Prerequisites

In order to run BDE platform your system should fulfill these requirements:

- `docker` ≥ 1.13.0 [instructions]
- `docker-compose` ≥ 1.10.0 [instructions]

Please note that you will need apx 10 -15 GB free disk space and around 10 GB memory.

Installing the BDI Platform

The Big Data Integrator is an application that can be thought of as a "starter kit" to start working and implementing big data pipelines in your process. It is the minimal standalone system so you can create a project with multiple docker containers, upload it & make it run using a nice GUI.

You can find the BDI platform here.

1. Clone the repository: git clone https://github.com/big-data-europe/app-bdi-ide.git
2. Move into the newly created directory: cd app-bdi-ide
3. Run the install script (which will configure the docker-compose to your environment and modify your hosts file): sudo ./scripts/install.sh
4. Start the platform: docker-compose up

Your BDI platform should now be reachable here.

Running your stack

Now that our BDI platform is running, we can clone and execute a big data pipeline on this platform.

Creating the stack

Browse to http://integrator-ui.big-data-europe.aksw.org/ and click "swarm-ui" in the headers. You can now create a new stack by pressing the “create new stack” button and by providing a name, an optional icon and a docker-compose file from either a git repository or already stored in the database.

For this presentation, we will use an existing stack from https://github.com/GezimSejdiu/pilot-sc4-fcd-applications.git

Once the stack is running on the BDE platform,(Please note that It will take sometime to load all the docker images, when you try it for the first time) the pipeline could be monitored by pipeline monitor applications.
This monitor will show the status of individual running containers.
As an application-pipeline running on the platform, we can scale our services “up”, or “down”, as needed.
Once we see that all the services are running, we move on the interface of the application.

Application

The application fetches continuously the FCD (Floating Car Data) data about taxies in Thessaloniki from CERTH web service. A Kafka Producer connects to the web service and sends the data into a Kafka topic. The data is consumed by a Flink job, enriched using a map matching algorithm, aggregated and finally stored into Elasticsearch. A visualization based on Kibana is used to visualize the aggregated data in a map (average speed in road segments per time windows).

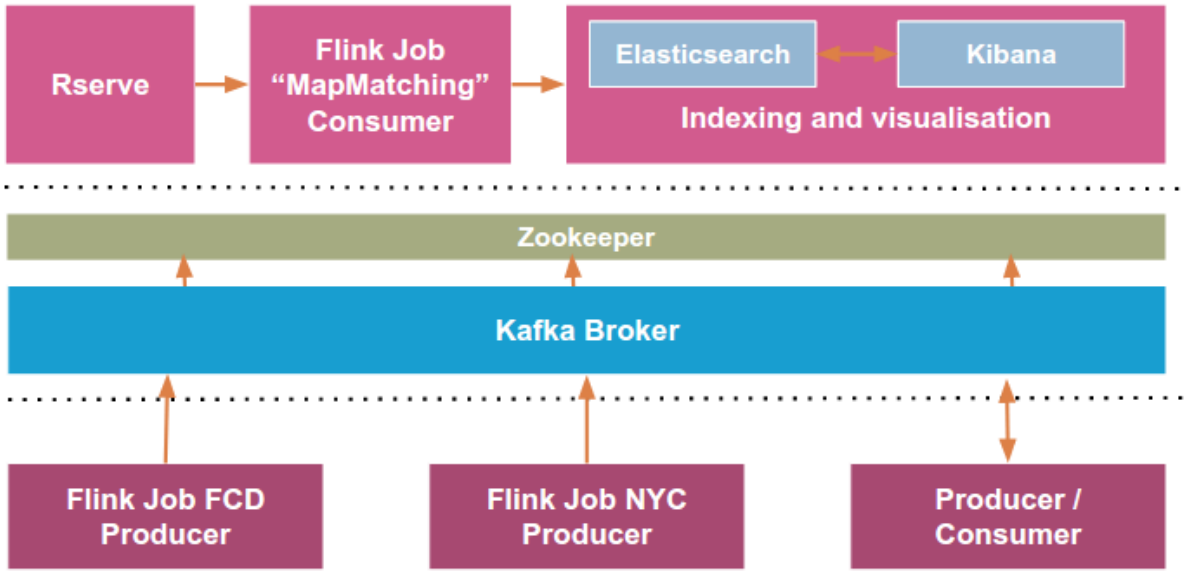

Fig.1 Application architecture.

The architecture has been designed in order to handle many data sources, mostly stream data, process it in time windows and store the result so that critical features such as geographic information and time can be indexed enabling low latency for search and queries.

The workflow of the pipeline is as follows:

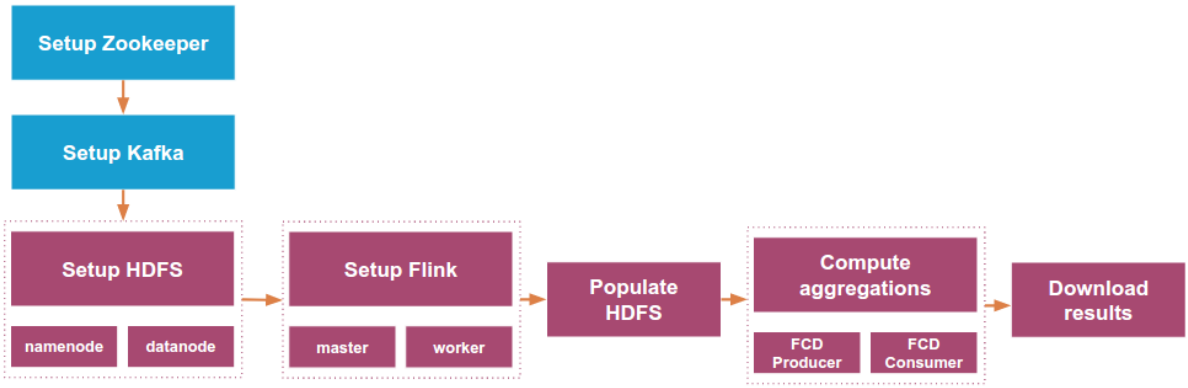

Fig.2 Pipeline workflow.

The web-interface of the pipeline will be accessible at localhost:http://localhost:9999/