Data science lifecycle with Apache Flink and Apache Zeppelin (incubating)

Moon moon@nflabs.com

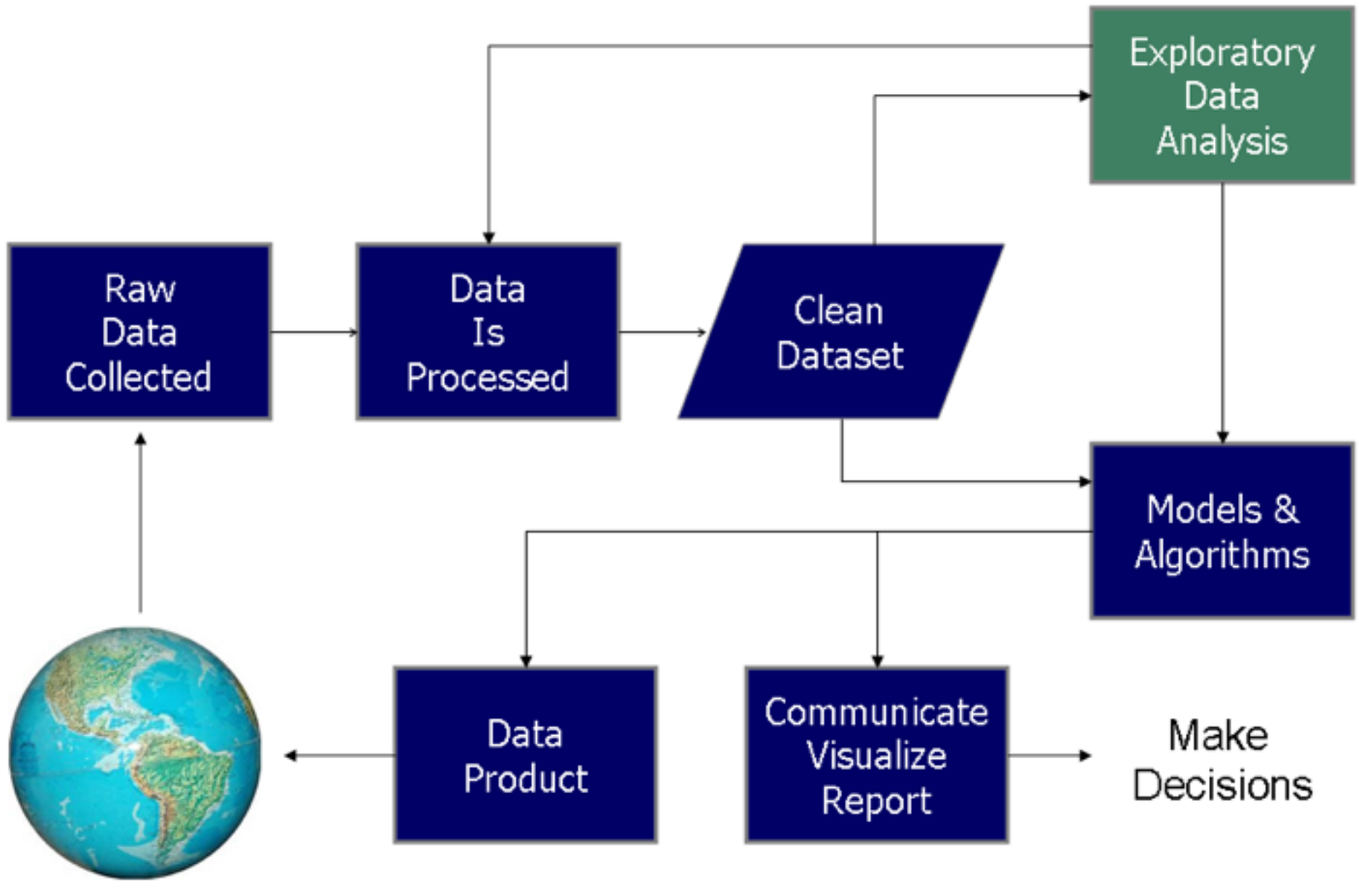
NFLabs www.nflabs.com

Content

1. **Data science lifecycle**
2. Zeppelin for data science
3. Zeppelin and Flink
4. Project Roadmap

Data science lifecycle

Data Science: process



https://en.wikipedia.org/wiki/Data_analysis

Data Science: tools

MLlib

Data Science: people

Engineer

Data Scientist

DevOps

Business

http://aarondavis.design/

Content

1. Data science lifecycle
2. **Zeppelin for data science**
3. Zeppelin and Flink
4. Project Roadmap

Zeppelin for data scientist

Project Timeline

http://zeppelin.incubator.apache.org

12.2012 Commercial Product for data analysis

10.2013 Open sourced a single feature

08.2014 Started getting adoption

12.2014 ASF Incubation

Hadoop Landscape

ML-base

MRQL

?

Cloudera-ML

Flink

Shark

Cassandra

accumulo

HBASE

Spark

Sqoop

HDFS

Map Reduce

HIVE

Commercial Product

12.2012

Zeppelin

10.2013

Zeppelin ZQL ZAN

New

Jobs

- Import bank marketing data

Import bank marketing data

History Latest

```
    housing STRING,
    loan STRING,
    contract STRING,
    day INT,
    month INT,
    duration INT,
    campain INT,
    pdays INT,
    previous INT,
    poutcome STRING,
    y STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY "59"
STORED AS TEXTFILE
;

LOAD DATA LOCAL INPATH '/tmp/bank-full.csv' OVERWRITE INTO TABLE bank;
```

Run

Schedule - **None** | Share | Delete

Last run Feb 5, 2014 10:25:53 AM, took 14.0 seconds

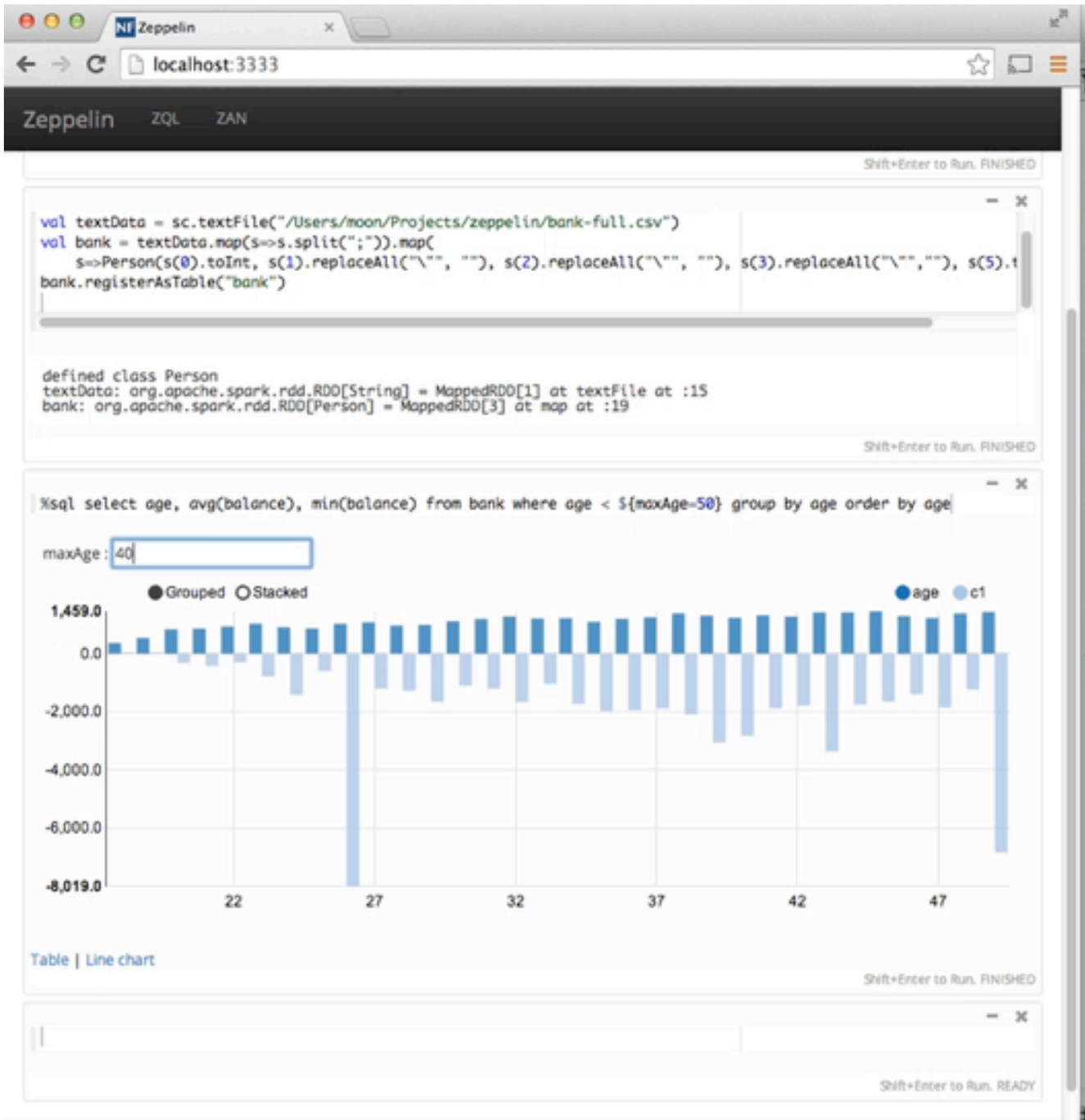
curl -s http://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank.zip -o

