SENSITIVE METADATA RISK – CLIENT SETUP

1 System architecture

1.1 Component

Component	IP address
Client Laptop (Windows OS/Ubuntu OS)	CLIENT_IP
MITM Proxy Server	SERVER_IP
Virtual Phone (AVD)	Running on Client's Laptop

1.2 System architecture

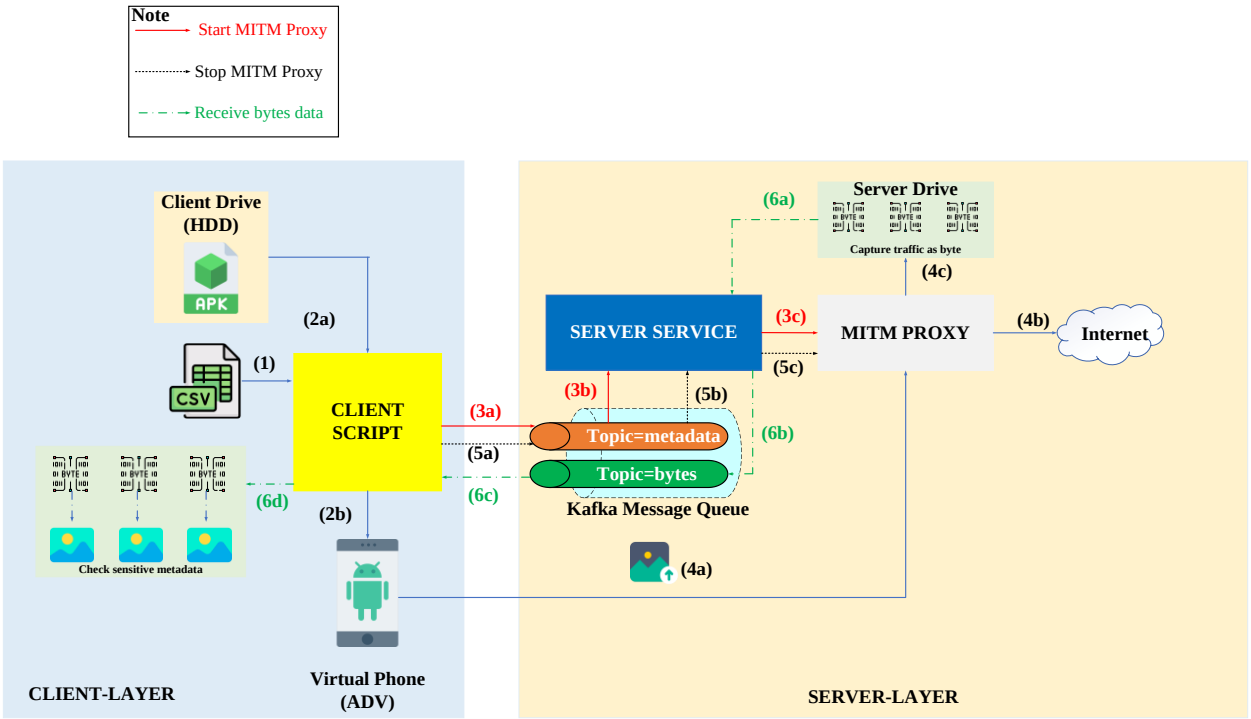


Figure 1: System architecture

The system includes two layers: a **client layer** and a **server layer**.

In the client layer, we have two components: client script and virtual phone (ADV).

We have three server layer components: the server service, the Man-in-the-middle proxy (MITM Proxy), and the Kafka message Queue. In Kafka, we will have two topics auto-created by the server service. The **metadata** topic is used to control (start/stop) MITM Proxy, and the **bytes** topic is used to receive the traffic captured by MITM Proxy.

Firstly, the client script reads APK information from the CSV file (1). The APK information includes the APK name and APK package. Then, the client script uses APK's information to query APK in the drive and install it on ADV (2a-2b).

After installation, the client script will auto-launch the app and push the "start" command to Kafka on topic=metadata. This command receives by the server service and starts the MITM Proxy (3a-3b-3c).

The user upload/send/share the image with sensitive metadata via the app to the internet through MITM Proxy (4a-4b). At this time, MITM Proxy will capture all upload traffic from the app and save it as bytes (4c).

When the user finishes the test, they uninstall the app. This action triggers the client script to send the "stop" command to Kafka on topic=metadata (5a-5b-5c).

After the MITM Proxy stop, the server service reads all captured traffic bytes and sends them to the client script on Kafka topic=bytes (6a-6b-6c).

When the client script receives all messages, it will convert these bytes to images and check if they contain sensitive metadata (6d).

1.1 Demo video

Install the client on Ubuntu OS: https://youtu.be/feTJm_ykjYQ

Install the client on Windows OS: https://youtu.be/rUCk2TM58Z0

Running test code (whole system): https://youtu.be/Xj_IluWAnMw

2 Install Ubuntu dependencies

server_machine:~# apt-get update

server_machine:~# apt-get upgrade

2.1 git

server_machine:~# apt-get install git

server_machine:~# git --version

```
root@ubuntu-desktop:~# git --version
git version 2.34.1
```

2.2 Java version 11

server_machine:~# apt install openjdk-11-jdk

server_machine:~# java -version

```
root@ubuntu-desktop:~# java -version
openjdk version "11.0.19" 2023-04-18
OpenJDK Runtime Environment (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1, mixed mode, sharing)
root@ubuntu-desktop:~#
```

2.3 Install snapd

server_machine:~# apt install snapd

2.4 Install net-tools

server_machine:~# apt-get install net-tools

3 Install a Virtual Phone (on the client)

3.1 Install Android Studio on Client Windows OS

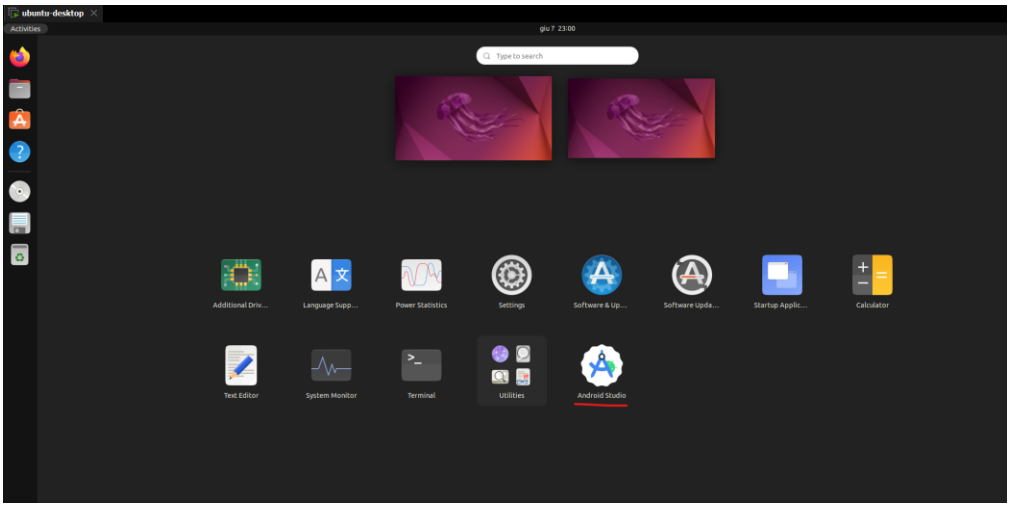
Download Android Studio https://developer.android.com/studio

Install as normal software

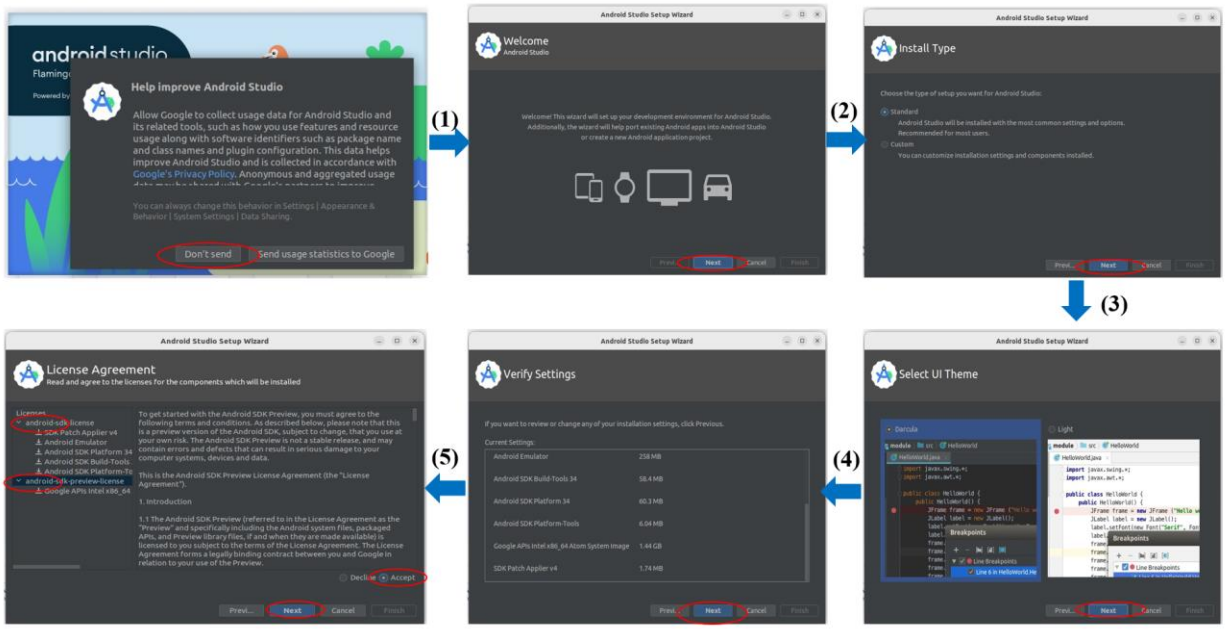
3.2 Install Android Studio on Client Ubuntu OS (desktop version)

ubuntu_client:~# snap install android-studio --classic

```
root@ubuntu-desktop:~# snap install android-studio --classic
android-studio 2022.2.1.19 from Snapcrafters installed
```

