

# DWH Cluster Analytics

# Prototype Architecture

User Interface Layer

IBM Cognos Analytics 11.1 (Web-Based)  
 Location: User Browser


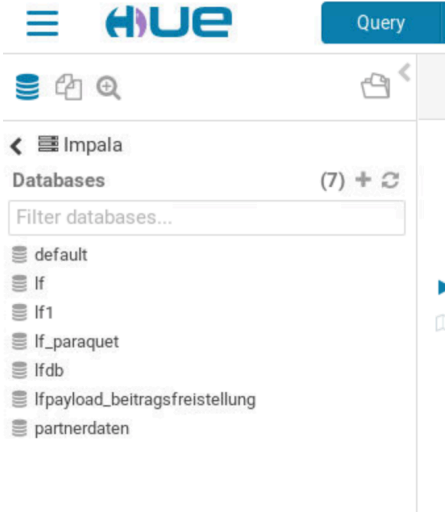
Application Layer

IBM Cognos Analytics Server  
 Location: 10.85.52.13

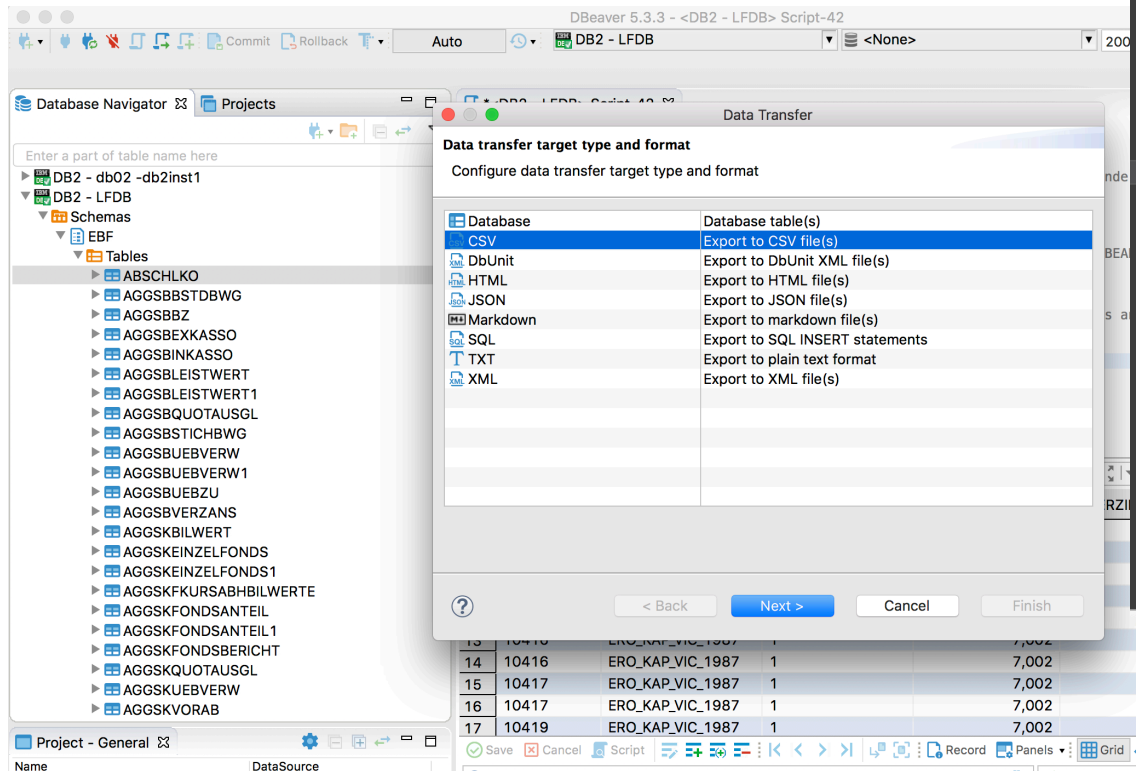
Data Layer

IBM Cognos DB2  
 Content Store  
 Location:  
 10.85.200.180

Cloudera CDH 6.2

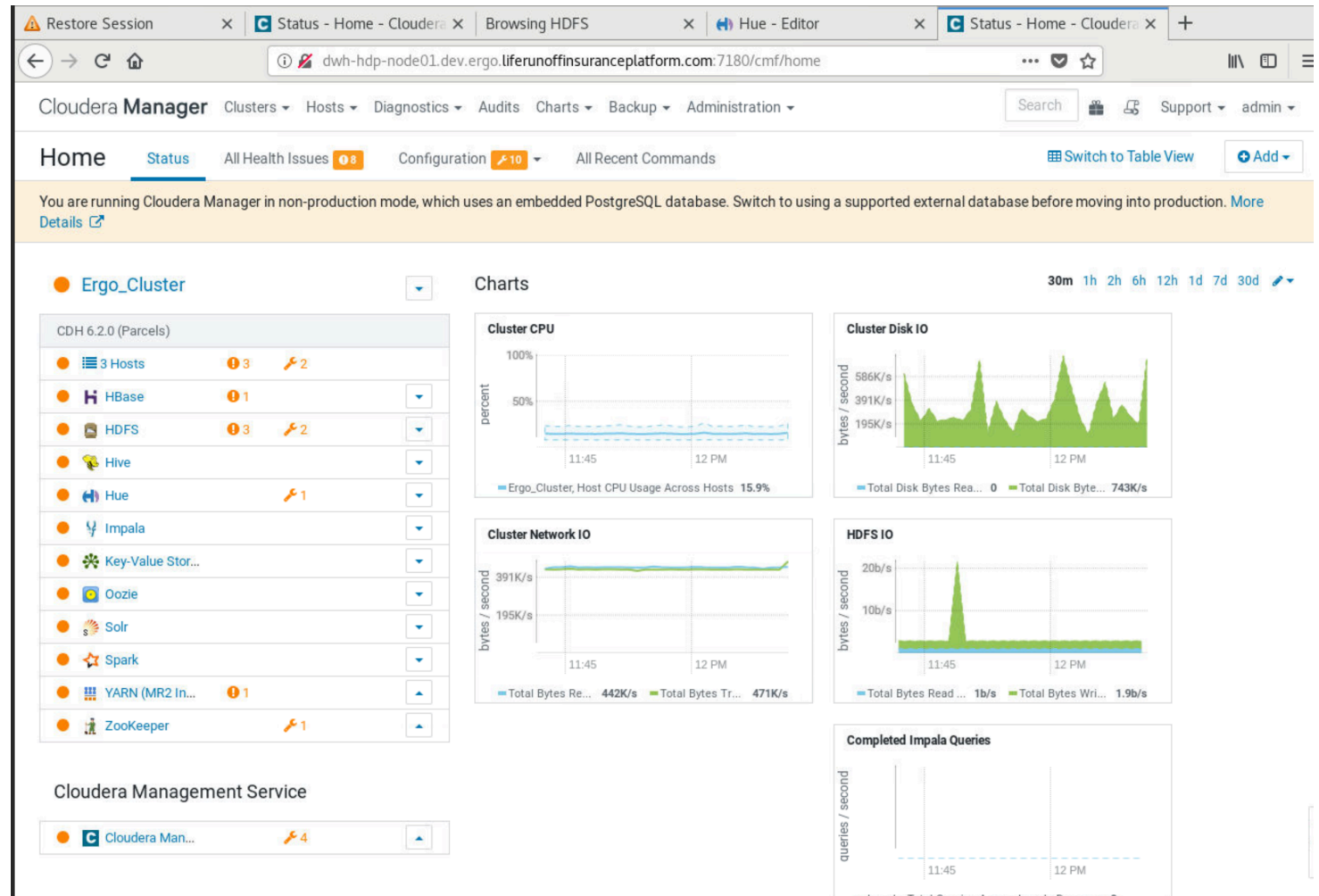
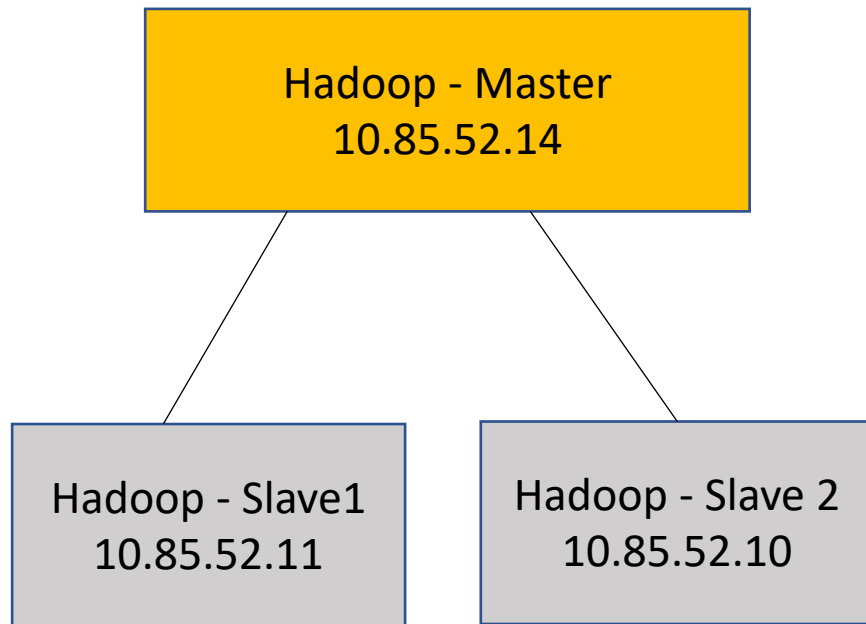
# Data Extraction