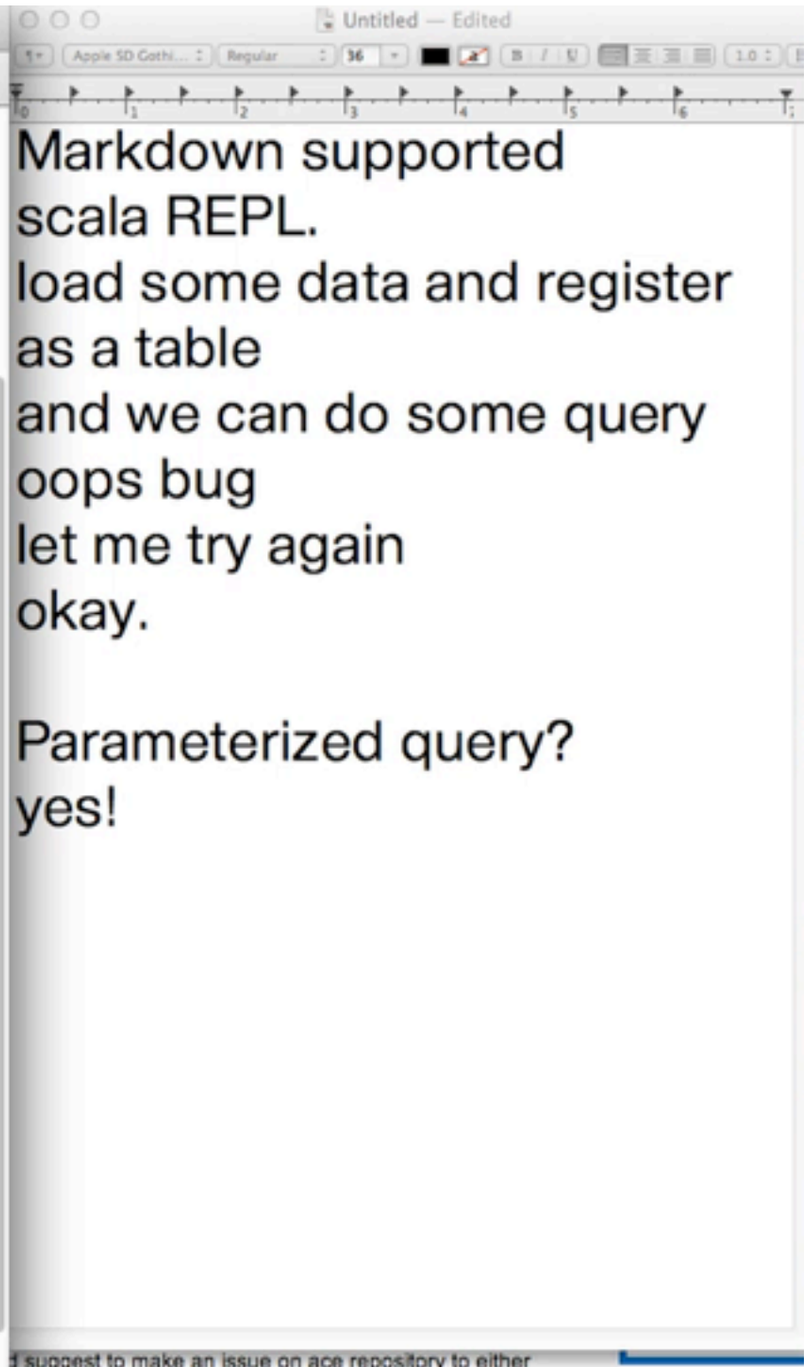
Zeppelin

10.2013

Zeppelin

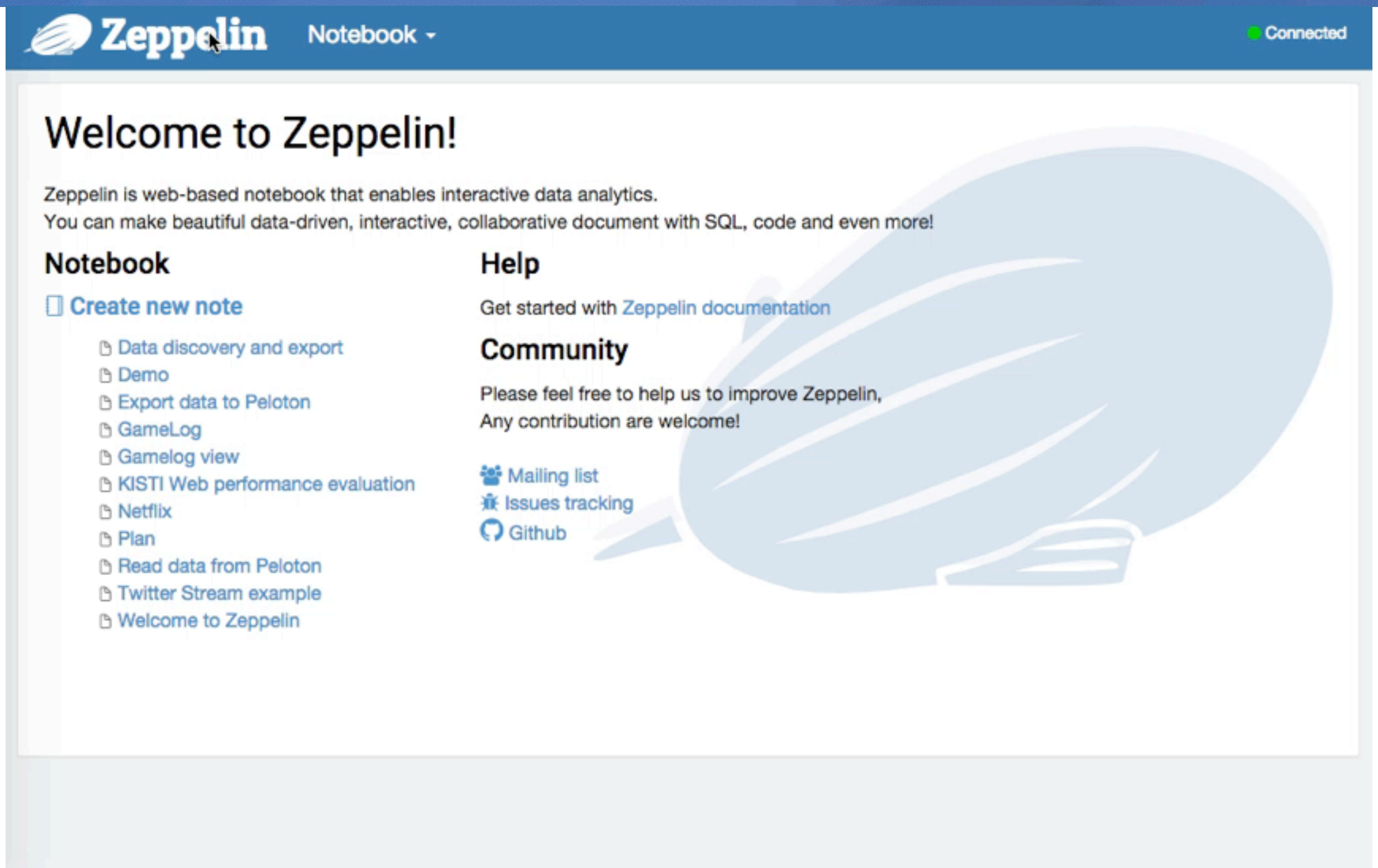
08.2014





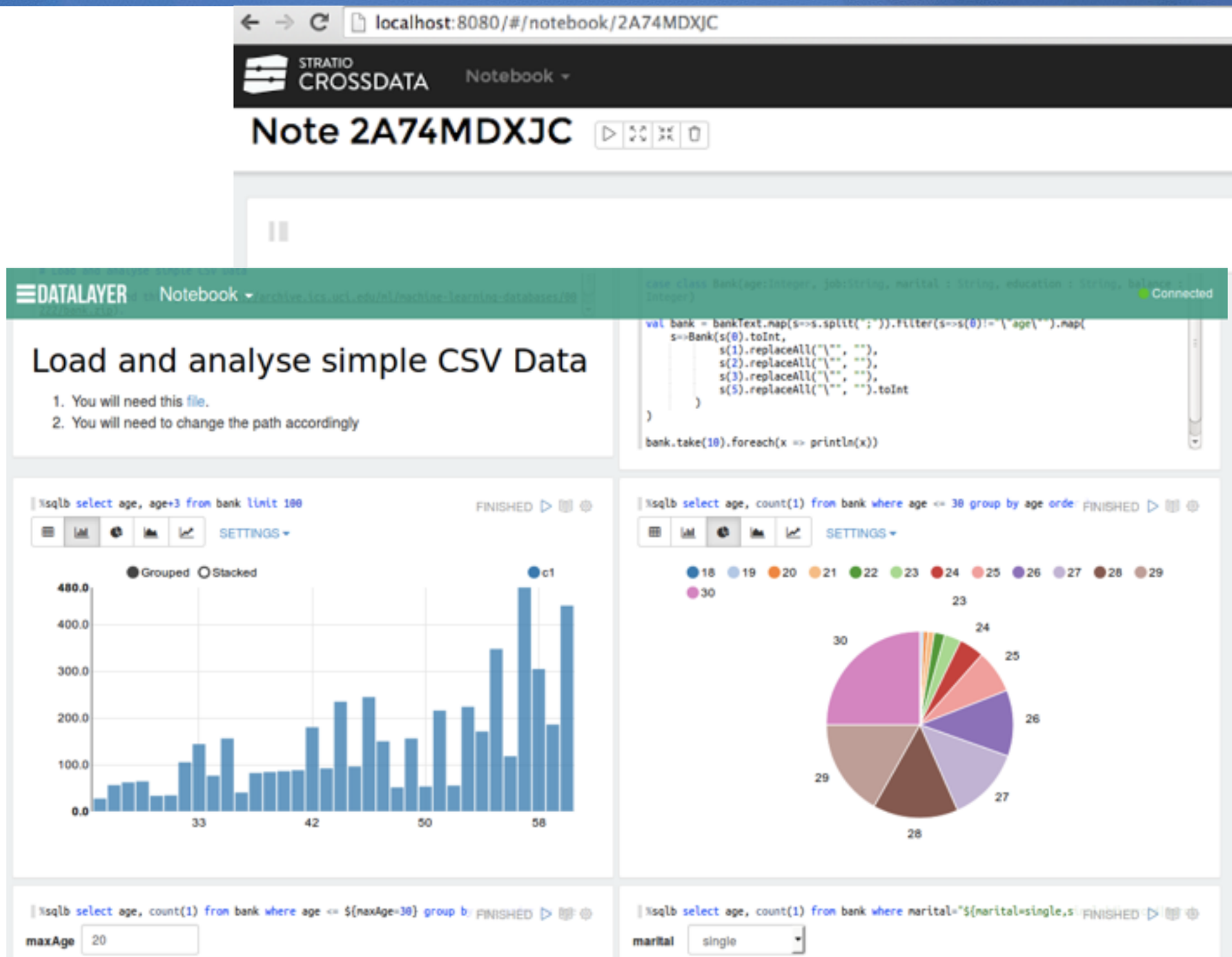
Zeppelin

08.2014



Third-party Products

10.2014



Apache Incubation Proposal

11.2014

Abstract

Zeppelin is a collaborative data analytics and visualization tool for distributed, general-purpose data processing systems such as Apache Spark, Apache Flink, etc.

Proposal

Zeppelin is a modern web-based tool for the data scientists to collaborate over large-scale data exploration and visualization projects. It is a notebook style interpreter that enable collaborative analysis sessions sharing between users. Zeppelin is independent of the execution framework itself. Current version runs on top of Apache Spark but it has pluggable interpreter APIs to support other data processing systems. More execution frameworks could be added at a later date i.e Apache Flink, Crunch as well as SQL-like backends such as Hive, Tajo, MRQL.

We have a strong preference for the project to be called Zeppelin. In case that may not be feasible, alternative names could be: “Mir”, “Yuga” or “Sora”.

Background

Large scale data analysis workflow includes multiple steps like data acquisition, pre-processing, visualization, etc and may include inter-operation of multiple different tools and technologies. With the widespread of the open source general-purpose data processing systems like Spark there is a lack of open source, modern user-friendly tools that combine strengths of interpreted language for data analysis with new in-browser visualization libraries and collaborative capabilities.

Zeppelin initially started as a GUI tool for diverse set of SQL-over-Hadoop systems like Hive, Presto, Shark, etc. It was open source since its inception in Sep 2013. Later, it became clear that there was a need for a greater web-based tool for data scientists to collaborate on data exploration over the large-scale projects, not limited to SQL. So Zeppelin integrated full support of Apache Spark while adding a collaborative environment with the ability to run and share interpreter sessions in-browser

