APT FILE PATH : /var/cache/apt/archives/

Etape 1 : Installation de l’image ubuntu

SUIVRE CE VIDEO :

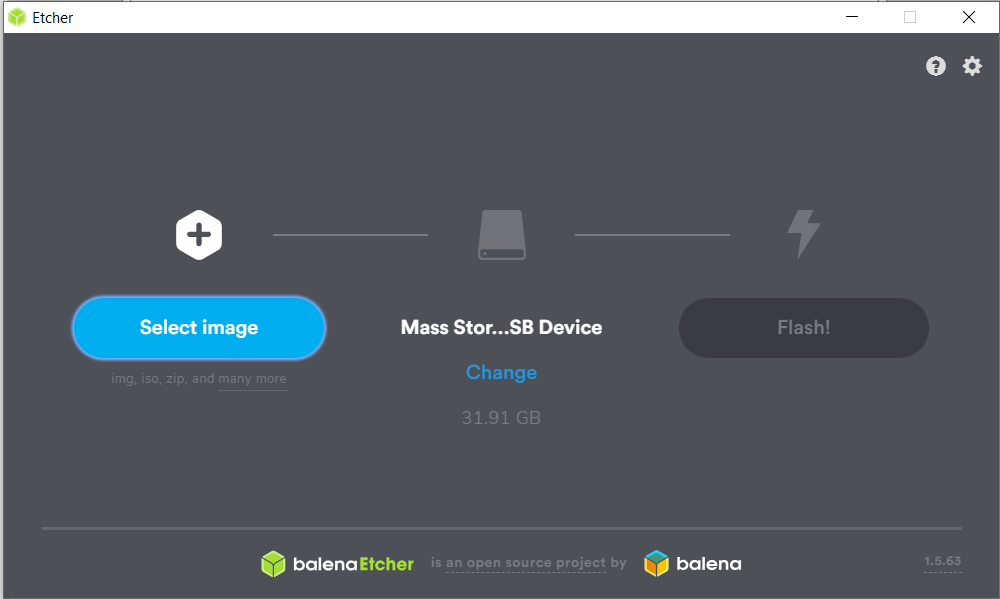
<https://www.youtube.com/watch?v=GVgMM_TFeOw>

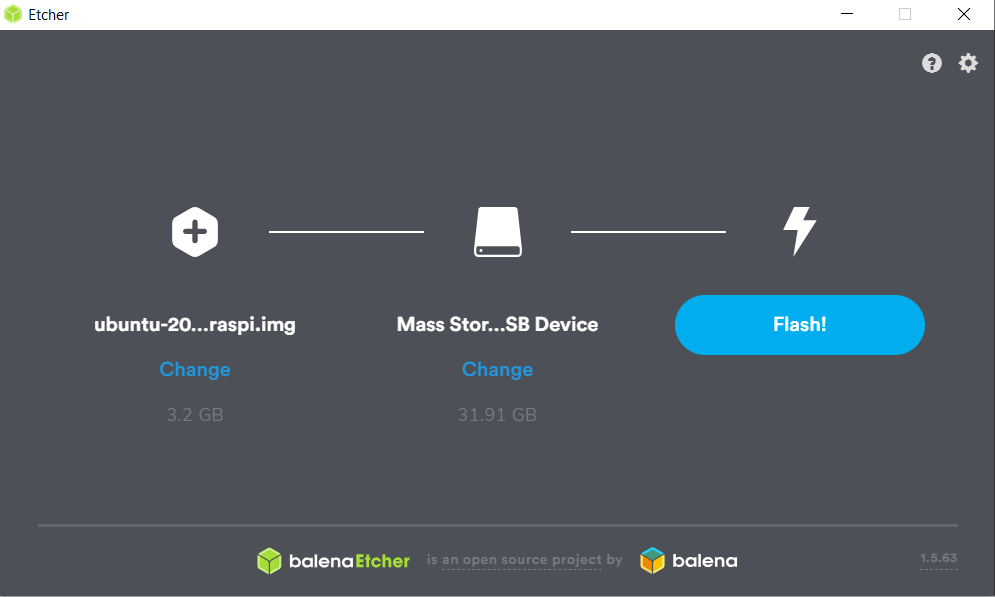
<https://github.com/DavidUnboxed/Ubuntu-20.04-WiFi-RaspberyPi4B/blob/master/Files%20to%20Copy%20to%20SD%20Card/user-data>

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

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:*

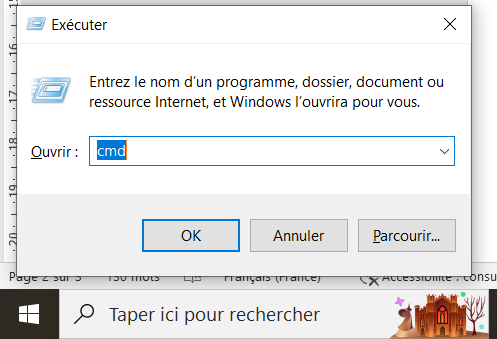
*addresses: [192.168.43.1,8.8.8.8,8.8.4.4]*

