1. Setup: AWS EC2 GPU Instance (g6.xlarge)

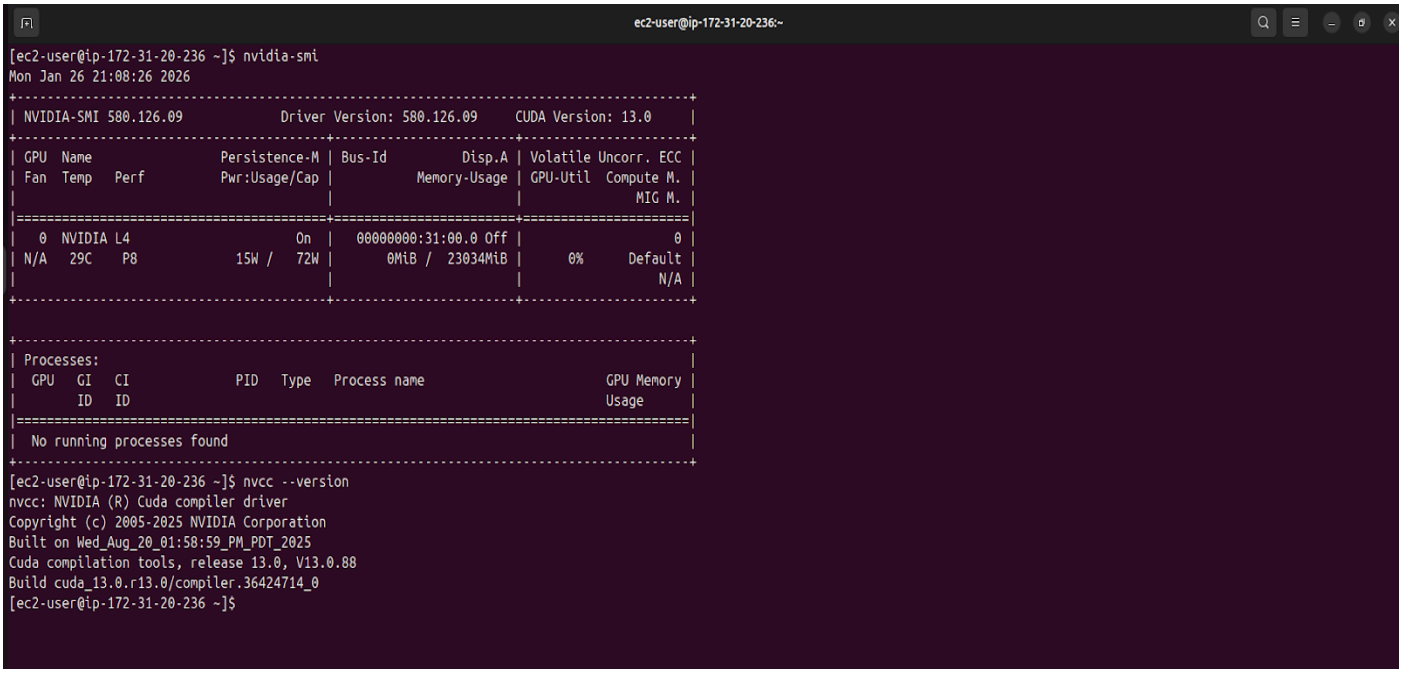
For cost-efficiency, I used a g6.xlarge single-GPU instance.

Launch Steps:

1. **EC2 Console** - Launch new instance
2. **AMI Selection** - Use "Deep Learning Base AMI with Single CUDA (Amazon Linux 2023) 20260120"
    - Alternatively, choose any base OS and install drivers manually
    - Note: Deep learning Ubuntu 22.X variant had NVIDIA driver detection issues. So this did not work for me
3. **Instance Type** - Select g6.xlarge
    - **Important**: Request vCPU quota increase beforehand (approval may take time; good luck if you account is new)
4. **SSH Access** - Configure key pair (.pem file)
5. **Network** - Default settings; customized security group to allow SSH from my public IP
6. **Storage** - gp3, 125 GB root volume

Verification:

After connecting via SSH, run `nvidia-smi` to confirm GPU availability.



In this instance, python was already installed.

2. Set up remote SSH access to your EC2 instance for VS Code

On your local machine: Get your public key:

cat .ssh/id_abc123.pub

Copy the entire output.

On your EC2 instance terminal: Add your public key to the authorized keys list:

echo "paste_your_public_key_here" >> .ssh/authorized_keys

Next, retrieve your EC2 instance's public IP address:

curl ifconfig.me

In VS Code:

1. Navigate to Remote Explorer
2. Click the + icon
3. Enter: `ssh <username>@your_ec2_public_ip`

You should now be connected.

Important: Each time you restart your EC2 instance, you'll need to get the new public IP (using `curl ifconfig.me`) and reconnect via SSH in VS Code, since the IP address typically changes.

3. Install Nsight Compute

Note: This guide focuses on Nsight Compute. While Nsight Systems is skipped here, it's worth mentioning that it's an excellent tool for analyzing both GPU and CPU performance and their interaction in great detail.

To verify Nsight Compute is available:

which ncu