- The qemu-user-static package has been installed on the Linux host:

```
sudo apt-get install qemu-user-static
```

```
docker load -i
./sdkmanager-[version].[build#]-[base_OS]_docker.tar.gz
```

Example:

```
kjw@concepta:~/Downloads$ sudo docker load -i "./sdkmanager-1.8.0.10363-Ubuntu_18.04_docker (1).tar.gz"
[sudo] password for kjw:
Loaded image: sdkmanager:1.8.0.10363-Ubuntu_18.04
```

```
docker tag sdkmanager:[version].[build#] sdkmanager:latest
```

```
kjw@concepta:~/Downloads$ sudo docker tag sdkmanager:1.8.0.10363-Ubuntu_18.04 sdkmanager:test
```

Run a docker container. Will exit so you will need to rerun and enter the

```
kjw@concepta:~/Downloads$ sudo docker run -it --privileged -v /dev/bus/usb:/dev/bus/usb/ -v /dev:/dev -v /media/$USER:/media/nvidia:slave --name JetPack_NanoTest --network host
 sdkmanager --cli install --logintype devzone --product Jetson --target JETSON_NANO --targetos Linux --archivedversions --version 4.4.1 --select 'Jetson OS' --deselect 'Jetson
SDK Components' --flash skip --license accept --staylogin true --datacollection enable

 ===== INSTALLATION COMPLETED SUCCESSFULLY. =====
      - Drivers for Jetson: Installed
      - File System and OS: Installed
      - Device Mode Host Setup in Flash: Skipped
      - Flash Jetson Nano (Devkit): Skipped
      - Flash Jetson Nano: Skipped
      - Flash Jetson Nano 2GB (Devkit): Skipped
      - Device Mode Host Setup in Target SDK: Skipped
      - DateTime Target Setup: Skipped
      - CUDA Toolkit for L4T: Skipped
      - cuDNN on Target: Skipped
      - TensorRT on Target: Skipped
      - OpenCV on Target: Skipped
      - VisionWorks on Target: Skipped
      - VPI on Target: Skipped
      - NVIDIA Container Runtime with Docker integration (Beta): Skipped
      - Multimedia API: Skipped

 ===== Installation completed successfully - Total 16 components =====
 ===== 2 succeeded, 0 failed, 0 up-to-date, 14 skipped =====
```

Restart docker container
Enter docker container terminal

```
kjw@concepta:~/Downloads$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
kjw@concepta:~/Downloads$ sudo docker start JetPack_NanoTest
JetPack_NanoTest
kjw@concepta:~/Downloads$ sudo docker ps
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS         PORTS     NAMES
433883581ae7   sdkmanager   "docker-entrypoint.s…"   14 minutes ago   Up 2 seconds             JetPack_NanoTest
kjw@concepta:~/Downloads$ sudo docker exec -it JetPack_NanoTest bash
nvidia@concepta:~$ ls
Downloads  nvidia
```

