1. Install NVIDA GPU Drivers:

Detect your device with:

> ubuntu-drivers devices and compatible drivers

Install recommended driver:

> sudo apt install nvidia-driver-470

Si arroja el siguiente error (o similar):

desviación de /usr/lib/i386-linux-gnu/libGL.so.1 a /usr/lib/i386-linux-gnu/libGL.so.1.distrib por nvidia-340

dpkg-divert: error: diferencia en el paquete

Remove previous installations with:

> apt remove -y --purge nvidia-\* libnvidia-ifr1-\*:i386 libnvidia-ifr1-\* nvidia-driver-\*

>

> dpkg-divert --remove "/usr/lib/i386-linux-gnu/libGL.so.1"

> dpkg-divert --remove "/usr/lib/x86\_64-linux-gnu/libEGL.so"

> dpkg-divert --remove "/usr/lib/x86\_64-linux-gnu/libEGL.so.1"

Then type:

sudo apt --fix-broken install

2. Install Cuda Drivers

Download Cuda Toolkit 9 from

<https://developer.nvidia.com/cuda-90-download-archive?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1704&target_type=deblocal>

and follow instructions there, for installing

dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 sudo dpkg --purge # remove previous cuda intallations

dpkg --install cuda-repo-ubuntu\*-9.0-local\*.deb

sudo apt-get update

sudo apt-get install cuda

Change environmental variables related to cuda on .bashrc:

export PATH=/usr/local/cuda-9.0/bin:$PATH

export LD\_LIBRARY\_PATH=/usr/local/cuda-9.0/lib64:$LD\_LIBRARY\_PATH

export CUDA\_HOME=/usr/local/cuda-9.0

3. Downgrade compilers

Another problem during installation is that gcc, gfortran compilers on ubuntu20.04 are too new for amber16:

#error -- unsupported GNU version! gcc versions later than 6 are not supported!

So you have to add gcc-6, gfortran-6 and g++-6 to the sources list

<https://askubuntu.com/questions/1236552/how-can-i-downgrade-gcc-to-version-6-on-20-04>

sudo vi /etc/apt/sources.list

and add deb http://dk.archive.ubuntu.com/ubuntu/ bionic main universe

sudo apt-get update

sudo apt install gcc-6 g++-6 gfortran-6

If there is conflict with gcc-6-base, remove it and try again

sudo apt remove gcc-6-base

sudo apt install gcc-6 g++-6 gfortran-6

then you will have to point to these old compilers (instead of the current 9.xxx versions in ubunto20.04) by creating symbolic links

sudo ln -sfn /usr/bin/gfortran-6 /usr/bin/gfortran

sudo ln -sfn /usr/bin/gcc-6 /usr/bin/gcc

sudo ln -sfn /usr/bin/g++-6 /usr/bin/g++

4. Install Amber

As of Sept 2021, the last release of AmberTools is 21. It does not work with amber16 (our last version of Amber). So, we install Amber16 along with AmberTools17

Unpack both Amber16 and AmberTools17 into the same folder Amber16 in the home directory

tar xvvf /tmp/Amber16.tar.bz2

tar xvvf /tmp/AmberTools17.tar.bz2

cd /home/user/amber16

For users that connect to the internet through a proxy server, you set the http\_proxy environment variable yourself in the .bashrc

vi ~/.bashrc

add the following line at the on of the file

export http\_proxy="10.40.1.252:3128"

This will allow patches updates when installing amber

Then install:

> ./configure --with-python “python2.7” gnu

> make install

Next, install the cuda version:

> ./configure -cuda gnu

> make install