

Prepare your TENSORFLOW resource on Saturn Cloud

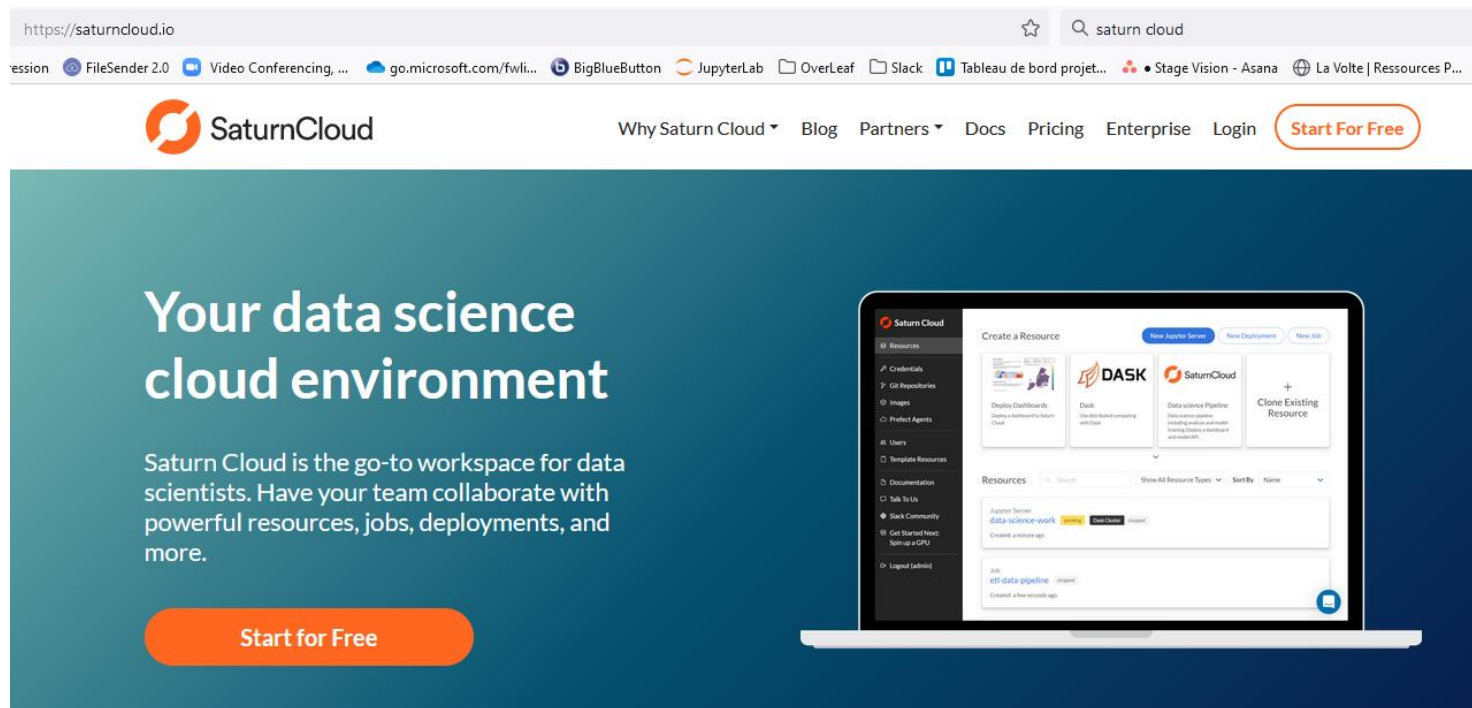
Hands-on 1 Classification
Hands-on 2 Segmentation

Steps required

1. Create an account on a cloud platform (Saturn Cloud)
2. Create a jupyter server and parametrize it on this cloud
3. Start the jupyter server
4. Launch the Jupyter lab environment

1- Create an account (if not already done !)

