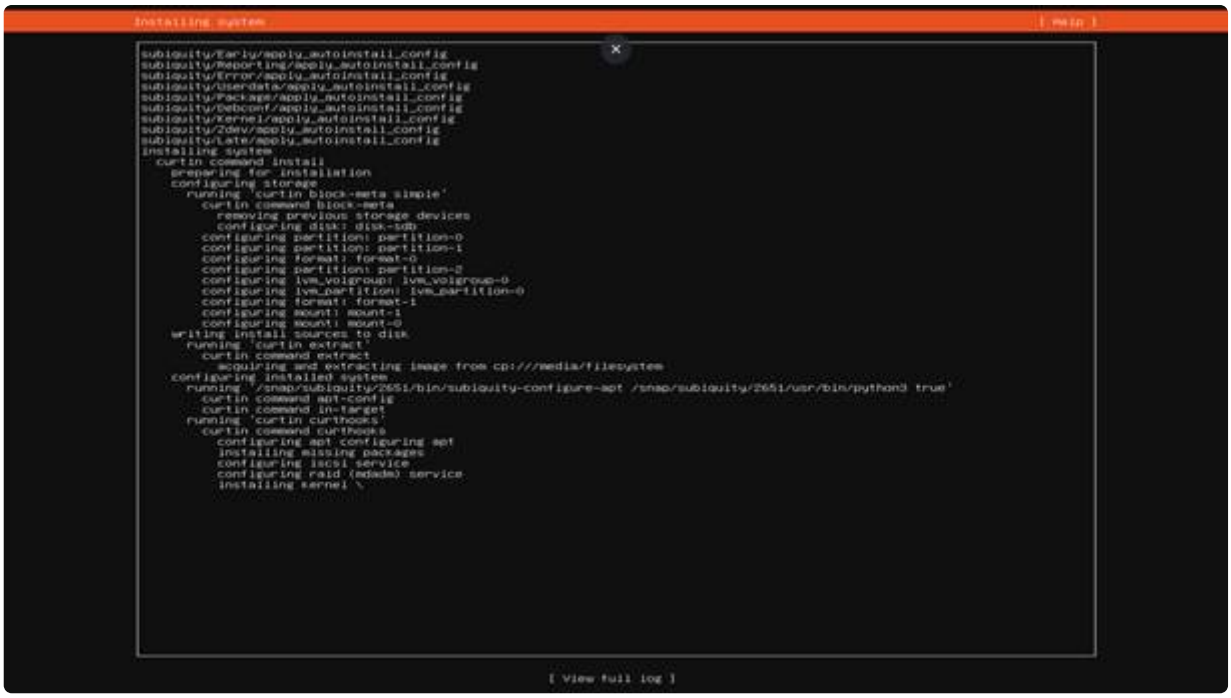
在Ubuntu 20.04中安装Nvidia V100 GPU驱动

导读 **本文介绍如何在Ubuntu 20.04 操作系统中禁用nouveau驱动，并安装Nvidia Tesla V100型号GPU的驱动，并安装CUDA11.6和cuDNN。**

安装Ubuntu 20.04.3 LTS版本

安装Ubuntu 20.04按照安装提示，仔细选择每一项，基本默认即可。



系统中查看GPU信息

系统安装完成之后，进入系统，使用lspci 命令查询一下GPU是否存在、型号信息是什么。

```
bpang@bobpang:~$ sudo lspci |grep -i nvidia
2f:00.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 PCIe 16GB] (rev a1)
86:00.0 3D controller: NVIDIA Corporation GV100GL [Tesla V100 PCIe 16GB] (rev a1)
```

下载NVIDIA Tesla V100驱动

通过lspci查询到GPU的型号之后，访问官网 nvidia.com 下载驱动程序可以按照如下选择，选择产品类型、系列、型号、然后根据自己的操作系统来选择。

注意：如果操作系统是Linux，尽量选择Linux 32-bit/Linux 64-bit，不需要选择详细的Linux发行版本。测试是发现选择详细的Linux发行版本，安装驱动之后，找不到nvidia-smi命令。

安装gcc等依赖包

当安装GPU驱动时，提示缺少相关的依赖包，在此，我们需要提前安装相关的依赖包，目前需要用到的是gcc , g++ , make ：

```
bpang@bobpang:~$ sudo apt install gcc g++ make
```

