



ELKH Cloud

Mélytanulást támogató referencia architektúra az ELKH Cloudon

Farkas Attila

farkas.attila@sztaki.hu



Bemutatkozás



Farkas Attila

- ▶ SZTAKI - tudományos segédmunkatárs, kutató
- ▶ PERL - Párhuzamos és Elosztott Rendszerek Kutatólaboratórium

- ▶ farkas.attila@sztaki.hu

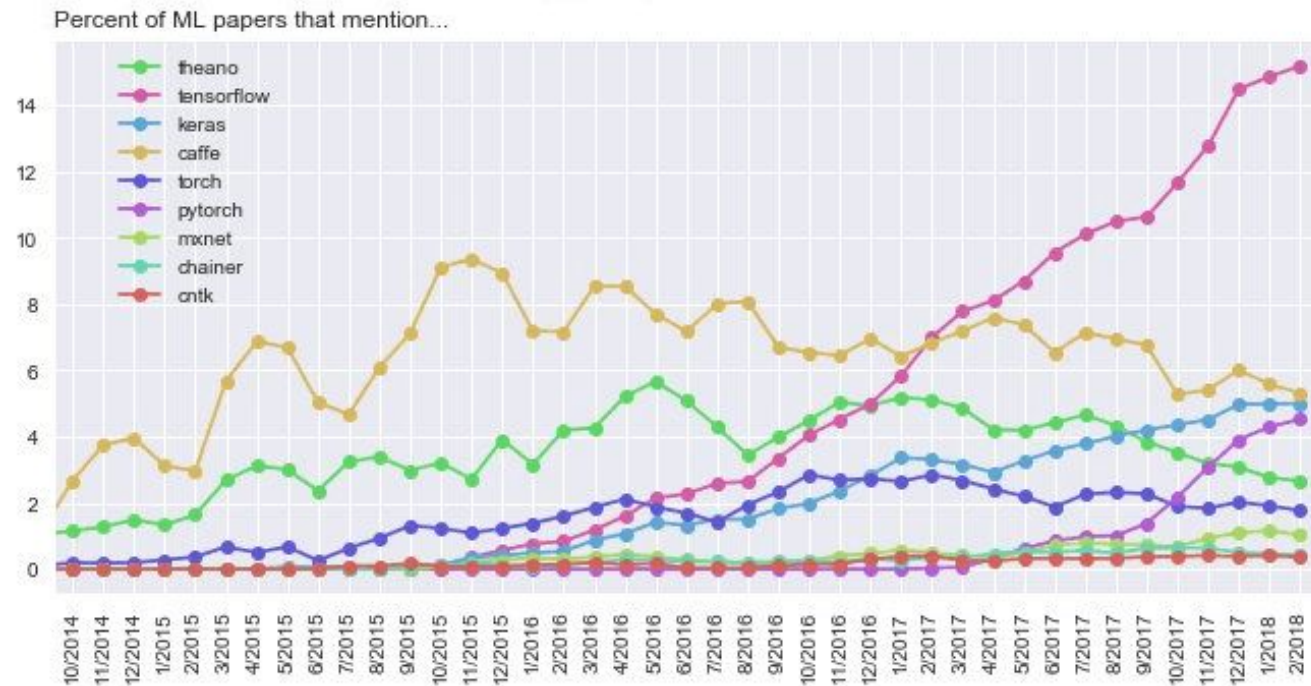


ELKH Cloud

TensorFlow és Jupyter

TensorFlow

- ▶ Google Brain csapat által fejlesztett Python (C++, JS) függvénykönyvtár adatfolyam programozáshoz
- ▶ 2015 novemberében publikálták először
- ▶ CPU/GPU támogatás, több platformon



