MTA
SZTAKI

BIG DATA ÉS GÉPI TANULÁS KÖRNYEZET AZ MTA CLOUD-ON

KACSUK PÉTER, NAGY ENIKŐ, PINTYE ISTVÁN, HAJNAL ÁKOS, LOVAS RÓBERT

TARTALOM

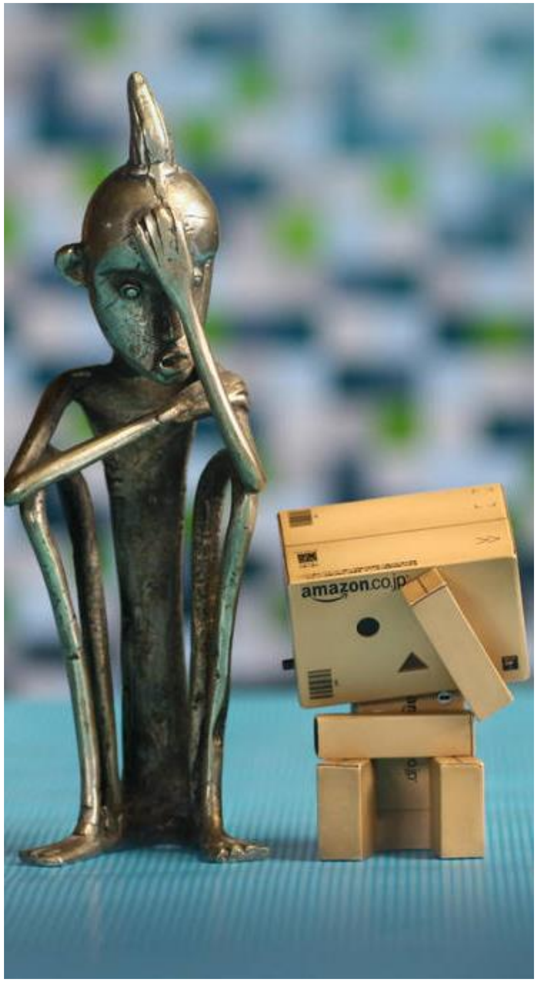
- MTA Cloud
- Big Data és gépi tanulást támogató szoftver eszközök
- Apache Spark keretrendszer
- Occopus felhő menedzser és orkesztrátor
- Rstudio, R, SparklyR, Spark klaszter, HDFS környezet létrehozása
- Jupyter, Python, Spark ml, Spark klaszter, HDFS környezet létrehozása
- Továbbfejlesztési irányok

MTA CLOUD

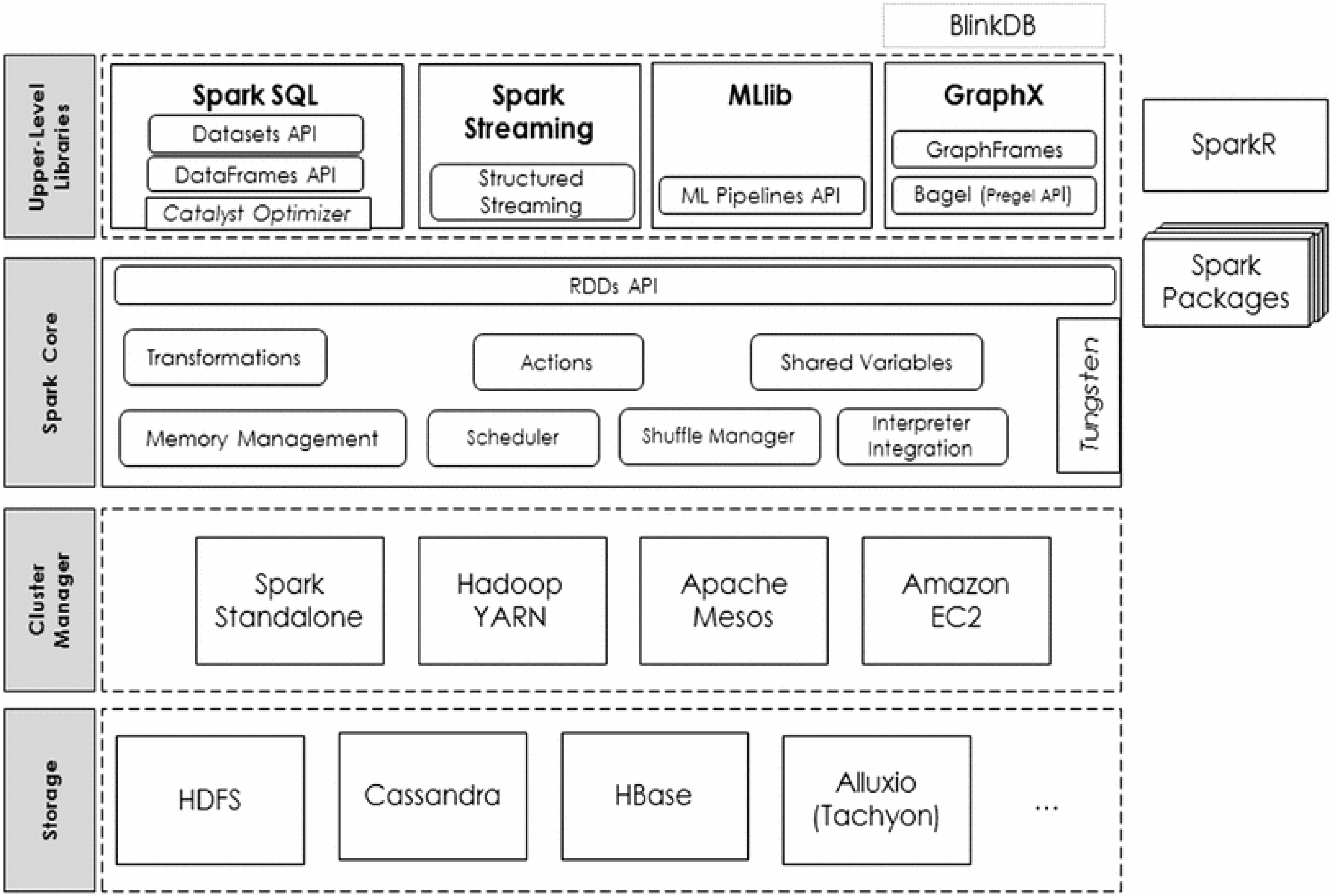
- Két telephely: Wigner Adatközpont és MTA SZTAKI
- OpenStack és Docker konténer alapú IaaS felhő infrastruktúra
- Ingyenes használat MTA kutatói számára
- Jelenleg 95 aktív projekt, 2016 óta több, mint 20 különböző MTA intézetből pld.:
  - Nyelvtudományi Intézet, Konkoly Thege Miklós Csillagászati Intézet, Szociológiai Intézet, Rényi Alfréd Matematikai Kutatóintézet
- 4000 vCPU, 5,25 TB memória, 762 TB tároló kapacitás
  - 2017-es bővítéssel GPGPU kártyák: Wigner 4 db nVidia V100, SZTAKI oldalon 8 darab Tesla K80 GPU