屏蔽nouveau开源版本的GPU驱动

当系统安装完成之后，会安装系统开源的NVIDIA驱动版本，名称为nouveau。下面将屏蔽该驱动。
首先，创建/etc/modprobe.d/blacklist-nouveau.conf文件，

```
bpang@bobpang:~$ sudo vim /etc/modprobe.d/blacklist-nouveau.conf
```

将下面内容添加进去：

```
blacklist nouveau
blacklist lbm-nouveau
options nouveau modeset=0
alias nouveau off
alias lbm-nouveau off
```

创建/etc/modprobe.d/nouveau-kms.conf文件，将options nouveau mdeset=0添加进去：

```
bpang@bobpang:~$ echo options nouveau modeset=0 | sudo tee -a /etc/modprobe.d/nouveau-kms.conf
```

更新一下initramfs：

```
bpang@bobpang:~$ sudo update-initramfs -u
```

重启服务器：

```
bpang@bobpang:~$ sudo reboot
```

查看nouveau模块是否加载，不显示的话就表示已经禁用：

```
bpang@bobpang:~$ sudo lsmod | grep nouveau
```

```
bpang@bobpang:~$ vim lspci.txt
bpang@bobpang:~$ lsmod |grep -i nouveau
bpang@bobpang:~$
```

安装NVIDIA GPU驱动

安装下载的GPU驱动：NVIDIA-Linux-x86_64-510.47.03.run ，目前驱动版本为：510.47.03，如下执行该驱动文件，即可安装。

```
bpang@bobpang:~$ ./ NVIDIA-Linux-x86_64-510.47.03.run
```

安装完成之后，可以使用lspci看到GPU的驱动信息：

```
        Capabilities: [128 v1] Power Budgeting <?>
        Capabilities: [420 v2] Advanced Error Reporting
                UESta:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq- ACSViol-
                UEMsk:  DLP- SDES- TLP- FCP- CmpltTO- CmpltAbrt- UnxCmplt- RxOF- MalfTLP- ECRC- UnsupReq+ ACSViol-
                UESvrt: DLP+ SDES+ TLP- FCP+ CmpltTO- CmpltAbrt- UnxCmplt- RxOF+ MalfTLP+ ECRC- UnsupReq- ACSViol-
                CESta:  RxErr- BadTLP- BadDLLP- Rollover- Timeout- AdvNonFatalErr-
                CEMsk:  RxErr+ BadTLP+ BadDLLP+ Rollover+ Timeout+ AdvNonFatalErr+
                AERCap: First Error Pointer: 00, ECRCGenCap- ECRCGenEn- ECRCChkCap- ECRCChkEn-
                        MultHdrRecCap- MultHdrRecEn- TLPPfxPres- HdrLogCap-
                HeaderLog: 00000000 00000000 00000000 00000000
        Capabilities: [600 v1] Vendor Specific Information: ID=0001 Rev=1 Len=024 <?>
        Capabilities: [900 v1] Secondary PCI Express
                LnkCtl3: LnkEquIntrruptEn-, PerformEqu-
                LaneErrStat: 0
        Capabilities: [ac0 v1] Designated Vendor-Specific <?>
        Kernel driver in use: nvidia
        Kernel modules: nvidiafb, nouveau, nvidia_drm, nvidia

57:02.0 PCI bridge [0604]: Intel Corporation Sky Lake-E PCI Express Root Port C [8086:2032] (rev 04) (prog-if 00 [Normal de
        Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr+ Stepping- SERR+ FastB2B- DisINTx+
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
        Latency: 0, Cache Line Size: 64 bytes
        Interrupt: pin A routed to IRQ 32
        NUMA node: 0
        Bus: primary=57, secondary=58, subordinate=58, sec-latency=0
        I/O behind bridge: 00008000-00008fff [size=4K]
        Memory behind bridge: e3d00000-e3efffff [size=2M]
        Prefetchable memory behind bridge: 00000000e3b00000-00000000e3cfffff [size=2M]
        Secondary status: 66MHz- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort+ <SERR- <PERR-
        BridgeCtl: Parity+ SERR+ NoISA- VGA- VGA16+ MAbort- >Reset- FastB2B-
                PriDiscTmr- SecDiscTmr- DiscTmrStat- DiscTmrSERREn-
        Capabilities: [40] Subsystem: Lenovo Sky Lake-E PCI Express Root Port C [17aa:7800]
        Capabilities: [60] MSI: Enable+ Count=1/2 Maskable+ 64bit-
                Address: fee00038  Data: 0000
                Masking: 00000002  Pending: 00000000
```

