Code

Table of Contents

If you encounter any error, feel free to enter an issue on GitHub.

TODO

1. kafkatrain

See on GitHub

1.1. Avoir un train à l'heure, c'est kafkaïen

> Repository principal de notre présentation à Snowcamp 2019

1.1.1. What does this repository contains ?

- `src/build` contains various build scripts
  - `0-install.sh` installs the environment, provided the secrets are known
  - `1-write-reader-code.bat` copies reader code in its own repository
  - `2-write-web-ui.bat` copies web ui in its own repository
  - `delete.bat` deletes the cluster, and the various generated projects
- `src/k8s` contains all deployed into k8s cluster
  - `elastic` provides ingresses for Kibana and Elasticsearch (**DON'T DO THAT IN PROD**)
  - `kafka` installs all additionnal applications for kafka

1.1.2. Meta

- Logan Hauspie – @lhauspie
- Nicolas Delsaux – @Riduidel

1.1.3. Contributing

1. Fork it (<https://github.com/Riduidel/snowcamp-2019/fork>)
2. Create your feature branch (`git checkout -b feature/fooBar`)

3. Commit your changes (`git commit -am 'Add some fooBar'`)
4. Push to the branch (`git push origin feature/fooBar`)
5. Create a new Pull Request

1.2. sncf-reader

See on GitHub

1.2.1. sncf-reader application

This application allows us to inject SNCF timesheets into our Kafka engine, for later processing.

Configuration

This application requires the following environment variables to be set

- `SNCF_READER_TOKEN` access token for Navitia API
- `SNCF_READER_READ_AT_STARTUP` When set to true, immediatly start reading SNCF timesheet
- `SNCF_READER_KAFKA_BOOTSTRAP_SERVER` url of Kafka server to connect to
- `SNCF_READER_TOPIC_SCHEDULE` Topic where to post schedule. Defaults to `sncfReaderSchedule`

1.3. web-ui

See on GitHub

1.3.1. node

Simple Hello World that listens on localhost:8080