

AI Infrastructure Service

An introduction

Slices Summer School 2026

09/06/2026

Thijs Walcarius (imec)

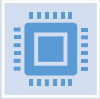


Table of Contents

- Features of the AI Infrastructure Service
- Architecture
- How to use:
 - jobDefinition
 - Storage
 - Opening network ports
 - Accessing logs
 - SSH access
- JupyterHub on the AI Infrastructure Service
- Hands on!



Features of the AI Infrastructure Service



Access to a lot of GPUs:

18x H200 – 141GB, 8x A100 – 80GB, 12x A40 – 48 GB, 40x Tesla V100 – 32 GB, 10x RTX4090 – 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 the AI Infrastructure

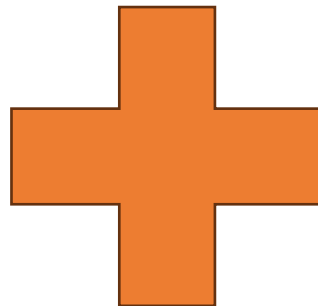
Thin wrapper around GPU-enabled Docker containers:

- 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 Slices-RI portal
- Concept of 'projects' for sharing of resources

Topology of the AI Infrastructure Service

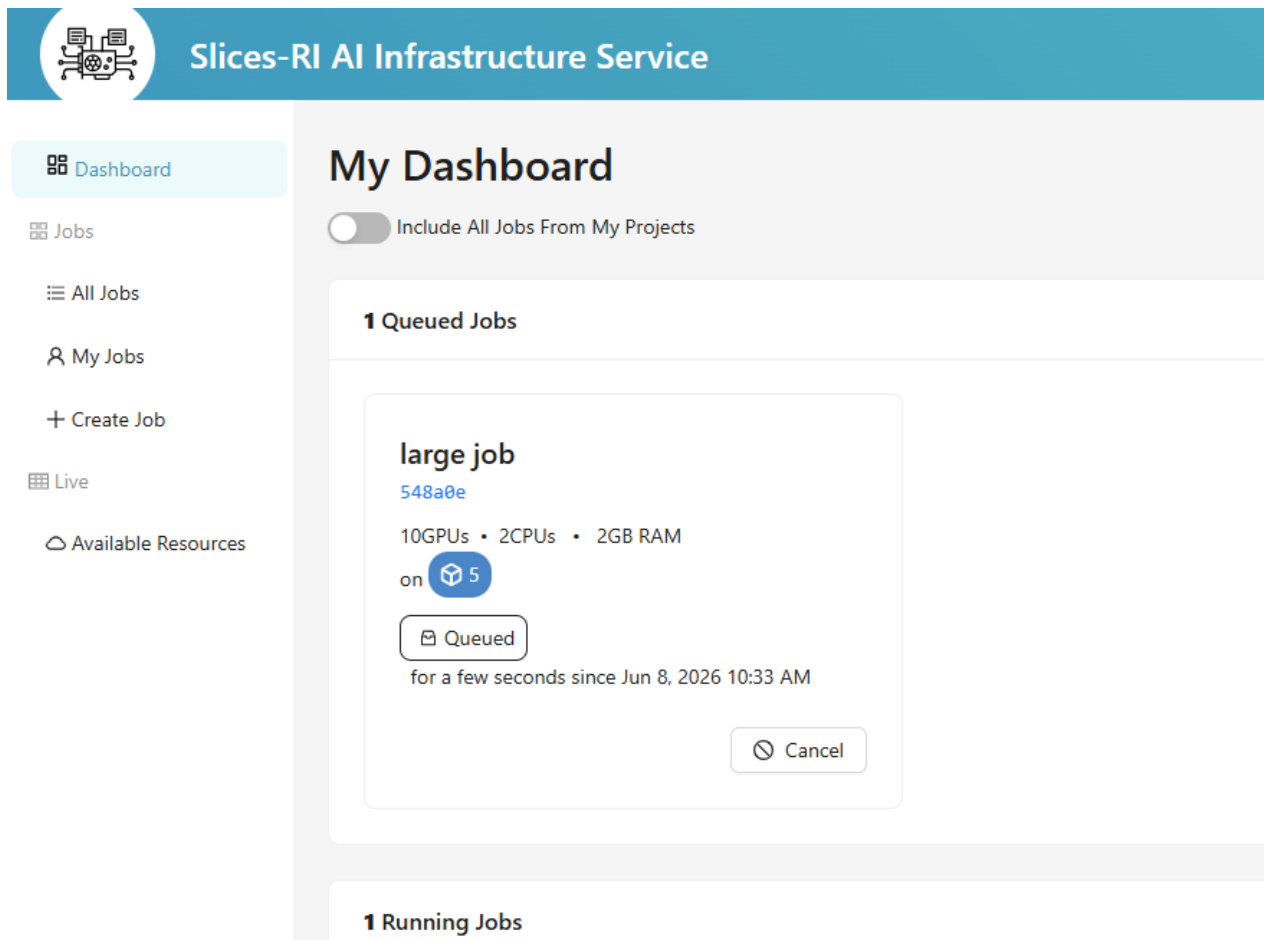
The machines are:

- Spread over **3 locations**: iGent DC, UAntwerp DC, UC3M DC:
 - Different storage options
 - Different network
- Divided into **clusters**: servers with the similar properties:
 - Same location,
 - Same GPU type

Using the AI Infrastructure Service

- Website <https://ai.slices-ri.eu> for monitoring/submitting simple jobs

- `slices ai` for submitting jobs from the command line



The screenshot shows the 'My Dashboard' of the Slices-RI AI Infrastructure Service. On the left, a sidebar contains navigation links: Dashboard, Jobs, All Jobs, My Jobs, Create Job, Live, and Available Resources. The main content area shows a toggle for 'Include All Jobs From My Projects' and a section for '1 Queued Jobs'. A job card for 'large job' (ID 548a0e) is displayed, showing it is using 10GPUs, 2CPUs, and 2GB RAM on a specific node. The job status is 'Queued' and it has been in this state for a few seconds since June 8, 2026, at 10:33 AM. A 'Cancel' button is visible at the bottom right of the job card.

```
> slices ai
```

```
Usage: slices ai [OPTIONS] COMMAND [ARGS]...
```

AI Infrastructure Service Commands.

