Exercice 1 : Installation des environnements

* Téléchargez ubuntu20.04 destiné pour raspberry pi pour la suite des TP
* Téléchargez balena etcher pour insérer l’image dans une carte mémoire
* Enlever la carte puis réinsérer dans l’ordinateur pour configurer network-config (dynamique ou statique).

Pour le cas d’adressage dynamique : Installer dans l’ordinateur freeipscanner ou bien nmap (n’utiliser pas advanced ip scanner)

* Insérer carte mémoire et démarrez raspberry pi
* Configurer les mots des passes
* Installer

AVANT DE FAIRE L’EXERCICE : SUIVRE CE VIDEO :

<https://www.youtube.com/watch?v=GVgMM_TFeOw>

<https://github.com/DavidUnboxed/Ubuntu-20.04-WiFi-RaspberyPi4B/blob/master/Files%20to%20Copy%20to%20SD%20Card/user-data>

REPONSES :

Etape 1 : Installation de l’image ubuntu

SUIVRE CE VIDEO :

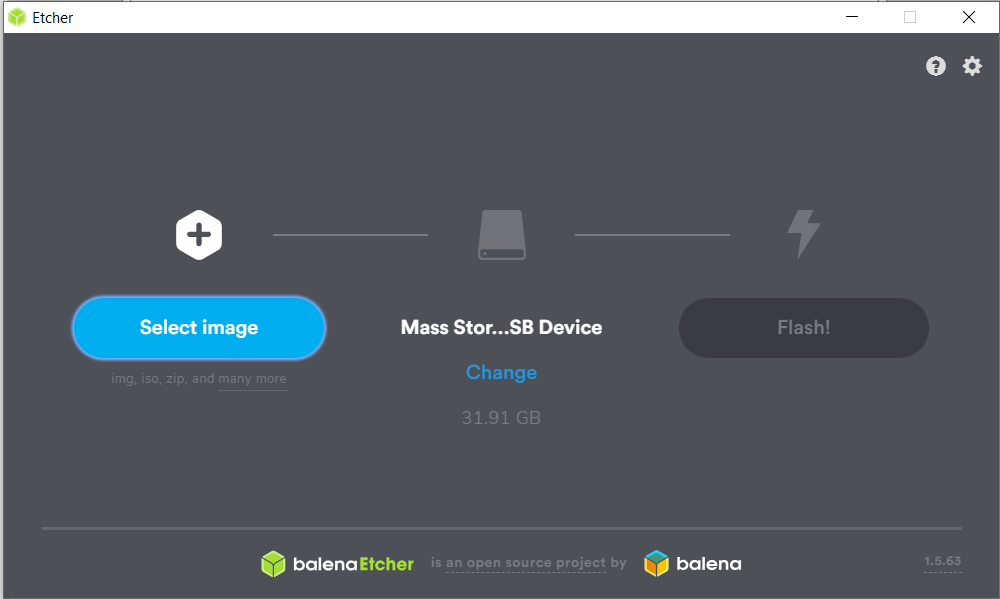
<https://www.youtube.com/watch?v=GVgMM_TFeOw>

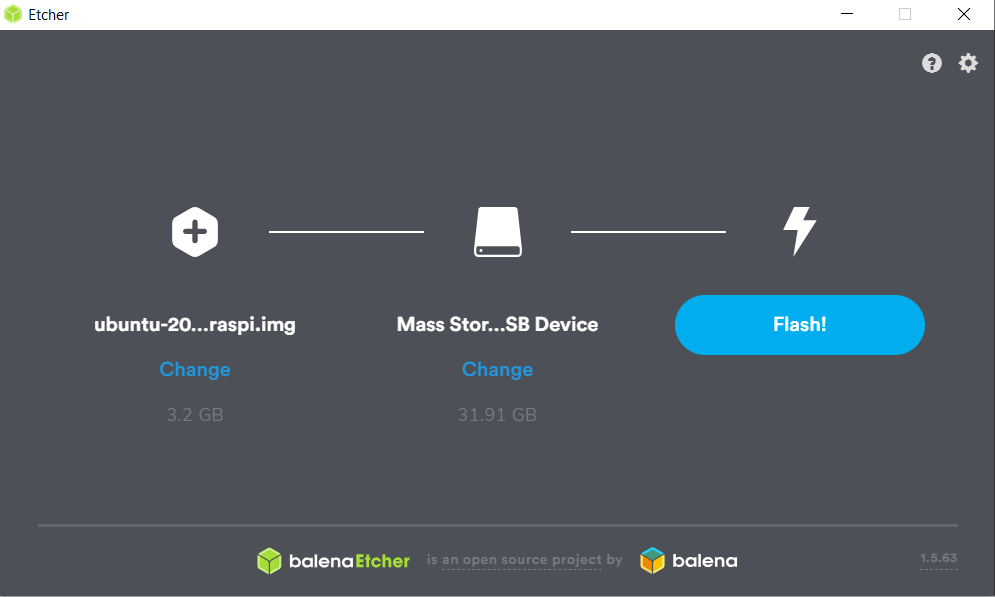
<https://github.com/DavidUnboxed/Ubuntu-20.04-WiFi-RaspberyPi4B/blob/master/Files%20to%20Copy%20to%20SD%20Card/user-data>

Les versions des logiciels et des images sont très importantes, sinon l’installation ne se fait pas :

