Introduction + transforms API

PyTorch BIVL²ab

Agenda

- Use the BIVL²ab server
- Set up an environment for the course
- Access the course resources
- `transforms` API

Use the BIVL²ab server

Creating an account

Sign up

To sign up or create an account in the cluster please deliver an email to:

- erangp00@estudiantes.unileon.es
- famarcar@saber.uis.edu.co (in the CC)

The subject must be **Sign Up to BivL2ab Cluster** (in English and literally like is shown here).

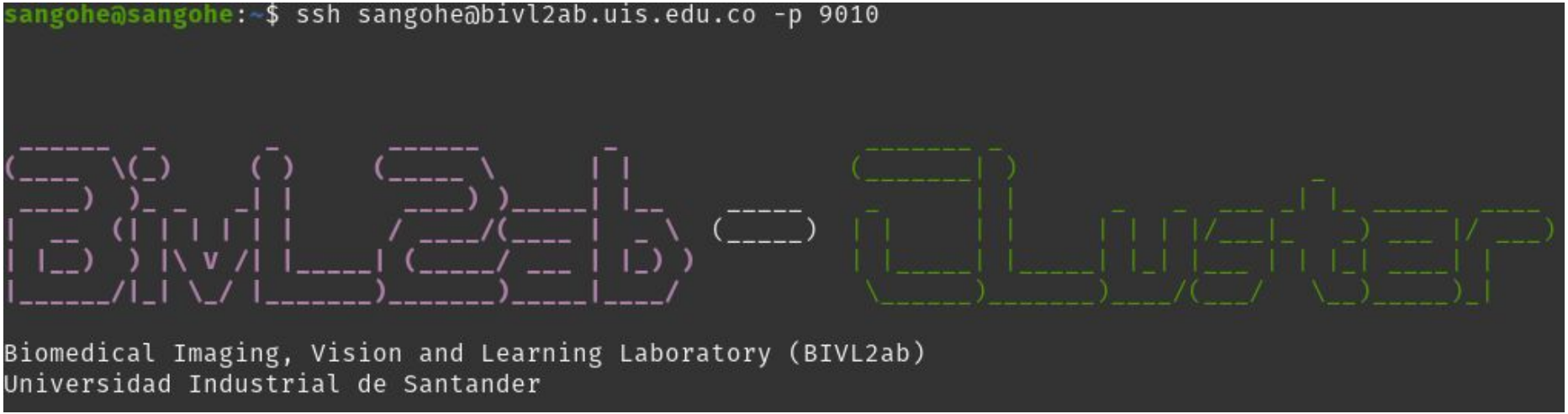
And the content of message must contain the data show below:

- Username (Must be in lowercase).
- Full name
- CC or passport number.
- Cell phone with whatsapp.
- Role (Undergraduate, Magister, Doctor, Researcher, External).
- [Optional] Desired port (See Available ports).

Accessing the server via SSH

It's really simple, the command in general is:

```
ssh <your_username>@bivl2ab.uis.edu.co -p 9010
```



More information

Check out the detailed guide on [how to use the server], made by the server master itself.

Set up an environment for the course

Useful commands

One of the first things you'll notice after accessing the cluster is the information. Pay close attention to the commands at the bottom.

```
================================================================================
Para cualquier inquietud con el servidor puedes consultar la biblia en:
https://docs.google.com/presentation/d/1nGQlwlPUTUx4YnGXETGOxDKdXL1gQxMpIwYMbghB76k/edit?

Para revisar el cronograma de uso de las GPUs puedes consultarlo en:
https://calendar.google.com/calendar/embed?src=c_5rrgq63fpgitqfcjih6hujfefg%40group.calen

Para solicitar un puerto web, recuerda inscribirlo en este documento:
https://docs.google.com/spreadsheets/d/18C2F_b-o8qUx1kWuCNa0i6bvgUcC49-LNEjuUTZkEow/edit?
================================================================================

Te recuerdo algunos comandos utiles para usar en la consola:
        - htop -> Muestra el estado de uso de la CPU.
        - gpus -> Muestra el estado de uso de las GPUs.
        - check_gpus -> Muestra los usuario que están usando las GPUs
        - active_dockers -> Lista los docker activos (encendidos) actualmente.
        - list_isos -> Lista las imagenes de docker disponibles.
        - create_docker -> Script que facilita la creacion de nuevos dockers.
        - run_jupyter -> Pone en ejecucion el jupyter notebook en un docker configurado.
        - create_event -> Crear un evento de GPU. (Autoaprovado)
        - check_event -> Listar los eventos de GPU
```