MUNKÁNK CÉLJA

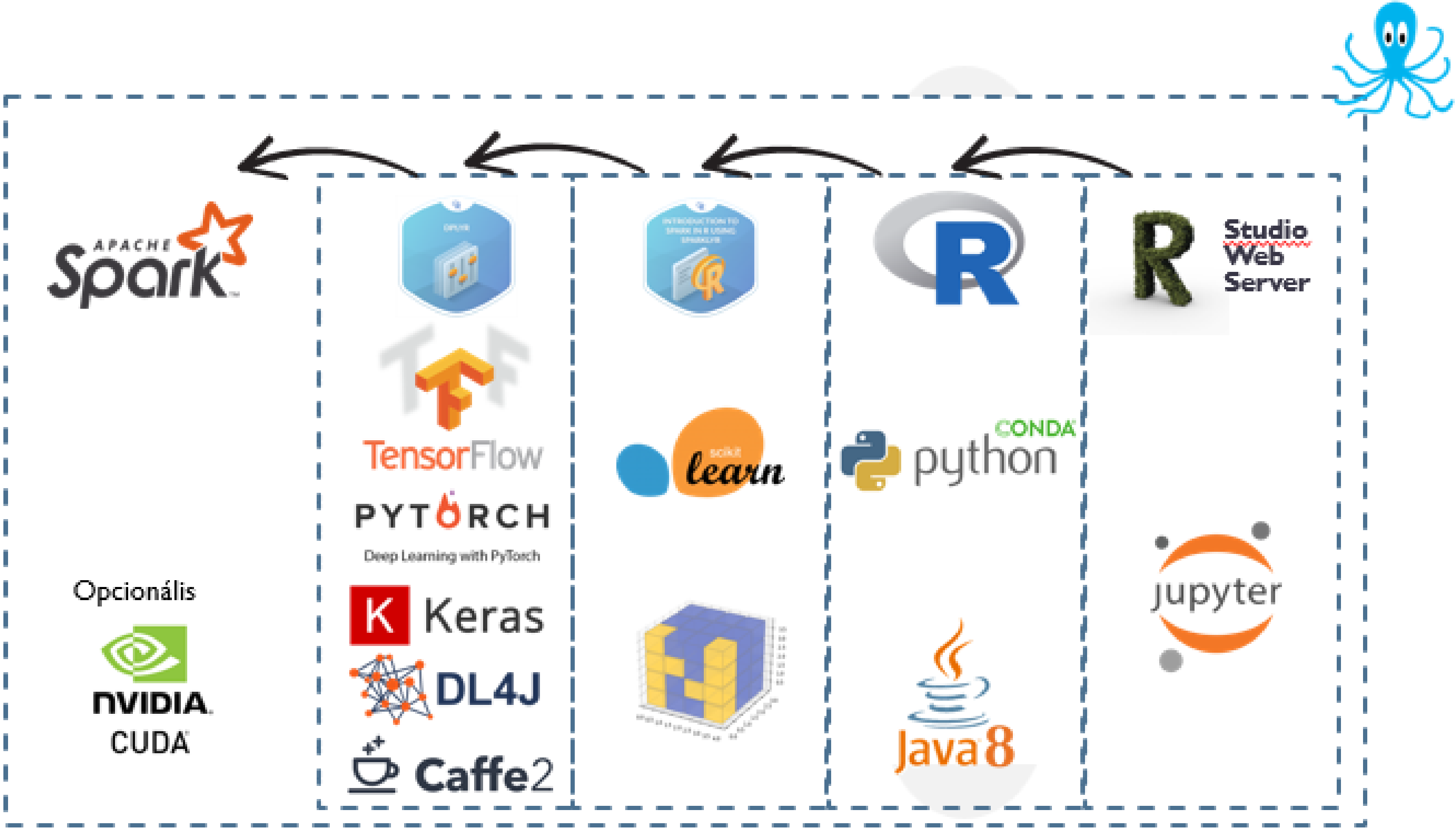
- Gépi tanulás egyre fontosabb
- **DE:** nagy számítási erőforrás igény
    - MTA Cloud az MTA kutatóknak
- Apache Spark keretrendszer elterjedt
- **DE:** kiépítése nem triviális, az MTA Cloud felhasználók döntő többsége nem informatikus
- **CÉLUNK:** megkönnyíteni az MTA Cloud felhasználók számára a Big Data és gépi tanulást támogató környezetek felépítését
    - telepítési mechanizmus elkészítése, a megoldás használata Occopus orkesztrációs eszközzel



