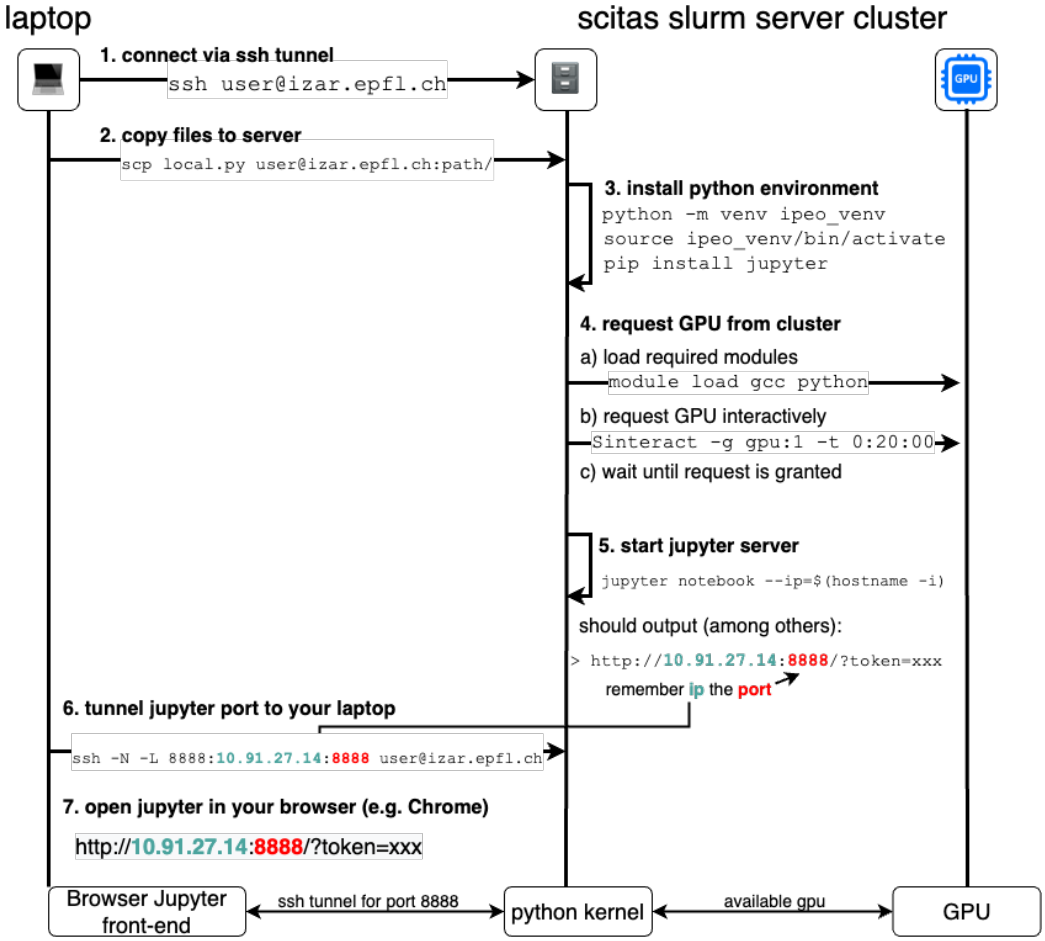
SCITAS GPU-enabled Jupyter Server

Introduction

Scitas provides access to GPU machines on the common cluster. This document shows you how to start a jupyter server on this infrastructure. To run a script, we refer you to the SCITAS documentation (and don't hesitate to ask the TAs!) but the first 4 steps are similar. You can store up to 100 Gb in your personal space (again, we refer you to the Scitas documentation for more details).
The **General Workflow** and **General Steps** sections provide a schematic and high level steps for a general understanding.
Detailed commands are in the **Detailed Execution Steps** section.

General Workflow



We can conceptually think of three entities that interact: your laptop (left), a scitas interface server (center) accessible under `izar.epfl.ch`, and a

cluster of GPU machines (right) that are managed by a Slurm Workload Manager. At the end, your laptop browser will display the jupyter front-end (like in the last exercises), but the code will be executed by a python kernel on the cluster (center). This kernel will be able to use a GPU provided by SLURM.

General Steps

All commands in `typewriter style` can be executed in a terminal window (more parameters and details later).
The general steps are:

1. connecting to a terminal in the scitas cluster via `ssh` (ssh stands for "secure shell").
2. copying files from your local computer (e.g. local.py) to the scitas computer with `scp` (scp stands for "secure copy protocol")
3. setting up a python environment with virtual environment `venv`, which is a lightweight alternative to anaconda. This involves creating the environment (named `ipeo_venv`), activating it with `source activate`, and installing required packages with `pip`.
4. requesting a GPU with the `Sinteract` command and waiting until this request is granted.
5. starting the jupyter server on the scitas cluster.
6. tunneling the port and ip connection back to your local laptop
7. opening jupyter in the browser on your laptop.

