

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ВАНГАРД СОФТ»**

**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА
ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ
ДЛЯ ДЕЖУРНО-ДИСПЕТЧЕРСКИХ СЛУЖБ**

Инструкция по установке

Листов 14

2024

АННОТАЦИЯ

Документ содержит инструкцию по установке программного обеспечения «Интеллектуальная система поддержки принятия решений для дежурно-диспетчерских служб» (далее - программа, ИСППР, приложение).

Программное обеспечение ИСППР предназначено для поддержки принятия решений дежурно-диспетчерских служб в отношении дорожно-транспортных происшествий, пожарной охраны, медицинской помощи, в сфере жилищно-коммунального хозяйства, оказывающих влияние на гражданскую безопасность на территории муниципальных образований.

Программное обеспечение ИСППР может эксплуатироваться на объектах любого масштаба и предназначено для организаций, входящих в территориальную подсистему РСЧС муниципального уровня и обеспечивающих деятельность служб (организаций) в области защиты населения и территорий от чрезвычайных ситуаций, управления силами и средствами, предназначенными и привлекаемыми для предупреждения и ликвидации чрезвычайных ситуаций (происшествий).

СОДЕРЖАНИЕ

1. Загрузка и запуск программы.....	4
1.1 Подготовка к запуску	4
1.2 Загрузка и запуск ПО ИСППР	4
1.2.1 Загрузка общесистемного ПО	4
1.2.2 Скачивание и распаковка архива.....	4
1.2.3 Установка ПО из архива.	5
1.2.4 Загрузка общедоступных образов	5
1.2.5 Запуск ПО.....	5
1.2.6 Контроль и мониторинг запуска ПО	5
2. Конфигурация	7
2.1 Настройка параметров запуска контейнеров.....	7
2.2 Настройка конфигурационного файла	8
3. ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ	14

1. ЗАГРУЗКА И ЗАПУСК ПРОГРАММЫ

1.1 Подготовка к запуску

Данный раздел описывает развертывание приложения со всеми зависимостями в Docker-контейнерах. Запуск всех контейнеров в нужной конфигурации происходит согласно файлу конфигурации `docker-compose.yml`. Для подготовки запуска приложения необходимо выполнить следующие шаги:

- обновить серверную операционную систему до последней версии;
- установить программное обеспечение Docker, согласно инструкции, расположенной по адресу: <https://docs.docker.com/engine/install/>;
- установить программное обеспечение Docker Compose, согласно инструкции, расположенной по адресу: <https://docs.docker.com/compose/install/>.

1.2 Загрузка и запуск ПО ИСППР

Данный раздел описывает загрузку дистрибутива и запуск ПО ИСППР, которые включают в себя следующие шаги, указанные ниже по тексту настоящего документа.

1.2.1 Загрузка общесистемного ПО

Для работы ПО ИСППР необходимо установить и настроить общесистемное ПО PostgreSQL, MinIO, Redis. Они не входят в поставку и могут быть получены из общедоступного хранилища образов. Однако для удобства часть настроенных и протестированных образов доступны на сайте. Скачать и установить их можно командой:

```
wget -qO- https://www.vanguardsoft.ru/redgift/redgift-minio.img.gz | docker load
```

Оставшиеся образы будут скачены автоматически при первом запуске, см. ниже.

1.2.2 Скачивание и распаковка архива

1. Создайте выделенную папку для приложения, например `/opt/isprr`, с помощью следующих команд:

```
mkdir /opt/isprr
```

```
cd /opt/isppr
```

2. Загрузите в эту папку архив `isppr.tar.gz` по ссылке, полученной от разработчика ПО.

3. Распакуйте скачанный архив с помощью команды:

```
tar xvzf isppr.tar.gz
```

1.2.3 Установка ПО из архива.

1. После распаковки архива перейдите в директорию с файлами образов с помощью команды:

```
cd /opt/isppr
```

2. Заполните конфигурационный файл `.env` как указано в разделах ниже.

3. При необходимости произведите настройку файла `settings.toml`.

4. Запустите команду установки и дождитесь окончания.

```
for archive in deploy/*.img.gz; do docker load -i ${archive}; done
```

5. Загруженные образы теперь можно удалить с диска с помощью команды:

```
rm deploy/*.img.gz
```

6. При необходимости скопируйте имеющиеся датафреймы в папку `storage/datasets` и загрузите подготовленные модели.

