

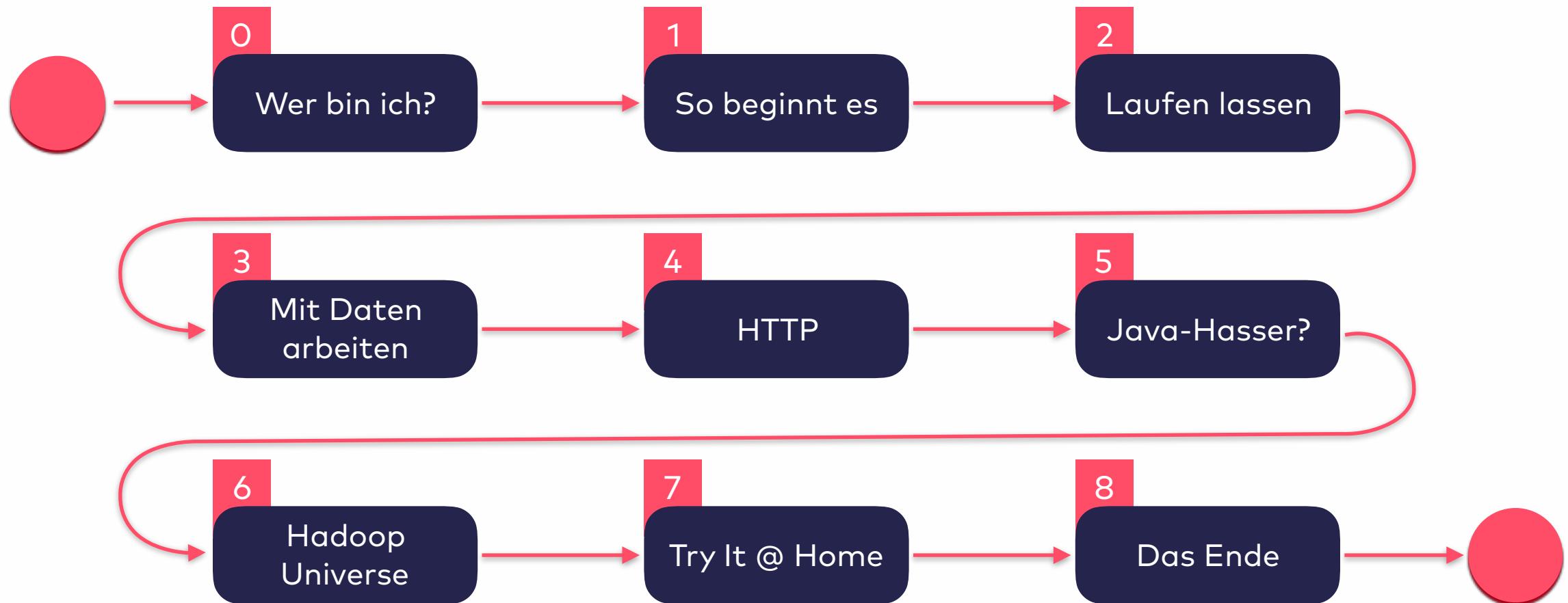
2019-01-24

Java User Group, Hamburg

# Hadoop - Taming the Elephant (With a Whale)

**INNOQ**

# Unsere Reise

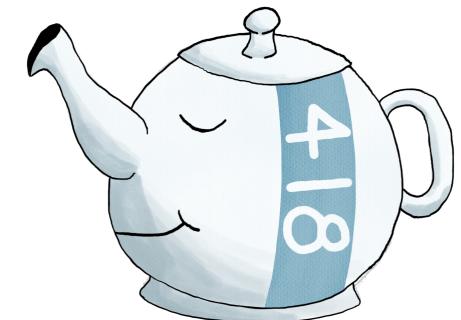


# Wer bin ich?



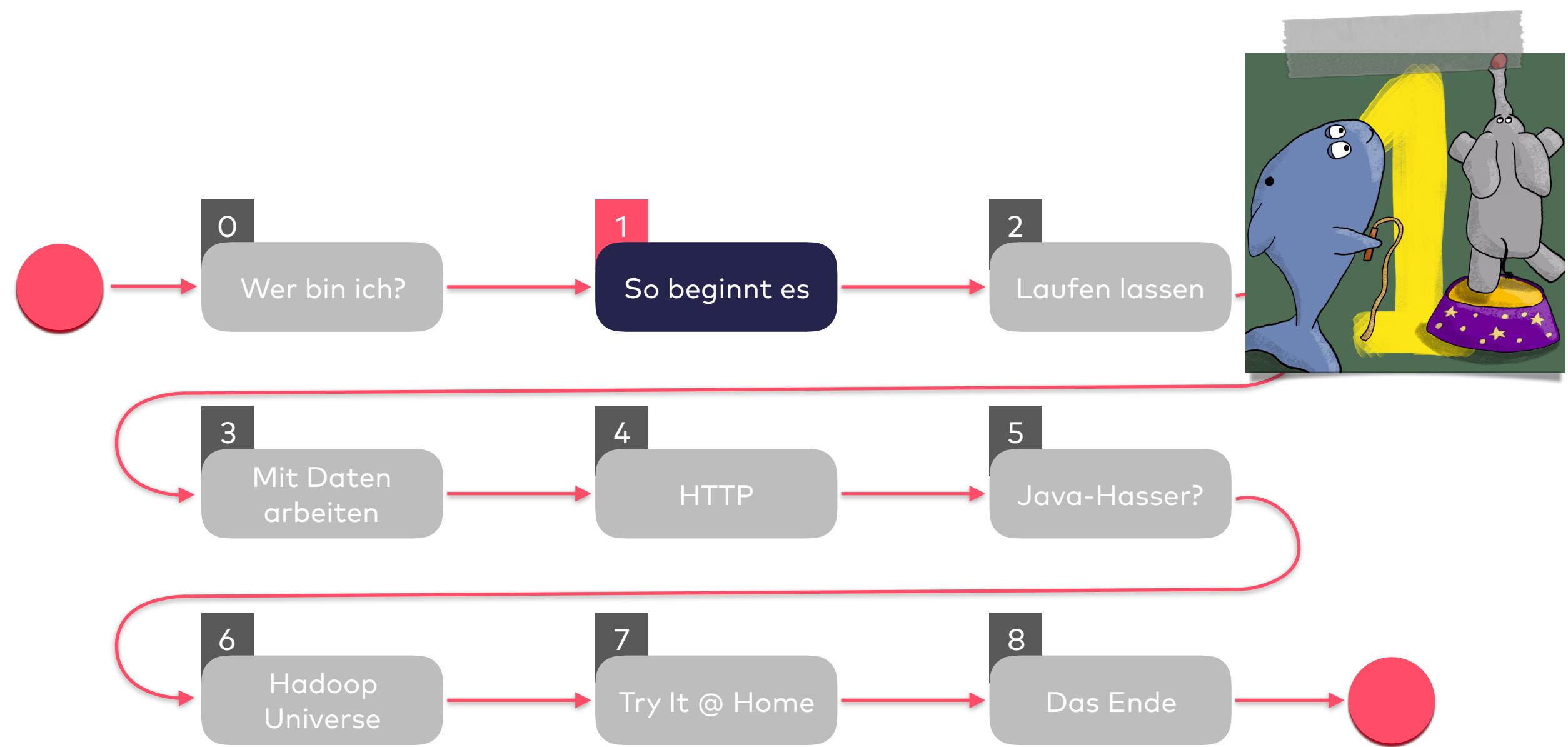
**Lisa Maria Moritz**  
[lisa.moritz@innoq.com](mailto:lisa.moritz@innoq.com)

