1. Environment preparation
2. The version of operating System: The official version only supports the Linux kernel operating system, recommending the use of ubuntu 16.04/18.04 LTS
3. Hardware configuration requirements:

|  |  |
| --- | --- |
| **Configuration Items** | **Recommend to Use** |
| CPU | Intel 4 Core, main frequency no less than 2.5GHz |
| RAM | Recommend to use 16G Byte |
| Graphics card | Nvidia Geforce series graphics card，with the RAM of 4G Byte |

1. Graphics Card Driver Installation & Developer Dependent Library Installation

The document takes Nvidia Geforce GTX 1050Ti series graphics cards as an example (operating system version is Ubuntu 16.04 LTS), installing drivers and required dependent libraries, and other types of graphics cards require developers to choose the corresponding version for installation.

1. Download the corresponding version driver of the graphics card on the official website, as shown in the following figure:



* Delete old driver

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

* Install Linux kernel header

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

* Disable old nouveau driver

$ cd /etc/modprobe.d/

$ sudo vim blacklist-nouveau.conf

blacklist nouveau

options nouveau modeset=0

$ sudo update-initramfs –u

* Restart the computer. Implement the following commands, and the configuration complete when there’s no information output.

$ lsmod | grep nouveau

* Close X graphical interface service

sudo service lightdm stop

* Enter the downloading directory of driver, and install the driver

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

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

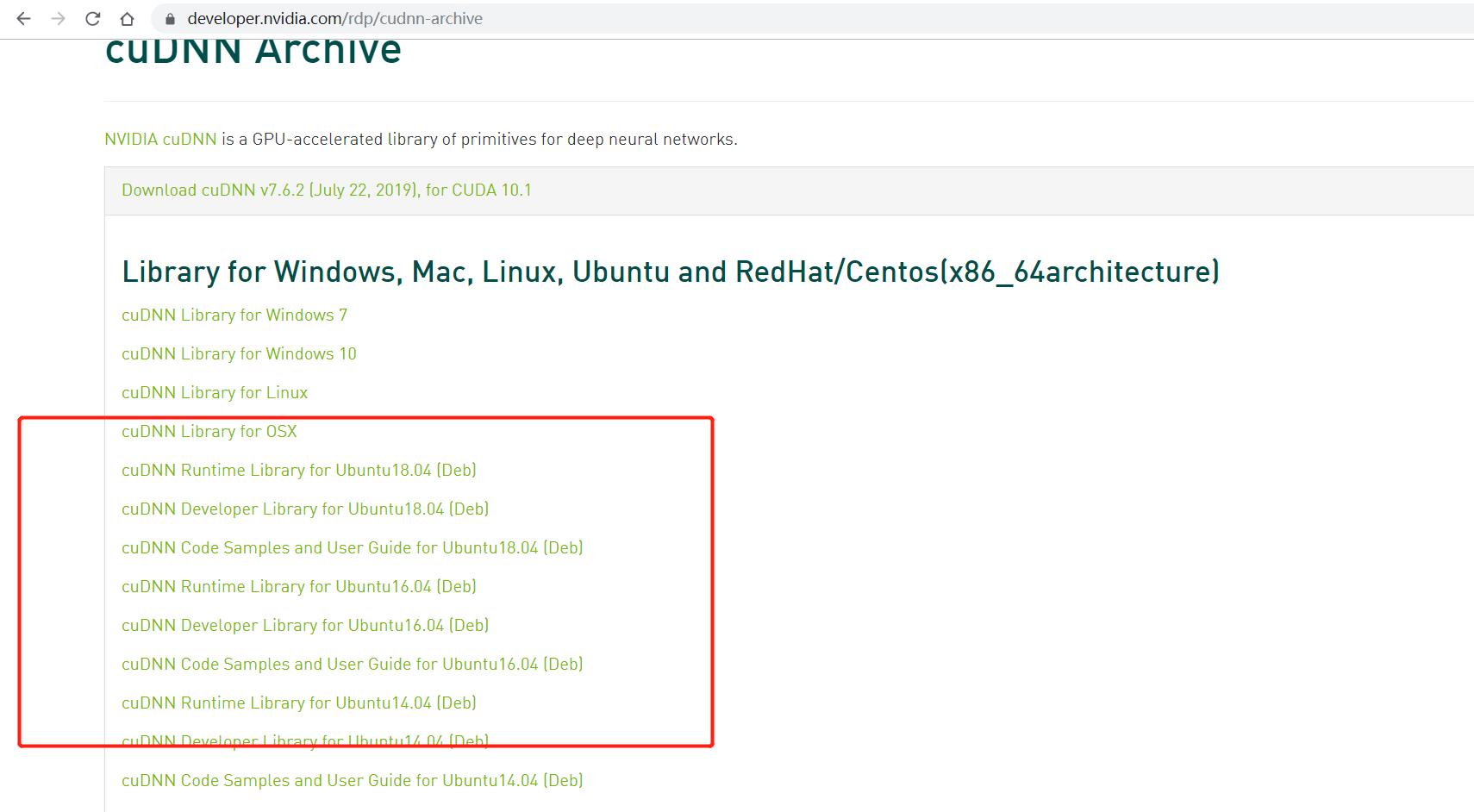
1. Install CUDA

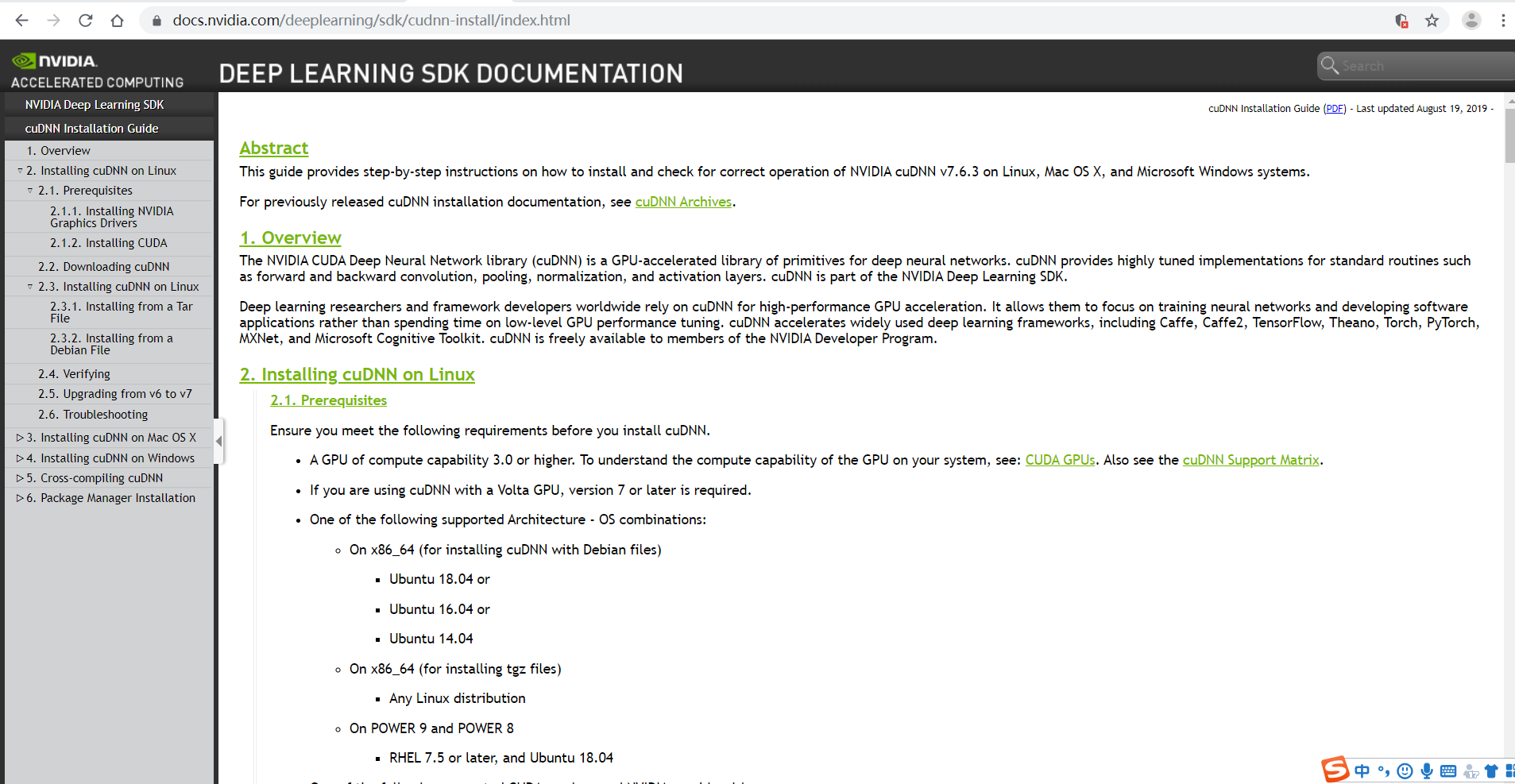
Go to the CUDA download address on the official website (https://developer.nvidia.com/cuda-toolkit-archive). The following figure takes the 10.1 version (CUDA>=8.0) as an example. Follow the installation wizard to complete the installation.