> Installer balena etcher

> Flasher l’image de ubuntu 64bit arm (ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz) en utilisant balena etcher

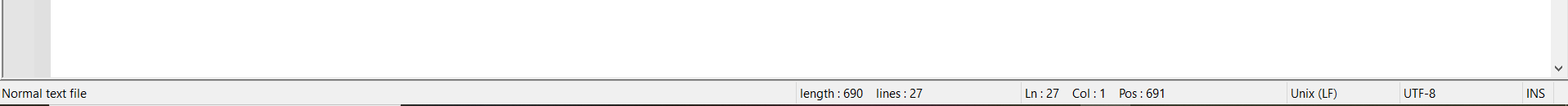




* Cliquer sur flash

Ajouter la configuration network-config, enregistrer en tant que UTF-8 et UNIX-LF (avec notepad ++) dans :

* Encodage > UTF-8
* Edition > Convertir les sauts de ligne > unix LF



Choix de configuration : statique ou dynamique

* Configuration pour une adresse dynamique

*# This file contains a netplan-compatible configuration which cloud-init*

*# will apply on first-boot. Please refer to the cloud-init documentation and*

*# the netplan reference for full details:*

*#*

*# https://cloudinit.readthedocs.io/*

*# https://netplan.io/reference*

*#*

*# Some additional examples are commented out below*

*version: 2*

*renderer: networkd*

*ethernets:*

*eth0:*

*dhcp4: true*

*optional: true*

*wifis:*

*wlan0:*

*dhcp4: true*

*optional: true*

*access-points:*

*"imsicatcher":*

*password: "imsicatcher"*

Configuration pour le cas d’une adresse statique :

*# This file contains a netplan-compatible configuration which cloud-init*

*# will apply on first-boot. Please refer to the cloud-init documentation and*

*# the netplan reference for full details:*

*#*

*# https://cloudinit.readthedocs.io/*

*# https://netplan.io/reference*

*#*

*# Some additional examples are commented out below*

*version: 2*

*renderer: networkd*

*ethernets:*

*eth0:*

*dhcp4: true*

*optional: true*

*wifis:*

*wlan0:*

*dhcp4: no*

*addresses: [192.168.43.205/24]*

*gateway4: 192.168.43.1*

*nameservers:*

*addresses: [192.168.43.1,8.8.8.8,8.8.4.4]*

*access-points:*

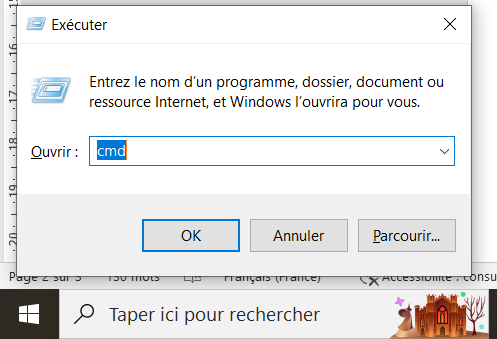
*"alcatel POP 4 3a8a":*

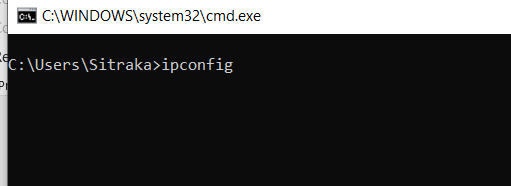
*password: "mi0123456789"*

Lien :

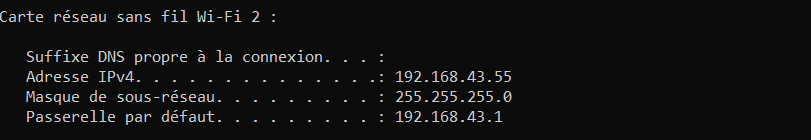
<https://drive.google.com/drive/folders/1izdP7fxxxae9nORYUSIOnE8xHGRxHeCQ>

* Démarrer l’AP concerné
* Win+R puis cmd et tapez ipconfig

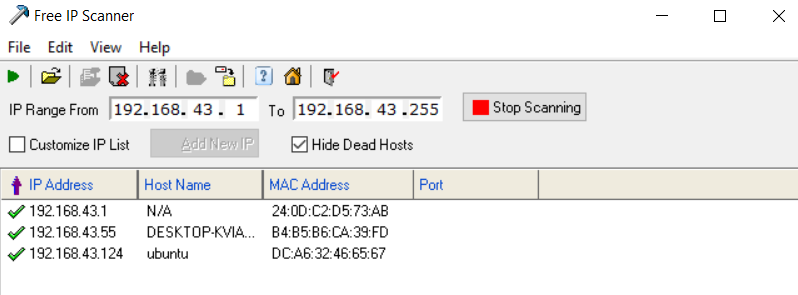




* Rechercher carte réseau sans fil



* Faire une recherche approfondie sur le réseau (dans mon cas 192.168.43.1 – 192.168.43.255) en utilisant de préférence freeipscanner (à telecharger) ou namp



Dans le cas d’adresse statique : ubuntu devra être l’adresse statique que vous avez mentionnée 192.168.43.205

(PARFOIS LE PREMIER DEMARRAGE NE MONTRE PAS LE RPI connecté au WIFI, enlever l’alimentation de rpi et redémarrez le rpi)