 **Teapot4181**

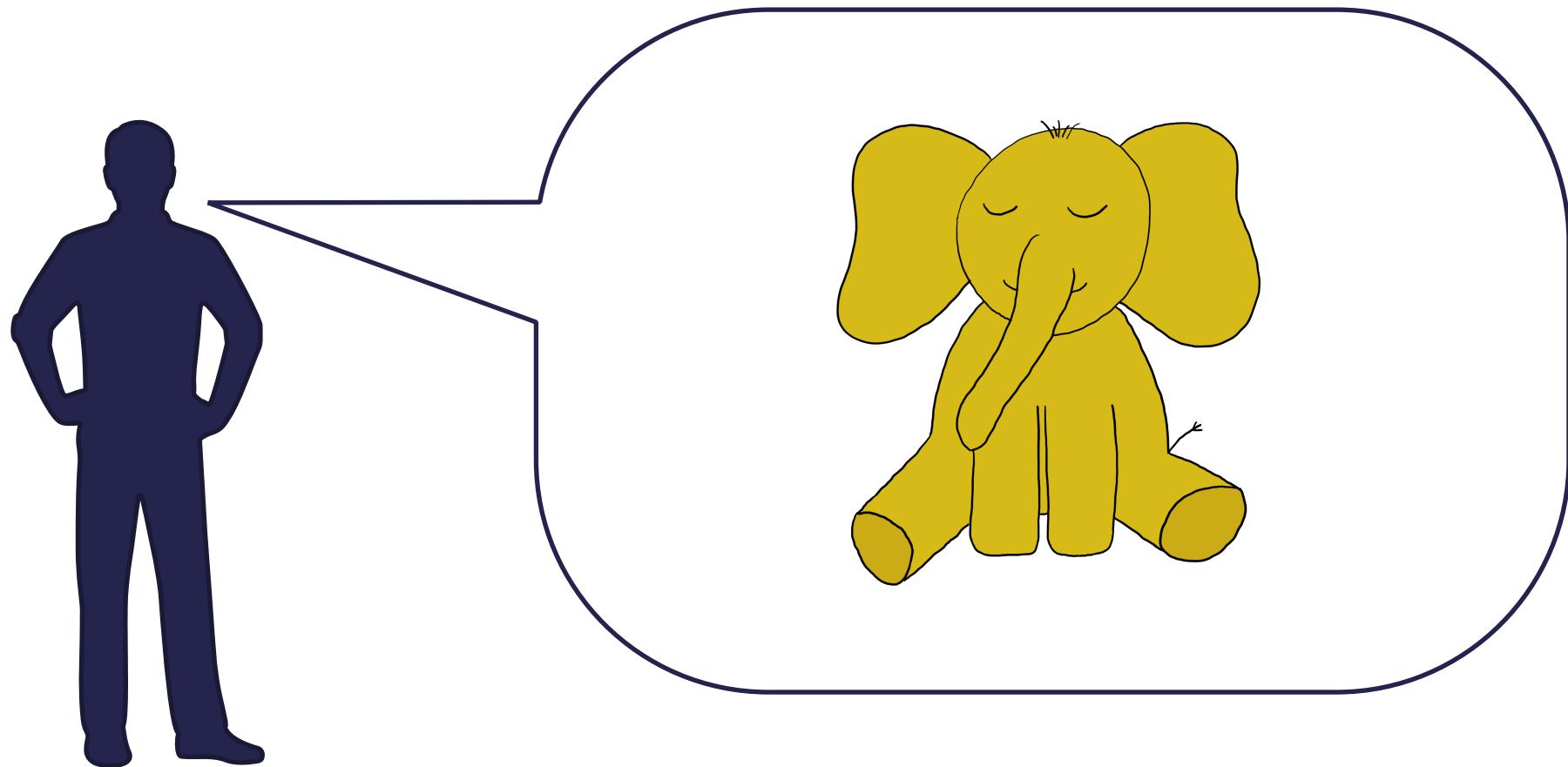


**Consultant seit  
September 2018**

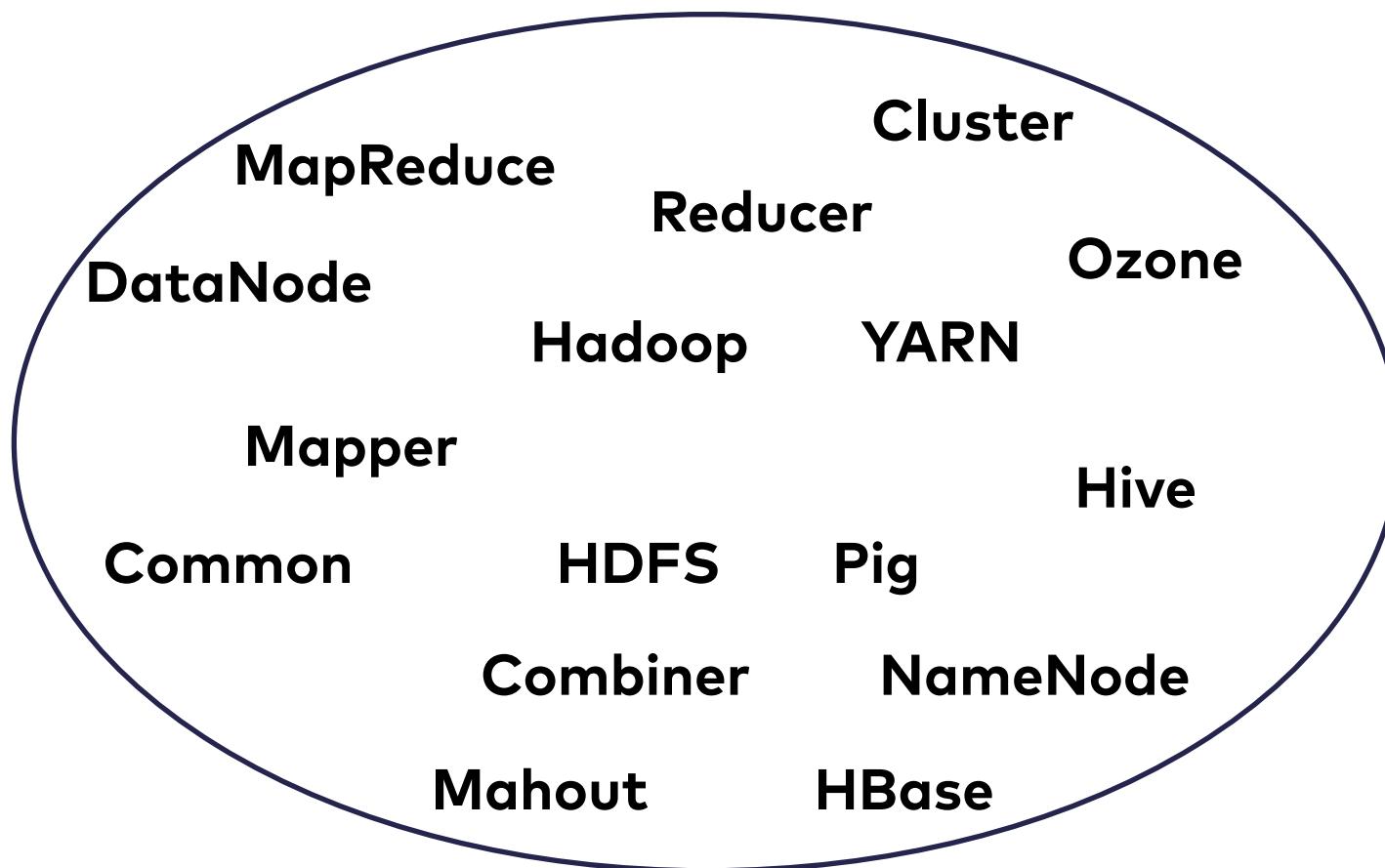
**INNOQ**  
[www.innoq.com](http://www.innoq.com)



# Wie alles begann...

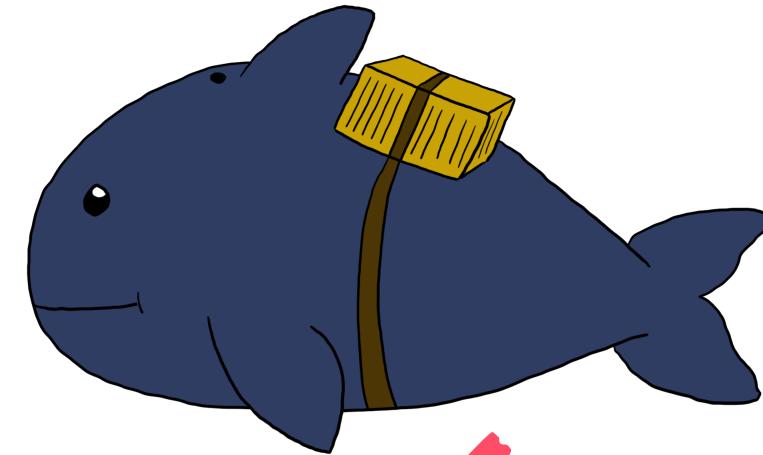
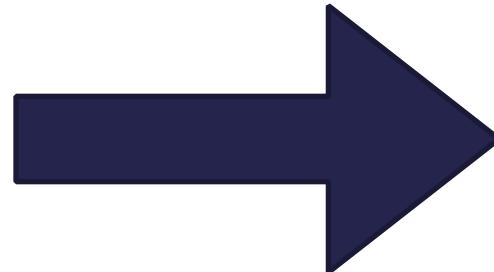


# Mehr als nur ein Elefant!

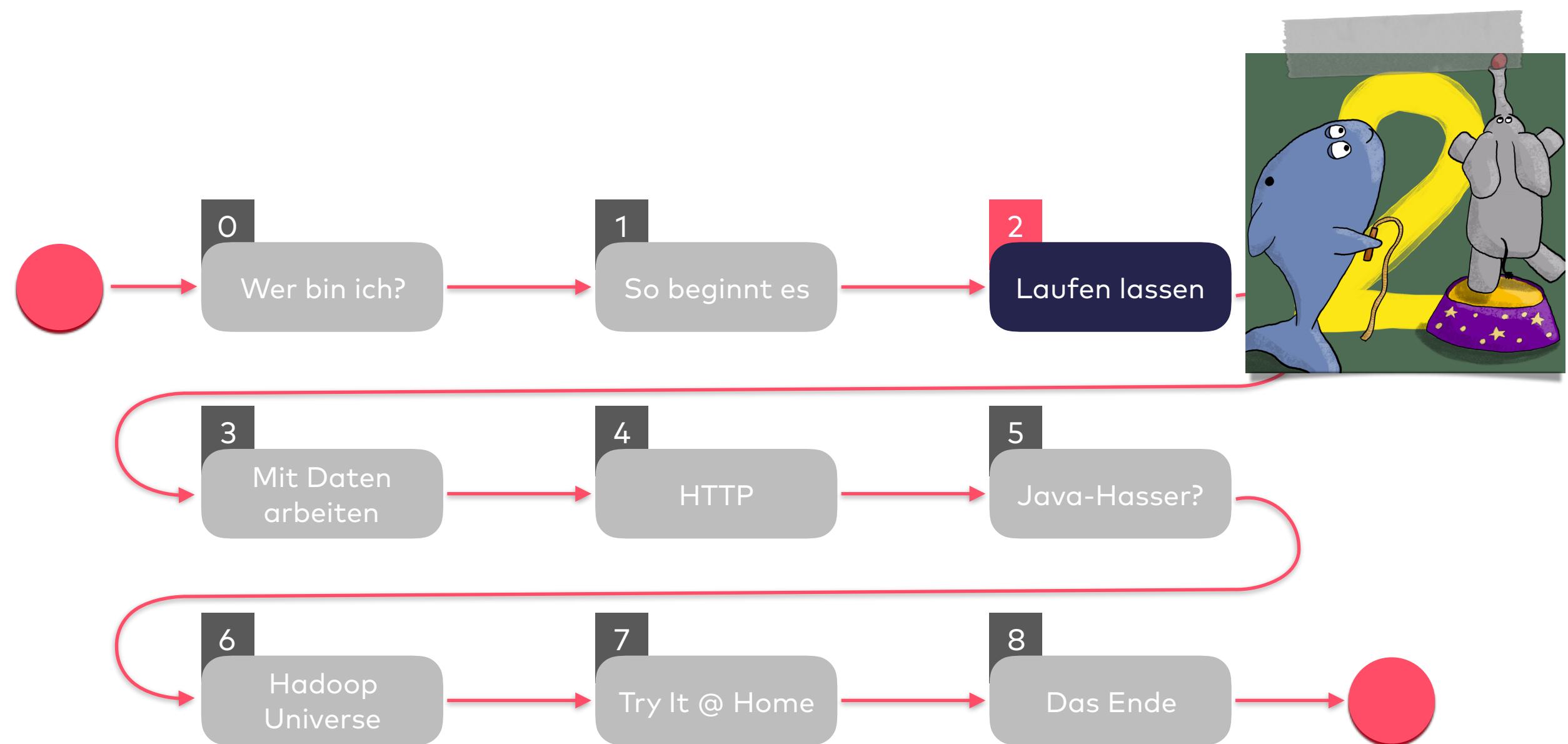


# Den Wal zur Hilfe nehmen

Cluster

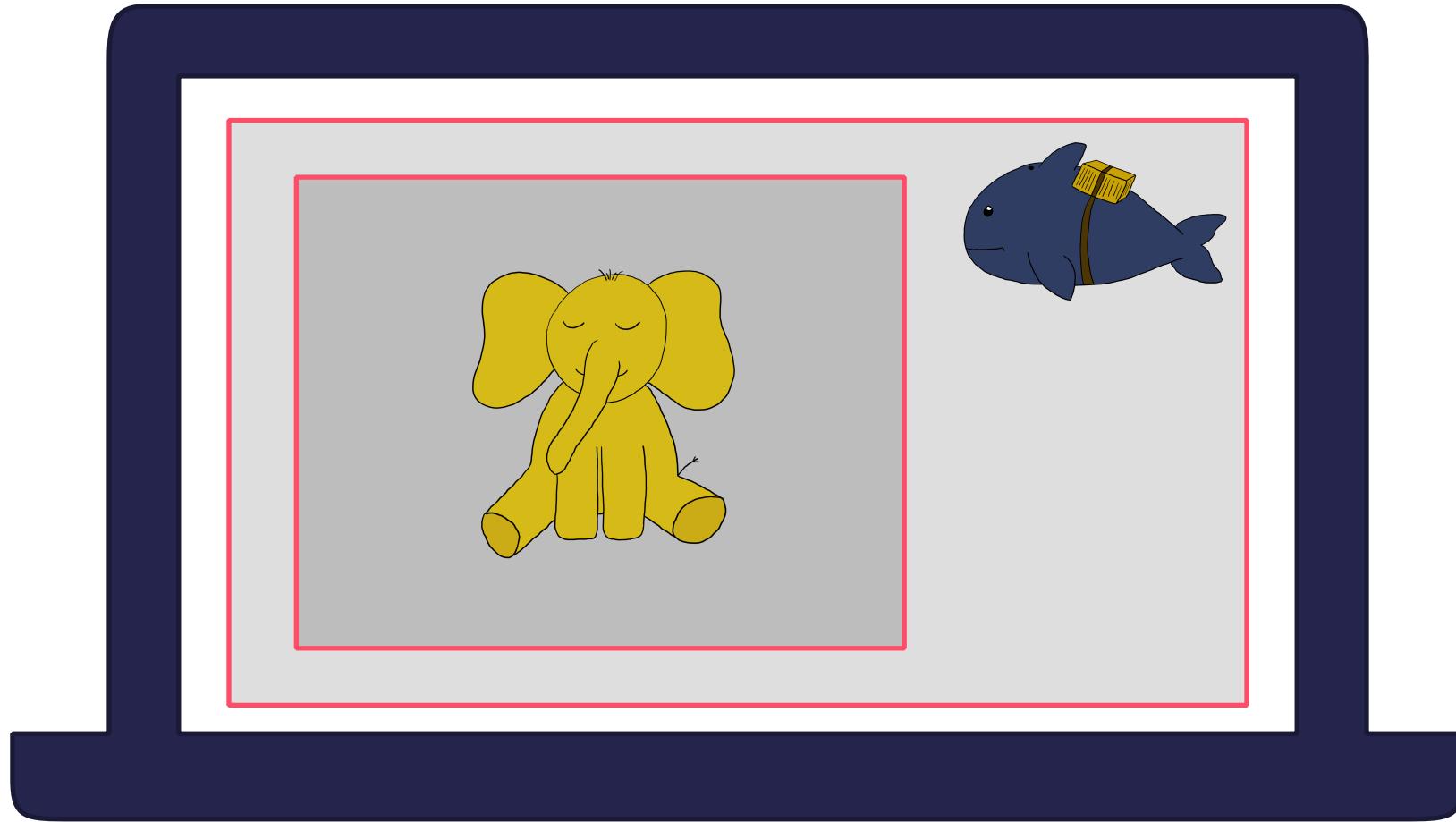


sequenceiq/hadoop-docker:2.7.0





# Ziel - Kapitel 2





# Ein paar Docker-Befehle

**Zeige alle laufenden Container**

```
docker ps
```

**Bash in laufenden Container**

```
docker exec -it <container-id> /bin/bash
```

**Container stoppen**

```
docker container stop <container-id>
```

**Container starten**

```
docker container start <container-id>
```



# Image starten

```
docker run -it \
-p 50070:50070 \
-p 8088:8088 \
-p 50075:50075 \
sequenceiq/hadoop-docker:2.7.0 \
/etc/bootstrap.sh -bash
```

Image in interaktivem Modus starten

Web-UI port weiterleiten

Port zum Job-Tracking weiterleiten

Port zum Ergebnis-Download weiterleiten

Name & Version des Docker-Images

Kommando ausführen

## BITTE BEACHTEN

Es ist nicht üblich, Hadoop in Docker zu starten!