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ZCfrLfReadcompletepartnerResponse>
3   <Return>
4     <Messages>
5       <item>item</item>
6     </Messages>
7     <ProcessNumber>ProcessNumber</ProcessNumber>
8   </Return>
9   <Value>
10    <AddressList>
11      <item>
12        <AddressAddition1>AddressAddition1</AddressAddition1>
13        <AddressAddition2>AddressAddition2</AddressAddition2>
14        <AddressNr>224</AddressNr>
15        <AddressState>AddressState</AddressState>
16        <AddressType>AddressType</AddressType>
17        <City1>Hamburg</City1>
18        <City2>City2</City2>
19      </item>
20    </AddressList>
21  </Value>
22 </ZCfrLfReadcompletepartnerResponse>
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
```

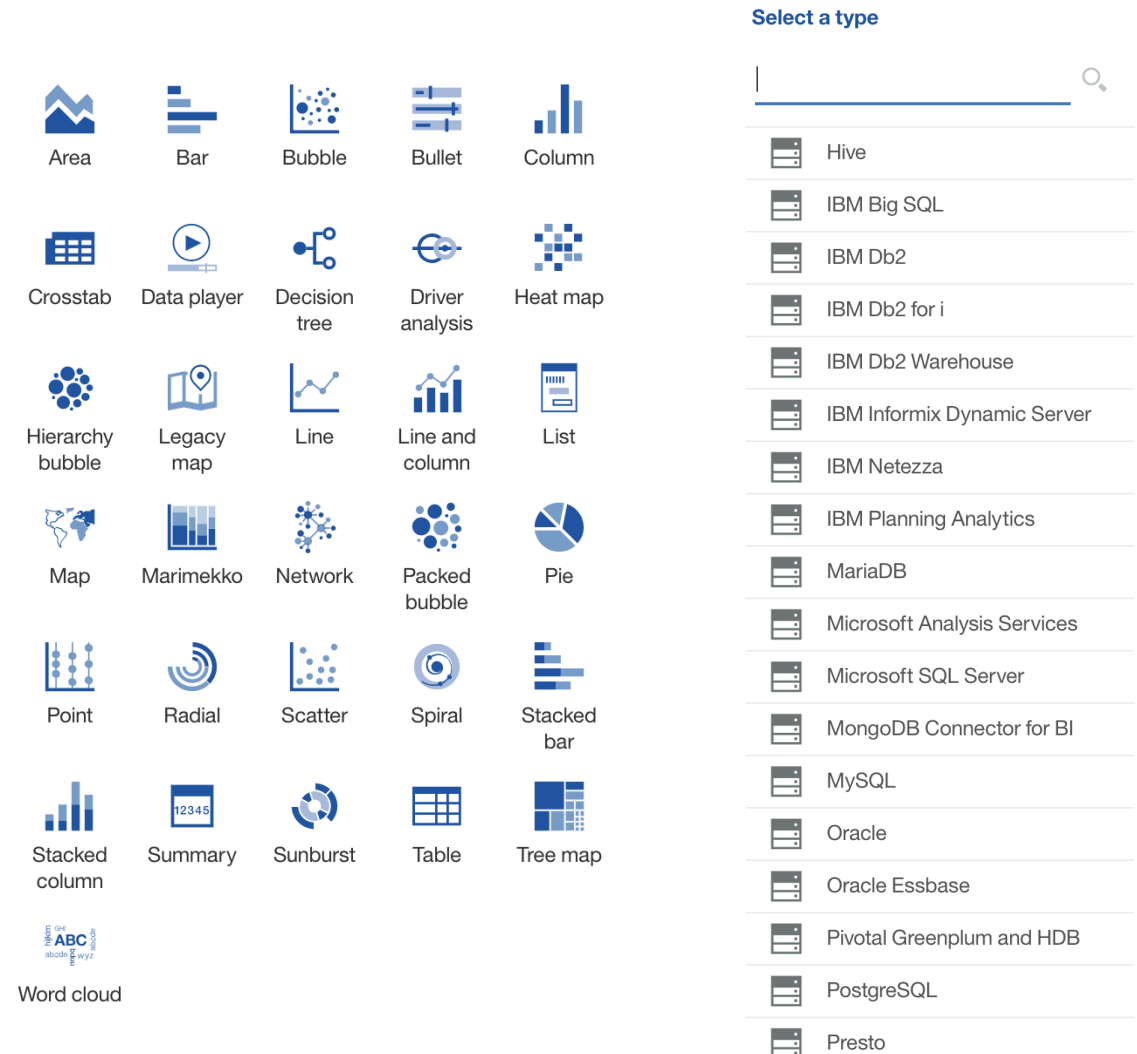
```
1 import collections
2 from pyspark.sql import SparkSession, SQLContext, HiveContext
3 import pandas as pd
4 import itertools
5 from pyspark.sql.types import *
6 from pyspark.sql.functions import *
7 import numpy
8
9 # initialise sparksession and set config
10 spark = SparkSession.builder. \
11   config('spark.jars.packages', 'com.databricks:spark-xml_2.11:0.5.0'). \
12   enableHiveSupport(). \
13   getOrCreate()
14
15 spark.conf.set("spark.debug.maxToStringFields", 10000)
16
17 # Initilaize hive repo
18 sc = spark.sparkContext
19 sc.setSystemProperty("hive.metastore.warehouse.dir", "hdfs:///user/hive/warehouse")