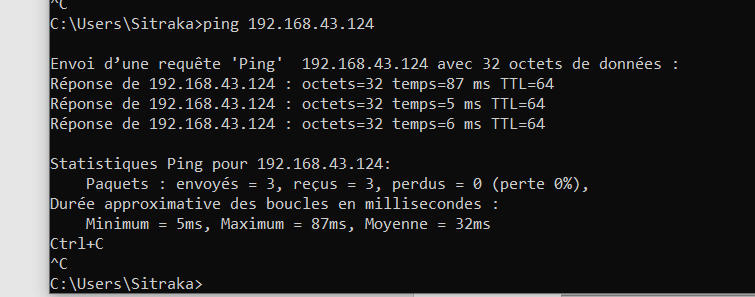
* Tester :

Pour la configuration dynamique

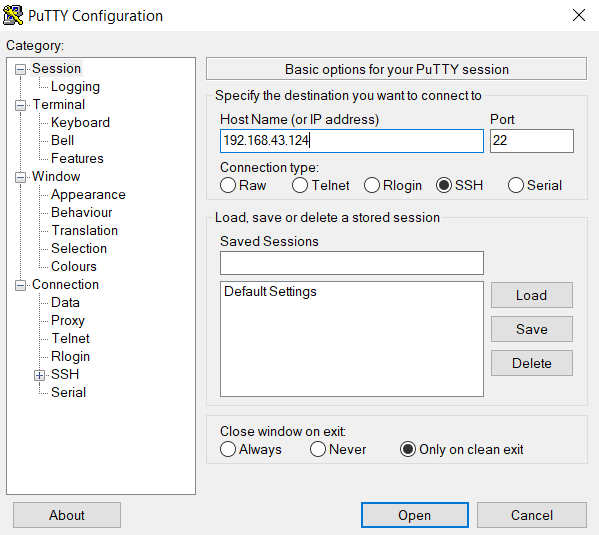
**ping 192.168.43.124**

**Pour la configuration statique**

**ping 192.168.43.205**

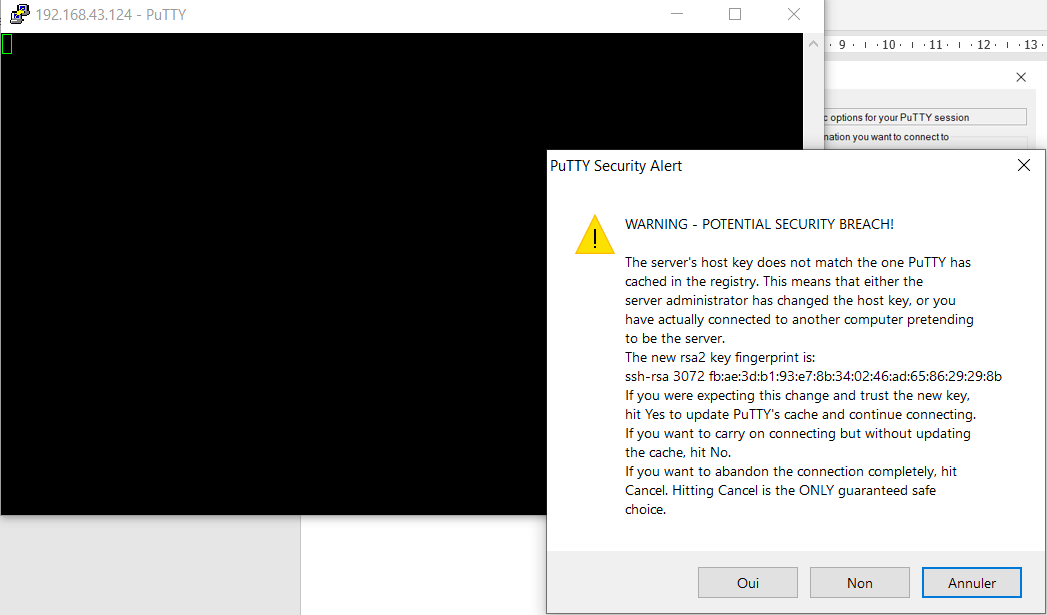


* Installer putty puis entrer l’adresse ip et accéder via ssh



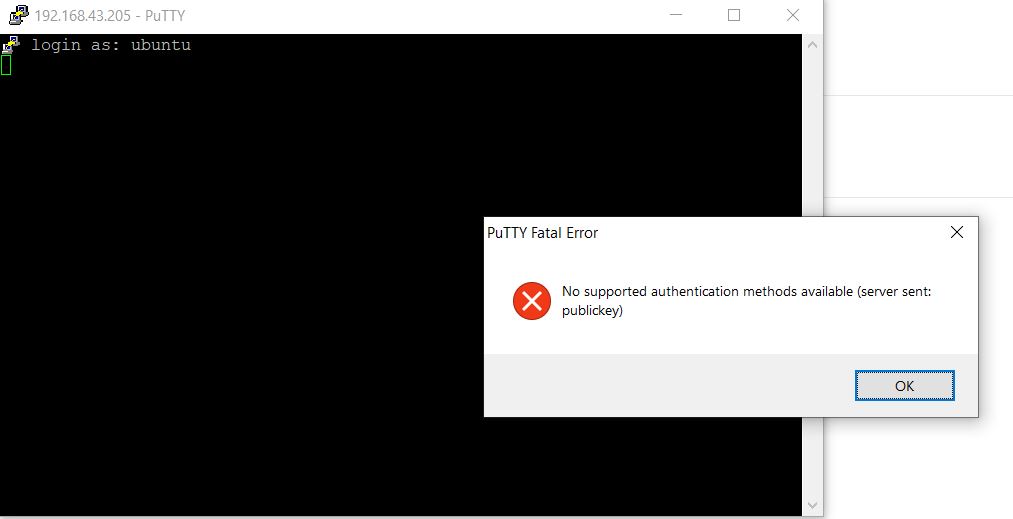
Pour le cas de l’adressage statique : tapez 192.168.43.205