# /etc/bootstrap.sh?

- In sequenceiq Docker-Image enthaltenes Skript
- Relevant zu wissen:
  - Startet Services
  - Hat zwei Argument-Optionen:
    - -d : Detached, im Background
    - -bash: Bash nach Start öffnen



# Hat alles geklappt? 1 / 3

Ordner mit u.a. Beispielen

```
$HADOOP_PREFIX/share/hadoop/mapreduce
```

MapReduce-Beispiel (Java)

```
hadoop-mapreduce-examples-2.7.0.jar
```

Ausführen eines MapReduce-Jobs

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-
examples-2.7.0.jar grep input output 'dfs[a-z.]+'
```



# Hat alles geklappt? 2 / 3

```
bin/hdfs dfs -cat output/*
```

```
bash-4.1# bin/hdfs dfs -cat output/*
6      dfs.audit.logger
4      dfs.class
3      dfs.server.namenode.
2      dfs.period
2      dfs.audit.log.maxfilesize
2      dfs.audit.log.maxbackupindex
1      dfsmetrics.log
1      dfsadmin
1      dfs.servers
1      dfs.replication
1      dfs.file
```



# Hat alles geklappt? 3 / 3

Browsing HDFS    +

localhost:50070/explorer.html#/user/root/output

Hadoop Overview Datanodes Snapshot Startup Progress Utilities ▾

## Browse Directory

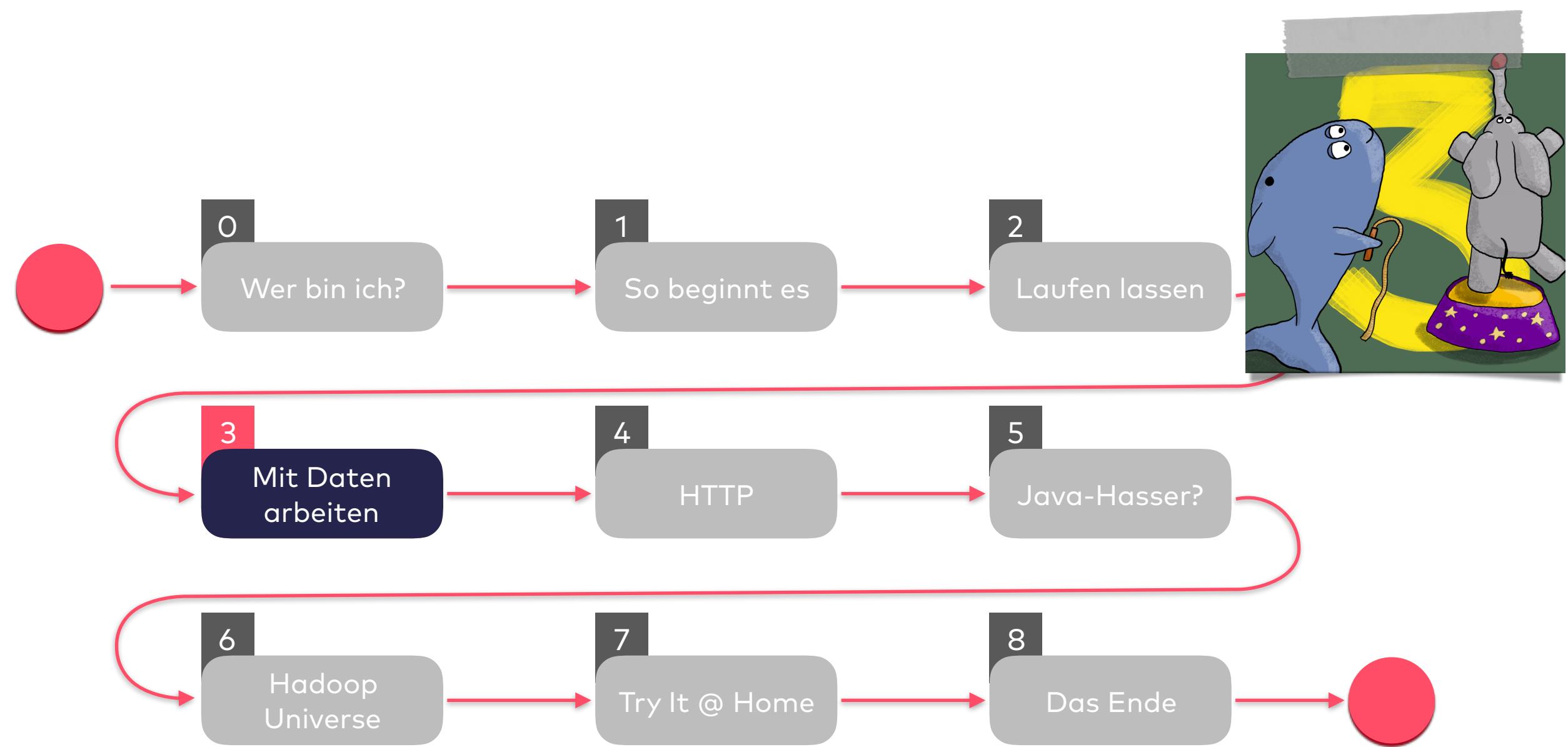
/user/root/output

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	1/9/2019, 4:55:08 PM	1	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	197 B	1/9/2019, 4:55:08 PM	1	128 MB	part-r-00000

Hadoop, 2014.

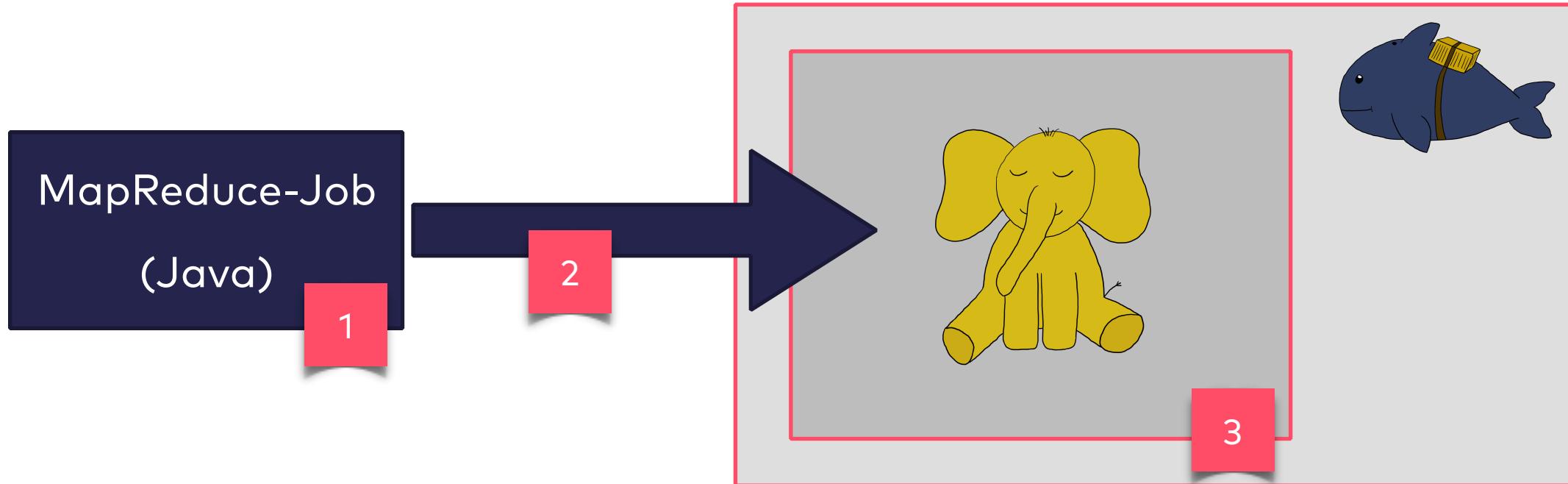


# Hadoop läuft in Docker





# Ziel - Kapitel 3





# MapReduce?

## MapReduce

Mapper

Reducer

Combiner

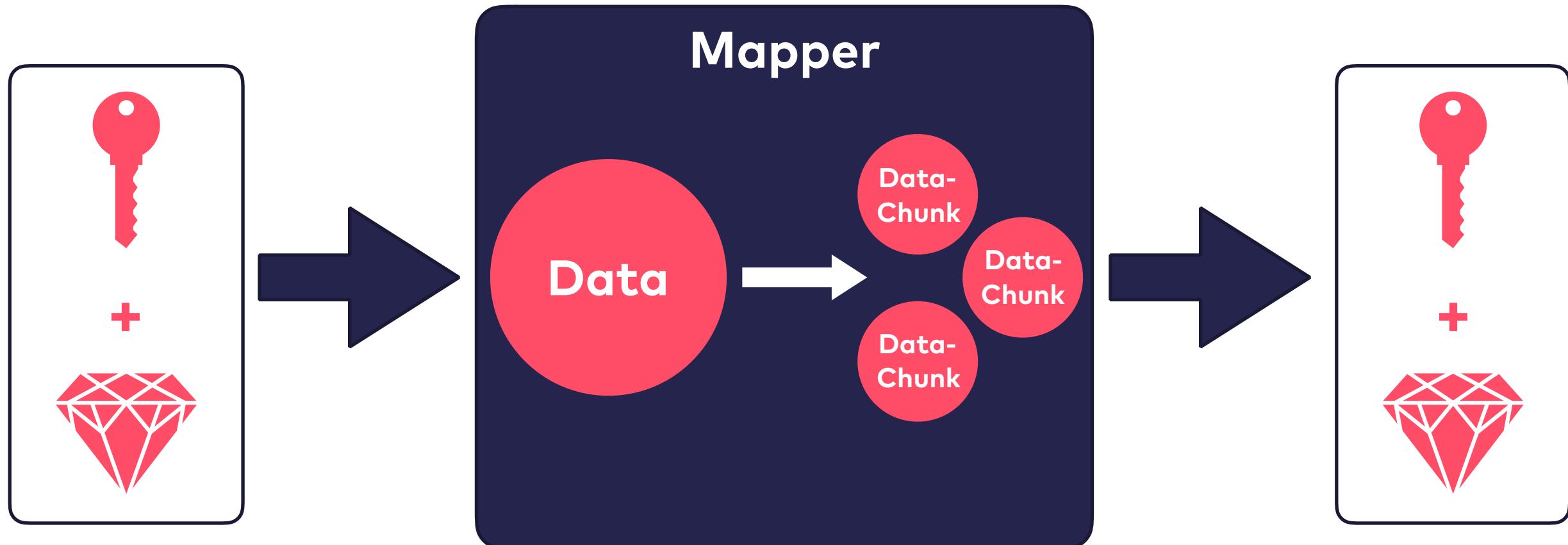


# MapReduce: Beispiel

ID	Tierart	Name
1	Hund	Berta
2	Katze	Miezi
3	Katze	Nyan
4	Hund	Napoleon
...		



# Mapper



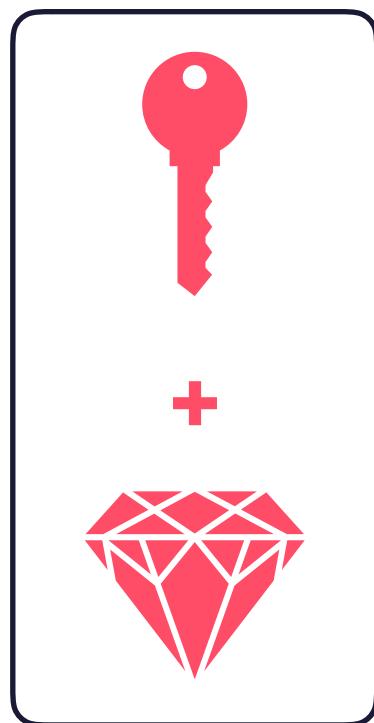
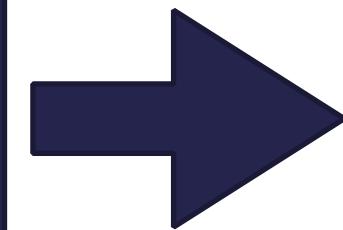
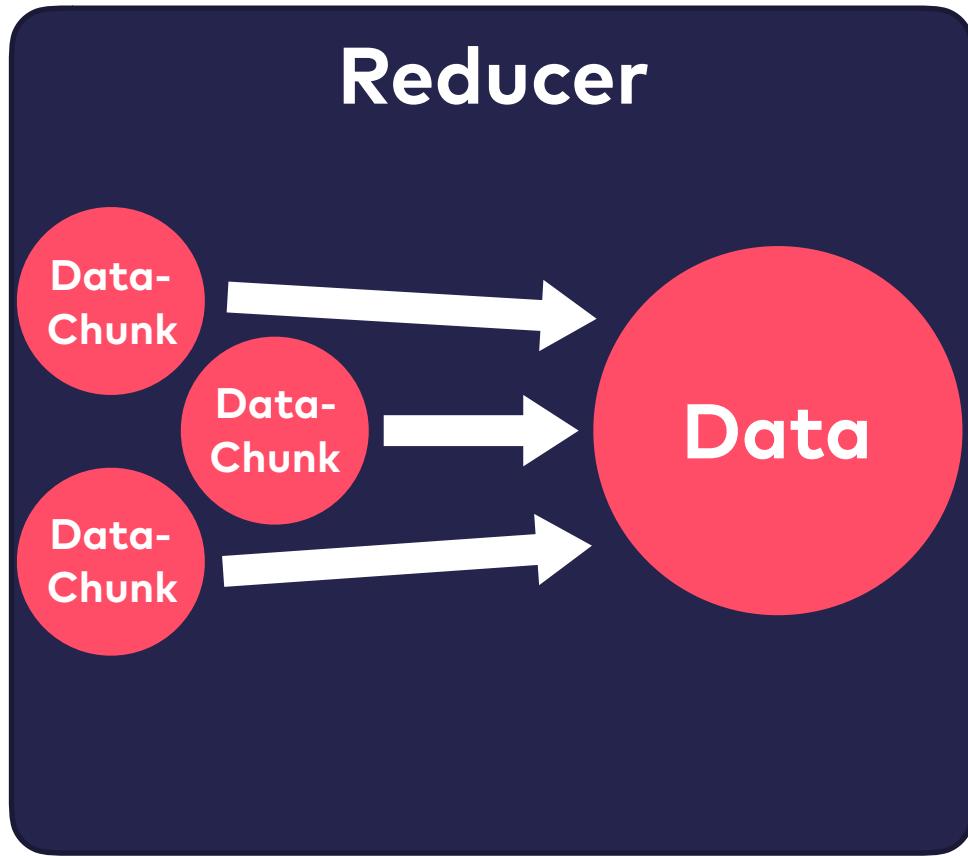
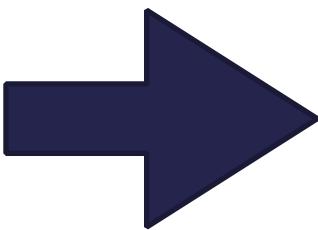
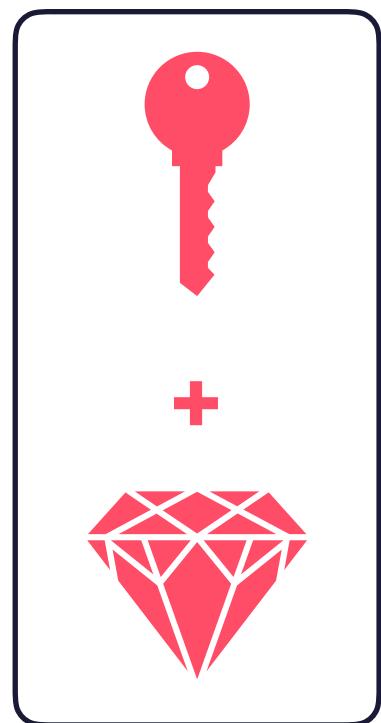


# Mapper: Beispiel





# Reducer





# Reducer: Beispiel





# Combiner

## BITTE BEACHTEN

Ein Combiner kann den Reducer nicht ersetzen

- **Optional**
- **Netzwerklast reduzieren**
  - **Den Output des Mappers kombinieren und kombinierte Daten zum Reducer schicken**
- **Oftmals kann der Reducer als Combiner eingesetzt werden**



# Wir brauchen Daten!

## Columns

- A Vendor
- A Category
- A Item
- A Item Description
- A Price
- A Origin
- A Destination
- A Rating
- A Remarks

- **Kaggle:**  
**Dark Net Marketplace Data  
(Agora 2014-2015)**
- **30.98 MB**
- **CSV-Datei**

Original Daten: <https://www.kaggle.com/philipjames11/dark-net-marketplace-drug-data-agora-20142015>

Leicht adaptiert: <https://github.com/Teapot-418/hadoop-taming-the-elephant/blob/master/darknet-data.csv>

# HDFS



# Daten ins HDFS

**Daten in Container kopieren**