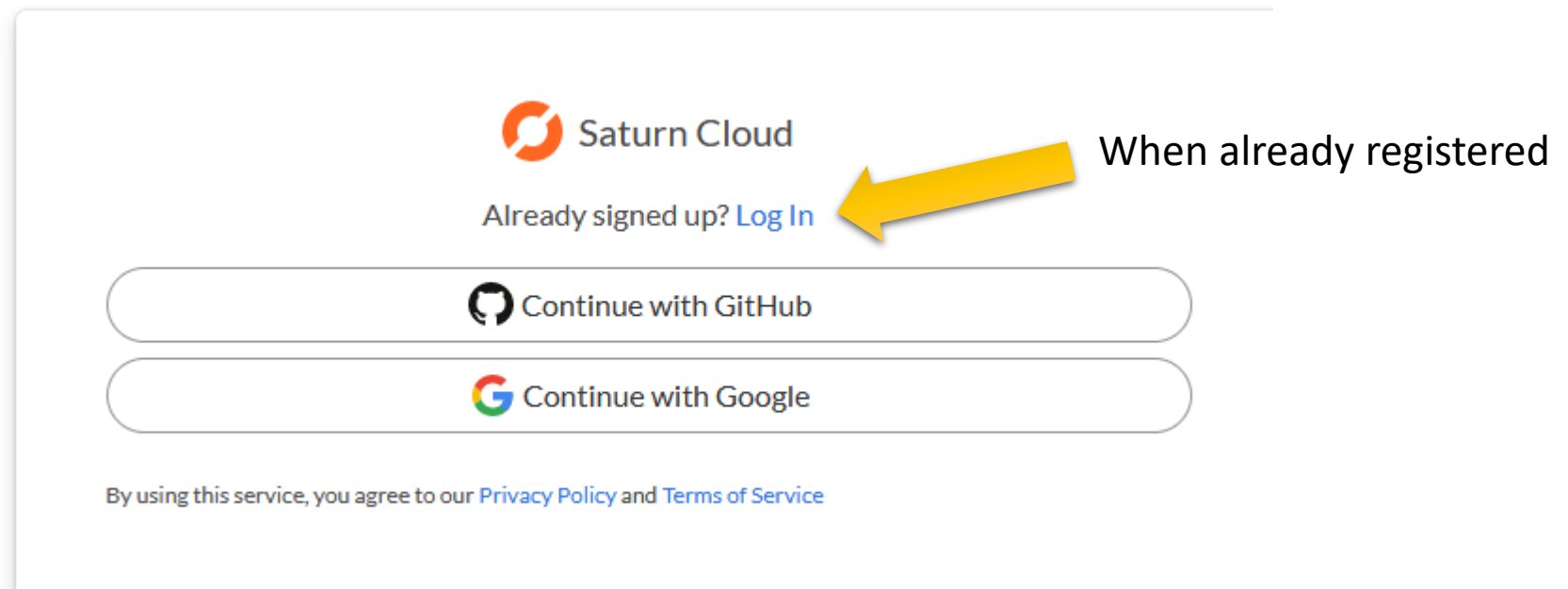
- It's free, « just » need an email adress
- Connect on <https://saturncloud.io> and clic on « Start for Free »



The screenshot shows the Saturn Cloud website. The browser address bar is <https://saturncloud.io>. The page header includes the Saturn Cloud logo, navigation links for 'Why Saturn Cloud', 'Blog', 'Partners', 'Docs', 'Pricing', 'Enterprise', 'Login', and a 'Start For Free' button. The main content area features the text 'Your data science cloud environment' and a 'Start for Free' button. A laptop in the foreground displays the Saturn Cloud interface, showing a 'Create a Resource' dialog with options for DASK, SaturnCloud, and Clone Existing Resource, and a list of existing resources.

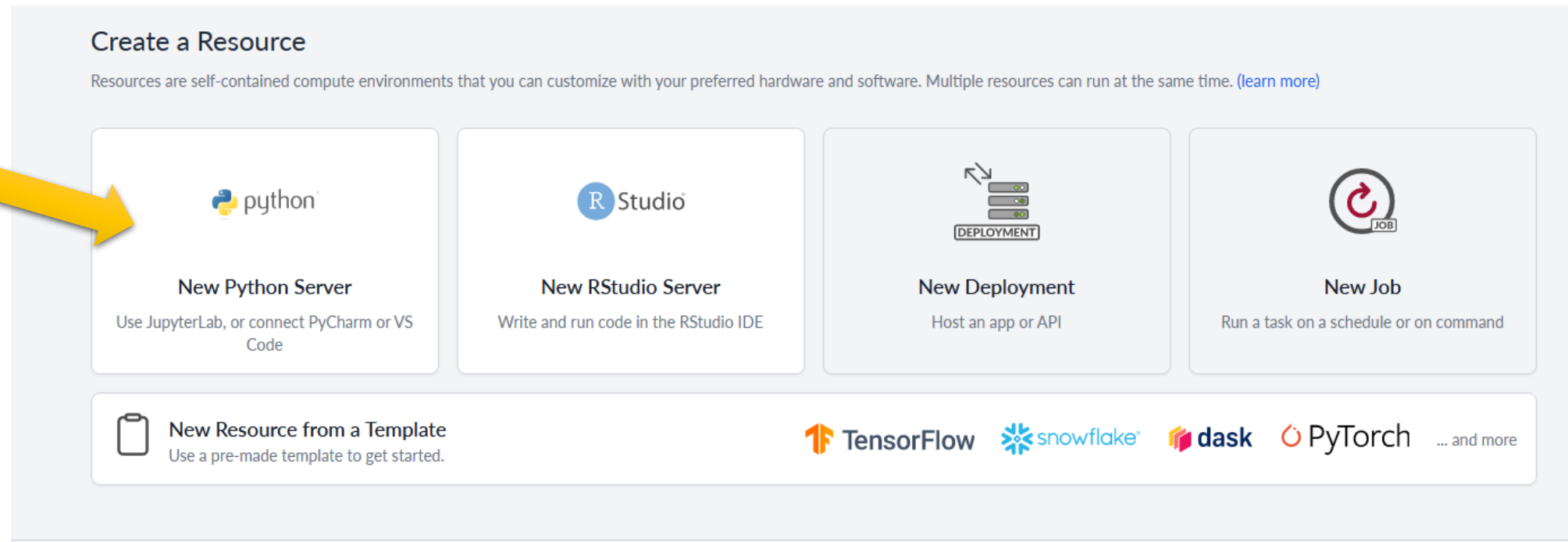
1- create an account

- You can use one of your GitHub or Google account
- Optionnaly, you can create a Saturn Cloud account