APACHE SPARK ÖKO SZISZTÉMA



BIG DATA ÉS GÉPI TANULÁST TÁMOGATÓ ESZKÖZÖK



Opcionális

ELŐRE DOBOZOLT MEGOLDÁSOK

- **Google Cloud - Cloud Dataproc**
  - Google Cloud Platform
- **Amazon Elastic MapReduce (EMR)**
- **CloudBreak from Hortonworks**
  - Hortonworks platforms

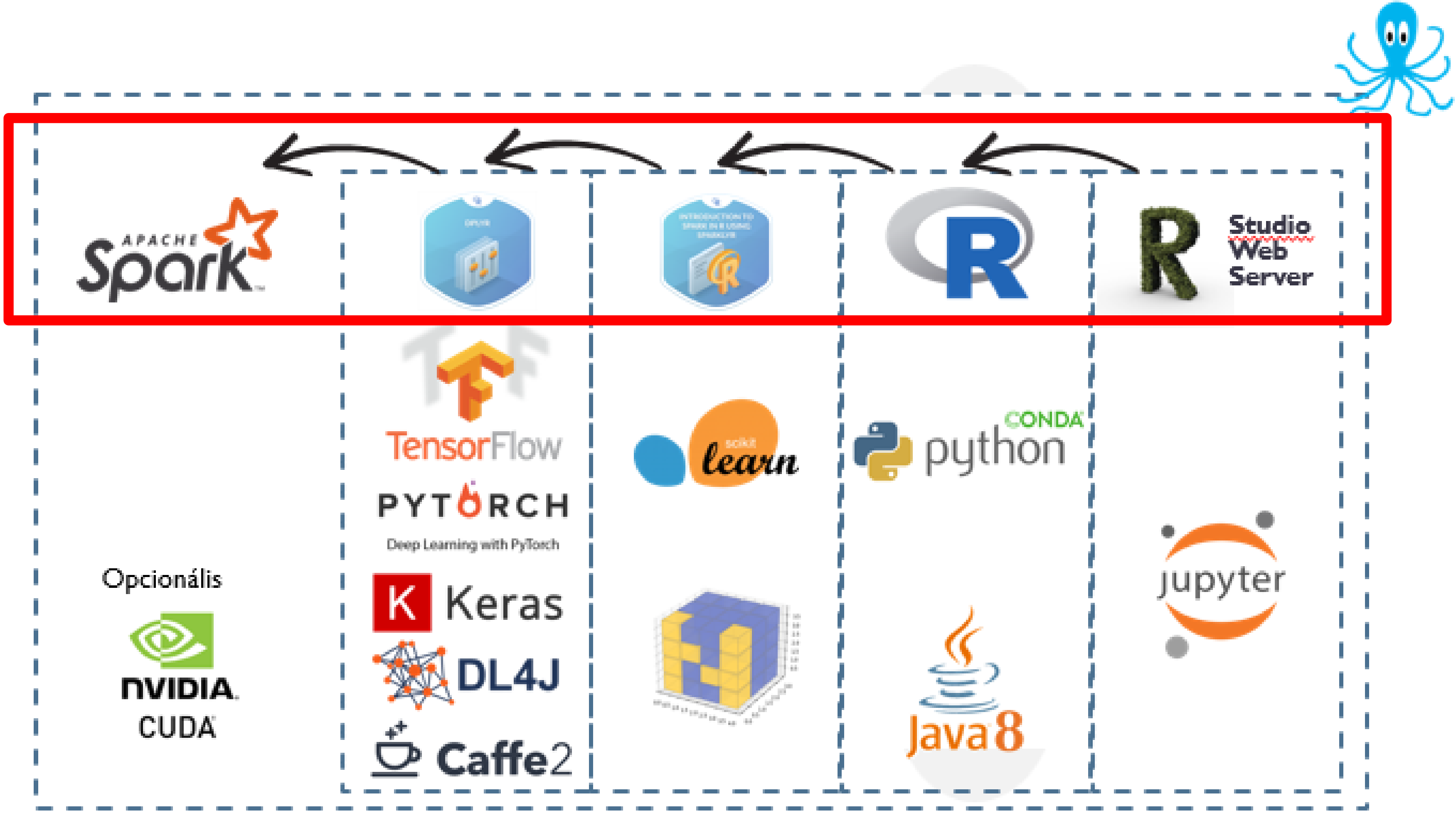
Problémák:

- Kereskedelmi felhők
- Beszállítói függőség (vendor lock-in)
- Az MTA Cloud-on nem állnak rendelkezésre

