

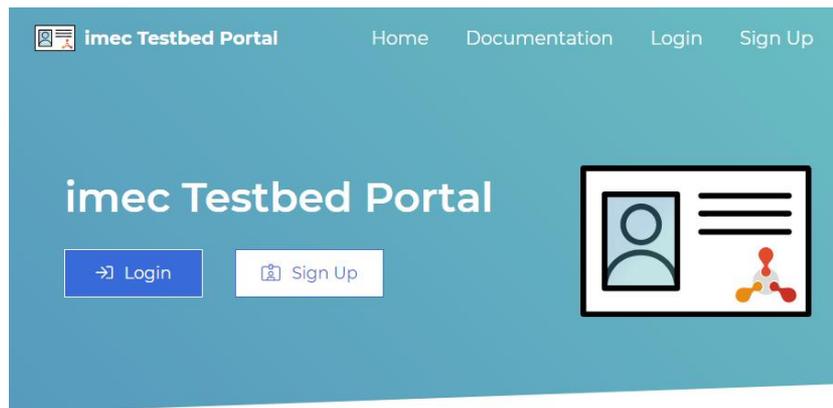


## GPULAB AND JUPYTERHUB INTRODUCTION

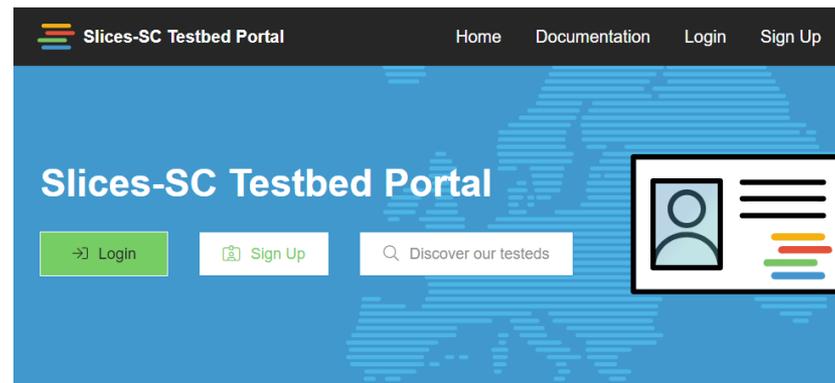
# PREREQUISITES

## REQUEST OR JOIN A PROJECT

**For IDLab members/students:**  
<https://account.ilabt.imec.be>



**For External Users:**  
<https://portal.slices-sc.eu>



# TABLE OF CONTENTS

- Features of GPULab
- Architecture of GPULab
- JupyterHub on GPULab
- Using GPULab directly
  - jobDefinition
  - Storage
  - Opening network ports
  - Accessing logs
  - SSH access

# FEATURES OF GPULAB



Access to a lot of GPUs:

12x A40 – 48 GB,  
40x Tesla V100 – 32 GB,  
11x RTX 4090 – 24GB,  
33x GTX 1080 Ti – 12 GB, ...



Your pip/conda packages are installed and ready to use!

Choose any Docker image with your packages pre-installed



Isolated Storage

Separate storage per project



Automatic Job Scheduling

Jobs are started in FIFO order

# ARCHITECTURE OF GPULAB

## Thin wrapper around GPU-enabled Docker containers:

- Hides complexities of mounting storage, CPU/GPU isolation, etc.
- No need to install CUDA, Tensorflow, PyTorch, etc. on the machine yourself



## Job Scheduler:

- Over multiple machines
- With 1 or more GPUs
- Reservations possible

## Authentication:

- Via imec iLab.t or Slices portal
- Concept of 'projects' for sharing of resources

# TOPOLOGY OF GPULAB

The GPU Lab slaves (=servers) are:

- Spread over 2 locations: iGent DC and UAntwerp DC:
  - Different storage options
  - Different network
- Divided into clusters: servers with the similar properties:
  - Same location,
  - Same GPU type

# JUPYTERHUB ON GPULAB

Available on <https://jupyterhub.ilabt.imec.be>



- Hosted Jupyter Notebook service which provides easy access to GPULab resources
  - Similar to Google CoLab
  - No setup required!
- Ideal for: interactive development of your code and short training runs
- Use GPULab directly for long running jobs

# JUPYTERHUB ON GPULAB

Available on <https://jupyterhub.ilabt.imec.be>

## General settings

Load Configuration Save Configuration

Select a project:

twlocalproj

## Docker settings

Choose a default Docker Image:

Minimal Jupyter Notebook Stack  
Basic Python and Tex packages.



More information

Jupyter Notebook R Stack

R interpreter with popular packages from the R ecosystem. Based on Minimal Stack.



More information

Jupyter Notebook Scientific Python Stack

Popular packages from the scientific Python ecosystem. Based on Minimal Stack.



More information

Jupyter Notebook Deep Learning Stack

TensorFlow and Keras ML libraries. Based on Scientific Python Stack.



More information

Jupyter Notebook Deep Learning Stack

Latest TensorFlow and Keras ML libraries. Based on Scientific Python Stack.