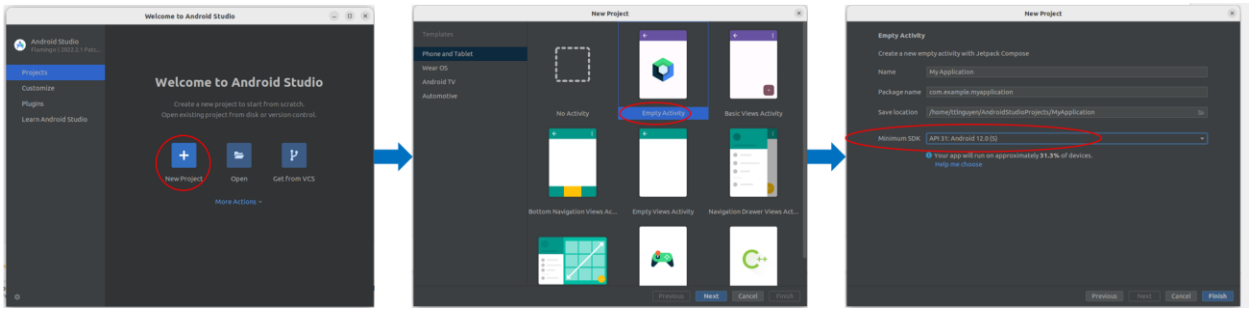


Config



3.3 Create new project

Note: do the same for both Windows OS and Ubuntu OS

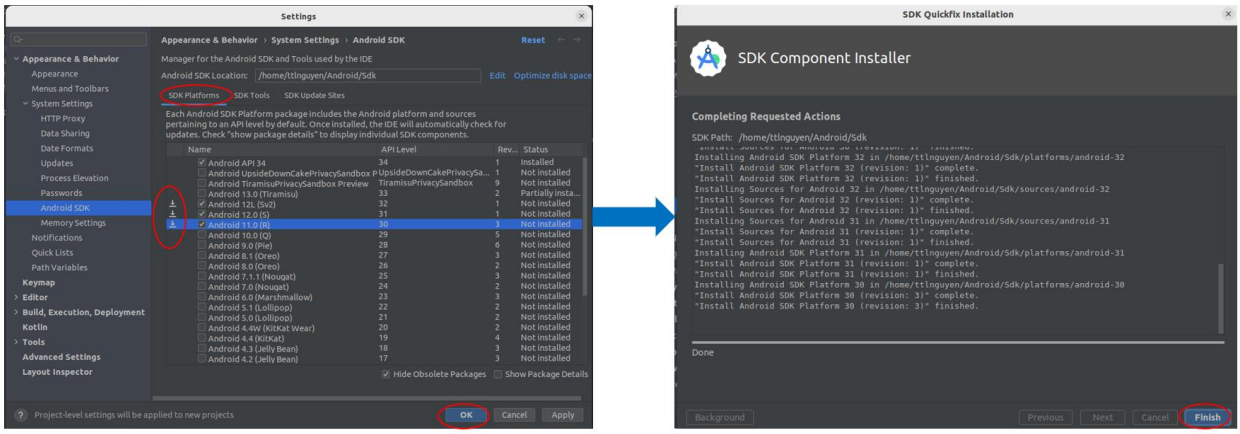


3.4 Install SDK

Note: do the same for both Windows OS and Ubuntu OS

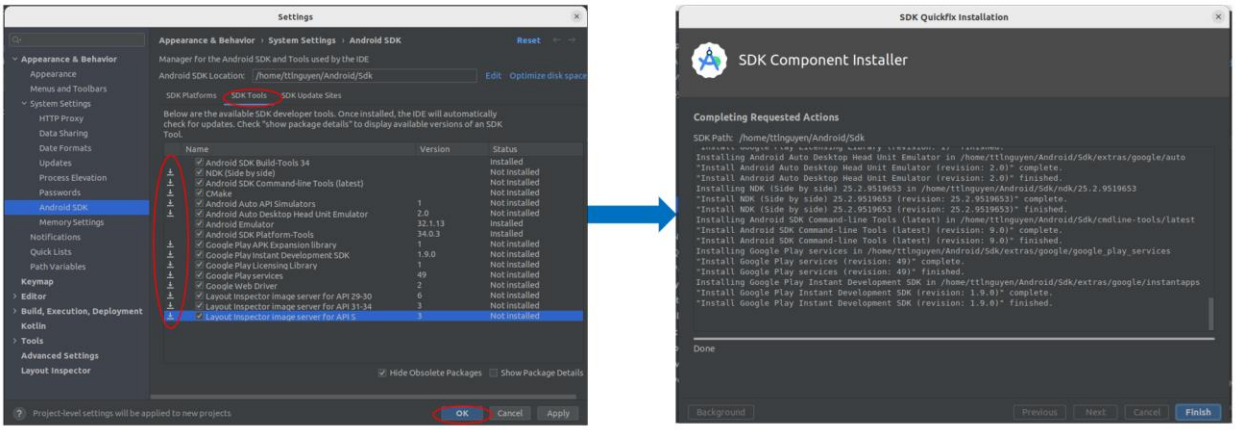
3.4.1 SDK Platforms

Tools → SDK Manager → SDK Platforms



3.4.2 SDK Tools

Tools → SDK Manager → SDK Tools



3.5 Setup adb

Note: do only for Ubuntu OS

Check adb path

ubuntu_client:~# ls -al /home/ttlnguyen/Android/Sdk/platform-tools/

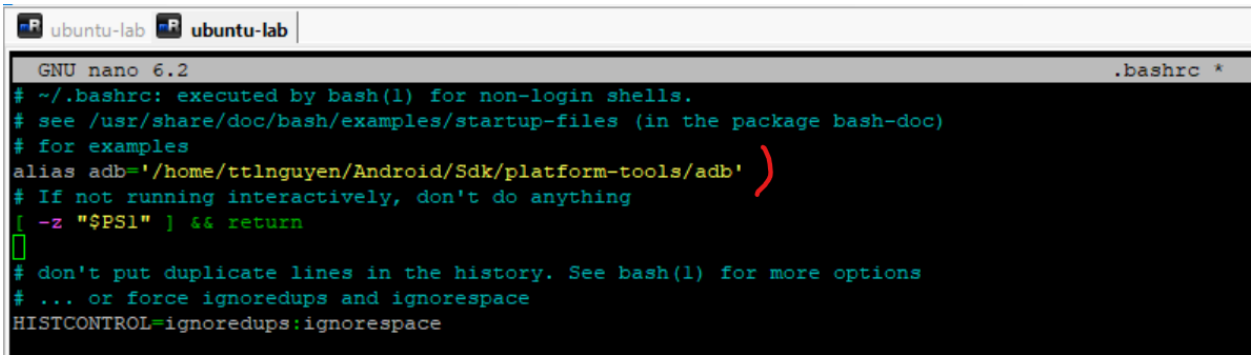
```
root@distarossi-ttln:~# ls -al /home/ttlnguyen/Android/Sdk/platform-tools/
total 14436
drwxrwxr-x  3 ttlnguyen ttlnguyen    4096 giu  8 10:58 .
drwxrwxr-x 19 ttlnguyen ttlnguyen    4096 giu  8 11:25 ..
-rwxrwxr-x  1 ttlnguyen ttlnguyen 7890744 giu  8 10:58 adb
-rwxrwxr-x  1 ttlnguyen ttlnguyen   57216 giu  8 10:58 dmtracedump
-rwxrwxr-x  1 ttlnguyen ttlnguyen  305368 giu  8 10:58 etc1tool
-rwxrwxr-x  1 ttlnguyen ttlnguyen 2502240 giu  8 10:58 fastboot
-rwxrwxr-x  1 ttlnguyen ttlnguyen   13376 giu  8 10:58 hprof-conv
drwxrwxr-x  2 ttlnguyen ttlnguyen    4096 giu  8 10:58 lib64
-rwxrwxr-x  1 ttlnguyen ttlnguyen  261896 giu  8 10:58 make_f2fs
-rwxrwxr-x  1 ttlnguyen ttlnguyen  261880 giu  8 10:58 make_f2fs_casefold
-rwxrwxr-x  1 ttlnguyen ttlnguyen  874624 giu  8 10:58 mke2fs
-rw-rw-r--  1 ttlnguyen ttlnguyen    1157 giu  8 10:58 mke2fs.conf
-rw-rw-r--  1 ttlnguyen ttlnguyen 1070930 giu  8 10:58 NOTICE.txt
-rw-rw-r--  1 ttlnguyen ttlnguyen   18335 giu  8 10:58 package.xml
-rw-rw-r--  1 ttlnguyen ttlnguyen      37 giu  8 10:58 source.properties
-rwxrwxr-x  1 ttlnguyen ttlnguyen 1490224 giu  8 10:58 sqlite3
root@distarossi-ttln:~#
```

Create link

ubuntu_client:~# nano ~/.bashrc

ADD

alias adb='/home/ttlnguyen/Android/Sdk/platform-tools/adb'



ubuntu_client:~# source ~/.bashrc

Check

root@ubuntu-desktop:~# adb devices

```
root@ubuntu-desktop:~# adb devices
List of devices attached

root@ubuntu-desktop:~#
```

Note: there is no AVD now!