2- Create your jupyter server

- Hands-on are Jupyter notebooks, so create first a new Python server on Saturn Cloud



Create a Resource
Resources are self-contained compute environments that you can customize with your preferred hardware and software. Multiple resources can run at the same time. [\(learn more\)](#)

New Python Server
Use JupyterLab, or connect PyCharm or VS Code

New RStudio Server
Write and run code in the RStudio IDE

New Deployment
Host an app or API

New Job
Run a task on a schedule or on command

New Resource from a Template
Use a pre-made template to get started.

TensorFlow snowflake dask PyTorch ... and more

2- Server Parameters 1/4

Give a name (TF-Hands-On is nice)



The screenshot shows a configuration form for a Jupyter server. At the top left, there is a rocket icon and the word "Overview". At the top right, there is a link "Hide Advanced Options". Below this, there are two columns: "Owner" and "Name". The "Owner" field contains the text "thomasgre /". The "Name" field is a text input box containing "ML-HandsOn-2". A yellow arrow points from the rocket icon to the "Name" field. Below the "Name" field is a "Description" section with a large text area. At the bottom of the description area, there is a character count: "Briefly describe this Jupyter server. (Characters left: 255/255)".

2- Server Parameters 2/4

- Select CPU or **GPU** resources



Hardware

The hardware your Jupyter server will run on.

Hardware

CPU

An instance with only CPU processors.



GPU

An instance with both CPU and GPU processors.

Size

Large - 2 cores - 16 GB RAM

Disabled options are not supported due to your account limit. To increase the limit, please



Hardware

The hardware your Jupyter server will run on.

Hardware

CPU

An instance with only CPU processors.

GPU

An instance with both CPU and GPU processors.



Size

T4-XLarge - 4 cores - 16 GB RAM - 1 GPU

Disabled options are not supported due to your account limit. To increase the limit, please

2- Server Parameters 3/4

- Select the desired image : saturncloud/saturn-python-tensorflow for GPU
- For version, use the 2022.06.01 or, if not available, the latest available


 **Environment** [Show Advanced Options](#)
The software your Jupyter server will use. This includes libraries, packages, environment variables, and other attributes.

Image **Version**

saturncloud/saturn-python-tensorflow 2022.06.01

Extra Packages
Extra packages are installed every time the resource starts up - right before the start script. Use spaces to separate packages.
If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)

Conda	Pip	Apt
-------	-----	-----

2- Server Parameters 4/4



ATTENTION

→ **Pip Install**, add the following
opencv-python

→ **Apt Packages** add the following
htop zip unzip python3-opencv

Both must be done

Extra Packages
Extra packages are installed every time the resource starts up - right before the start script. Use spaces to separate packages. If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)

Conda Install Pip Install Apt Packages

```
htop zip unzip python3-opencv
```

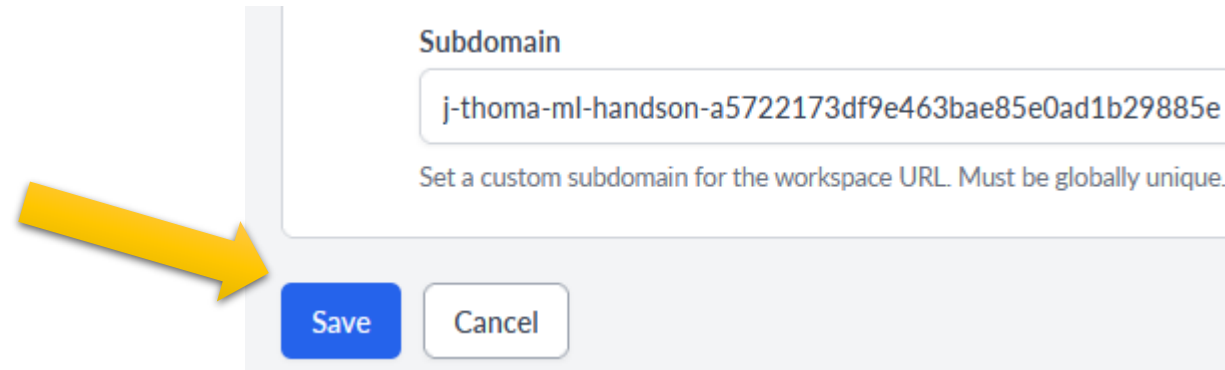
The packages together will run the following script:

```
apt-get install htop zip unzip python3-opencv  
pip install opencv-python
```

Environment Variables

```
name = value
```

2- Server Parameters : Save



Subdomain

Set a custom subdomain for the workspace URL. Must be globally unique.

