DeepStream 5.0 Installation Guide V1.1

Mehul Vast

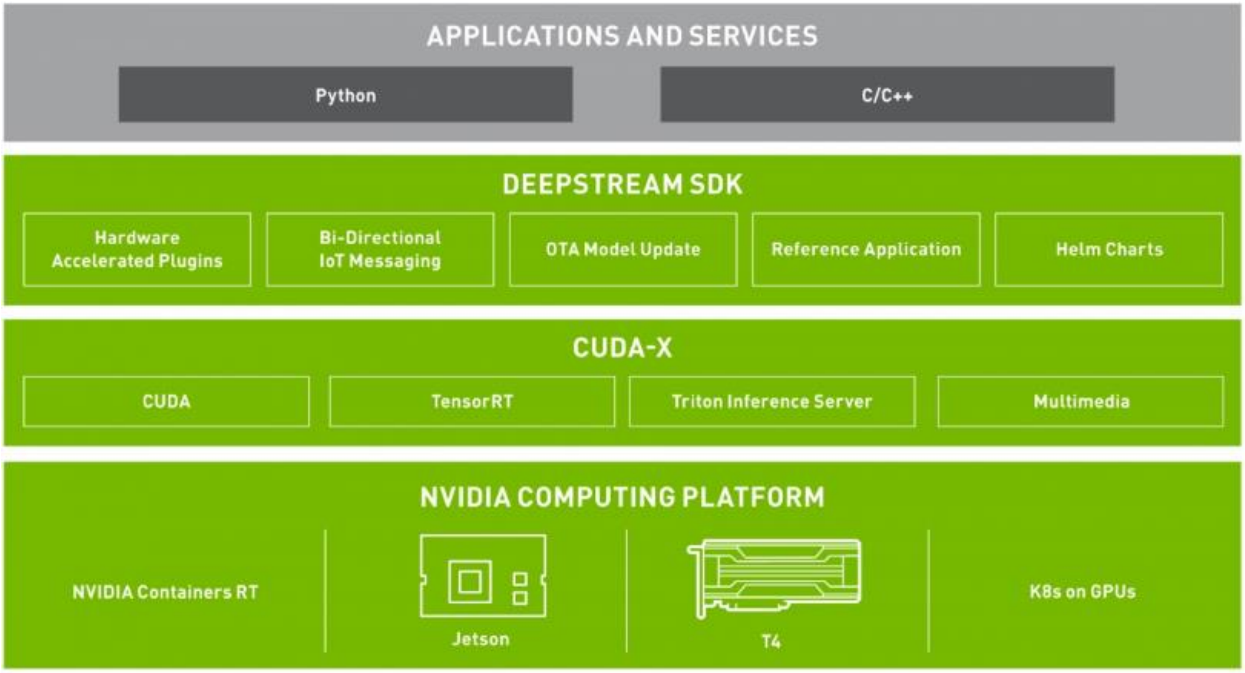
Table of Contents

1. DEEPSTREAM OVERVIEW 4
2. DEEPSTREAM SDK BUILDING BLOCKS 5
3. SETUP & INSTALLATION 5
4. DGPU MODEL PLATFORM AND OS COMPATIBILITY 12