MTA KUTATÓK TÁMOGATÁSA

- MTA TK Politikatudományi Intézet
  - Feladat: nyomtatott és online médiában megjelenő újságcikkek osztályozása
  - textanalitika, neurális háló használata, R nyelvben
  - Spark klaszter használata
- MTA CSFK Csillagászati és Földtudományi Kutatóközpont
  - Feladat: fényerősség intenzitásának változása alapján van-e bolygója egy csillagnak?
  - klasszifikációs feladat, konvolúciós neurális hálóval, Python, Keras, TensorFlow, GPU használat

MTA TK POLITIKATUDOMÁNYI INTÉZET FELADATÁHOZ SZÜKSÉGES KÖRNYEZET



OCCOPUS

- Nyílt forráskódú hibrid orkesztrációs eszköz
- MTA SZTAKI által fejlesztett
- Felhőfüggetlen megoldás
- Hordozható leírók
- Skálázási lehetőség
- Kontextualizáció cloud-init segítségével

openstack
CLOUD SOFTWARE

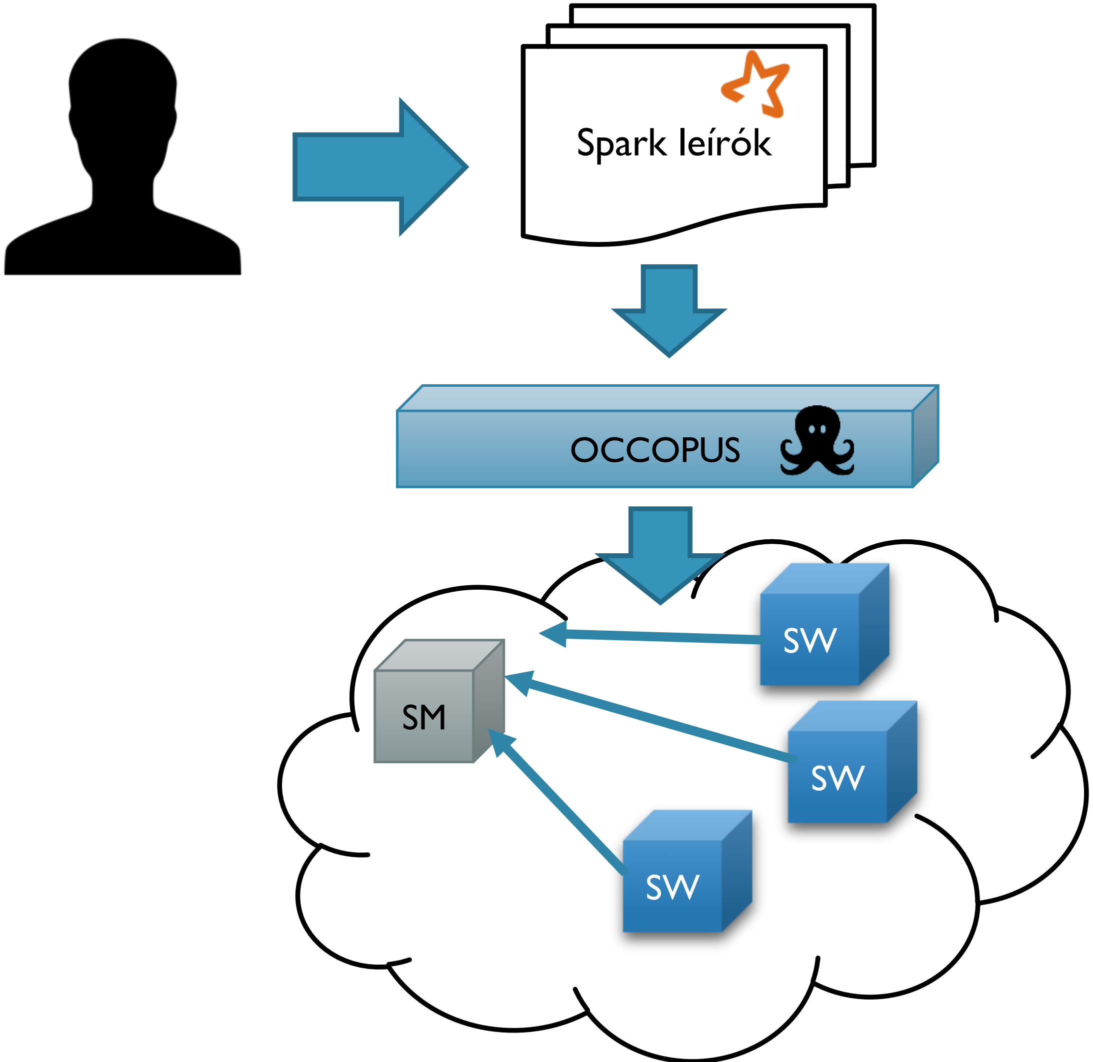
CloudSigma

Open
Nebula

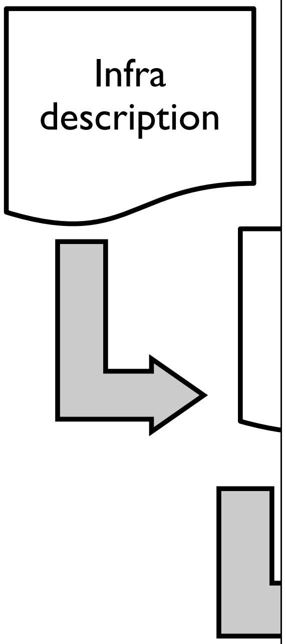
amazon
web services

CloudBroker

A MEGOLDÁS ARCHITEKTÚRÁJA



OCCOPUS LEÍRÓK

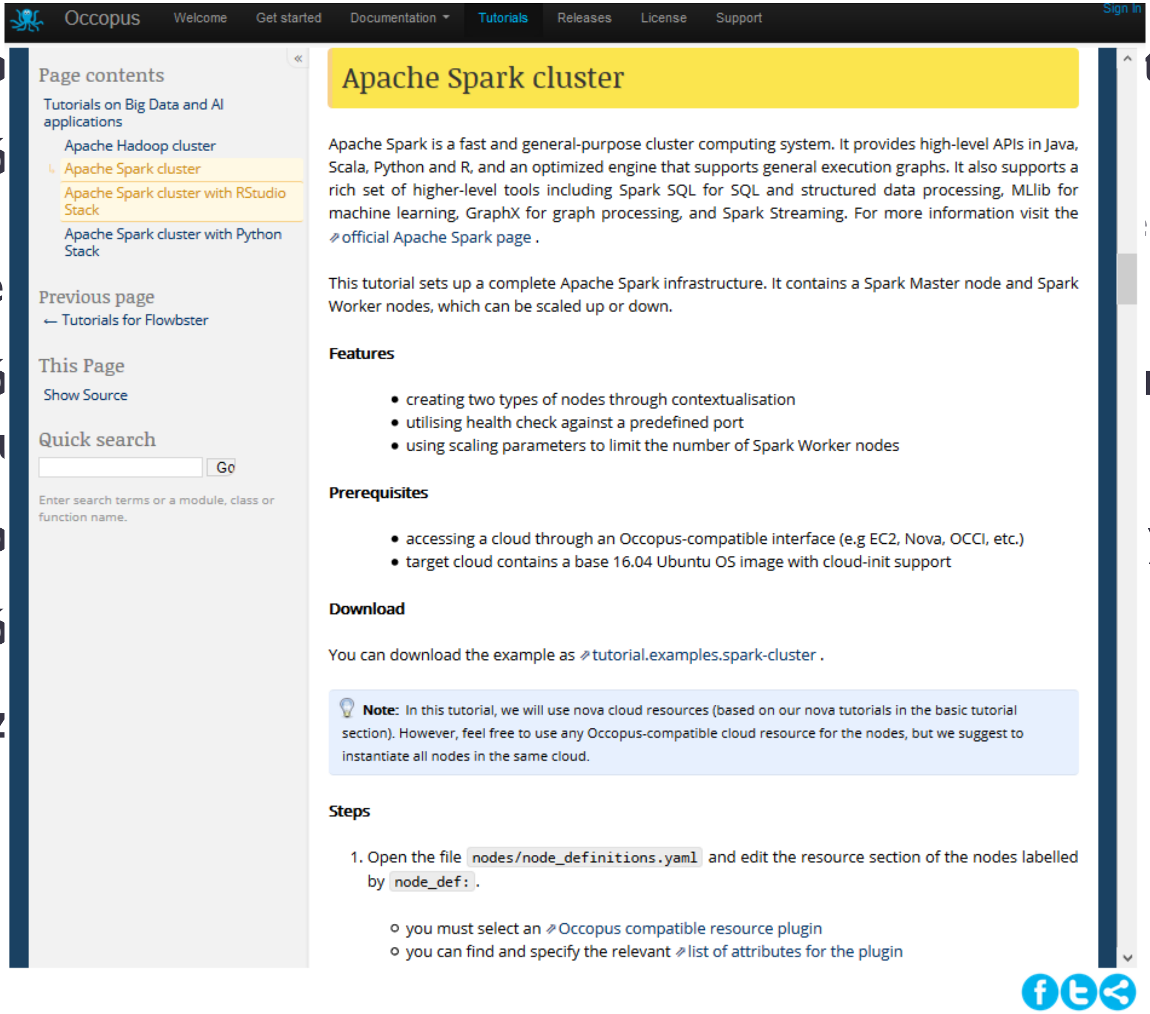


```
'node_def: spark_master_node ':
  −
    resource:
      type: nova
      endpoint: https://sztaki.cloud.mta.hu...
      image_id: ...
      network_id: ...
      flavor_name: ...
      security_groups:...
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_spark_master.yaml
    health_check:
      ports:
        − 50070
'node_def: spark_worker_node ':
  −
    resource:
      type: nova
      endpoint: https://sztaki.cloud.mta.hu...
      project_id: a9c30db63ddf47a98045ef9c726c7436
      image_id: ...
      network_id: ...
      flavor_name: ...
      security_groups:...
    contextualisation:
      type: cloudinit
      context_template: !yaml_import
        url: file://cloud_init_spark_workere.yaml
    health_check:
```

- Binárisok telepítése
- Konfigurációs fájlok telepítése (testreszabhatóság)
- Spark konfiguráció
- Spark démonok elindítása

A MEGOLDÁS HASZNÁLATA

1. **Occo**
tall manual”
2. **Leíró**
„Tuto
Spark
cluste
3. **Leíró**
ng-resource-
attribu
4. **Occo**
yezetét
5. **Leíró**
6. **Klasz**



Page contents

Tutorials on Big Data and AI applications
- Apache Hadoop cluster
- Apache Spark cluster
- Apache Spark cluster with RStudio Stack
- Apache Spark cluster with Python Stack

Previous page

← Tutorials for Flowbster

This Page

Show Source

Quick search

Enter search terms or a module, class or function name.