Options

```
--site, --site-id TEXT AI Infrastructure Service site
--help -h Show this message and exit.
```

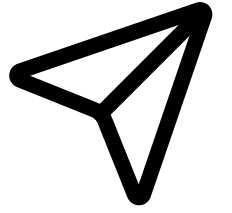
Auxiliary commands

```
wait Wait until the requested status has been reached (o
modify Change maxDuration, minDuration or notAfter of a jo
debug Show internal event log for a job.
```

Lifecycle commands

```
submit Submit a job described in a file.
cancel Cancel a job.
rm Delete a job.
halt Halt a job. Halted jobs may be automatically re-QUE
hold Hold a QUEUED job, preventing it from running.
requeue Requeue/Release a held job, QUEUEing it, so it wait
```

Submitting a job via the CLI

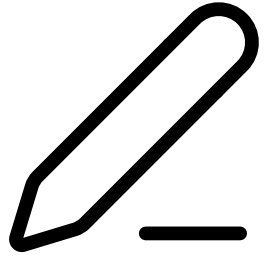


```
> slices ai submit nvidia_smi.json  
✨ Created Job 7bd9d1dc-8398-48c0-8bb8-b3925bcba583
```



Defining a Job

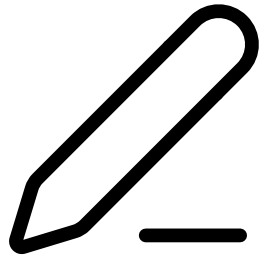
AI Infrastructure-specific
bookkeeping



```
{
  "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 when starting
the container

Defining a job



- Modify an example
 - Consult documentation on <https://doc.slices-ri.eu/BasicServices/AI/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

```
{
  "owner" : {
    "projectUrn" :
      "urn:publicid:IDN+ilabt.imec.be+project+twlocalproj"
  }
  "name" : "helloworld"
  "request" : {
```

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

gpulab

0 / 4 GPUs available
12 / 48 CPUs available
11 / 527 GB CPU RAM available

Active Jobs

7



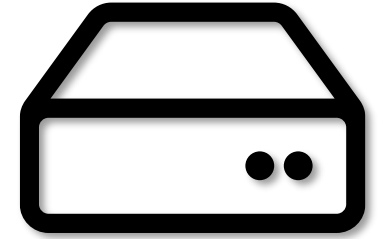
Determining the cluster and resources to request – via the CLI

