**ПРОГРАММНЫЙ ПРОДУКТ**

**СИСТЕМА УПРАВЛЯЮЩИХ ВОЗДЕЙСТВИЙ   
INNOSTAGE ORCHESTRATOR**

**Руководство по развертыванию**

2022

**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](#_Toc100741499)

[1.1 Требования к серверной части 6](#_Toc100741500)

[1.2 Требования к рабочим станциям 6](#_Toc100741501)

[1.3 Требования к квалификации персонала 6](#_Toc100741502)

[2 Развертывание Системы в ОС Ubuntu 7](#_Toc100741503)

[2.1 Установка окружения 7](#_Toc100741504)

[2.1.1 Создание служебной учетной записи ОС 7](#_Toc100741505)

[2.1.2 Установка СУБД PostgreSQL 7](#_Toc100741506)

[2.1.3 Установка службы очереди сообщений Redis 7](#_Toc100741507)

[2.1.4 Установка системных библиотек ОС 8](#_Toc100741508)

[2.1.5 Установка ассоциаций ОС 8](#_Toc100741509)

[2.1.6 Настройка каталога приложений 9](#_Toc100741510)

[2.2 Развертывание базы данных 10](#_Toc100741511)

[2.2.1 Настройка конфигурационного файла 10](#_Toc100741512)

[2.2.2 Настройка доступа пользователей 10](#_Toc100741513)

[2.2.3 Создание пользователя и базы данных 10](#_Toc100741514)

[2.3 Установка Системы 11](#_Toc100741515)

[2.3.1 Установка переменных окружения 11](#_Toc100741516)

[2.3.2 Установка дистрибутива 11](#_Toc100741517)

[2.3.3 Настройка Системы 12](#_Toc100741518)

[2.3.4 Инициализация базы данных 12](#_Toc100741519)

[2.4 Настройка сервисов и логирования 13](#_Toc100741520)

[2.4.1 Настройка сервисов 13](#_Toc100741521)

[2.4.2 Настройка логирования сервисов 15](#_Toc100741522)

[2.5 Запуск Системы 15](#_Toc100741523)

[3 Развертывание Системы в ОС Astra Linux 16](#_Toc100741524)

[3.1 Установка окружения 16](#_Toc100741525)

[3.1.1 Создание служебной учетной записи ОС 16](#_Toc100741526)

[3.1.2 Установка СУБД PostgreSQL 16](#_Toc100741527)

[3.1.3 Установка службы очереди сообщений Redis 16](#_Toc100741528)

[3.1.4 Установка системных библиотек ОС 17](#_Toc100741529)

[3.1.5 Установка ассоциаций ОС 17](#_Toc100741530)

[3.1.6 Настройка каталога приложений 18](#_Toc100741531)

[3.2 Развертывание базы данных 19](#_Toc100741532)

[3.2.1 Настройка конфигурационного файла 19](#_Toc100741533)

[3.2.2 Настройка доступа пользователей 19](#_Toc100741534)

[3.2.3 Создание пользователя и базы данных 19](#_Toc100741535)

[3.3 Установка Системы 20](#_Toc100741536)

[3.3.1 Установка переменных окружения 20](#_Toc100741537)

[3.3.2 Установка дистрибутива 20](#_Toc100741538)

[3.3.3 Настройка Системы 21](#_Toc100741539)

[3.3.4 Инициализация базы данных 21](#_Toc100741540)

[3.4 Настройка сервисов и логирования 22](#_Toc100741541)

[3.4.1 Настройка сервисов 22](#_Toc100741542)

[3.4.2 Настройка логирования сервисов 24](#_Toc100741543)

[3.5 Запуск Системы 24](#_Toc100741544)

[4 Развертывание Системы в РЕД ОС 25](#_Toc100741545)

[4.1 Установка окружения 25](#_Toc100741546)

[4.1.1 Создание служебной учетной записи ОС 25](#_Toc100741547)

[4.1.2 Установка СУБД PostgreSQL 25](#_Toc100741548)

[4.1.3 Установка службы очереди сообщений Redis 25](#_Toc100741549)

[4.1.4 Установка системных библиотек ОС 26](#_Toc100741550)

[4.1.5 Установка ассоциаций ОС 26](#_Toc100741551)

[4.1.6 Настройка каталога приложений 27](#_Toc100741552)

[4.2 Развертывание базы данных 28](#_Toc100741553)

[4.2.1 Настройка конфигурационного файла 28](#_Toc100741554)

[4.2.2 Настройка доступа пользователей 28](#_Toc100741555)

[4.2.3 Создание пользователя и базы данных 28](#_Toc100741556)

[4.3 Установка Системы 29](#_Toc100741557)

[4.3.1 Установка переменных окружения 29](#_Toc100741558)

[4.3.2 Установка дистрибутива 29](#_Toc100741559)

[4.3.3 Настройка Системы 30](#_Toc100741560)

[4.3.4 Инициализация базы данных 30](#_Toc100741561)

[4.4 Настройка сервисов и логирования 31](#_Toc100741562)

[4.4.1 Настройка сервисов 31](#_Toc100741563)

[4.4.2 Настройка логирования сервисов 33](#_Toc100741564)

[4.5 Запуск Системы 33](#_Toc100741565)

[Перечень используемых сокращений 34](#_Toc100741566)

# Требования к программно-аппаратной части и персоналу

## 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: ­

* ОC (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. ­

## 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.

## Требования к квалификации персонала

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

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

# Развертывание Системы в ОС Ubuntu

## Установка окружения

### 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

### 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

### 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

### 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

### 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

### 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

## Развертывание базы данных

### 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

### 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

### 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

## Установка Системы

### 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

### 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

### 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 c расширением файла .pem и ключ c 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

### 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

## Настройка сервисов и логирования

### 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

### 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

## 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

# Развертывание Системы в ОС Astra Linux

## Установка окружения

### 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

### 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

### 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

### 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

### 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

### 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

## Развертывание базы данных

### 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

### 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

### 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

## Установка Системы

### 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

### 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

### 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 c расширением файла .pem и ключ c 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

### 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

## Настройка сервисов и логирования

### 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

### 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

## 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

# Развертывание Системы в РЕД ОС

## Установка окружения

### 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

### 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

### 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

### 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

### 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

### 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

## Развертывание базы данных

### 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

### 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

### 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

## Установка Системы

### 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

### 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

### 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 c расширением файла .pem и ключ c 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

### 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

## Настройка сервисов и логирования

### 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

### 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

## 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 |