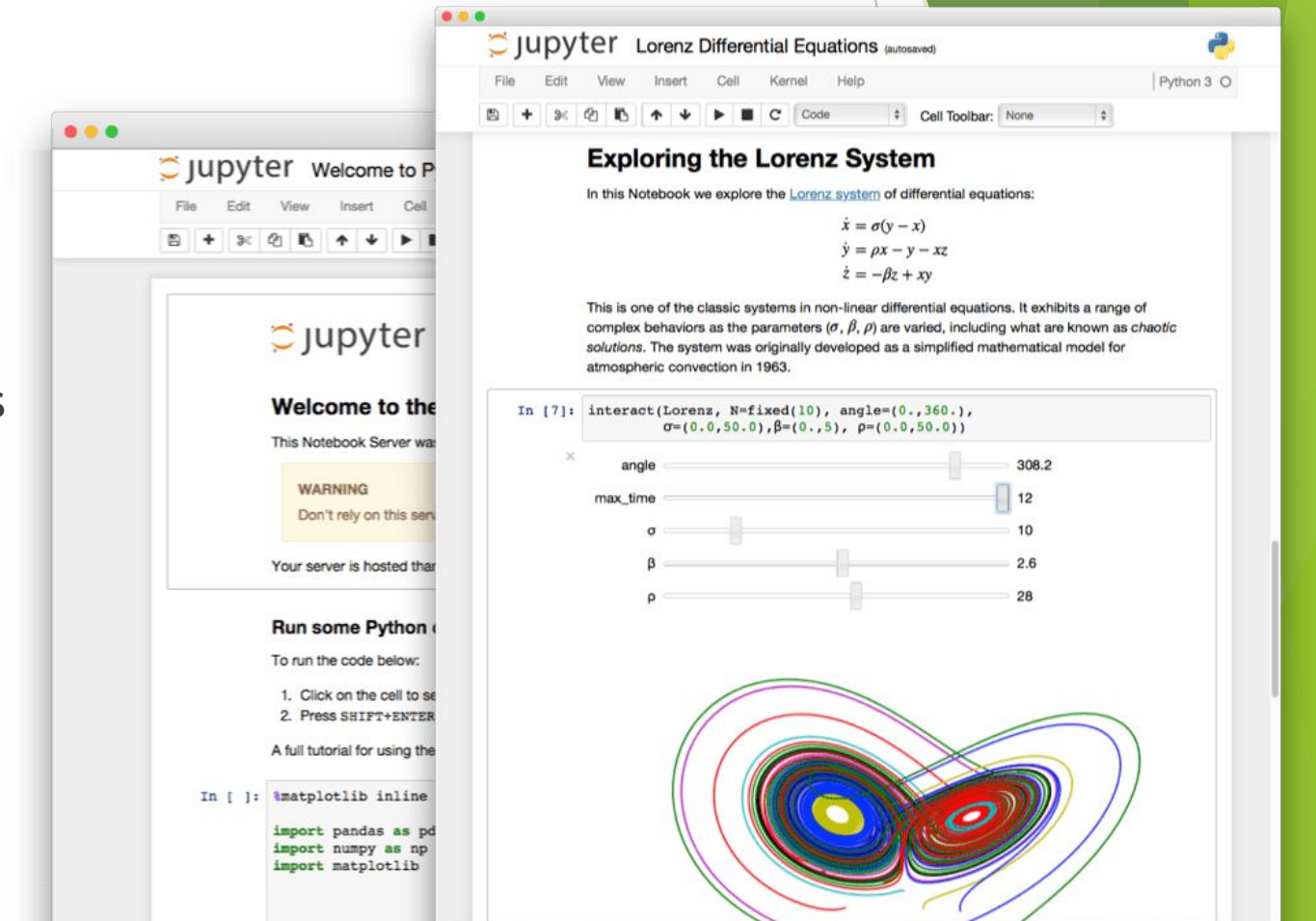
Keras



- ▶ Főbb tervezési szempontok a megtervezésekor
 - ▶ Felhasználóbarátság
 - ▶ Modularitás
 - ▶ Egyszerű bővíthetőség
 - ▶ Python
- ▶ A TensorFlow 2017 óta az alacsony szintű interfész mellett a Kerast hivatalosan támogatja, olyannyira, hogy a TF csomag részeként elérhető
 - ▶ A TensorFlow 2.0 (2019 szept) már alapértelmezetten Keras csomaggal támogatott

Jupyter Notebook

- ▶ Nyílt forráskódú webalkalmazás
- ▶ Fejlesztő környezet biztosít
- ▶ Adat vizualizációs megoldás
- ▶ Széleskörű programozási nyelv támogatás
- ▶ A Notebookok könnyedén megoszthatók



jupyter Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Exploring the Lorenz System

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ , β , ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))
```

angle 308.2
max_time 12
 σ 10
 β 2.6
 ρ 28

`!matplotlib inline`

```
import pandas as pd
import numpy as np
import matplotlib
```

JupyterLab



- ▶ Jupyter Notebook továbbfejlesztett verziója
- ▶ Web alapú interaktív fejlesztőkörnyezet
- ▶ Terminál biztosítása
- ▶ Moduláris felépítés
- ▶ Bővítmények támogatása

The screenshot displays the JupyterLab web interface. The main notebook, 'In Depth: Linear Regression', is open, showing a text editor with code and a terminal window. The code includes a simple linear regression example:

```
[1]: rng = np.random.randn(10, 2)
x = 10 * x
y = 2 * x + 10
plt.scatter(x, y)
```

 The terminal shows the output:

```
[3]: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X, y)
```

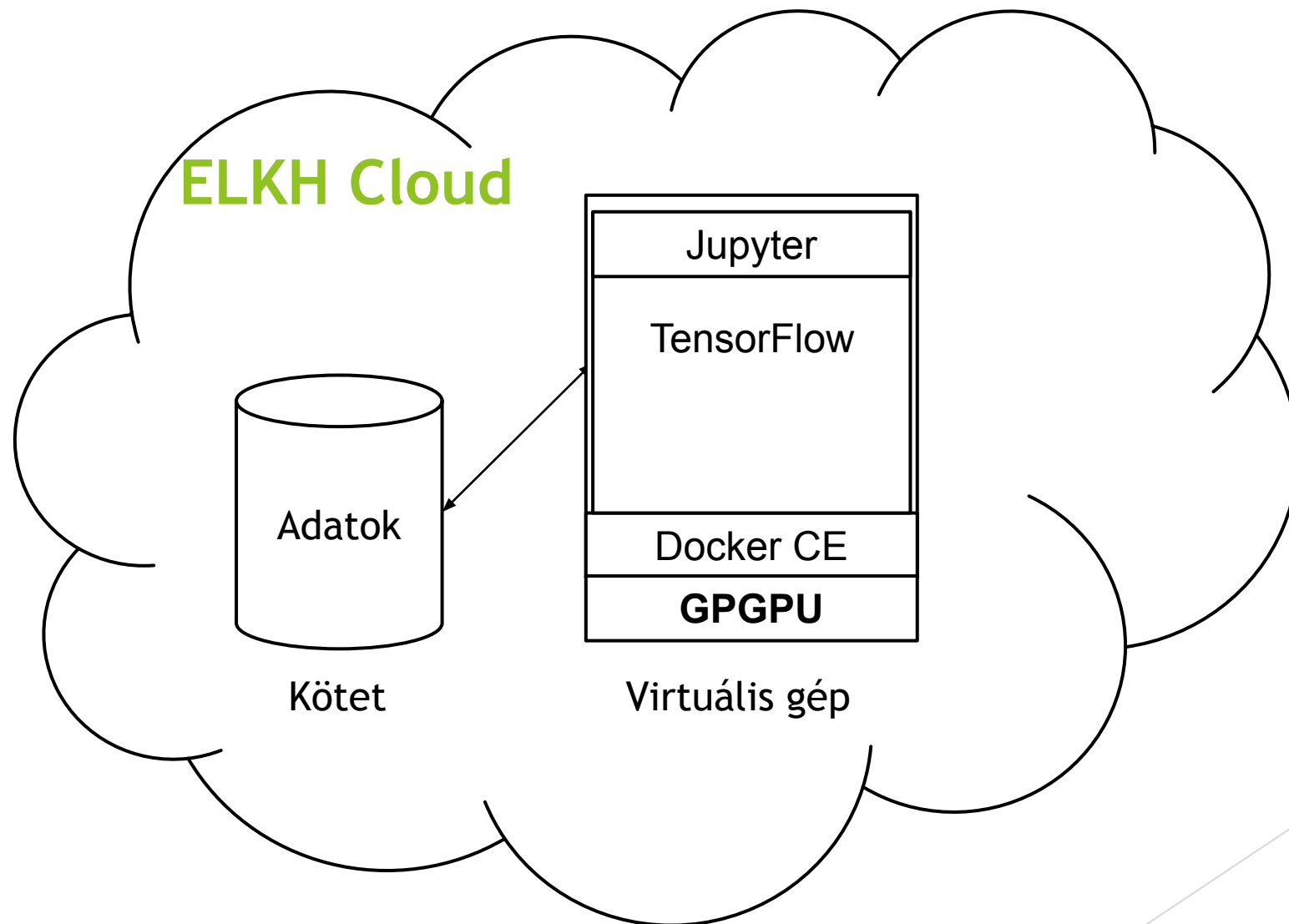
 The interface also features a file browser on the left, a launcher for other notebooks, and a console window. Other notebooks visible include 'Julia', 'python notebook', and 'R'. The 'python notebook' shows a Lorenz attractor plot and the Lorenz equations:
$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\rho z + xy \end{aligned}$$
 The 'R' notebook shows a scatter plot of Sepal.Length vs Sepal.Width for the Iris dataset.



ELKH Cloud

TensorFlow referencia architektúra ELKH Cloudon GPU erőforrásokkal

Architektúra



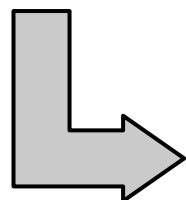
Occopus leírók

Infrastruktúra leíró
(infrastructure
description)

```
infra_name: tensorflow
user_id: somebody@somewhere

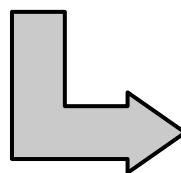
variables:
  JUPYTER_PASSWORD: tensorflow

nodes:
  - &M
    name: tensorflow
    type: tensorflow_node
```



Csomópont leíró
(Node definition)

```
'node_def:tensorflow_node':
-
  resource:
    type: nova
    endpoint: replace_with_endpoint_of_nova_interface_of_your_cloud
    project_id: replace_with_projectid_to_use
    user_domain_name: Default
    image_id: replace_with_id_of_your_image_on_your_target_cloud
    network_id: replace_with_id_of_network_on_your_target_cloud
    server_name: tensorflow-{{cut(infra_id,0,4)}}
    flavor_name: replace_with_id_of_the_flavor_on_your_target_cloud
    key_name: replace_with_name_of_keypair_or_remove
    security_groups:
      - replace_with_security_group_to_add_or_remove_section
    floating_ip: add_yes_if_you_need_floating_ip_or_remove
  contextualisation:
    type: cloudinit
    context_template: !yaml_import
      url: file://cloud_init_tensorflow.yaml
  health_check:
    ports:
      - 8888
    timeout: 3600
```



Cloud-init
fájlok

```
#cloud-config
...