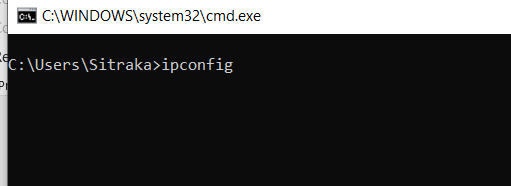
*access-points:*

*"alcatel POP 4 3a8a":*

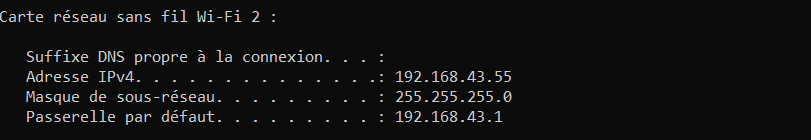
*password: "mi0123456789"*

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

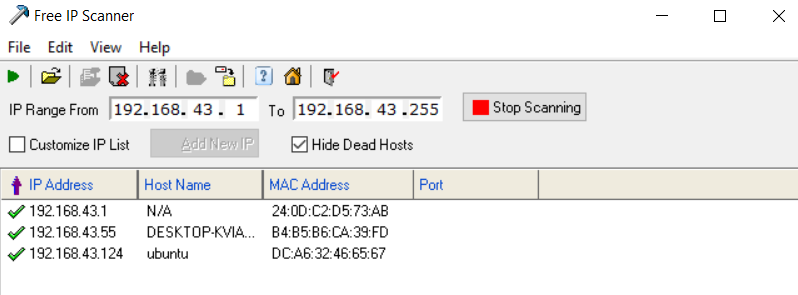




* Rechercher carte réseau sans fil



* 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)

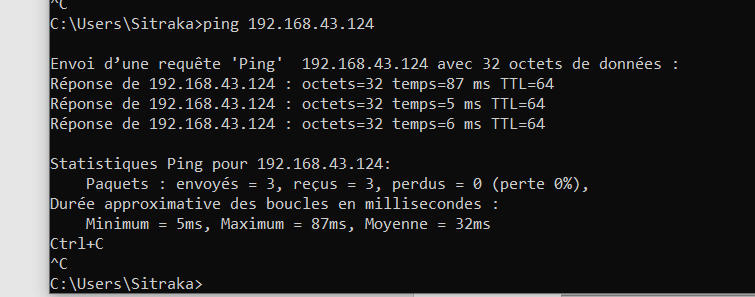
* Tester :