GPU-enabled

More information

VSCode Deep Learning Stack

VSCode with everything from the Deep Learning Stack.



GPU-enabled

More information

Or specify your custom Docker Image:

jupyter/minimal-notebook:latest



## Requested resources

### Storage

- Mount Antwerp Project storage to /project\_antwerp
- Mount Ghent Project storage to /project\_ghent
- Mount Scratch storage to /project\_scratch
- Mount Antwerp Dataset storage to /datasets\_antwerp
- Mount Ghent Dataset storage to /datasets\_ghent

### Number of resources

#CPUs:

1



#CPU GB Memory:

2

#GPUs:

0

The image that you selected has no support for GPU usage.

Cluster ID:

4

Currently available: Overview

12 GPU's, 40 CPU's, 266 GB of CPU memory.

Show Advanced Options

Start

# JUPYTERHUB ON GPULAB

## HOW DOES IT WORK?



- Generates and starts a GPULab job for you
- Redirects you to your Jupyter notebook server once started
  
- **Gotchas:**
  - Server start will timeout after 5 minutes (ex. no GPUs available in chosen cluster)
  - Job will be cancelled after 1 hour of inactivity in the browser, even if a computation is running!
  - Job will always stop after 24 hours
  - Custom docker images must descend from `jupyter/base-notebook`

# JUPYTERHUB ON GPULAB

## FAIR USAGE POLICY

- Use JupyterHub only for interactive development and short training runs.
- Request 1 GPU, maximum 2 if you need to validate concurrency
- Request a reasonable amount of CPU's and CPU memory
  
- Does your code need multiple hours to run? Do you need to scale up the number of GPU's? → Use GPULab directly instead.

# JUPYTERHUB ON GPULAB

## HANDS ON

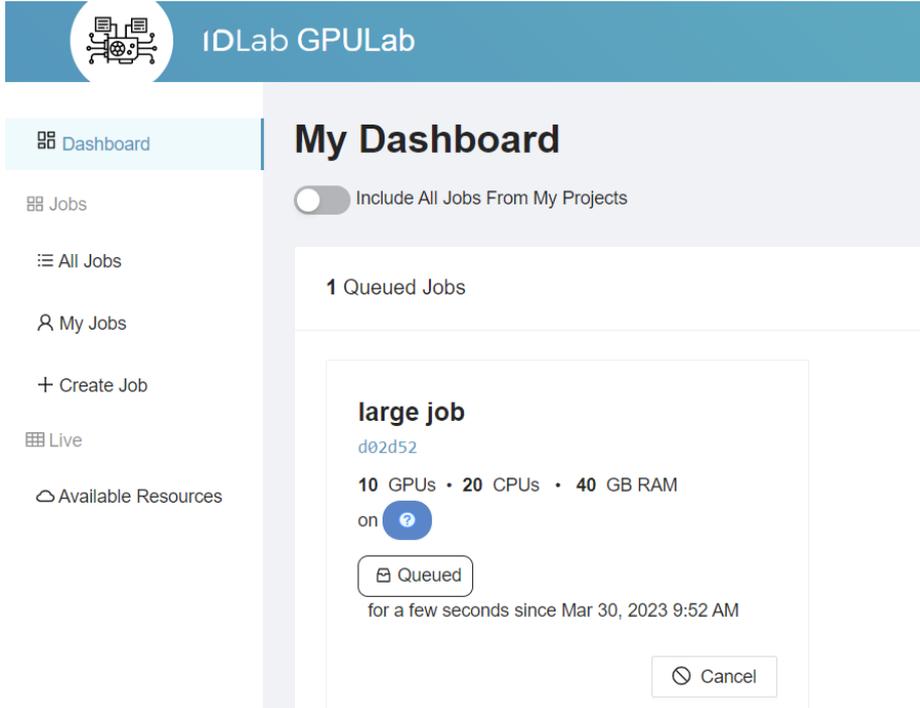


- Go to <https://jupyterhub.ilabt.imec.be>
- Choose a default Docker image of your liking
- Select the storage you want to use
  - Note: illegal combinations are automatically prevented
- Define number of CPU's/GPU's memory
- Optional: define a cluster ID
- Click 'Start'
  