使用nvidia-smi命令查看GPU的信息

```
bpang@bobpang:~$  nvidia-smi
```

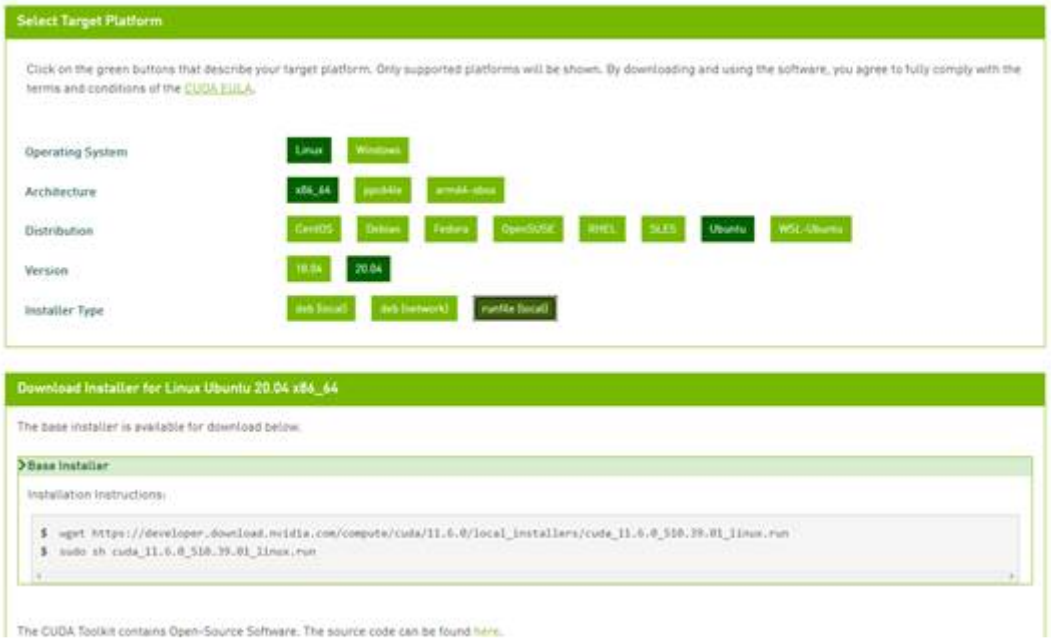
```
bpang@bobpang:/data/cuda$ nvidia-smi
Fri Feb 11 03:31:14 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.47.03    Driver Version: 510.47.03    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  Off  | 00000000:2F:00.0 Off |                    0 |
| N/A   36C    P0    26W / 250W |      4MiB / 16384MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   35C    P0    24W / 250W |      4MiB / 16384MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      2160      G   /usr/lib/xorg/Xorg                  4MiB |
|    1   N/A  N/A      2160      G   /usr/lib/xorg/Xorg                  4MiB |
+-----------------------------------------------------------------------------+
bpang@bobpang:/data/cuda$
```