Pour la configuration dynamique

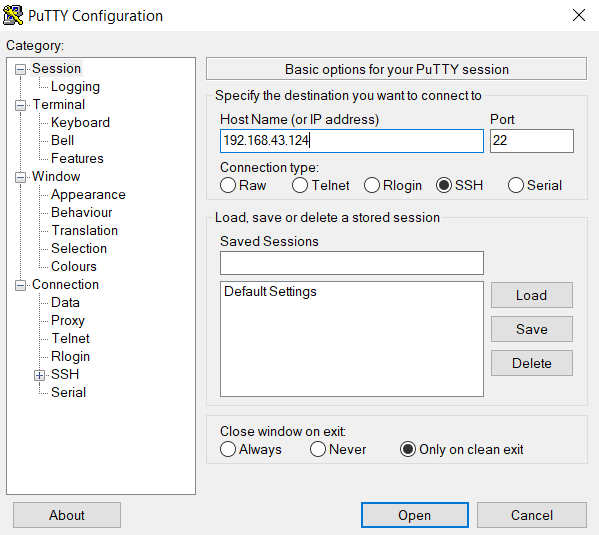
**ping 192.168.43.124**

**Pour la configuration statique**

**ping 192.168.43.205**

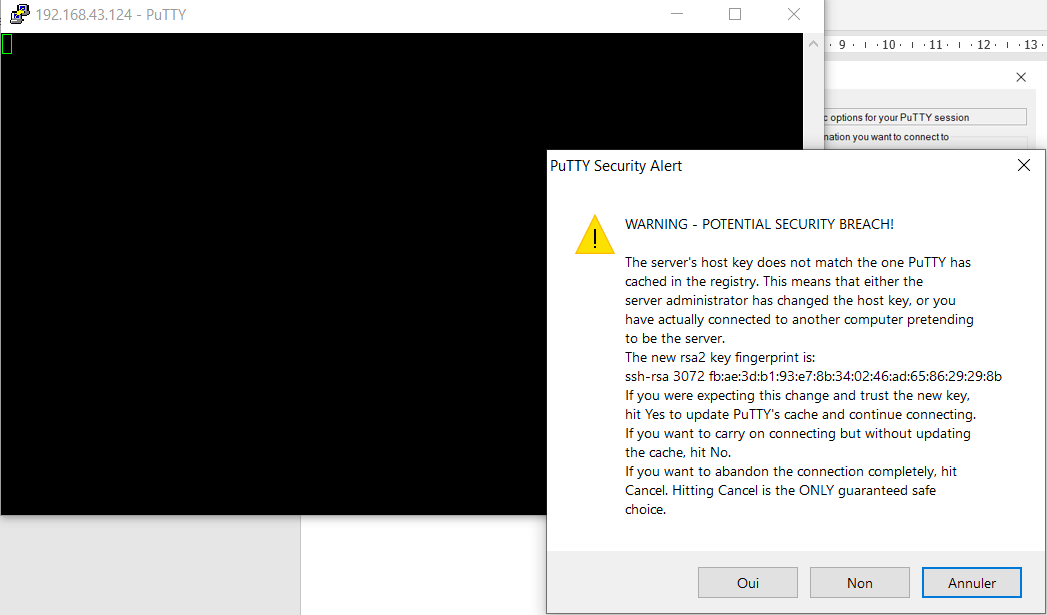


* Installer putty puis entrer l’adresse ip et accéder via ssh



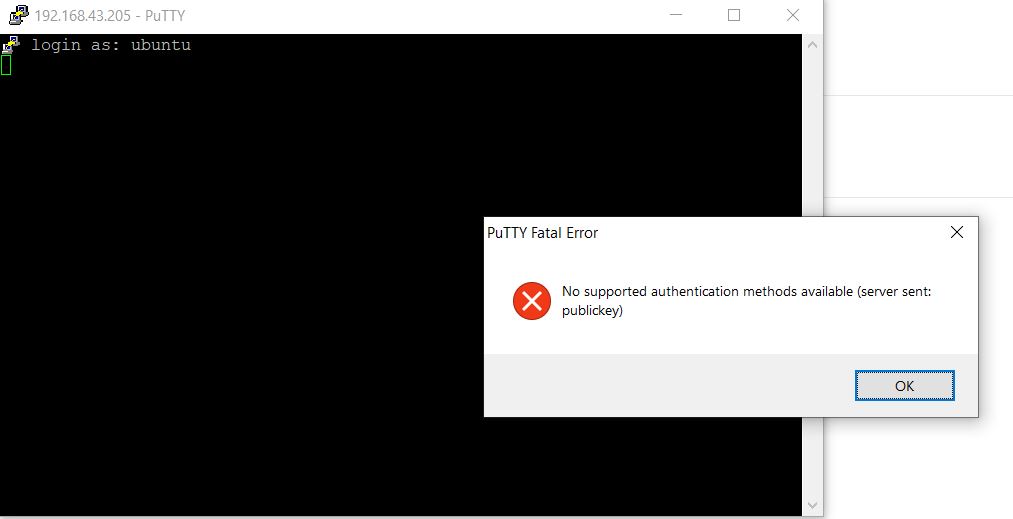
Pour le cas de l’adressage statique : tapez 192.168.43.205

* Cliquez sur OK



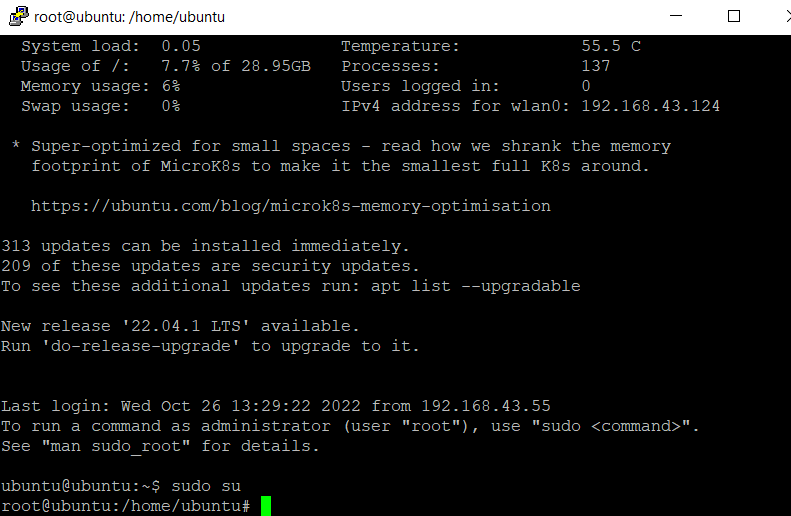
* Tapez comme login : ubuntu et mdp : ubuntu puis renouvelez-en (imsicatcher ou ubuntupi) (putty se ferme automatiquement)