- Create or Upload a Python script / Jupyter notebook that you want to execute

# USING GPULAB DIRECTLY

- Website <https://gpulab.ilabt.imec.be> for monitoring/submitting simple jobs

- `gpulab-cli` for submitting jobs from the command line



IDLab GPU Lab

Dashboard

Jobs

All Jobs

My Jobs

Create Job

Live

Available Resources

## My Dashboard

Include All Jobs From My Projects

1 Queued Jobs

**large job**  
d02d52

10 GPUs • 20 CPUs • 40 GB RAM  
on 

Queued

for a few seconds since Mar 30, 2023 9:52 AM

Cancel

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli --help
Usage: gpulab-cli [OPTIONS] COMMAND [ARGS]...
```

### Options:

<code>--cert</code> PATH	Login certificate [required]
<code>-p, --password</code> TEXT	Password associated with the login certificate
<code>--dev</code>	Use the GPU Lab staging environment (this option is only kept for backward compatibility. It was renamed to <code>--staging</code> )
<code>--staging</code>	Use the GPU Lab staging environment
<code>--stable</code>	Use the GPU Lab production environment (this option is only kept for backward compatibility. It was renamed to <code>--production</code> )
<code>--production</code>	Use the GPU Lab production environment (default)
<code>--custom-master-url</code> TEXT	Use a custom URL as GPU Lab master
<code>--debug</code>	Some extra debugging output
<code>--servercert</code> PATH	The file containing the servers (self-signed) certificate. Only required when the server uses a self signed certificate.
<code>--version</code>	Show the version and exit.
<code>-h, --help</code>	Show this message and exit.

### Commands:

<code>bugreport</code>	Get context info for including in a bug report
<code>cancel</code>	Cancel running job
<code>clusters</code>	Retrieve info about the available clusters. If a <code>cluster_id</code> is specified, detailed info about the slaves of that cluster is shown.
<code>convert</code>	Convert a Job to Job2 format

# DEFINING A GPULAB JOB

GPULab-specific  
bookkeeping

```
{
  "jobDefinition": {
    "name": "helloworld",
    "description": "Hello world!",
    "request": {
      "resources": { "gpuModel": "1080",
                    "gpus": 1, "cpus": 2, "cpuMemoryGb": 2000, },
    },
    "docker": { "image": "nvidia/cuda:12.1.0-runtime-ubuntu22.04",
                "command": "echo 'Hello World'",
                "environment": { },
                "storage": [ { "containerPath": "/project_ghent" } ],
                "portMappings": [ { "containerPort": 80 } ],
              },
    }
}
```

Passed to Docker for  
starting the container

# USING GPULAB

## SUBMITTING A JOB VIA THE CLI

```
thijs@ibcn055:~$ gpulab-cli submit --project twalcari-test < jupyter-scipy.json
87914bc6-10ec-11ea-93a1-d7177117bc9b
thijs@ibcn055:~$ gpulab-cli jobs 87914bc6-10ec-11ea-93a1-d7177117bc9b
  Job ID: 87914bc6-10ec-11ea-93a1-d7177117bc9b
   Name: Jupyter SciPy
  Description: Scipy jupyter notebook server
   Project: twalcari-test
  Username: twalcari
 Docker image: jupyter/scipy-notebook:latest
   Command:
   Status: RUNNING
   Created: 2019-11-27T09:04:26+01:00
  State Updated: 2019-11-27T09:04:35+01:00
   Queued: 2019-11-27T09:04:26+01:00
  Cluster ID: 4
   Worker ID: 3
   Worker Name: n051-02
  Port Mappings: 8888/tcp -> 32814
   Worker Host: n051-02.wall2.ilabt.iminds.be
   SSH login: ssh FCXDN232@n051-02.wall2.ilabt.iminds.be
   Started: 2019-11-27T09:04:34+01:00
   Duration: 11 seconds
   Finished: -
   Deadline: 2019-11-27T19:04:34+01:00
```

# DEFINING A GPULAB JOB

- Modify an example
  - Consult documentation on <https://doc.ilabt.imec.be/ilabt/gpulab/jobdefinition.html>
- Use the 'Create job' function on the website

## Create a job

↻ Load jobDefinition File...

📄 Load saved job ▾

📄 Load template ▾

## Resulting JSON:

📄 Download jobDefinition File

📄 Save in Browser

## Job Definition

### Core info

Show advanced fields

Project: twlocalproj ▾

Name: helloworld

Description: Writes 'Hello World'

```
{
  "owner" : {
    "projectUrn" :
      "urn:publicid:IDN+ilabt.imec.be+project+twlocalproj"
  }
  "name" : "helloworld"
  "request" : {
    "docker" : {
      "image" : "nvidia/cuda:12.1.0-runtime-ubuntu22.04"
      "command" : "echo 'Hello World'"
      "equipment" : {
```

# DEFINING A GPULAB JOB

## DETERMINING THE CLUSTER AND RESOURCES TO REQUEST: VIA THE WEBSITE

Live

Available Resources

Cluster  8

 GPU Model

NVIDIA A40  
with 44 GB RAM

 Storage

/datasets\_ghent/  
/project\_ghent/

Nodes

gpulab8A

● ONLINE

 0 / 4 GPUs available  
 0 / 48 CPUs available  
 213 / 527 GB CPU RAM available

Active Jobs

6

gpulab8B

● ONLINE

 0 / 4 GPUs available  
 0 / 48 CPUs available  
 0 / 527 GB CPU RAM available

Active Jobs

4

gpulab8C

 0 / 4 GPUs available  
 1 / 48 CPUs available  
 1 / 527 GB CPU RAM available

Active Jobs

7

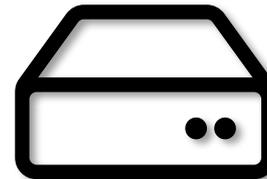
# DEFINING A GPULAB JOB

## DETERMINING THE CLUSTER AND RESOURCES TO REQUEST: VIA THE CLI

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli clusters
```