Save Cancel

3- Start the jupyter server (few minutes)

Jupyter Server
thomasgre / ML-HandsOn

Image: saturncloud/saturn-tensorflow:2021.07.26-2
Working Directory: /home/jovyan/git-repos
Extra Packages:
`apt-get install htop python3-opencv`
`pip install opencv-python`

Edit Delete

Logs

Resource Details

Jupyter Server

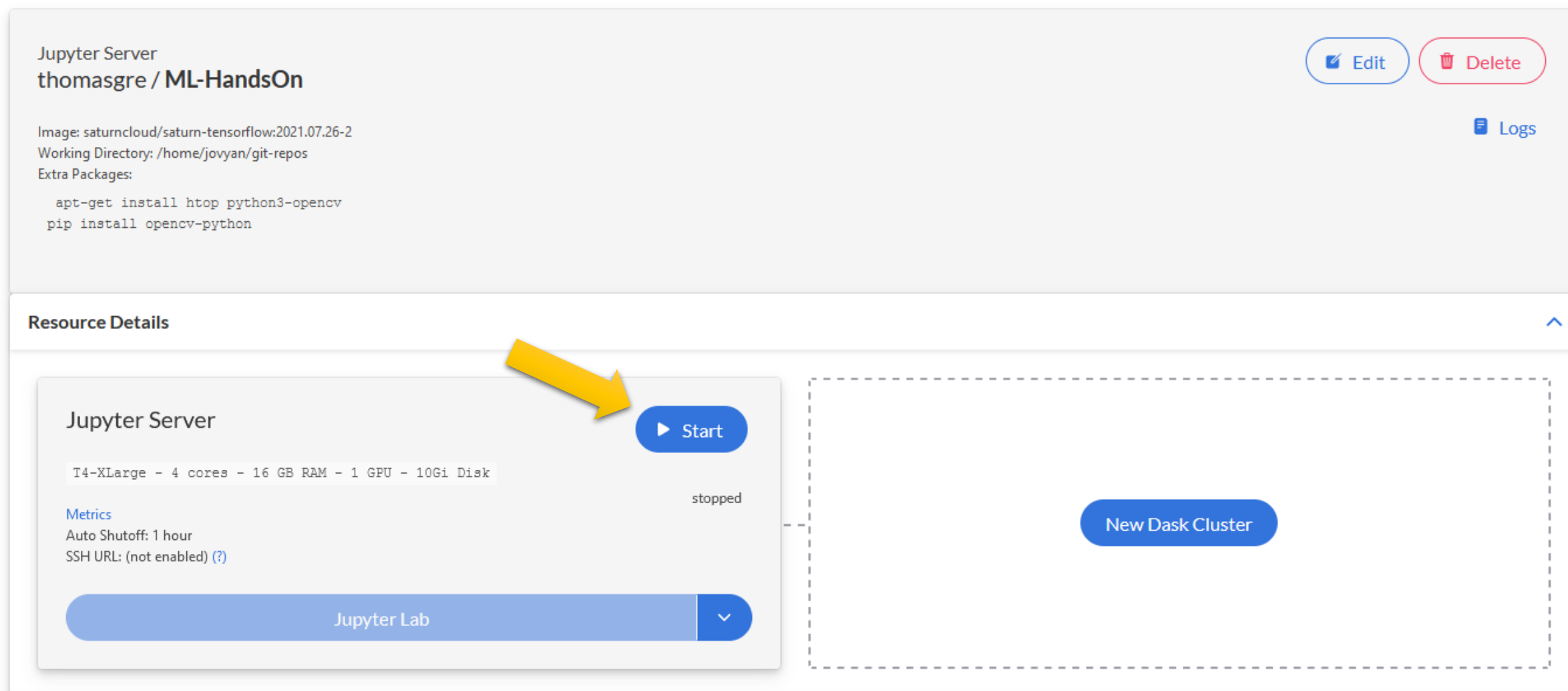
T4-XLarge - 4 cores - 16 GB RAM - 1 GPU - 10Gi Disk

stopped

Metrics
Auto Shutoff: 1 hour
SSH URL: (not enabled) (?)

Jupyter Lab

New Dask Cluster



4- Launch the Jupyter lab environment

- First, wait until the server is started 😊



5- Enjoy your Hands on (in a new tab)

The image shows a JupyterLab interface. On the left is a file browser for the directory `/git-repos/`. It has a table with columns for 'Name' and 'Last Modified'. A red arrow points to the file browser area with the text "Drag and drop files here (or via the upload button)". On the right is the 'Launcher' panel, which contains several options: 'Notebook', 'Console', and 'Other'. Under 'Notebook', there are two 'saturn (Python 3)' options. Under 'Other', there are icons for 'Terminal', 'Python File', 'Text File', 'Markdown File', and 'Show Contextual Help'. A green arrow points to the 'Launcher' tab in the top bar with the text "Launcher tab".