Create $PATH environment

ubuntu_client:~# nano ~/.profile

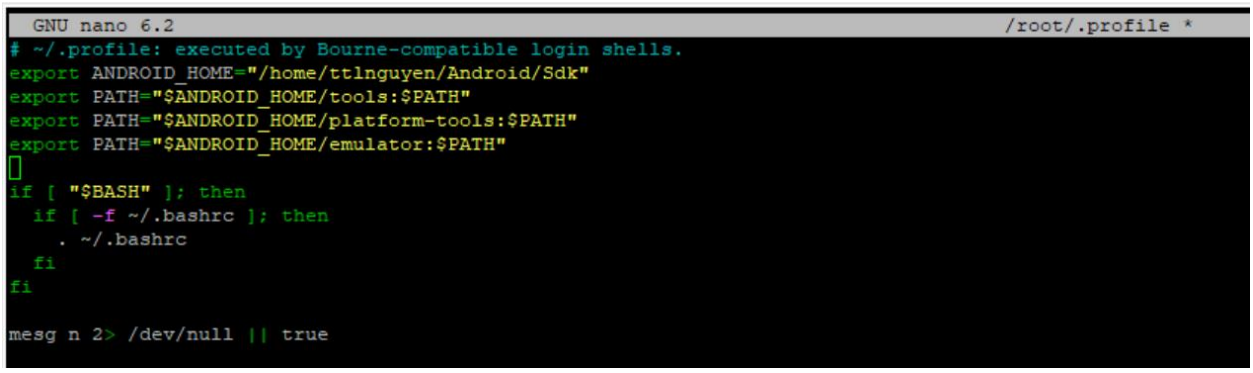
ADD

export ANDROID_HOME="/home/ttlnguyen/Android/Sdk"

export PATH="$ANDROID_HOME/tools:$PATH"

export PATH="$ANDROID_HOME/platform-tools:$PATH"

export PATH="$ANDROID_HOME/emulator:$PATH"



ubuntu_client:~# source ~/.profile

Check $PATH

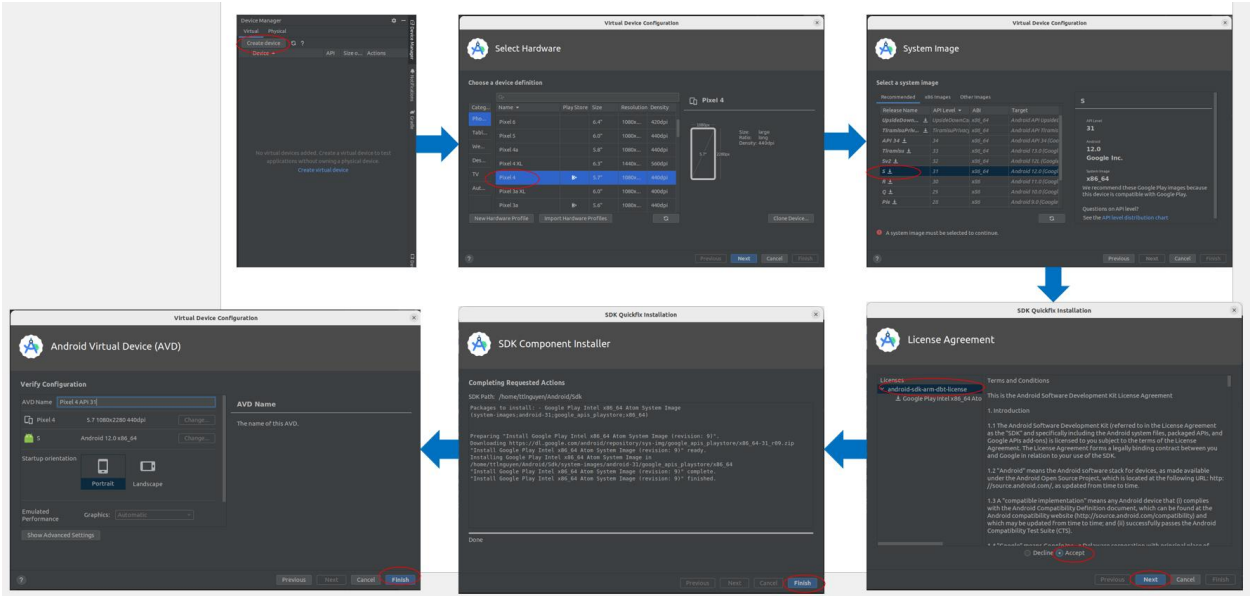
ubuntu_client:~# echo $PATH

3.6 Create AVD (Virtual Phone)

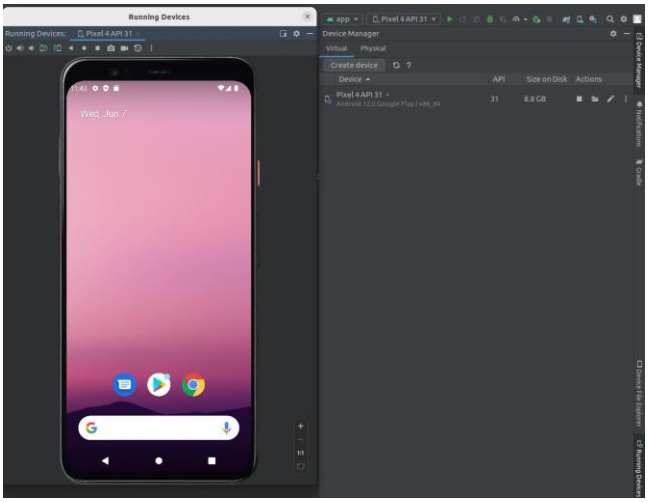
Note: do the same for both Windows OS and Ubuntu OS

Chose Pixel 4

Chose Android version 12 (API 31)



Run AVD



Check

ubuntu_client:~# adb devices

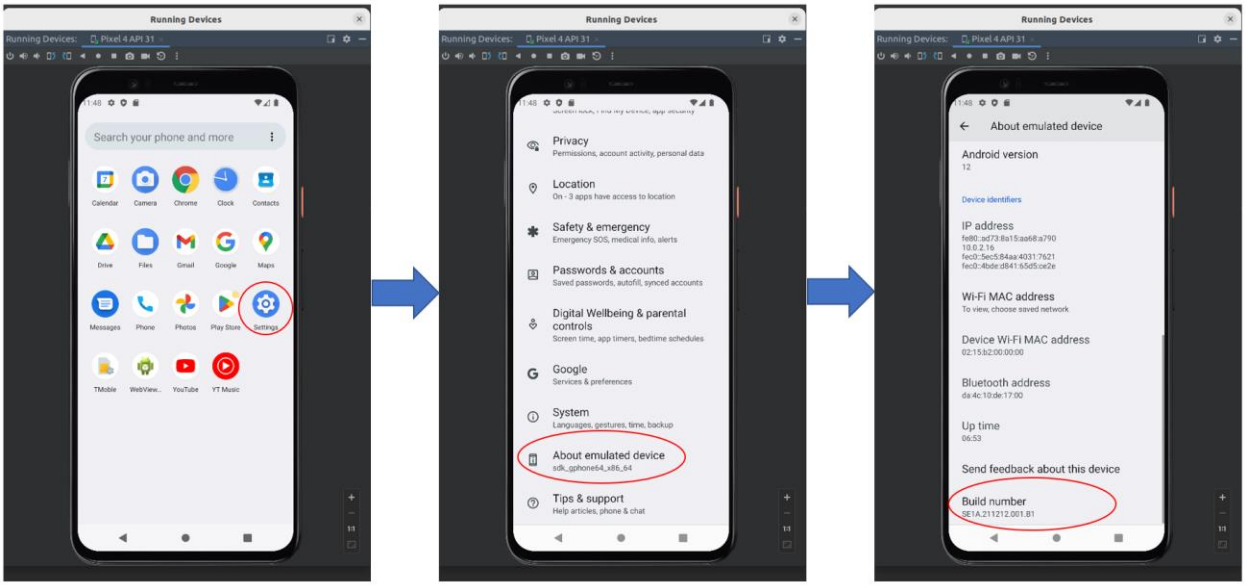
```
root@lam-G550JK:~# adb devices
List of devices attached
emulator-5554   device

root@lam-G550JK:~#
```

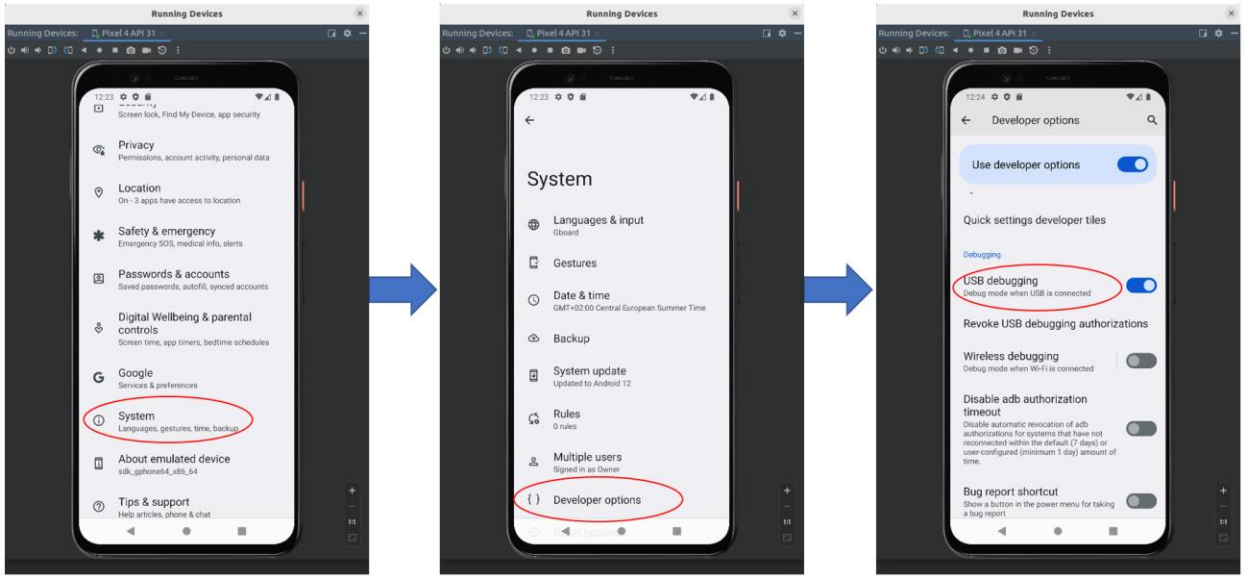
3.7 Root AVD for Ubuntu OS

3.7.1 Enable Developer Mode

Settings → About emulated device → Build number (click 7 times)