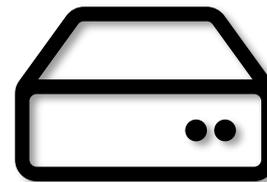
ID	GPU Model	Comment	Slaves	GPUs	CPUs
1 stable		1x 2x GF GTX 1080 Ti	0	0/0	0/0
2 stable		temporary cluster	0	0/0	0/0
3 stable	NVIDIA GeForce RTX 2080 Ti	1x RTX2080	1	1/1	10/12
4 stable	NVIDIA GeForce GTX 1080 Ti	4x 11x GF GTX 1080 Ti	4	19/44	55/128
5 stable	Quadro RTX 4000	UAntwerp: 2x 4x RTX4000	3	0/8	21/80
6 stable	Tesla V100-SXM3-32GB	1x HGX-2 - 16xTesla V100	1	1/16	46/96
7 stable	Tesla V100-SXM2-32GB, Tesla V100-SXM3-32GB	UAntwerp: DGX-1 + DGX-2 - (8+16)x Tesla V100	4	0/24	17/176
8 stable	NVIDIA A40	12x A40 48GB	3	0/12	12/144

# STORAGE ON GPULAB



- File system within your GPULab job is ephemeral:
  - Jobs start with file system defined within the Docker image that you start
  - when the job ends, all changes to the file system are lost
- Exception: any permanent storage that you attach to the GPULab job: use this for storing all your datasets, code, logs, results

# STORAGE ON GPULAB

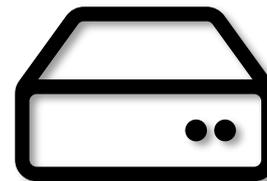


- GPU Lab slaves are spread over 2 datacenters, each with their own storage:
  - /project\_antwerp : cluster 5 & 7
  - /project\_ghent : all other clusters
- Some slaves have very fast local scratch storage: /project\_scratch
- Use CPU memory as storage by mounting it as tmpfs

- Use request → docker → storage to set storage mount points

```
“storage”: [  
  {  
    "hostPath": "/project_ghent"  
  },  
  {  
    "hostPath": "/project_ghent/work",  
    "containerPath": "/custom/dir"  
  }  
]
```

# STORAGE ON GPULAB



`/project_ghent`

115 TB backed by NVMe disks

Connected by Infiniband to Cluster 6

“Fair use policy”, but no hard limits

`/project_antwerp`

105TB backed by DDN A<sup>3</sup>I storage

Connected by Infiniband to Cluster 5 & 7

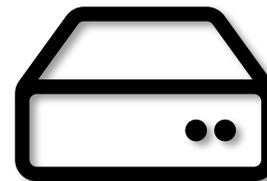
“Fair use policy”, but no hard file size limits

Quota of 500.000 inodes

**Documentation:**

<https://doc.ilabt.imec.be/ilabt/gpulab/storage.html>

# STORAGE ON GPULAB



## /project\_scratch

- Slave-specific:
  - **6A**: 94TB
  - **7A**: 28TB
  - **7B**: 7TB
- Use request → resources → slaveName to request specific slave

```
"resources": {  
  ...  
  "slaveName": "slave6a"  
},
```

- RAID0 storage over multiple local NVMe disks

Example: 6A uses 16  
enterprise-grade NVMe disks  
with each a MTBF  
of 2.000.000 hours

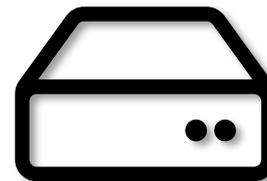


### **WARNING**

Do NOT store anything here that  
you cannot afford to lose!

# STORAGE ON GPULAB

## TMPFS



Due to limited local storage on each slave, you can use a **maximum of 10GB** of disk space outside of the mounted storage paths.

Using more will kill your job.

**Solution 1:** make sure that all necessary dependencies are already installed in the Docker image that you are using

**Solution 2:** mount a part of the available CPU memory as an ephemeral tmpfs

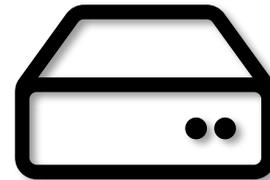
```
“storage”: [  
  {  
    "hostPath": "tmpfs",  
    "containerPath": "/my_tmp_dir",  
    "sizeGb": 4  
  }  
]
```

**Documentation:**

<https://doc.ilabt.imec.be/ilabt/gpulab/storage.html#tmpfs>

# STORAGE ON GPULAB

## IMPORTING YOUR DATA INTO GPULAB



### Small uploads?

Start a JupyterHub session and upload via the browser

A screenshot of a JupyterHub web interface. The browser address bar shows 'jupyterhub.ilabt.imec.be/user/twlocal@ilabt.imec.be/lab/tree/project\_ghent'. The interface includes a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help) and a file browser on the left. The file browser shows a search bar and a list of files: 'Untitled.ipynb' (9 months ago), 'untitled.txt' (9 months ago), and 'Untitled1.ipynb' (8 months ago). On the right, there is a 'Launcher' section with a 'Notebook' button and a 'Console' button, both with Python 3 (ipykernel) icons.