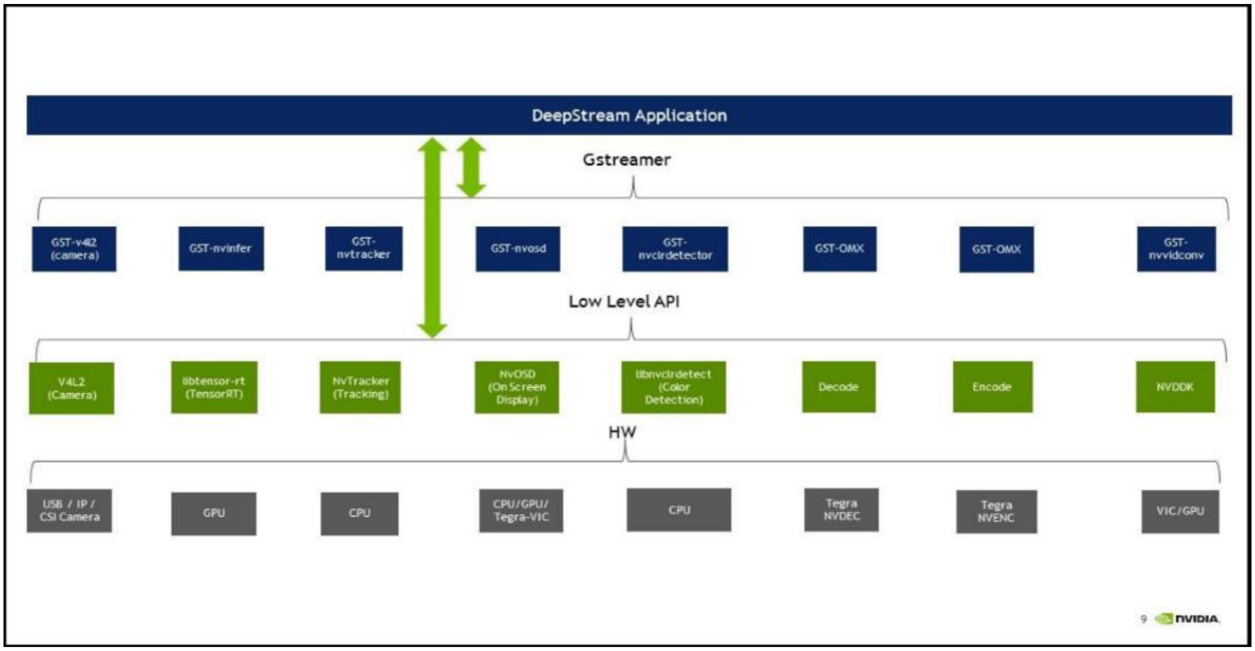
DEEPSTREAM OVERVIEW

There are billions of cameras and sensors worldwide, capturing an abundance of data that can be used to generate business insights, unlock process efficiencies and improve revenue streams. Whether it's at a traffic intersection to reduce vehicle congestion, health and safety monitoring at hospitals, surveying retail aisles for better customer satisfaction, sports analytics or at a manufacturing facility to detect component defects- every application demands reliable, real-time Intelligent Video Analytics (IVA).

With DeepStream SDK you can apply AI to streaming video and can simultaneously optimize video decode/encode, image scaling and conversion and edge-to-cloud connectivity for complete end-to-end performance optimization.



DEEPSTREAM SDK BUILDING BLOCKS



SETUP & INSTALLATION

NVIDIA® DeepStream Software Development Kit (SDK) is an accelerated AI framework to build intelligent video analytics (IVA) pipelines. DeepStream runs on NVIDIA® T4 and platforms such as NVIDIA® Jetson™ Nano, NVIDIA® Jetson AGX Xavier™, NVIDIA® Jetson Xavier NX™, NVIDIA® Jetson™ TX1 and TX2.

Make sure you have a Linux platform,Deepstream does not support Windows.

You must install the following components in your Ubuntu Platform(18/20): -

- GCC-7.5
- PYTHON 3.7
- GStreamer 1.14.1
- NVIDIA driver 450.51
- CUDA 10.2
- CUDNN 7.6.5
- TensorRT 7.0.X

1. Remove all previous DeepStream installations

Enter the following commands to remove all previous DeepStream 3.0 or prior installations:

```
$ sudo rm -rf /usr/local/deepstream /usr/lib/x86_64-linux-gnu/gstreamer-1.0/libgstnv*      /usr/bin/deepstream* /usr/lib/x86_64-
linux-gnu/gstreamer-1.0/libnvdsgst*
/usr/lib/x86_64-linux-gnu/gstreamer-1.0/deepstream*
/opt/nvidia/deepstream/deepstream*
$ sudo rm -rf /usr/lib/x86_64-linux-gnu/libv41/plugins/libcuvidv4l2_plugin.so
```