Creating a PyTorch container

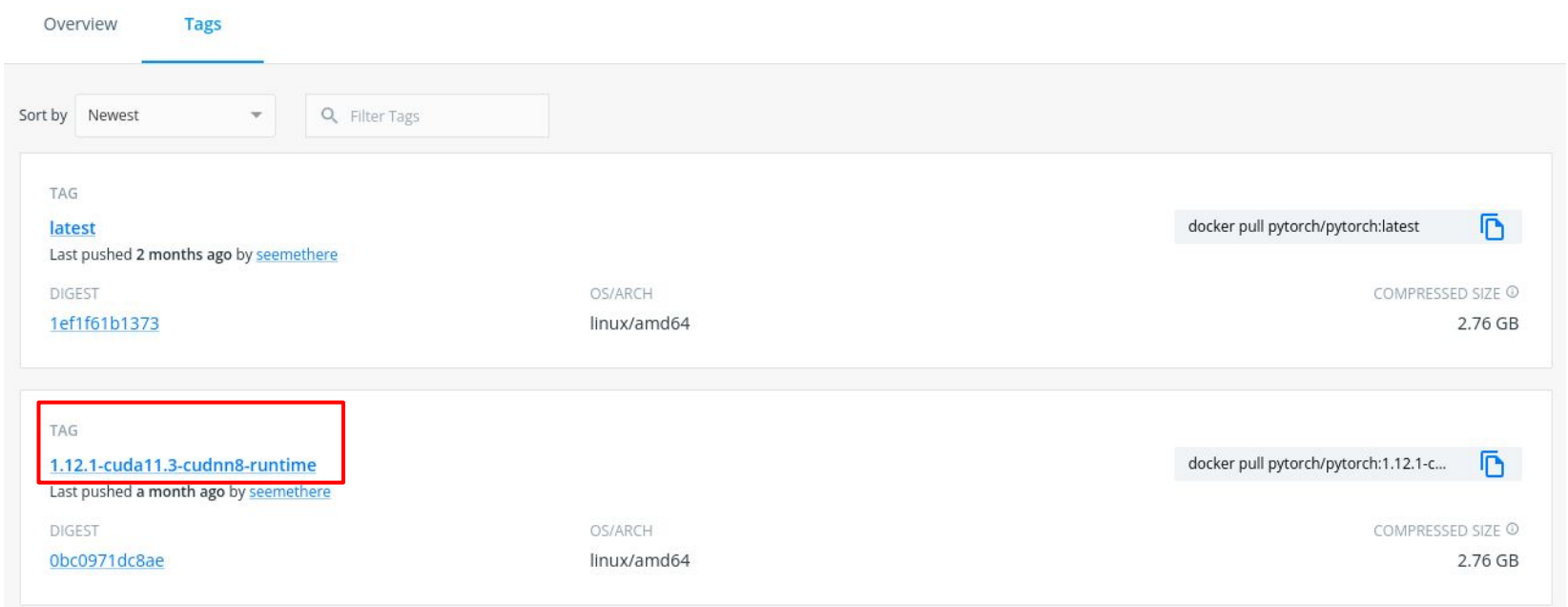
The first command we will use is `create_docker`. Type the command in the CLI to get a detailed example of how to use it.

```
sangohe@bivl2ab-cluster:~$ create_docker
usage: create_docker <image_name> <container_name> <port>
  <image_name> - Image name of the docker that will be in use (You can retrieve images names with 'list_isos')
  <container_name> - The name of your docker container (Generally is with your name)
  <port> - The port number which you will work in the server (Check the ports availables in drive)
```

Next, go to https://hub.docker.com/ and search for pytorch containers

Creating a PyTorch container

After searching, click on the tags tab and search for a version with ***cuda*** and ***cudnn runtime***.