### Large uploads?

Connect via SFTP to your job

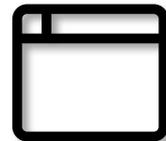
- Use the SSH credentials supplied by your job to connect
- Use login certificate as credentials

A screenshot of a 'Job State' summary box. It contains the following information: 'Host Summary: 4C', 'Cluster: 4', 'Slave: gpu1ab4C', and 'SSH: 046FACLU @ 4c.gpulab.ilabt.imec.be'. The SSH key is highlighted in a light blue box.

**Documentation:**

<https://doc.ilabt.imec.be/ilabt/gpulab/storage.html>

# EXPOSING PORTS OF YOUR CONTAINER



- You can define ports to be exposed in request → docker → portMappings
- Use containerPort to specify which port of your container you want to access
- Host address/port is determined during job scheduling
- Ghent-based nodes:
  - only have public IPv6 address
  - use iGent VPN for IPv4 access
- Antwerp-based nodes:
  - use IDLab Antwerp VPN for access

```
"portMappings": [  
  {  
    "containerPort": 5000  
  },  
  {  
    "containerPort": 5001,  
    "hostPort": 5001  
  }  
]
```

**WARNING:** Job will fail if another container is already mapped to that port!

# EXPOSING PORTS OF YOUR CONTAINER

## FINDING THE HOST/PORT VIA THE WEBSITE OR CLI



```
$ gpulab-cli jobs <job_id>
```

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli jobs ae9e8923
Job ID: ae9e8923-f2a9-4f1a-ad76-ae1680a9efd9
Name: JupyterHub-singleuser
Description: Single Jupyter notebook instance for JupyterHub
Project: twlocalproj
User URN: urn:publicid:IDN+ilabt.imec.be+user+twlocal
User ID: twlocal@ilabt
Docker image: jupyter/minimal-notebook:latest
Command: start-notebook.sh --notebook-dir=/ --SingleUserNotebookApp.default_url='/lab/tr
Status: RUNNING
Cluster ID: 4
Worker ID: 8
Worker Name: gpulab4C
Port Mappings: 8888 -> 33400, 8888 -> 33400
Worker Host: 4c.gpulab.ilabt.imec.be
SSH login:: ssh -i '/home/thijs/.ssh/twlocal_unlocked.pem' ELWHKEIY@4c.gpulab.ilabt.imec.be
em' fffftwlocal@bastion.ilabt.imec.be -W %h:%p"
Timing:
  Created: 2023-03-30T11:25:35+02:00 (5 minutes, 48 seconds ago)
  Queued: 2023-03-30T11:25:35+02:00 (less than 1 second after job creation)
  Assigned: 2023-03-30T11:25:38+02:00 (3 seconds after QUEUED)
  Starting: 2023-03-30T11:25:41+02:00 (3 seconds after ASSIGNED)
  Running: 2023-03-30T11:25:43+02:00 (2 seconds after STARTING)
  Ended: -
Duration: 5 minutes, 40 seconds
State Updated: 2023-03-30T11:25:43+02:00 (5 minutes, 40 seconds ago)
```

Job State

Host Summary: 4C

Cluster: 4

Slave: gpulab4C

SSH: ELWHKEIY@4c.gpulab.ilabt.imec.be

Port mappings

Port 8888 → 4c.gpulab.ilabt.imec.be:33400

Port 8888 → 4c.gpulab.ilabt.imec.be:33400

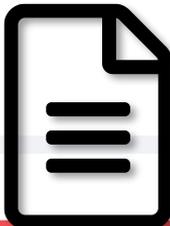
# CHECKING THE LOGS OF YOUR CONTAINER ON THE CLI



```
$ gpulab-cli log <job_id>
```

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli log ae9e8923
2023-03-30T11:25:42.854890278+02:00 WARNING: using start-singleuser.sh instead of start-notebook.sh to start a server a
2023-03-30T11:25:42.859173254+02:00 Entered start.sh with args: jupyterhub-singleuser --ip=0.0.0.0 --notebook-dir=/ --S
b/tree/project_ghent'
2023-03-30T11:25:42.923211639+02:00 Updated the jovyan user:
2023-03-30T11:25:42.923275604+02:00 - username: jovyan      -> twlocal
2023-03-30T11:25:42.923370399+02:00 - home dir: /home/jovyan -> /home/twlocal
2023-03-30T11:25:42.930787366+02:00 Update twlocal's UID:GID to 1000:7123
2023-03-30T11:25:42.971664391+02:00 userdel: group twlocal not removed because it is not the primary group of user twlo
2023-03-30T11:25:43.051739125+02:00 Attempting to copy /home/jovyan to /home/twlocal...
2023-03-30T11:25:43.059861132+02:00 Success!
2023-03-30T11:25:43.059889846+02:00 Changing working directory to /home/twlocal/
2023-03-30T11:25:43.067453762+02:00 Granting twlocal passwordless sudo rights!
2023-03-30T11:25:43.067769575+02:00 Running as twlocal: jupyterhub-singleuser --ip=0.0.0.0 --notebook-dir=/ --SingleUser
roject_ghent'
2023-03-30T11:25:46.163553566+02:00 [I 2023-03-30 09:25:46.163 SingleUserLabApp mixins:609] Starting jupyterhub single-
2023-03-30T11:25:46.163744884+02:00 [I 2023-03-30 09:25:46.163 SingleUserLabApp mixins:623] Extending jupyterlab.labhub
3.6.2
```