1.2.4 Загрузка общедоступных образов

Загрузка общедоступных образов, не входящих в поставку, осуществляется командой:

```
docker-compose pull
```

1.2.5 Запуск ПО

1. Запустите Docker Compose для развертывания всех компонентов с помощью следующей команды:

```
docker-compose up -d
```

1.2.6 Контроль и мониторинг запуска ПО

1. Убедитесь, что все контейнеры запущены и работают корректно, выполнив

команду:

docker-compose ps

Результат команды должен отражать список запущенных контейнеров (рисунок 1).

```

root@isppr:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
4e9a102eb201  registry.kiap.dev/redgift/isppr-mlflow  "/bin/sh -c 'mlflow -"  5 days ago    Up 5 days (healthy)  0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  isppr-mlflow-1
42d13ef07d07  registry.kiap.dev/redgift/isppr-airflow  "/usr/bin/dumb-init -"  5 days ago    Up 5 days (healthy)  8080/tcp                                isppr-airflow-triggerer-1
15a54ceba7ca  registry.kiap.dev/redgift/isppr-airflow  "/usr/bin/dumb-init -"  5 days ago    Up 5 days (healthy)  8080/tcp                                isppr-airflow-scheduler-1
4e8a3490b594  registry.kiap.dev/redgift/isppr-airflow  "/usr/bin/dumb-init -"  5 days ago    Up 5 days (healthy)  8080/tcp                                isppr-airflow-worker-1
fa44294da29c  registry.kiap.dev/redgift/minio:RELEASE.2024-11-07T00-52-20Z  "/usr/bin/docker-ent-"  5 days ago    Up 5 days (healthy)  0.0.0.0:9000-9001->9000-9001/tcp, :::9000-9001->9000-9001/tcp  isppr-minio-1
48dd339f698c  registry.kiap.dev/redgift/isppr-airflow  "/usr/bin/dumb-init -"  5 days ago    Up 5 days (healthy)  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  isppr-airflow-webserver-1
29e336c4c97e  registry.kiap.dev/redgift/isppr-frontend  "/bin/sh -c 'python -"  5 days ago    Up 5 days (healthy)  0.0.0.0:8282->8000/tcp, :::8282->8000/tcp  isppr-frontend-1
55f8772308c5  redis:7.2.0                              "docker-entrypoint.sh"  4 months ago  Up 5 days (healthy)  6379/tcp                                isppr-redis-1
f6eb349752d8  postgres:14.12                          "docker-entrypoint.sh"  4 months ago  Up 5 days (healthy)  0.0.0.0:5432->5432/tcp, :::5432->5432/tcp  isppr-postgres-1

```

Рисунок 1 - Результат команды мониторинга

2. Для мониторинга логов каждого из контейнеров можно выполнить команду:

docker-compose logs -f

3. При необходимости остановки контейнеров, используйте команду:

docker-compose down

4. При необходимости перезапуска контейнеров, используйте команду:

docker-compose restart

2. КОНФИГУРАЦИЯ

Данный раздел описывает предварительную конфигурацию файлов для корректного запуска приложения. Настройка приложения осуществляется с помощью env-параметров запуска контейнеров (основные параметры) и/или с помощью конфигурационного файла (основные параметры и тонкие настройки). Приоритет имеют env-параметры.

2.1 Настройка параметров запуска контейнеров

Параметры запуска контейнеров задаются в файле .env. В нем располагаются основные параметры микросервисов, такие как параметры доступа к базам данных, пароли доступа к системе, сетевые порты для доступа к сервисам.

Пример .env файла приведен ниже:

```
# Логин и пароль доступа к веб-интерфейсу Airflow
AIRFLOW_WWW_USER_USERNAME='login'
AIRFLOW_WWW_USER_PASSWORD='password'

# Логин и пароль доступа к моделям
MINIO_ACCESS_KEY='mlflow_login'
MINIO_SECRET_KEY='mlflow_password'

# Параметры базы данных метаданных и прогноза
DB_NAME='airflow'
DB_USER='airflow'
DB_PASSWORD='password'
MLFLOW_DB_NAME='mlflow'

# Параметры доступа к базе-источнику происшествий
SRC_HOSTNAME='HOSTNAME'
SRC_DATABASENAME='DATABASENAME'
SRC_USERNAME='USERNAME'
```

```
SRC_PASSWORD='PASSWORD'

# Параметры доступа к базе-источнику обращений
PG_HOSTNAME='HOSTNAME'
PG_DATABASENAME='DATABASENAME'
PG_PORTNUMBER='PORTNUMBER'
PG_USERNAME='USERNAME'
PG_PASSWORD='PASSWORD'

# Сетевые порты для доступа к микросервисам
FRONTEND_PORT=8282
AIRFLOW_PORT=8080
MINIO_PORT=9000
MINIO_CONSOLE_PORT=9001
POSTGRES_PORT=5432
MLFLOW_PORT=5000

# Не менять! Параметры запуска Airflow в контейнере
AIRFLOW_UID=50000
AIRFLOW_HOME='/opt/airflow/'
```

2.2 Настройка конфигурационного файла

Конфигурационный файл является компонентом хранения статических данных конфигурации микросервисов, в нем располагаются более тонкие настройки процесса загрузки, такие как наименования датафреймов, адресный путь к данным о погоде и прочие служебные данные. Конфигурационный файл `settings.toml` находится в папке `dags` контейнера `airflow` и может быть заменен через механизм монтирования `volumes` в `docker compose`.