```
docker cp darknet-data.csv \
ab0978def5ae:/tmp/
```

**Daten ins HDFS**

```
$HADOOP_PREFIX/bin/hdfs dfs \
-put /tmp/darknet-data.csv \
/user/root/input/darknet
```



# CLI - Check

```
/usr/local/hadoop/bin/hdfs dfs -ls /user/root/input
```

```
bash-4.1# /usr/local/hadoop/bin/hdfs dfs -ls /user/root/input
Found 32 items
-rw-r--r-- 1 root supergroup          4436 2015-05-16 05:43 /user/root/input/capacity-scheduler.xml
-rw-r--r-- 1 root supergroup          1335 2015-05-16 05:43 /user/root/input/configuration.xsl
-rw-r--r-- 1 root supergroup          318  2015-05-16 05:43 /user/root/input/container-executor.cfg
-rw-r--r-- 1 root supergroup          155  2015-05-16 05:43 /user/root/input/core-site.xml
-rw-r--r-- 1 root supergroup          154  2015-05-16 05:43 /user/root/input/core-site.xml.template
-rw-r--r-- 1 root supergroup 4311104 2019-01-10 04:15 /user/root/input/darknet
-rw-r--r-- 1 root supergroup          3670 2015-05-16 05:43 /user/root/input/hadoop-env.cmd
-rw-r--r-- 1 root supergroup          4302 2015-05-16 05:43 /user/root/input/hadoop-env.sh
-rw-r--r-- 1 root supergroup          2490 2015-05-16 05:43 /user/root/input/hadoop-metrics.properties
-rw-r--r-- 1 root supergroup          2598 2015-05-16 05:43 /user/root/input/hadoop-metrics2.properties
-rw-r--r-- 1 root supergroup          9683 2015-05-16 05:43 /user/root/input/hadoop-policy.xml
-rw-r--r-- 1 root supergroup          126  2015-05-16 05:43 /user/root/input/hdfs-site.xml
-rw-r--r-- 1 root supergroup          1449 2015-05-16 05:43 /user/root/input/httpfs-env.sh
-rw-r--r-- 1 root supergroup          1657 2015-05-16 05:43 /user/root/input/httpfs-log4j.properties
-rw-r--r-- 1 root supergroup          21   2015-05-16 05:43 /user/root/input/httpfs-signature.secret
-rw-r--r-- 1 root supergroup          620  2015-05-16 05:43 /user/root/input/httpfs-site.xml
-rw-r--r-- 1 root supergroup          3518 2015-05-16 05:43 /user/root/input/kms-acls.xml
-rw-r--r-- 1 root supergroup          1527 2015-05-16 05:43 /user/root/input/kms-env.sh
-rw-r--r-- 1 root supergroup          1631 2015-05-16 05:43 /user/root/input/kms-log4j.properties
```





# Web UI - Check

Hadoop Overview Datanodes Snapshot Startup Progress Utilities ▾

## Browse Directory

/user/root/input

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	4.33 KB	5/16/2015, 11:43:03 AM	1	128 MB	<a href="#">capacity-scheduler.xml</a>
-rw-r--r--	root	supergroup	1.3 KB	5/16/2015, 11:43:03 AM	1	128 MB	<a href="#">configuration.xsl</a>
-rw-r--r--	root	supergroup	318 B	5/16/2015, 11:43:03 AM	1	128 MB	<a href="#">container-executor.cfg</a>
-rw-r--r--	root	supergroup	155 B	5/16/2015, 11:43:03 AM	1	128 MB	<a href="#">core-site.xml</a>
-rw-r--r--	root	supergroup	154 B	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">core-site.xml.template</a>
-rw-r--r--	root	supergroup	4.11 MB	1/10/2019, 10:15:36 AM	1	128 MB	<a href="#">darknet</a>
-rw-r--r--	root	supergroup	3.58 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hadoop-env.cmd</a>
-rw-r--r--	root	supergroup	4.2 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hadoop-env.sh</a>
-rw-r--r--	root	supergroup	2.43 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hadoop-metrics.properties</a>
-rw-r--r--	root	supergroup	2.54 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hadoop-metrics2.properties</a>
-rw-r--r--	root	supergroup	9.46 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hadoop-policy.xml</a>
-rw-r--r--	root	supergroup	126 B	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">hdfs-site.xml</a>
-rw-r--r--	root	supergroup	1.42 KB	5/16/2015, 11:43:04 AM	1	128 MB	<a href="#">https-env.sh</a>

← →



# Maven Dependencies

hadoop-common

hadoop-mapreduce-client-core

 Was wollen wir?

## Columns

- A Vendor
- A Category
- A Item
- A Item Description
- A Price
- A Origin
- A Destination
- A Rating
- A Remarks



## Katgeorie

**Drugs/Cannabis/Weed**



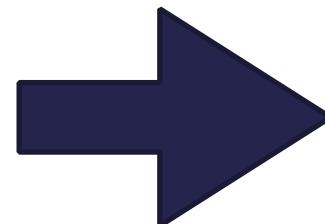
**Drugs/Ecstasy/Pills**



**Services/Other**



...



## Hauptkategorie

**Drugs**

**418**

**Services**

**42**

...



# Mapper erstellen

Extend Hadoop's Mapper

```
Mapper< [input-key], [input-value],  
        [output-key], [output-value] >
```

“map” überschreiben

```
public void map(  
    [input-key] key,  
    [input-value] value,  
    Context context  
)
```



# Mapper erstellen

```
public class MainCategoryMapper extends  
Mapper<LongWritable, Text, Text, LongWritable> {  
    @Override  
    public void map(LongWritable key, Text value, Context context) throws [...] {  
        String line = value.toString();  
        String[] lineData = line.split(",");  
        String[] categories = lineData[1].split("/");  
        if(categories.length > 0) {  
            context.write(new Text(categories[0]), key);  
        }  
    }  
}
```

Output erzeugen

## BITTE BEACHTEN

Hadoop verwendet spezielle VariablenTypen.



# Reducer erstellen

**Extend Hadoop's Reducer**

```
Reducer< [input-key], [input-value],  
         [output-key], [output-value] >
```

**“reduce” überschreiben**

```
public void reduce(  
    [input-key] key,  
    Iterable<[input-value]> values,  
    Context context  
)
```



# Reducer erstellen

```
public class CategoryCountReducer extends  
    Reducer<Text, LongWritable, Text, IntWritable> {  
    @Override  
    public void reduce(Text key, Iterable<LongWritable> values, Context context)  
        throws [...] {  
        int count = 0;  
        for (LongWritable value : values) {  
            count++;  
        }  
        context.write(key, new IntWritable(count));  
    }  
}
```

Output erzeugen



# Einstiegspunkt

A

Job erzeugen

B

Ort von Input und Output spezifizieren

C

Mapper und Reducer definieren

D

Mapper-Output definieren (Key und Value)

E

Reducer-Output definieren (Key und Value)

F

Job starten und auf Fertigstellung warten

```
A job = Job.getInstance();
job.setJobName("Main category count");

B FileInputFormat.addInputPath(job, new Path(inputPath));
FileOutputFormat.setOutputPath(job, new Path((outputPath)));

C job.setMapperClass(MainCategoryMapper.class);
job.setReducerClass(CategoryCountReducer.class);

D job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(LongWritable.class);

E job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

F job.setJarByClass(MainCategoryCount.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
```



# Job starten

**Executable Jar bauen**

```
mvn clean package
```

jar liegt in /target

**In Container kopieren**

```
docker cp darknet-mapreduce.jar\  
ab0978def5ae:/tmp/darknet-mapreduce.jar
```

**Ausführen**

```
$HADOOP_PREFIX/bin/hadoop \  
jar /tmp/darknet-mapreduce.jar
```



# Job starten