* Cliquez sur OK



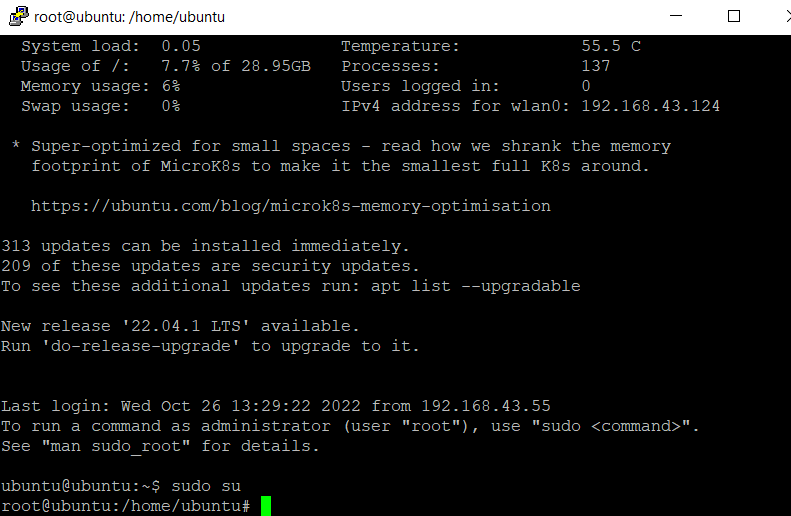
* Tapez comme login : ubuntu et mdp : ubuntu puis renouvelez-en (imsicatcher ou ubuntupi) (putty se ferme automatiquement)

*REMARQUES : Erreur lors de la connexion ssh*

**

*Veuillez attendre un peu et puis ré-utiliser putty*

* Ré-entrer l’addresse IP via putty avec le nouveau mot de passe
* Tapez sudo su

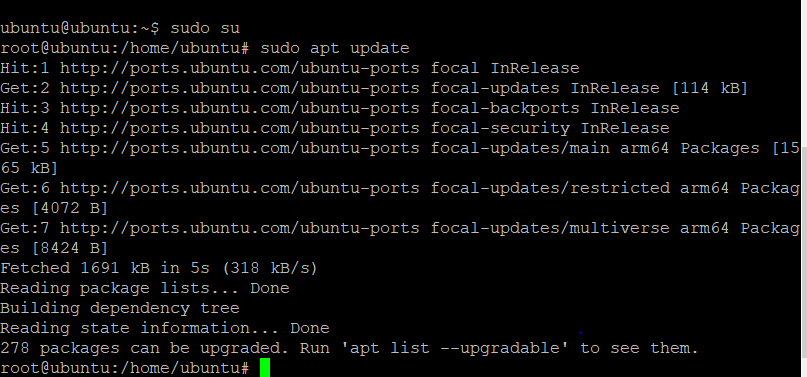


Etape 2 : Installation Desktop xfce et tiger VNC

SUIVRE CE TUTORIEL : <https://bytexd.com/how-to-install-configure-vnc-server-on-ubuntu/>

**rm -rf /var/lib/apt/lists/lock**

**sudo apt update**



**rm -rf /var/lib/dpkg/lock-frontend**

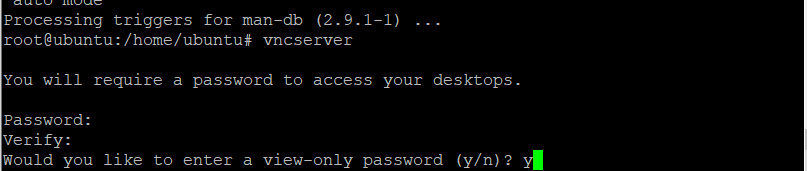
**rm -rf /var/lib/dpkg/lock**

**sudo apt-get install xfce4 xfce4-goodies**

**sudo apt-get install tigervnc-standalone-server**

* Changer mot de passe (ubuntu) de vnc en utilisant  la commande:

**vncserver**



**vncserver -kill :1**

**nano /root/.vnc/xstartup**

*#!/bin/sh*

*# Start up the standard system desktop*

*unset SESSION\_MANAGER*

*unset DBUS\_SESSION\_BUS\_ADDRESS*

*/usr/bin/startxfce4*

***[*** *-x /etc/vnc/xstartup* ***]******&&*** *exec /etc/vnc/xstartup*

***[*** *-r $HOME/.Xresources* ***]******&&*** *xrdb $HOME/.Xresources*

*x-window-manager &*

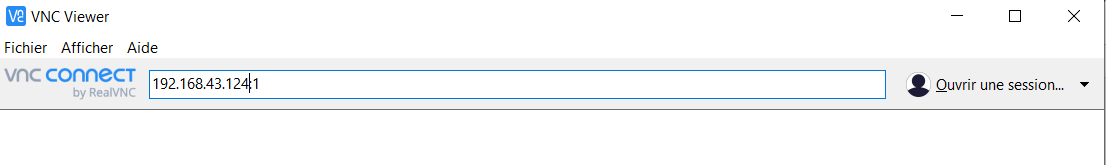
**chmod +x /root/.vnc/xstartup**

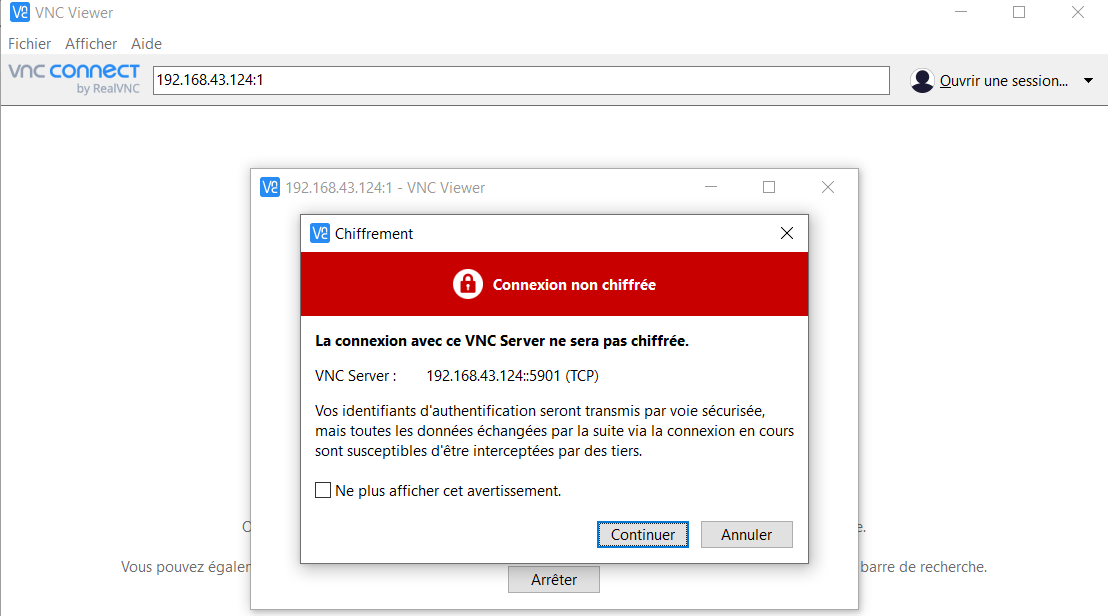
**vncserver -localhost no :1**

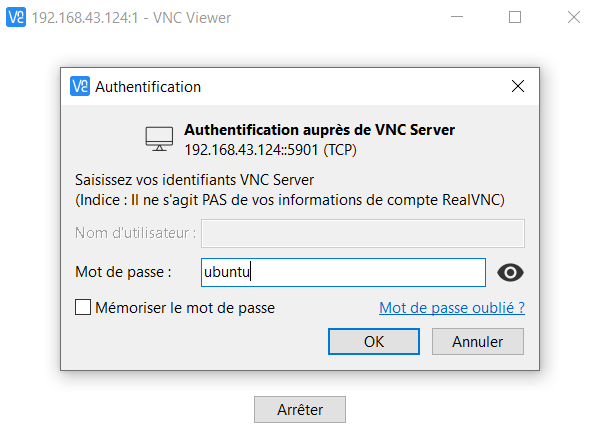
* Telecharger puis installer vncviewer sous windows, puis tapez addresseip:1

