DHBW Mannheim

Big Data Analytics

Kafka und Projekt

Frank Schulz

Installation

Download Apache Kafka

https://kafka.apache.org/quickstart

Alternatively:

Download Confluent Platform

https://www.confluent.io/download/

https://docs.confluent.io/current/quickstart/index.html

TL;DR

```
git clone https://github.com/confluentinc/cp-all-in-one.git
cd cp-all-in-one/cp-all-in-one
docker-compose up
```

Confluent Platform on Docker

Check Docker containers

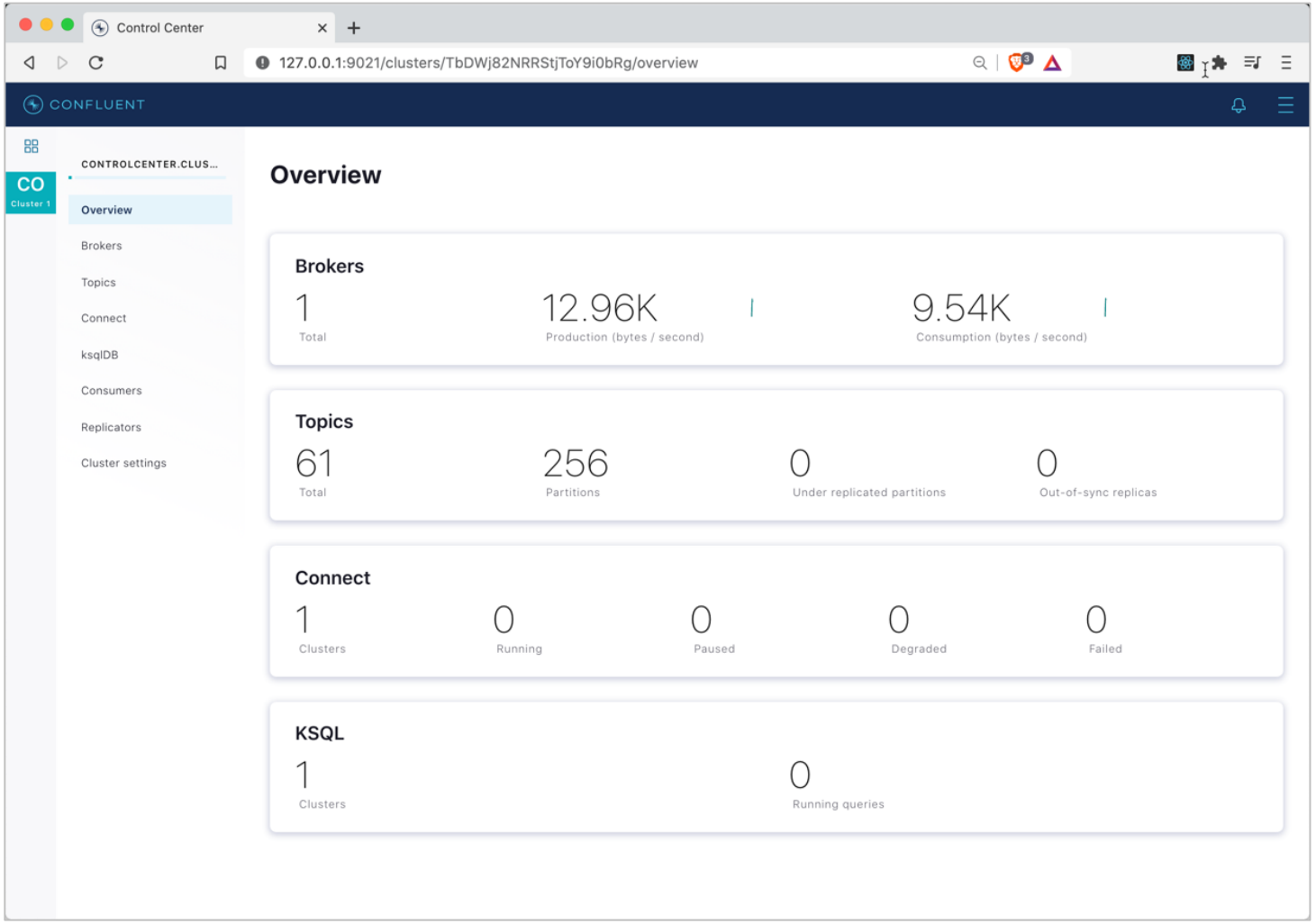
```
$ docker-compose ps
```

```
     Name                   Command                  State                          Ports
------------------------------------------------------------------------------------------------------
broker            /etc/confluent/docker/run        Up                   0.0.0.0:9092->9092/tcp
connect           /etc/confluent/docker/run        Up                   0.0.0.0:8083->8083/tcp, 9092/tcp
control-center    /etc/confluent/docker/run        Up                   0.0.0.0:9021->9021/tcp
ksql-datagen      bash -c echo Waiting for K ...   Up
ksqldb-cli        /bin/sh                          Up
ksqldb-server     /etc/confluent/docker/run        Up                   0.0.0.0:8088->8088/tcp
rest-proxy        /etc/confluent/docker/run        Up                   0.0.0.0:8082->8082/tcp
schema-registry   /etc/confluent/docker/run        Up                   0.0.0.0:8081->8081/tcp
zookeeper         /etc/confluent/docker/run        Up                   0.0.0.0:2181->2181/tcp, 2888/tcp, 3888/tcp
```

