FLOWX.AI

PLATFORM SETUP GUIDES / scheduler-setup-guide

Contents

- PLATFORM SETUP GUIDES / Scheduler setup guide
  - Introduction
  - Infrastructure prerequisites
  - Dependencies
  - Dependencies
    - MongoDB helm example
  - Configuration
    - Configuring MongoDB
    - Configuring Kafka
    - Configuring logging

PLATFORM SETUP GUIDES / Scheduler setup guide

Introduction

This guide will walk you through the process of setting up the Scheduler service using a Docker image.

Infrastructure prerequisites

- **MongoDB** - version 4.4 or higher for storing taxonomies and contents
- **Kafka** - version 2.8 or higher

Dependencies

- MongoDB database
- ability to connect to a Kafka instance used by the engine

The service comes with most of the needed configuration properties filled in, but there are a few that need to be set up using some custom environment variables.

Dependencies

MongoDB helm example

Basic MongoDB configuration - helm values.yaml

```
scheduler-mdb:
    existingSecret: {{secretName}}
    mongodbDatabase: {{SchedulerDatabaseName}}
    mongodbUsername: {{SchedulerDatabaseUser}}
    persistence:
      enabled: true
      mountPath: /bitnami/mongodb
      size: 4Gi
    replicaSet:
      enabled: true
      name: rs0
      pdb:
        enabled: true
        minAvailable:
          arbiter: 1
          secondary: 1
      replicas:
        arbiter: 1
        secondary: 1
```

```
    useHostnames: true
  serviceAccount:
    create: false
  usePassword: true
```

DANGER

This service needs to connect to a Mongo database that has replicas, in order to work correctly.

Configuration

Configuring MongoDB

The MongoDB database is used to persist scheduled messages until they are sent back. The following configurations need to be set using environment variables:

- `SPRING_DATA_MONGODB_URI` - the URI for the MongoDB database

Configuring Kafka

The following Kafka related configurations can be set by using environment variables:

- `SPRING_KAFKA_BOOTSTRAP_SERVERS` - address of the Kafka server
- `SPRING_KAFKA_CONSUMER_GROUP_ID` - group of consumers
- `KAFKA_CONSUMER_THREADS` - the number of Kafka consumer threads

- `KAFKA_AUTH_EXCEPTION_RETRY_INTERVAL` - the interval between retries after `AuthorizationException` is thrown by `KafkaConsumer`

Each action available in the service corresponds to a Kafka event. A separate Kafka topic must be configured for each use-case.

> **CAUTION**
>
> Make sure the topics configured for this service don't follow the engine pattern.

Configuring logging

The following environment variables could be set in order to control log levels:

- `LOGGING_LEVEL_ROOT` - root spring boot microservice logs
- `LOGGING_LEVEL_APP` - app level logs

Was this page helpful?