安装CUDA 11.6

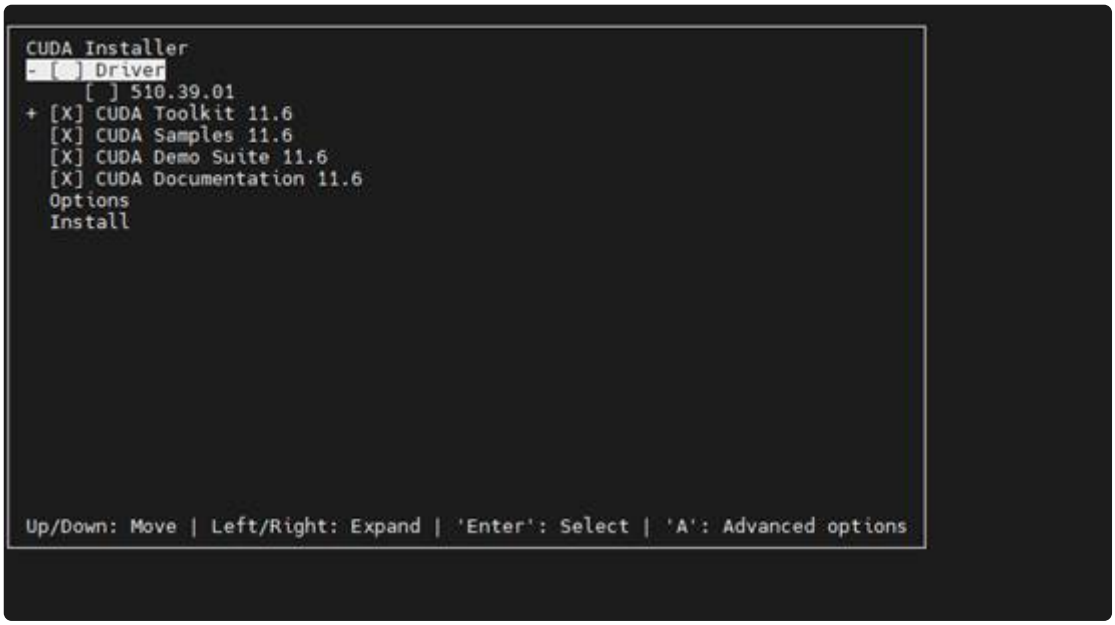
访问nvidia官网，下载CUDA，cuda的链接为：https://developer.nvidia.com/cuda-downloads

选择runfile文件来安装。




```
bpang@bobpang:~$  wget https://developer.download.nvidia.com/compute/cuda/11.6.0/local_installers/cuda_11.6.0_510.39.01_linux.run
bpang@bobpang:~$  sudo sh cuda_11.6.0_510.39.01_linux.run
```

如下图，Driver选项不要勾选了，前面已经安装GPU驱动了。



创建环境变量，编辑 ~/.bashrc 文件：

```
bpang@bobpang:/data/cuda$ vim ~/.bashrc
```

将下面命令追加到文件最后面：

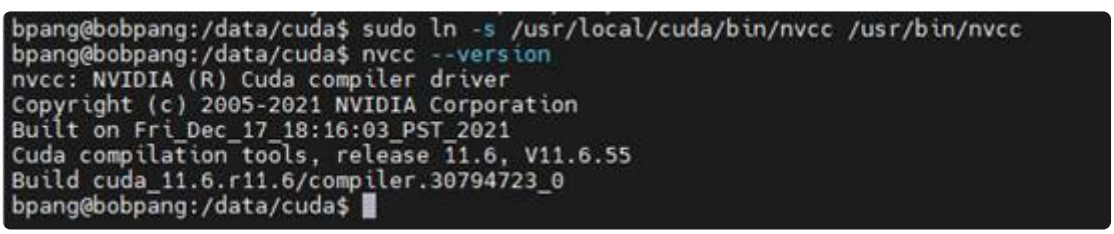
```
export CUDA_HOME=/usr/local/cuda
export PATH=$PATH:$CUDA_HOME/bin
export LD_LIBRARY_PATH=/usr/local/cuda-11.6/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

为 nvcc 命令创建一个软连接到/usr/bin目录：