Creating a PyTorch container

Copy the tag and execute the command in the CLI to create the docker container.

```
sangohe@bivl2ab-cluster:~$ create_docker pytorch/pytorch:1.12.1-cuda11.3-cudnn8-runtime pt_santiago 8081
```

image_name (image_name:tag) container_name port

Next, enter the container and install jupyter to have a functional IDE to work in.

```
sangohe@bivl2ab-cluster:~$ docker exec -it pt_santiago bash
root@79e260fb9b7f:/home/sangohe# pip install jupyter
```

After the installation is done, exit the container

```
root@79e260fb9b7f:/home/sangohe# exit
exit
sangohe@bivl2ab-cluster:~$
```

Exposing a Jupyter Notebook attached to the PyTorch container

Configure the container to expose a Jupyter Notebook instance on the internet. To do this, run `configure_jupyter` command and set a secure password

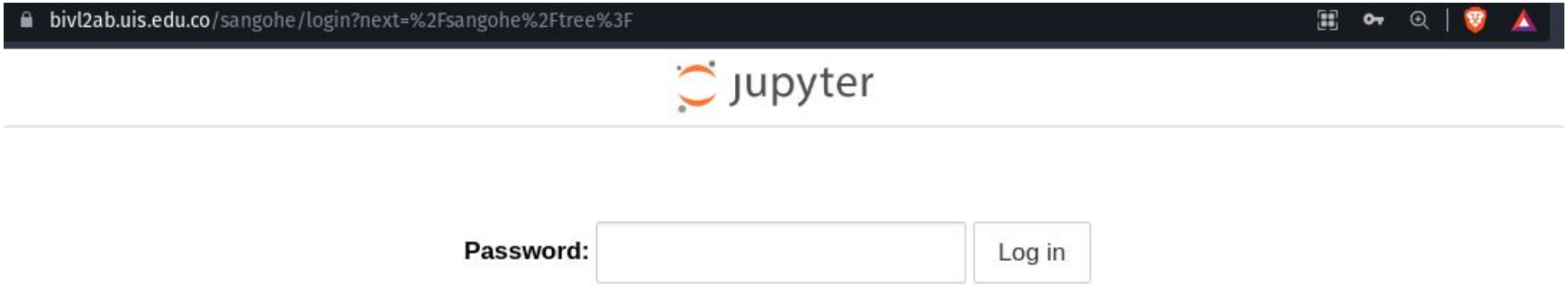
```
sangohe@bivl2ab-cluster:~$ configure_jupyter pt_santiago 8081
Creating jupyter config folder...
Success!
Now its time to configure the password for your notebook...
Enter password:
Verify password:
[NotebookPasswordApp] Wrote hashed password to /root/.jupyter/jupyter_notebook_config.json
Success!
Now your can start the jupyter server with run_jupyter command will
```

Now run the `run_jupyter` command to start the Jupyter

```
sangohe@bivl2ab-cluster:~$ run_jupyter pt_santiago
Starting your jupyter notebook in your docker...
Now enter to bivl2ab.uis.edu.co/sangohe
```

Access to the Jupyter instance

Open your web browser and go to `https://bivl2ab.uis.edu.co/<your_username>/`



Enter the password you set before to enter your workspace

Access to the Jupyter instance

The only thing left is to create a symbolic link to the ***Courses*** directory. Notice that when you execute the `ls` command, the name ***Courses*** appears in a different color