runcmd:
- echo "==> DEPLOYMENT started."
- export DEBIAN_FRONTEND=noninteractive
- apt update && apt upgrade -y
- /bin/install-nvidia-driver.sh
- /bin/install-nvidia-container-toolkit.sh
- /bin/start-tensorflow-jupyterlab.sh
- echo "==> DEPLOYMENT finished."
```

Megoldás használatának lépései

Felhasználó feladatköre:

0. Lépés: Előkészítés (ELKH Cloud projekt, Üres Ubuntu VM elindítás)

1. Lépés: Occopus telepítés/konfigurálás a virtuális gépen

2. Lépés: Leírók letöltése a virtuális gépre
Occopus/ELKH Cloud weboldala

3. Lépés: Tűzfalszabályok létrehozása
ELKH Cloud OpenStack felületén

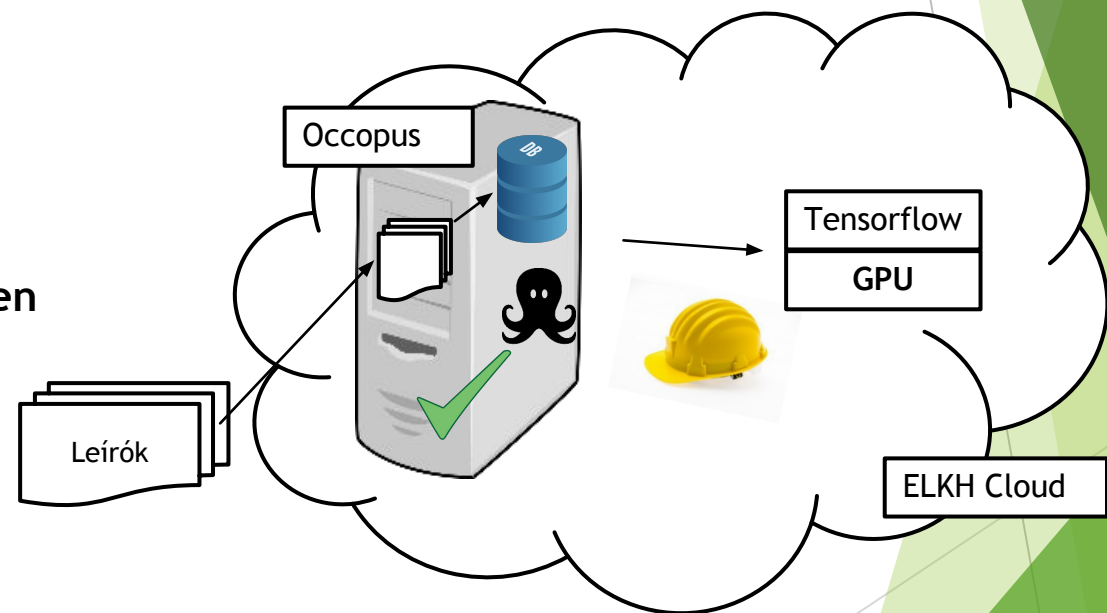
4. Lépés: Leírók személyre szabása a virtuális gépen

5. Lépés: Occopus aktiválása
`$ source ~/occopus/bin/activate`

6. Lépés: Leírók importálása Occopus számára
`$ occopus-import nodes/node_definitions.yaml`

7. Lépés: Infrastruktúra kiépítése
`$ occopus-build infra-tensorflow.yaml`

8. Lépés: Infrastruktúra használata



2. Lépés: Leírók letöltése a virtuális gépre

A rendelkezésre álló referencia architektúrák és leírásuk:

- Occopus cloud orchestrator indítása
- JupyterLab
- DataAvenue
- Cloud alkalmazásokat támogató portál indítása
- Flowbster - Autodock Vina
- CQueue klaszter
- Docker-Swarm klaszter kiépítése *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- Kubernetes klaszter
- Apache Hadoop klaszter kiépítése
- Apache Spark klaszter RStudio stack-el
- Apache Spark klaszter Python stack-el *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- TensorFlow, Keras, Jupyter Notebook stack
- **TensorFlow, Keras, Jupyter Notebook GPU stack** *(Frissítés: ELKH Cloud - Microsoft Azure hibrid felhő támogatással)*
- Horovod klaszter
- Kafka klaszter
- Slurm klaszter

2. Lépés: Leírók letöltése a virtuális gépre #2

Címlap

TensorFlow, Keras, Jupyter Notebook GPU Stack

Áttekintés:

A TensorFlow egy nyílt forráskódú platform gépi tanuláshoz. Átfogó, rugalmas eszközeinek, könyvtárainak és közösségi erőforrásainak ökoszisztémája lehetővé teszi a kutatók számára, hogy az ML-ben a legmodernebbek legyenek, és a fejlesztők könnyen fejlesszenek és futtassanak ML-alapú alkalmazásokat. A TensorFlow-t a Google Brain csapata fejlesztette ki a Google belső használatához. 2015. november 9-én adták ki az Apache licenz 2.0 alatt. További információkért látogasson el a hivatalos TensorFlow oldalra.

Keras egy magas szintű neurális hálózatok API, amelyet Pythonban írtak és képes TensorFlow, CNTK vagy Theano felett futni. Kifejlesztették a gyors kísérletezés lehetőségét. A jó kutatás elvégzésének kulcsa az, hogy az ötletből a megoldásig a lehető legkisebb kísérlettel tudjunk menni. A Keras számos általánosan használt neurális hálózatépítő blokk, például rétegek, célok, aktiválási funkciók, optimalizálók és számos olyan eszközt tartalmaz, amely megkönnyíti a kép- és szövegadatok kezelését. A standard neurális hálózatokon kívül a Keras támogatja a konvolúciós és ismétlődő neurális hálózatokat is. Támogatja a többi segéd réteget, például a lemorzsolódást, a kötegelt normalizálást és az összevonást. További információ a Keras hivatalos lapján található.

A teljes gépi tanulási környezet a következő komponensekből áll: Jupyter Notebook, Keras, TensorFlow egy GPU kártya teljesítményét felhasználva.

Fontos: Ha szeretné használni ezt a bemutatót, a virtuális gépnek rendelkeznie kell egy hozzácsatolt NVIDIA GPU kártyával. Ha módosítani szeretné a CUDA illesztőprogramot, nyugodtan személyre szabhatja az install-cuda.sh szkriptet a nodes/cloud_init_jupyter_server_gpu.yaml fájlban.

Használati és telepítési útmutató:

<https://occpus.readthedocs.io/en/latest/tutorial-bigdata-ai.html#tensorflow-and-keras-with-jupyter-notebook-stack-using-nvidia-gpu-card>

Features

- creating a node through contextualisation
- utilising health check against a predefined port

Prerequisites

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

Download

You can download the example as [tutorial.examples.tensorflow-keras-jupyter-gpu](https://occpus.readthedocs.io/en/latest/tutorial-bigdata-ai.html#tensorflow-and-keras-with-jupyter-notebook-stack-using-nvidia-gpu-card).

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.

3. Lépés: Tűzfalszabályok létrehozása

| <input type="checkbox"/> | Direction | Ether Type | IP Protocol | Port Range | Remote IP Prefix |
|--------------------------|-----------|------------|-------------|------------|------------------|
| <input type="checkbox"/> | Egress | IPv4 | Any | Any | 0.0.0.0/0 |
| <input type="checkbox"/> | Egress | IPv6 | Any | Any | ::/0 |
| <input type="checkbox"/> | Ingress | IPv4 | TCP | 22 (SSH) | 0.0.0.0/0 |
| <input type="checkbox"/> | Ingress | IPv4 | TCP | 8888 | 0.0.0.0/0 |

} Kimenő forgalom

→ SSH hozzáférés

→ JupyterLab hozzáférés

Displaying 4 items

4. Lépés: Leírók személyre szabása a virtuális gépen

- ▶ Infrastruktúra definíciós fájl
 - ▶ JupyterLab jelszó megadás
 - ▶ Alapértelmezett jelszó:
 - ▶ tensorflow