3.7.2 Enable USB Debugging



3.7.3 Download root script

ubuntu_client:~# git clone https://github.com/thanhlam2110/rootAVD-1.git

ubuntu_client:~# cd rootAVD-1/

ubuntu_client:~/rootAVD-1# chmod -R 777 *

3.7.4 Install Magisk

Check ramdisk.img path

ubuntu_client:~# ls -al /home/lam/Android/Sdk/system-images/android-31/google_apis_playstore/x86_64

```
root@lam-G550JK:~# ls -al /home/lam/Android/Sdk/system-images/android-31/google_apis_playstore/x86_64
total 3006312
drwxrwxr-x 3 lam  lam        4096 giu  3 23:07 .
drwxrwxr-x 3 lam  lam        4096 giu  3 22:40 ..
-rw-rw-r-- 1 lam  lam         659 giu  3 22:40 advancedFeatures.ini
-rw-rw-r-- 1 lam  lam        4341 giu  3 22:40 build.prop
drwxrwxr-x 3 lam  lam        4096 giu  3 22:40 data
-rw-rw-r-- 1 lam  lam    18874368 giu  3 22:40 encryptionkey.img
-rw-rw-r-- 1 lam  lam    22839216 giu  3 22:40 kernel-ranchu
-rw-rw-r-- 1 lam  lam    12639126 giu  3 22:40 NOTICE.txt
-rw-rw-r-- 1 lam  lam       19143 giu  3 22:40 package.xml
-rw-r--r-- 1 root root    3848074 giu  3 23:07 ramdisk.img
-rw-r--r-- 1 root root    3441343 giu  3 23:06 ramdisk.img.backup
-rw-rw-r-- 1 lam  lam         302 giu  3 22:40 source.properties
-rw-rw-r-- 1 lam  lam  2864709632 giu  3 22:40 system.img
-rw-rw-r-- 1 lam  lam     1048576 giu  3 22:40 userdata.img
-rw-rw-r-- 1 lam  lam   150994944 giu  3 22:40 vendor.img
-rw-rw-r-- 1 lam  lam         356 giu  3 22:40 VerifiedBootParams.textproto
root@lam-G550JK:~#
```

ubuntu_client:~# cd rootAVD-1/

ubuntu_client:~/rootAVD-1# chmod -R 777 *

ubuntu_client:~/metadata-root-adv/rootAVD # ./rootAVD.sh /home/lam/Android/Sdk/system-images/android-31/google_apis_playstore/x86_64/ramdisk.img

Output log

[!] and we are NOT in an emulator shell

[*] Set Directorys

[-] Test if ADB SHELL is working

[-] In any AVD via ADB, you can execute code without root in /data/data/com.android.shell

[*] Cleaning up the ADB working space

[*] Creating the ADB working space

[-] Magisk installer Zip exists already

[*] Push Magisk.zip into /data/data/com.android.shell/Magisk

[-] ./Magisk.zip: 1 file pushed, 0 skipped. 190.7 MB/s (11278270 bytes in 0.056s)

[-] ramdisk.img Backup exists already

[*] Push ramdisk.img into /data/data/com.android.shell/Magisk/ramdisk.img

[-] /home/lam/Android/Sdk/system-images/android-31/google_apis_playstore/x86_64/ramdisk.img: 1 file pushed, 0 skipped. 224.4 MB/s (3848074 bytes in 0.016s)

[*] Push rootAVD.sh into /data/data/com.android.shell/Magisk

[-] rootAVD.sh: 1 file pushed, 0 skipped. 97.3 MB/s (76222 bytes in 0.001s)

[-] run the actually Boot/Ramdisk/Kernel Image Patch Script

[*] from Magisk by topjohnwu and modded by NewBit XDA

[!] We are in a ranchu emulator shell

[-] Api Level Arch Detect

[-] Device Platform is x64 only

[-] Device SDK API: 31

[-] First API Level: 31

[-] The AVD runs on Android 12

[-] Switch to the location of the script file

[*] Looking for an unzip binary

[-] unzip binary found

[*] Extracting busybox and Magisk.zip via unzip ...

[*] Finding a working Busybox Version

[!] Found a working Busybox Version

[!] BusyBox v1.34.1-Magisk (2022-03-22 04:11:29 PDT) multi-call binary.

[*] Move busybox from lib to workdir

[-] Checking AVDs Internet connection...

[!] AVD is online

[!] Checking available Magisk Versions

[?] Choose a Magisk Version to install and make it local

[s] (s)how all available Magisk Versions

[1] local stable '25.2' (ENTER)

[2] stable 26.1

[3] canary 47d2d4e3(26102)

[4] alpha 709f25f6-alpha(26101)

[-] You choose Magisk local stable Version '25.2'

[*] Re-Run rootAVD in Magisk Busybox STANDALONE (D)ASH

[-] We are now in Magisk Busybox STANDALONE (D)ASH

[*] rootAVD with Magisk '25.2' Installer

[-] Get Flags

[*] System-as-root, keep dm/avb-verity

[-] Encrypted data, keep forceencrypt

[*] RECOVERYMODE=false

[-] KEEPVERITY=true

[*] KEEPFORCEENCRYPT=true

[-] copy all x86_64 files from /data/data/com.android.shell/Magisk/lib/x86_64 to /data/data/com.android.shell/Magisk

[*] Detecting ramdisk.img compression

[!] Ramdisk.img uses lz4_legacy compression

[-] taken from shakalaca's MagiskOnEmulator/process.sh

[*] executing ramdisk splitting / extraction / repacking

[-] API level greater then 30

[*] Check if we need to repack ramdisk before patching ..

[*] After decompressing ramdisk.img, magiskboot will work

Detected format: [lz4_legacy]

[-] Checking ramdisk STATUS=1

[-] Magisk patched boot image detected

[*] Verifying Boot Image by its Kernel Release number:

[-] This AVD = 5.10.66-android12-9-00041-gfa9c9074531e-ab7914766

[-] Ramdisk = 5.10.66-android12-9-00223-gfa9c9074531e-ab7914766

[!] Ramdisk is probably NOT from this AVD

[*] repacking back to ramdisk.img format

[!] Rename Magisk.zip to Magisk.apk

[*] Pull ramdiskpatched4AVD.img into ramdisk.img

[-] /data/data/com.android.shell/Magisk/ramdiskpatched4AVD.img: 1 file pulled, 0 skipped. 208.5 MB/s (3848074 bytes in 0.018s)

[*] Pull Magisk.apk into

[-] /data/data/com.android.shell/Magisk/Magisk.apk: 1 file pulled, 0 skipped. 205.9 MB/s (11278270 bytes in 0.052s)

[-] Clean up the ADB working space

[-] Install all APKs placed in the Apps folder

[*] Trying to install Apps/Magisk.apk

[*] Performing Streamed Install

[*] Success

[-] Shut-Down & Reboot (Cold Boot Now) the AVD and see if it worked

[-] Root and Su with Magisk for Android Studio AVDs

[-] Trying to shut down the AVD

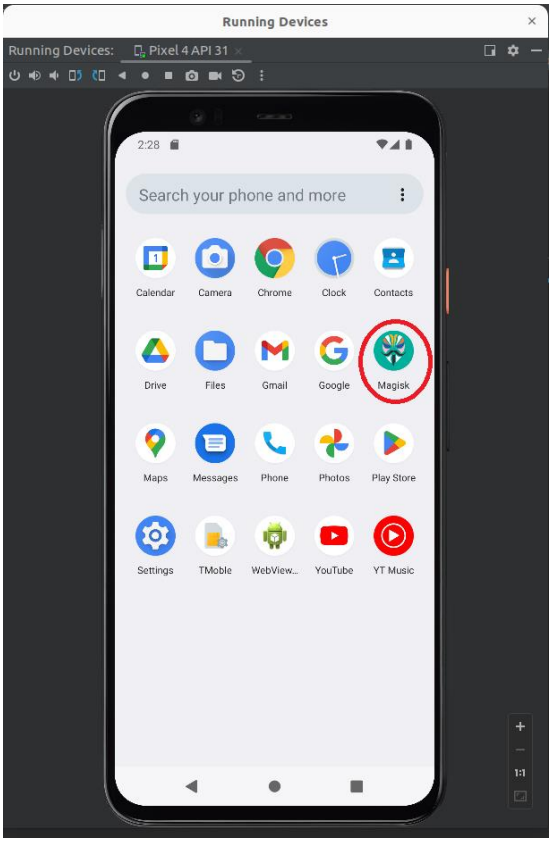
[!] If the AVD doesn't shut down, try it manually!