```
> slices ai clusters
```

Clusters

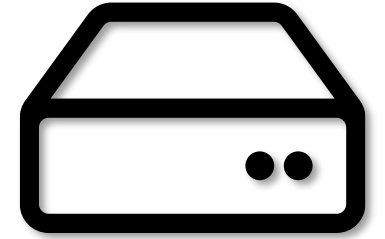
ID	Site	GPU Model	CUDA	# Nodes	GPUs	CPUs	Memory (GB)
3	be-gent1	NVIDIA RTX 2080 Ti (11GB)	13.0	1	0/1	3/12	14/30
4	be-gent1	NVIDIA GTX 1080 Ti (11GB)	13.0	3	28/33	61/96	592/750
5	be-gent1	NVIDIA RTX 4090 (24GB)	13.0	1	1/10	2/32	98/250
6	be-gent1	NVIDIA V100-SXM3 (32GB)	13.0	1	0/16	5/96	727/1509
8	be-gent1	NVIDIA A40 (45GB)	13.0	3	0/12	43/144	900/1504
9	be-gent1	NVIDIA RTX 5090 (32GB)	13.0	1	0/0	0/0	0/0
10	be-gent1	NVIDIA RTX A6000 (48GB)	13.0	1	0/2	26/32	56/124
11	be-gent1	NVIDIA H200 NVL (141GB)	13.0	2	0/10	58/192	2535/3019
12	be-gent1	NVIDIA RTX PRO 4500 Blackwell (32GB)	13.2	2	5/14	28/64	844/940
13	be-gent1	NVIDIA RTX PRO 6000 Blackwell Workstation Edition (96GB)	13.2	1	0/4	4/32	254/502
101	be-antwerpen1	NVIDIA A100 (80GB)	12.2	2	0/8	124/192	368/1004
102	be-antwerpen1	NVIDIA RTX 4000 (8GB)	12.2	2	0/8	13/80	75/500
103	be-antwerpen1	NVIDIA V100-SXM3 (32GB), NVIDIA V100-SXM2 (32GB)	12.2	2	3/24	120/176	1395/2011
201	es-madrid1	NVIDIA H200 (141GB)	12.8	1	0/8	144/224	1561/1985

Storage



- File system within your 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 job: use this for storing all your datasets, code, logs, results
 - The S3 storage service: best suited for your datasets!

Storage

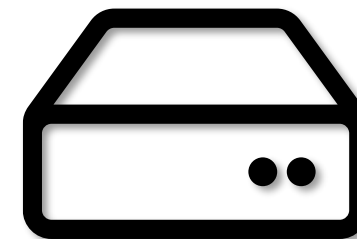


- Machines are spread over 3 datacenters, each with their own storage:
 - /project_madrid: all 2xx clusters
 - /project_antwerp: all 1xx clusters
 - /project_ghent: all other clusters
- Some machines have very fast local scratch storage: /project_scratch
→ clusters 6, 9, 12, 13, 101, 103
- Use CPU memory 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



/project_ghent

115 TB backed by NVMe disks

“Fair use policy”, but no hard limits

/project_antwerp

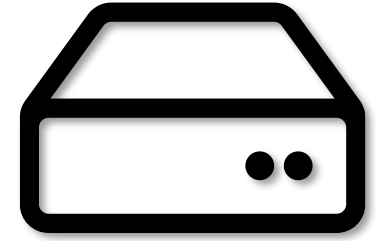
105TB backed by DDN A³I storage

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

Quota of 500.000 inodes



Storage



/project_scratch

- Slave-specific:
 - **6A**: 94TB
 - **9A, 12A, 12B, 13A**: 7TB
 - **101A**: 6TB
 - **101B**: 7TB
 - **103A**: 28TB
 - **103B**: 7TB
- RAID0 storage over multiple local NVMe disks
- Example: 6A uses 16 enterprise-grade NVMe disks with each a MTBF of 2.000.000 hours
- Use request → resources → slaveName to request specific slave



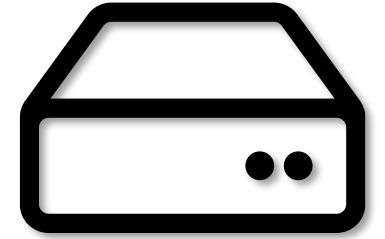
```
"resources": {  
    ...  
    "slaveName": "slave6a"  
},
```



WARNING

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

Storage



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.

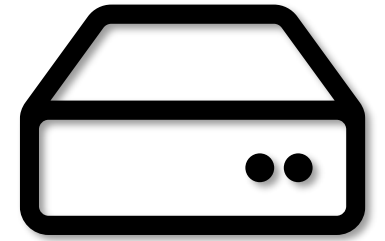
Solution 1: make sure that all necessary dependencies are already installed in the Docker image that you are using

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

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



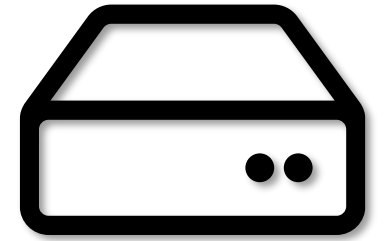
Storage: S3 compatible storage



- Mirrored storage in iGent and Antwerp DC.
- Authentication not handled by AI Infrastructure. No need to mention it in the jobDefinition.
- Example: usage in PyTorch with s3torchconnector
- 1/ Retrieve credentials from MINIO webinterface @ <https://s3.slices-be.eu/>
- 2/ Store them in .env file:
AWS_ACCESS_KEY_ID=my..access..key
AWS_ENDPOINT_URL=https://s3.slices-be.eu
AWS_REGION=us-east-1
AWS_SECRET_ACCESS_KEY=my..secret..key



Storage: S3 compatible storage



Example: usage in PyTorch with s3torchconnector

- 1/ Retrieve credentials from MINIO webinterface @ <https://s3.slices-be.eu/>
- 2/ Store them in .env file:

```
AWS_ACCESS_KEY_ID=my..access..key
AWS_ENDPOINT_URL=https://s3.slices-be.eu
AWS_REGION=us-east-1
AWS_SECRET_ACCESS_KEY=my..secret..key
```

- 3/ Use:

```
from dotenv import load_dotenv
from s3torchconnector import S3MapDataset, S3IterableDataset

load_dotenv()

bucket_name = 'ilabt.imec.be-project-ilabt-dev'
prefix="folder"
dataset_uri=f"s3://{bucket_name}/{prefix}"

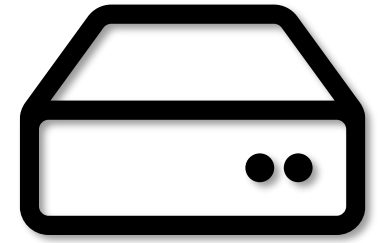
iterable_dataset = S3IterableDataset.from_prefix(dataset_uri, region=os.environ['AWS_REGION'])

# Datasets are also iterators.
for item in iterable_dataset:
    print(item.key)
```



```
folder/a.txt
folder/b.txt
```

Storage: importing your data



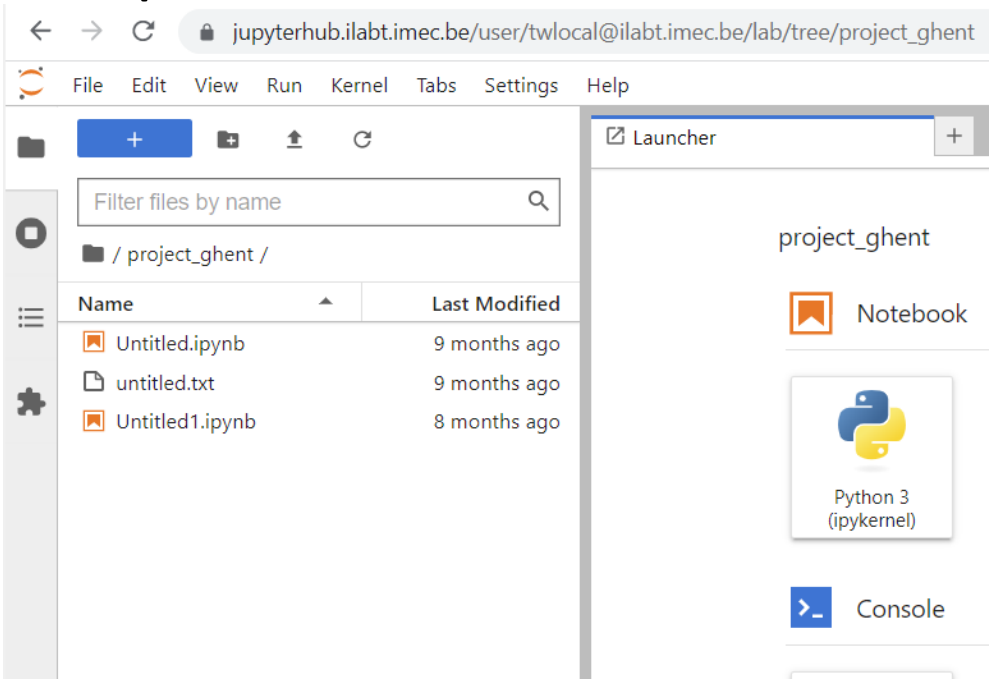
Small uploads?

Start a JupyterHub session and upload via the browser

Large uploads?

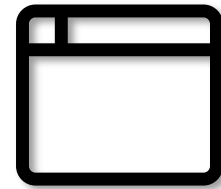
Connect via SFTP to your job

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



Documentation:

<https://doc.slices-ri.eu/BasicServices/AI/storage.html>



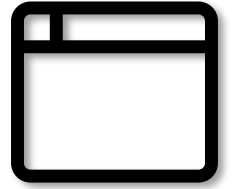
Exposing ports of your container

- 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:
 - Geo-limited IPv4 access (Belgium only)
 - use IDLab Antwerp VPN for access
- Madrid-based nodes: public IPv4

