo nano /var/lib/pgsql/data/pg_hba.conf

# Установить значение секции local
# local all all md5
# Все ident изменить на md5

# Перезапуск сервиса postgresql
sudo systemctl restart postgresql
```

4.2.3 Создание пользователя и базы данных

Открыть терминал и подключиться к консоли управления СУБД (`psql`) под пользователем `postgres`, создать базу данных и пользователя с необходимым паролем, проверить подключение:

```
# Подключение к консоли управления СУБД
sudo -u postgres psql

# Создание пользователя БД: <password> заменить на необходимое значение
psql => CREATE USER orchestrator WITH PASSWORD '<password>';

# Создание БД
psql => CREATE DATABASE orchestrator OWNER=orchestrator ENCODING=utf8;

# Добавление прав на БД
psql => GRANT ALL PRIVILEGES ON DATABASE orchestrator TO orchestrator;

# Выход из консоли
psql => \q

# Проверка подключения к созданной БД под созданным пользователем
psql -U orchestrator orchestrator
--/Ввести пароль/--
orchestrator => \q
```

4.3 Установка Системы

4.3.1 Установка переменных окружения

Добавить переменные окружения и осуществить перезагрузку ОС после добавления:

```
# Добавление переменной окружения AIRFLOW_HOME
echo "AIRFLOW_HOME=/opt/orchestrator" | sudo tee -a /etc/environment > /dev/null

# Добавление переменной окружения CONFIG_PROCESSOR_MANAGER_LOGGER
echo "CONFIG_PROCESSOR_MANAGER_LOGGER=True" | sudo tee -a /etc/environment > /dev/null

# Перезагрузка ОС
sudo reboot
```

4.3.2 Установка дистрибутива

Из распакованного архива с поставкой дистрибутивов **Системы** скопировать дистрибутивы на сервер, где должна быть развернута **Система**, в папку /opt/orchestrator.

Создать изолированное виртуальное окружение, активировать и осуществить установку **Системы**:

```
# Создание изолированного виртуального окружения
virtualenv /opt/orchestrator/venv

# Активация созданного виртуального окружения
source /opt/orchestrator/venv/bin/activate

# Установка дистрибутива
pip install -U /opt/orchestrator/nextstage-<version>-py3-none-any.whl
```

4.3.3 Настройка Системы

Произвести настройку **Системы**: проверить установленную версию, установить необходимые адреса БД, службы очереди сообщений.

```
# Проверка установленной версии системы
nextstage version

# Конфигурация подключения к БД
nextstage config db

# Конфигурация подключения к службе очереди сообщений:
# Redis, localhost, 6379
nextstage config broker
```

Произвести настройку параметров работы **Системы**.

Примечание: при использовании HTTPS, предварительно необходимо выпустить сертификат с расширением файла .pem и ключ с расширением файла .key, и разместить их на сервере в папке /opt/orchestrator.

```
# Конфигурация параметров работы системы:
# IP или HostName, HTTP, 8080, auth_db
nextstage config webserver

# Предоставление прав на бинд порта 443 в случае работы по HTTPS
sudo setcap CAP_NET_BIND_SERVICE=+eip /usr/bin/python3.8
```

4.3.4 Инициализация базы данных

Выполнить команду для инициализации базы данных **Системы**, создать пользователя системы с ролью «Администратор»:

```
# Инициализация БД
nextstage db init

# Создание системного пользователя с ролью "Администратор"
nextstage config create_admin
```

4.4 Настройка сервисов и логирования

4.4.1 Настройка сервисов

Создать сервис `webserver` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса webserver
sudo nano /etc/systemd/system/webserver.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Webserver

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow webserver'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=webserver
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис `scheduler` и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса scheduler
sudo nano /etc/systemd/system/scheduler.service

# Настройки запуска сервиса (записать в файл сервиса)
[Unit]
Description = Orchestrator Scheduler

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow scheduler'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=scheduler
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```

Создать сервис worker и добавить настройки запуска в файле с настройками сервиса:

```
# Создание сервиса worker
sudo nano /etc/systemd/system/worker.service

# Настройки запуска сервиса
[Unit]
Description = Orchestrator Worker

[Service]
ExecStart=/bin/bash -c 'source /opt/orchestrator/venv/bin/activate && airflow celery worker'
Restart=always
RestartSec=30
User=orchestrator
SyslogIdentifier=worker
Environment="AIRFLOW_HOME=/opt/orchestrator"
Environment="CONFIG_PROCESSOR_MANAGER_LOGGER=True"

[Install]
WantedBy=multi-user.target
```


Обновить данные ОС о наличии сервисов:

```
# Обновление данных о сервисах  
sudo systemctl daemon-reload
```

4.4.2 Настройка логирования сервисов

Добавить конфигурационный файл настроек логирования службы rsyslog для созданных сервисов и перезапустить службу:

```
# Создание конфигурационного файла для rsyslog  
sudo nano /etc/rsyslog.d/orchestrator.conf  
  
# Добавить в конфигурационный файл следующее содержимое  
if $programname == 'webserver' then /var/log/orchestrator/webserver.log  
if $programname == 'scheduler' then /var/log/orchestrator/scheduler.log  
if $programname == 'worker' then /var/log/orchestrator/worker.log  
  
# Перезапуск службы rsyslog  
sudo systemctl restart rsyslog
```

4.5 Запуск Системы

Управление запуском/остановкой/перезапуском **Системы** осуществляется с помощью созданных служб ОС: webserver, scheduler, worker.

Осуществить добавление данных сервисов в автозагрузку и запуск:

```
# Добавление сервисов в автозагрузку  
sudo systemctl enable worker  
sudo systemctl enable scheduler  
sudo systemctl enable webserver  
  
# Запуск сервисов  
sudo service worker start  
sudo service scheduler start  
sudo service webserver start
```

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

| Сокращение | Полное наименование |
|-------------------|--|
| IP | Internet Protocol |
| БД | База данных |
| ОС | Операционная система |
| Система | Система управляющих воздействий Innostage Orchestrator |
| СУБД | Система управления базами данных |