FLOWX.AI

**PLATFORM SETUP GUIDES / search-data-service-setup-guide**

# Contents



# PLATFORM SETUP GUIDES / Data search service setup guide

## Introduction

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

## Infrastructure prerequisites

Before proceeding with the setup, ensure that the following components have been set up:

- **Redis** - version 6.0 or higher

- **Kafka** - version 2.8 or higher
- **Elasticsearch** - version 7.11.0 or higher

# Dependencies

- **Kafka** - used for communication with the engine
- **Elasticsearch** - used for indexing and searching data
- **Redis** - used for caching

# Configuration

## Configuring Kafka

Set the following Kafka-related configurations using environment variables:

- `SPRING_KAFKA_BOOTSTRAP_SERVERS` - address of the Kafka server
- `KAFKA_TOPIC_DATA_SEARCH_IN`
- `KAFKA_TOPIC_DATA_SEARCH_OUT`
- `KAFKA_CONSUMER_THREADS` - the number of Kafka consumer threads

## Configuring Elasticsearch

Set the following Elasticsearch-related configurations using environment variables:

- `SPRING_ELASTICSEARCH_REST_URIS`

- `SPRING_ELASTICSEARCH_REST_DISABLESSL`
- `SPRING_ELASTICSEARCH_REST_USERNAME`
- `SPRING_ELASTICSEARCH_REST_PASSWORD`
- `SPRING_ELASTICSEARCH_INDEX_SETTINGS_NAME` - the index can be customized for data-search and it should be similar to what is configured on the process-engine

## Configuring authorization & access roles

Set the following environment variables to connect to the identity management platform:

- `SECURITY_OAUTH2_BASE_SERVER_URL`
- `SECURITY_OAUTH2_CLIENT_CLIENT_ID`
- `SECURITY_OAUTH2_REALM`

## Configuring logging

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

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

## Elasticsearch

Data search in Elasticsearch runs against an index pattern representing multiple indices. The index pattern is derived from the configuration property:

`spring.elasticsearch.index-settings.name`

Below is an example of a filter to be used in Kibana (as generated by data search):

```
{
  "query": {
    "bool": {
      "adjust_pure_negative": true,
      "boost": 1,
      "must": [
        {
          "nested": {
            "boost": 1,
            "ignore_unmapped": false,
            "path": "keyIdentifiers",
            "query": {
              "bool": {
                "adjust_pure_negative": true,
                "boost": 1,
                "must": [
                  {
                    "match": {
                      "keyIdentifiers.key.keyword": {

"auto_generate_synonyms_phrase_query": true,
                        "boost": 1,
                        "fuzzy_transpositions": true,
                        "lenient": false,
                        "max_expansions": 50,
                        "operator": "OR",
                        "prefix_length": 0,
```

```
                        "query": "astonishingAttribute",
                        "zero_terms_query": "NONE"
                      }
                    }
                  },
                  {
                    "match": {

"keyIdentifiers.originalValue.keyword": {

"auto_generate_synonyms_phrase_query": true,
                        "boost": 1,
                        "fuzzy_transpositions": true,
                        "lenient": false,
                        "max_expansions": 50,
                        "operator": "OR",
                        "prefix_length": 0,
                        "query": "OriginalGangsta",
                        "zero_terms_query": "NONE"
                      }
                    }
                  }
                ]
              }
            },
            "score_mode": "none"
          }
        },
        {
          "terms": {
            "boost": 1,
            "processDefinitionName.keyword": [
              "TEST_PORCESS_NAME_0",
              "TEST_PORCESS_NAME_1"
            ]
```

```
            }
          }
        ]
      }
    }
}
```

**Was this page helpful?**