```
"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 port of your container



```
$ slices ai show jobs <job_id>
```

```
> slices ai show 2df72c
  Job ID: 2df72cd8-1baa-44f0-8611-8b4ae4ca74ec
  Name: JupyterHub-singleuser
  Description: Single Jupyter notebook instance for JupyterHub
  Project: proj_account.ilabt.imec.be_59z3qackp19veshw8yr8kws0yz
  User ID: user_account.ilabt.imec.be_0ma4rks06s9kxahxrgjhg6y41b
  Docker image: quay.io/jupyter/minimal-notebook:latest
  Command: start-notebook.sh --LabApp.check_for_updates_class=jupyter
  Status: RUNNING
  Cluster ID: 4
  Machine ID: m_be-gent1-ai-cluster4_074mc8ec30ag2ahcjzcb12z1b8
  Worker Name: gpulab4B
  Port Mappings: 8888 -> 5004, 22 -> 5005
  Worker Host: 4b.gpulab.ilabt.imec.be
```

Job State

Host Summary: 4B

Cluster: 4

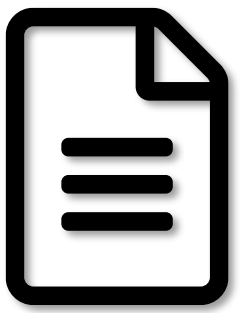
Worker: gpulab4B

SSH:  ssh -p 5005 root@4b.gpulab.ilabt.imec.be 

Port mappings

Port 8888 → 4b.gpulab.ilabt.imec.be:5004

Port 22 → 4b.gpulab.ilabt.imec.be:5005



Checking the logs of your container: CLI

`$ slices ai output <job_id>`

```
> slices ai output 2df72c
INFO[2026-06-05 08:35:28] Entered start.sh with args: start-notebook.sh --LabApp.check_for_updates_class=jupyterlab.NeverCheckForUpdate --notebook-dir=/
INFO[2026-06-05 08:35:28] Running hooks in: /usr/local/bin/start-notebook.d as uid: 0 gid: 0
INFO[2026-06-05 08:35:28] Done running hooks in: /usr/local/bin/start-notebook.d
INFO[2026-06-05 08:35:28] Updated the jovyan user:
INFO[2026-06-05 08:35:28] - username: jovyan -> twlocal
INFO[2026-06-05 08:35:28] - home dir: /home/jovyan -> /home/twlocal
INFO[2026-06-05 08:35:29] Update twlocal's UID:GID to 1000:7123
userdel: group twlocal not removed because it is not the primary group of user twlocal.
INFO[2026-06-05 08:35:29] Attempting to copy /home/jovyan to /home/twlocal...
INFO[2026-06-05 08:35:29] Success!
INFO[2026-06-05 08:35:29] Changing working directory to /home/twlocal/
INFO[2026-06-05 08:35:29] Granting twlocal passwordless sudo rights!
INFO[2026-06-05 08:35:29] Running hooks in: /usr/local/bin/before-notebook.d as uid: 0 gid: 0
INFO[2026-06-05 08:35:29] Sourcing shell script: /usr/local/bin/before-notebook.d/1
```

Checking the logs of your container: website



← Job 2df72cd8-1baa-44f0-8611-8b4ae4ca74ec (RUNNING

Refresh

Cancel

...

Creator: twlocal

Project: twlocalproj

Name: JupyterHub-singleuser

Description: Single Jupyter notebook instance for JupyterHub

General Info

Logs

Debugging Logs

Usage Graphs

Raw Job JSON

Console

Admin

Scheduling

Start time

→ End time



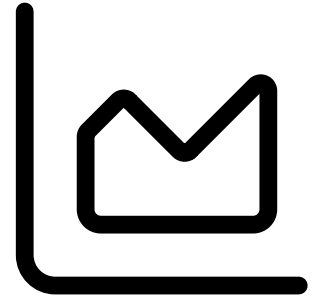
Live

Manual scroll

0 loaded of 117 total

```
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] Entered start.sh with args: start-notebook.sh --LabApp.check_for_updates_class=jupyterlab.NeverCheckForUpdate --
notebook-dir=/
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] Running hooks in: /usr/local/bin/start-notebook.d as uid: 0 gid: 0
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] Done running hooks in: /usr/local/bin/start-notebook.d
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] Updated the jovyan user:
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] - username: jovyan      -> twlocal
06/05/2026, 12:35:28.000 INFO[2026-06-05 08:35:28] - home dir: /home/jovyan -> /home/twlocal
06/05/2026, 12:35:29.000 INFO[2026-06-05 08:35:29] Update twlocal's UID:GID to 1000:7123
06/05/2026, 12:35:29.000 userdel: group twlocal not removed because it is not the primary group of user twlocal.
06/05/2026, 12:35:29.000 INFO[2026-06-05 08:35:29] Attempting to copy /home/jovyan to /home/twlocal...
```