Hands-on with , download zip files :

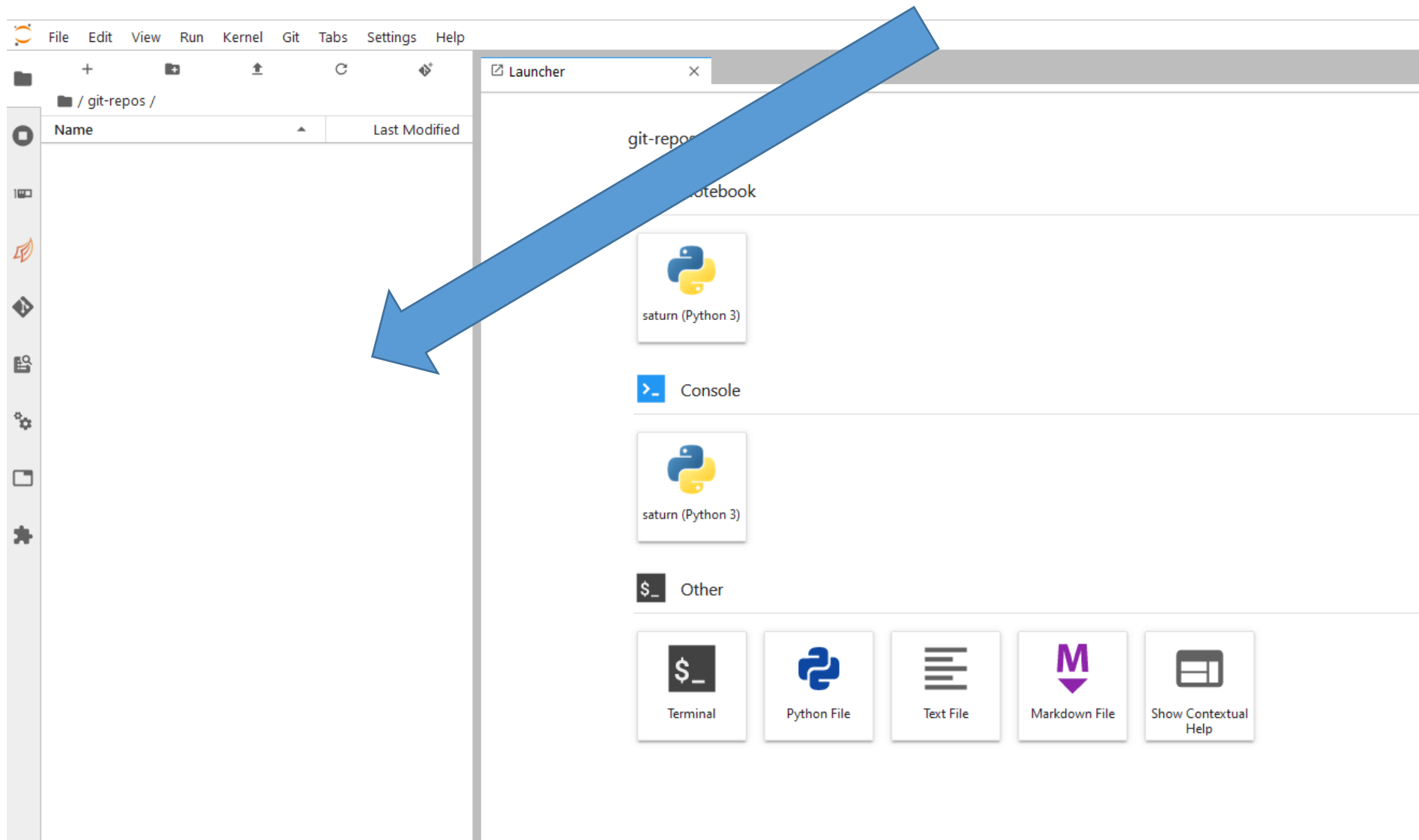
Classification (GPU recommended):

https://gitlab.in2p3.fr/thomas.grenier/tp1ss_classification/-/raw/master/TP_Classification_v03.zip

Segmentation (GPU needed):

https://gitlab.in2p3.fr/thomas.grenier/tp4ss_segmentation/-/raw/master/TP_Segmentation_v03.zip

Drag and drop zip file in the file browser area



Open a Terminal

The image shows the JupyterLab Launcher interface. On the left is a file browser showing the workspace path `/ workspace / Classification /`. It contains a table with the following entries:

Name	Last Modified
<code>.ipynb_checkpoints</code>	2 minutes ago
<code>TP_Classification_v03.zip</code>	12 hours ago

The main area is the Launcher, which is currently in the `workspace/Classification` directory. It features several sections:

- Notebook**: A section containing a `saturn (Python 3)` icon.
- Console**: A section containing another `saturn (Python 3)` icon.
- Other**: A section containing five icons: a terminal icon (labeled "Terminal" and "Start a new terminal session"), a list icon, a markdown icon (labeled "Markdown File"), a Python icon (labeled "Python File"), and a help icon (labeled "Show Contextual Help").