Rationale

Acceptance by Incubator

23.12.2014

Current Status

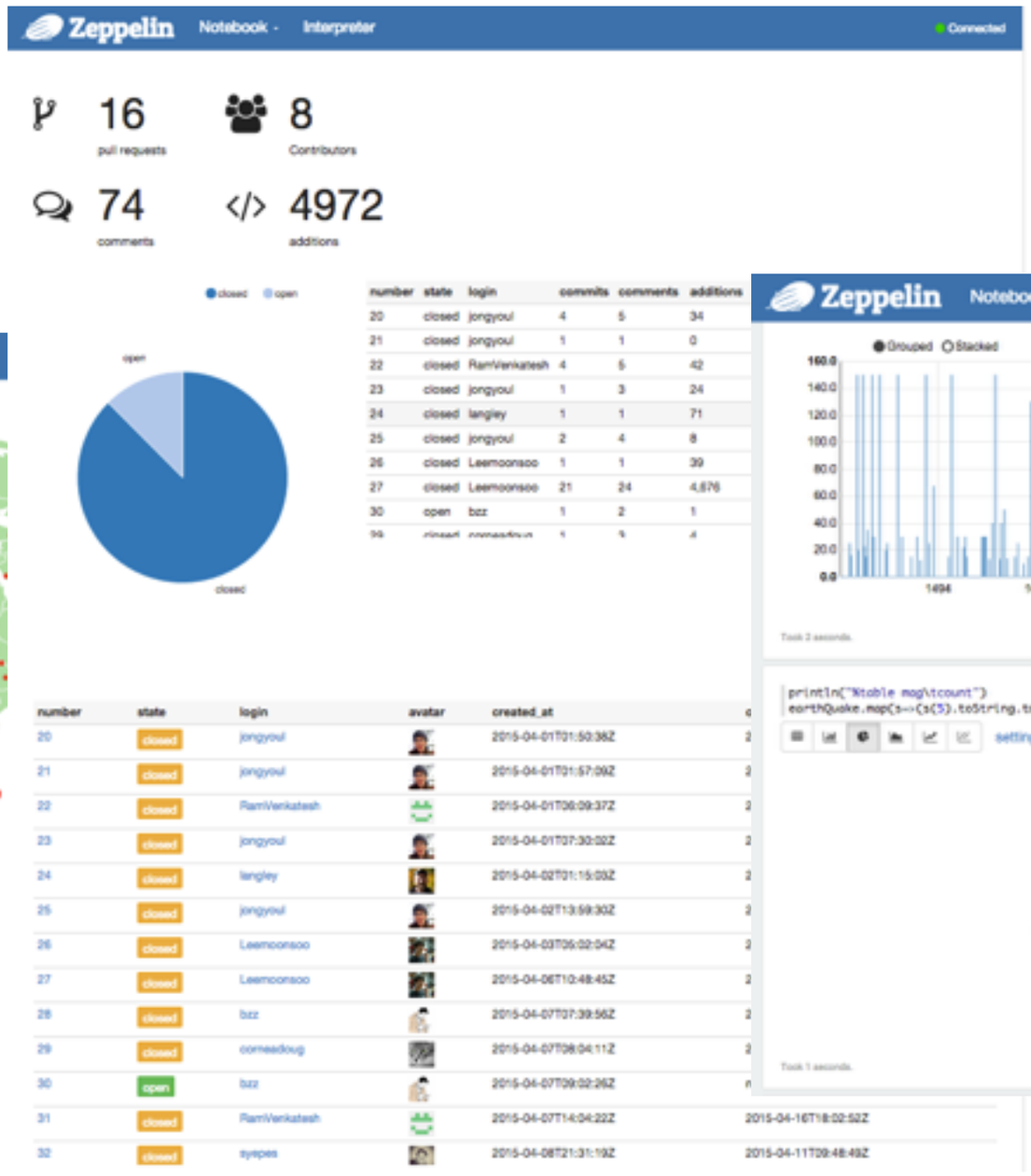
68 Contributors worldwide

722 Stars on GH

300/900 Emails at users/dev @i.a.o

1 Release

Interactive Notebooks

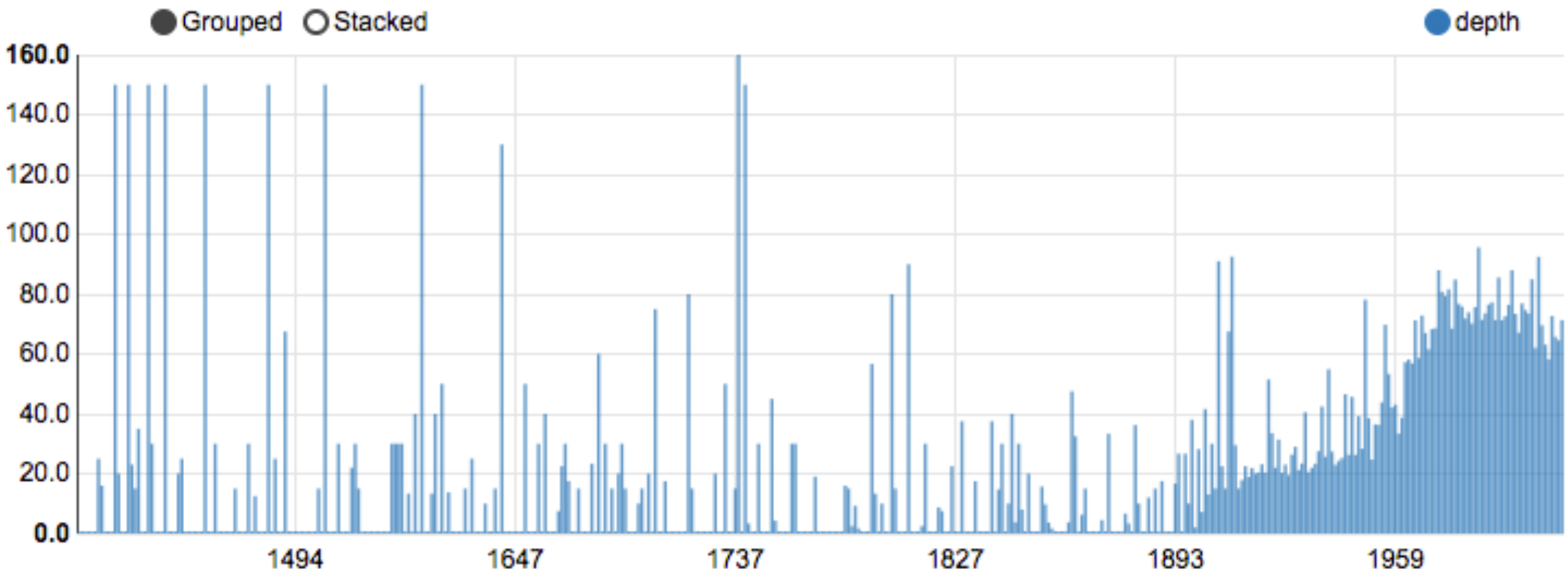


Interactive Visualization

```
%sql select substring(date, 0, 4) as dt, avg(depth) as depth from eq group by substri
```

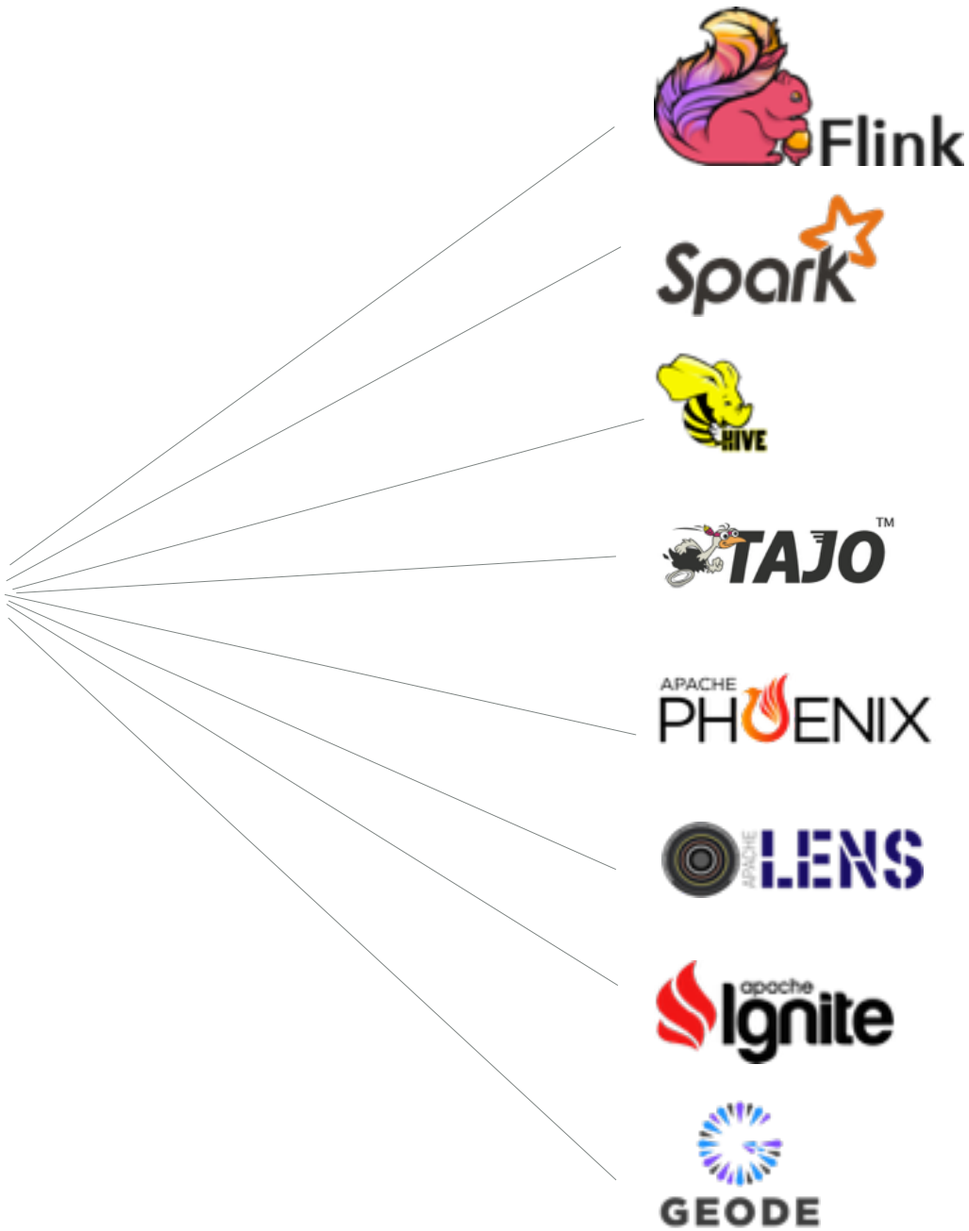
FINISHED

settings



Took 2 seconds.

Multiple Backends



Zeppelin & Friends

Collaboration/Sharing

ZeppelinHub

Packaging & Deployment

Hortonworks

Ambari

Z-Manager

Zeppelin + Full stack on a cloud

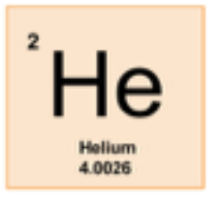
Microsoft Azure

SEQUENCEIQ

Qubole

DATALAYER

Packages



Backend Integration

Flink

Spark

HIVE

...

apache Ignite

LENS

TAJO

Online Viewer

https://www.zeppelinhub.com/viewer/

Zeppelin Viewer

Paste a link to a notebook

This viewer currently supports direct urls or notebooks hosted on github and dropbox. We will support other methods in the very near future.

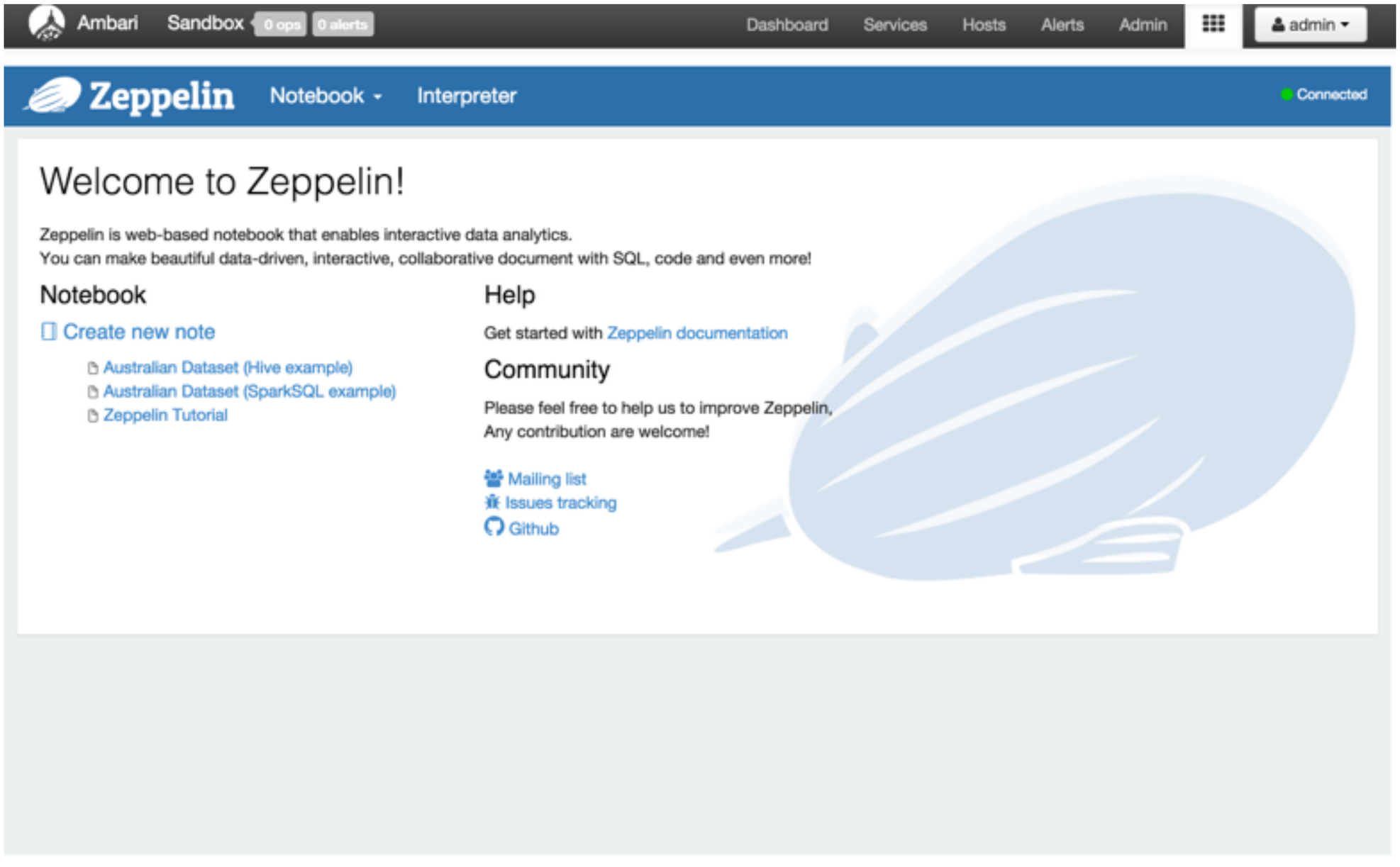
http(s)://your-note-url.json

view

for example, intro to Zeppelin

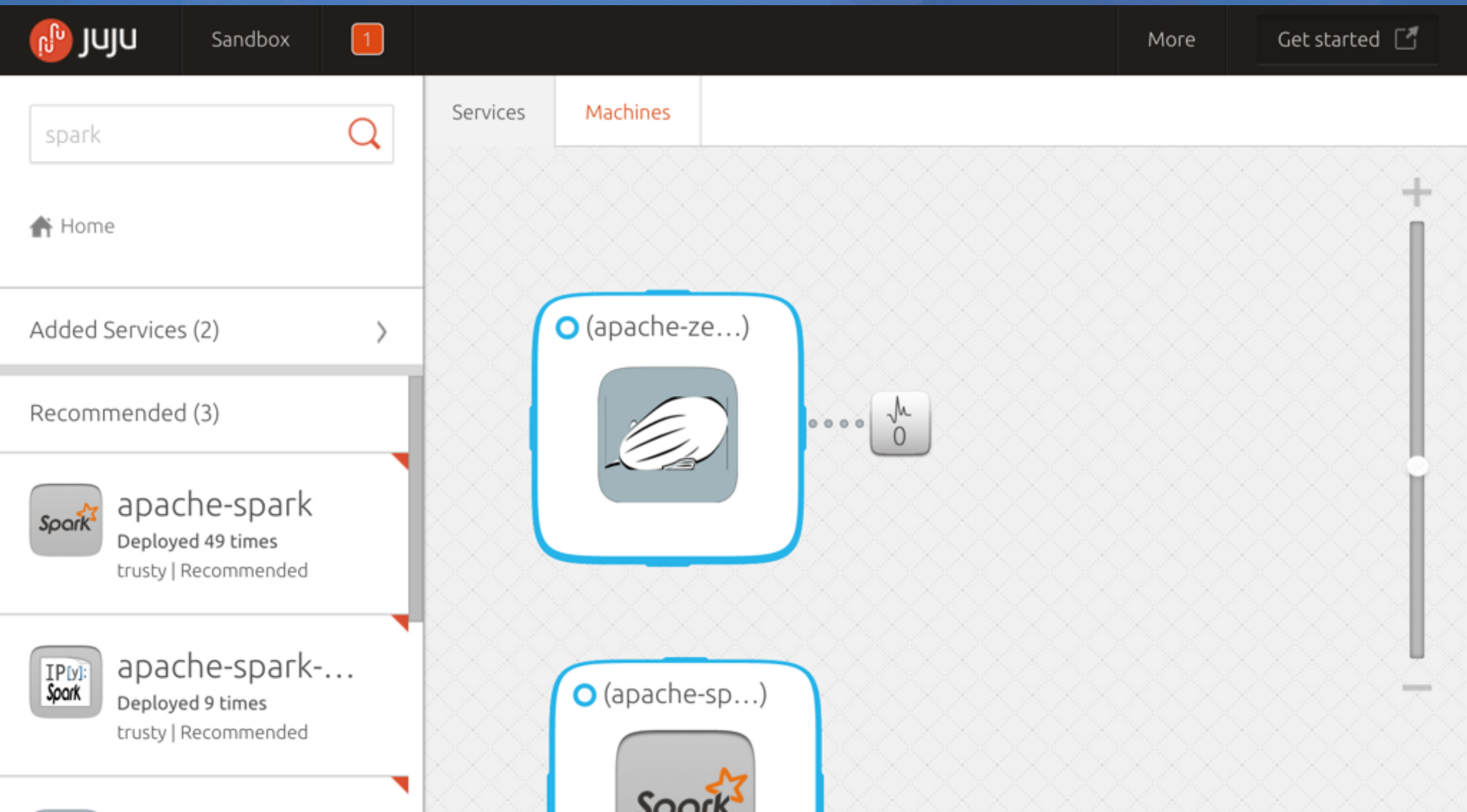
We heard you and we want to thank you. As a gratitude to the wonderful community of Apache Zeppelin (incubating)—both users and contributors—we would like to offer this Zeppelin Viewer service. So go ahead, share your Zeppelin notebooks with anyone, anywhere. And please, share your thoughts and feedback as well. Thank you to the greatest open source community in the world