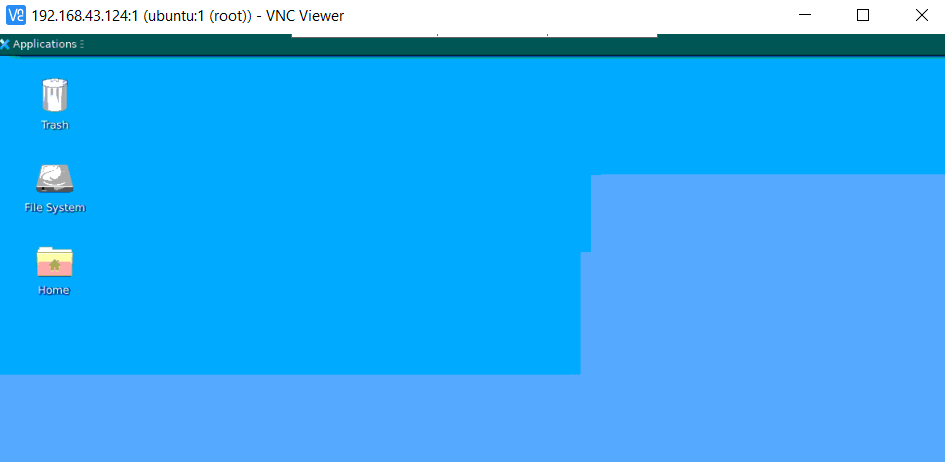
Dans mons cas 192.168.43.124:1

Dans le cas d’adresse static : 192.168.43.205 :1









* Fermer VNCviewer

**nano /etc/systemd/system/vncserver@.service**

*[Unit]*

*Description=Start TigerVNC server at startup*

*After=syslog.target network.target*

*[Service]*

*Type=forking*

*User=ubuntu*

*Group=ubuntu*

*WorkingDirectory=/home/ubuntu*

*PIDFile=/home/ubuntu/.vnc/%H:%i.pid*

*ExecStartPre=-/usr/bin/vncserver -kill :%i > /dev/null 2>&1*

*ExecStart=/usr/bin/vncserver -depth 24 -geometry 1280x800 -localhost :%i*

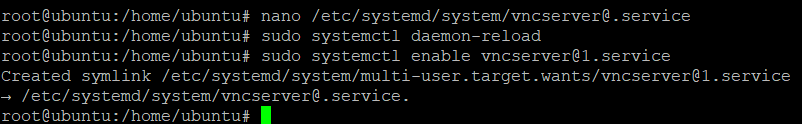
*ExecStop=/usr/bin/vncserver -kill :%i*

*[Install]*

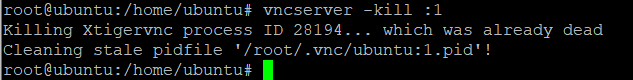
*WantedBy=multi-user.target*

**sudo systemctl daemon-reload**

**sudo systemctl enable** [**vncserver@1.service**](mailto:vncserver@1.service)

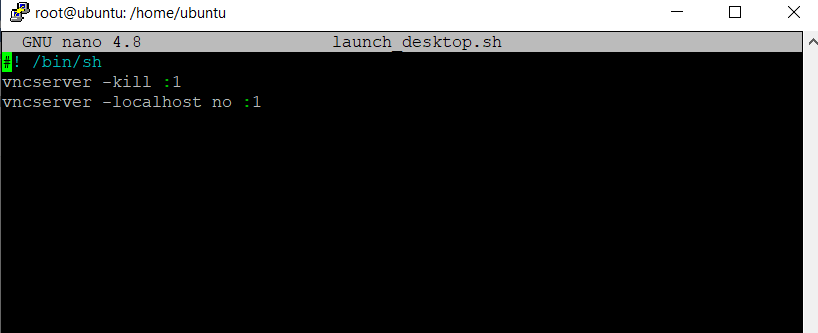


**vncserver -kill :1**



**sudo systemctl enable vncserver@1.service**

**nano launch\_desktop.sh**



*#! /bin/sh*

*vncserver -kill :1*

*vncserver -localhost no:1*

**chmod +x launch\_desktop.sh**

RESUME POUR OUVRIR UNE SESSION VNC :

1. Tapez adresse ip via putty
2. Tapez login et mdp
3. sudo su
4. **bash launch\_desktop.sh**
5. Lancer vncview
6. Tapez addresse\_ip :1
7. Tapez mdp : ubuntu

INSTALLATION KARLI BTS :

(commande en une seul ligne)

**sudo apt-get install libboost-all-dev libusb-1.0-0-dev doxygen python3-docutils python3-mako python3-numpy python3-requests python3-ruamel.yaml python3-setuptools cmake build-essential**

**apt-get remove texinfo**

**mkdir /root/Desktop/karli**

**cd /root/Desktop/karli/**

**apt-get install alsa-base alsa-oss alsa-utils**

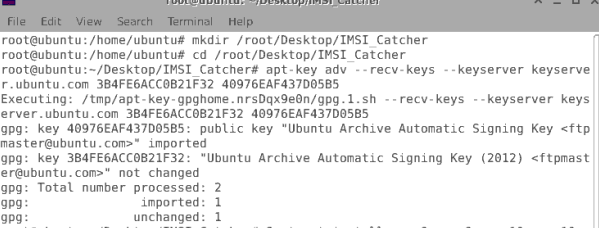
**apt-get install oss-compat**

**modprobe snd-pcm-oss**

**modprobe snd-mixer-oss**

**(commande en un seul ligne)**

**apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 3B4FE6ACC0B21F32 40976EAF437D05B5**



**apt-get install gcc-9 g++-9 gcc-10 g++-10 git -y**

**mousepad /etc/apt/sources.list**

* Ajouter à la fin :

*# adding this one*

*deb http://ports.ubuntu.com/ubuntu-ports xenial main restricted universe multiverse*

**apt update**

**apt-get install gcc-4.9 g++-4.9 gcc-7 g++-7 -y**

**mousepad /etc/apt/sources.list**

* changer à la fin en :

*# adding this one*

*# deb http://ports.ubuntu.com/ubuntu-ports xenial main restricted universe multiverse*

**apt update**

**(commande en un seul ligne)**

**apt-get install -y build-essential libusb-1.0-0-dev libsqlite3-dev libsctp-dev libgmp-dev libx11-6 libx11-dev flex libncurses5 libdbd-sqlite3 libdbi-dev libncurses5-dev libncursesw5 libpcsclite-dev zlib1g-dev libmpc3 lemon aptitude libtinfo-dev libtool shtool autoconf git-core pkg-config make libmpfr-dev libmpc-dev libtalloc-dev libfftw3-dev libgnutls28-dev libtool-bin libxml2-dev sofia-sip-bin libsofia-sip-ua-dev sofia-sip-bin libncursesw5-dev bison libgmp3-dev alsa-oss**

**(commande en un seul ligne)**

**update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 49 --slave /usr/bin/g++ g++ /usr/bin/g++-4.9**

**(commande en un seul ligne)**

**update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 70 --slave /usr/bin/g++ g++ /usr/bin/g++-7**

**(commande en un seul ligne)**

**update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 90 --slave /usr/bin/g++ g++ /usr/bin/g++-9**

**(commande en un seul ligne)**

**update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-10 100 --slave /usr/bin/g++ g++ /usr/bin/g++-10**

**mousepad /etc/apt/sources.list**

Ajouter à la fin :