```
bpang@bobpang:/data/cuda$ sudo ln -s /usr/local/cuda/bin/nvcc /usr/bin/nvcc
```

使用nvcc命令查看cuda的版本：

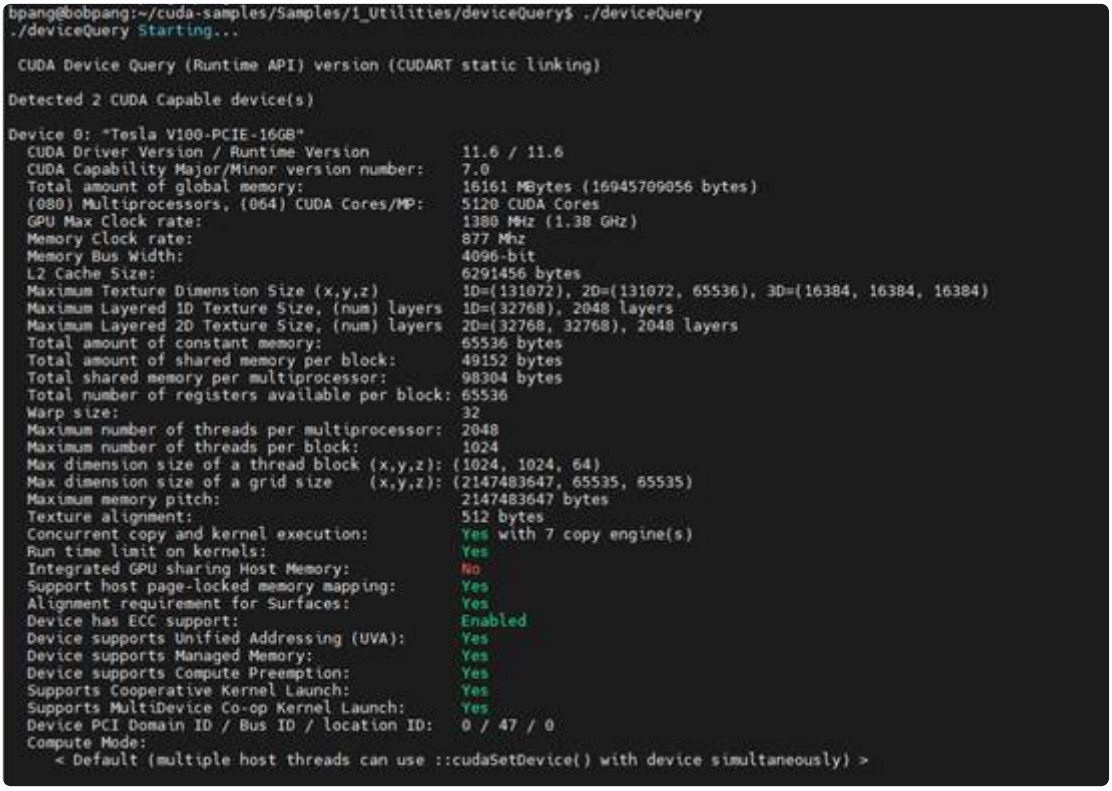
```
bpang@bobpang:/data/cuda$ nvcc --version
```



验证CUDA是否安装成功

下载CUD实例文件。因为CUDA 11.6版本中 /usr/local/cuda/samples里面没有实例文件，只有一个README文件，内容中告诉你需要从github下载，因为github太慢，所以从gitee中下载实例文件：