```
[...] INFO mapreduce.Job: The url to track the job: http://ab0978def5ae:8088/proxy/application_1547046677403_0012/  
[...] INFO mapreduce.Job: Running job: job_1547046677403_0012  
[...] INFO mapreduce.Job: Job job_1547046677403_0012 running in uber mode : false  
[...] INFO mapreduce.Job: map 0% reduce 0%  
[...] INFO mapreduce.Job: map 100% reduce 0%  
[...] INFO mapreduce.Job: map 100% reduce 100%  
[...] INFO mapreduce.Job: Job job_1547046677403_0012 completed successfully
```



# CLI - Check

```
$HADOOP_PREFIX/bin/hdfs dfs -cat output/darknet/main-
category-count/*
```

Chemicals	1
Counterfeits	202
Data	38
Drug paraphernalia	2
Drugs	11834
Electronics	41
Forgeries	101
Info	15
Information	13
Jewelry	23
Other	97
Services	179
Tobacco	6
Weapons	83

Hauptkategorie	Anzahl
<b>Chemicals</b>	1
<b>Counterfeits</b>	202
<b>Drug paraphernalia</b>	2
<b>Drugs</b>	11834
...	



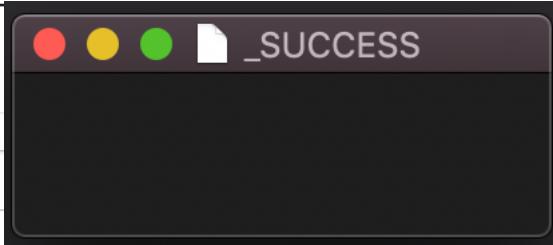
# Web UI - Check

## Browse Directory

/user/root/output/darknet/main-category-count

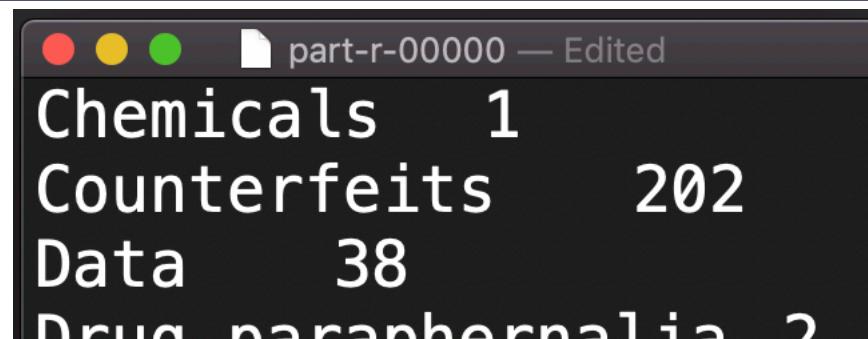
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	1/10/2019, 2:13:26 PM	1	128 MB	<a href="#">_SUCCESS</a>
-rw-r--r--	root	supergroup	176 B	1/10/2019, 2:13:26 PM	1	128 MB	<a href="#">part-r-00000</a>

Hadoop, 2014.



The screenshot shows a file viewer window titled "part-r-00000 — Edited". The file contains the following text:

```
Chemicals 1
Counterfeits 202
Data 38
Drug paraphernalia ?
```





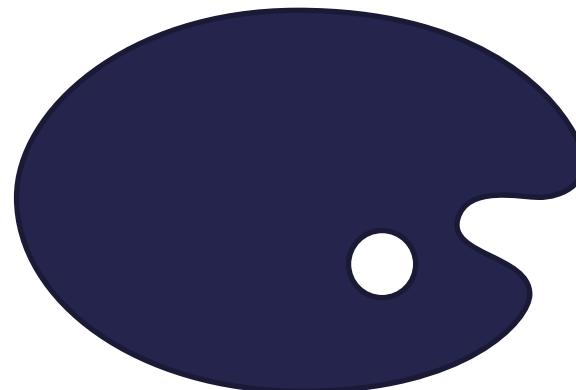
# Aus dem HDFS löschen

```
$HADOOP_PREFIX/bin/hdfs dfs -rm -r <folder-name>
```



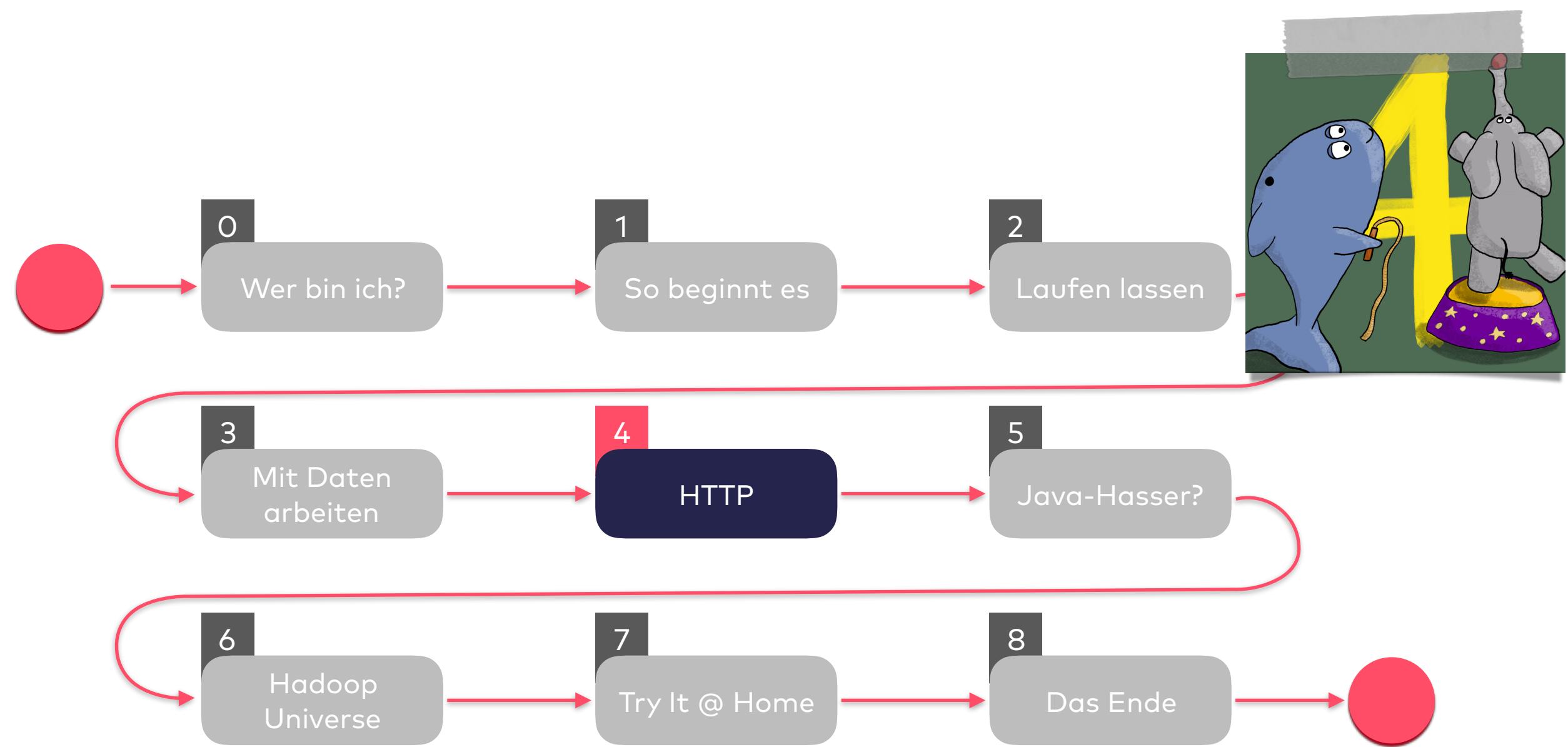
# Werdet kreativ

- Versucht doch mal:
  - Die Länder in die versendet wird zu zählen
  - Den maximalen Preis jeder Hauptkategorie aufzulisten
  - ...



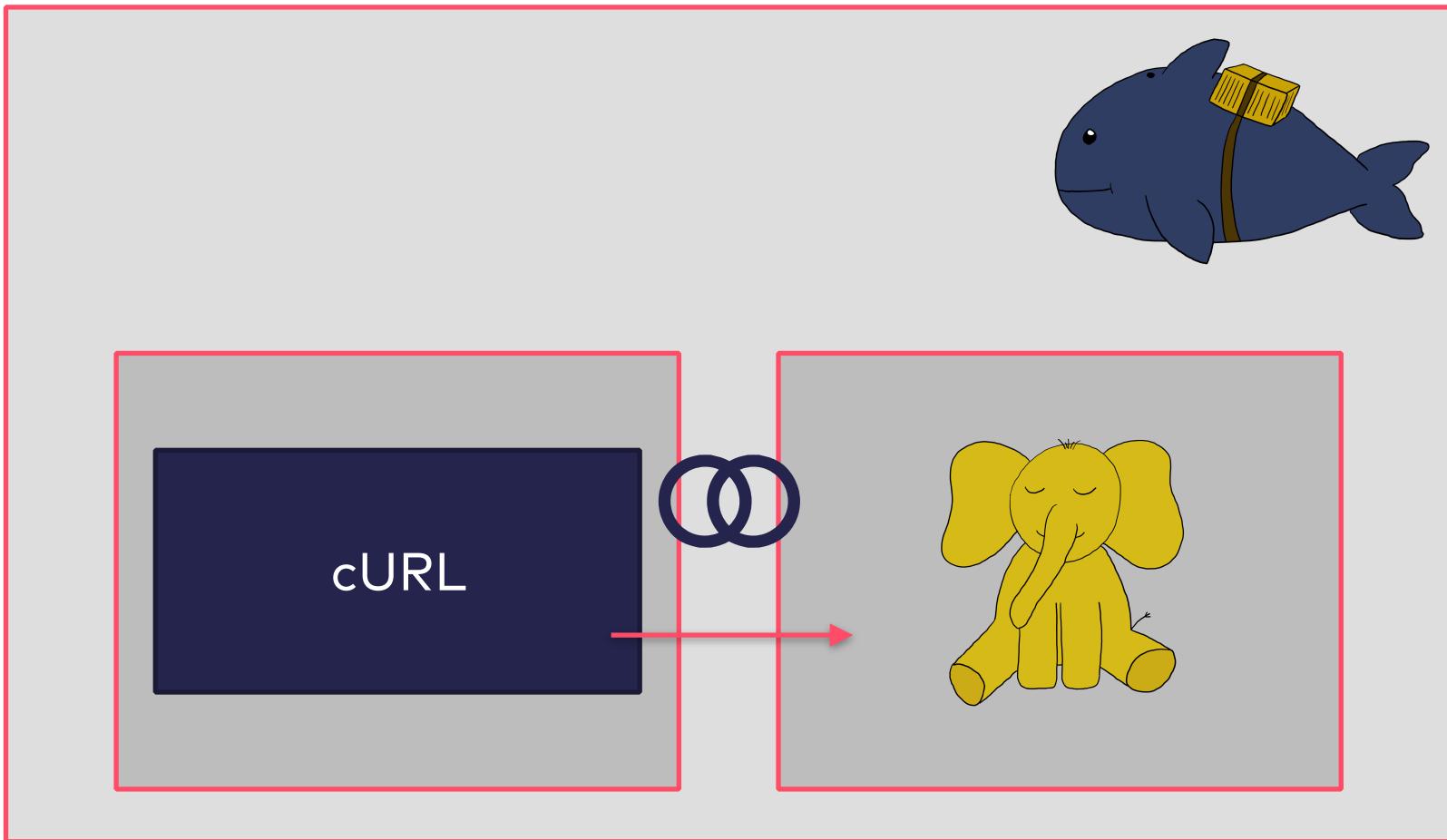


# **Erster MapReduce-Job**





# Ziel - Kapitel 4





# HttpFS?

Hadoop HDFS over HTTP



# Docker-Container linken

```
docker run \  
-it \  
--link 7680676ced68:hadoop \  
-name curl_container \  
ubuntu:latest
```

Im interaktiven Modus starten

Container linken

```
/ # cat /etc/hosts  
127.0.0.1      localhost  
::1      localhost ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
172.17.0.2      hadoop 7680676ced68 compassionate_robinson  
172.17.0.3
```



# Durch's HDFS navigieren

**Basis URL**

```
http://hadoop:50070/webhdfs/v1/user
```

**Ordner browsen**

```
curl -i -L \
"http://hadoop:50070/webhdfs/v1/user/
root/output/?op=LISTSTATUS"
```

**Dateien öffnen**

