

# Spark Wordcloud

Team [Miracle Fruit](#)

Webapp with Docker, Spark, MariaDB and Flask



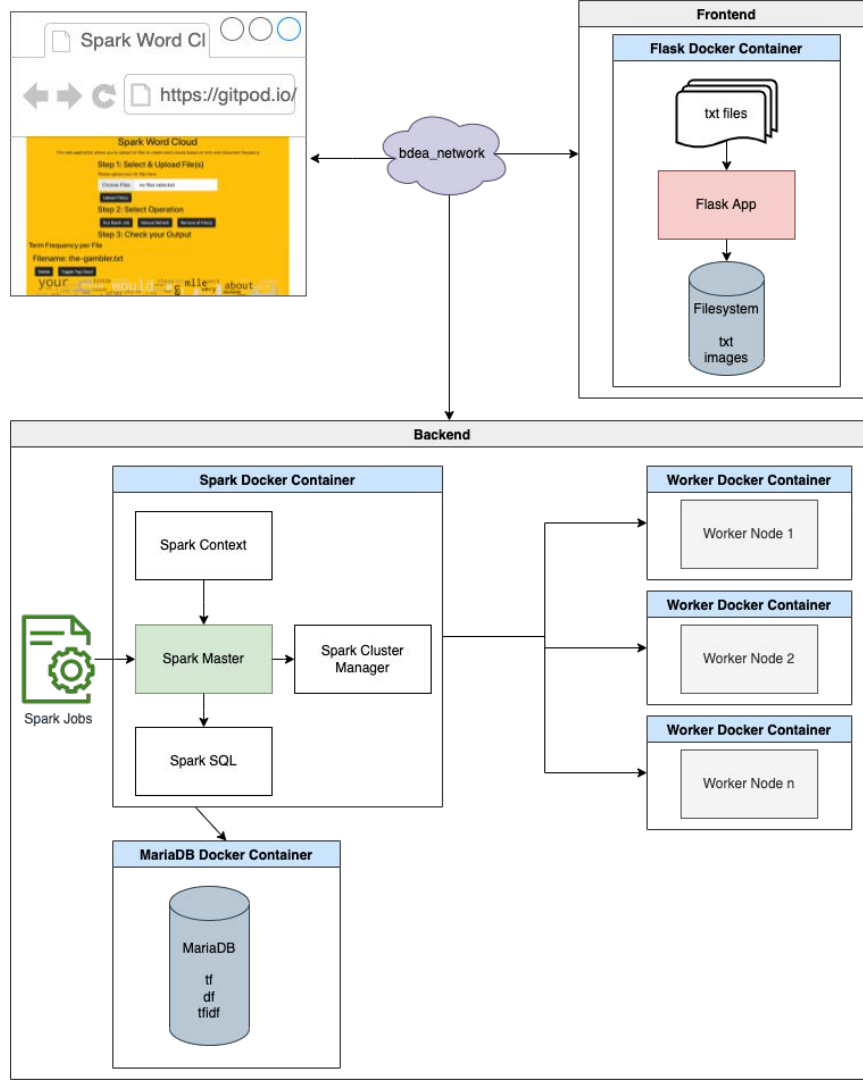
# Architecture

**tf\_job** which is executed every time a new text file is uploaded on the website

**df\_job** which is the batch job that can be manually executed to generate the global term frequency word cloud based on the document frequency

## Database:

- **tf** (stores the term frequency for all documents)
- **df** (stores the document frequency for all words)
- **tfidf** (stores the TFIDF value for each document)



# Live Demo

# Spark Word Cloud

This web application allows you to upload txt files to create word clouds based on term and document frequency

## Step 1: Select & Upload File(s)

Please upload your txt files here:

Choose Files

no files selected

**Upload File(s)**

## Step 2: Select Operation

## Run Batch Job

Manual Refresh

Remove all File(s)

## Step 3: Check your Output

## Term Frequency per File

Filename: the-gambler.txt



Delete

### Toggle Tag Cloud



<https://github.com/Miracle-Fruit/spark-wordcloud>

# Problems, Lessons Learned & Discussion

- Docker setup with all components 🖱️ try and error until it works
- Setup with database and JDBC Driver 🖱️ Link driver by spark-submit and in docker-image
- Read and write in the same spark job 🖱️ Cache the table 
- Special characters are not correctly saved in database 🖱️ Change *char\_set* and collate to UTF-8
- Bootstrap elements center 🖱️ Frontend was annoying 
- Spark Jobs depends strongly on systems hardware e. g. 4 MB .txt-file PC 11-20 Sec., Laptop 🖱️ 2-3 Min. 