Detailed Execution Steps

The first four steps are the same if you want to run a job via SLURM or if you want to use a jupyter notebook.

1 Connect to the Scitas Cluster via ssh

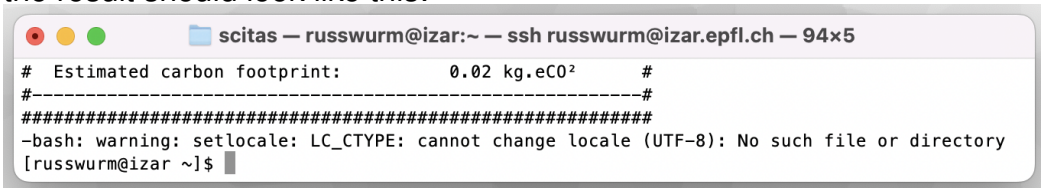
1.1 make sure, you are in eduroam or connected via VPN.
1.2 open a terminal window:

- Linux: press Ctrl + Alt + T
- macOS: Cmd + Space, type "Terminal"
- Windows: open Windows PowerShell (search in the search bar)

1.3 use `ssh` to open a terminal on the scitas GPU cluster (named izar). Replace `user` with your GASPAR username. The terminal will ask you for your GASPAR password.

```
ssh user@izar.epfl.ch
```

1.4 the result should look like this:



2 Copy files to the Scitas Cluster

To work on the server, you will need to copy files with `scp`. For instance, `ex8.ipynb` or additional datasets. You execute the `scp` command on your laptop.

2.1 to copy files, you can use the `scp` program. This line will copy the local `ex8.ipynb` file into a `exercise_7` folder in your home directory (located under `/home/user/`). You may have to create the folder `exercise_7` first by calling `mkdir exercise_7` in the terminal.

```
scp ex8.ipynb user@izar.epfl.ch:exercise_7/
```

3 Set up a Python environment on the Scitas Cluster

You will need to create a python environment. Execute these steps on the scitas server. These steps are identical (except 3.1) to setting up a virtual environment on your own computer via terminal.

3.1 load `gcc` and `python` modules in the scitas node. `gcc` is necessary to install (compile) new python packages.

```
module load gcc
module load python
```

3.2 create a new empty environment called `ipeo_venv`

```
python -m venv ipeo_venv
```

3.3 activate the environment by calling the `activate` script in the environment with the `source` program

```
source ipeo_venv/bin/activate
```

3.4 you can check if you use the correct `python` and `pip` programs by calling `which`

```
which python
which pip
```

this should print the full path to `python` and `pip` in your virtual python environment, e.g., `/home/user/ipeo_venv/bin/python`

3.5 make sure your pip is up-to-date

```
pip install --upgrade pip
```

3.6 install python dependencies. If you're running a notebook, you can install those dependencies within jupyter later as well.

```
pip install jupyter
```