Navigate to the flash script and Call the flash.sh specifying the jetson board and dev root

```
nvidia@concepta:~$ cd nvidia/nvidia_sdk/JetPack_4.4.1_Linux_JETSON_NANO/Linux_for_Tegra/
nvidia@concepta:~/nvidia/nvidia_sdk/JetPack_4.4.1_Linux_JETSON_NANO/Linux_for_Tegra$ ls
README_Massflash.txt            jetson-nano-devkit.conf   nv_tegra                        p3448-0000.conf.common             rootfs
TX1_boot-firmware-redundancy.txt jetson-nano-emmc.conf    nv_tools                        p3449-0000+p3448-0000-qspi-sd.conf source
apply_binaries.sh               jetson-nano-qspi-sd.conf  nvmassflashgen.sh               p3449-0000+p3448-0000-qspi.conf    source_sync.sh
bootloader                      jetson-nano-qspi.conf     p2371.conf                      p3449-0000+p3448-0002.conf         tools
build_l4t_bup.sh                jetson-tx1-devkit.conf    p2597-0000+p2180-1000-24x7.conf p3450.conf
flash.sh                        jetson-tx1.conf           p2597-0000+p2180-1000.conf      p3542-0000+p3448-0003-qspi-sd.conf
jetson-nano-2gb-devkit.conf     kernel                    p3448-0000-max-spi-sd.conf      p3542-0000+p3448-0003-qspi.conf
jetson-nano-devkit-emmc.conf    l4t_generate_soc_bup.sh   p3448-0000-max-spi.conf         p3542.conf
nvidia@concepta:~/nvidia/nvidia_sdk/JetPack_4.4.1_Linux_JETSON_NANO/Linux_for_Tegra$ ./flash.sh jetson-nano-emmc mmcblk0p1
################################################################################
# L4T BSP Information:
# R32 , REVISION: 4.4
################################################################################
flash.sh requires root privilege
nvidia@concepta:~/nvidia/nvidia_sdk/JetPack_4.4.1_Linux_JETSON_NANO/Linux_for_Tegra$ sudo ./flash.sh jetson-nano-emmc mmcblk0p1
```

```
[ 208.5008 ] Writing partition RP4 with rp4.blob
[ 208.5371 ] [.....................................................] 100%
[ 208.5913 ]
[ 208.5925 ] tegradevflash --write BCT P3448_A00_lpddr4_204Mhz_P987.bct
[ 208.5935 ] Cboot version 00.01.0000