Deployment



https://github.com/hortonworks-gallery/ambari-zeppelin-service

Deployment


juju
Sandbox
1
More
Get started
spark
Services
Machines
Home
Added Services (2)
Recommended (3)
apache-spark
Deployed 49 times
trusty | Recommended
apache-spark-…
Deployed 9 times
trusty | Recommended
(apache-ze…)
0
(apache-sp…)

As a Service

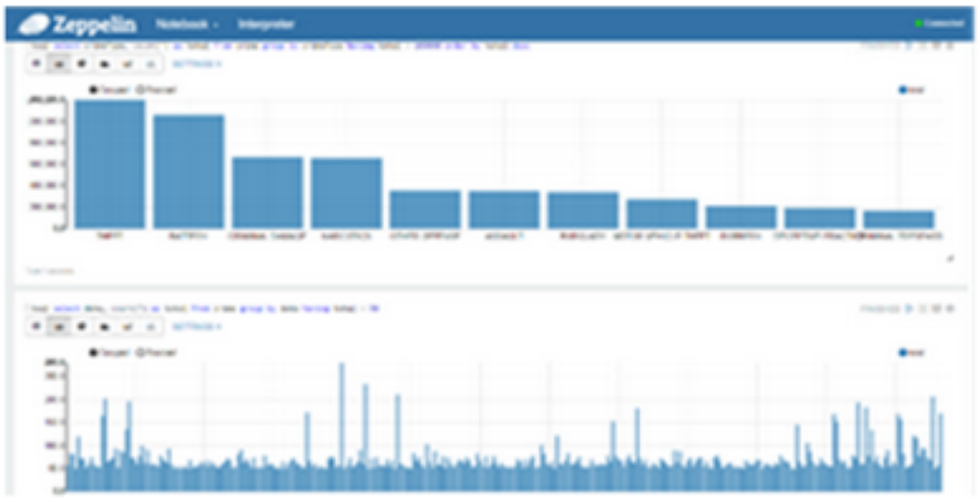
Microsoft Azure

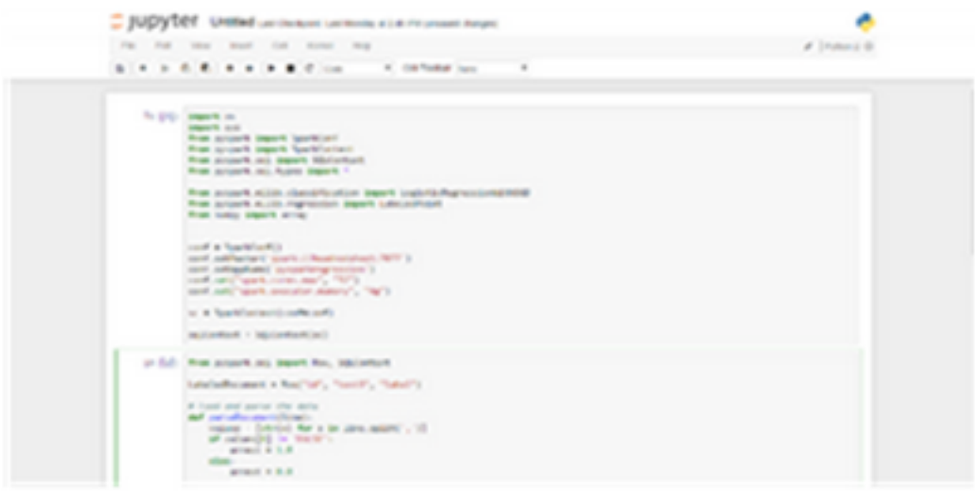
SALES 1-800-867-1389 | MY ACCOUNT | PORTAL

Why Azure | Products | Documentation | Pricing | Downloads | Partners | Blog | Community | Support

Using open-source notebooks

Notebooks can also be used to visualize data running in Spark for Azure HDInsight. Notebooks are open source tools that give data scientists an ability to combine live code, statistical equations, narrative text, and visualizations to tell a story of their data. We have enabled popular Jupyter (IPython) and Zeppelin notebooks to run on Spark for Azure HDInsight. Jupyter will come with standard IPython visual libraries and ideal for those who code using Python. Zeppelin is ideal for those who write in Scala while also supporting Spark SQL and Markdown.





Before

ML-base

MRQL

?

Cloudera-ML

Flink

Shark

Sqoop

HDFS

MapReduce

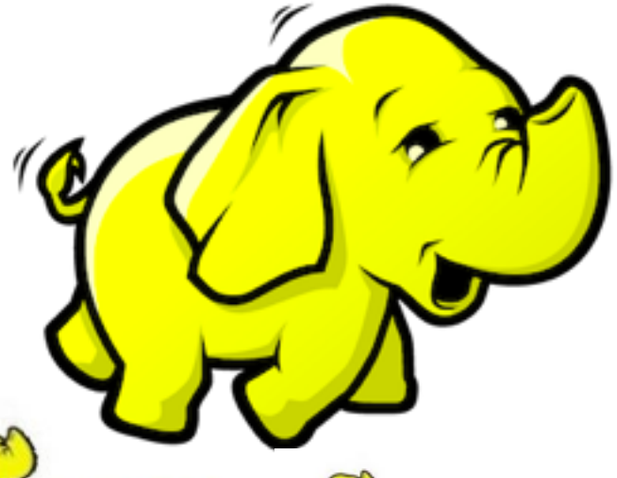
After

ML-base

MRQL

Cloudera-ML

Shark



Content

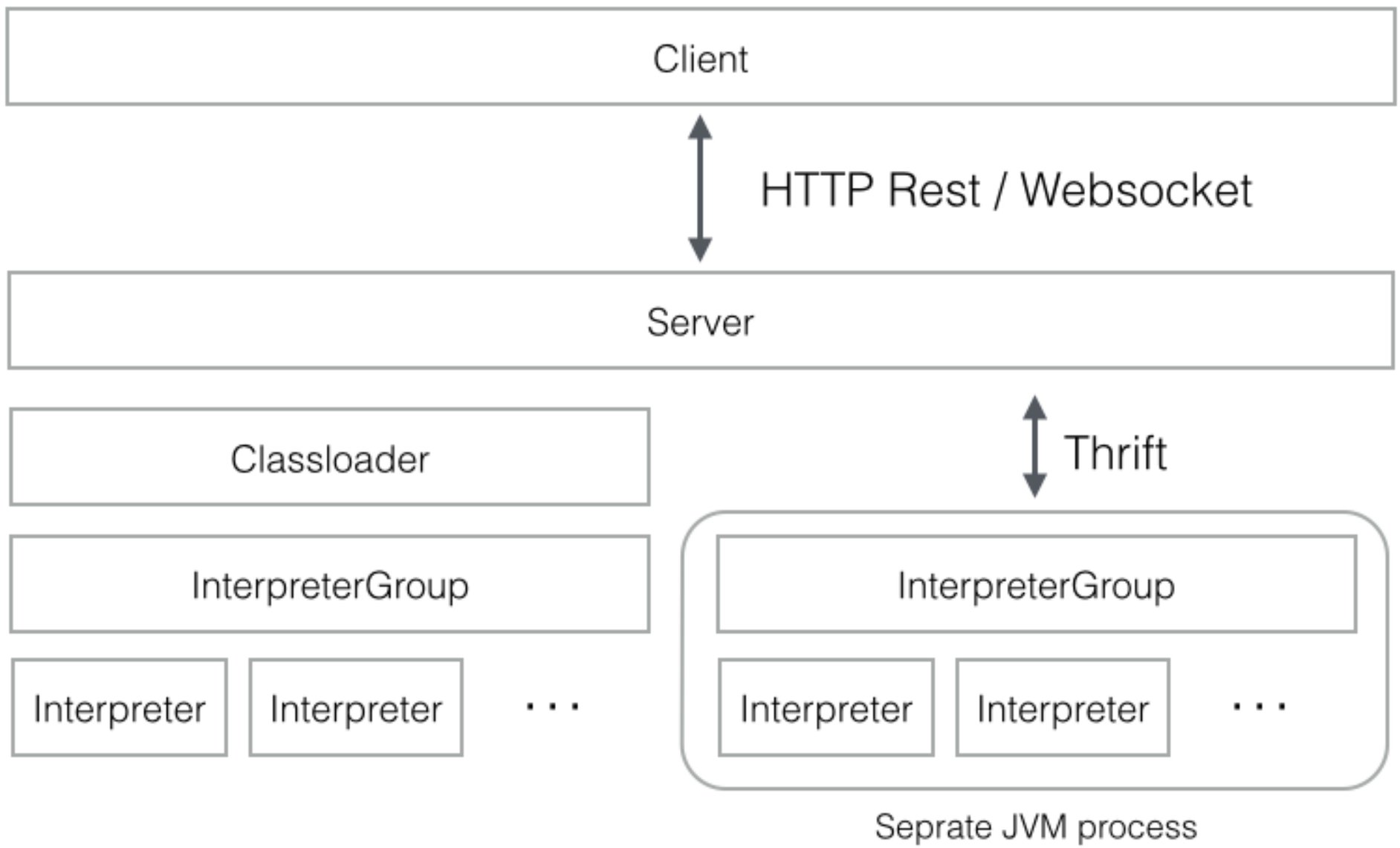
1. Data science lifecycle
2. Zeppelin for data science
3. **Zeppelin and Flink**
4. Project Roadmap

Flink integration

Integrated through Interpreter

Data processing system abstraction in Zeppelin

Interpreter



http://zeppelin.incubator.apache.org/docs/development/writingzeppelininterpreter.html

Writing an Interpreter

Must have

```
public abstract void open();

public abstract void close();

public abstract InterpreterResult interpret(String st, InterpreterContext context);
```

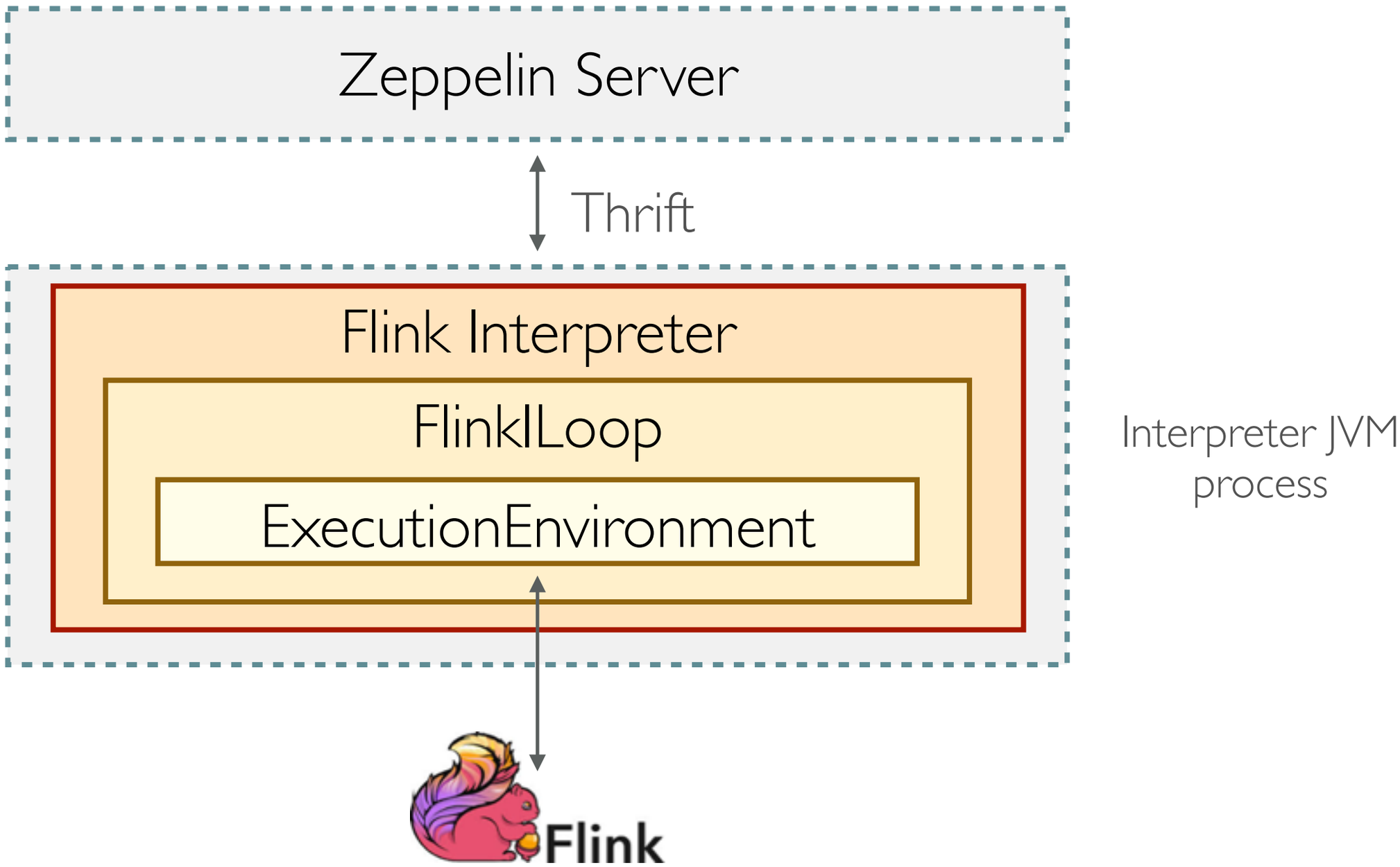
Good to have

```
public abstract void cancel(InterpreterContext context);

public abstract int getProgress(InterpreterContext context);

public abstract List<String> completion(String buf, int cursor);
```