```
infra_name: tensorflow
user_id: somebody@somewhere

variables:
  JUPYTER_PASSWORD: tensorflow

nodes:
  - &M
    name: tensorflow
    type: tensorflow_node
```

4. Lépés: Leírók személyre szabása a virtuális gépen #2

- ▶ Ubuntu 20.04-es alap képfájl használata
 - ▶ Docker CE-t előre telepítve tartalmaz

Images / Ubuntu 20.04 LTS

| | |
|-------------|--|
| Name | Ubuntu 20.04 LTS |
| Description | Ubuntu 20.04 LTS Cloud image with Docker. username: ubuntu, password: ubuntu |
| ID | 88bafb03-b169-4289-8f6f-c0cfc9177ca |

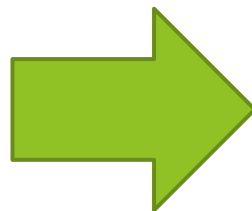
- ▶ GPU erőforrást is tartalmazó flavour kiválasztása

| Flavor Details: oktatas.k80 | |
|-----------------------------|--------------------------------------|
| ID | b0af1664-0d30-4339-90ee-8638b00b832c |
| VCPUs | 4 |
| RAM | 8GB |
| Size | 50GB |

4. Lépés: Leírók személyre szabása a virtuális gépen #3

► Csomópont definíciós fájl (nodes/node_definition.yaml)

```
'node_def:tensorflow_node':  
-  
  resource:  
    type: nova  
    endpoint: replace_with_endpoint_of_nova_interface_of_your_cloud  
    project_id: replace_with_projectid_to_use  
    user_domain_name: Default  
    image_id: replace_with_id_of_your_image_on_your_target_cloud  
    network_id: replace_with_id_of_network_on_your_target_cloud  
    server_name: tensorflow-{{cut(infra_id,0,4)}}  
    flavor_name: replace_with_id_of_the_flavor_on_your_target_cloud  
    key_name: replace_with_name_of_keypair_or_remove  
    security_groups:  
      - replace_with_security_group_to_add_or_remove_section  
    floating_ip: add_yes_if_you_need_floating_ip_or_remove  
  contextualisation:  
    type: cloudinit  
    context_template: !yaml_import  
      url: file://cloud_init_tensorflow.yaml  
  health_check:  
    ports:  
      - 8888  
    timeout: 3600  
  ...
```



```
'node_def:tensorflow_node':  
-  
  resource:  
    type: nova  
    endpoint: https://sztaki.cloud.mta.hu:5000/v3  
    project_id: a9c30db63ddf47a98045ef9c726c7436  
    user_domain_name: Default  
    image_id: 88bafb03-b169-4289-8f6f-c0cfc9177ca  
    network_id: 01efee1c-858c-4047-a48a-e2fab056f82a  
    server_name: tensorflow-{{cut(infra_id,0,4)}}  
    flavor_name: b0af1664-0d30-4339-90ee-8638b00b832c  
    key_name: key_name  
    security_groups:  
      - a52445ab-ab61-4eff-9d71-948285106d2f  
    floating_ip: yes  
  contextualisation:  
    type: cloudinit  
    context_template: !yaml_import  
      url: file://cloud_init_tensorflow.yaml  
  health_check:  
    ports:  
      - 8888  
    timeout: 3600  
  ...
```

5. - 6. Lépés: Occopus aktiválása és leírók importálása

► 5. Lépés: Occopus aktiválása

```
ubuntu@occo:~/tensorflow$ source $HOME/occopus/bin/activate  
(occopus) ubuntu@occo:~/tensorflow$
```

► 6. Leírók importálása Occopus számára

```
$ occopus-import nodes/node_definitions.yaml  
Successfully imported nodes: tensorflow_node
```

7. Lépés: Infrastruktúra kiépítése