1. Install CuDNN

Go to the CUDA download address on the official website (https://developer.nvidia.com/rdp/cudnn-archive), and you need to register for Nvidia account to download. The following figure takes 10.1 version as an example, follow the installation wizard to complete the installation.





1. Compile dependent library installation

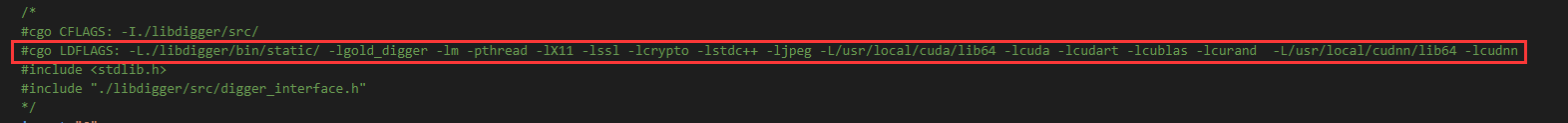
（1）Go：1.7 version or the updated version

（2）gcc or other C compiler

（3）Install libx11、openssl、libjpeg dependent library

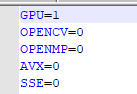
* + - sudo apt-get install libx11-dev
    - sudo apt-get install openssl
    - sudo apt-get install libssl-dev
    - sudo apt-get install libjpeg-turbo8-dev

1. GPU mining lib library compilation
2. Git clone <https://github.com/MatrixAINetwork/go-matrix.git>
3. Enter /github.com/MatrixAINetwork/go-matrix/aidigger directory，and confirm the correctness of digger.go content



Note：/usr/local/cuda/lib64 is the installation location of the graphics driver command in step 2. Please confirm whether it is consistent.

1. Enter /github.com/MatrixAINetwork/go-matrix/aidigger/libdigger directory, and confirm the correctness of makefile content.



1. Implement make clean&&make in the above directory, and complete GPU mining lib library compilation.
2. Gman compilation

（1）Compile gman with reference to the official website manual, <https://github.com/MatrixAINetwork/go-matrix>

（2）GPU mining lib library static link, if you start the gman program locally in the compiled host, you can ignore the following steps.

（3）Static link dependent tool installation

* 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）Go to gman to generate directory github.com/MatrixAINetwork/go-matrix/build/bin, execute statifier gman new\_gman (name can be customized by user), and copy new\_gman to run host to launch

1. Deploy gman stake Masternodes

（1）Deploy the following files to the corresponding location of the corresponding gman node (shown directly at the full path location).

/gman # Text file

/MANGenesis.json # Text file

/chaindata # Directory file

/chaindata/man.json # Text file

/chaindata/coco.names # Text file

/chaindata/yolov3.cfg # Text file

/chaindata/yolov3.weights # Text file

/chaindata/picstore # Directory file

/chaindata/picstore/test\_0.jpg # Picture file

……

/chaindata/picstore/test\_15.jpg # Picture file

/root # Directory file

（2）Initialize the Genesis block，and execute the command as follows:

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

（3）Create a keystore, account address, and password on the wallet (remember password), and place the contents of the keystore under the keystore file generated in step 2. The keystore is stored in the path:

/chaindata/keystore

（4）Encrypted the password in plain text into cipher text:

First create a new signAccount.json file in the /root path. The contents of the new file are as follows:

[

{

"Address":"MAN.gQAAHUeTBxvgbzf8tFgUtavDceJP",

"Password":"pass123456"

}

]

Second, generate cipher text, execute the command under /root as follows:

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

When you execute this command, you will be asked to enter a password. The password must have capitalization, numbers, and special characters.

Deploy the generated entrust.json file to the root directory ("/").

（5）Launch gman (the launched command is integrated into gman\_run.sh as follows:), the password that needs to be entered when launch is the password set in step 5:

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

（6）Implement Attach command: /gman attach /chaindata/gman.ipc

Now you can view the information with some commands.

For example, enter man.blockNumber to view the block height of the current sync.