Advanced

```
public abstract FormType getFormType();

public Scheduler getScheduler();
```

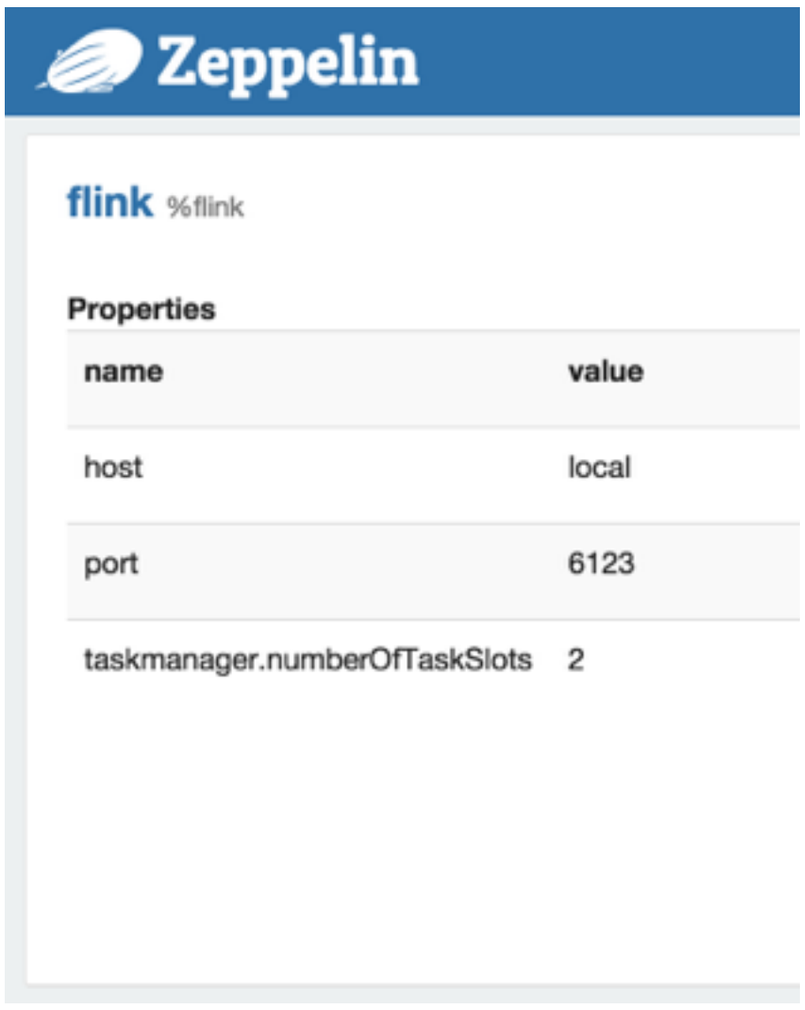
Flink Interpreter



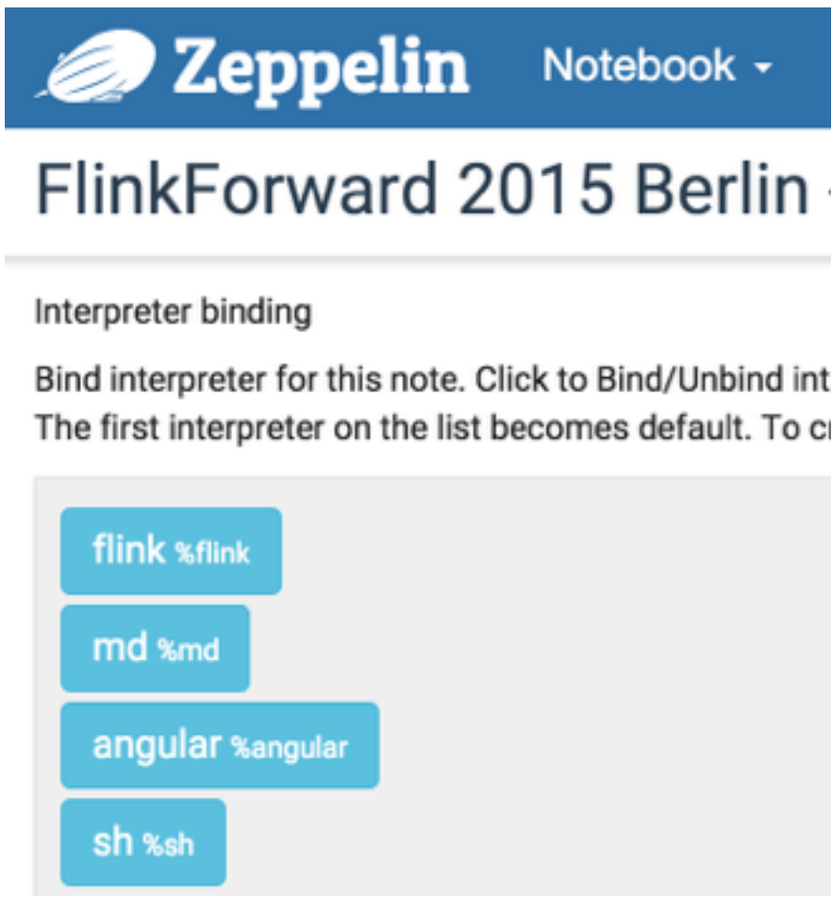
https://github.com/apache/incubator-zeppelin/blob/master/flink/src/main/java/org/apache/zeppelin/flink/FlinkInterpreter.java

Using interpreter

Configure



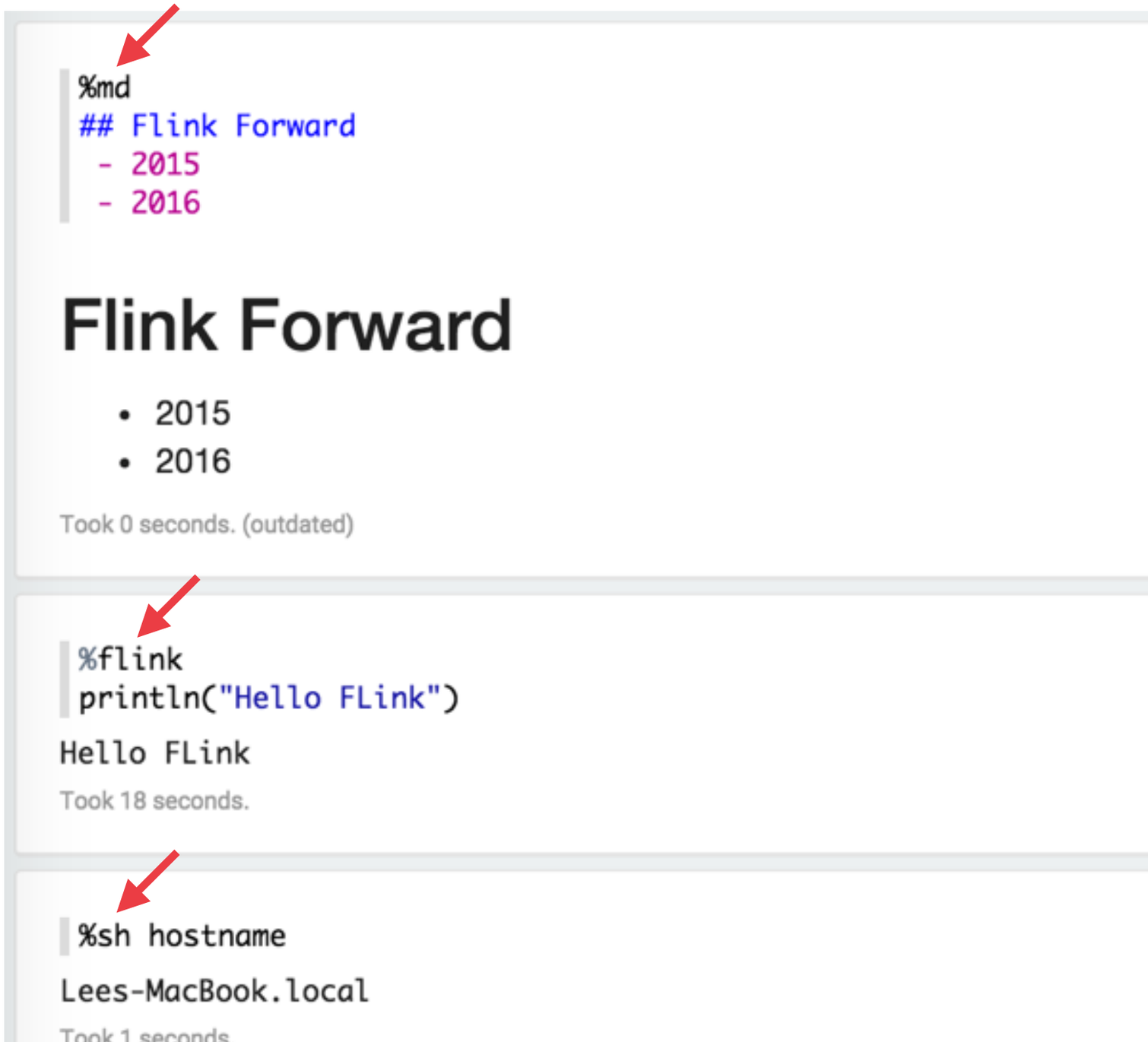
Bind



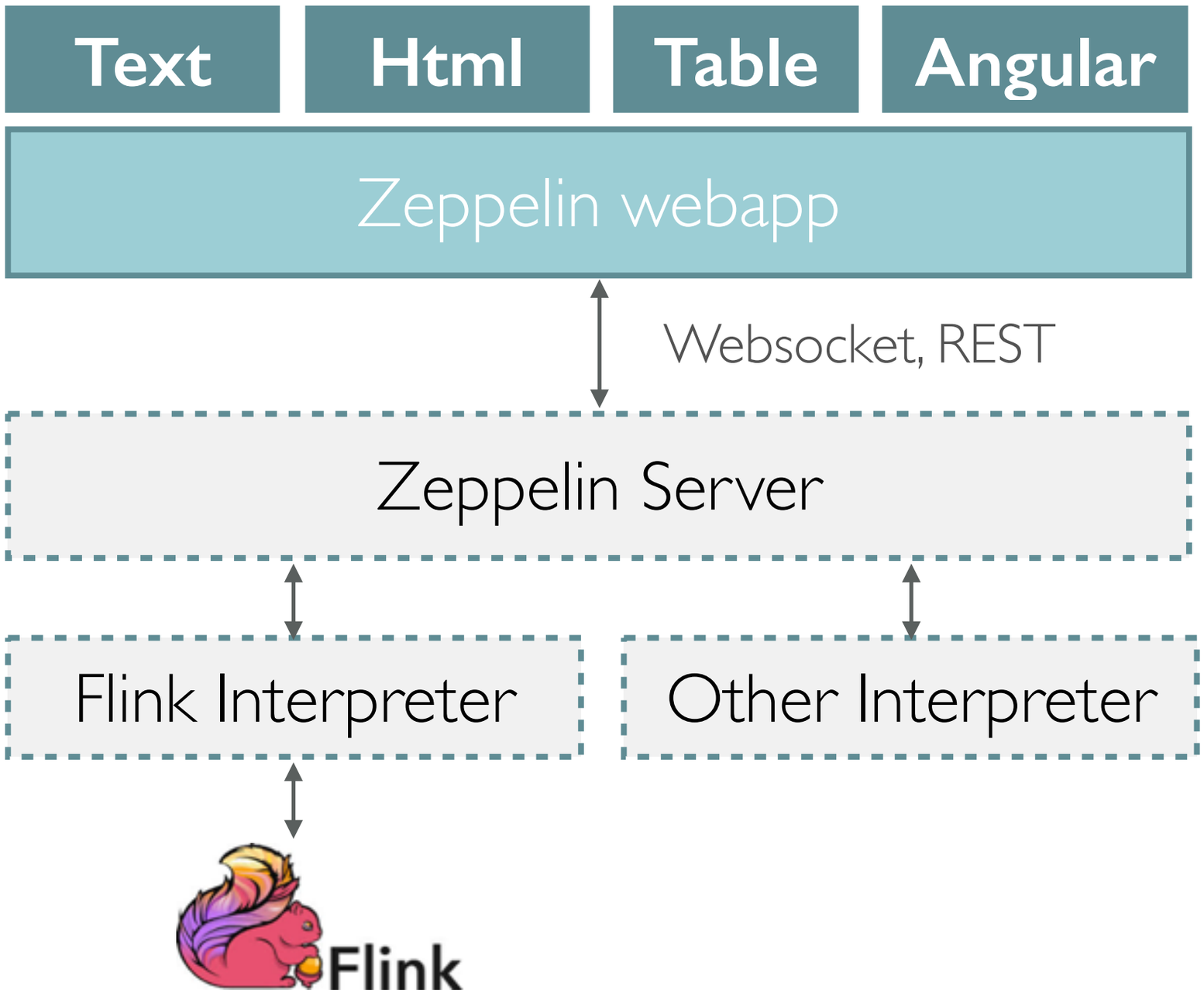
use

Using interpreter

Use different interpreters in the same notebook



Display System



Display System

Select display system through output

```
println("Default display system is text")
```

```
Default display system is text
```

Took 1 seconds.