[-] Modded by NewBit XDA - Jan. 2021

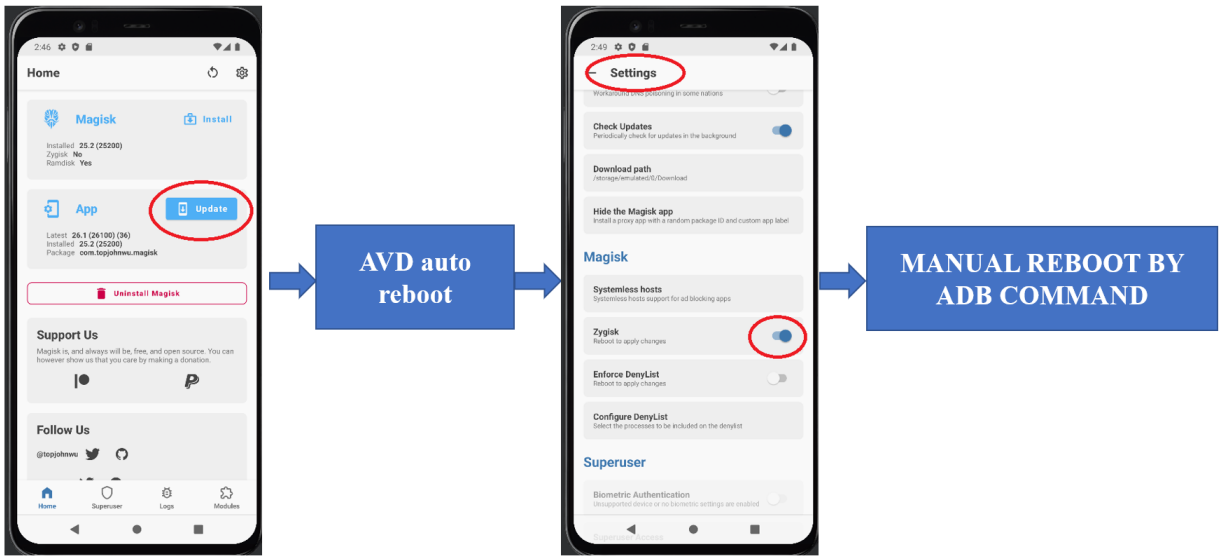
[!] Huge Credits and big Thanks to topjohnwu, shakalaca, vvb2060 and HuskyDG

```
[!] Checking available Magisk Versions
[?] Choose a Magisk Version to install and make it local
[s] (s)how all available Magisk Versions
[1] local stable '25.2' (ENTER)
[2] stable 26.1
[3] canary 47d2d4e3(26102)
[4] alpha 709f25f6-alpha(26101)
[-] You choose Magisk local stable Version '25.2'
[*] Re-Run rootAVD in Magisk Busybox STANDALONE (D)ASH
[-] We are now in Magisk Busybox STANDALONE (D)ASH
[*] rootAVD with Magisk '25.2' Installer
[-] Get Flags
[*] System-as-root, keep dm/avb-verity
[-] Encrypted data, keep forceencrypt
[*] RECOVERYMODE=false
[-] KEEPVERITY=true
[*] KEEPFORCEENCRYPT=true
[-] copy all x86_64 files from /data/data/com.android.shell/Magisk/lib/x86_64 to /data/data/com.android.shell/Magisk
[*] Detecting ramdisk.img compression
[!] Ramdisk.img uses lz4_legacy compression
[-] taken from shakalaca's MagiskOnEmulator/process.sh
[*] executing ramdisk splitting / extraction / repacking
[-] API level greater then 30
[*] Check if we need to repack ramdisk before patching ..
[*] After decompressing ramdisk.img, magiskboot will work
Detected format: [lz4_legacy]
[-] Checking ramdisk STATUS=1
[-] Magisk patched boot image detected
[*] Verifying Boot Image by its Kernel Release number:
[-] This AVD = 5.10.66-android12-9-00041-gfa9c9074531e-ab7914766
[-]  Ramdisk = 5.10.66-android12-9-00223-gfa9c9074531e-ab7914766
[!] Ramdisk is probably NOT from this AVD
[*] repacking back to ramdisk.img format
[!] Rename Magisk.zip to Magisk.apk
[*] Pull ramdiskpatched4AVD.img into ramdisk.img
[-] /data/data/com.android.shell/Magisk/ramdiskpatched4AVD.img: 1 file pulled, 0 skipped. 208.5 MB/s (3848074 bytes in 0.018s)
[*] Pull Magisk.apk into
[-] /data/data/com.android.shell/Magisk/Magisk.apk: 1 file pulled, 0 skipped. 205.9 MB/s (11278270 bytes in 0.052s)
[-] Clean up the ADB working space
[-] Install all APKs placed in the Apps folder
[*] Trying to install Apps/Magisk.apk
[*] Performing Streamed Install
[*] Success
[-] Shut-Down & Reboot (Cold Boot Now) the AVD and see if it worked
[-] Root and Su with Magisk for Android Studio AVDs
[-] Trying to shut down the AVD
[!] If the AVD doesn't shut down, try it manually!
[-] Modded by NewBit XDA - Jan. 2021
[!] Huge Credits and big Thanks to topjohnwu, shakalaca, vvb2060 and HuskyDG
root@lam-G550JK:~/root_android_phone/rootAVD-master#
```

Result



3.7.5 Config Root by Magisk



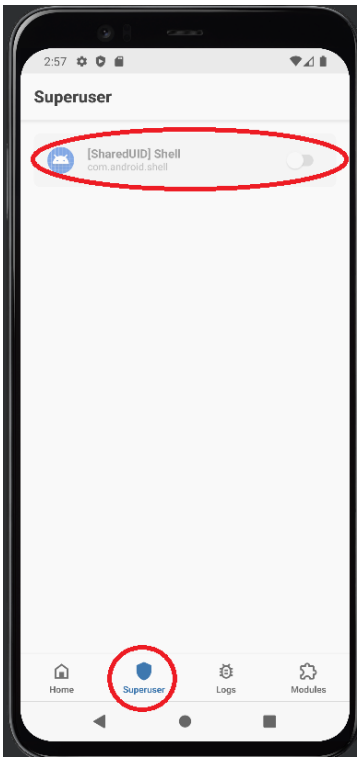
Reboot by adb command

ubuntu_client:~# adb reboot

3.7.6 Grant root user for adb shell

ubuntu_client:~# adb shell

emulator64_x86_64_arm64:/ $ su



emulator64_x86_64_arm64:/ # whoami

```
root@lam-G550JK:~# adb shell
emulator64_x86_64_arm64:/ $ su
emulator64_x86_64_arm64:/ # whoami
root
emulator64_x86_64_arm64:/ #
```

3.8 Root AVD for Windows OS

3.8.1 Enable Developer Mode

Do the same step 3.7.1

3.8.2 Enable USB Debugging

Do the same step 3.7.2

3.8.3 Download root script

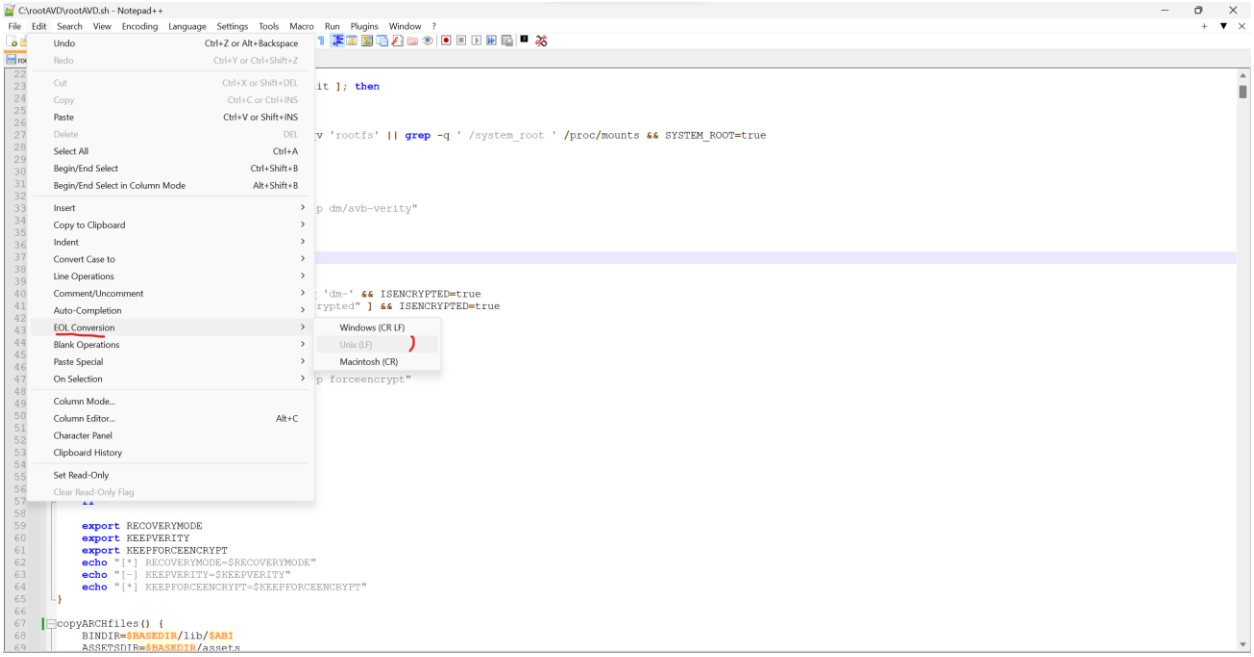
Do the same step 3.7.3

3.8.4 Install Magisk

Open file rootAVD.bat with Notepad++

Fix EOL Conversion from Windows (CR LF) to Unix (LF)

Edit → EOL Conversion → chose Unix (LF)



Run this command below in CMD (Administrator)