Пример файла `settings.toml` приведен ниже.

```
#Общие настройки
#папка хранения датасетов
DATASET_DIR = "/tf/datasets"
#имя папки airflow (может быть пустым "" если датасеты в DATASET_DIR)
AIRFLOW_DIRECTORY_NAME = "/airflow/"
#папка где хранятся функции
EXT_DIR = "/tf/ext/"
#IP адрес сервера
SERVER_IP = "ip"
#порт веб интерфейса MLflow
MLFLOW_PORT = 5000
#путь к веб интерфейсу MLflow
MLFLOW_URI = "@format http://{this.SERVER_IP}:{this.MLFLOW_PORT}"
#ключ доступа к MLflow (логин и пароль)
AWS_ACCESS_KEY_ID = "login"
AWS_SECRET_ACCESS_KEY = "password"

#адрес веб интерфейса minio
MLFLOW_S3_ENDPOINT_URL = "http://minio:9000/"

#данные подключения к БД хранения прогнозов
PREDICTIONS_DATABASE = "databasename"
PREDICTIONS_USER = "user"
PREDICTIONS_PASSWORD = 'password'
PREDICTIONS_HOST = "ip базы данных"
PREDICTIONS_PORT = "5432"

#константы для прогнозирования кол-ва операторов
OPERATORS_DEPLOY_FILENAME = "dataframe_operators_full.csv"
OPERATORS_DATASET = "dataframe_operators.csv"
OPERATORS_TARGET_COLUMN = "incidents_count"
```

```

OPERATORS_EXPREIMENT_NAME = "operators_forecast"
OPERATORS_DIMENSIONS = 1
OPERATORS_EPOCHS = 3000
OPERATORS_BATCH_SIZE = 10
OPERATORS_FILTERS_VALUE = 64
OPERATORS_N_STEPS = 2
OPERATORS_LAG = 15
OPERATORS_TEST_SIZE = 0.25
OPERATORS_PREDICTIONS_COLUMNS = "predictions_accidents target_value
accident_name aggregation_method predictions_date"
OPERATORS_PREDICTIONS_TABLE = "predictions_operators_forecast"
OPERATORS_DATAFRAME_PATH = "@format
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.OPERATORS_DA
TASET}"
OPERATORS_DEPLOY_DATAFRAME_PATH = "@format
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.OPERATORS_DE
PLOY_FILENAME}"

#константы для прогнозирования кол-ва инцидентов
ACCIDENTS_FORECAST_CNT_01_ID = 1080
ACCIDENTS_FORECAST_CNT_02_ID = 1081
ACCIDENTS_FORECAST_CNT_03_ID = 1082
ACCIDENTS_FORECAST_CNT_JKH_ID = 1083
ACCIDENTS_FORECAST_CNT_DTP_ID = 1161
ACCIDENTS_FORECAST_DEPLOY_DATAFRAME =
"accidents_forecast_full.csv"
ACCIDENTS_FORECAST_DATAFRAME_NAME =
"dataframe_accidents_forecast.csv"
ACCIDENTS_FORECAST_ACC_COLUMNS = "cnt_01 cnt_02 cnt_03 cnt_jkh
cnt_dtp"