*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

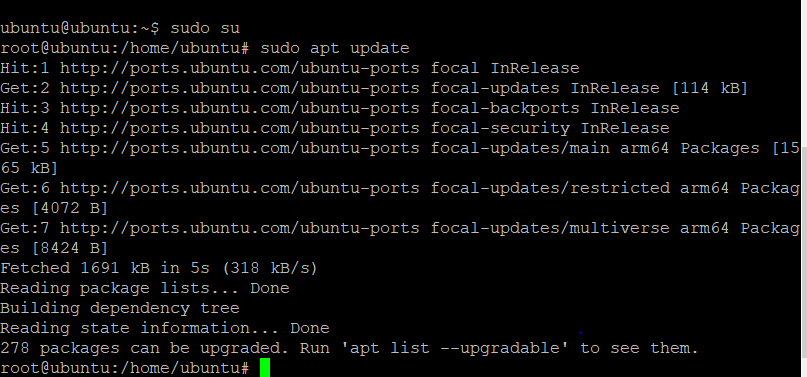


Etape 2 : Installation Desktop xfce et tiger VNC

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**

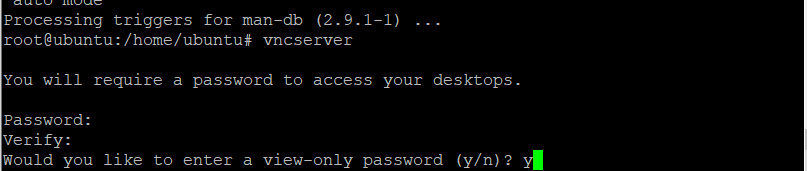
**sudo apt-get install xfce4 xfce4-goodies**