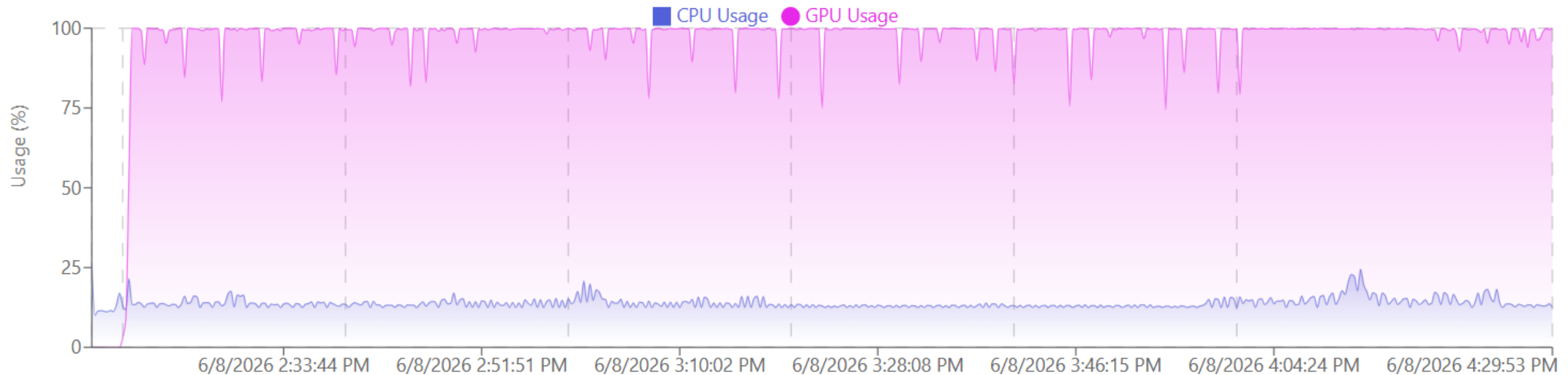
Checking the resource usage of your job

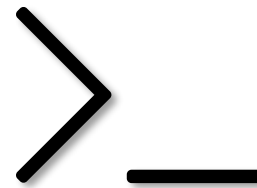


General Info Logs Debugging Logs **Usage Graphs** Raw Job JSON Console

▼ CPU/GPU Utilization

CPU ▼ GPU ▼ Network ▼ Other ▼





SSH-access: CLI

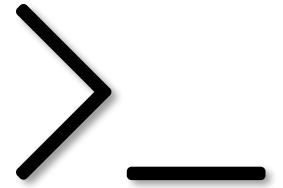
```
$ slices ai ssh <job-id>
```

```
> slices ai ssh 95288e
ssh -p 5010 -J proxy@bastion2.slices-be.eu root@4c.gpulab.ilabt.imec.be
Last login: Fri Jun  5 09:52:34 2026 from 172.17.0.1

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@9f53f06851fc:~#
```

SSH-access: website



← Job 2df72cd8-1baa-44f0-8611-8b4ae4ca74ec

🟢 RUNNING

🔄 Refresh

🛑 Cancel



Creator: twlocal

Name: JupyterHub-singleuser

Project: twlocalproj

Description: Single Jupyter notebook instance for JupyterHub

📘 General Info

📄 Logs

🔍 Debugging Logs

📊 Usage Graphs

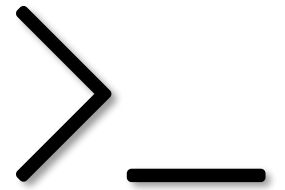
{ } Raw Job JSON

🖥️ Console

🔑 Admin

🕒 Scheduling

```
root@bc242f9676fb:~# ps ajfx
  PPID    PID    PGID    SID TTY        TPGID STAT   UID   TIME COMMAND
    0        1        1        1 ?          -1 Ss       0   0:00 tini -g -- start.sh start-notebook.sh --LabApp.check_for_updates_class=jupyterlab.NeverCheckForUpdate -
    1        7        7        1 ?          -1 S        0   0:00 sudo --preserve-env --set-home --user twlocal LD_LIBRARY_PATH= PATH=/opt/conda/bin:/opt/conda/condabin:
    7       79        7        1 ?          -1 S1      1000 0:04  \_ /opt/conda/bin/python3.13 /opt/conda/bin/jupyterhub-singleuser --LabApp.check_for_updates_class=jup
    1       52       51       51 ?          -1 S        0   0:00 sshd: /gpulab-ssh/sbin/sshd -E /gpulab-ssh/var/log [listener] 0 of 10-100 startups
   52       94       94       94 ?          -1 Ss       0   0:00  \_ sshd-session: root [priv]
   94       96       94       94 ?          -1 S        0   0:00    \_ sshd-session: root@pts/0
   96       97       97       97 pts/0      106 Ss       0   0:00        \_ -bash
   97      106      106       97 pts/0      106 R+       0   0:00          \_ ps ajfx
root@bc242f9676fb:~#
```

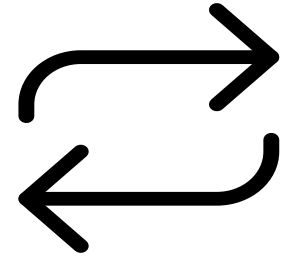


SSH-access: how does this work?