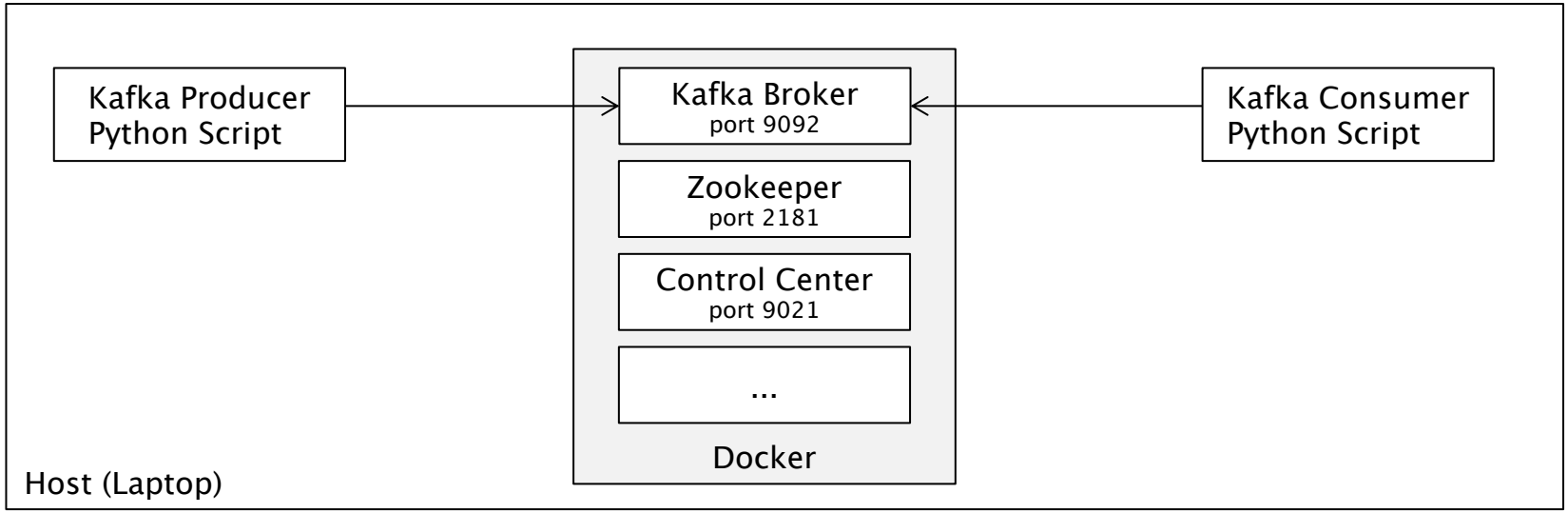
Confluent control center at http://localhost:9021/