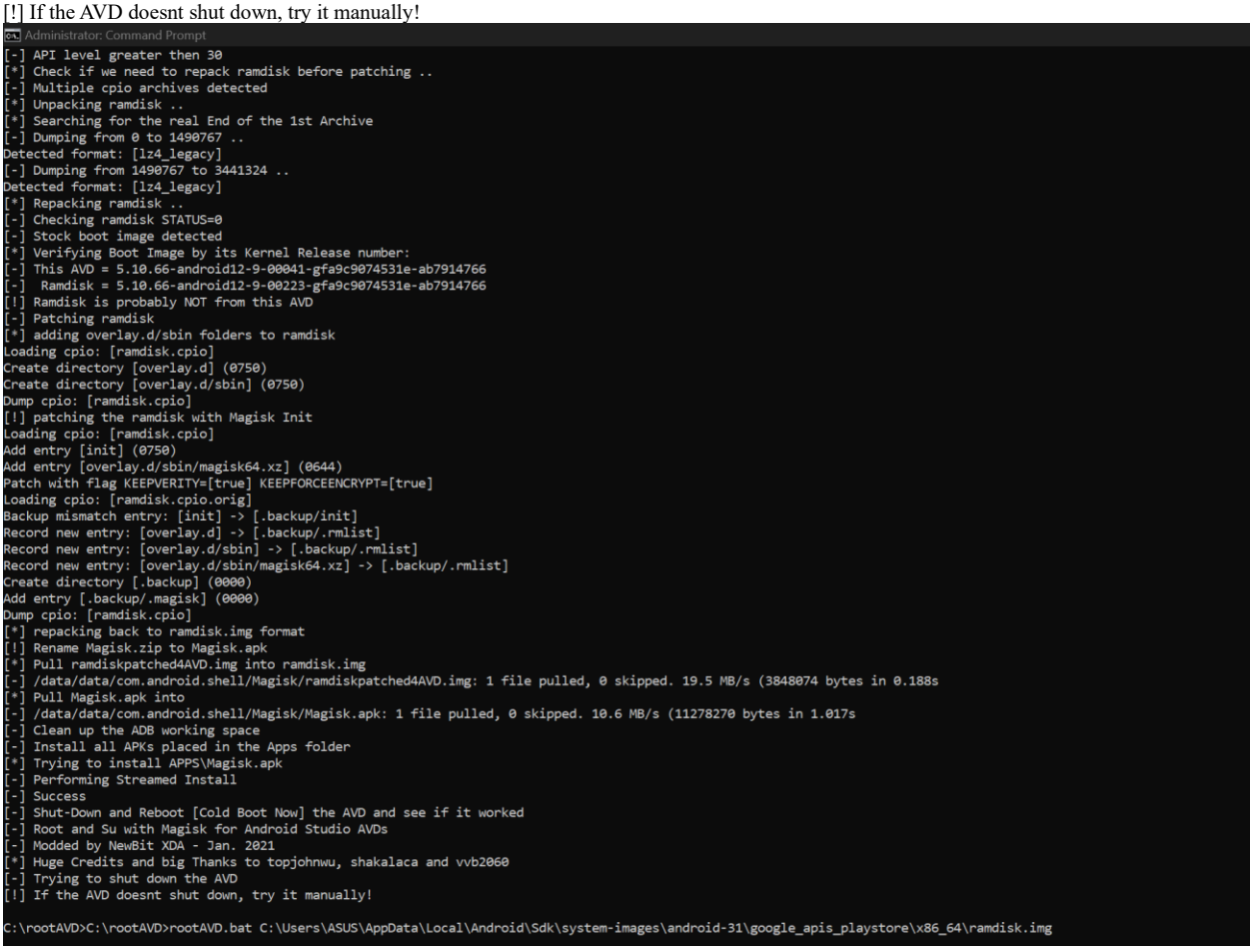
C:\rootAVD>rootAVD.bat C:\Users\ASUS\AppData\Local\Android\Sdk\system-images\android-31\google_apis_playstore\x86_64\ramdisk.img

Note: change **ASUS** based on your machine!

Running Output log

```
[*] Set Directorys
[-] Test if ADB SHELL is working
[-] ADB connectoin possible
[-] In any AVD via ADB, you can execute code without root in /data/data/com.android.shell
[*] looking for Magisk installer Zip
[*] Cleaning up the ADB working space
[*] Creating the ADB working space
[*] Push Magisk.zip into /data/data/com.android.shell/Magisk
[-] C:\rootAVD\Magisk.zip: 1 file pushed, 0 skipped. 28.5 MB/s (11278270 bytes in 0.377s)
[-] Backup exists already
[*] Push ramdisk.img into /data/data/com.android.shell/Magisk
[-] C:\Users\ASUS\AppData\Local\Android\Sdk\system-images\android-31\google_apis_playstore\x86_64\ramdisk.img: 1 file pushed, 0
skipped. 182.2 MB/s (3441343 bytes in 0.018s)
[-] Copy rootAVD Script into Magisk DIR
rootAVD.sh: 1 file pushed, 0 skipped. 76.9 MB/s (76222 bytes in 0.001s)
[-] run the actually Boot/Ramdisk/Kernel Image Patch Script
[*] from Magisk by topjohnwu and modded by NewBit XDA
[!] We are in a ranchu emulator shell
[-] Api Level Arch Detect
[-] Device Platform is x64 only
[-] Device SDK API: 31
[-] First API Level: 31
[-] The AVD runs on Android 12
[-] Switch to the location of the script file
[*] Looking for an unzip binary
[-] unzip binary found
[*] Extracting busybox and Magisk.zip via unzip ...
[*] Finding a working Busybox Version
[!] Found a working Busybox Version
[!] BusyBox v1.34.1-Magisk (2022-03-22 04:11:29 PDT) multi-call binary.
[*] Move busybox from lib to workdir
[-] Checking AVDs Internet connection...
[!] AVD is offline
[*] Re-Run rootAVD in Magisk Busybox STANDALONE (D)ASH
[-] We are now in Magisk Busybox STANDALONE (D)ASH
[*] rootAVD with Magisk '25.2' Installer
[-] Get Flags
[*] System-as-root, keep dm/avb-verity
[-] Encrypted data, keep forceencrypt
[*] RECOVERYMODE=false
[-] KEEPVERITY=true
[*] KEEPFORCEENCRYPT=true
[-] copy all x86_64 files from /data/data/com.android.shell/Magisk/lib/x86_64 to /data/data/com.android.shell/Magisk
[*] Detecting ramdisk.img compression
[!] Ramdisk.img uses lz4_legacy compression
[-] taken from shakalaca's MagiskOnEmulator/process.sh
[*] executing ramdisk splitting / extraction / repacking
[-] API level greater then 30
[*] Check if we need to repack ramdisk before patching ..
[-] Multiple cpio archives detected
[*] Unpacking ramdisk ..
[*] Searching for the real End of the 1st Archive
[-] Dumping from 0 to 1490767 ..
Detected format: [lz4_legacy]
[-] Dumping from 1490767 to 3441324 ..
Detected format: [lz4_legacy]
[*] Repacking ramdisk ..
[-] Checking ramdisk STATUS=0
[-] Stock boot image detected
[*] Verifying Boot Image by its Kernel Release number:
[-] This AVD = 5.10.66-android12-9-00041-gfa9c9074531e-ab7914766
[-]  Ramdisk = 5.10.66-android12-9-00223-gfa9c9074531e-ab7914766
[!] Ramdisk is probably NOT from this AVD
[-] Patching ramdisk
[*] adding overlay.d/sbin folders to ramdisk
Loading cpio: [ramdisk.cpio]
Create directory [overlay.d] (0750)
Create directory [overlay.d/sbin] (0750)
```

Dump cpio: [ramdisk.cpio]
[!] patching the ramdisk with Magisk Init
Loading cpio: [ramdisk.cpio]
Add entry [init] (0750)
Add entry [overlay.d/sbin/magisk64.xz] (0644)
Patch with flag KEEPVERITY=[true] KEEPFORCEENCRYPT=[true]
Loading cpio: [ramdisk.cpio.orig]
Backup mismatch entry: [init] -> [.backup/init]
Record new entry: [overlay.d] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin] -> [.backup/.rmlist]
Record new entry: [overlay.d/sbin/magisk64.xz] -> [.backup/.rmlist]
Create directory [.backup] (0000)
Add entry [.backup/.magisk] (0000)
Dump cpio: [ramdisk.cpio]
[*] repacking back to ramdisk.img format
[!] Rename Magisk.zip to Magisk.apk
[*] Pull ramdiskpatched4AVD.img into ramdisk.img
[-] /data/data/com.android.shell/Magisk/ramdiskpatched4AVD.img: 1 file pulled, 0 skipped. 19.5 MB/s (3848074 bytes in 0.188s
[*] Pull Magisk.apk into
[-] /data/data/com.android.shell/Magisk/Magisk.apk: 1 file pulled, 0 skipped. 10.6 MB/s (11278270 bytes in 1.017s
[-] Clean up the ADB working space
[-] Install all APKs placed in the Apps folder
[*] Trying to install APPS\Magisk.apk
[-] Performing Streamed Install
[-] Success
[-] Shut-Down and Reboot [Cold Boot Now] the AVD and see if it worked
[-] Root and Su with Magisk for Android Studio AVDs
[-] Modded by NewBit XDA - Jan. 2021
[*] Huge Credits and big Thanks to topjohnwu, shakalaca and vvb2060
[-] Trying to shut down the AVD
[!] If the AVD doesnt shut down, try it manually!



3.8.5 Config Root by Magisk

Do the same step 4.7.5

3.8.6 Grant root user for adb shell

Do the same step 4.7.6

4 Setup MITM proxy certificate on AVD

Note: do the same for both AVD on Windows OS and Ubuntu OS

4.1 Copy source

ubuntu_client:~# git clone https://github.com/thanhlam2110/metadata-root-adv.git

ubuntu_client:~# cd metadata-root-adv/

ubuntu_client:~/metadata-root-adv # adb push AlwaysTrustUserCerts.zip /sdcard/Download/

```
root@lam-G550JK:~/root_android_phone# adb push AlwaysTrustUserCerts.zip /sdcard/Download/
AlwaysTrustUserCerts.zip: 1 file pushed, 0 skipped. 6.7 MB/s (3574 bytes in 0.001s)
```

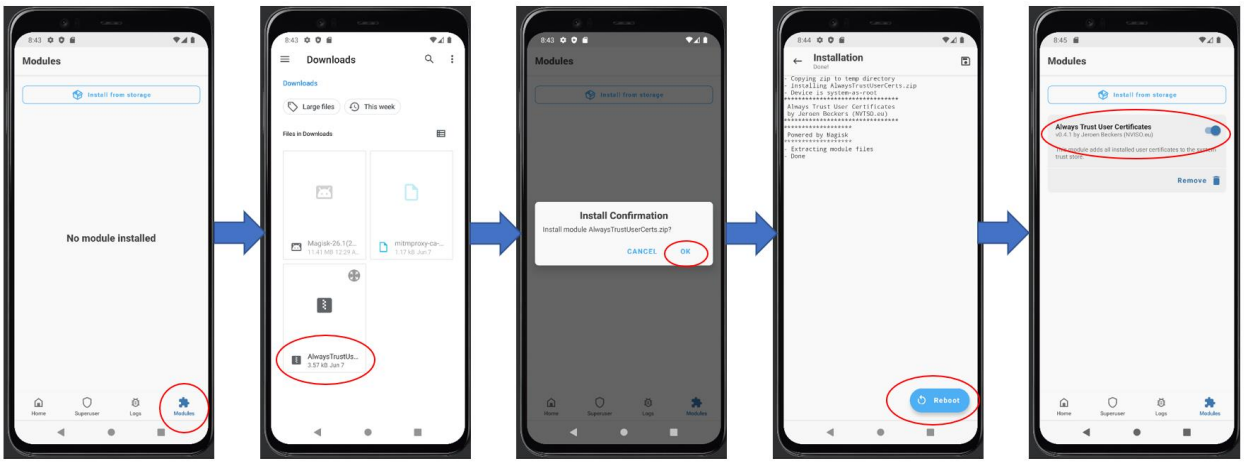
ubuntu_client:~/metadata-root-adv# adb push mitmproxy-ca-cert.cer /sdcard/Download/

```
root@lam-G550JK:~/root_android_phone# adb push mitmproxy-ca-cert.cer /sdcard/Download/
mitmproxy-ca-cert.cer: 1 file pushed, 0 skipped. 0.3 MB/s (1172 bytes in 0.004s)
root@lam-G550JK:~/root_android_phone#
```

Note:

The file mitmproxy-ca-cert.cer is automatically generated during the installation of the MITM Proxy server, and each server has a corresponding certificate at path (/root/.mitmproxy). So, at this step, please ask mitmproxy-ca-cert.cer from the MITM Proxy Server administrator.