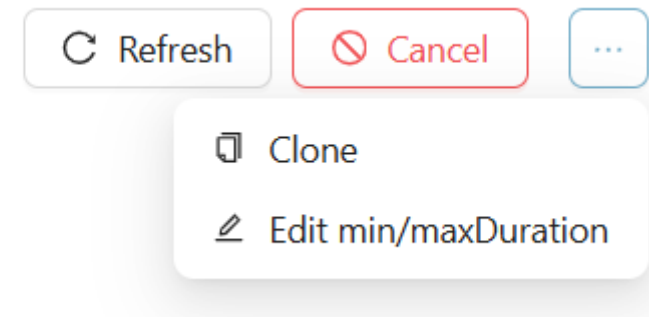
- An SSH-server is injected into your container
- Authentication via SSH public key
 - The `slices ai` CLI automatically adds SSH keys registered in `slices pubkey`
 - The AI infrastructure automatically adds an SSH-key for the console on the website
 - Register extra ssh keys via `request → extra → sshPubKeys`

```
"extra": {  
  "sshPubKeys": [  
    "ssh-ed25519 AAAAC3NzaC..."  
  ]  
}
```

Repeating a job



- Via the CLI: execute the `slices ai submit` command again
- Via the website: use the Clone option in the menu



Jupyterhub: <https://jupyterhub.ai.slices-ri.eu>

Project

Current project: **twlocalproj** [↔ Switch project](#)

Docker settings

Choose a default Docker Image:

Tensorflow Stack ⓘ
TensorFlow and Keras ML libraries.
Based on Scientific Python Stack.
[latest](#) [cuda](#)

Tensorflow + VSCode ⓘ
TensorFlow and Keras ML libraries.
Based on Scientific Python Stack. With VSCode.
[latest](#) [cuda](#)

Minimal Stack ⓘ
Basic Python and Tex packages.
[latest](#)

PyTorch Stack ⓘ
PyTorch libraries. Based on Scientific Python Stack.
[latest](#) [cuda12](#) [cuda13](#)

PyTorch + VSCode ⓘ
PyTorch libraries. Based on Scientific Python Stack. With VSCode.
[latest](#) [cuda11](#) [cuda12](#)

Scientific Python Stack ⓘ
Popular packages from the scientific Python ecosystem. Based on Minimal Stack.
[latest](#)

▼ Show more Docker images ▼

[Load Configuration](#) [Save Configuration](#)

Or specify your custom Docker Image:

Requested resources

Storage

- ☐ Mount Antwerp Project storage to /project_antwerp ⓘ
- ☒ Mount Ghent Project storage to /project_ghent ⓘ
- ☐ Mount Madrid Project storage to /project_madrid ⓘ
- ☐ Mount Scratch storage to /project_scratch ⓘ
- ☐ Mount Antwerp Dataset storage to /datasets_antwerp ⓘ
- ☐ Mount Ghent Dataset storage to /datasets_ghent ⓘ

Number of resources

#CPU's:

#CPU Memory:

GB

#GPU's:

Cluster ID:

Currently available: [Overview](#)

29 GPU's, **80** CPU's, **633** GB of CPU memory.

[Show Advanced Options](#)

[▶ Start](#)

JupyterHub: how does this work?



- Generates and starts a 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!
 - Custom docker images must descend from `jupyter/base-notebook`



JupyterHub: available images

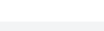
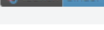
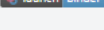
Tensorflow Stack ⓘ TensorFlow and Keras ML libraries. Based on Scientific Python Stack. latest cuda 	PyTorch Stack ⓘ PyTorch libraries. Based on Scientific Python Stack. latest cuda12 cuda13 	R Stack ⓘ R interpreter with popular packages from the R ecosystem. Based on Minimal Stack. latest 	Julia Stack ⓘ Includes popular packages from the Julia ecosystem latest 
Tensorflow + VSCode ⓘ TensorFlow and Keras ML libraries. Based on Scientific Python Stack. With TensorFlow VSCode. latest cuda 	PyTorch + VSCode ⓘ PyTorch libraries. Based on Scientific Python Stack. With VSCode. latest cuda11 cuda12 	Data Science Stack ⓘ Everything in the jupyter/scipy-notebook , jupyter/r-notebook , and jupyter/julia-notebook images. latest 	Spark Stack ⓘ Includes Python support for Apache Spark latest 
Minimal Stack ⓘ Basic Python and Tex packages. latest 	Scientific Python Stack ⓘ Popular packages from the scientific Python ecosystem. Based on Minimal Stack. latest 	All Spark Stack ⓘ Includes Python and R support for Apache Spark latest 	

JupyterHub: need a custom software stack?

- Docker image needs to descend from `jupyter/docker-stacks`
- Lots of customization examples documented at <https://jupyter-docker-stacks.readthedocs.io/en/latest/using/>
- Check for ‘Community stacks’ first on <https://jupyter-docker-stacks.readthedocs.io/en/latest/using/>