**# INSTEAD : sudo apt-get install ubuntu-desktop-minimal**

**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*

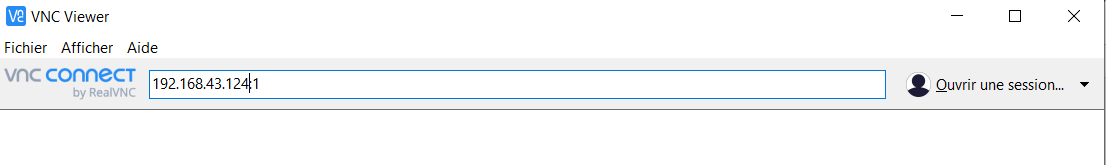
*x-window-manager &*

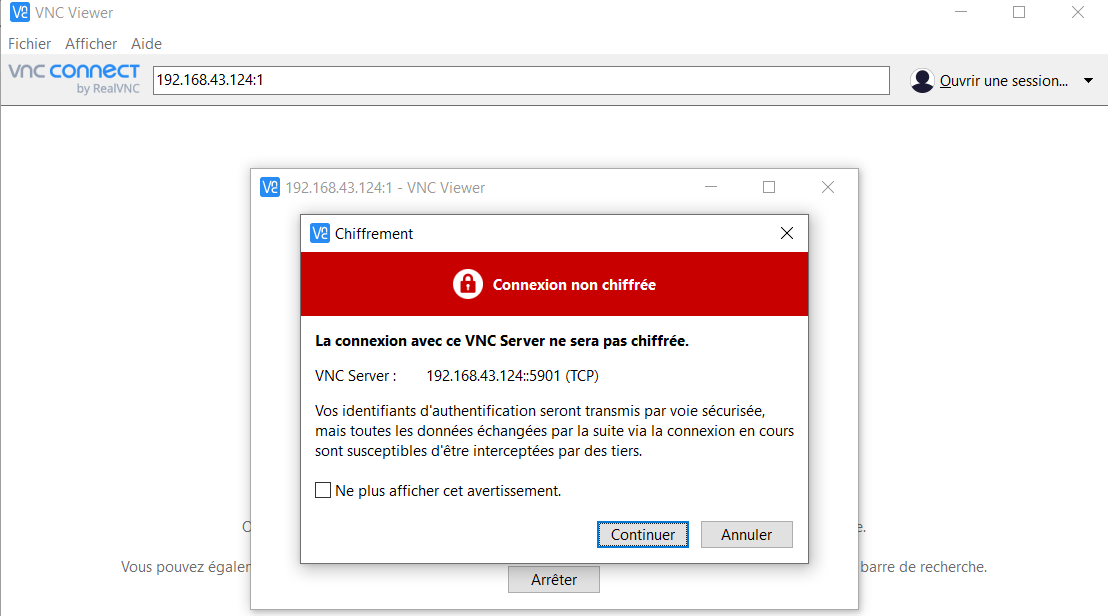
**chmod +x /root/.vnc/xstartup**

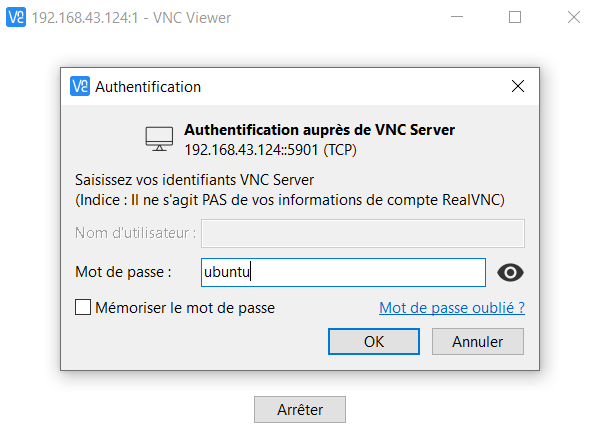
**vncserver -localhost no :1**

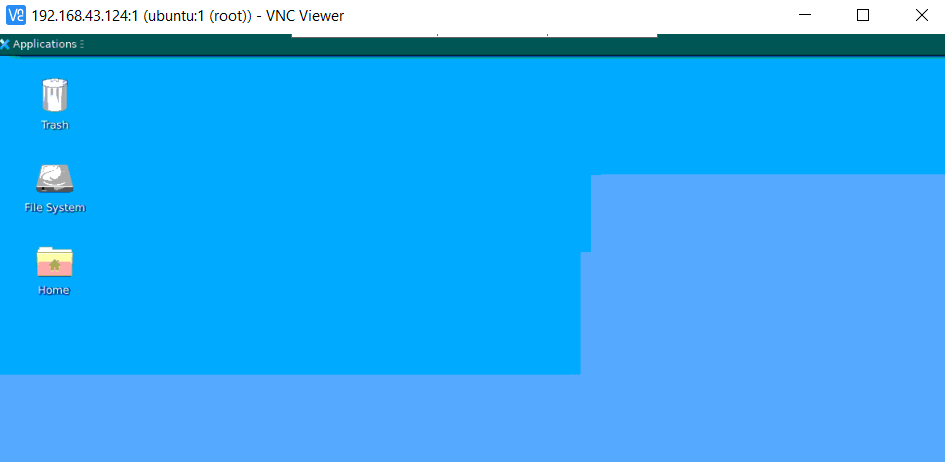
* Telecharger puis installer vncviewer sous windows, puis tapez addresseip:1

Dans mons cas 192.168.43.124:1









* 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*

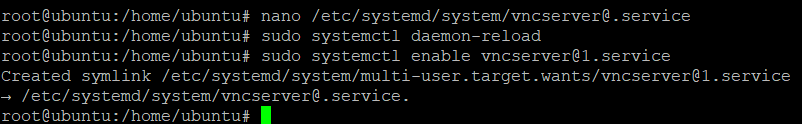
*ExecStop=/usr/bin/vncserver -kill :%i*

*[Install]*

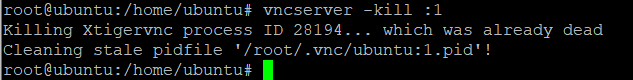
*WantedBy=multi-user.target*

**sudo systemctl daemon-reload**

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

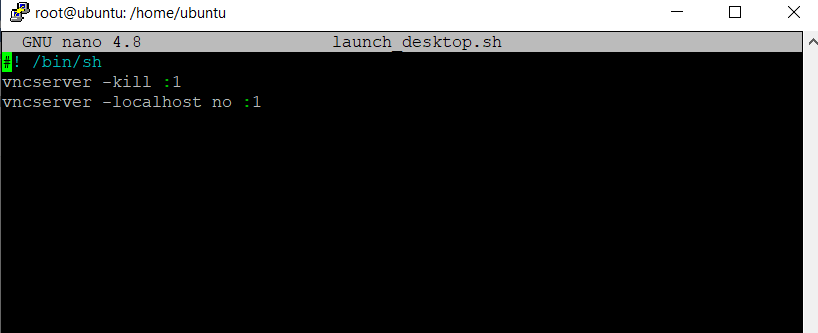


**vncserver -kill :1**



**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

Etape 4 : Installation de l’IMSI-Catcher

Suivre le tutoriel de Bastien Baranoff (j’ai changé /opt/IMSI\_Catcher en /root/Desktop/IMSI\_Catcher) :

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

**apt-get install oss-compat**

**modprobe snd-pcm-oss**

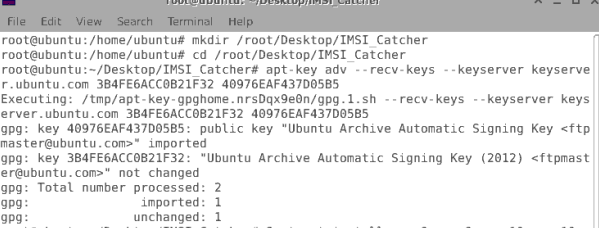
**modprobe snd-mixer-oss**

**mkdir /root/Desktop/IMSI\_Catcher**

**cd /root/Desktop/IMSI\_Catcher**

**(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/

**apt remove texinfo**

**cd /root/Desktop/IMSI\_Catcher/**

**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**

CREATION COMPTE: [imsicatcher.bastienbaranoff@gmail.com](mailto:imsicatcher.bastienbaranoff@gmail.com)