20 sql = SQLContext(sc)
21
22 # Point to xml files
23 hdfs_loc = 'hdfs:///user/lfpayload/lfpayload_beitragsfreistellung.xml'
24 local_loc = '/Users/krishna/PycharmProjects/xmlParser/xml/lfpayload_beitragsfreistellung.xml'
25
26 # Parse xml to spark dataframe df
27 df = spark.read.format('xml'). \
28   options(rowTag='ns2:bigDocumentPolicyEnvelope'). \
29   load(hdfs_loc)
30
31 # database name in the hive metastore
32 database_name = "lfpayload_beitragsfreistellung"
33
34 # Flatten the xml to fit into hive relational model
35 list1 = []
36 prefix_list = []
37 def flatten(schema, prefix=None):
```

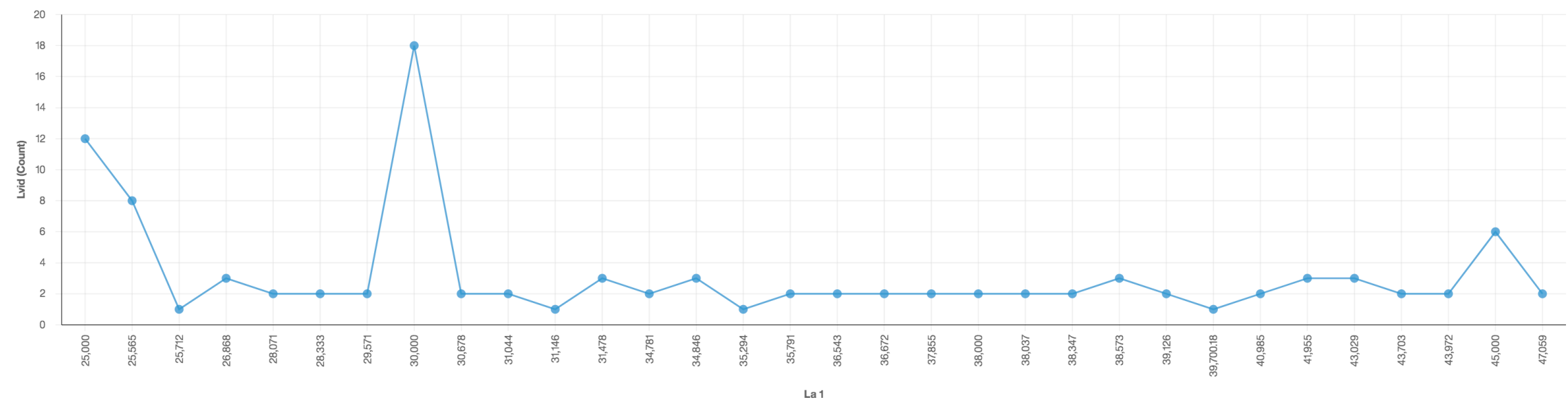