Community Stacks

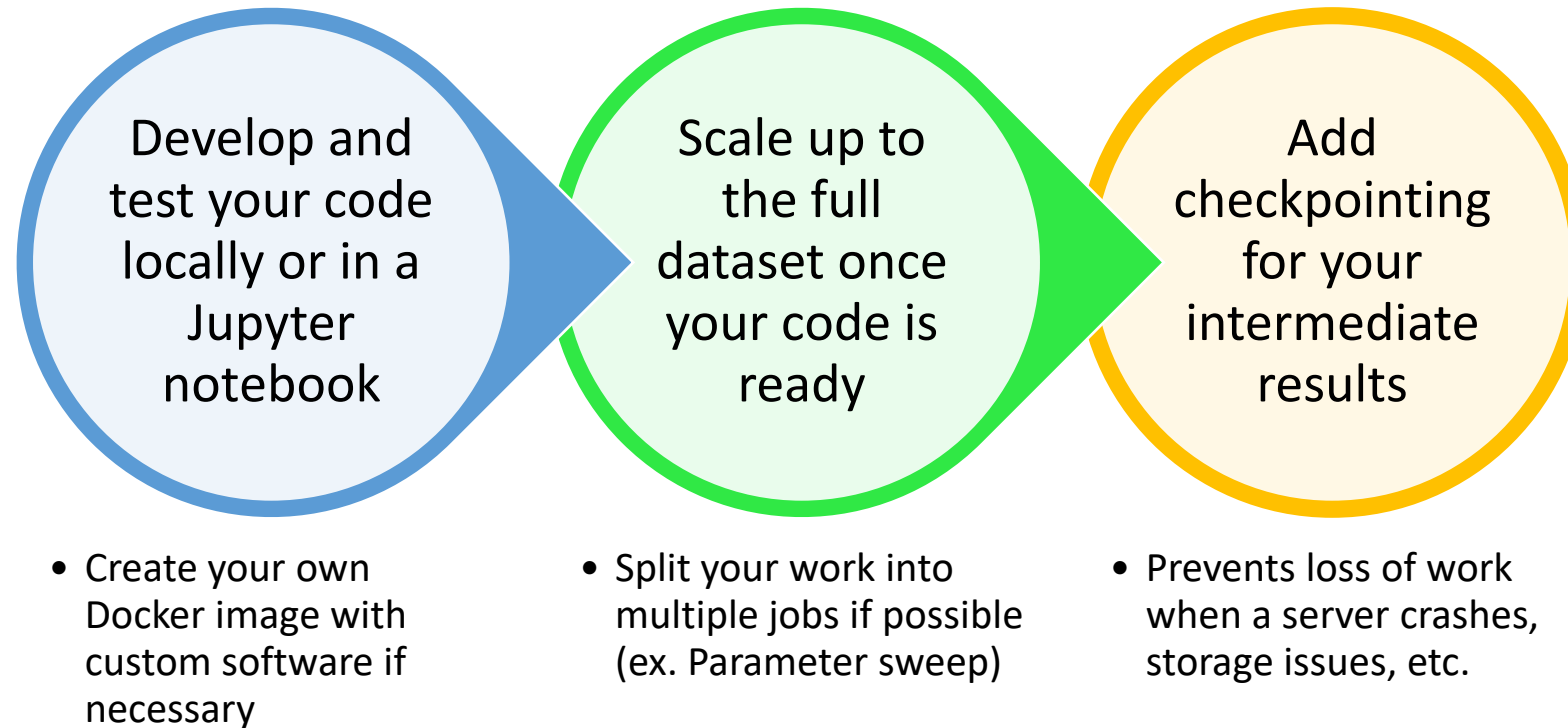
The core stacks are but a tiny sample of what’s possible when combining Jupyter with other technologies. We encourage members of the Jupyter community to create their own stacks based on the core images and link them below. See the [contributing guide](#) for information about how to create your own Jupyter Docker Stack.

Flavor	Binder	Description
csharp		More than 200 Jupyter Notebooks with example C# code
education		nbgrader and RISE on top of the datascience-notebook image
ihaskell		Based on IHaskell . Includes popular packages and example notebooks
java		IJava kernel on top of the minimal-notebook image
sage		sagemath kernel on top of the minimal-notebook image
cgspatial		Major geospatial Python & R libraries on top of the datascience-notebook image
kotlin		Kotlin kernel for Jupyter/IPython on top of the base-notebook image
transformers		Transformers and NLP libraries such as Tensorflow , Keras , Jax and PyTorch
scraper		Scraper tools (selenium , chromedriver , beatifulsoup4 , requests) on minimal-notebook image
almond		Scala kernel for Jupyter using Almond on top of the base-notebook image
lisp-stat		Common Lisp statistical computing environment on top of the minimal-notebook image
sequencing		Collection for bioinformatics sequencing data analysis, covering bulk RNA-seq, single-cell RNA-seq, spatial sequencing, and multi-omics

Fair Usage Policy

- Non-interactive 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

Best practices



Get Started!

- **Documentation:** <https://doc.slices-ri.eu/BasicServices/AI/>
- **Website:** <https://ai.slices-ri.eu>
- **JupyterHub:** <https://jupyterhub.ai.slices-ri.eu>

SUPPORT 

`user-support@slices-ri.eu`

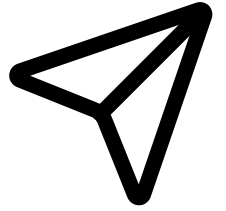
Hands on!



Start your first Jupyter session

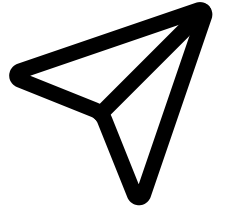


- Go to <https://jupyterhub.ai.slices-ri.eu>
- Choose a default Docker image of your liking
- Select the storage you want to use
 - Note: illegal combinations are automatically prevented
- Define number of CPU's/GPU's memory
- Optional: define a cluster ID
- Click 
- Create or Upload a Python script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



Submit your first job via the website

- Go to <https://ai.slices-ri.eu>
- Click on 
- 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'



Submit your first job via the CLI

- Install the CLI:

```
$ pip install slices-cli-ai --extra-index-url=https://doc.slices-ri.eu/pypi/
```

```
> slices ai
```

```
Usage: slices ai [OPTIONS] COMMAND [ARGS]...
```

```
AI Infrastructure Service Commands.
```

Options

<code>--site, --site-id</code>	TEXT	AI Infrastructure Service site ID. [env var: SLICES_AI_S
<code>--help</code>	<code>-h</code>	Show this message and exit.