Mdp : IMSIBB9874563210

User name github : ImsicatcherBastienbaranoff

UTILISER GCC-9

**update-alternatives --set gcc /usr/bin/gcc-9**

**cd /root/Desktop/IMSI\_Catcher**

Telecharger gnuarm\_for\_ubuntu : <https://drive.google.com/drive/folders/11CPg-WdmBTo9DCexn_FVHiQ7yFOM6Z11?usp=share_link>

**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 :







Le path se trouve dans : /root/Desktop/IMSI\_Catcher/gnuarm/install/bin

**(commande en un seul ligne)**

**export PATH=$PATH:/root/Desktop/IMSI\_Catcher/gnuarm/install/bin**

**cd /root/Desktop/IMSI\_Catcher**

**git clone git://git.osmocom.org/libosmocore.git**

**cd libosmocore**

**git checkout 1.3.0**

**cd ..**

**apt-get install zip**

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

**cd libosmocore**

**autoreconf -i**

**./configure**

**make -j$(nproc)**

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

**ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**git clone git://git.osmocom.org/libosmo-dsp.git**

**zip -r libosmo-dsp\_for\_ubuntu.zip libosmo-dsp/**

**cd libosmo-dsp**

**autoreconf -i**

**./configure**

**make -j$(nproc)**

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

**ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

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

**cd trx**

**git checkout jolly/testing**

**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**

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

*# adding this last one*

*#deb http://old-releases.ubuntu.com/ubuntu/ zesty main*

*#deb-src http://old-releases.ubuntu.com/ubuntu/ zesty main*

**#apt update**

**#apt-get install python-minimal**

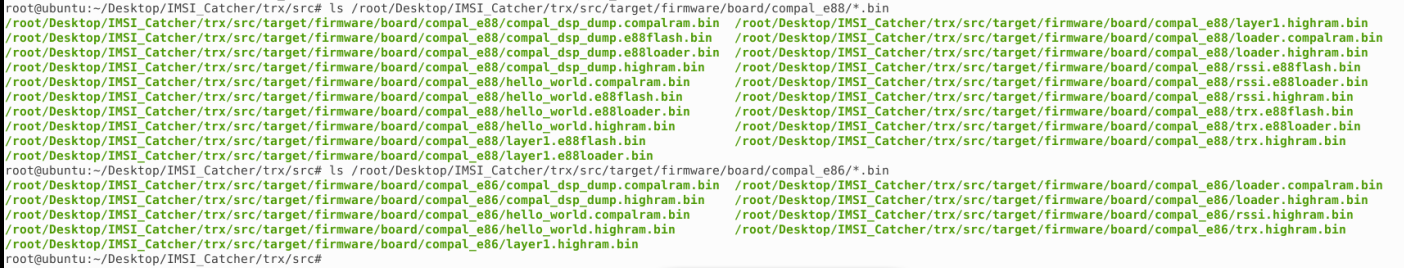
REMARQUES : PAS BESOIN DE CE COMMANDE

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

INSTALLATION DE TRX :

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

AFFICHAGE DES FICHIERS BIN



**#apt remove python-minimal**

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

#Changer en :

*# adding this last one*

*# deb http://old-releases.ubuntu.com/ubuntu/ zesty main*

*# deb-src http://old-releases.ubuntu.com/ubuntu/ zesty main*

**#apt update**

**apt-get install -y libortp-dev**

**cd /root/Desktop/IMSI\_Catcher**

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

**cd libosmo-abis**

**git checkout 0.8.1**

**cd ..**

**zip -r libosmo-abis\_for\_ubuntu.zip libosmo-abis/**

**cd libosmo-abis**

**(commande en un seul ligne)**

**autoreconf -fi && ./configure --disable-dahdi && make -j$(nproc) && make -j$(nproc) install &&ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**git clone https://github.com/osmocom/libosmo-netif**

**cd libosmo-netif/**

**git checkout 0.7.0**

**cd ..**

**zip -r libosmo-netif\_for\_ubuntu.zip libosmo-netif/**

**cd libosmo-netif**

**(commande en un seul ligne)**

**autoreconf -fi && ./configure && make -j$(nproc) && make -j$(nproc) install && ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**git clone** [**https://github.com/osmocom/openbsc**](https://github.com/osmocom/openbsc)

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

**cd openbsc/openbsc**

**(commande en un seul ligne )**

**autoreconf -fi && ./configure --with-lms && make -j$(nproc) && make -j$(nproc) install && ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**git clone https://github.com/osmocom/osmo-bts**

**cd osmo-bts**

**git checkout 0.8.1**

**cd ..**

**zip -r osmo-bts\_for\_ubuntu.zip osmo-bts/**

**cd osmo-bts**

**(commande en un seul ligne )**

**autoreconf -fi && ./configure --enable-trx && make -j$(nproc) && make -j$(nproc) install && ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**(commande en un seul ligne )**

