ML3D Compute Cluster Guide

We provide access to a small compute cluster for the exercises and projects, consisting of a login node and 4 compute nodes with one dedicated RTX 3090 GPU each. Please send us a short email with your name and preferred username so we can add you as a user.

Login Node and Sessions

To access the cluster, connect to the login node first. Please note that you’ll need to use TUM eduVPN if you want to connect to the cluster outside TUM.

```
$ ssh <user>@ml3d.vc.in.tum.de
```

This login node has limited resources and no GPU installed. DO NOT RUN ANY CODE HERE. Instead, first start an interactive session on one of the compute nodes:

```
$ salloc --gpus=1
```

This will initiate a new session for your user. Each session has a maximum runtime of 8 hours after which it will get killed. If all 4 compute nodes are in use right now, your request will be queued until one of the previous sessions times out or is ended by the other user. You can take a look at the current queue from the login node:

```
$ squeue
```

If you are done working in your session, just exit it which will take you back to the login node:

```
$ exit
```

Directories

These directories are available across all nodes:

/rhome/<user>	This is your personal home directory. You can use it to store your code, python environments, and other small files like training results.
/cluster/51 /cluster/52 /cluster/53 /cluster/54	These are the hard drives of the compute nodes. Here, you can store your datasets and other large files you might need during training. All of these four hard drives are available across all four nodes.

(Only once) Setting up the environment for the exercise

```
$ ssh <user>@ml3d.vc.in.tum.de
```

Create a folder for your data on one of the compute nodes' hard drives. You can choose any node (51, 52, 53, or 54) for that since all hard drives are accessible from all nodes. We recommend using the node that has the most freespace (you can check with `df -h`).

```
$ mkdir /cluster/51/<user>

# We don't want other users to see our solutions
$ chmod 700 /cluster/51/<user>
```

Note that stored data that don't follow `/cluster/[51-54]/<user>` will be **removed** regularly.

Copy all exercise files to the cluster. We move the code to a hard drive since you will download the datasets into the same directory.

```
$ scp E2.zip <user>@ml3d.vc.in.tum.de:
$ ssh <user>@ml3d.vc.in.tum.de
$ mv E2.zip /cluster/51/<user>
$ cd /cluster/51/<user>
$ unzip E2.zip
$ cd E2/
```

(Option 1) Setup python environment with Conda

Please run the following part in the compute node

```
$ wget
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
$ bash Miniconda3-latest-Linux-x86_64.sh
$ eval "$(conda shell.bash hook)"
$ conda create -n ml3d python=3.8
$ cd E2/

$ conda activate ml3d
# Install all the packages specify by requirements.txt
$ pip install -r requirements.txt
```

Run the jupyter notebook:

```
$ conda activate ml3d
$ jupyter notebook --no-browser --ip=0.0.0.0 --port 8888
```

(Option 2) Setup python environment with Poetry

Please run the following part in the compute node. Check out the official documentation to know more about poetry for more information.

```
$ curl -sSL https://install.python-poetry.org | python3 -
$ export PATH="/rhome/<user>/.local/bin:$PATH"

# Show all the commands provided py poetry
$ poetry

# Install all the packages specify by pyproject.toml
$ cd E2/
$ poetry install
```

Run the jupyter notebook:

```
$ poetry run jupyter notebook --no-browser --ip=0.0.0.0 --port 8888
```

Connect to the jupyter notebook from your local machine

Forward the port by create ssh tunnel (make sure you are connected to TUM VPN) when login with ssh:

```
$ ssh -L 8888:TUINI15-VC51.vc.in.tum.de:8888 <user>@ml3d.vc.in.tum.de
```

You can then open the notebook on your local machine at `localhost:8888`
Note that here `VC[51,52,53,54]` has to be set according to the node that is allocated to you, and the port here (8888) should be adjusted based on your juypter notebook port.

Don’t run vscode server or other IDE server on the login node

Since the login node will crash if everyone is running **vscode or cursor (or other IDE)** server on it, it’s **not allowed** to do so. You should write your code on your local machine.

Here are a few ways to write and upload your codes to the server:

1. (Linux based) mount the directory to your local machine using sshfs.

Execute this on your machine to install sshfs:

```
$ sudo apt install sshfs
```

Create a local folder in which you want to mount the remote directory and mount it with sshfs:

```
$ sudo mkdir /home/{your_username}/{your_wished_folder_name}
$ sudo chmod 777 /home/{your_username}/{your_wished_folder_name}
$ sudo sshfs -o allow_other,ssh_command='ssh -4'
{your_ml3d_cluster_username}@ml3d.vc.in.tum.de:/rhome/{your_ml3d_cluster
_username} /home/{your_username}/{your_desired_folder_name}
```

2. Use SFTP (a VSCode extension)

Install the SFTP extension in your vscode. Set up the SFTP config following the template (Ctrl+Shift+P and select "SFTP:Config") as follows. This will automatically upload your code to the server when saving.

```
{
    "name": "{exmaple config}",
    "host": "ml3d.vc.in.tum.de",
    "protocol": "sftp",
    "port": 22,
    "username": " {your_ml3d_cluster_username}",
    "privateKeyPath": "~/.ssh/id_rsa",
    "remotePath": "/rhome/
{your_ml3d_cluster_username}/{your_desired_location}",
    "uploadOnSave": true,
    "openSsh": false,
    "useTempFile": false,
    "ignore": [
        ".vscode",
        ".DS_Store",
        ".ipynb_checkpoints",
        "__pycache__",
        "assets",
        "output",
        "backup",
        "*.egg-info",
        "logs",
        "src",
        "runs",
        "wandb",
        ".log",
        "data"
    ],
    "connectTimeout": 100000
}
```

3. Use tools like FileZilla or rsync command to upload your code to the server manually.

4. Use editor tools like vim/emacs to write your code on the server