*# adding this another one*

*deb* [*http://ports*](http://ports)*.ubuntu.com/ubuntu-ports bionic main restricted universe multiverse*

**apt update**

**apt-get install -y gcc-5 g++-5 libssl1.0-dev**

**(commande en un seul ligne)**

**update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-5 50 --slave /usr/bin/g++ g++ /usr/bin/g++-5**

**mousepad /etc/apt/sources.list**

changer à la fin en :

*# adding this another one*

*# deb http://ports.ubuntu.com/ubuntu-ports bionic main restricted universe multiverse*

**apt update**

**update-alternatives --set gcc /usr/bin/gcc-4.9**

TOUS LES DEB SE TROUVENT DANS : /var/cache/apt/archives/

**cd /root/Desktop/karli/**

**wget** [**http://ftp.gnu.org/gnu/texinfo/texinfo-4.13.tar.gz**](http://ftp.gnu.org/gnu/texinfo/texinfo-4.13.tar.gz)

**tar xvf texinfo-4.13.tar.gz**

**cd texinfo-4.13**

**./configure --build=aarch64-unknown-linux-gnu**

**make -j$(nproc)**

**make -j$(nproc) install**

**ldconfig**

TELECHARGER GNU\_ARM\_FOR UBUNTU DANS:

<https://drive.google.com/file/d/1xV3RLZg3Lm4m3UPd1nDfO_xEjIH2F5lQ/view?usp=share_link>

**cd ..**

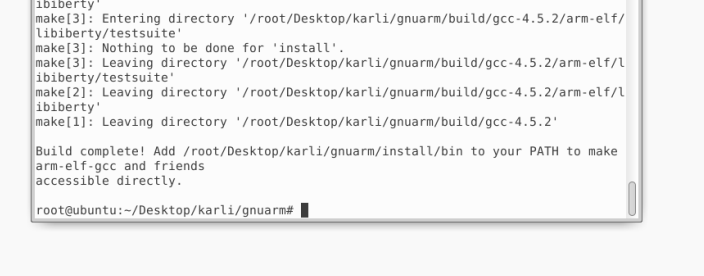
**unzip gnuarm\_for\_ubuntu.zip**

**cd gnuarm**

**chmod +x gnu-arm-build.2.sh**

**bash gnu-arm-build.2.sh**

BOIRE QUELQUES BIERRES PARCE QUE CA VA PRENDRE ENVION 5H :



make[1]: Leaving directory '/root/Desktop/karli/gnuarm/build/gcc-4.5.2'

Build complete! Add /root/Desktop/karli/gnuarm/install/bin to your PATH to make arm-elf-gcc and friends

accessible directly.