# CHECKING THE LOGS OF YOUR CONTAINER ON THE WEBSITE



← Job **ae9e8923-f2a9-4f1a-ad76-ae1680a9efd9** ( **RUNNING** )

Refresh

Cancel

Halt

Creator: twlocal

Project: twlocalproj

Name: JupyterHub-singleuser

Description: Single Jupyter notebook instance for JupyterHub

General Info

**Logs**

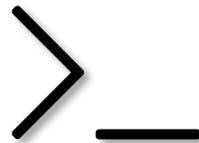
Debugging Logs

Usage Graphs

Raw Job JSON

```
2023-03-30T11:25:42.854890278+02:00 WARNING: using start-singleuser.sh instead of start-notebook.sh to start a server associated with JupyterHub.
2023-03-30T11:25:42.859173254+02:00 Entered start.sh with args: jupyterhub-singleuser --ip=0.0.0.0 --notebook-dir=/ --SingleUserNotebookApp.default_url='
2023-03-30T11:25:42.923211639+02:00 Updated the jovyan user:
2023-03-30T11:25:42.923275604+02:00 - username: jovyan      -> twlocal
2023-03-30T11:25:42.923370399+02:00 - home dir: /home/jovyan -> /home/twlocal
2023-03-30T11:25:42.930787366+02:00 Update twlocal's UID:GID to 1000:7123
2023-03-30T11:25:42.971664391+02:00 userdel: group twlocal not removed because it is not the primary group of user twlocal.
2023-03-30T11:25:43.051739125+02:00 Attempting to copy /home/jovyan to /home/twlocal...
2023-03-30T11:25:43.059861132+02:00 Success!
2023-03-30T11:25:43.059889846+02:00 Changing working directory to /home/twlocal/
2023-03-30T11:25:43.067453762+02:00 Granting twlocal passwordless sudo rights!
2023-03-30T11:25:43.067769575+02:00 Running as twlocal: jupyterhub-singleuser --ip=0.0.0.0 --notebook-dir=/ --SingleUserNotebookApp.default_url='/lab/tree
2023-03-30T11:25:46.163553566+02:00 [I 2023-03-30 09:25:46.163 SingleUserLabApp mixins:609] Starting jupyterhub single-user server version 3.1.1
2023-03-30T11:25:46.163744884+02:00 [I 2023-03-30 09:25:46.163 SingleUserLabApp mixins:623] Extending jupyterlab.labhubapp.SingleUserLabApp from jupyterl
2023-03-30T11:25:46.163979141+02:00 [I 2023-03-30 09:25:46.163 SingleUserLabApp mixins:623] Extending jupyter_server.serverapp.ServerApp from jupyter_ser
2023-03-30T11:25:46.188658587+02:00 /opt/conda/lib/python3.10/site-packages/traitlets/traitlets.py:2548: FutureWarning: Supporting extra quotes around st
2023-03-30T11:25:46.188700697+02:00 warn(
2023-03-30T11:25:46.199510390+02:00 [W 2023-03-30 09:25:46.199 SingleUserLabApp configurable:200] Config option `open_browser` not recognized by `Single
```

# GETTING SSH-ACCESS TO YOUR CONTAINER



```
$ gpulab-cli ssh <job-id>
```

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli ssh ae9e8923
Warning: Permanently added the ECDSA host key for IP address '10.2.47.43' to the list of known hosts
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-58-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
```

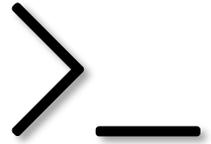
```
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
```

```
To restore this content, you can run the 'unminimize' command.
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