Apache Spark cluster

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming. For more information visit the official Apache Spark page .

This tutorial sets up a complete Apache Spark infrastructure. It contains a Spark Master node and Spark Worker nodes, which can be scaled up or down.

Features

- creating two types of nodes through contextualisation
- utilising health check against a predefined port
- using scaling parameters to limit the number of Spark Worker nodes

Prerequisites

- accessing a cloud through an Occopus-compatible interface (e.g EC2, Nova, OCCI, etc.)
- target cloud contains a base 16.04 Ubuntu OS image with cloud-init support

Download

You can download the example as tutorial.examples.spark-cluster .

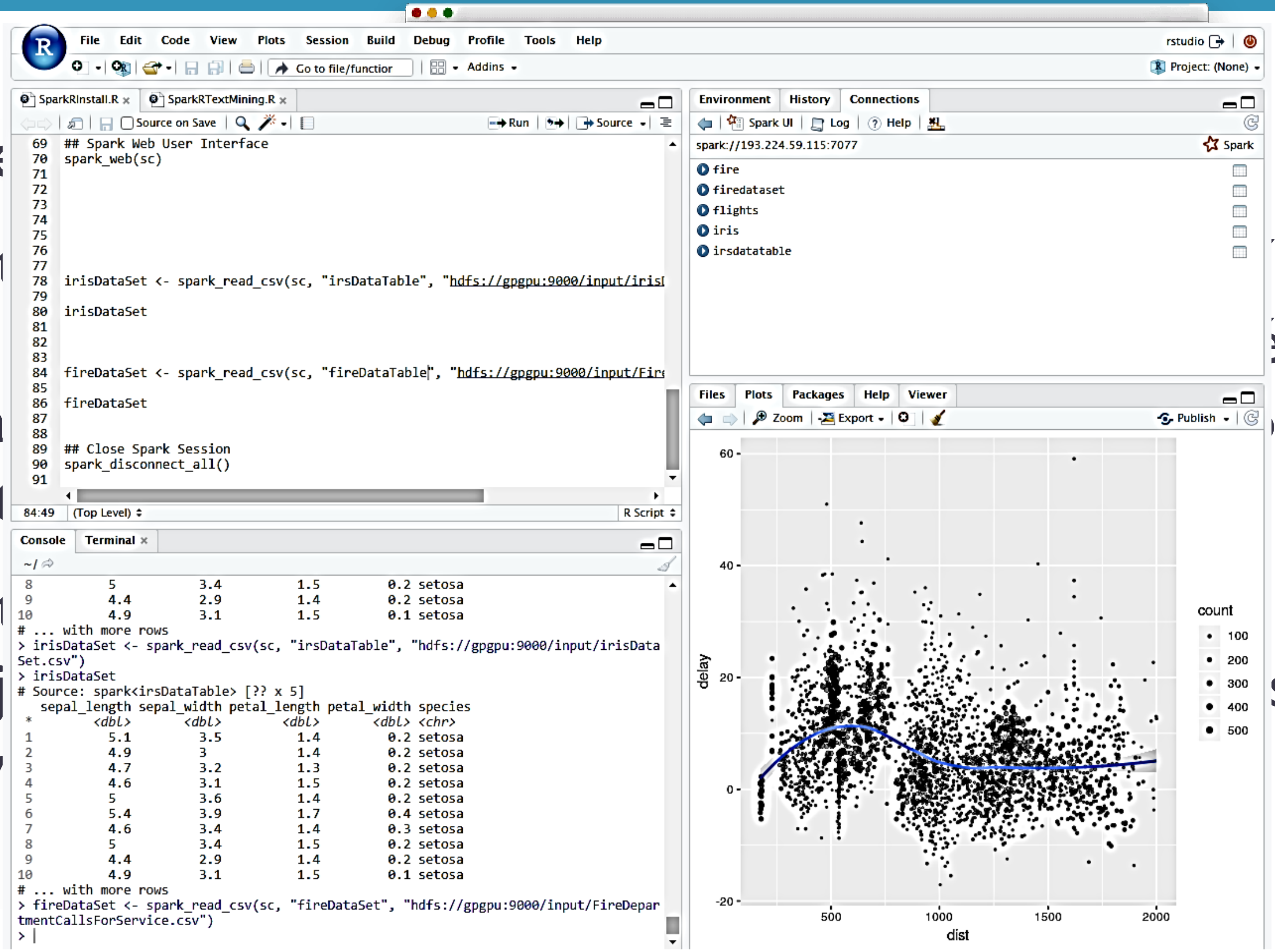
Note: In this tutorial, we will use nova cloud resources (based on our nova tutorials in the basic tutorial section). However, feel free to use any Occopus-compatible cloud resource for the nodes, but we suggest to instantiate all nodes in the same cloud.