A large blue arrow points from the bottom left towards the terminal icon in the "Other" section.

Unzip the material

Enter the command line: `unzip TP_Classification_v03.zip`
then « enter »

The image displays two screenshots of a JupyterLab interface, illustrating the process of unzipping a file. A blue arrow on the left points from the top screenshot to the bottom one.

Top Screenshot: The terminal window shows the command `unzip TP_Classification_v03.zip` being entered in the `Classification/` directory.

```
jovyan@w-dlmi2-tensorflow-handson-c61a1ab2154b45699816594d0d3b964f7rfb:~/workspace$ cd Classification/  
jovyan@w-dlmi2-tensorflow-handson-c61a1ab2154b45699816594d0d3b964f7rfb:~/workspace/Classification$ unzip TP_Classification_v03.zip
```

Bottom Screenshot: The terminal window shows the output of the `unzip` command, listing the files extracted from the archive:

```
jovyan@w-dlmi2-tensorflow-handson-c61a1ab2154b45699816594d0d3b964f7rfb:~/workspace$ cd Classification/  
jovyan@w-dlmi2-tensorflow-handson-c61a1ab2154b45699816594d0d3b964f7rfb:~/workspace/Classification$ unzip TP_Classification_v03.zip  
Archive: TP_Classification_v03.zip  
  inflating: 01_LoadData.ipynb  
  inflating: 02_Classification_TF22.ipynb  
  inflating: 03_Interpretability_TF22.ipynb  
  inflating: 04_KnowledgeDistillation_TF22.ipynb  
  inflating: 09_GradAndAutoGrad.ipynb  
  inflating: tools.py  
  inflating: figures/architecture_cnn_en.png  
  inflating: figures/architecture_fcn_en.png  
  inflating: figures/architecture_resnet_en.png  
  inflating: figures/decision_tree.png  
  inflating: figures/Figure1_SagittalAxialCoronal_small.png  
  inflating: figures/ResNet_TransferLearning.png  
  inflating: figures/TensorBoard.png  
jovyan@w-dlmi2-tensorflow-handson-c61a1ab2154b45699816594d0d3b964f7rfb:~/workspace/Classification$
```

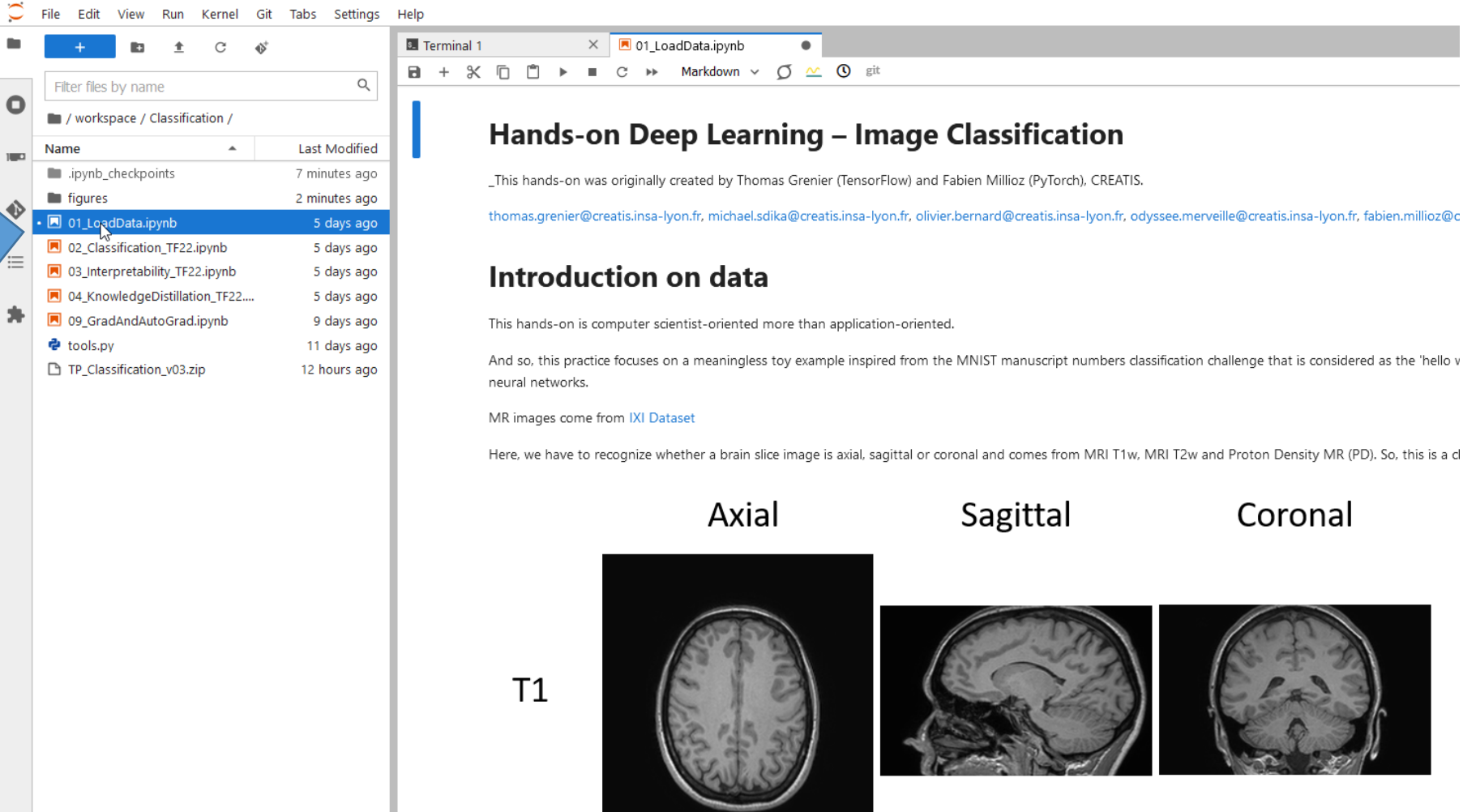
The file explorer on the left shows the contents of the `Classification/` directory, including the unzipped files and a `figures` directory.