# GETTING SSH-ACCESS TO YOUR CONTAINER



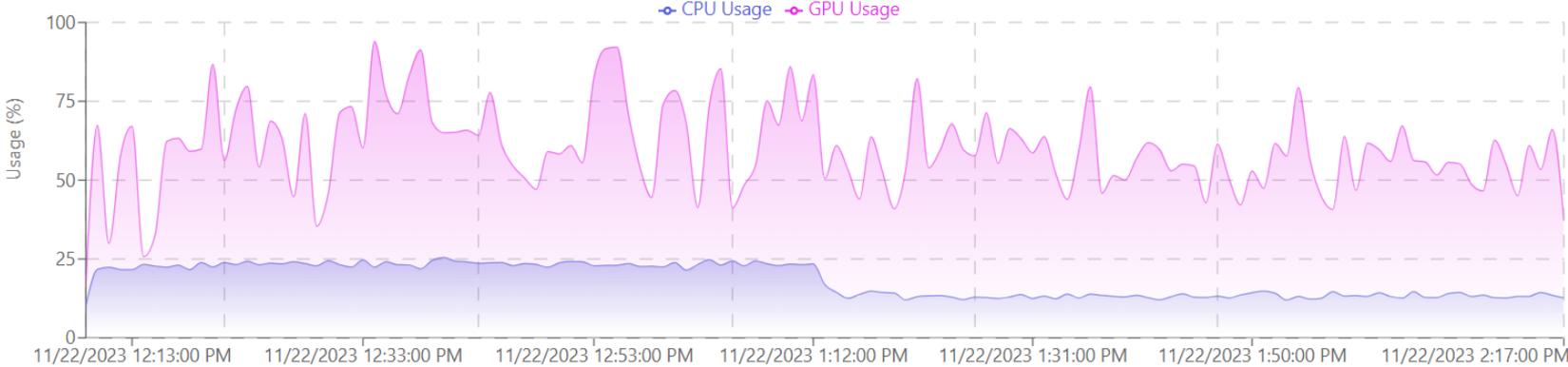
- SSH access to GPULab jobs is a hack on top of ``docker exec -ti /bin/bash`` into the container.
- Can only be used for shell access and SFTP access
- No support for additional SSH channels:
  - No port forwarding
  - No SCP
  - No support for VSCode Remote Development

# CHECKING CPU/GPU UTILIZATION

- General Info
- Logs
- Debugging Logs
- Usage Graphs
- Raw Job JSON

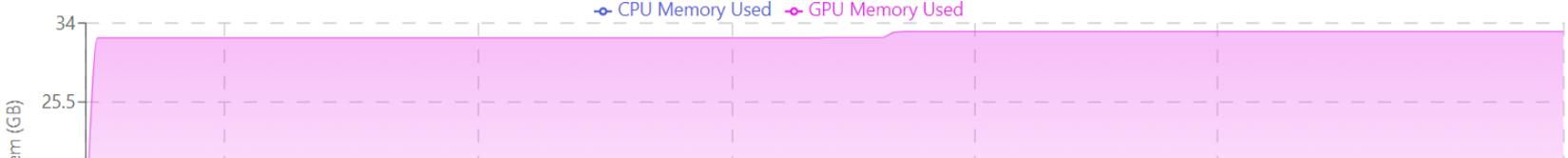
## CPU/GPU Utilization

- CPU
- GPU
- Network
- Other



## Memory Utilization

- CPU
- GPU
- Network
- Other



# FAIR USAGE POLICY

- Non-interactive GPU Lab jobs should:
  - Not require manual intervention to start their computations
  - Stop automatically when computations have ended
  - Be efficient:
    - Use all the GPU's they requested
    - Request enough CPU/memory to support your GPU ...
    - ... but not more than necessary

# SUGGESTED WORKFLOW

1. Develop and test your code locally or in a Jupyter notebook
  - Create your own Docker image with custom software if necessary
2. Scale up to the full dataset once your code is ready:
  - Split your work into multiple jobs if possible (ex. Parameter sweep)
3. Add checkpointing for your intermediate results (e.g. if your job or server would crash)

# GET STARTED!

- **Documentation:** <http://doc.ilabt.imec.be>
- **GPULab:** <https://gpulab.ilabt.imec.be>
- **JupyterHub:** <https://jupyterhub.ilabt.imec.be>

## SUPPORT



helpdesk@ilabt.imec.be

imec iLab.t documentation

Docs / Welcome to the imec iLab.t testbeds' documentation!

### Welcome to the imec iLab.t testbeds' documentation!

Get familiar with the imec iLab.t offering and explore their features:

- Virtual Wall**  
Perform large networking and cloud experiments.
- w-iLab.t testbeds**  
Wireless Testbed and OfficeLab
- GPULab**  
Batch system for GPU-enabled docker containers
- jupyterhub**

# HANDS ON: YOUR FIRST GPULAB JOB

- Go to <https://gpulab.ilabt.imec.be>
- Click on  + New Job
- Complete the form to execute the Python script / Jupyter notebook:
  - Image: you can reuse the Jupyter 'Docker stacks' images
  - Command:
    - `python /project_ghent/path/to/my/script.py`
    - `bash -c '/project_ghent/my/script.py > /project_ghent/log.txt'`
    - `jupyter nbconvert --to notebook --execute /project_ghent/example.ipynb`
  - Resources / input and output: complete as necessary
  - Scheduling: leave 'Interactive' unchecked.
- Click 'Start'

Interactive jobs will never queue, but fail instead. Useful for when you need to interact with the job (via SSH/SFTP/Jupyter server/ ...)

