ВИЛИОН

ПРОГРАММНЫЙ ПРОДУКТ

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММЫ ДЛЯ ЭВМ «ПРЕДИКТИВНОЕ ОПОВЕЩЕНИЕ О ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЯХ И ИХ ПОСЛЕДСТВИЯХ»

Краснодар

2022

Оглавление

1. Назначение

Настоящий документ содержит инструкцию по установке демонстрационного макета программного обеспечения «Предиктивное оповещение о чрезвычайных ситуациях и их последствиях» и описывает развертывание ПО.

Данный макет допускает промышленное использование при отказе от загрузки тестовых данных во время процедуры развертывания и последующей конфигурации и подключения к реальным источники данных.

2. Системные Требования

Дистрибутив собран и протестирован в следующей конфигурации окружения:

- Ubuntu Linux версии 22.04 или совместимый дистрибутив
- Docker версии 20.10 и выше
- Docker Compose версии v2.6 и выше

на аппаратной платформе

- CPU от 4х ядер
- MEM от 4Гб
- HDD/SDD не менее 10Гб

3. Установка

Для упрощения процедуры развертывания, дистрибуция ПО «Предиктивное оповещение о чрезвычайных ситуациях и их последствиях» осуществляется в виде контейнера Docker с преднастроенной сборкой docker-compose, включающий в себя брокер сообщений kafka.

Порядок установки:

1. Скачайть и распаковать архив дистрибутива
   tar xfz py-predict-docker.tgz
2. Загрузить образ контейнера:
   docker load < py-predict-latest.img
3. Произвести запуск контейнеров
   docker-compose up -d

4. Произвести первоначальную загрузку тестовых данных
   docker-compose exec -e PP_LOGGING_LEVEL=DEBUG pypredict-api pypredict-loader-init
5. Проконтролировать работу контейнеров
   docker-compose logs pypredict-api pypredict-router pypredict-predictor pypredict-criticallevel pypredict-colorlimit

4. Конфигурация и настройка

Конфигурирование контейнера осуществляется путем переопределения конфигурации **conf.yml** с помощью env-переменных. Для этого к параметрам из **conf.yml** добавляется префикс “**PP_**” (без кавычек).

Второй вариант переопределения конфигурации — это использование механизма volume docker-compose с полным переорпеделением файла **conf.yml**.

Настройки, необходимые для подключения внутри контейнера хранятся в файле **.env**.

5. Управление контейнерами

После запуска доступны следующие контейнеры py-predict:

- **pypredict-api** — REST API подключения датчиков и получения результатов, доступен по адресу http://localhost:6065
- **pypredict-router** — Модуль сортировки
- **pypredict-colorlimit** — Модуль классификации уровня опасности
- **pypredict-criticallevel** — Модуль определения зоны подтопления
- **pypredict-predictor** — Диспетчер модулей прогноза

А так же следующие контейнеры брокера kafka:

- **cp-zookeeper** — балансировщик брокера, доступен по адресу localhost:2181
- **broker** — непосредственно брокер сообщений kafka, доступен по адресу localhost:9092
- **schema-registry** — каталог схем данных брокера, доступен по адресу localhost:8081
- **control-center** — центр управления брокером, доступен по адресу http://localhost:9021

Для перезагрузки контейнера используется команда:

docker-compose restart *<имя контейнера>*

Остановка с полная очисткой данных контейнера:

docker-compose down *<имя контейнера>*

Просмотр логов контейнера:

docker-compose logs *<имя контейнера>*

6. Работа с конфигурационным файлом

Формат конфигурационного файла **conf.yml** с комментариями и примером заполнения:

```
# Порт запуска REST сервисов
    REST_SERVICE_PORT: 6066

# Параметры подключения к Kafka и Schema Registry
    KAFKA_CONSUMER_GROUP: 'pyPredictRouter-producer'
    KAFKA_BOOTSTRAP_SERVERS: 'localhost:9092'
    SCHEMA_REGISTRY_URL: 'http://localhost:8081'

# Входной топпик для датчиков/данных
    SENSORS_TOPIC: 'sensors'

# Роутер входных данных - сортировка источников по разным очередям
    GRAINS_ROUTER: {
      'river:a1': 'route-kuban',
      'river:a2': 'route-belaya'
    }

# Привязка алгоритмов к разным очередям
    GRAINS_PREDICTOR: {
      'route-kuban': { CLASS: 'PredictorKuban', OUT_TOPIC: 'predicted-kuban' },
      'route-belaya': { CLASS: 'PredictorBelaya', OUT_TOPIC: 'predicted-belaya' },
```

```
}


# Уровни опасности для рек
COLOR_IN_TOPICS: ['predicted-kuban', 'predicted-belaya']
COLOR_LEVELS: {
 'river:a1': { 15: "Зеленый", 20: "Желтый", 25: "Красный" },
 'river:a2': { 15: "Зеленый", 20: "Желтый", 25: "Красный" },
}
COLOR_OUT_TOPIC: 'warn-levels'
```

Определение зоны затопления для рек

```
CRITICAL_IN_TOPICS: ['predicted-kuban', 'predicted-belaya']
CRITICAL_LEVELS: {
 'river:a1': {
   24: [ 'с.Калинино','мрн.Ньювасюки' ],
   26: [ 'Дальнее Бутово','х.Ленина','пос.Грушевка' ],
   28: [ 'аул Веселое' ],
 },
 'river:a2': {
   19: [ 'снт.Люблино' ],
   22: [ 'Гадюкино', 'х.Светлое' ],
   30: [ 'с.Грустное' ],
 },
}
CRITICAL_OUT_TOPIC: 'warn-levels'
```

Дополнительные настройки отображения имен датчиков

```
FAMILY_NAMES: {
```

```
  'river:a1': 'Кубань (Усть-Лабинс)',
  'river:a2': 'Белая (Майкоп)'
 }

 # Настройки логгирования
 LOGGING_LEVEL: 'INFO'
 LOGGING_FORMAT: '%(asctime)-15s | %(levelname)s |
%(filename)s#%(lineno)d | %(message)s'
```

7. Перечень сокращений

ОС — Операционная система

ПО — Программное обеспечение