```
$ occopus-build infra-tensorflow.yaml
** 2021-06-27 11:01:56,835    Creating node 'tensorflow/'d72fd6b6-12cf-4f43-b309-4fc13594a4ce'
...
** 2021-06-27 11:23:41,223    Health checking for node 'tensorflow/'d72fd6b6-12cf-4f43-b309-4fc13594a4ce'
** 2021-06-27 11:23:42,405    Checking node reachability (d72fd6b6-12cf-4f43-b309-4fc13594a4ce):
** 2021-06-27 11:23:42,418    193.224.59.182 => ready
** 2021-06-27 11:23:42,418    Checking port availability (d72fd6b6-12cf-4f43-b309-4fc13594a4ce):
** 2021-06-27 11:23:42,423    8888 => ready
** 2021-06-27 11:23:42,424    Health checking result: ready
** 2021-06-27 11:23:43,560    Submitted infrastructure: '249ba186-4333-4f6b-b8ec-58e72d96a4d4'
** 2021-06-27 11:23:43,585    List of nodes/instances/addresses:
** 2021-06-27 11:23:43,585    tensorflow:
** 2021-06-27 11:23:43,585    d72fd6b6-12cf-4f43-b309-4fc13594a4ce:
** 2021-06-27 11:23:43,585    193.224.59.182
249ba186-4333-4f6b-b8ec-58e72d96a4d4
```

7. Lépés: Infrastruktúra kiépítése #2

Instances

Instance Name = Filter Launch Instance Delete Instances More Actions

| Instance Name | Image Name | IP Address | Size | Key Pair | Status | Availability Zone | Task | Power State | Time since created | Actions |
|--|------------------|--|-------------|----------|--------|-------------------|----------|-------------|--------------------|---------------------------------|