```
bpang@bobpang:~$ git clone https://gitee.com/liwuhaoooo/cuda-samples.git
bpang@bobpang:~$ mv cuda-samples/Samples/* /usr/local/cuda/samples/*
bpang@bobpang:~$ cd /usr/local/cuda/samples/1_Utilities/deviceQuery
$ sudo make
$ ./deviceQuery
```

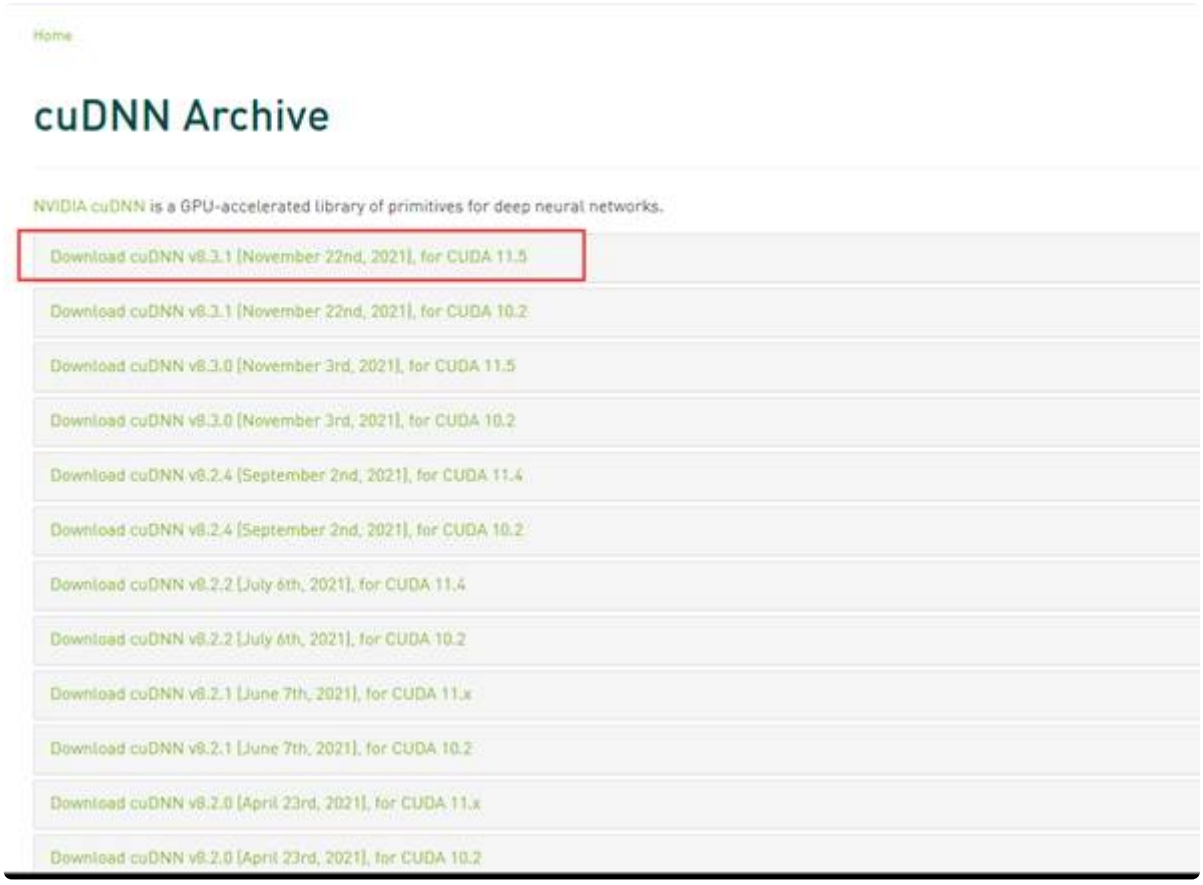


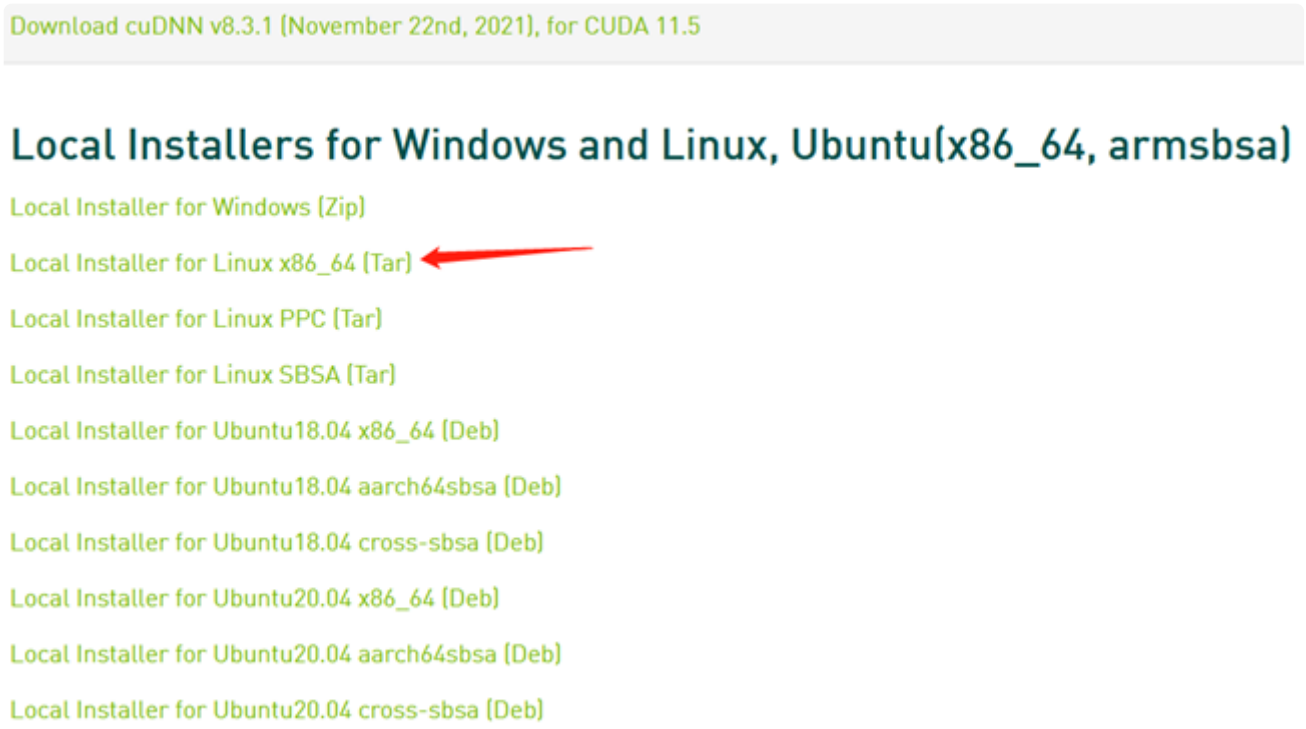
```
Device 1: "Tesla V100-PCIE-16GB"
  CUDA Driver Version / Runtime Version          11.6 / 11.6
  CUDA Capability Major/Minor version number:    7.0
  Total amount of global memory:                 16161 MBytes (16945709056 bytes)
  (080) Multiprocessors, (064) CUDA Cores/MP:    5120 CUDA Cores
  GPU Max Clock rate:                            1380 MHz (1.38 GHz)
  Memory Clock rate:                             877 Mhz
  Memory Bus Width:                              4096-bit
  L2 Cache Size:                                 6291456 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total shared memory per multiprocessor:        98304 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 7 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Enabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Managed Memory:                Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 134 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
> Peer access from Tesla V100-PCIE-16GB (GPU0) -> Tesla V100-PCIE-16GB (GPU1) : Yes
> Peer access from Tesla V100-PCIE-16GB (GPU1) -> Tesla V100-PCIE-16GB (GPU0) : Yes