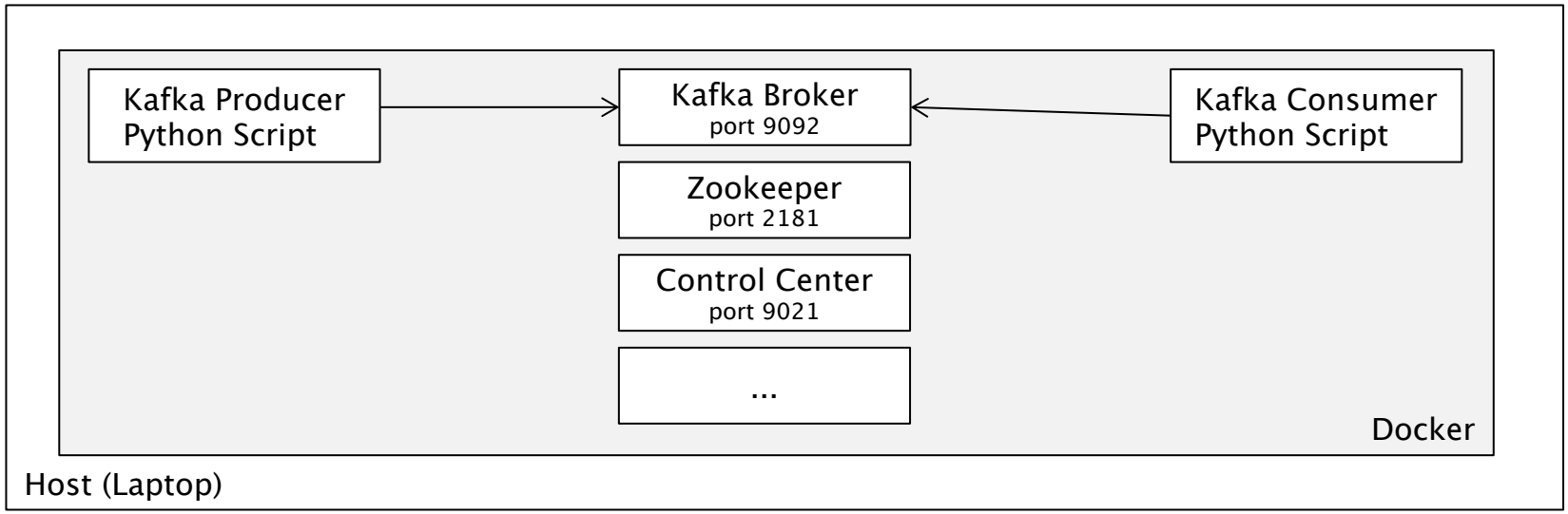
Broker (server) at localhost:9092 -> used by producers and consumers to connect

Confluent Control Center



Mögliche Setups





Projekt

Projektbeschreibung

Projektgruppen

- Gruppe aus 2-3 Teilnehmern
- **Ziel**: Prototypische Entwicklung einer **Data Pipeline**, die das Konsumieren, Speichern, Verarbeiten und Visualisieren von Daten umfasst

Deliverables

(1) Projektbericht

- Ein Bericht pro Gruppe, ca. 15 Seiten (± 5 Seiten).
- Jeder soll einen oder mehrere Abschnitte beitragen, dabei soll vermerkt werden, wer welchen Abschnitt geschrieben hat.
- Einreichen per Moodle bis **Sonntag, 15. November 2020**

(2) Präsentation am Mittwoch, 18. November 2020

- 10-15 Minuten Präsentation, jedes Gruppenmitglied soll einen Teil präsentieren
- 5 Minuten Diskussion

Projektbericht

Project report

- Little formal requirements:
  no abstract, no table of contents, no list of figures or list of tables.
- All used external sources have to be cited properly.

Goal

The project report should be a short instruction for repeating the work done in the prototype. Don't give any general information, please be specific to your prototype.

- Which tools have been chosen?
- Which steps have been taken (possibly with some selected snippets of commands or code)?
- Which difficulties have occurred, and how were they solved?
- Which decisions have been made (without theoretical justification)?

Projektpräsentation

Project presentation should contain the following:

- Short introduction of the chosen data source
- Motivation from application perspective: which questions will be answered with help of the data
- Main part: Description and explanation of the data pipeline, ideally with a live demo of the running pipeline
- Conclusion: Answer the questions from the initial motivation section

Goal

The project presentation should give an overview of the prototype that you have built.

- What are the components and how are they connected with each other?
- Did you use any specific configurations of the components?
- Which difficulties have occurred, and how were they solved?

Projektaufgaben

Data Ingestion (-> Kafka Producer)

- Either from API call
- Or from a data stream (Twitter stream, Wikipedia live changes,...)