```
curl -i -L \
"http://hadoop:50070/webhdfs/v1/user/
root/output/darknet/category-count/
part-00000?op=OPEN"
```



# Durch's HDFS navigieren

## Ordner browsen

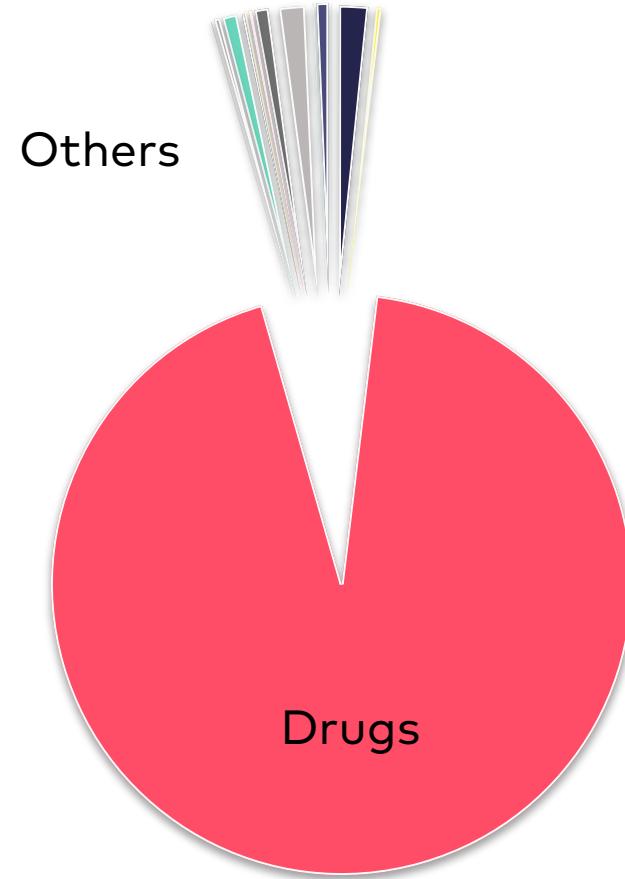
```
{
  "FileStatuses": [
    "FileStatus": [
      "accessTime": 0,
      "blockSize": 0,
      "childrenNum": 1,
      "fileId": 16435,
      "group": "supergroup",
      "length": 0,
      "modificationTime": 1547137495047,
      "owner": "root",
      "pathSuffix": "darknet",
      "permission": "755",
      "replication": 0,
      "storagePolicy": 0,
      "type": "DIRECTORY"
    ]
  }
}
```

## Dateien öffnen

```
Pragma: no-cache
Content-Type: application/octet-stream
Location: http://7680676ced68:50075/webhdfs/read?path=/Counterfeits%2F&offset=0&length=7680676ced68:9000&offset=0
Content-Length: 0
Server: Jetty(6.1.26)

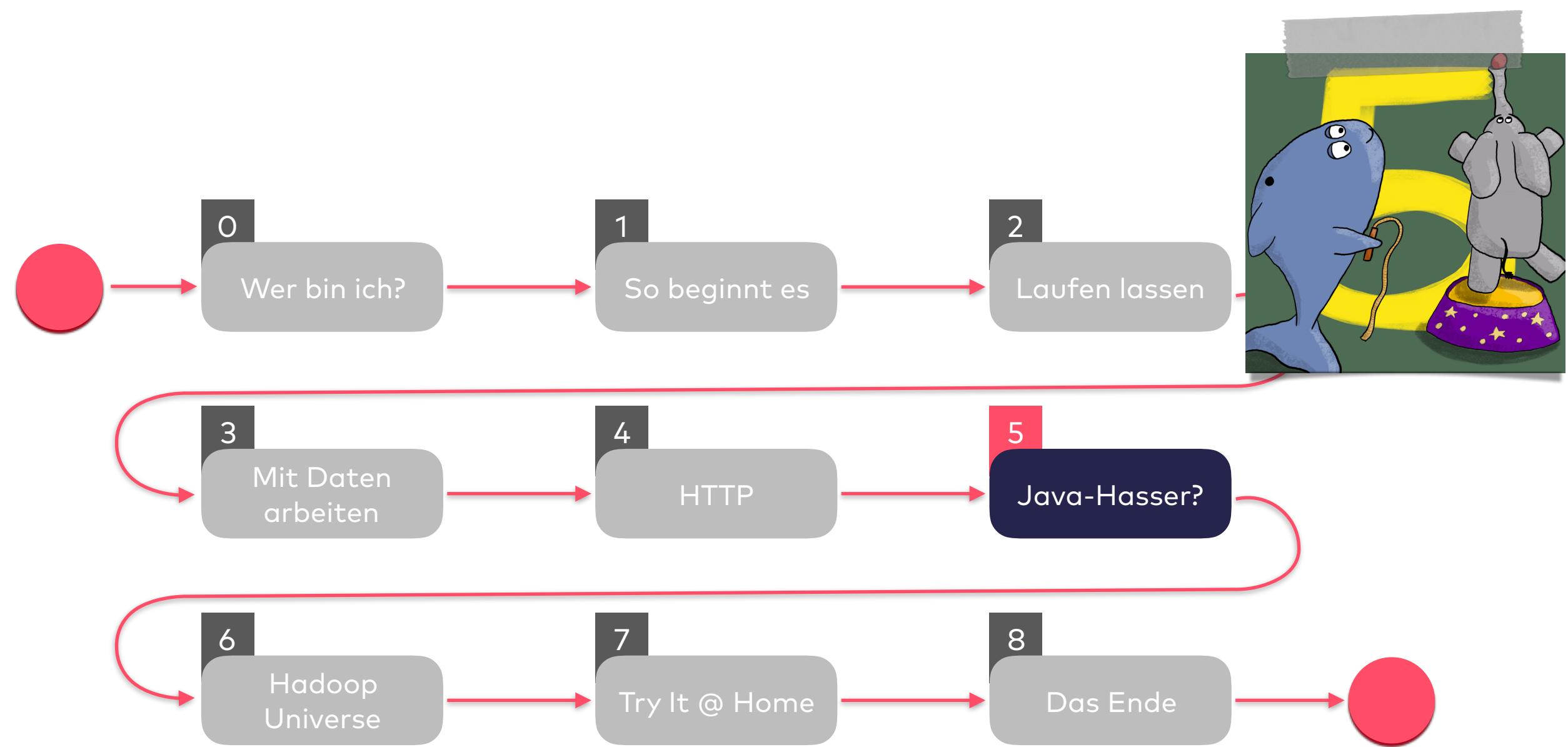
HTTP/1.1 200 OK
Access-Control-Allow-Methods: GET
Access-Control-Allow-Origin: *
Content-Type: application/octet-stream
Connection: close
Content-Length: 1905

Counterfeits/Accessories 1
Counterfeits/Clothing 8
Counterfeits/Electronics 10
Counterfeits/Money 5
Counterfeits/Watches 36
```

 Idee

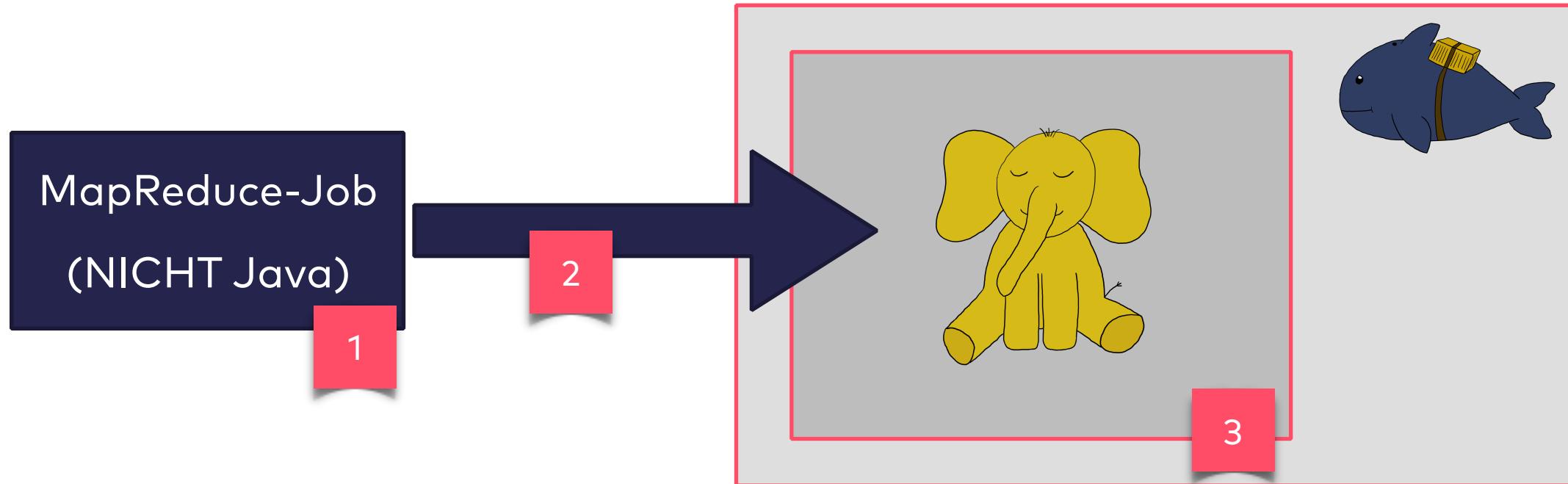


# Daten über HttpFS





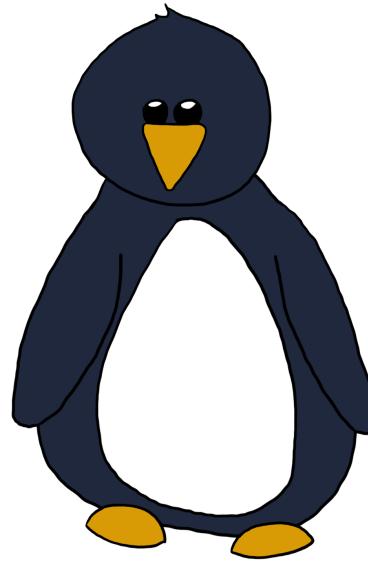
# Ziel - Kapitel 5





# Hadoop Streaming API

stdin



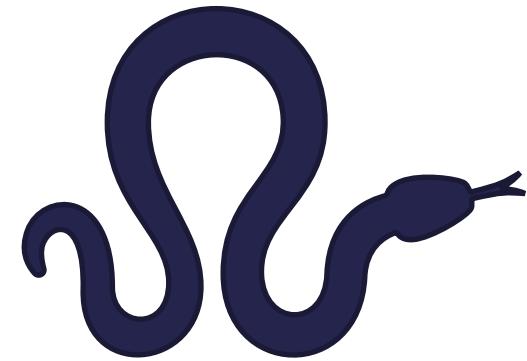
stdout



Reducer erhält Keys geordnet

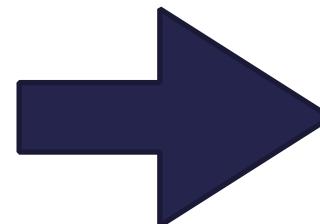


# Beispiel - Einleitung



Columns
A Vendor
A Category
A Item
A Item Description
A Price
A Origin
A Destination
A Rating
A Remarks

Kategorie
<b>Drugs/Cannabis/Weed</b>
<b>Drugs/Ecstasy/Pills</b>
<b>Services/Other</b>
...



Kategorie	Anzahl
<b>Drugs/Cannabis/Weed</b>	<b>418</b>
<b>Drugs/Ecstasy/Pills</b>	<b>42</b>
...	

 Mapper

```
#!/usr/bin/env python
import sys

id = 0

for line in sys.stdin:
    val = line.strip()
    data = val.split(',')
    category = data[1]
    print(category + '\t' + str(id))
    id = id + 1
```

stdin Zeile für Zeile verarbeiten

Output an Reducer aushändigen

 Reducer

```
previous_key = None
count = 0

for line in sys.stdin:
    (key, val) = line.strip().split('\t') Key und Value der Zeile
    if previous_key is None:
        previous_key = key
    if key == previous_key:
        count = count + 1
    else:
        print(key + '\t' + str(count))  
        count = 1  
        previous_key = key
```

Letzter? Ergebnis ausgeben

Key merken



# Ab in den Container

**Mapper in den Container**

```
docker cp mapper.py ab0978def5ae:/tmp/
```

**Reducer in den Container**

```
docker cp reducer.py ab0978def5ae:/tmp/
```

**Ausführbar machen**

```
chmod +x mapper.py  
chmod +x reducer.py
```



# Ausführen - Basis

```
$HADOOP_PREFIX/bin/hadoop jar \  
$HADOOP_PREFIX/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar
```





# Ausführen - Argumente

