WEXAC GPU Jupyter – Full Correct, Minimal & Reliable Workflow

This document describes the proven, manual workflow for running JupyterLab on a WEXAC GPU node and connecting to it from macOS / VS Code using SSH tunneling.

Step 0 – Clean Previous Jobs (MANDATORY)

- bjobs -u $USER (verify running jobs)
- bkill 0 (kill all jobs if needed)
- pkill -u $USER jupyter (kill stray Jupyter)

Step 1 – Request GPU Node (login2)

bsub -q interactive-gpu -R "rusage[mem=8GB]" -gpu "num=1:j_exclusive=no:gmem=8GB" -Is bash

Wait until you see: <<Starting on lgnXX>>

Step 2 – Activate Conda (GPU node)

```
module load miniconda
source /apps/easybd/programs/miniconda/24.11_environmentally/etc/profile.d/conda.sh
conda activate /home/projects/galvardi/yoado/.conda/envs/rec
```

Step 3 – Launch JupyterLab

```
PORT=8899
while lsof -nP -iTCP:$PORT -sTCP:LISTEN >/dev/null 2>&1; do PORT=$((PORT+1)); done
jupyter lab --no-browser --ip=127.0.0.1 --port=$PORT --ServerApp.port_retries=0
```

Step 4 – Tunnel login2 → GPU node

```
ssh wexac
PORT=8899 # must match Jupyter output
ssh -N -L 127.0.0.1:8899:127.0.0.1:$PORT lgnXX
```

Step 5 – Tunnel Mac → login2

ssh -N -L 8899:127.0.0.1:8899 wexac

Step 6 – Open in Browser / VS Code

http://127.0.0.1:8899/lab?token=...

VS Code → Jupyter → Select Kernel → Existing Jupyter Server → paste URL.

Common Errors

- 403 Forbidden → using https:// instead of http://
- Port already in use → kill stopped jobs (jobs -l)
- conda not found → forgot module load miniconda
- Bad forwarding spec → $PORT not set in that shell