Name	Last Modified
.ipynb_checkpoints	3 minutes ago
TP_Classification_v03.zip	12 hours ago

Name	Last Modified
.ipynb_checkpoints	5 minutes ago
figures	seconds ago
01_LoadData.ipynb	5 days ago
02_Classification_TF22.ipynb	5 days ago
03_Interpretability_TF22.ipynb	5 days ago
04_KnowledgeDistillation_TF22....	5 days ago
09_GradAndAutoGrad.ipynb	9 days ago
tools.py	11 days ago
TP_Classification_v03.zip	12 hours ago

Select and open the desired notebook

As example, double clic on : 01_LoadData.ipynb



The screenshot shows the JupyterLab interface. On the left, the file browser displays a list of files and folders in the workspace. A blue arrow points to the file '01_LoadData.ipynb'. The main area shows the notebook content, which includes the title 'Hands-on Deep Learning – Image Classification' and the text 'Introduction on data'. Below the text, there are three MRI brain slice images labeled 'Axial', 'Sagittal', and 'Coronal'. The label 'T1' is positioned to the left of the Axial image.

Name	Last Modified
.ipynb_checkpoints	7 minutes ago
figures	2 minutes ago
01_LoadData.ipynb	5 days ago
02_Classification_TF22.ipynb	5 days ago
03_Interpretability_TF22.ipynb	5 days ago
04_KnowledgeDistillation_TF22....	5 days ago
09_GradAndAutoGrad.ipynb	9 days ago
tools.py	11 days ago
TP_Classification_v03.zip	12 hours ago

Hands-on Deep Learning – Image Classification

_This hands-on was originally created by Thomas Grenier (TensorFlow) and Fabien Millioz (PyTorch), CREATIS.

thomas.grenier@creatis.insa-lyon.fr, michael.sdika@creatis.insa-lyon.fr, olivier.bernard@creatis.insa-lyon.fr, odyssee.merveille@creatis.insa-lyon.fr, fabien.millioz@creatis.insa-lyon.fr

Introduction on data

This hands-on is computer scientist-oriented more than application-oriented.

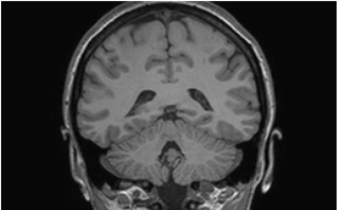
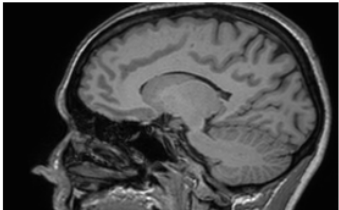
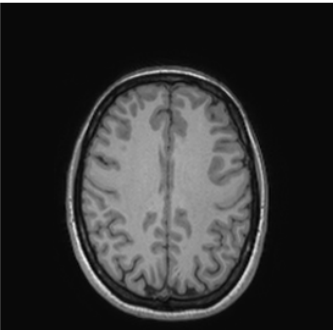
And so, this practice focuses on a meaningless toy example inspired from the MNIST manuscript numbers classification challenge that is considered as the 'hello world' of neural networks.