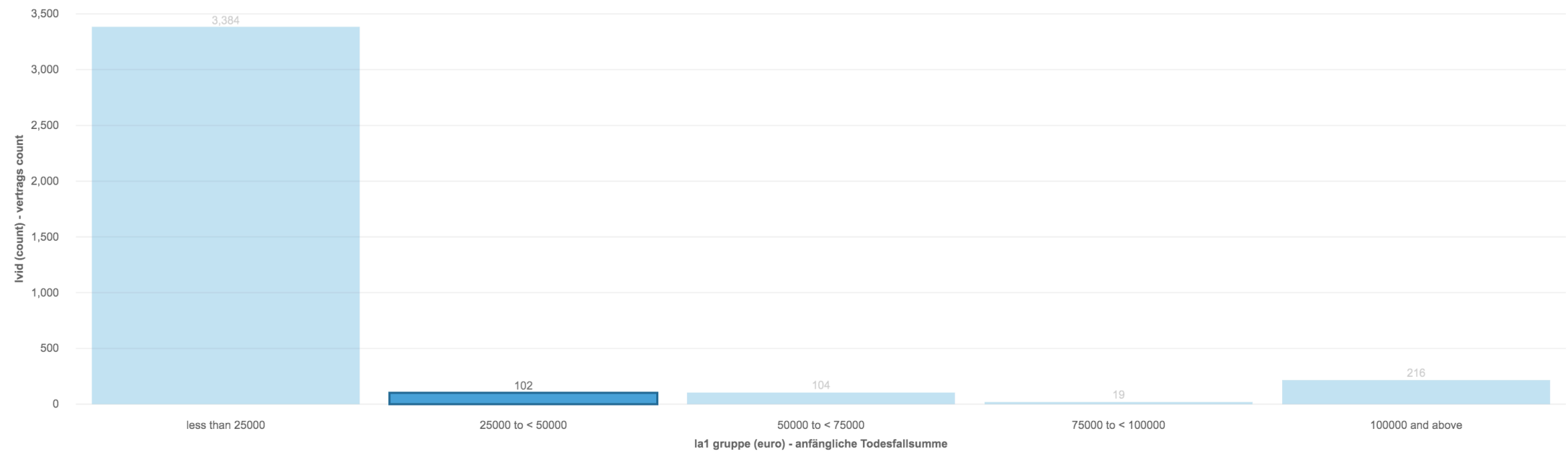
# Cloudera CDH 6.2 (3 node cluster)

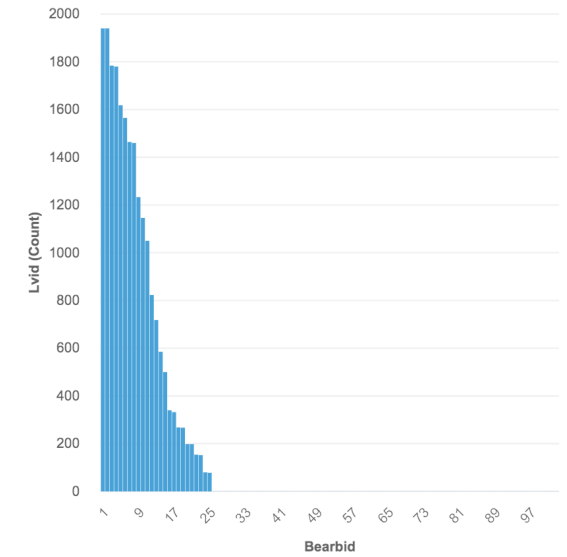
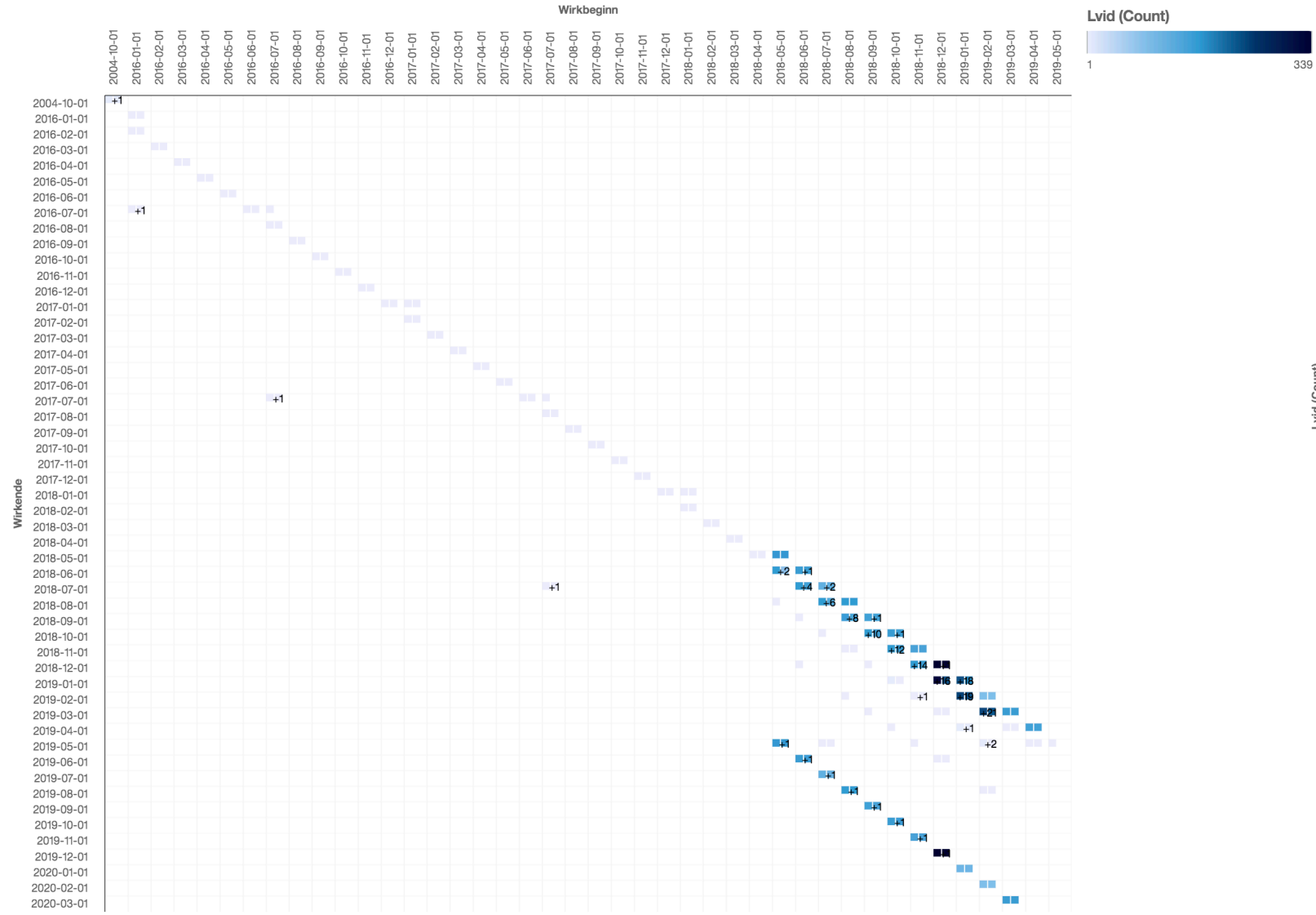


# IBM Cognos Analytics 11.1 Functionalities

- > 30 interactive visualizations that meets customer needs
- Plugins for > 30 data sources
- Filter data to excel / pdf files for sharing important insights
- Integration with complex data modelling tools
- Customization flexibility
- And many others....







# Demo

- Cloudera hadoop cluster setup

<https://sys1010.dev.ergo.liferunoffinsuranceplatform.com/>

- Reporting with cognos

<https://sys1014.dev.ergo.liferunoffinsuranceplatform.com/bi/?perspective=home>



THANK YOU