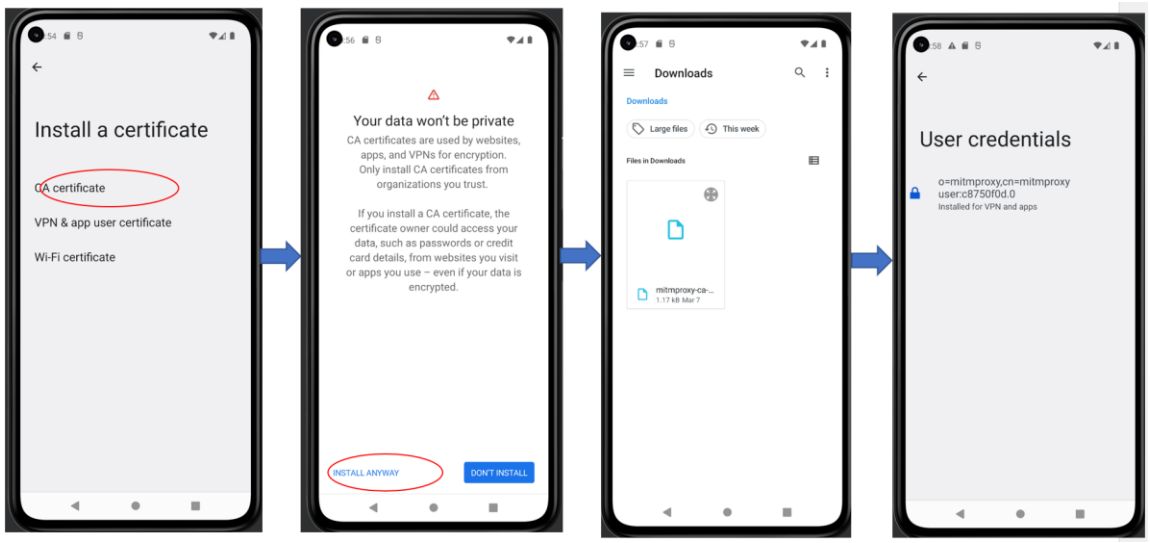
4.2 Install AlwaysTrustUserCerts



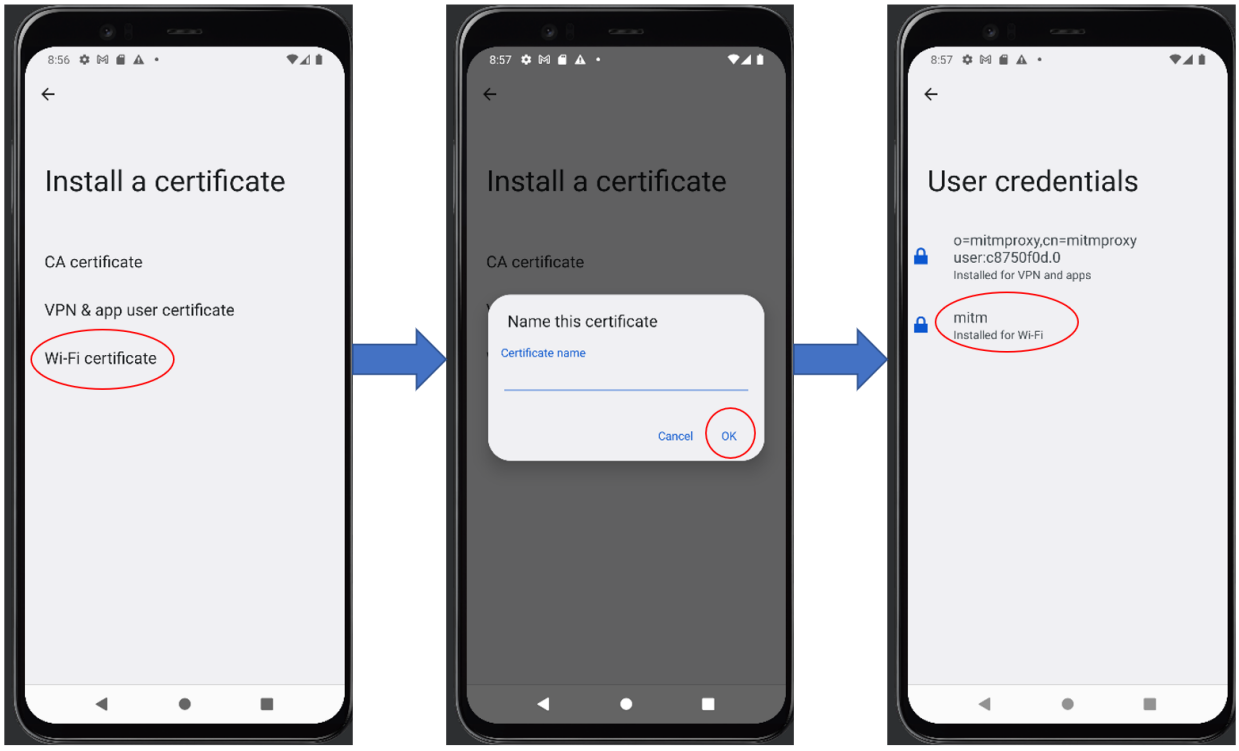
4.3 Install MITM certificate

Settings → Security → Encryption & Credentials → Install a certificate

CA certificate

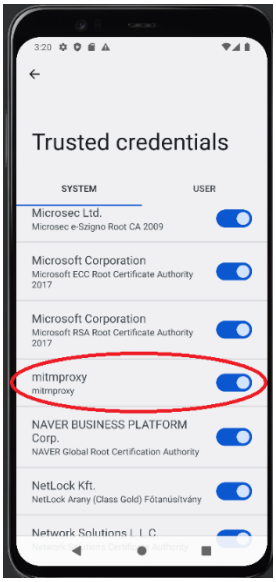


Wifi Certificate



ubuntu_client:~/root_android_phone# adb reboot

Result



4.4 Check using MITM Proxy Server capture upload image traffic

4.4.1 Copy image has metadata

ubuntu_client:~/root_android_phone# adb push LANDSCAPE_ORIGIN.jpg /sdcard/Download/

```
root@lam-G550JK:~/root_android_phone# adb push LANDSCAPE_ORIGIN.jpg /sdcard/Download/
LANDSCAPE_ORIGIN.jpg: 1 file pushed, 0 skipped. 10.7 MB/s (74490 bytes in 0.007s)
root@lam-G550JK:~/root_android_phone# adb push LAPTOP-ORIGIN.jpg /sdcard/Download/
LAPTOP-ORIGIN.jpg: 1 file pushed, 0 skipped. 22.3 MB/s (116896 bytes in 0.005s)
root@lam-G550JK:~/root_android_phone#
```

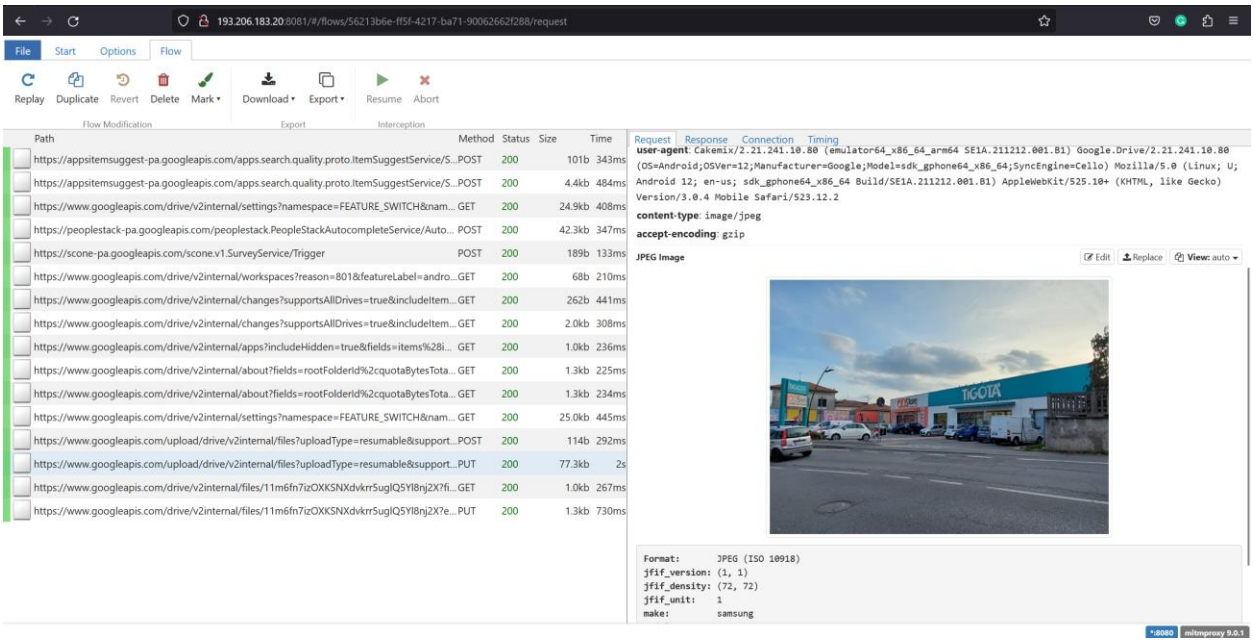
4.4.2 Redirect traffic to MITM proxy

ubuntu_client:~# adb shell settings put global http_proxy **SERVER_IP**:8080

4.4.3 Run MITM proxy server

server_machine:~# mitmweb --web-host 0.0.0.0 --web-port 8081 --set block_global=false

4.4.4 Test upload image and capture traffic my MITM Proxy



Use Google Drive on Virtual Phone and upload one image. If you can capture traffic by the MITM Proxy like figure. You succeed.