MR images come from [IXI Dataset](#)

Here, we have to recognize whether a brain slice image is axial, sagittal or coronal and comes from MRI T1w, MRI T2w and Proton Density MR (PD). So, this is a classification task.

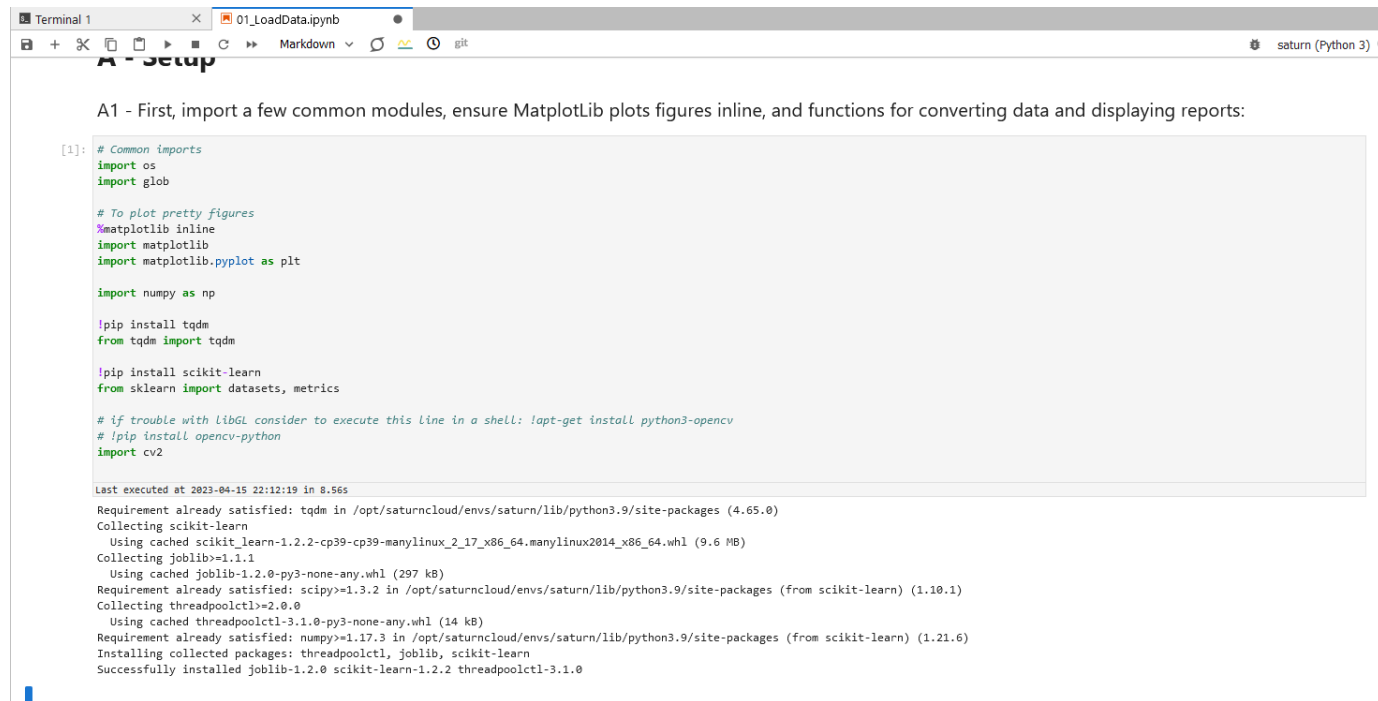
Axial **Sagittal** **Coronal**

T1



Cells and running

- Notebooks are made of cells : can be text or python code
- To go to next cell use the play button or « shift+enter »



A1 - First, import a few common modules, ensure Matplotlib plots figures inline, and functions for converting data and displaying reports:

```
[1]: # Common imports
import os
import glob

# To plot pretty figures
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt

import numpy as np

!pip install tqdm
from tqdm import tqdm

!pip install scikit-learn
from sklearn import datasets, metrics

# if trouble with libGL consider to execute this line in a shell: !apt-get install python3-opencv
# !pip install opencv-python
import cv2
```

Last executed at 2023-04-15 22:12:19 in 8.56s

Requirement already satisfied: tqdm in /opt/saturncloud/envs/saturn/lib/python3.9/site-packages (4.65.0)
Collecting scikit-learn
Using cached scikit_learn-1.2.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.6 MB)
Collecting joblib=1.1.1
Using cached joblib-1.2.0-py3-none-any.whl (297 kB)
Requirement already satisfied: scipy>=1.3.2 in /opt/saturncloud/envs/saturn/lib/python3.9/site-packages (from scikit-learn) (1.10.1)
Collecting threadpoolctl=2.0.0
Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
Requirement already satisfied: numpy>=1.17.3 in /opt/saturncloud/envs/saturn/lib/python3.9/site-packages (from scikit-learn) (1.21.6)
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.2.0 scikit-learn-1.2.2 threadpoolctl-3.1.0