NYU HPC Cloud Bursting Student Guide (2026.01)

This guide covers the specific infrastructure for **Embodied Learning and Vision** .

Please reach out to TAs if you have any questions about HPC before emailing HPC staff.

1. Course Account & Access Information

All compute resources for this semester are accessed via the **Cloud Bursting Open OnDemand (OOD)** portal.

- **OOD URL:** https://ood-burst-001.hpc.nyu.edu/
- **VPN:** You **must** be connected to the NYU VPN if accessing from off-campus.
- **Slurm Account:** ds_ga_3001_003-2026sp
- **Allocation:** 300 GPU hours (18,000 minutes) per student.

Available Partitions

Partition	**Resource**	**Use Case**
interactive	Varies	Quick testing/debugging
n2c48m24	CPU Only	Data preprocessing
g2-standard-12	1x L4 GPU	Standard training
g2-standard-24	2x L4 GPUs	Distributed training
g2-standard-48	4x L4 GPUs	Distributed training
c12m85-a100-1	1x A100 (40GB)	Standard training
c24m170-a100-2	2x A100 (40GB)	Distributed training
n1s8-t4-1	1x T4 GPU	Light GPU tasks

2. Setup Environments

Step 1: Create a Writable Overlay

1. Open a **Terminal** in OOD (get a jupyter notebook → terminal).
2. Run these commands to create a 15GB persistent workspace:

```
cd /scratch/$USER
cp /share/apps/overlay-fs-ext3/overlay-15GB-500K.ext3.gz .
gunzip overlay-15GB-500K.ext3.gz
mv overlay-15GB-500K.ext3 my_env.ext3
```

Step 2: Set up Conda Inside the Container

Launch the container in "read-write" mode (rw):

```
singularity exec --nv \
--overlay /scratch/$USER/my_env.ext3:rw \
/share/apps/images/cuda12.1.1-cudnn8.9.0-devel-ubuntu22.04.2.sif  /bin/bash
```

Inside the container (Singularity>), set up Miniconda:

```
# Install Miniconda to the overlay
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -b -p /ext3/miniconda3

# Create an activation script
cat << 'EOF' > /ext3/env.sh
#!/bin/bash
source /ext3/miniconda3/etc/profile.d/conda.sh
export PATH=/ext3/miniconda3/bin:$PATH
EOF

source /ext3/env.sh
conda create -n my_env python=3.10 -y
conda activate my_env
```

Step 3: Install Packages

pip/conda install or use requirements.txt: `pip install -r requirements.txt`

```
torch

torchvision
```

```
torchaudio

wandb

matplotlib

numpy

tqdm

ipykernel
```

3. Submitting Jobs (Handling Spot Instances)

All cloud nodes are **GCP Spot Instances**, meaning they can be terminated at any time. You **must** use the --requeue flag, save checkpoints frequently and resume from checkpoints.

Example Slurm Script (run_job.slurm)

```
#!/bin/bash
#SBATCH --job-name=dl_train
#SBATCH --account=ds_ga_3001_003-2026sp
#SBATCH --partition=g2-standard-12
#SBATCH --gres=gpu:1
#SBATCH --time=00:10:00
#SBATCH --requeue
#SBATCH --output=slurm_%j.out

singularity exec --nv \
--overlay /scratch/$USER/my_env.ext3:ro \
/share/apps/images/cuda12.1.1-cudnn8.9.0-devel-ubuntu22.04.2.sif   \
    /bin/bash -c "source /ext3/env.sh; conda activate my_env; python train.py"
```

train.py example

```
import torch

print("CUDA available:", torch.cuda.is_available(), flush=True)
print("CUDA version:", torch.version.cuda, flush=True)
print("GPU count:", torch.cuda.device_count(), flush=True)

if torch.cuda.is_available():
    print("GPU name:", torch.cuda.get_device_name(0), flush=True)
```

Submit with: sbatch run_job.slurm

Tips: Weights and Bias is a useful tool for keeping track of your training! [Demo]

More tutorials can be found in https://docs.wandb.ai/tutorials

4. Data Transfer from Torch

To bring data from the main NYU Torch cluster to the Cloud Bursting nodes:

- **Command:** scp -rp dtn.torch.hpc.nyu.edu:/path/to/your/data .
- **Note:** This requires NYU MFA (Duo push).

5. Using Jupyter via OOD

1. Go to **Interactive Apps** -> **Jupyter Notebook**.
2. **Singularity Image:** /share/apps/images/cuda11.8.0-cudnn8-devel-ubuntu22.04.2.sif
3. **Overlay File:** /scratch/$USER/my_env.ext3:ro
4. Once the notebook opens, ensure you select your custom Conda kernel (you may need to install ipykernel).