# Ubuntu 20 (Jetpack >=5.0 & < 6.0)

## Expanding Platform's disk space with SD card

On Linux Host Machine:

This section is needed in order to build big packages from source.

- Reformat SD card as ext4 using GParted (SD card needs to use the file system  ext4 )

```
# install gparted
$ sudo apt install gparted
# after you have installed it (execute it)
$ gparted
```

- Create partition as follow:

  Partition Format:  ext4
  Partition Label:  slamr01_sd

- Exit gparted ( ensure to  generate partition).
- insert Sd card to the Jetson and unmount it.

```
# you can use 'df' to ensure this the partition of the SD card.
$ sudo umount /dev/mmcblk1p1
```

On Jetson:

- make the directory for mounting the SD card (make sure the name matches)

> *$ sudo mkdir -p /mnt/{name_of_sd_card_label}*

- Use fstab to assign a mounting location for the SD card :

> *$ sudo vi /etc/fstab*

  Edit file by adding the following line

```
/dev/mmcblk1p1      /mnt/slamr01_sd          ext4       defaults,nofail,x-systemd.mount-timeout=5      0 0
```

- mount SD card  using the following command:

> *$ sudo mount -a*

- Once the drive have mounted the desired directory, you can set it for its folders to  be own by the user.

```
#
$ cd /mnt/slamr01_sd

# set subfolder to be own by the user:
$ sudo chown -R slamr01:slamr01 ./

# now you should be able to create folder and show to be owned by the user.

$ mkdir programs workspace data
```

- Basic packages to configure you image:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt autoremove

$ sudo apt install git cmake cmake-curses-gui htop ccache python3-pip dnsutils curl apt-utils build-essential

# upgrade pip
$ sudo -H python3 -m pip install --upgrade pip

# install jetson-stat
# (I believe this is for the jtop application, run with command "sudo jtop")
$ sudo -H python3 -m pip install -U jetson-stats
```

- Create symbolic links to SD card:

```
# from user space
$ cd ~

$ ln -s /mnt/${NAME_OF_DRIVE}/data data
$ ln -s /mnt/${NAME_OF_DRIVE}/programs programs
$ ln -s /mnt/${NAME_OF_DRIVE}/workspace workspace
```

- if you have installed ccache

the cache is located in the user space and it takes a lot of disk

> *add the path the ccache directory,  in the file ~/.ccache/ccache.conf*
>
> *cache_dir=${HOME}/workspace/.ccache*

```
# if folder has not been created
$ mkdir ~/.ccache
$ mkdir ~/workspace/.ccache
$ cd ~/.ccache
$ vi ccache.conf

# --- write in the file:
cache_dir=${HOME}/workspace/.ccache
# ------- save file

# check file
$ ccache --show-config
$ ccache --show-stats
```

- If using Visual Studio code from a remote machine, it creates a big folder that gets big over time.

Create a symbolic link to the location where you want vscode to install it's packages for remote access.

```
$ mkdir ~/workspace/.vscode-server

$ cd ~
$ ln -s /home/slamr01/workspace/.vscode-server .vscode-server
```

It looks like when installing ROS, the OS is forced to use Python2, In order to avoid this behavior , use update-alternative:

```
$ sudo update-alternatives --install /usr/bin/python python /usr/bin/python2.7 2

# For Orin with Ubuntu 20 (JetPack 5)
$ sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.8 3

#check the config
update-alternatives --list python
update-alternatives --query python
```