**wget https://github.com/bbaranoff/telco\_install\_sh/raw/main/opencore-amr-0.1.5.tar.gz**

**tar xvzf opencore-amr-0.1.5.tar.gz**

**cd opencore-amr-0.1.5**

**./configure**

**make -j$(nproc)**

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

**ldconfig**

**cd /lib/modules/$(uname -r)/build/certs**

**(commande en un seul ligne )**

**openssl req -new -x509 -newkey rsa:2048 -keyout signing\_key.pem -outform DER -out signing\_key.x509 -nodes -subj "/CN=Owner/"**

**cd /root/Desktop/IMSI\_Catcher**

**git clone** [**https://github.com/isdn4linux/mISDN**](https://github.com/isdn4linux/mISDN)

**cd mISDN**

**(commande en un seul ligne )**

**wget https://raw.githubusercontent.com/bbaranoff/PImpMyPi/main/octvqe.patch**

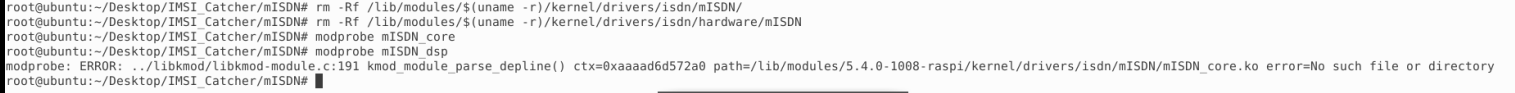
**cd ..**

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

**cd mISDN**

**rm -Rf /lib/modules/$(uname -r)/kernel/drivers/isdn/hardware/mISDN**

**rm -Rf /lib/modules/$(uname -r)/kernel/drivers/isdn/mISDN/**



**(commande en un seul ligne )**

**cp /boot/System.map-$(uname -r) /usr/src/linux-headers-$(uname -r)/System.map**

**(commande en un seul ligne )**

**ln -s /lib/modules/$(uname -r)/build /lib/modules/$(uname -r)/source**

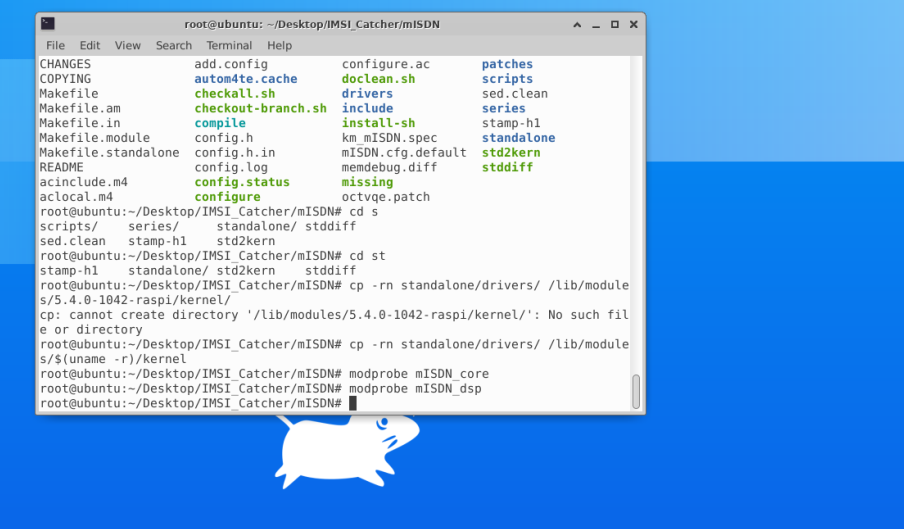
**aclocal && automake --add-missing**

**./configure**

**patch -p0 < octvqe.patch**

**make modules**

**cp -rn standalone/drivers/ /lib/modules/$(uname -r)/kernel**



*Rémarques :*

*NE PAS UTILISER CLASSIQUE COMPILATION SINON RPI CRASH*

*cp standalone/drivers/isdn/mISDN/modules.order /usr/src/linux-headers-$(uname -r)*

*cp -rn /usr/lib/modules/$(uname -r)/. /usr/src/linux-headers-$(uname -r)*

*make -j$(nproc) modules\_install*

*depmod -a*

**update-alternatives --set gcc /usr/bin/gcc-7**

**apt install bison flex -y**

**cd /root/Desktop/IMSI\_Catcher**

**git clone** [**https://github.com/isdn4linux/mISDNuser**](https://github.com/isdn4linux/mISDNuser)

**zip -r mISDNuser\_for\_ubuntu.zip mISDNuser**

**cd mISDNuser**

**make -j$(nproc)**

**./configure**

**make -j$(nproc)**

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

**ldconfig**

**cd example**

**./configure**

**make -j$(nproc)**

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

**ldconfig**

**cd /root/Desktop/IMSI\_Catcher**

**(commande en un seul ligne )**

**wget** [**http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-11.25.3.tar.gz**](http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-11.25.3.tar.gz)