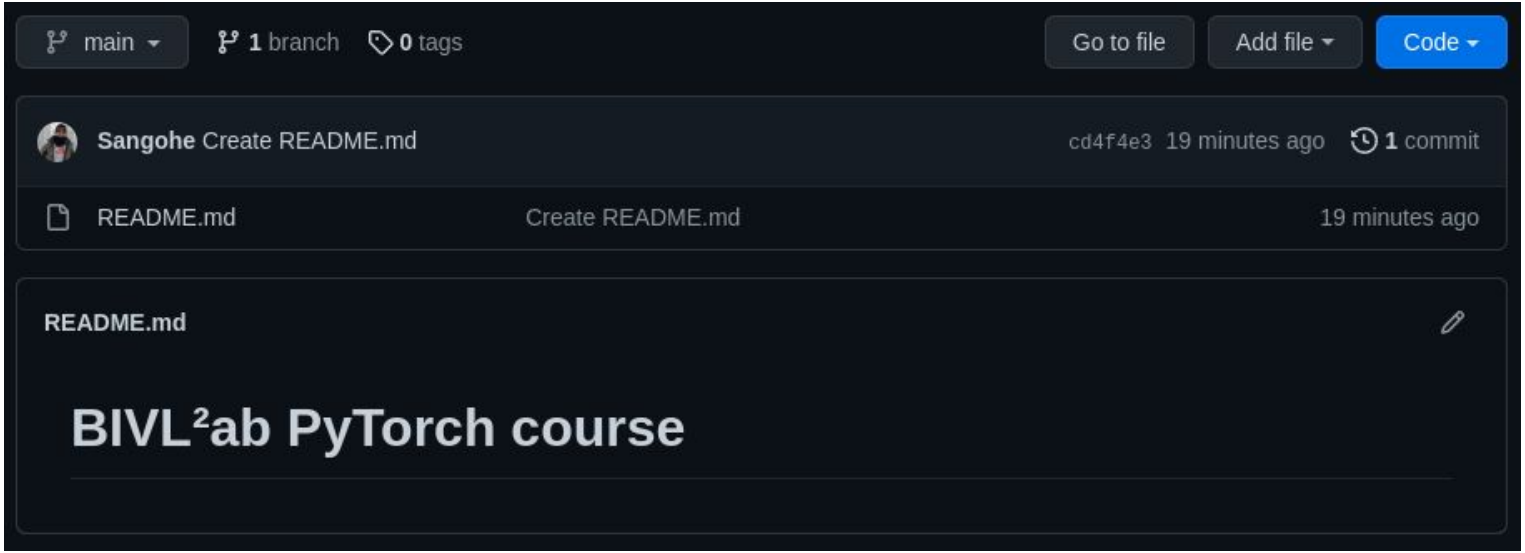
```
sangohe@bivl2ab-cluster:/home/Data/Courses/bivl2ab-pytorch$ docker exec -it pt_santiago bash
root@79e260fb9b7f:/home/sangohe# ln -s /data/Courses/ /home/sangohe/Courses
root@79e260fb9b7f:/home/sangohe# ls
Courses  Data  bivlab-pytorch  projects  projects.zip  tutorials
```

Voila! The Courses folder is now accessible from Jupyter.

Course resources

Github repository

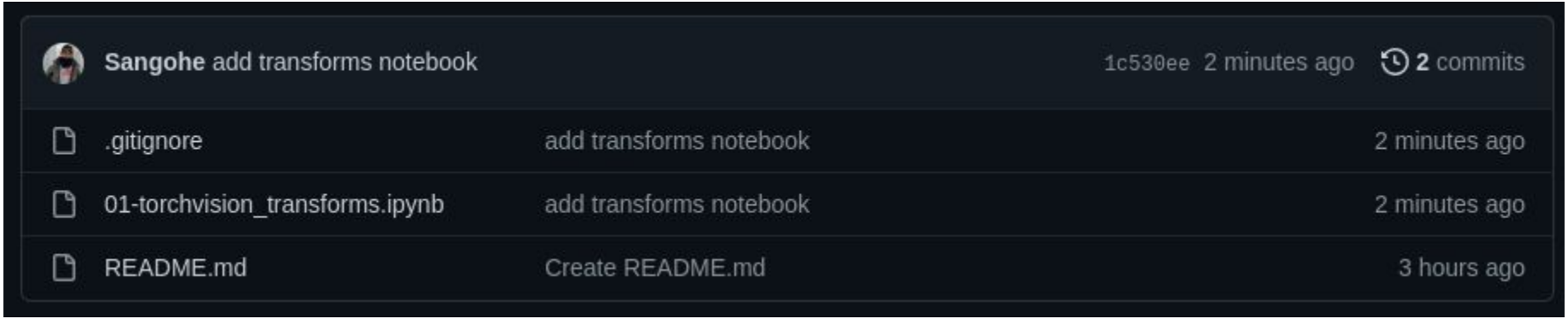
You can find the repository for the course here:
https://github.com/Sangohe/bivl2ab-pytorch/tree/main



`torchvision.transforms` module

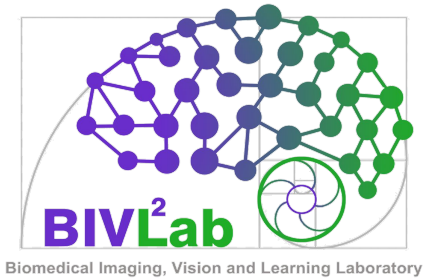
Transforms module

You can find the notebook for this class in the Github repository.



Thank You!


Universidad
Industrial de
Santander


BIVL²ab
Biomedical Imaging, Vision and Learning Laboratory