 --------------------------------------------------------------------------

**apt-get install zip**

**(commande en un seul ligne)**

**export PATH=$PATH:/root/Desktop/karli/gnuarm/install/bin**

**cd /root/Desktop/karli/**

**git clone https://github.com/osmocom/libosmocore**

**cd libosmocore**

**git checkout cf70aa0c40c574c32b832454f725cc37459c5d8d**

**cd ..**

**zip -r libosmocore.zip libosmocore/**

**cd libosmocore**

**autoreconf -i**

**./configure**

**make -j4**

**make install**

**ldconfig -i**

**apt-get install python2**

sudo ln -s /usr/bin/python2 /usr/bin/python

**cd /root/Desktop/karli/**

**git clone https://github.com/osmocom/osmocom-bb**

**cd osmocom-bb/**

**git checkout 4f677e6ba8434dab376495cd996d140548fa6e93**

**mousepad src/target/firmware/Makefile**

**\*| #uncomment CFLAGS += -DCONFIG\_TX\_ENABLE |\***

**\*| #ctrl+o return ctrl+x**

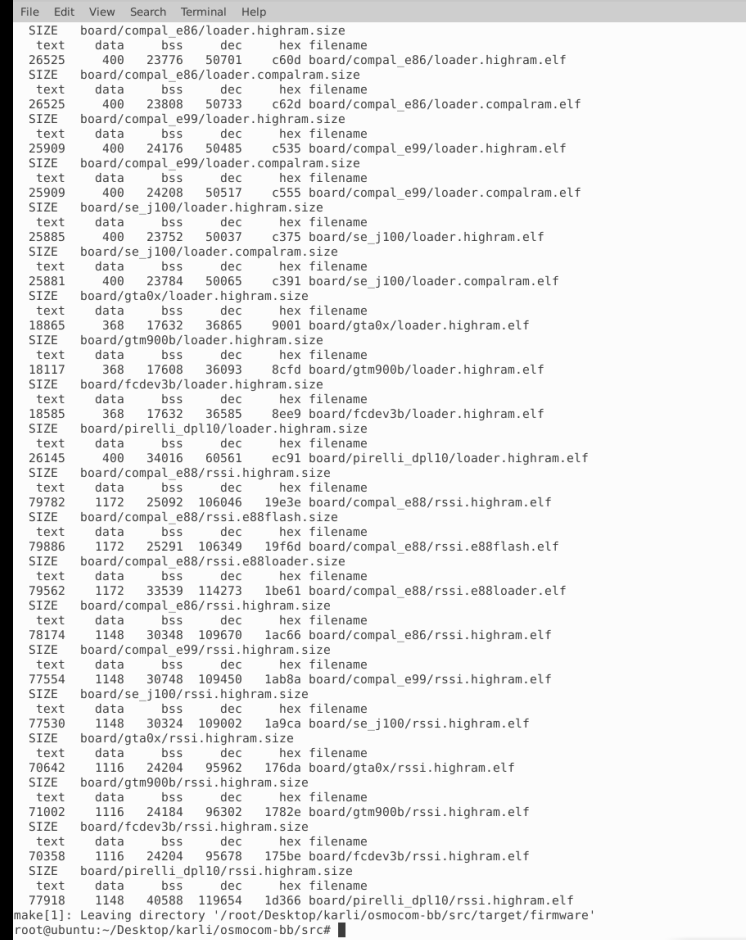
**cd ..**

**zip -r osmocom-bb.zip osmocom-bb/**

**cd osmocom-bb/**

**cd src**

**make -j4**

****

**cd /root/Desktop/karli/**

**git clone** [**https://github.com/osmocom/libosmo-dsp**](https://github.com/osmocom/libosmo-dsp)

**cd libosmo-dsp**

**git checkout 551b9752bcd5d3d21bb2df0736b1801bda3d0d10**

**cd ..**

**zip -r libosmo-dsp.zip libosmo-dsp**

**cd /root/Desktop/karli/libosmo-dsp/**

**autoreconf -i**

**./configure**

**make -j4**

**make install**

**ldconfig -i**

**cd /root/Desktop/karli/**

**git clone https://github.com/osmocom/osmocom-bb trx**

**cd trx**

**git checkout jolly/testing**

**#git checkout 620fe497efa492feff4550e336cc3f8167715936**

**cd src**

**wget https://github.com/bbaranoff/telco\_install\_sh/raw/main/trx.highram.bin**

**(commande en un seul ligne)**

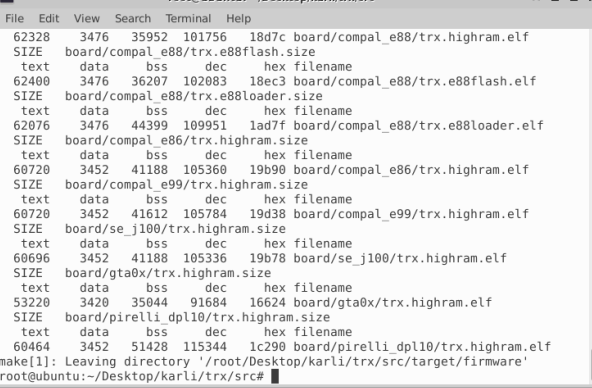
**sed -i -e 's/#CFLAGS += -DCONFIG\_TX\_ENABLE/CFLAGS += -DCONFIG\_TX\_ENABLE/g' target/firmware/Makefile**

**cd ../..**

**zip -r trx\_for\_ubuntu.zip trx/**

**cd trx/src**

**make -j4 HOST\_layer23\_CONFARGS=--enable-transceiver**



apt-get install libortp-dev

apt-get install dahdi-source

# gcc gcc-arm-none-eabi

apt-get install libtool shtool automake dahdi-source libssl-dev sqlite3 libsqlite3-dev libsctp-dev libfftw3-dev libfftw3-3 autoconf libsctp-dev libgnutls28-dev libcurl4-gnutls-dev git-core pkg-config make doxygen libtalloc-dev libpcsclite-dev libusb-1.0-0-dev

**cd /root/Desktop/karli/**

**git clone https://github.com/osmocom/osmocom-bb trx2**

**cd trx2**

**git checkout jolly/testing**

**#git checkout 620fe497efa492feff4550e336cc3f8167715936**

**cd src**

**wget https://github.com/bbaranoff/telco\_install\_sh/raw/main/trx.highram.bin**

**(commande en un seul ligne)**

**sed -i -e 's/#CFLAGS += -DCONFIG\_TX\_ENABLE/CFLAGS += -DCONFIG\_TX\_ENABLE/g' target/firmware/Makefile**

**cd ../..**

**zip -r trx2\_for\_ubuntu.zip trx2/**

**cd trx2/src**

**(commande en une seul ligne)**

**make -j4 HOST\_layer23\_CONFARGS=--enable-transceiver -e CROSS\_TOOL\_PREFIX=arm-none-eabi-**

**cd /root/Desktop/karli/**

cd libosmo-abis

git checkout 39dffb6c29a8d78ba8527aa4ccc13f34d1c3b319

cd ..

zip -r libosmo-abis.zip libosmo-abis/

cd libosmo-abis

autoreconf -i

./configure

make -j4

make install && \

ldconfig && \

\

**cd /root/Desktop/karli/**

cd /home/debian/osmocom/libosmo-netif/

git checkout 09c71b04f5a8d82515d0d4d541b8368b585dbd31

autoreconf -i

./configure

make -j4

make install

ldconfig

**cd /root/Desktop/karli/**

cd /home/debian/osmocom/openbsc/openbsc/ && \

git checkout d2550da76f9974bb1957f74c5d3eb75fdae923d9

autoreconf -i && \

./configure && \

make -j4 && \

make install && \

ldconfig && \

\

**cd /root/Desktop/karli/**

cd /home/debian/osmocom/osmo-bts/ && \

git checkout 59e7773055335a12d749faf84d88a8ed9fa0f201 && \

autoreconf -i && \

./configure --enable-trx && \

make -j4 && \

make install && \

ldconfig && \

\