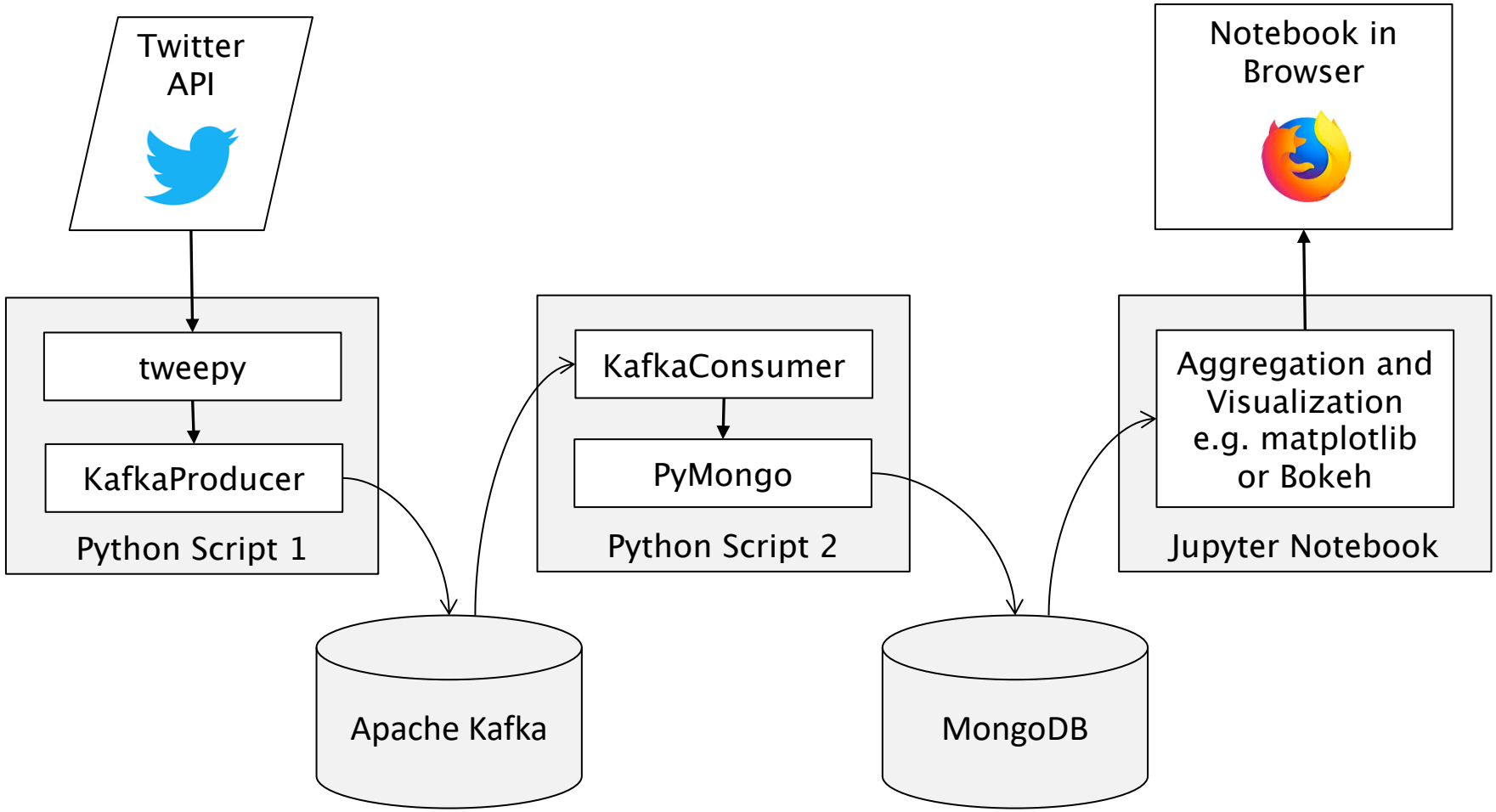
Data Storage

- Using Kafka as a buffer storage
- Store data in a NoSQL database (e.g. MongoDB), relational database, or directly process the data (e.g. Spark Streaming)

Simple data processing and visualization

- Queries with Aggregation
- Display of results: as table and/or with simple visualizations (e.g. Jupyter or Zeppelin Notebook)

Beispiel für eine Data Pipeline



Datenquelle 1: Twitter

Twitter

- Access to the Twitter live stream
- Description see https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter
- Using endpoint https://stream.twitter.com/1.1/statuses/filter.json

- Prerequisite: Twitter account und registration of an app (https://apps.twitter.com/), for getting access to the consumer key/secret and access token key/secret.
- Note: The registration as app developer might need some special justification.