| <input type="checkbox"/> tensorflow-249b | Ubuntu 20.04 LTS | 192.168.10.203 Floating IPs: 193.224.59.182 | oktatas.k80 | farkas | Build | nova | Spawning | No State | 0 minutes | Delete Floating IP |

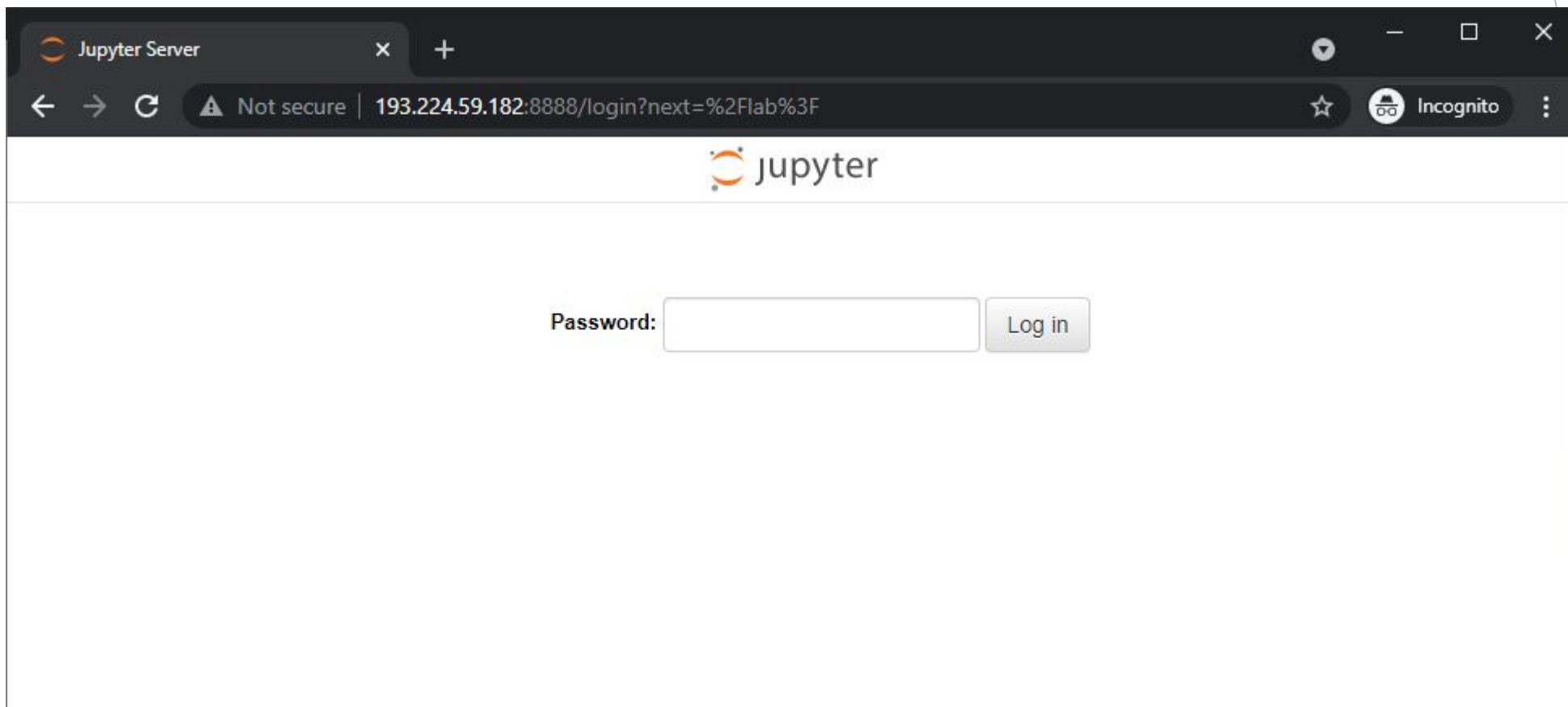
Instances

Instance Name = Filter Launch Instance Delete Instances More Actions

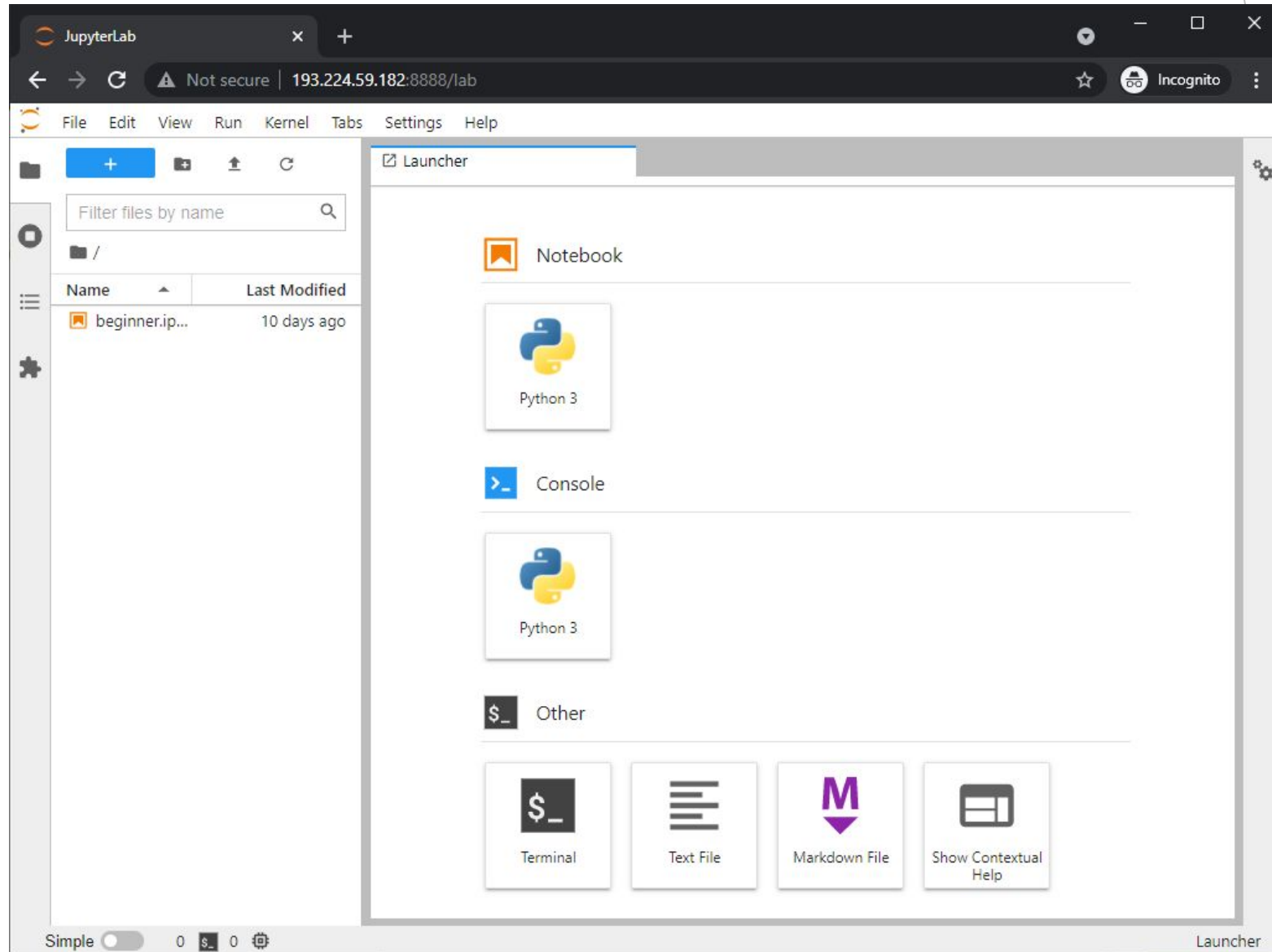
| Instance Name | Image Name | IP Address | Size | Key Pair | Status | Availability Zone | Task | Power State | Time since created | Actions |
|--|------------------|--|-------------|----------|--------|-------------------|------|-------------|--------------------|------------------------------|
| <input type="checkbox"/> tensorflow-249b | Ubuntu 20.04 LTS | 192.168.10.203 Floating IPs: 193.224.59.182 | oktatas.k80 | farkas | Active | nova | None | Running | 0 minutes | Create Snapshot |

8. Lépés: Infrastruktúra használata

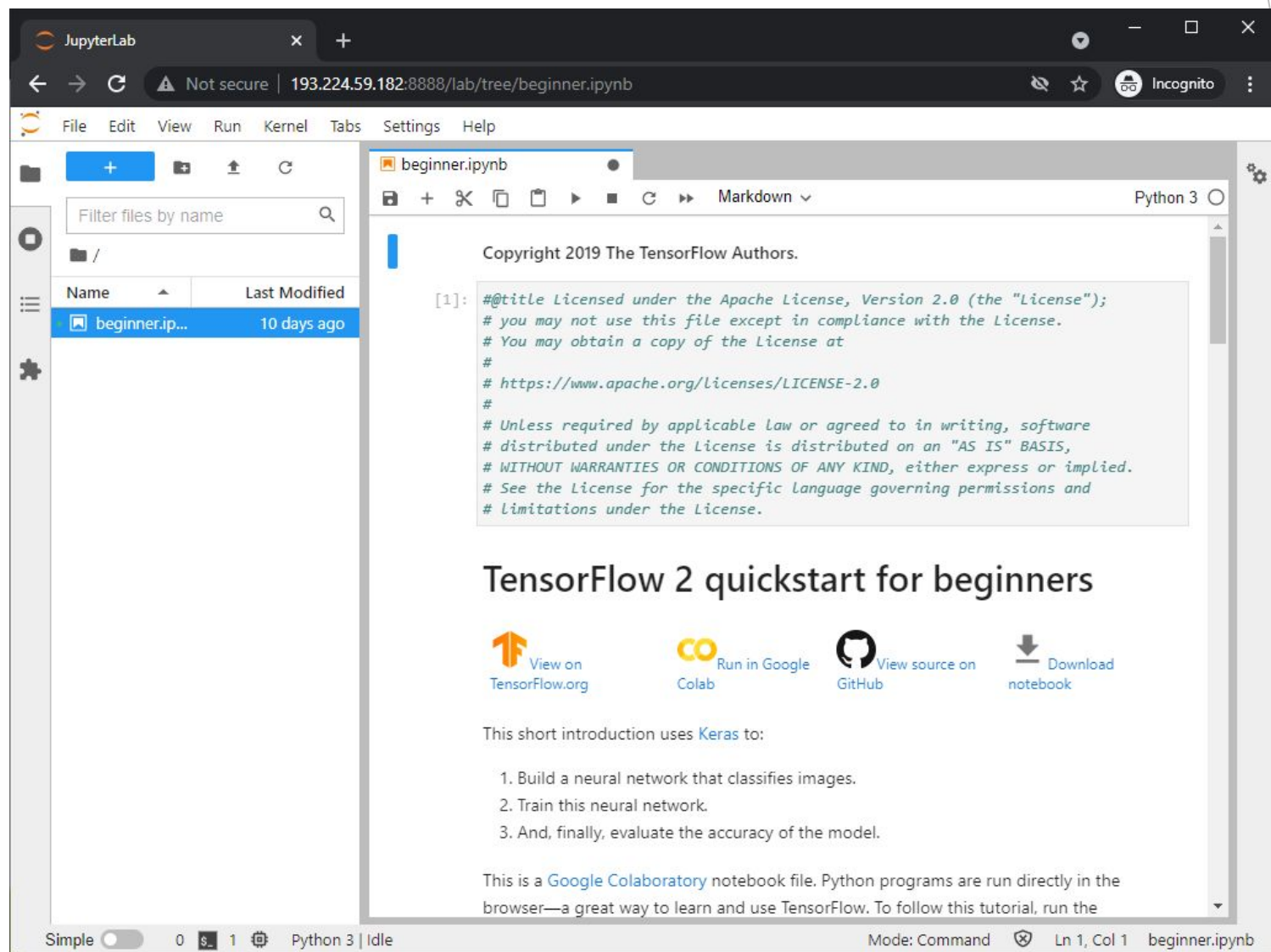
- ▶ Belépés JupyterLab felületre:
 - ▶ `http(s)://<JupyterLabIP>:8888/lab`



8. Lépés: Infrastruktúra használata #2



8. Lépés: Infrastruktúra használata #3



The screenshot displays the JupyterLab web interface. The browser address bar shows the URL `193.224.59.182:8888/lab/tree/beginner.ipynb`. The interface includes a file browser on the left, a main editor area, and a status bar at the bottom.

The notebook content is as follows:

```
Copyright 2019 The TensorFlow Authors.
```

```
[1]: ##@title Licensed under the Apache License, Version 2.0 (the "License");  
# you may not use this file except in compliance with the License.  
# You may obtain a copy of the License at  
#  
# https://www.apache.org/licenses/LICENSE-2.0  
#  
# Unless required by applicable law or agreed to in writing, software  
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.
```



TensorFlow 2 quickstart for beginners

[View on TensorFlow.org](#) [Run in Google Colab](#) [View source on GitHub](#) [Download notebook](#)

This short introduction uses [Keras](#) to:

1. Build a neural network that classifies images.
2. Train this neural network.
3. And, finally, evaluate the accuracy of the model.

This is a [Google Colaboratory](#) notebook file. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To follow this tutorial, run the

Simple 0 `$` 1  Python 3 | Idle Mode: Command  Ln 1, Col 1 beginner.ipynb

Infrastruktúra törlése

► Infrastruktúra ID lekérése

```
$ occopus-maintain -l  
Using default configuration file: '/root/.occopus/occopus_config.yaml'  
** 2021-06-27 11:36:28,009 Starting up; PID = 186  
List of active infrastructure:  
249ba186-4333-4f6b-b8ec-58e72d96a4d4
```

► Infrastruktúra törlése

```
$ occopus-destroy -i 249ba186-4333-4f6b-b8ec-58e72d96a4d4  
Using default configuration file: '/root/.occopus/occopus_config.yaml'  
** 2021-06-27 11:37:17,094 Starting up; PID = 191  
** 2021-06-27 11:37:17,096 Start dropping infrastructure 249ba186-4333-4f6b-b8ec-58e72d96a4d4  
** 2021-06-27 11:37:17,116 Dropping node 'tensorflow/'d72fd6b6-12cf-4f43-b309-4fc13594a4ce'  
** 2021-06-27 11:37:20,998 Finished dropping infrastructure 249ba186-4333-4f6b-b8ec-58e72d96a4d4
```

Összefoglalás



- ▶ Mélytanulást támogató referencia architektúra kiépítése
 - ▶ GPU erőforrások használata
- ▶ TensorFlow és JupyterLab alkalmazása
- ▶ ELKH Cloud technikai támogatás:
info@science-cloud.hu



ELKH Cloud

Köszönöm a figyelmet!