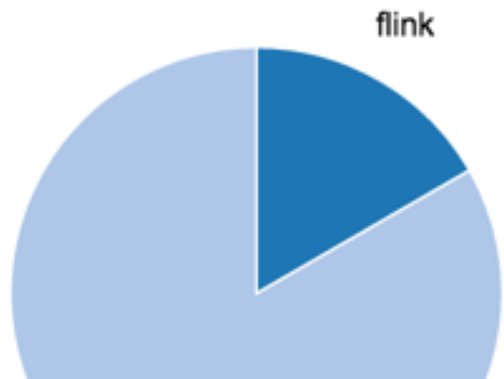
```
println("%html <h4>html display system</h4>")
```

html display system

Took 1 seconds.

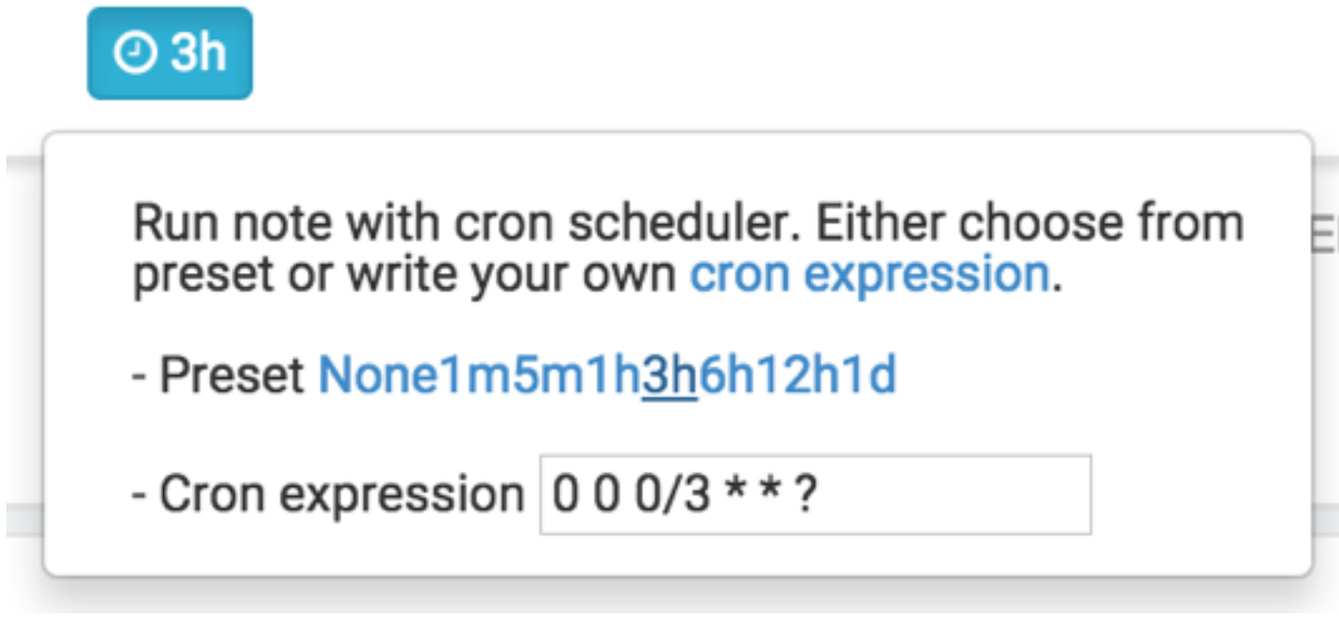
```
println(s"""%table key\tvalue
flink\t20
forward\t100""")
```

settings



Built in scheduler

Built-in scheduler runs your notebook with cron expression.



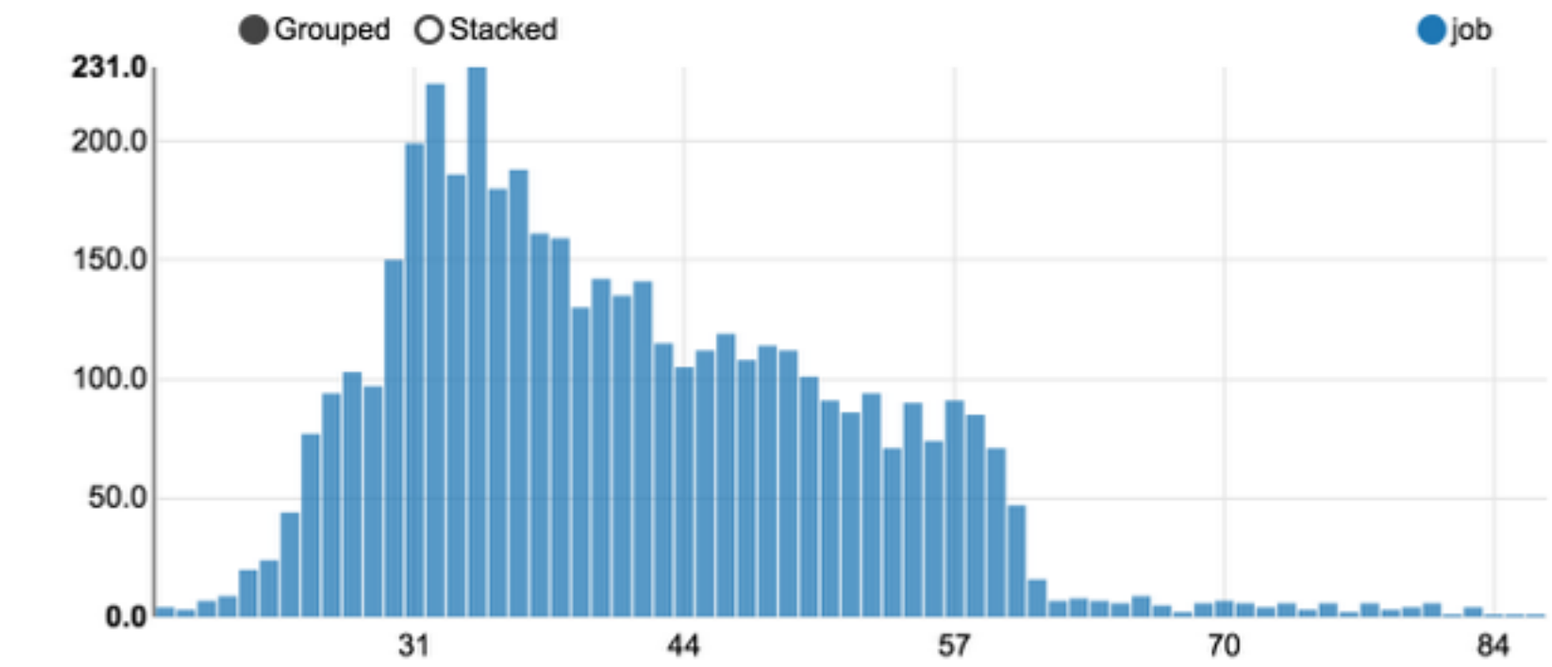
Flexible layout

Flexible layout

```
println("%table " + Array("age", "job", "marital", "education", "bal
("\t"))
bank.collect.foreach(b=>
  println(Array(b.age, b.job, b.marital, b.education, b.balance).mkString("\t"))
```

FINISHED

settings

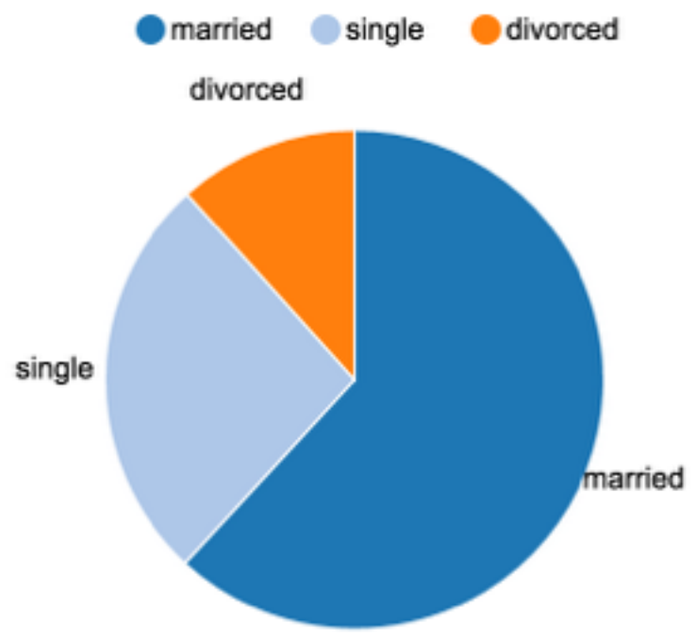


Took 5 seconds. (outdated)

```
println("%table " + A
"marital", "education", "balance"
).mkString("\t"))
bank.collect.foreach(b=>
```

FINISHED

settings



Took 2 seconds. (outdated)

DEMO

Content

1. Data science lifecycle
2. Zeppelin for data science
3. Zeppelin and Flink
4. **Project Roadmap**

Flink Integration

- ZeppelinContext : Access to Zeppelin provided features
- - Dynamic form
- - Angular display system

- Dependency loading

- Auto completion
- Cancel
- Get progress information

Thank you

Q & A

http://zeppelin.incubator.apache.org/

Moon
moon@nflabs.com

NFLabs
www.nflabs.com

Project roadmap

Multi-tenancy

Two approaches

1. Implement authentication, ACL inside of Zeppelin

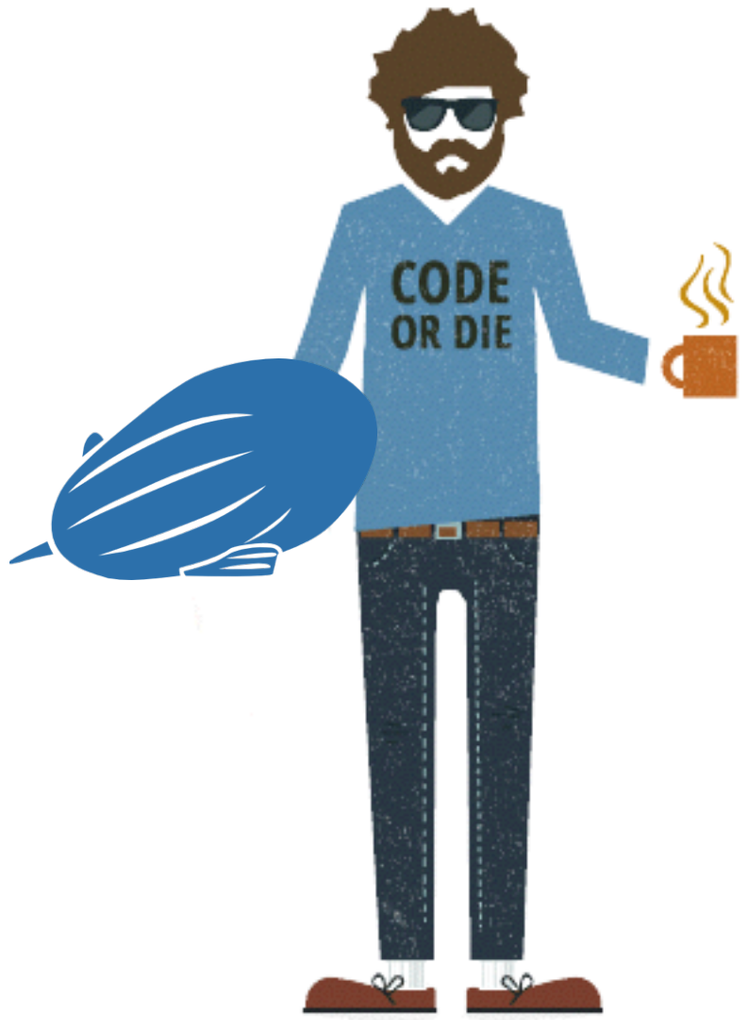
   https://github.com/apache/incubator-zeppelin/pull/53