[ 208.5953 ] Writing partition BCT with P3448_A00_lpddr4_204Mhz_P987.bct
[ 208.5957 ] [.....................................................] 100%
[ 208.9938 ]
[ 208.9938 ] Flashing completed

[ 208.9939 ] Coldbooting the device
[ 208.9960 ] tegradevflash --reboot coldboot
[ 208.9970 ] Cboot version 00.01.0000
[ 208.9990 ]
*** The target t210ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

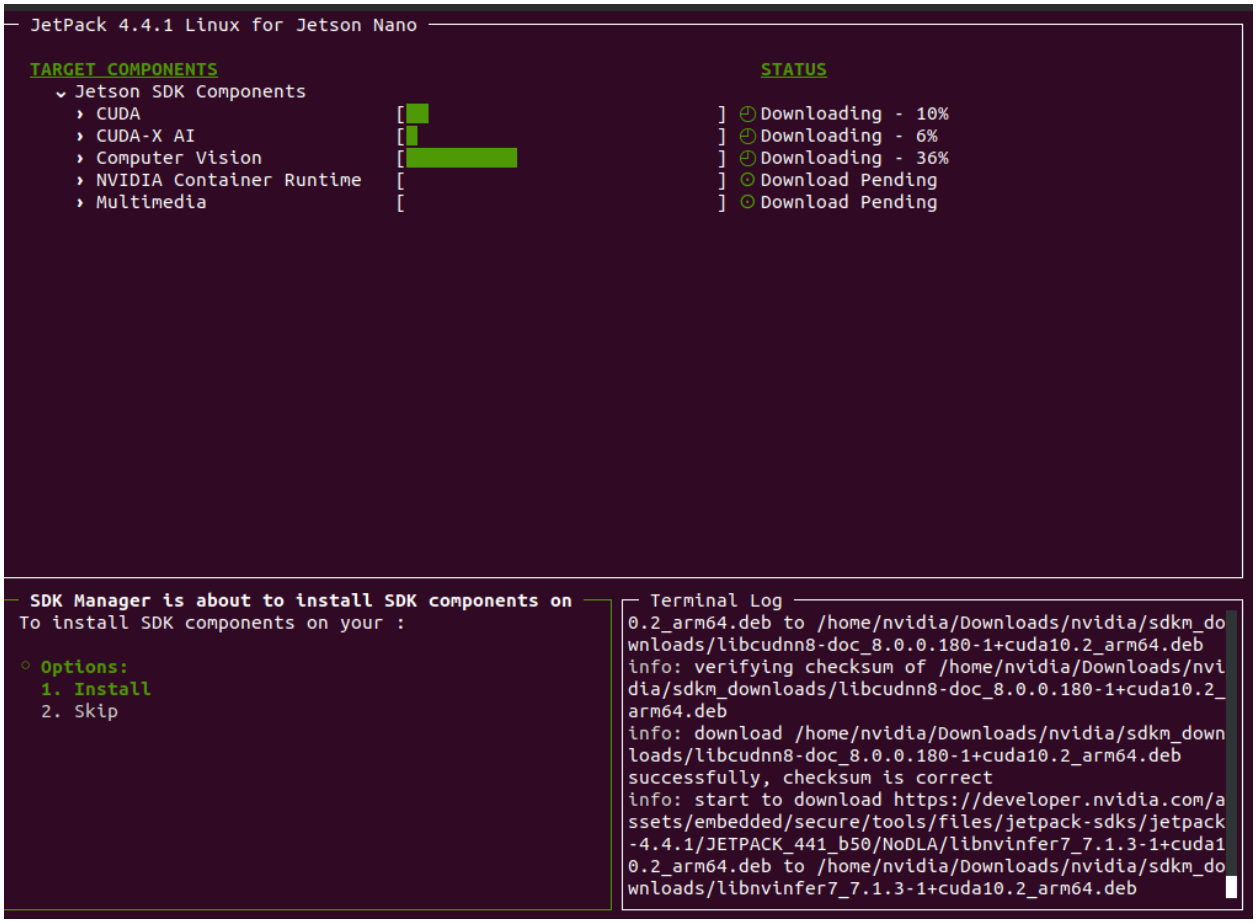
ALT: Can alternatively call massflash with correct argument for flashing multiple

Unpower and repower the Nano, not in recovery mode.

Step though the System Configuration and note the new login credentials
Login and not the ip-address
Ifconfig → eth0 → inet

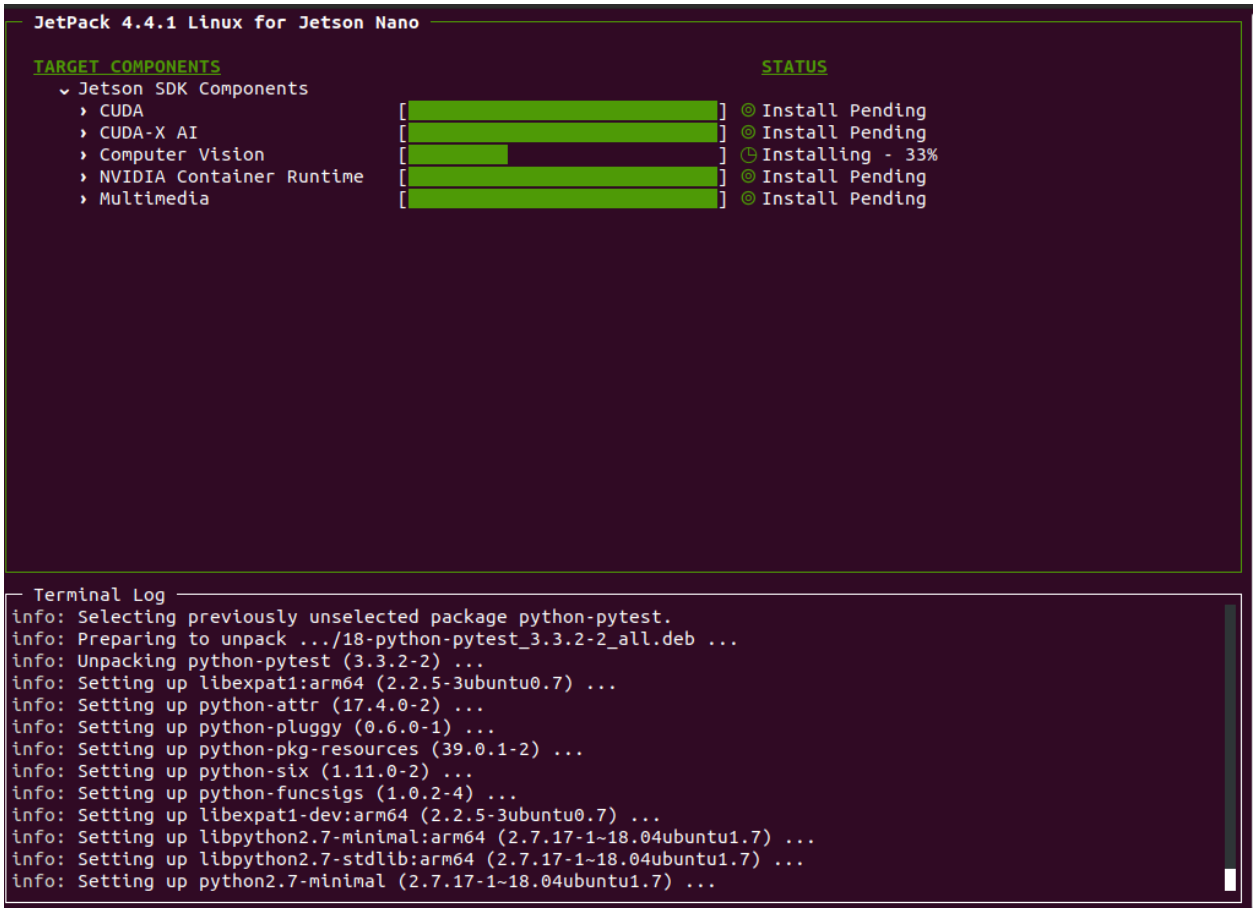
Flashing the Jetson SDK Components

```
kjw@concepta:~$ sudo docker run -it --privileged -v /dev/bus/usb:/dev/bus/usb/ -v /dev:/dev -v /media/$USER:/medi
a/nvidia:slave --name SDK_Tools --network host sdkmanager --cli install --logintype devzone --product Jetson --ta
rget P3448-0002 --targetos Linux --archivedversions --version 4.4.1 --deselect 'Jetson OS' --select 'Jetson SDK C
omponents' --license accept --staylogin true --datacollection enable
```



Select Install > Connect via USB Cable > Choose the connected board that shows up > IPv4 > Change IP address to that of the nano > username and password

Allow all downloading and installation to complete




```
===== INSTALLATION COMPLETED SUCCESSFULLY. =====
     - Device Mode Host Setup in Target SDK: Installed
     - DateTime Target Setup: Installed
     - CUDA Toolkit for L4T: Installed
     - cuDNN on Target: Installed
     - TensorRT on Target: Installed
     - OpenCV on Target: Installed
     - VisionWorks on Target: Installed
     - VPI on Target: Installed
     - NVIDIA Container Runtime with Docker integration (Beta): Installed
     - Multimedia API: Installed

===== Installation completed successfully - Total 10 components =====
===== 10 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```