Steps

1. Open the file `nodes/node_definitions.yaml` and edit the resource section of the nodes labelled by `node_def:`.
    - you must select an Occopus compatible resource plugin
    - you can find and specify the relevant list of attributes for the plugin

FELHASZNÁLÁSI LEHETŐSÉGEK

- Le
- Int lt
  fo és
  na ok
  lét
- Int lt
  fej si
  ny

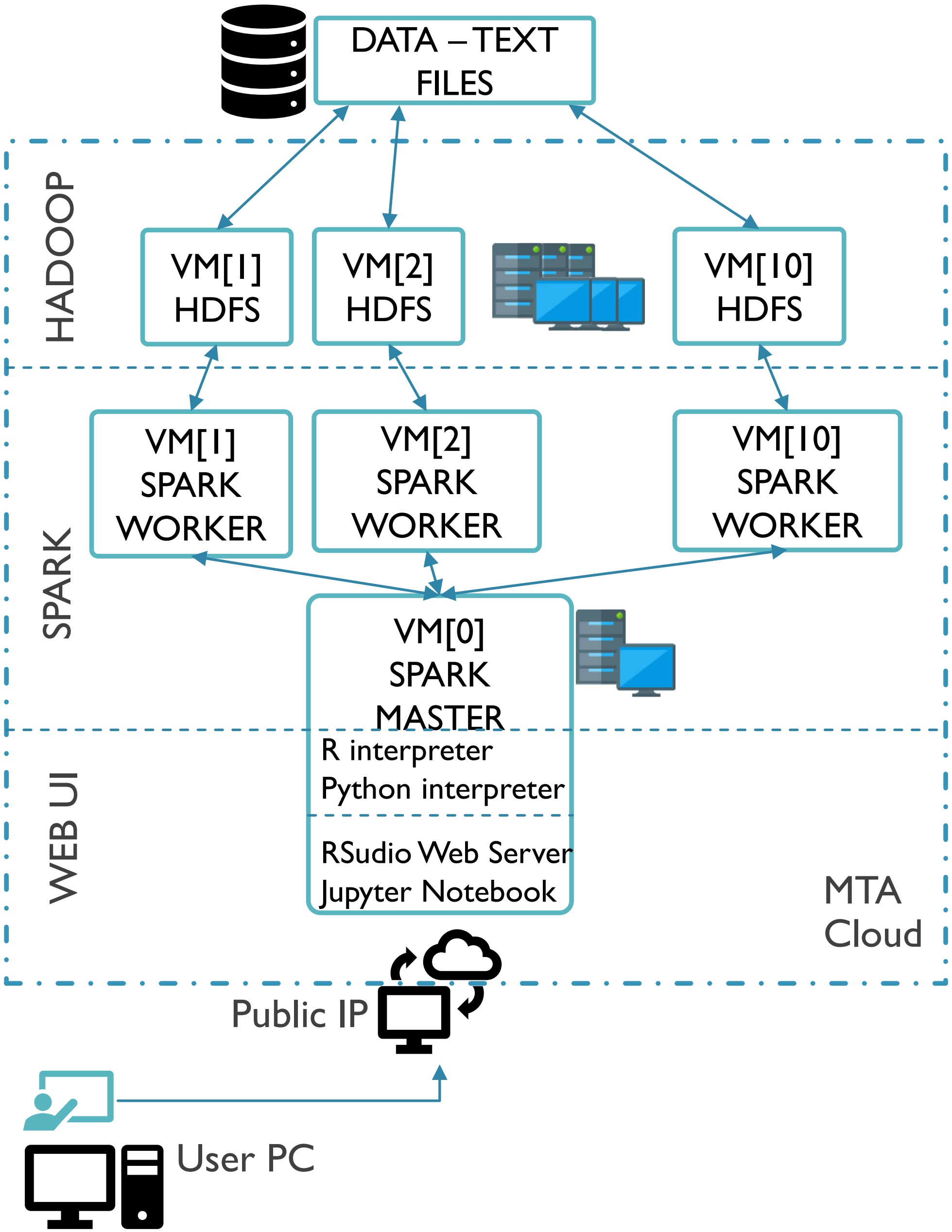


```
## Spark Web User Interface
spark_web(sc)







irisDataSet <- spark_read_csv(sc, "irsDataTable", "hdfs://gpgpu:9000/input/irisD

irisDataSet



fireDataSet <- spark_read_csv(sc, "fireDataTable", "hdfs://gpgpu:9000/input/Fire

fireDataSet



## Close Spark Session
spark_disconnect_all()
```