# YOUR FIRST GPULAB JOB VIA THE CLI

## SETTING UP THE CLI

- Only supported on Linux/Mac (On Windows: use WSL)
- Install the CLI: `sudo pip install imecilabt-gpulab-cli`
- Get your Login Certificate (PEM) from: <https://account.ilabt.imec.be/profile> or <https://portal.slices-sc.eu/>

### Login Certificate

Your login certificate is valid until 2023-07-13 09:34 CEST



Download Login Certificate (PEM)



Download PKCS12 Certificate

# YOUR FIRST GPULAB JOB VIA THE CLI

## USING THE CLI

```
$ gpulab-cli --cert /path/to/cert.pem clusters
```

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli --cert /home/thijs/.ssh/login_ilabt_imec_be_twlocal.pem clusters  
Enter PEM pass phrase:
```

ID	GPU Model	Comment	Slaves	GPUs	CPUs
1 stable		1x 2x GF GTX 1080 Ti	0	0/0	0/0
2 stable		temporary cluster	0	0/0	0/0
3 stable	NVIDIA GeForce RTX 2080 Ti	1x RTX2080	1	0/1	8/12
4 stable	NVIDIA GeForce GTX 1080 Ti	4x 11x GF GTX 1080 Ti	4	9/44	35/128
5 stable	Quadro RTX 4000	UAntwerp: 2x 4x RTX4000	3	2/8	8/80
6 stable	Tesla V100-SXM3-32GB	1x HGX-2 - 16xTesla V100	1	1/16	45/96
7 stable	Tesla V100-SXM2-32GB, Tesla V100-SXM3-32GB	UAntwerp: DGX-1 + DGX-2 - (8+16)x Tesla V100	4	0/24	2/176
8 stable	NVIDIA A40	12x A40 48GB	3	0/12	17/144

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ |
```

# YOUR FIRST GPULAB JOB VIA THE CLI

## USING THE CLI

Tired of passing your certificate as a parameter?  
Add the following line to your `~/ .bashrc` :

```
export GPULAB_CERT=/path/to/cert.pem
```

Tired of entering your password?  
Decrypt your private key with OpenSSL  
→ see CLI documentation

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli clusters
Enter PEM pass phrase:
```

ID	GPU Model	Comment
1 stable		1x 2x GF GTX 1080 Ti
2 stable		temporary cluster
3 stable	NVIDIA GeForce RTX 2080 Ti	1x RTX2080
4 stable	NVIDIA GeForce GTX 1080 Ti	4x 11x GF GTX 1080 Ti
5 stable	Quadro RTX 4000	UAntwerp: 2x 4x RTX4000
6 stable	Tesla V100-SXM3-32GB	1x HGX-2 - 16xTesla V100
7 stable	Tesla V100-SXM2-32GB Tesla V100-SXM3-32GB	UAntwerp: DGX-1 + DGX-2 - (8+16)

# YOUR FIRST GPULAB JOB VIA THE CLI

## STARTING AN INTERACTIVE JOB

```
(gpulab-venv) thijs@thijs-xps:~/gpulab$ gpulab-cli interactive --help  
Usage: gpulab-cli interactive [OPTIONS]
```

### Options:

<code>--project TEXT</code>	[required]
<code>--duration-minutes TEXT</code>	[required]
<code>--docker-image TEXT</code>	[required]
<code>-p, --proxy</code>	Use the jFed proxy to reach the container
<code>-P, --no-proxy</code>	Don't use the jFed proxy to reach the container (requires IPv6 or VPN)
<code>--auto-proxy</code>	Automatically use the jFed proxy to reach the container
<code>--only-show</code>	Do nothing except showing the job request
<code>--cpus INTEGER</code>	Number of CPU cores required (default 1)
<code>--gpus INTEGER</code>	Number of GPU's required (default 1)
<code>--mem INTEGER</code>	Memory required, in GB (default 2)
<code>--cluster-id INTEGER</code>	Wanted cluster ID (default any)
<code>--ssh-pub-key TEXT</code>	Add an SSH public key. This can be used to access the running job's container. Uses the openssh public key format. This option can be specified more than once, to add multiple keys.

# YOUR FIRST GPULAB JOB VIA THE CLI

## STARTING A JOB USING A JOBDEFINITION

- Fill in the 'Create Job' form on the GPULab website, and click 'Download jobDefinition File':
- Manually tweak your jobDefinition file if necessary
- Start the job as follows:

Resulting JSON:

Download jobDefinition File

Save in Browser

```
{
  "owner": {
    "projectUrn": "urn:publicid:IDN+ilabt.imec.be+project+twlocal"
  }
  "name": "exampleJob"
  "request": {
    "docker": {
```

```
$ gpulab-cli submit --project my_project < my_jobdef_file.json
```

# ADVANCED TOPICS

## PARAMETER SWEEP

Want to do a parameter sweep?

Split it in multiple jobs!

1. Generate the correct jobDefinition JSON in the language of your choice
2. Pass it on the STDIN to ``gpuLab-cli submit --project my_project``

```
#!/bin/bash

sweep_parameter=({100..1000..50})

for param in ${sweep_parameter[@]}
do
    gpuLab-cli submit --project my_project <<EOF
{
  "jobDefinition": {
    "name": "My Parameter sweep with param ${param}",
    ...
    "command": "/project_ghent/my/script.sh ${param}'",
    ...
  }
}
EOF
done
```



# umec

embracing a better life