2. Run Zeppelin on top of Docker

   http://github.com/NFLabs/z-manager

Zeppelin for organizations

An Engineer

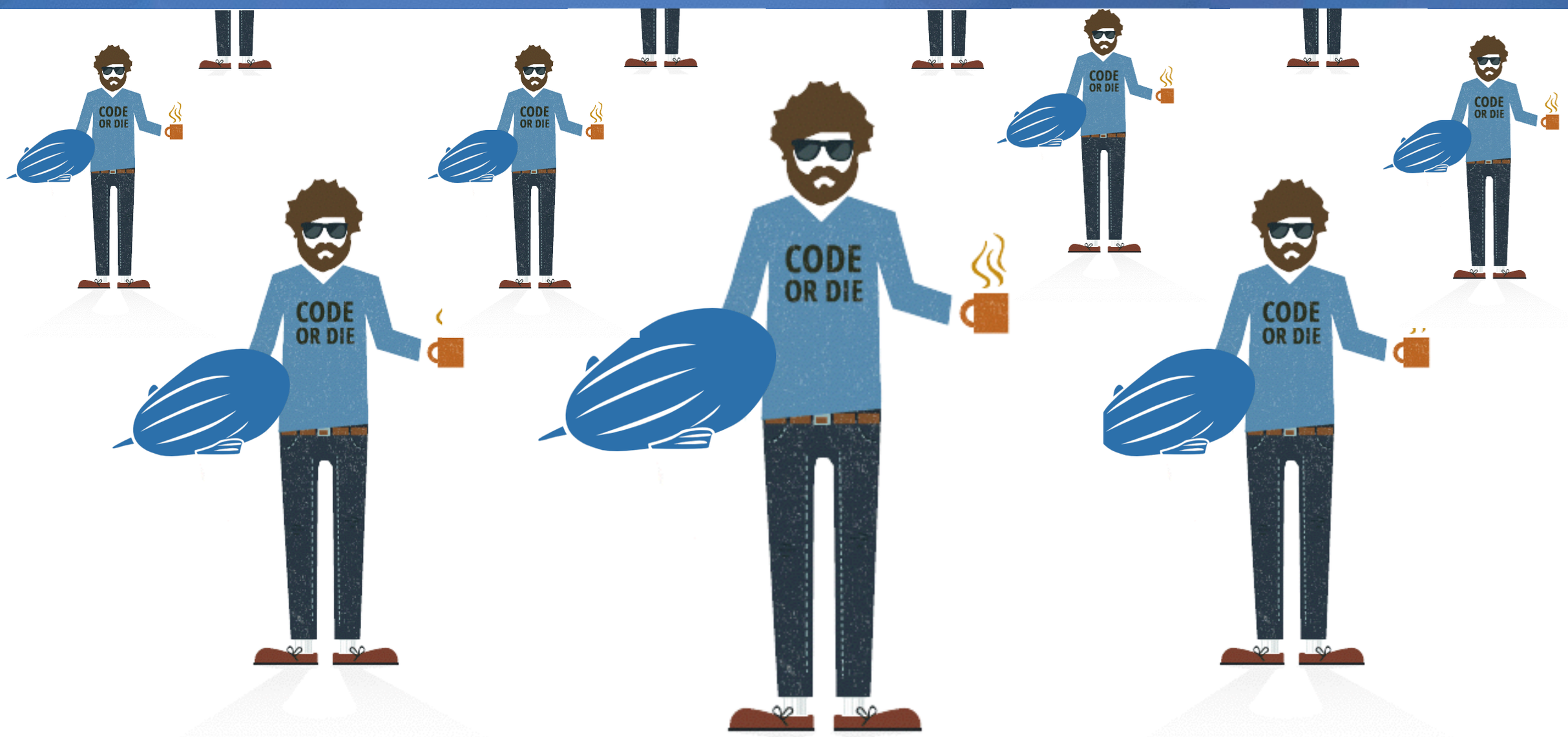


engineer by http://aarondavis.design/

A Team

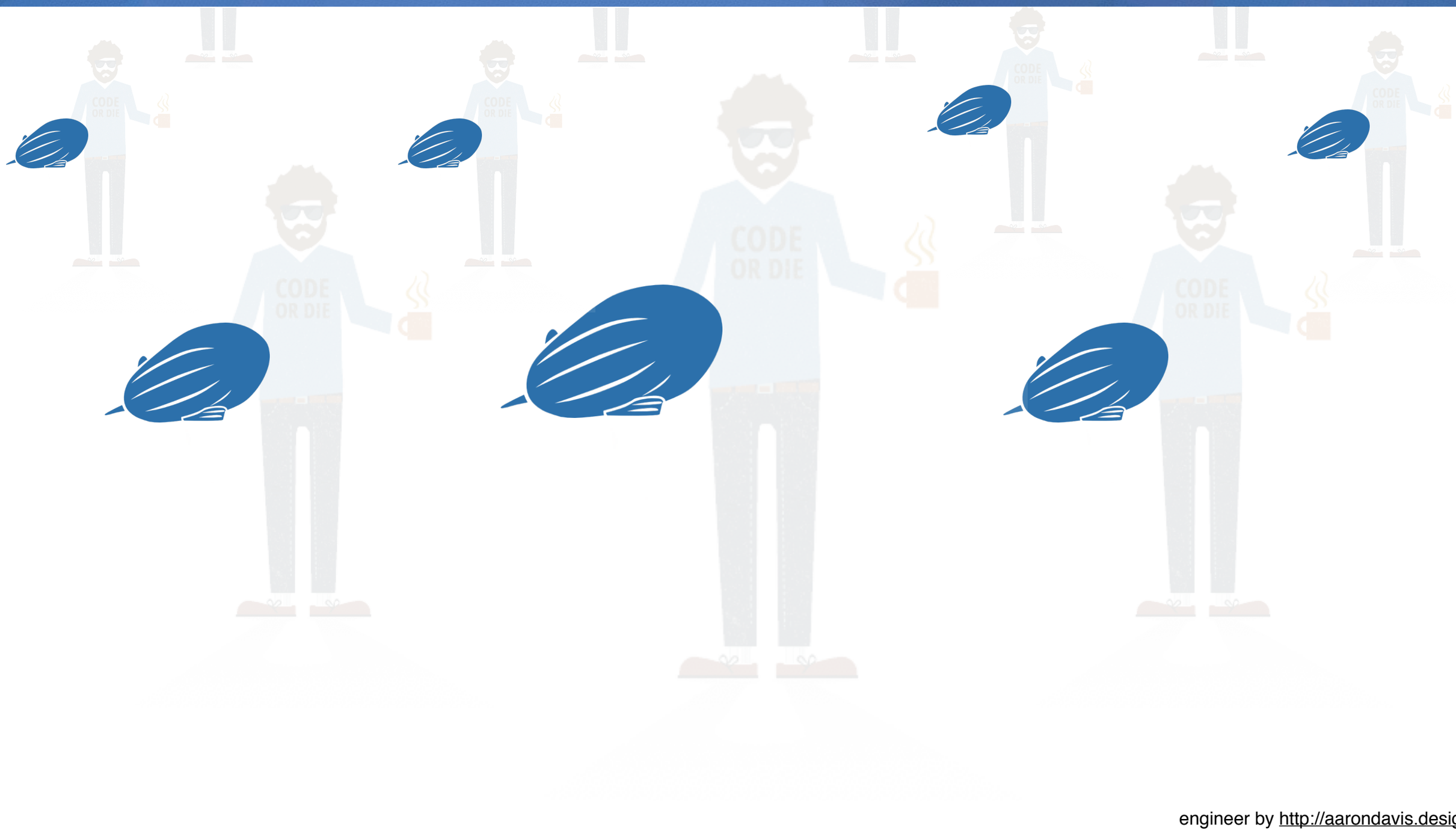
engineer by http://aarondavis.design/

An Organization



engineer by http://aarondavis.design/

That's too many!



engineer by http://aarondavis.design/

What is the problem?

Too much:

Install

Configure

Cluster resources

Solution?

We have containers

+

reverse proxy

Z Manager PoC

Linux box

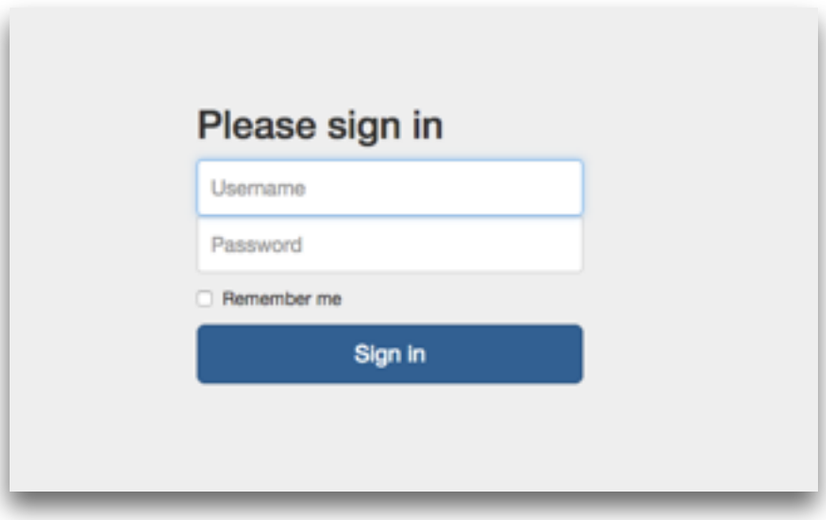
httpd + mod_php

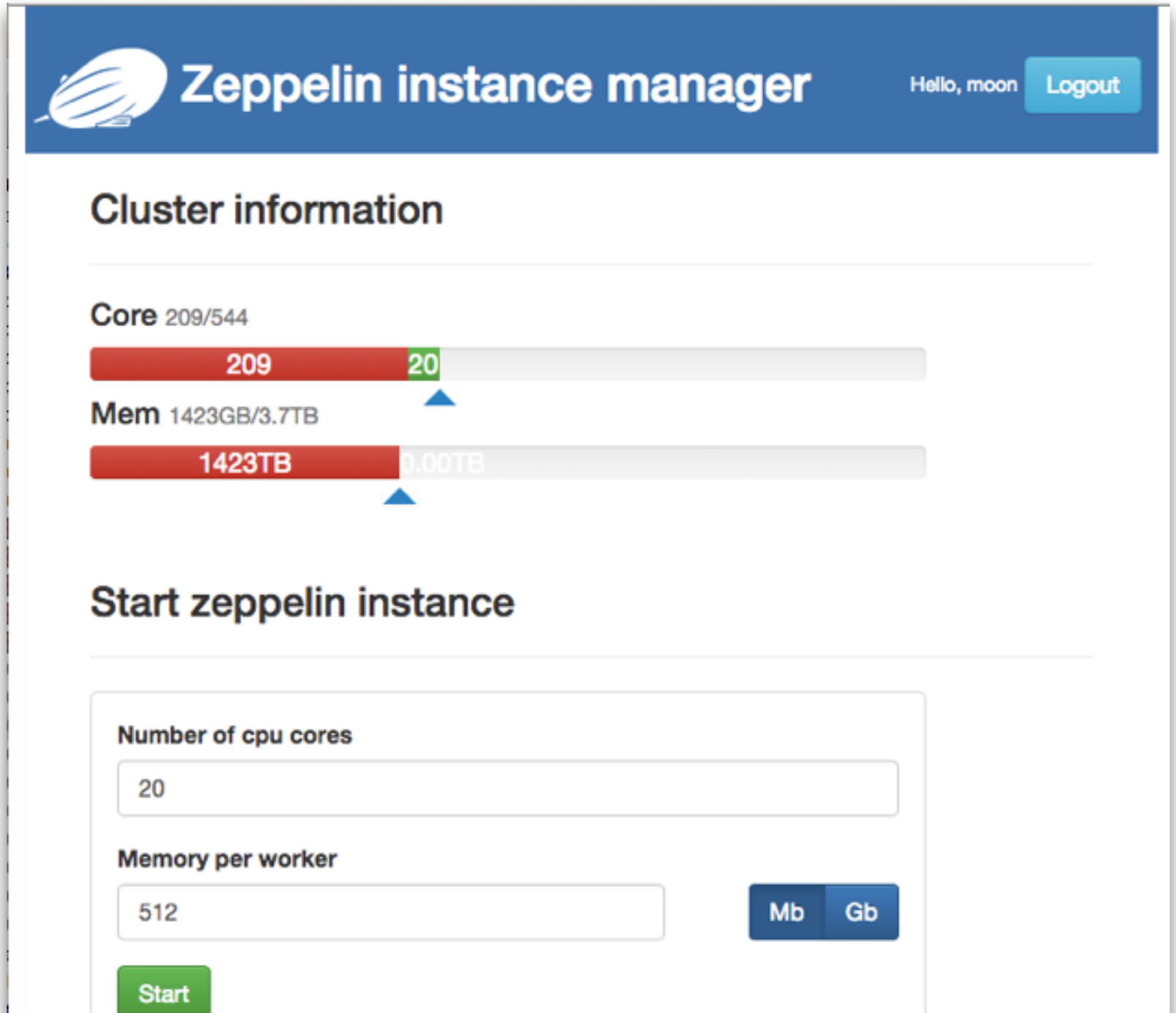
nginx

2 days, bash + php :(

engineer by http://aarondavis.design/

Z Manager PoC





Z Manager

Z Manager

http://github.com/NFLabs/z-manager

- Single binary
- Containerized deployment per user
- Reverse proxy
- Simple web application
- Apache 2.0 Licence
- SGA to ASF coming *