2. Uninstall Nvidia Drivers and CUDA

```
sudo apt-get --purge remove "*cublas*" "*cufft*" "*curand*" "*cusolver*" "*cusparse*" "*npp*"   "*nvjpeg*" "cuda*"
"nsight*"

sudo apt-get --purge remove "*nvidia*"

sudo apt-get autoremove
```

Enabling Nouveau driver (default graphic driver) for Ubuntu

```
echo ‘nouveau’ | sudo tee -a /etc/modules

dpkg -l | grep cuda

dpkg -l | grep nvidia
```

Reboot your System

```
sudo reboot
```

3. Install GCC and G++

```
sudo apt install build-essential

sudo apt -y install gcc-7 g++-7 gcc-8 g++-8 gcc-9 g++-9

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 7

sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-7 7

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 8
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-8 8
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 9
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 9

sudo update-alternatives --config gcc

sudo update-alternatives --config g++
gcc --version
```

4. Install Python 3.7

```
sudo apt-get install python3.7

sudo apt-get install python3-pip

sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 1
```

5. Install Gstreamer

```
sudo apt install \ libssl1.0.0 \
libgstreamer1.0-0 \
gstreamer1.0-tools \
gstreamer1.0-plugins-good \
gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-ugly \
gstreamer1.0-libav \
libgstrtspserver-1.0-0 \
libjansson4
```

While installing GSTREAMER replace Libssl1.0.0 with Libssl1.1 as Lib1.0 is not supported in Ubuntu 20.x

6. Install Cuda 10.2

Download CUDA Toolkit 10.2 + NVIDIA Graphic driver 440.33 from NVIDIA Downloads Page.

```
wget
http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda_10.2.89_440.33.01_linux.run
```

```
sudo apt update
sudo apt upgrade
```

Disable noveau driver (avoiding graphic driver installation error). Create new file with nano editor on /etc/modprobe.d/blacklist-nouveau.conf.

```
sudo nano /etc/modprobe.d/blacklist-nouveau.conf
```

Enter following content into it

```
blacklist nouveau
options nouveau modeset=0
```

```
sudo update-initramfs -u
```

Now, reboot and go to downloads and install Cuda

```
cd ~/Downloads
sudo sh cuda_10.2.89_440.33.01_linux.run
```

Got some error in this step ? check the installation log

```
cat /var/log/cuda-installer.log
```

If the error is like this : [ERROR]: Install of driver component failed.
open nvidia-installer log

```
/var/log/nvidia-installer.log
```

If you found error on nvidia-installer.log like this ‘Kernel module load error: Operation not permitted’. It means your machine is secured with secure boot. The solution for this problem you can goto this link https://askubuntu.com/questions/1048135/how-can-i-install-nvidia-drivers-on-ubuntu-18-04-with-secure-boot

After installation, usually your machine will reboot and when you first boot after installation, you will asked for entering MOK Key. This key is required for NVIDIA Graphic driver loading in secure boot and run in your startup. Attention, don’t miss it.

check if NVIDIA Driver have installed with command:

```
nvidia-smi
```

After that, please try to run the cuda installation run file again. uncheck NVIDIA Driver because you’ve already have one.

After instalation done edit the ~/.bashrc.

```
sudo gedit ~/.bashrc
```

Add the following to the end of file:

```
export PATH=/usr/local/cuda/bin:$PATH

export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

Run the ~/.bashrc again.