```
-files /tmp/mapper.py,/tmp/reducer.py \
      Dateien auf Cluster verteilen  
-input input/darknet \
      Input  
-output output/darknet/category-count \
      Output  
-mapper /tmp/mapper.py \
      Mapper setzen  
-reducer /tmp/reducer.py
      Reducer setzen
```



# CLI - Check

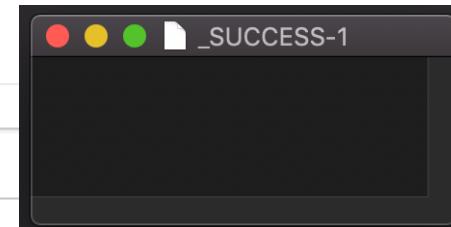
```
$HADOOP_PREFIX/bin/hdfs dfs -cat output/darknet/  
category-count/*
```

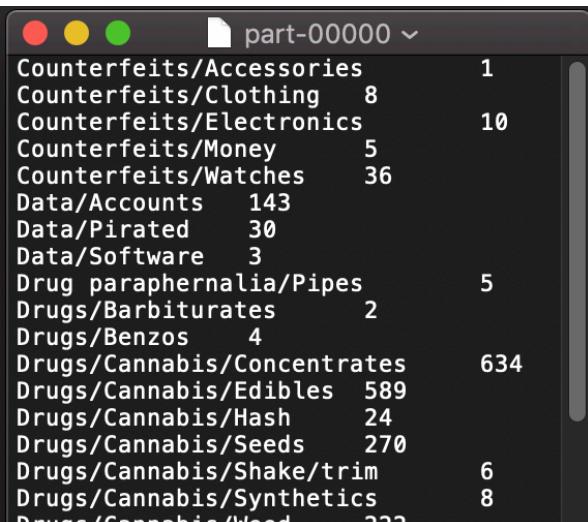
Counterfeits/Accessories	1
Counterfeits/Clothing	8
Counterfeits/Electronics	10
Counterfeits/Money	5
Counterfeits/Watches	36
Data/Accounts	143
Data/Pirated	30
Data/Software	3
Drug paraphernalia/Pipes	5
Drugs/Barbiturates	2
Drugs/Benzos	4
Drugs/Cannabis/Concentrates	634
Drugs/Cannabis/Edibles	589
Drugs/Cannabis/Hash	24

 Web UI - Check

## Browse Directory

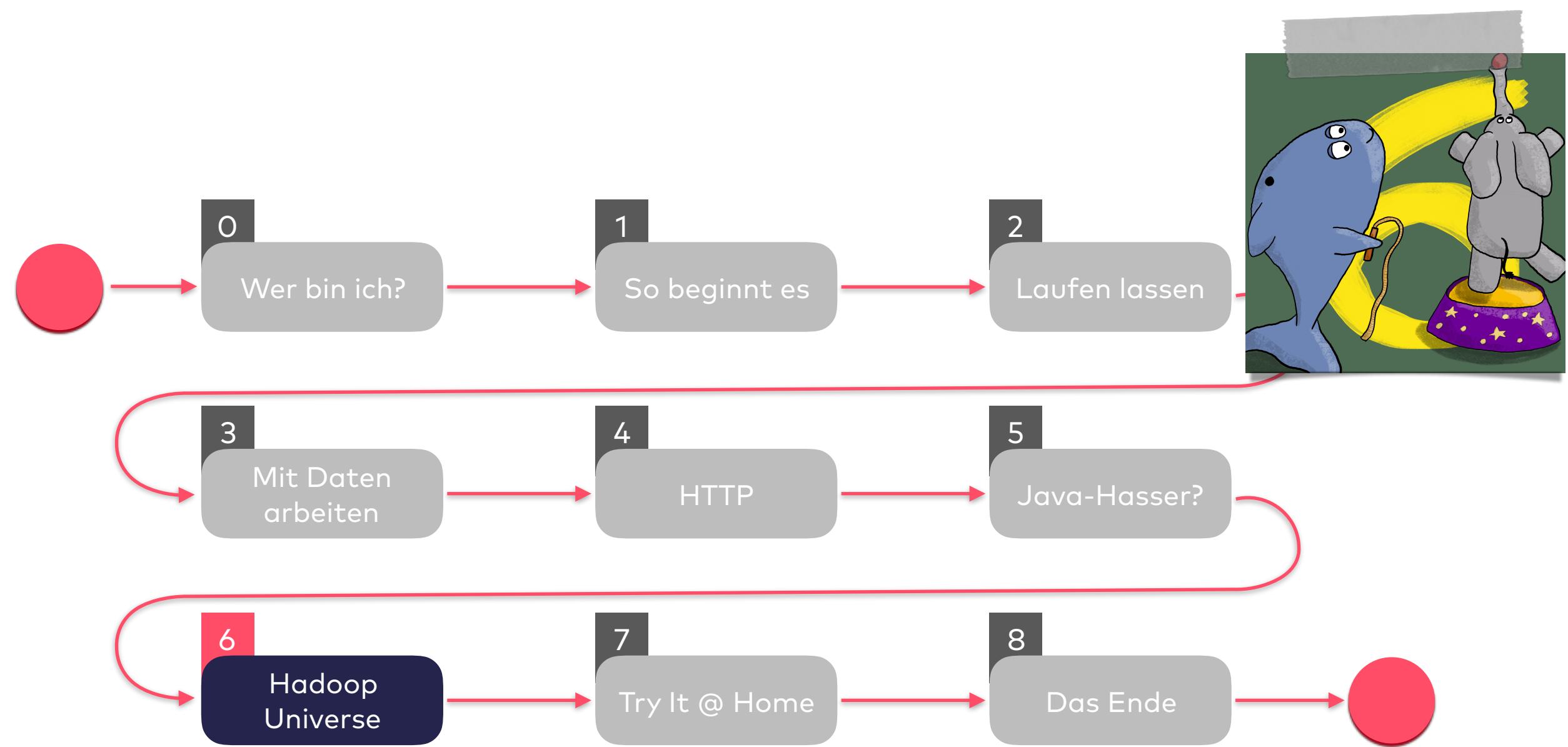
/user/root/output/darknet/category-count



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	0 B	1/10/2019, 5:25:10 PM	1	128 MB	_SUCCESS
-rw-r--r--	root	supergroup	1.86 KB	1/10/2019, 5:25:10 PM	1	128 MB	part-00000
Hadoop, 2014.							



# MapReduce ohne Java



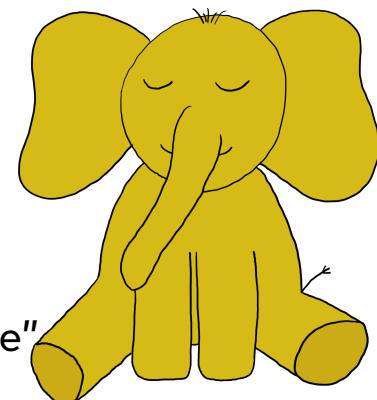


# Benamsung

"The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid's term."

(Doug Cutting)

Excerpt From: Tom White. "Hadoop: The Definitive Guide"





# Hadoop's Kern

Hadoop Common

MapReduce

HDFS

YARN

Ozone

Latest news

Ozone 0.3.0-alpha is released

2018 Nov 22



