1. 环境准备
2. 操作系统版本：官网版本只支持linux内核操作系统，推荐使用ubuntu 16.04/18.04 LTS
3. 硬件配置需求：

|  |  |
| --- | --- |
| **配置项** | **推荐使用** |
| CPU | Intel 4核心 主频不低于2.5GHz |
| 内存 | 推荐使用16G Byte |
| 显卡 | Nvidia Geforce系列显卡，显卡内存4G Byte |

1. 显卡驱动安装&&开发者依赖库安装

文档以Nvidia Geforce GTX 1050Ti系列显卡为例（操作系统版本ubuntu16.04 LTS），安装驱动和所需依赖库，其他类型的显卡请开发者选择对应版本进行安装

1. 官网下载显卡对应版本驱动，如下图所示：



* 删除旧的驱动

sudo apt-get autoremove --purge nvidia-\*

* Linux kernel header安装

sudo apt-get install linux-headers-$(uname -r)

* 禁用旧的nouveau驱动

$ cd /etc/modprobe.d/

$ sudo vim blacklist-nouveau.conf

blacklist nouveau

options nouveau modeset=0

$ sudo update-initramfs –u

* 重启机器。执行如下命令，如果没有信息输出，则配置完成

$ lsmod | grep nouveau

* 关闭X图形界面服务

sudo service lightdm stop

* 进入驱动下载目录，安装驱动

$ sudo chmod a+x NVIDIA-Linux-x86\_64-xxx.run

$ sudo ./NVIDIA-Linux-x86\_64-xxx.run --no-opengl-files

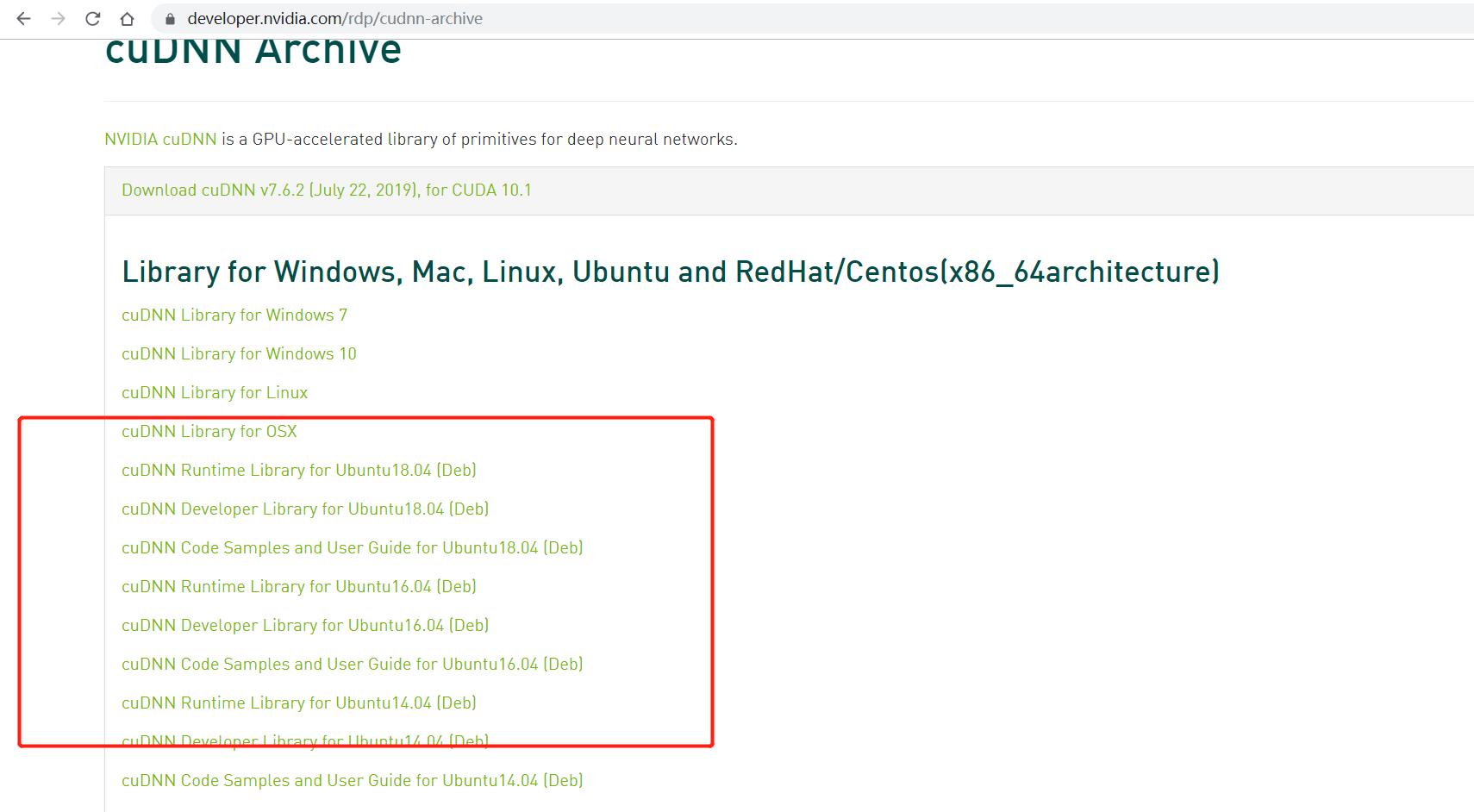
1. 安装CUDA

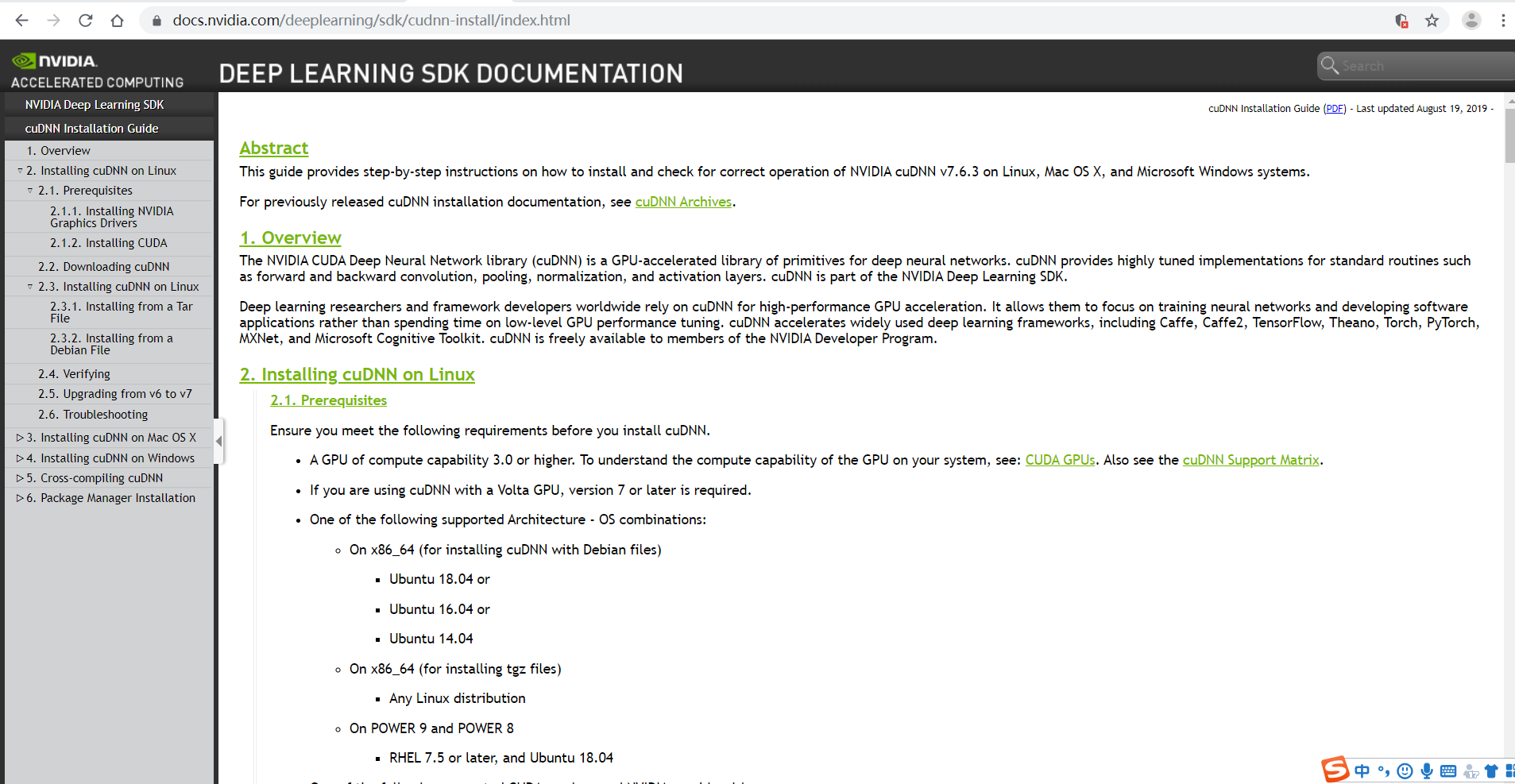
进入官网CUDA下载地址（<https://developer.nvidia.com/cuda-toolkit-archive>），下图以10.1版本（CUDA>=8.0）为例，按照安装向导完成安装