```
source ~/.bashrc
```

Verify CUDA Toolkit installation with

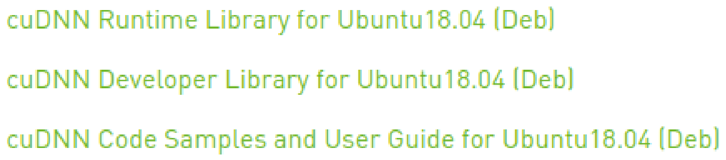
```
nvcc -V
```

7. Install CUDNN v8.3

Go to NVIDIA website and Login NVIDIA Developer Account

Goto https://developer.nvidia.com/rdp/cudnn-download

Choose installer Download cuDNN v7.6.5 (November 18th, 2019), for CUDA 10.2



cuDNN Runtime Library for Ubuntu18.04 (Deb)

cuDNN Developer Library for Ubuntu18.04 (Deb)

cuDNN Code Samples and User Guide for Ubuntu18.04 (Deb)

Download this 3 files :

```
sudo dpkg -i libcudnn7_7.6.5.32–1+cuda10.2_amd64.deb

sudo dpkg -i libcudnn7-dev_7.6.5.32–1+cuda10.2_amd64.deb

sudo dpkg -i libcudnn7-doc_7.6.5.32–1+cuda10.2_amd64.deb

cp -r /usr/src/cudnn_samples_v7/ ~

cd ~/cudnn_samples_v7/mnistCUDNN

make clean && make
```

If some issues follow the link: https://forums.developer.nvidia.com/t/cudnn-test-did-not-pass/54675

./mnistCUDNN

8. Install Tensor RT
    1. Go to NVIDIA website and Login NVIDIA Developer Account
    2. Goto https://docs.nvidia.com/deeplearning/tensorrt/install-guide/index.html
    3. Follow 4.5 section Tar file installation steps
    4. Check if TensorRT successfully installed.

```
dpkg -l | grep TensorRT
```

9. Install librdkafka

    Clone the librdkafka repository from GitHub: git clone https://github.com/edenhill/librdkafka.git

    Configure and build the library:

```
$ cd librdkafka
$ git reset --hard 7101c2310341ab3f4675fc565f64f0967e135a6a
./configure
$ make
$ sudo make install
```

Copy the generated libraries to the deepstream directory:

```
$ sudo mkdir -p /opt/nvidia/deepstream/deepstream-5.0/lib
$ sudo cp /usr/local/lib/librdkafka* /opt/nvidia/deepstream/deepstream-5.0/lib
```

10. Install Deepstream SDK

    Download the DeepStream tar package: https://developer.nvidia.com/assets/Deepstream/5.0/ga/secure/deepstream_sdk_5.0.1_x86_64.tbz2

Navigate to the location of the downloaded DeepStream package to extract and install the DeepStream SDK:

```
$ sudo tar -xvf deepstream_sdk_v5.0.1_x86_64.tbz2 -C /
$ cd /opt/nvidia/deepstream/deepstream-5.0/
$ sudo ./install.sh
$ sudo ldconfig
```

11. Run an Application