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.6, CUDA Runtime Version = 11.6, NumDevs = 2
Result = PASS
bpang@bobpang:~/cuda-samples/Samples/1_Utilities/deviceQuery$
```

安装cuDNN

https://developer.nvidia.com/cudnn 从官网下载cudnn





下载的文件名为：cudnn-linux-x86_64-8.3.2.44_cuda11.5-archive.tar.xz

文件拷贝到服务器。然后安装：

```
bpang@bobpang:/data$ tar xvf cudnn-linux-x86_64-8.3.2.44_cuda11.5-archive.tar.xz
bpang@bobpang:~$ cd /usr/local/cuda
bpang@bobpang:/usr/local/cuda$ sudo cp -p /data/cudnn-linux-x86_64-8.3.2.44_cuda11.5-archive/in
clude/cudnn*.h include/
bpang@bobpang:/usr/local/cuda$ sudo cp -p /data/cudnn-linux-x86_64-8.3.2.44_cuda11.5-archive/li
b/libcudnn* lib64/
bpang@bobpang:~$ sudo chmod a+r /usr/local/cuda-11.6/include/cudnn.h
bpang@bobpang:~$ sudo chmod a+r /usr/local/cuda-11.6/lib64/libcudnn*
```

本文原创地址：https://www.linuxprobe.com/ubuntu-nvidia-v100-gpu.html 编辑：逄增宝，审核员：逄增宝

为您推荐一些与本文相关的文章：

- Linux 6.2合并华为代码，为核心内核的性能带来大幅提升
- Rudra Saraswat近日宣布blendOS 3全面上市
- 代码托管平台：GitLab 8.3.1正式发布下载
- 《Python源码剖析》pdf电子书免费下载
- 捷讯：朱炫录6月27日深圳顺利通过RHCE认证。
- docker之删除none镜像
- 《iPhone狂：约会iPhone》pdf电子书免费下载
- 在Deepin操作系统中开启或关闭自动登录系统
- 在苹果 M1 上运行 Linux 虚拟机变得容易了
- 移动版 GNOME Shell：希望之始，期望满满