```

```

ACCIDENTS_FORECAST_EXPERIMENT_NAME = "accidents_forecast"
ACCIDENTS_FORECAST_MERRIDIAN_DATE = "2022-10-17"
ACCIDENTS_FORECAST_DATA_TEST_END_DATE = "2023-10-31"
ACCIDENTS_FORECAST_APPROXIMATION = 15
ACCIDENTS_FORECAST_DIMENSIONS = 1
ACCIDENTS_FORECAST_EPOCHS = 3000
ACCIDENTS_FORECAST_BATCH_SIZE = 10
ACCIDENTS_FORECAST_FILTERS_VALUE = 64
ACCIDENTS_FORECAST_N_STEPS = 2
ACCIDENTS_FORECAST_LAG = 15
ACCIDENTS_FORECAST_TEST_SIZE = 0.25
ACCIDENTS_FORECAST_PREDICTIONS_TABLE =
"predictions_accidents_forecast"
ACCIDENTS_FORECAST_PREDICTIONS_COLUMNS = "predictions
target_value accident_name aggregation_method predictions_date"
ACCIDENTS_FORECAST_DATAFRAME_PATH = "@format
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.ACCIDENTS_FORECAST_DATAFRAME_NAME}"
ACCIDENTS_FORECAST_DEPLOY_DATAFRAME_PATH = "@format
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.ACCIDENTS_FORECAST_DEPLOY_DATAFRAME}"

#константы аномалий звонков
CALLS_ANOMALY_DATAFRAME_NAME = "dataframe_calls_anomaly.csv"
CALLS_ANOMALY_EXPERIMENT_NAME = "calls_anomaly"
CALLS_ANOMALY_DDS_KRASNODAR = "ЦОВ ДДС-01 ДДС-02 ДДС-03
ДДС-04 ЕДДС"
CALLS_ANOMALY_DDS_SOCHI = "ЕДДС-Сочи ДДС-01-Сочи ДДС-02-Сочи
ДДС-03-Сочи ДДС-04-Сочи"
CALLS_ANOMALY_CONTAMINATION = 0.25

```

```
CALLS_ANOMALY_PREDICTIONS_TABLE = "predictions_calls_anomaly"  
CALLS_ANOMALY_DATAFRAME_PATH = "@format  
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.CALLS_ANOMA  
LY_DATAFRAME_NAME}"
```

```
#константы кластеризации
```

```
INCIDENT_CLUSTERING_DATAFRAME_KRASNODAR =  
"dataframe_incident_clustering_krasnodar.csv"
```

```
INCIDENT_CLUSTERING_DATAFRAME_SOCHI =  
"dataframe_incident_clustering_sochi.csv"
```

```
INCIDENT_CLUSTERING_EXPREIMENT_NAME = "incident_clustering"
```

```
INCIDENT_CLUSTERING_PREDICTIONS_TABLE =  
"predictions_incident_clustering"
```

```
INCIDENT_CLUSTERING_DATAFRAME_COLUMNS = "ic  
emergency_category_id region_id emergency_category date_received"
```

```
INCIDENT_CLUSTERING_DATAFRAME_KRASNODAR_PATH = "@format  
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.INCIDENT_CLUS  
TERING_DATAFRAME_KRASNODAR}"
```

```
INCIDENT_CLUSTERING_DATAFRAME_SOCHI_PATH = "@format  
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.INCIDENT_CLUS  
TERING_DATAFRAME_SOCHI}"
```

```
INCIDENT_CLUSTERING_KRASNODAR_COORDINATES = "39.0912  
45.0264"
```

```
INCIDENT_CLUSTERING_SOCHI_COORDINATES = "39.6209 43.6667"
```

```
INCIDENT_CLUSTERING_CLUSTER_COUNT = 9
```

```
#константы аномалий обработки заявок операторами
```

```
STATUSES_ANOMALY_DATAFRAME = "dataframe_statuses_anomaly.csv"
```

```
STATUSES_ANOMALY_EXPRIMENT_NAME = "statuses_anomaly"
```

```
STATUSES_ANOMALY_CITYES = "Сочи Краснодар"
```

```
STATUSES_ANOMALY_CONTAMINATION = 0.15
```

```
STATUSES_ANOMALY_PREDICTIONS_TABLE =  
"predictions_statuses_anomaly"  
STATUSES_ANOMALY_DATAFRAME_PATH = "@format  
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.STATUSES_ANO  
MALY_DATAFRAME}"  
  
#константы погодного датасета  
WEATHER_DATAFRAME = "dataframe_weather.csv"  
WEATHER_LINK = "ссылка на сервис с которого получаются погодные  
данные"  
WEATHER_DATAFRAME_PATH = "@format  
{this.DATASET_DIR}{this.AIRFLOW_DIRECTORY_NAME}{this.WEATHER_DAT  
AFRAME}"
```

3. ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОПРЕДЕЛЕНИЙ

ДДС	–	Дежурно-диспетчерская служба
ИСППР	–	Интеллектуальная система поддержки принятия решений для дежурно-диспетчерских служб
ПО	–	Программное обеспечение
СУБД	–	Система управления базами данных
БД	–	База данных
ДДС-01	–	Дежурно-диспетчерская служба пожарной охраны
ДДС-02	–	Дежурно-диспетчерская служба полиции
ДДС-03	–	Дежурно-диспетчерская служба скорой медицинской помощи
ДДС-04	–	Дежурно-диспетчерская служба газа
ЦОВ-112	–	Центр обработки вызовов Системы-112
ЕДДС	–	Единая дежурно-диспетчерская служба
РСЧС	–	Единая государственная система предупреждения и ликвидации чрезвычайных ситуаций