Z Manager

Auto-update

Z Manager process

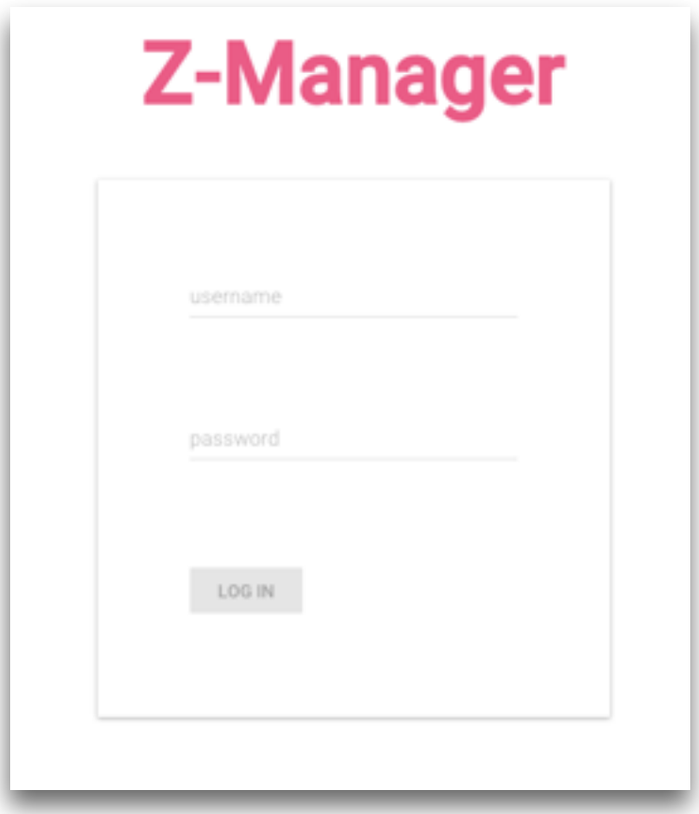
Linux box

Flink

go + react :)

engineer by http://aarondavis.design/

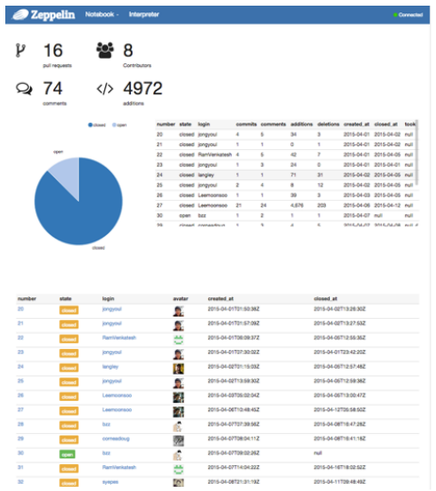
Z Manager


Z-Manager
username
password
LOG IN


anthonycorbacho
LOGOUT
Cores
Allocate
Free
Total
63
194
544
Memory (GB)
Allocate
Free
Total
600
1705.67
3808
RESET
CREATE

Helium

People do the similar work with different data

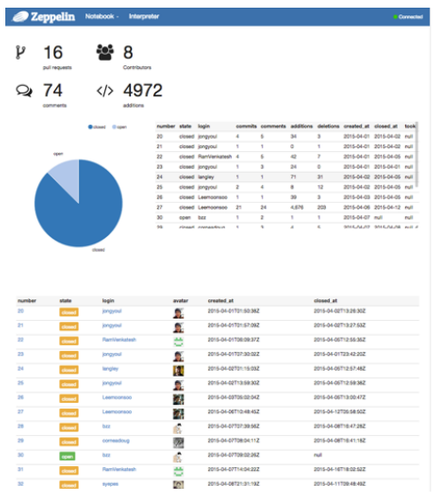


New visualization
Model & Algorithm
Data process pipeline

engineer by http://aarondavis.design/

Package and distribute work

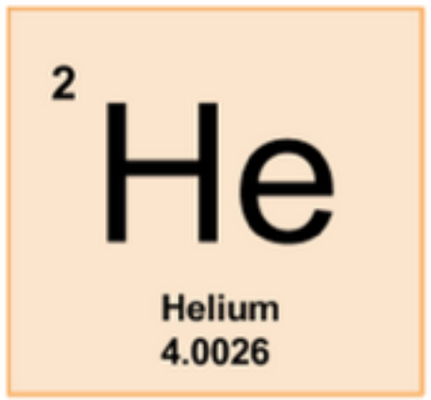
Pkg
Repo



New visualization
Model & Algorithm
Data process pipeline

engineer by http://aarondavis.design/

Helium



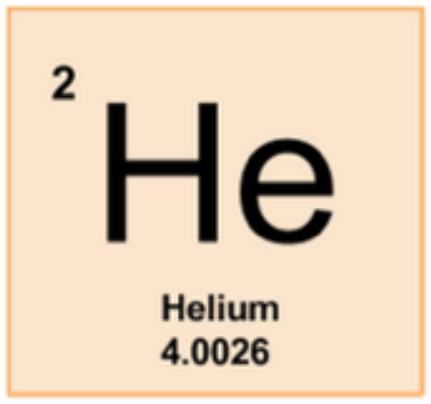
https://s.apache.org/helium

Platform for

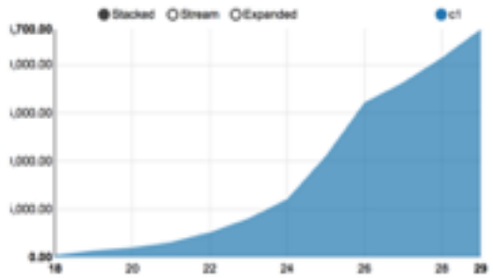
Data Analytics Application

on top of Apache Zeppelin

Helium Application



=



+

```
/** Configures the Oauth Credentials for accessing Twitter */
def configureTwitterCredentials(apiKey: String, apiSecret: String, acce
  val configs = new HashMap[String, String] ++= Seq(
    "apiKey" -> apiKey, "apiSecret" -> apiSecret, "accessToken" -> acce
ret)
  println("Configuring Twitter OAuth")
  configs.foreach{ case(key, value) =>
    if (value.trim.isEmpty) {
      throw new Exception("Error setting authentication - value for " +
    }
    val fullKey = "twitter4j.oauth." + key.replace("api", "consumer")
    System.setProperty(fullKey, value.trim)
    println("\tProperty " + fullKey + " set as [" + value.trim + "]")
  }
  println()
}
```

View

Algorithm

Zeppelin provided Resources

Resources

Data

- Result of last execution
- JDBC connection (from JDBC Interpreter)*

Computing

- SparkContext (from SparkInterpreter)
- Flink environment (from FlinkInterpreter)*

Any java object

- Provided by user created Interpreter
- Provided by user created Helium application

Application Examples

Data

- ex) get git commit log data
  https://github.com/Leemoonsoo/zeppelin-gitcommitdata

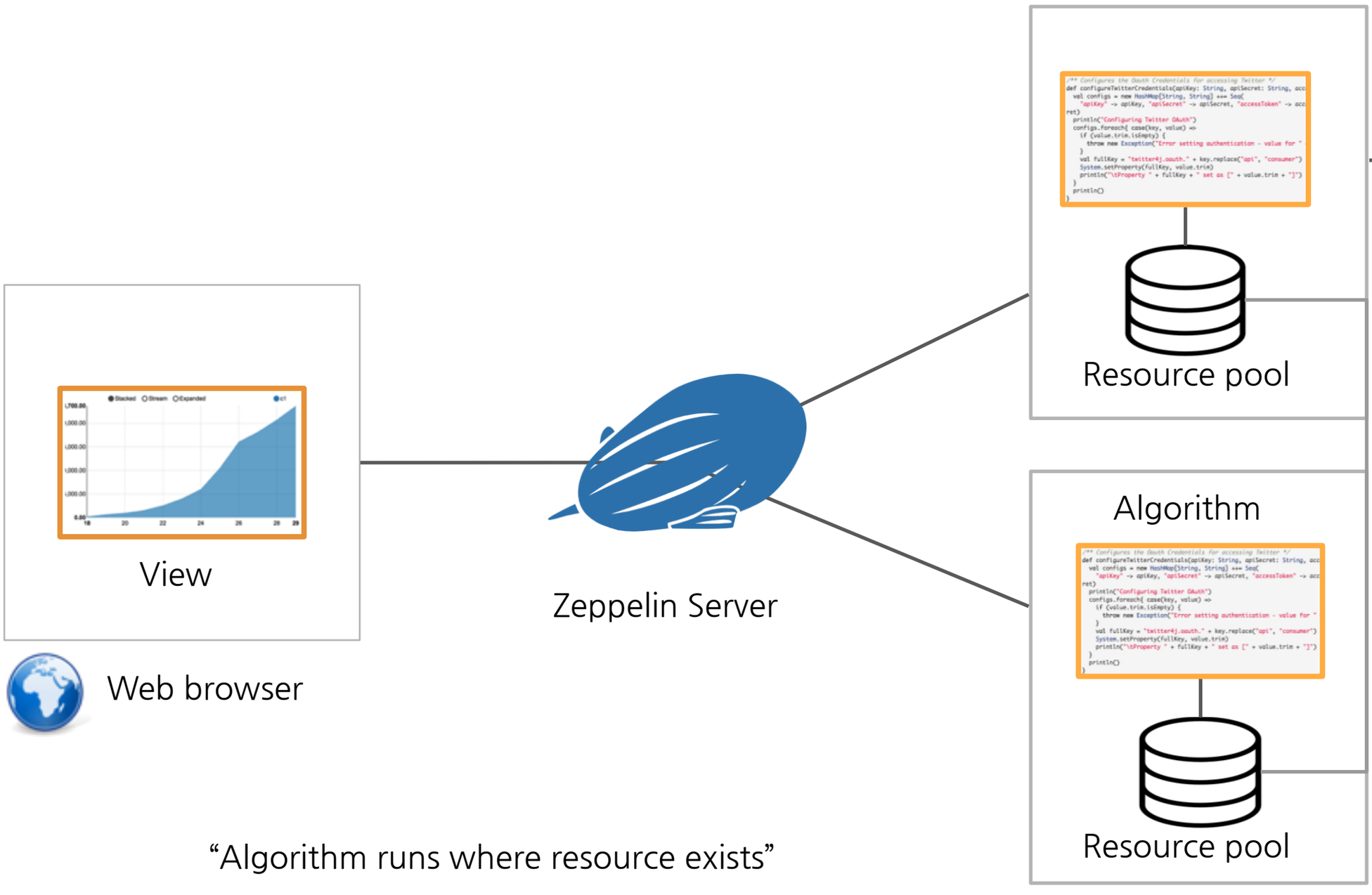
Computing

- ex) run cpu usage monitoring code across spark cluster, using SparkContext
  https://github.com/Leemoonsoo/zeppelin-sparkmon

Visualization

- ex) display result data as a wordcloud
  https://github.com/Leemoonsoo/zeppelin-wordcloud

How it works



“Algorithm runs where resource exists”

API

Easy API

```
class YourApplication extends org.apache.zeppelin.helium.Application {

  @Override
  public void run(ApplicationArgument arg, InterpreterContext context) {
      .....
  }
}
```

Just extend helium.Application

Application Spec

Simple

```
{
  mavenArtifact : "groupId:artifactId:version",
  className : "your.helium.application.Class",
  icon : "fa fa-cloud",
  name : "My app name",
  description : “some description",
  consume : [
    "org.apache.spark.SparkContext"
  ]
}
```

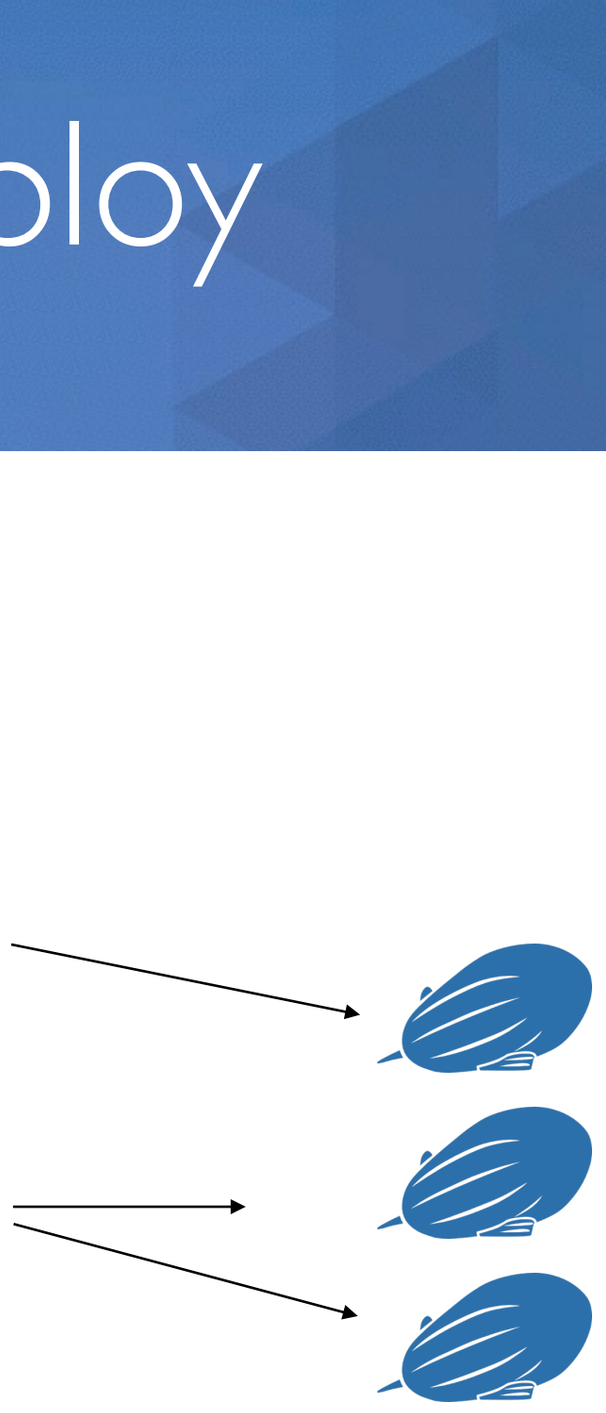
Writing a spec file allow Zeppelin load application

Deploy

Handy

Public

Private



Packaged to Jar and Distributed through Maven

Downloaded on the fly and run when user selects it

Thank you

Q & A

http://zeppelin.incubator.apache.org/

Moon
moon@nflabs.com

NFLabs
www.nflabs.com