Auxiliary commands

<code>wait</code>	Wait until the requested status has been reached (or can never be reached).
<code>modify</code>	Change maxDuration, minDuration or notAfter of a job.
<code>debug</code>	Show internal event log for a job.

Lifecycle commands

<code>submit</code>	Submit a job described in a file.
<code>cancel</code>	Cancel a job.
<code>rm</code>	Delete a job.
<code>halt</code>	Halt a job. Halted jobs may be automatically re-QUEUED later.
<code>hold</code>	Hold a QUEUED job, preventing it from running.
<code>requeue</code>	Requeue/Release a held job, QUEUEing it, so it waits to run.

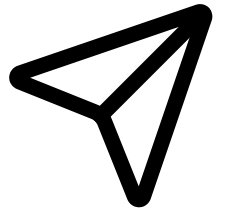
Submit your first job via the CLI

- Check availability with ``slices ai clusters``:

```
> slices ai clusters
```

Clusters

ID	Site	GPU Model	CUDA	# Nodes	GPUs	CPUs	Memory (GB)
3	be-gent1	NVIDIA RTX 2080 Ti (11GB)	13.0	1	1/1	12/12	30/30
4	be-gent1	NVIDIA GTX 1080 Ti (11GB)	13.0	3	29/33	81/96	636/750
5	be-gent1			0	0/0	0/0	0/0
6	be-gent1	NVIDIA V100-SXM3 (32GB)	13.0	1	1/16	2/96	1109/1509
8	be-gent1	NVIDIA A40 (45GB)	13.0	3	0/12	26/144	894/1504
9	be-gent1	NVIDIA RTX 5090 (32GB)	13.0	1	0/1	6/64	176/502
10	be-gent1	NVIDIA RTX A6000 (48GB)	13.0	1	0/2	20/32	32/124
11	be-gent1	NVIDIA H200 NVL (141GB)	13.0	2	1/10	66/192	2599/3019
101	be-antwerpen1	NVIDIA A100 (80GB)	12.2	2	1/8	139/192	514/1004
102	be-antwerpen1	NVIDIA RTX 4000 (8GB)	12.2	2	4/8	46/80	292/500
103	be-antwerpen1	NVIDIA V100-SXM3 (32GB), NVIDIA V100-SXM2 (32GB)	12.2	2	5/24	51/176	1545/2011
104	be-antwerpen1			0	0/0	0/0	0/0
201	es-madrid1	NVIDIA H200 (141GB)	12.8	1	0/8	144/224	1561/1985



Submit your first job via the CLI

- Submit via `slices ai submit`:

```
> slices ai submit --help
```

```
Usage: slices ai submit [OPTIONS] FILE
```

```
Submit a job described in a file.
```

Arguments

```
*   file          FILENAME  File with job definition. [required]
```

Options

```
--wait-run          Wait till the job is running.  
--wait-done         Wait till the job is done.  
--ssh-key           SSH-KEY  (Extra) SSH public key to register for login to the created job.  
--ssh-key-file      SSH-KEY  (Extra) SSH public key file for login to the created job. [env var: SLICES_SSH_KEY_FILE]  
--experiment        TEXT    Name of the experiment (default: job.name).  
--help             -h      Show this message and exit.
```

Creating your first jobDefinition

- Fill in the 'Create Job' form on the website, and click 'Download jobDefinition File': **Resulting JSON:**

[Download jobDefinition File](#) [Save in Browser](#)

```

{
  "root": {
    "name": "NVIDIA SMI"
    "description": "Writes the output of the command 'nvidia-smi' to t ..."
    "request": {
      "docker": {
        "image": "nvidia/cuda:12.1.0-runtime-ubuntu22.04"
        "command": "nvidia-smi"
      }
    }
  }
}
```

- Manually tweak your jobDefinition file if necessary

Submit your first job via the CLI

- `$ slices ai submit`

```
> slices ai submit nvidia_smi.json  
✦ Created Job 207c7339-a1f2-42ce-a398-a59fe15cf419
```

- `$ slices ai wait <job id>`

```
> slices ai wait 207c73  
2026-06-05 11:36:54 +0200 - Job status: STARTING  
✦ Job 207c7339-a1f2-42ce-a398-a59fe15cf419 reached status FINISHED.
```

Check the output

\$ slices ai output <job id>

```
> slices ai output 207c73
=====
== CUDA ==
=====

CUDA Version 12.1.0

Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.
By pulling and using the container, you accept the terms and conditions of this license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

*****
** DEPRECATION NOTICE! **
*****
THIS IMAGE IS DEPRECATED and is scheduled for DELETION.
  https://gitlab.com/nvidia/container-images/cuda/blob/master/doc/support-policy.md

Fri Jun  5 09:37:17 2026
+-----+
| NVIDIA-SMI 580.95.05                | Driver Version: 580.95.05   | CUDA Version: 13.0     |
+-----+-----+-----+
| GPU   Name                               | Persistence-M | Bus-Id               | Disp.A | Volatile Uncorr. ECC |
+-----+-----+-----+-----+-----+-----+

```

Advanced Topics: parameter sweeping

When possible, split your workload into multiple jobs

1. Generate the correct jobDefinition JSON file in the language of your choice
2. Pass it to the CLI via `slices ai submit`

```
#!/bin/bash

for param in {100..1000..50}; do
    echo "{
      \"jobDefinition\": {
        \"name\": \"Sweep $param\",
        \"command\": \"/project_ghent/my_script.sh $param\"
      }
    }" | slices ai submit /dev/stdin
done
```