**tar zxvf asterisk-11.25.3.tar.gz**

**cd asterisk-11.25.3**

**apt-get install libncurses-dev libxml2-dev**

**(commande en un seul ligne )**

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/tcptls.patch**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/tcptls.patch)

**patch -p1 < tcptls.patch**

**cd ..**

**zip -r asterisk-11.25.3\_for\_ubuntu.zip asterisk-11.25.3**

**cd asterisk-11.25.3**

**./configure**

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

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

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

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

**ldconfig**

**update-alternatives --set gcc /usr/bin/gcc-5**

**cd /root/Desktop/IMSI\_Catcher**

**git clone** [**https://github.com/fairwaves/lcr**](https://github.com/fairwaves/lcr)

**cd lcr**

**(commande en un seul ligne)**

**wget** [**https://raw.githubusercontent.com/bbaranoff/PImpMyPi/main/ast\_lcr.patch**](https://raw.githubusercontent.com/bbaranoff/PImpMyPi/main/ast_lcr.patch)

**cd ..**

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

**cd lcr**

**autoreconf -i**

**patch -p0 < ast\_lcr.patch**

**./configure --with-sip --with-gsm-bs --with-gsm-ms --with-asterisk --with-misdn**

**make -j$(nproc)**

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

**ldconfig**

**cp chan\_lcr.so /usr/lib/asterisk/modules/**

*(possible erreur network-config)*

*#apt-get autoremove python2.7-minimal*

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

*#apt-get install oss-compat*

**modprobe snd-pcm-oss**

**modprobe snd-pcm**

**modprobe snd-mixer-oss**

**modprobe mISDN\_core**

**modprobe mISDN\_dsp**

REDEMARREZ L’ORDINATEUR

**modprobe snd-pcm-oss**

**modprobe snd-pcm**

**modprobe snd-mixer-oss**

**modprobe mISDN\_core**

**modprobe mISDN\_dsp**

**cd /root/Desktop/IMSI\_Catcher**

**mousepad modprobe\_all.sh**

*#! /bin/bash*

*modprobe snd-pcm-oss*

*modprobe snd-pcm*

*modprobe snd-mixer-oss*

*modprobe mISDN\_core*

*modprobe mISDN\_dsp*

*echo "all modprobe OK"*

**bash modprobe\_all.sh**

**rm -rf /usr/local/etc/lcr**

**mkdir -p /usr/local/etc/**

**git clone** [**https://github.com/bbaranoff/lcr\_conf**](https://github.com/bbaranoff/lcr_conf%20)

**git clone** [**https://github.com/bbaranoff/lcr\_conf /usr/local/etc/lcr/**](https://github.com/bbaranoff/lcr_conf%20%20/usr/local/etc/lcr/)

**sudo chmod 755 /usr/local/etc/lcr**

**sudo chmod 644 /usr/local/etc/lcr/\***

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/sip.conf**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/sip.conf)

**(commande en un seul ligne)**

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/extensions.conf**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/extensions.conf)

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/openbsc.cfg**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/openbsc.cfg)

**cd /etc/asterisk**

**mv sip.conf sip.conf.bak**

**mv extensions.conf extensions.conf.bak**

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/sip.conf**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/sip.conf)

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/extensions.conf**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/extensions.conf)

**mkdir /root/Desktop/IMSI\_Catcher/nitb**

**cd /root/Desktop/IMSI\_Catcher/nitb**

**wget** [**https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/openbsc.cfg**](https://raw.githubusercontent.com/bbaranoff/telco_install_sh/main/openbsc.cfg)

**wget https://raw.githubusercontent.com/bbaranoff/telco\_install\_sh/main/nitb.sh**

**chmod +x nitb.sh**

**cd /root/Desktop/IMSI\_Catcher**

**git clone** [**https://github.com/bbaranoff/telco\_install\_sh**](https://github.com/bbaranoff/telco_install_sh)

**zip -r telco\_install\_sh\_for\_ubuntu.zip telco\_install\_sh/**

**apt-get install terminator**

**cd /root/Desktop/IMSI\_Catcher**

**chmod 777 -R launch\_all\_bts/\***

**update-alternatives --config x-terminal-emulator**

**apt-get install wireshark**

SUIVRE LE README DANS

In https://github.com/bbaranoff/telco\_install\_sh

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

OSMOCOM KARLI

* git clone git://git.osmocom.org/osmocom-bb.git
* cd osmocom-bb
* git checkout 4f677e6ba8434dab376495cd996d140548fa6e93
* mousepad src/target/firmware/Makefile
* DECOMMENTER : CFLAGS += -DCONFIG\_TX\_ENABLE/CFLAGS += -
* cd ..
* zip -r osmocom-bb\_for\_ubuntu.zip osmocom-bb/
* cd osmocom-bb/src

DO NOT USE make -j

* make

apt 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 gcc gcc-arm-none-eabi doxygen libtalloc-dev libpcsclite-dev libusb-1.0-0-dev