```
deepstream-app -c /opt/nvidia/deepstream/deepstream-5.0/samples/configs/deepstream-app/source4_1080p_dec_infer-resnet_tracker_sgie_tiled_display_int8.txt
```

watch nvidia-smi – This is a top-like utility for monitoring CUDA activity on a GPU.

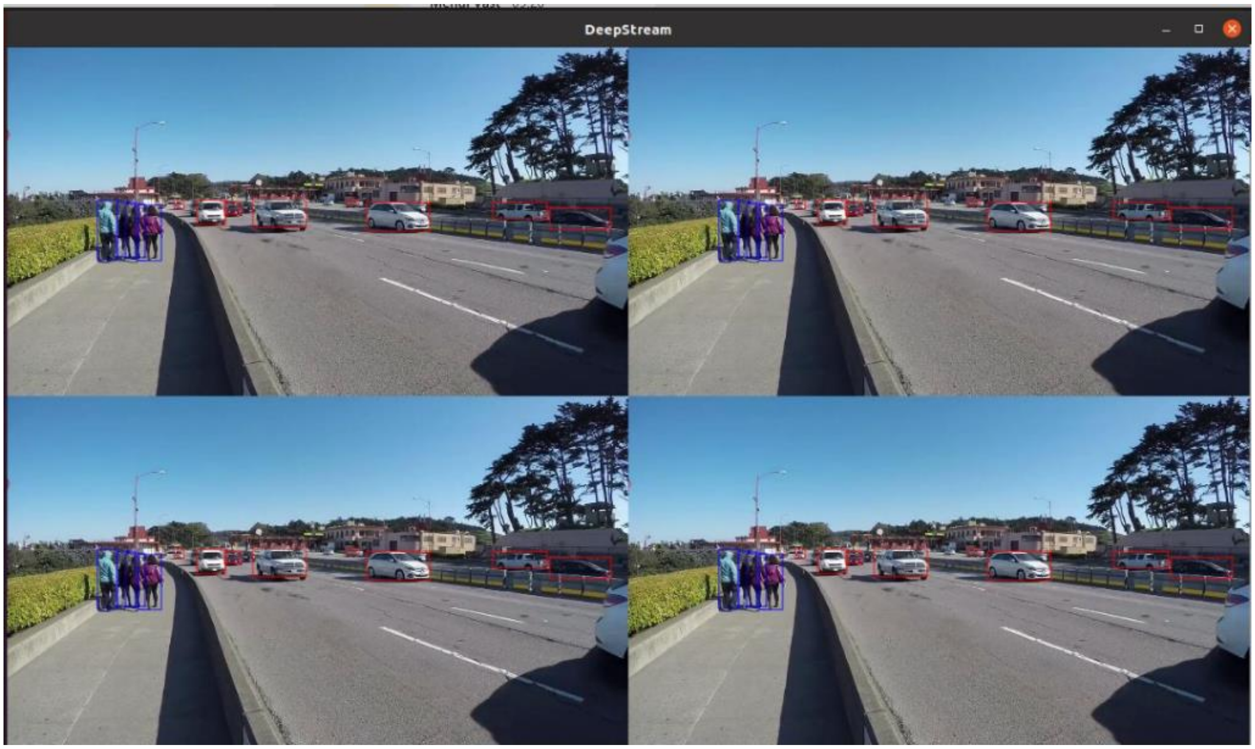
Need to Install opt/nvidia/deepstream/deepstream-5.0/sources/apps/sample_apps/deepstream-app/ README plugins to run a deepstream application:

```
sudo apt-get install libgstreamer-plugins-base1.0-dev
```

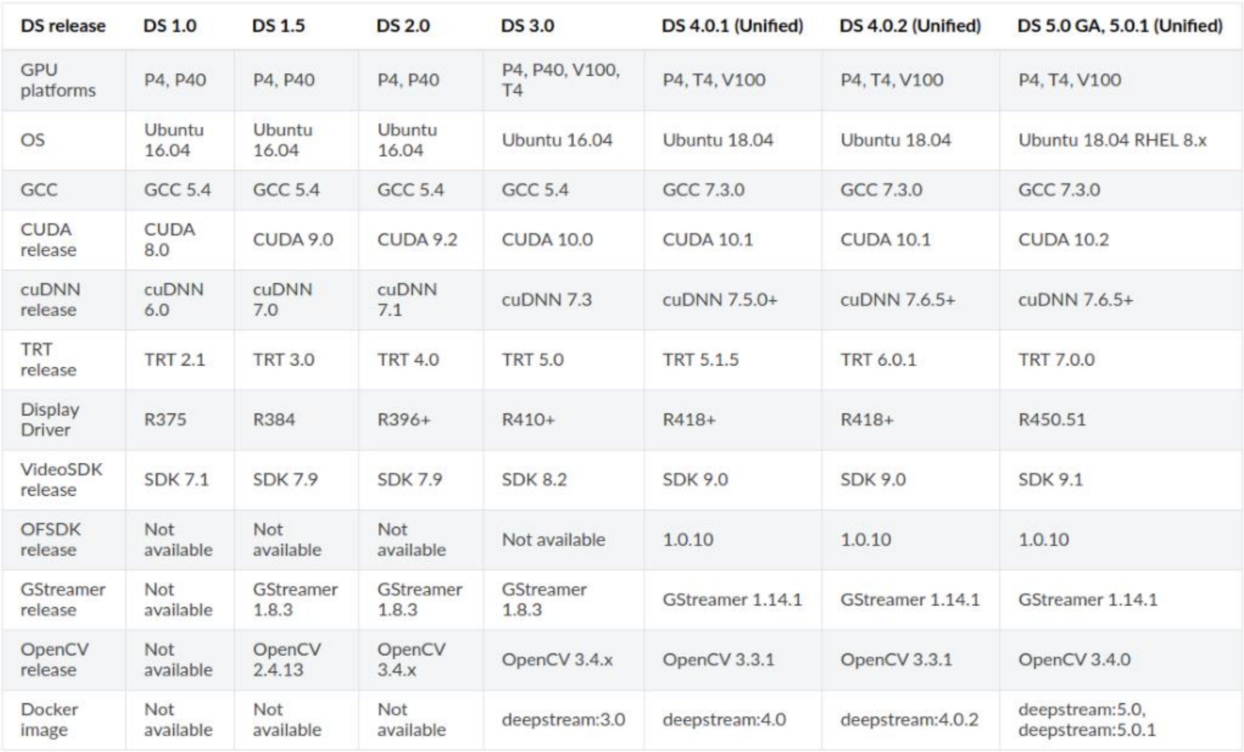
To run an NSIGHT ECLIPSE SDK:

```
./nsight -vm /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Output



DGPU MODEL PLATFORM AND OS COMPATIBILITY



DS release	DS 1.0	DS 1.5	DS 2.0	DS 3.0	DS 4.0.1 (Unified)	DS 4.0.2 (Unified)	DS 5.0 GA, 5.0.1 (Unified)
GPU platforms	P4, P40	P4, P40	P4, P40	P4, P40, V100, T4	P4, T4, V100	P4, T4, V100	P4, T4, V100
OS	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04 RHEL 8.x
GCC	GCC 5.4	GCC 5.4	GCC 5.4	GCC 5.4	GCC 7.3.0	GCC 7.3.0	GCC 7.3.0
CUDA release	CUDA 8.0	CUDA 9.0	CUDA 9.2	CUDA 10.0	CUDA 10.1	CUDA 10.1	CUDA 10.2
cuDNN release	cuDNN 6.0	cuDNN 7.0	cuDNN 7.1	cuDNN 7.3	cuDNN 7.5.0+	cuDNN 7.6.5+	cuDNN 7.6.5+
TRT release	TRT 2.1	TRT 3.0	TRT 4.0	TRT 5.0	TRT 5.1.5	TRT 6.0.1	TRT 7.0.0
Display Driver	R375	R384	R396+	R410+	R418+	R418+	R450.51
VideoSDK release	SDK 7.1	SDK 7.9	SDK 7.9	SDK 8.2	SDK 9.0	SDK 9.0	SDK 9.1
OFSDK release	Not available	Not available	Not available	Not available	1.0.10	1.0.10	1.0.10
GStreamer release	Not available	GStreamer 1.8.3	GStreamer 1.8.3	GStreamer 1.8.3	GStreamer 1.14.1	GStreamer 1.14.1	GStreamer 1.14.1
OpenCV release	Not available	OpenCV 2.4.13	OpenCV 3.4.x	OpenCV 3.4.x	OpenCV 3.3.1	OpenCV 3.3.1	OpenCV 3.4.0
Docker image	Not available	Not available	Not available	deepstream:3.0	deepstream:4.0	deepstream:4.0.2	deepstream:5.0, deepstream:5.0.1

What to do if Ubuntu OS gets corrupted for whatever reason:

1 Recommended to install “timeshift” application and save or back-up the OS at regular intervals of time so that the OS can be restored to any previous saved time instance. (https://linuxconfig.org/ubuntu-20-04-system-backup-and-restore)

2. The following links can also be helpful to check if OS can be fixed without reinstalling it.

https://ostechnix.com/how-to-fix-broken-ubuntu-os-without-reinstalling-it/
https://www.computersnyou.com/4981/how-to-reinstall-ubuntu-without-losing-data/

One Drive in Ubuntu 20.04

Install libcurl as shown below:

```
sudo apt-get install libcurl4-openssl-dev
```

Also make sure that you have installed pkg-config. If unsure install it with:

```
sudo apt-get install pkg-config
```

We download the 64-bit package from

```
wget http://downloads.dlang.org/releases/2.x/2.093.1/dmd_2.093.1-0_amd64.deb
```

and install it at system level

```
sudo dpkg -i dmd_2.093.1-0_amd64.deb
```

Then we clone the required repository files with git:

```
git clone https://github.com/abraunegg/onedrive.git
```

Then compile and install it:

```
cd onedrive
./configure
make
sudo make install
```

The next step, you need to authorize onedrive with Microsoft so it can access your account:

```
onedrive
```

Copy the web address appeared in the terminal window right after Authorize this app visiting:, then open any web browser and paste it into the web browser address line.

Log in into your account and grant the application permission to acess your account.

Wait for response ...

Then copy and paste the web address of response into corresponding part of the terminal window with prompt Enter the response uri:.

Now the installation has been done.

Before syncronization, let us create a onedrive directory for current user

```
mkdir -p ~/OneDrive
```

and copy a configuration file into default directory

```
cp config ~/.config/onedrive/config
```

Now we can leave the installation directory

```
cd
```

It is a time to check our current configuration

```
onedrive --display-config
```

You will see something like this

```
onedrive version                     = v2.4.0-25-g7f267a5
Config path                          = /home/yourcurrentusername/.config/onedrive
Config file found in config path     = true
Config option 'check_nosync'         = false
Config option 'sync_dir'             = /home/yourcurrentusername/OneDrive
Config option 'skip_dir'             =
Config option 'skip_file'            = ~*|.~*|*.tmp
Config option 'skip_dotfiles'        = false
Config option 'skip_symlinks'        = false
Config option 'monitor_interval'     = 45
Config option 'min_notify_changes'   = 5
Config option 'log_dir'              = /var/log/onedrive/
Config option 'classify_as_big_delete' = 1000
Config option 'sync_root_files'      = false
Selective sync configured            = false
```

For one time sync, run

```
onedrive --synchronize
```

or if you want to monitor your OneDrive storage for changes

```
onedrive –monitor
```

Create Custom Ubuntu ISO Image

First step is to download the Ubuntu 20.04 ISO image. Select any desired desktop flavour whether it be Ubuntu, Kubuntu, Lubuntu etc: https://linuxconfig.org/ubuntu-20-04-download

And install cubic

```
sudo apt-add-repository ppa:cubic-wizard/release

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 6494C6D6997C215E
sudo apt update

sudo apt install cubic
```

Then follow the steps mentioned in the following link to make ur custom bootable ubuntu iso file: https://www.techrepublic.com/article/how-to-create-a-custom-ubuntu-iso-with-cubic/

To install OpenCV with Cuda plugin

OpenCV 4.2.0 with CUDA 10.1 on Ubuntu 20.04 LTS (Focal Fossa).

Please follow the instructions mentioned in the link:

https://medium.com/@sb.jaduniv/how-to-install-opencv-4-2-0-with-cuda-10-1-on-ubuntu-20-04-lts-focal-fossa-bdc034109df3

Reference

1. **https://docs.nvidia.com/metropolis/deepstream/dev-guide/text/DS_Quickstart.html#dgpu-setup-for-ubuntu**