1. 安装CuDNN

进入官网CUDA下载地址（<https://developer.nvidia.com/rdp/cudnn-archive），需要注册Nvidia账户进行下载，下图以10.1>为例，按照安装向导完成安装





1. 编译依赖安装

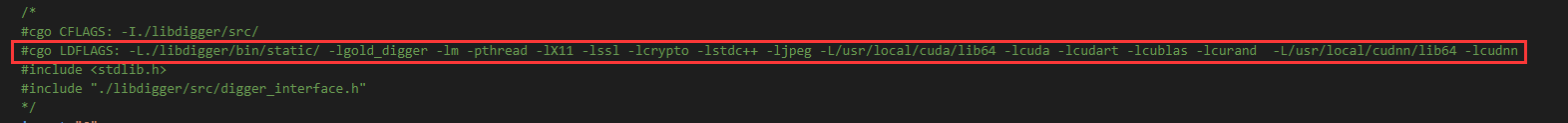
（1）Go：1.7版本或者更新

（2）gcc或其它C编译器

（3）安装libx11、openssl、libjpeg依赖库

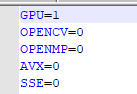
* + - sudo apt-get install libx11-dev
    - sudo apt-get install openssl
    - sudo apt-get install libssl-dev
    - sudo apt-get install libjpeg-turbo8-dev

1. GPU挖矿lib库编译
2. Git clone <https://github.com/MatrixAINetwork/go-matrix.git>
3. 进入/github.com/MatrixAINetwork/go-matrix/aidigger目录，确认digger.go文件内容正确



注意：/usr/local/cuda/lib64 为步骤2中显卡驱动指令安装位置，请确认是否一致

1. 进入/github.com/MatrixAINetwork/go-matrix/aidigger/libdigger目录，确认makefile文件内容正确



1. 上述目录下执行make clean&&make，完成GPU挖矿lib库编译
2. gman编译

（1）参考官网手册编译gman，<https://github.com/MatrixAINetwork/go-matrix>

（2）GPU挖矿lib库静态链接，如果在编译主机本地启动gman程序，可忽略以下步骤。

（3）静态链接依赖工具安装

* sudo wget <https://github.com/greenpau/statifier>
* in configs/config.x86\_64

change ELF32 := yes

to ELF32 := no

* make
* make install
* disable ASLR in Linux (https://askubuntu.com/questions/318315/how-can-i-temporarily-disable-aslr-address-space-layout-randomization)

echo 0 | sudo tee /proc/sys/kernel/randomize\_va\_space

* This won't survive a reboot, so you'll have to configure this in sysctl. Add a file /etc/sysctl.d/01-disable-aslr.conf containing:

kernel.randomize\_va\_space = 0

（4）进入gman生成目录github.com/MatrixAINetwork/go-matrix/build/bin，执行statifier gman new\_gman（名字用户可自定义），拷贝new\_gman到运行主机启动

1. 部署gman抵押节点

（1）将下列文件部署到对应的gman节点的对应位置（直接全路径位置显示）。

/gman #文本文件

/MANGenesis.json #文本文件

/chaindata #目录文件

/chaindata/man.json #文本文件

/chaindata/coco.names #文本文件

/chaindata/yolov3.cfg #文本文件

/chaindata/yolov3.weights #文本文件

/chaindata/picstore #目录文件

/chaindata/picstore/test\_0.jpg #图片文件

……

/chaindata/picstore/test\_15.jpg #图片文件

/root #目录文件

（2）初始化创始区块，执行命令如下：

/gman --datadir /chaindata/ init/MANGenesis.json

（3）在钱包端创建keystore、账户地址和密码password（切记密码）,将keystore的内容放到第2步生成的keystore文件下。keystore存放到路径：

/chaindata/keystore

（4）密码明文加密成密文：

首先在/root路径下新建signAccount.json文件，新建文件内容如下：

[

{

"Address":"MAN.gQAAHUeTBxvgbzf8tFgUtavDceJP",

"Password":"pass123456"

}

]

其次，生成密文，在/root下执行命令如下：

/gman --datadir /chaindata aes --aesin ./signAccount.json --aesout entrust.json

执行该命令时，会让输入密码，密码要求必须有大小写、数字、特殊字符。

将生成的entrust.json文件部署到根目录下（“/”）。

（5）启动gman(启动时的命令已集成到gman\_run.sh如下：)的命令，启动时需要输入的密码就是第5步骤设置的密码：

/gman --datadir /chaindata --networkid 1 --debug --verbosity 5 -- manAddress [your man.address here] --entrust /entrust.json --outputinfo 1 --syncmode full

（6）执行Attach命令: /gman attach /chaindata/gman.ipc

现在可以通过一些命令来查看信息了

例如，man.blockNumber 查看一下当前同步的块高