- Client Libraries
  - Python: http://www.tweepy.org/
  - Node.js: https://www.npmjs.com/package/node-tweet-stream

Datenquelle 2: Facebook und Instagram

Instagram

- Read-access to basic data
- https://developers.facebook.com/docs/instagram-basic-display-api

Facebook

- Graph API
- https://developers.facebook.com/docs/graph-api/

Python client library

- https://pypi.org/project/python-facebook-api/

Datenquelle 3: Wikipedia

Wikipedia

- Data is available here (Download, API, Recent Changes Stream)
  https://meta.wikimedia.org/wiki/Research:Data

- Event stream of recent changes
  https://wikitech.wikimedia.org/wiki/EventStreams
  https://www.mediawiki.org/wiki/API:Recent_changes_stream

  Endpoint for reading data: https://stream.wikimedia.org/v2/stream/recentchange

- Examples
  http://rcmap.hatnote.com/#en
  http://listen.hatnote.com/

Datenquelle 4: New York City Administration

New York Cabs

- Data on the taxi trips in New York
  http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

- Monthly CSV files in three categories
  - Yellow — around 800-900 MB each
  - Green — around 100 MB each
  - FHV (For Hire Vehicle) — around 400-800 MB each

- Contains geographical data (Pickup / Dropdown Zone)
  => Possibility of visualization on a map

More open data from New York City Government

- https://opendata.cityofnewyork.us/

Datenquelle 5: Benzinpreise

Fuel prices

- "Tankerkönig" offers access to current fuel prices of all fuel stations in Germany. The fuel stations are obliged to report their prices to the "Markttransparenzstelle für Kraftstoffe" (MTS-K): http://www.tankerkoenig.de/

- Historical data for download
- API for accessing current data: https://creativecommons.tankerkoenig.de/
  - Name, address and geographical coordinates of the fuel station
  - Current prices for different types of fuel
  - Opening times and information whether currently open or closed

Datenquelle 6: Nachrichten

- **New York Times** offers extensive APIs for querying news articles and related information: https://developer.nytimes.com/apis
  - Article search
  - Most popular articles
  - Geographical information
  - User comments

- **FiveThirtyEight**
  - Opinion polls and other news
  - https://data.fivethirtyeight.com/

Datenquelle 7: Wetter

- Open Weather
  - Free access to current weather and limited forecast, at most 1000 calls/day
  - https://openweathermap.org/api
  - https://openweathermap.org/api/one-call-api

- World Bank “Climate Change Knowledge Portal”
  - Historical data on temperature and rainfall, aggregated for each country
  - https://climateknowledgeportal.worldbank.org/download-data

- Weather Underground
  - Current and historical weather parameters
  - https://www.wunderground.com/

Datenquelle 8: Börse

- Alpha Vantage
  - Free real-time stock data
  - https://www.alphavantage.co
- IEX Cloud
  - Real-time data for financial applications
  - https://iexcloud.io/
- Quantopia
  - Quantitative finance data including real-time stock prices
  - https://www.quantopian.com
- Quandl
  - Economic and financial data
  - https://www.quandl.com/search

Datenquelle 9: Filme und Serien

- The Movie Database (TMDb)
  - Access to movie and series metadata
  - https://www.themoviedb.org/documentation/api

- Python client library
  - https://pypi.org/project/tmdbsimple/
  - and others

Datenquelle 10: Verkehr

- Pedestrians in Germany cities
  - Free access to number of people passing a specific point
  - https://hystreet.com/
  - Python client: https://github.com/JohannesFriedrich/hystReet

- Airplanes
  - FlightRadar24
  - https://www.flightradar24.com/
  - Only aggregate data can be downloaded

- Ships
  - MarineTraffic or Vesselfinder
  - https://www.marinetraffic.com/
  - https://www.vesselfinder.com/
  - No free download