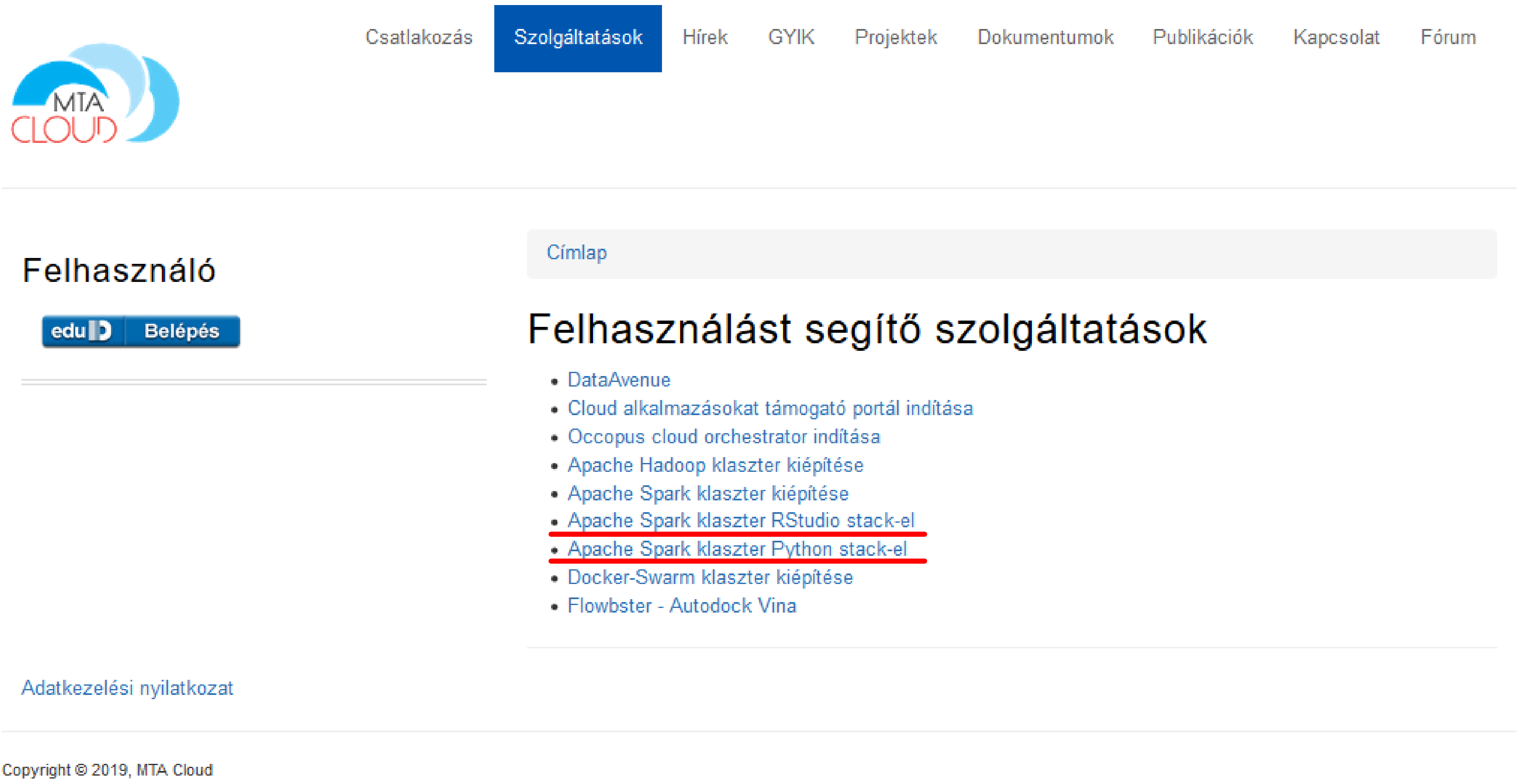
```
8          5          3.4          1.5          0.2 setosa
9          4.4        2.9          1.4          0.2 setosa
10         4.9        3.1          1.5          0.1 setosa
# ... with more rows
> irisDataSet <- spark_read_csv(sc, "irsDataTable", "hdfs://gpgpu:9000/input/irisData
Set.csv")
> irisDataSet
# Source: spark<irsDataTable> [?? x 5]
   sepal_length sepal_width petal_length petal_width species
 *        <dbl>       <dbl>        <dbl>       <dbl> <chr>
 1          5.1         3.5          1.4         0.2 setosa
 2          4.9         3            1.4         0.2 setosa
 3          4.7         3.2          1.3         0.2 setosa
 4          4.6         3.1          1.5         0.2 setosa
 5          5           3.6          1.4         0.2 setosa
 6          5.4         3.9          1.7         0.4 setosa
 7          4.6         3.4          1.4         0.3 setosa
 8          5           3.4          1.5         0.2 setosa
 9          4.4         2.9          1.4         0.2 setosa
10          4.9         3.1          1.5         0.1 setosa
# ... with more rows
> fireDataSet <- spark_read_csv(sc, "fireDataSet", "hdfs://gpgpu:9000/input/FireDepar
tmentCallsForService.csv")
> |
```

```
spark://193.224.59.115:7077
fire
firedataset
flights
iris
irsdatatable
```

A létrejött MI architekturák



ELÉRÉS AZ MTA CLOUD-ON



Felhasználó

Címlap

Felhasználást segítő szolgáltatások

- DataAvenue
- Cloud alkalmazásokat támogató portál indítása
- Occopus cloud orchestrator indítása
- Apache Hadoop klaszter kiépítése
- Apache Spark klaszter kiépítése
- Apache Spark klaszter RStudio stack-el
- Apache Spark klaszter Python stack-el
- Docker-Swarm klaszter kiépítése
- Flowbster - Autodock Vina

Adatkezelési nyilatkozat

Copyright © 2019, MTA Cloud

TOVÁBBFEJLESZTÉSI IRÁNYOK

- Különböző alkalmazás osztályokhoz szükséges szoftver környezetek felépítése Occopusszal és ezek publikálása az MTA Cloud web lapján
- Pl. Az MTA CSFK Csillagászati és Földtudományi Kutatóközpont alkalmazásához szükséges környezet:

ÖSSZEFOGLALÁS

- Cél, hogy a magyar kutatók minél gyorsabban kezdhessék el az MI-hez kapcsolódó kutató munkát az MTA Cloud-on
- Ehhez olyan szoftver környezet kell, ami az ehhez szükséges szoftver eszközöket azonnal, egymással együttműködve és **működőképesen** tartalmazza.
- Célunk, hogy ilyen jól működő és felhasználható MI környezeteket hozzunk létre az MTA Cloud-on
- Az eddig összeállított környezetek (Hadoop, Spark, Jupyter, Rstudio) **tutorial formájában** elérhetők és kipróbálhatók az MTA Cloud-on
- Várjuk további MI igények bejelentését
- A felhalmozott tudás segítségével konzultációs segítséget is vállalunk

KÖSZÖNÖM A FIGYELMET!

Kérdések?