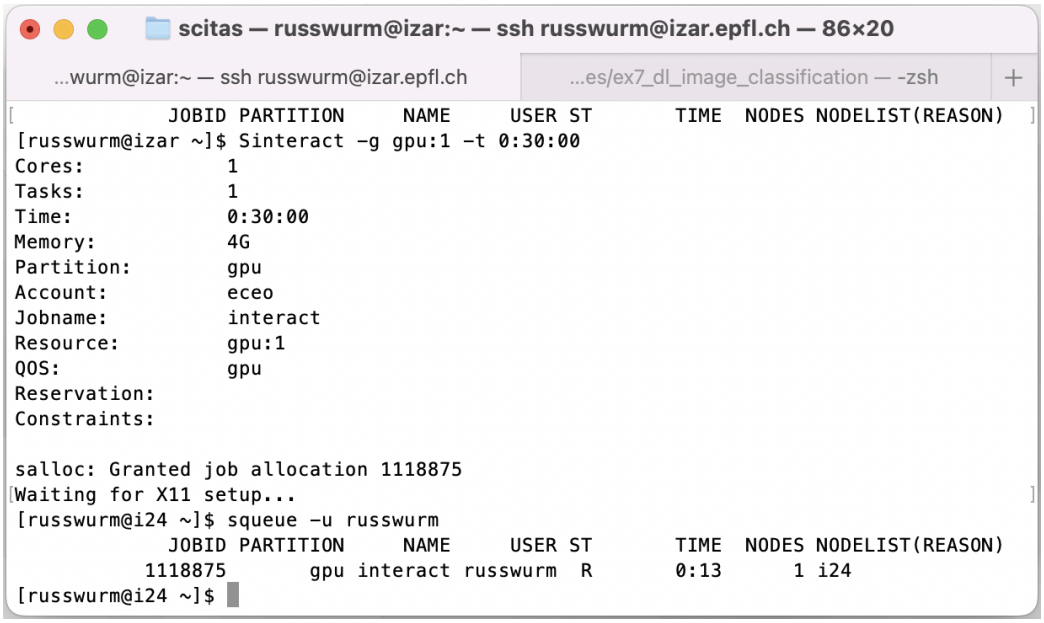
4 Request a GPU via SLURM

Izar cluster is available for educational purposes so you can request a GPU to train your models.

4.1 To run a notebook, we need to request a GPU in an interactive environment with `Sinteract`. The parameter `-t` defines the time frame (three hours). You also **have** to specify the account name with `-a env540` (it is the same for everyone in this course).

```
Sinteract -a env540 -g gpu:1 -t 3:00:00
```

4.2 you can see running SLURM processes in the SLURM queue with `squeue -u $(whoami)`. A successfully granted GPU should look like this:



5 Starting Jupyter Server on Scitas

Now, we need to start the Jupyter server on the scitas node. Execute this code on the izar terminal. Make sure the python environment is activated (should be indicated by `(ipeo_venv)` before your username)

5.1 start the jupyter server. The option `--ip=$(hostname -i)` passes explicitly the ip address of the scitas node you are on to the jupyter server (the url `izar.epfl.ch` can forward you to multiple internal interface servers with different ips).

```
jupyter notebook --ip=$(hostname -i)
```

5.2 the correct output should look like this

```
(ipeo_venv) [russwurm@izar ~]$ jupyter notebook --ip=$(hostname -i)
[I 15:50:43.285 NotebookApp] Serving notebooks from local directory: /home/russwurm
[I 15:50:43.285 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 15:50:43.285 NotebookApp] http://10.91.25.1:8888/?token=2ad8f7f50e5da03f5845e99046f3
9c39d62137cbf75452cd
[I 15:50:43.285 NotebookApp]  or http://127.0.0.1:8888/?token=2ad8f7f50e5da03f5845e9904
6f39c39d62137cbf75452cd
[I 15:50:43.285 NotebookApp] Use Control-C to stop this server and shut down all kernel
s (twice to skip confirmation).
[W 15:50:43.298 NotebookApp] No web browser found: could not locate runnable browser.
[C 15:50:43.298 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/russwurm/.local/share/jupyter/runtime/nbserver-5879-open.html
    Or copy and paste one of these URLs:
        http://10.91.25.1:8888/?token=2ad8f7f50e5da03f5845e99046f39c39d62137cbf75452cd
     or http://127.0.0.1:8888/?token=2ad8f7f50e5da03f5845e99046f39c39d62137cbf75452cd
```

5.3 look at the output and remember the ip address (here `10.91.25.1`) and port (here `8888`), we need it in the next section.

6 Tunnel the ip and port to your laptop via ssh

Usually, HPC servers do not expose web interfaces (like jupyter notebooks) to the public, as it is a security risk. So, we need to tunnel the jupyter web interface through a secure shell with `ssh`.

6.1 open a new terminal on your laptop. This terminal must remain open for the tunnel to exists.

6.2 connect to `izar.epfl.ch` again but also tunnel ip address (here `10.91.25.1`) and port (here `8888`) with `-L`. Use the ip and port from above (Step 5.3). This command has no output. As long as the terminal window remains open, the tunnel will be active.

```
ssh -N -L 8888:10.91.25.1:8888 user@izar.epfl.ch
```

7 Open Jupyter on your Browser

Now, that you tunneled the jupyter web interface to your laptop, it should be available under `localhost` like a server you started on your local machine (like the last weeks).

7.1 open the url `http://localhost:8888/` in your browser and provide the token `2ad8f7...` from Step 5.2 if asked.

7.2 open a new jupyter notebook and test if you have a GPU available with `!nvidia-smi` in a jupyter cell

In [1]:
```
!nvidia-smi
```

```
Tue Nov  1 17:22:58 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 455.23.05    Driver Version: 455.23.05    CUDA Version: 11.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:86:00.0 Off |                  Off |
| N/A   39C    P0    25W / 250W |      0MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```