# HDFS





# YARN

**Processing Frameworks**

Pig

Hive

Crunch

**APPLICATION**

MapReduce

Spark

Tez

**COMPUTE**

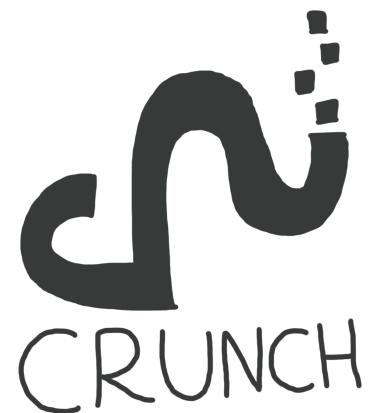
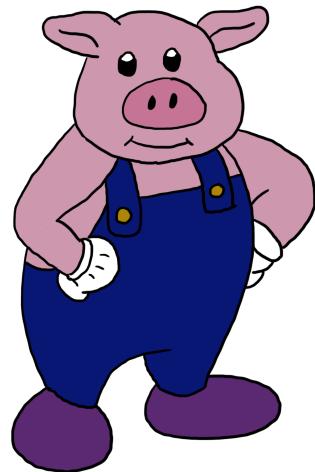
YARN

**STORE**

HDFS and HBase

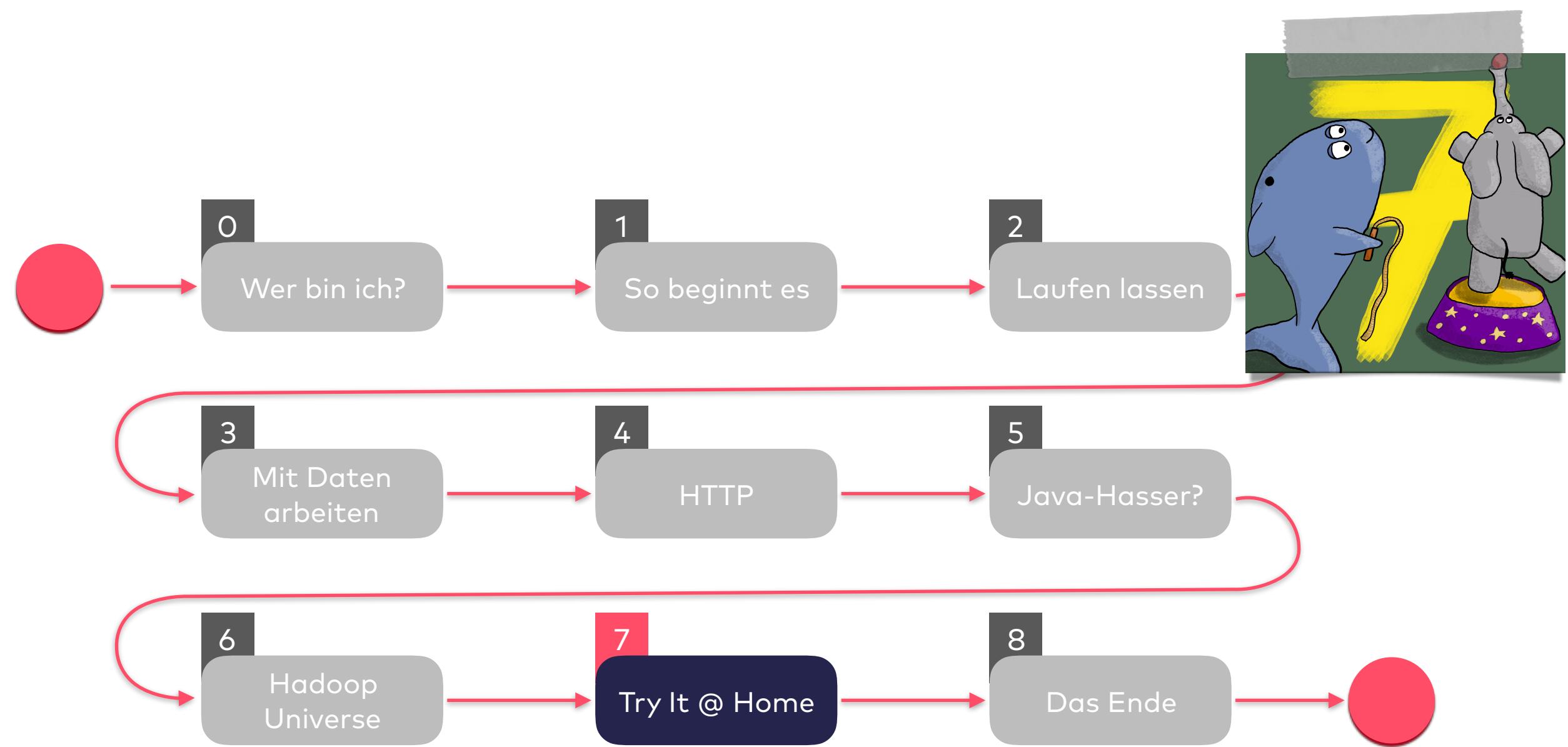


# Erwähnte Tools



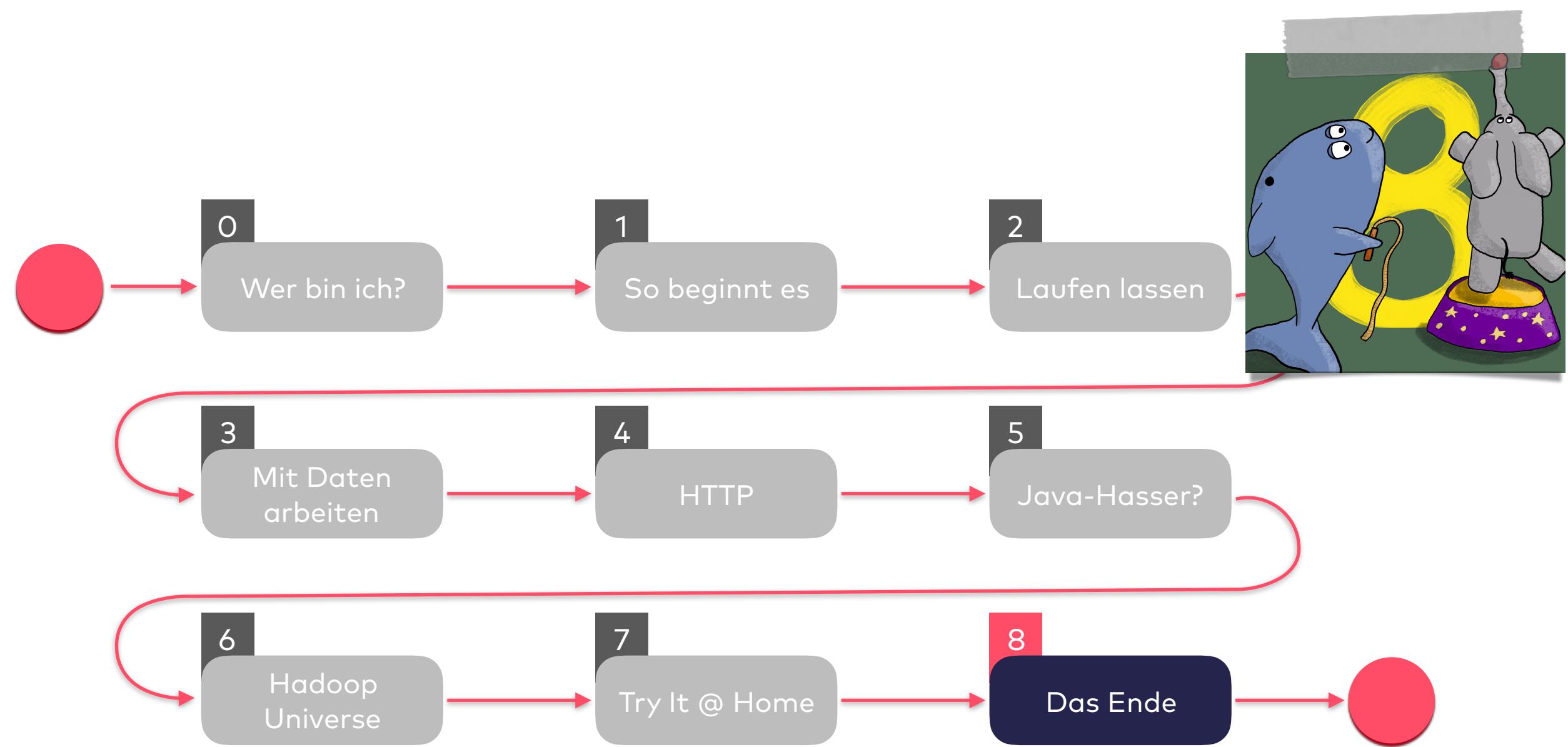


# **Ein bisschen Theorie**



# @ Home

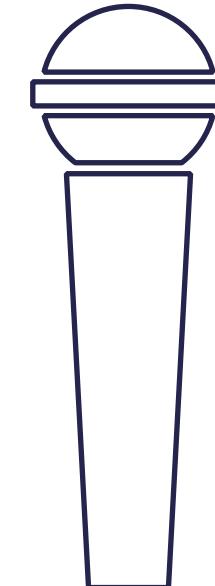
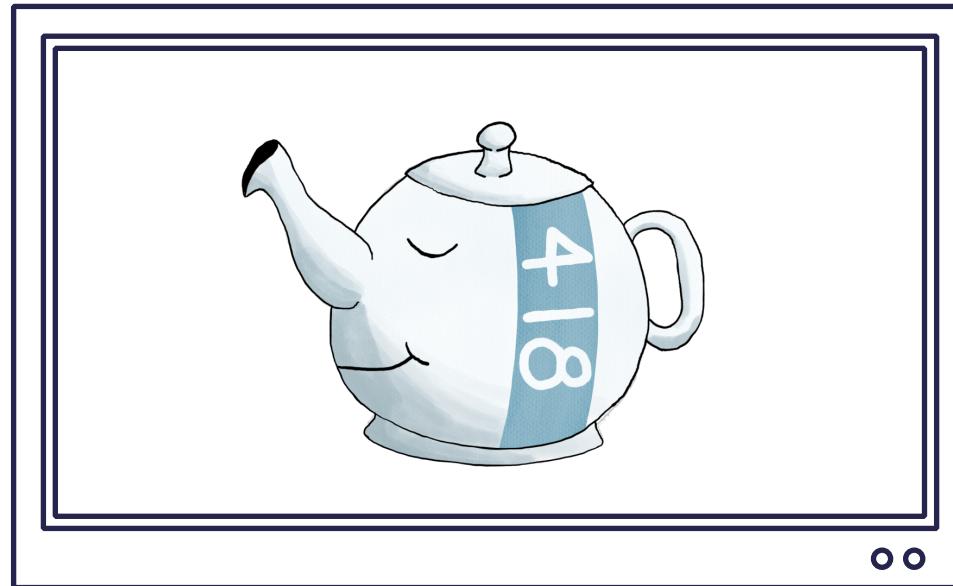
- Mehr Informationen über Hadoop
  - <http://hadoop.apache.org>
  - Tom White - "Hadoop - The Definitive Guide" (4th)
  - Alex Holmes - "Hadoop In Practice" (2nd)
- Mehr Informationen über diesen Vortrag
  - [www.teapot418.de](http://www.teapot418.de)
  - <https://github.com/Teapot-418/hadoop-taming-the-elephant>



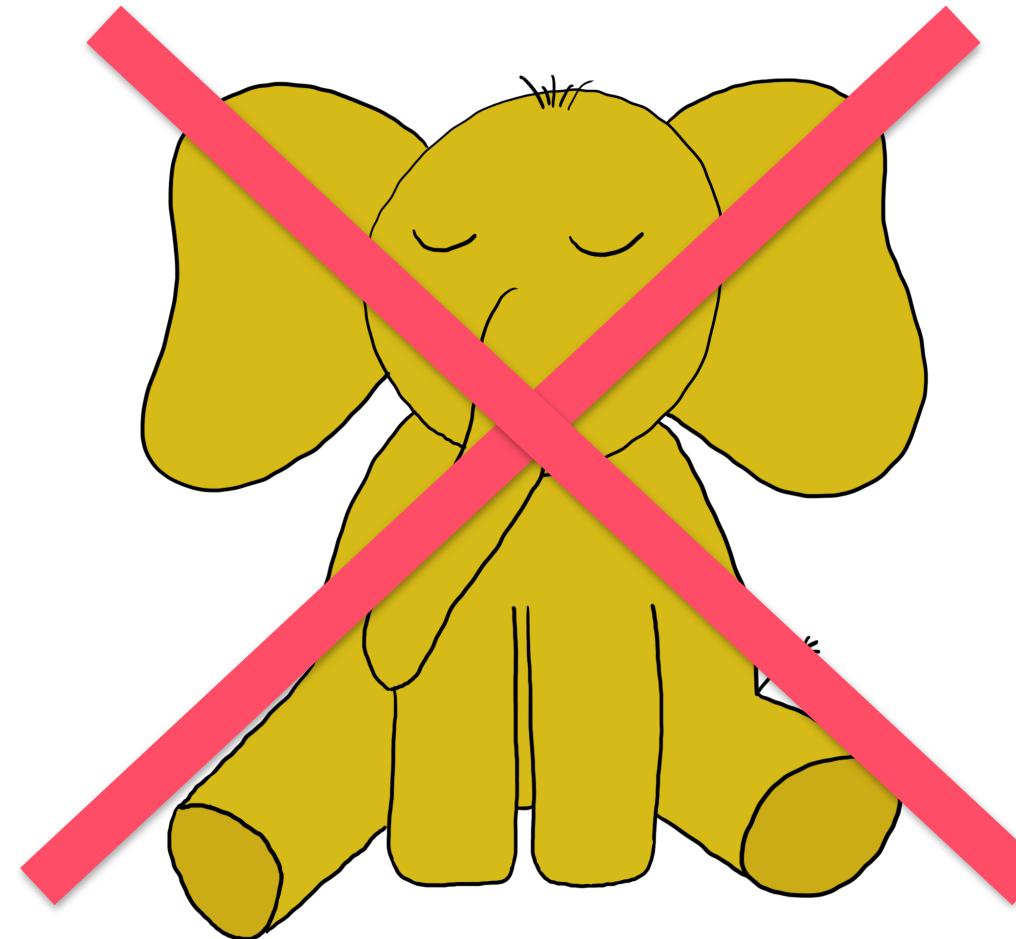
# Warum überhaupt?

**MEIN ZIEL**

Eis brechen und kurze Einführung  
geben



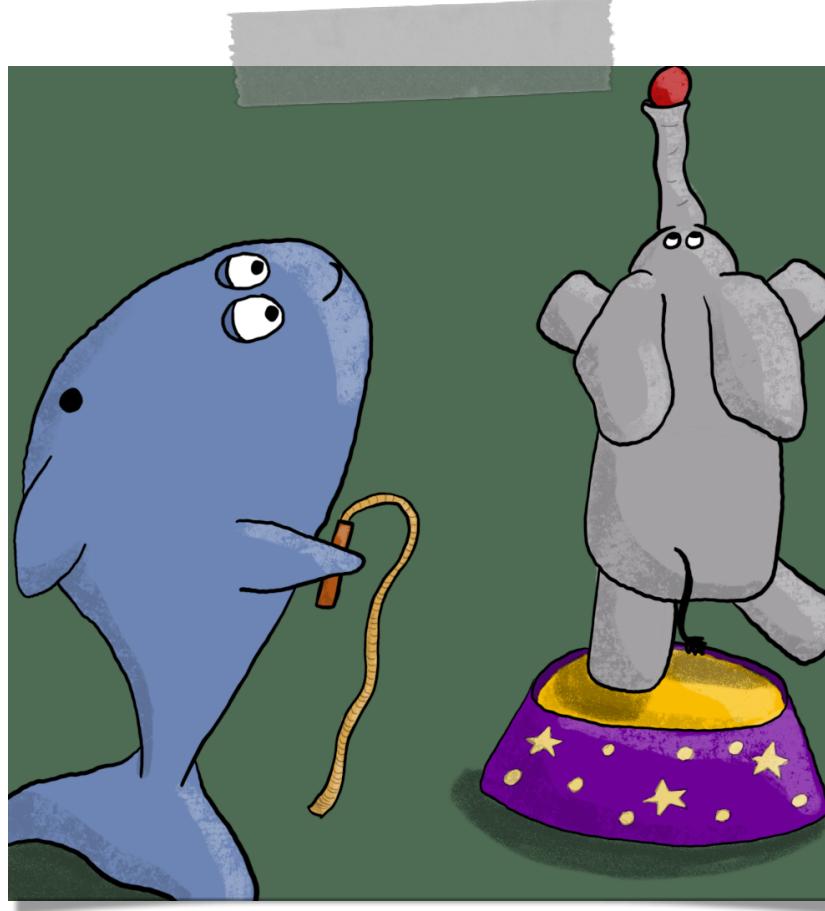
# Und nun?



# Vielen Dank!

Lisa Maria Moritz  
lisa.moritz@innoq.com

 Teapot4181



## innoQ Deutschland GmbH

Krischerstr. 100  
40789 Monheim am Rhein  
Germany  
+49 2173 3366-0

Ohlauer Str. 43  
10999 Berlin  
Germany  
+49 2173 3366-0

Ludwigstr. 180E  
63067 Offenbach  
Germany  
+49 2173 3366-0

Kreuzstr. 16  
80331 München  
Germany  
+49 2173 3366-0

## innoQ Schweiz GmbH

Gewerbestr. 11  
CH-6330 Cham  
Switzerland  
+41 41 743 0116

**INNOQ**  
[www.innoq.com](http://www.innoq.com)