5 Install Running-Script Environment on Client Ubuntu OS

5.1 Install python

Check python version

ubuntu_client:~# python3 --version

```
root@ubuntu-desktop:~# python3 --version
Python 3.10.6
```

Install pip

ubuntu_client:~# apt-get install python3-pip

ubuntu_client:~# pip --version

```
root@ubuntu-desktop:~# pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

ubuntu_client:~# pip install --upgrade pip

```
root@ubuntu-desktop:~# pip install --upgrade pip
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (22.0.2)
Collecting pip
  Downloading pip-23.1.2-py3-none-any.whl (2.1 MB)
     ---------------------------------------- 2.1/2.1 MB 4.2 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.2
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment /usr
    Can't uninstall 'pip'. No files were found to uninstall.
Successfully installed pip-23.1.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@ubuntu-desktop:~# pip --version
pip 23.1.2 from /usr/local/lib/python3.10/dist-packages/pip (python 3.10)
root@ubuntu-desktop:~#
```

5.2 Install miniconda

ubuntu_client:~# wget https://repo.anaconda.com/miniconda/Miniconda3-py310_23.3.1-0-Linux-x86_64.sh

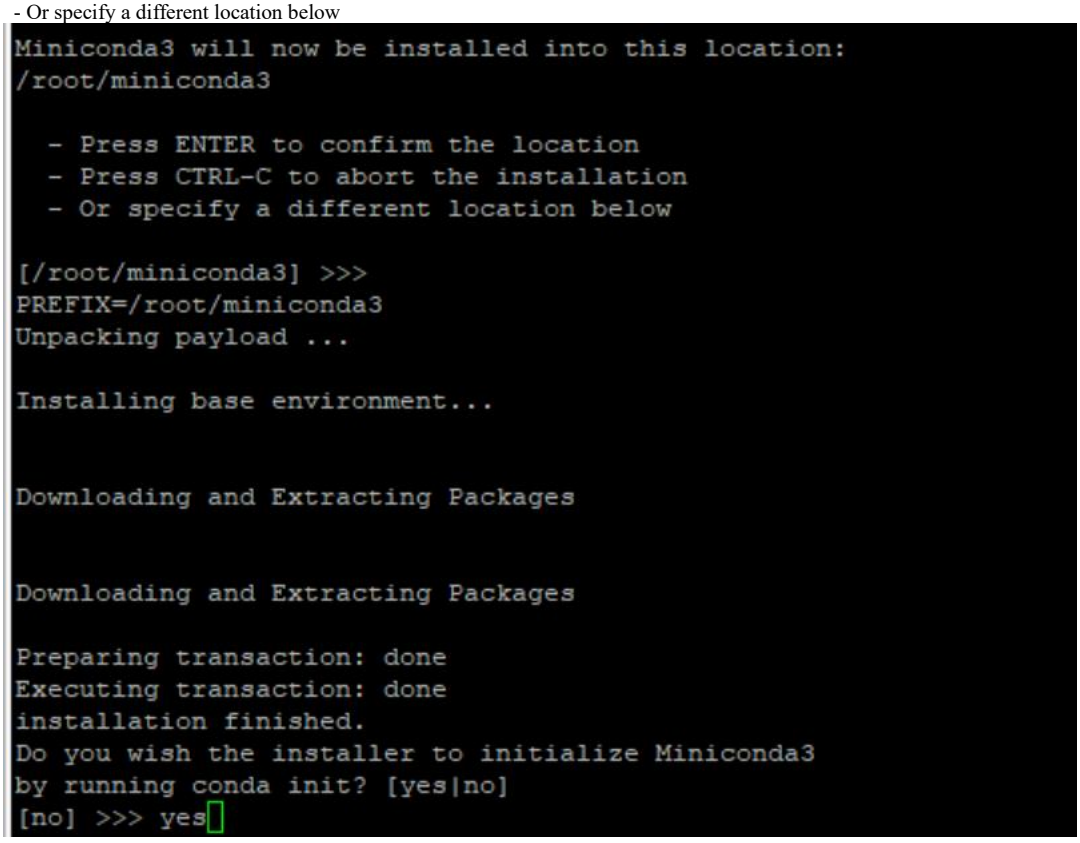
ubuntu_client:~# chmod -R 777 Miniconda3-py310_23.3.1-0-Linux-x86_64.sh

ubuntu_client:~# ./Miniconda3-py310_23.3.1-0-Linux-x86_64.sh

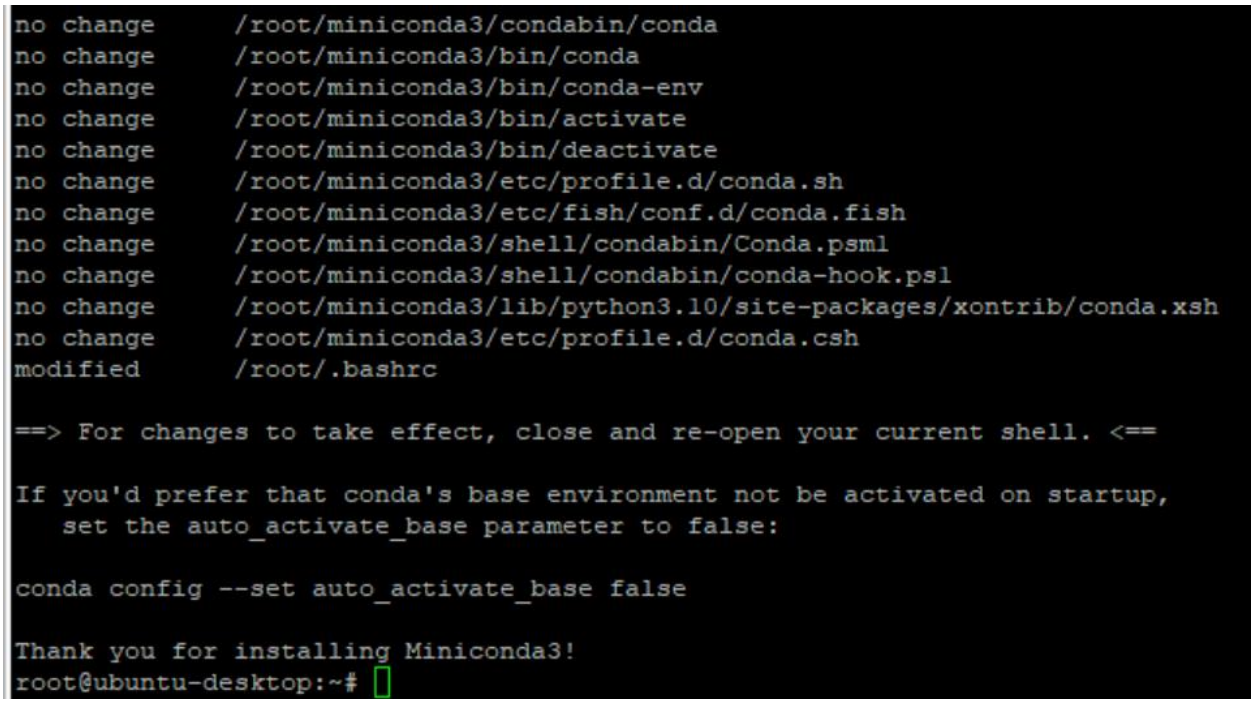
Press ENTER

Do you accept the license terms? [yes|no]
[no] >>>
Please answer 'yes' or 'no':'
>>> yes

Miniconda3 will now be installed into this location:
/root/miniconda3
- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below



Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes



no change /root/miniconda3/condabin/conda
no change /root/miniconda3/bin/conda
no change /root/miniconda3/bin/conda-env
no change /root/miniconda3/bin/activate
no change /root/miniconda3/bin/deactivate
no change /root/miniconda3/etc/profile.d/conda.sh
no change /root/miniconda3/etc/fish/conf.d/conda.fish
no change /root/miniconda3/shell/condabin/Conda.psm1
no change /root/miniconda3/shell/condabin/conda-hook.ps1
no change /root/miniconda3/lib/python3.10/site-packages/xontrib/conda.xsh
no change /root/miniconda3/etc/profile.d/conda.csh
modified /root/.bashrc
==> For changes to take effect, close and re-open your current shell. <==
If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:
conda config --set auto_activate_base false
Thank you for installing Miniconda3!

5.3 Install virtual environment

5.3.1 Create virtual environment

(base) ubuntu_client:~# conda create --name metadata python=3.10

Note: (base) means you are in the default (base) environment of miniconda

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
To activate this environment, use
#
$ conda activate metadata
#
To deactivate an active environment, use
#
$ conda deactivate

```
  ---------------------------|-----------------
  python-3.10.11             |       h7a1cb2a_2        26.8 MB
  setuptools-67.8.0          |  py310h06a4308_0         1.0 MB
  ------------------------------------------------------------
                                       Total:        27.8 MB

The following NEW packages will be INSTALLED:

  _libgcc_mutex      pkgs/main/linux-64::_libgcc_mutex-0.1-main
  _openmp_mutex      pkgs/main/linux-64::_openmp_mutex-5.1-1_gnu
  bzip2              pkgs/main/linux-64::bzip2-1.0.8-h7b6447c_0
  ca-certificates    pkgs/main/linux-64::ca-certificates-2023.05.30-h06a4308_0
  ld_impl_linux-64   pkgs/main/linux-64::ld_impl_linux-64-2.38-h1181459_1
  libffi             pkgs/main/linux-64::libffi-3.4.4-h6a678d5_0
  libgcc-ng          pkgs/main/linux-64::libgcc-ng-11.2.0-h1234567_1
  libgomp            pkgs/main/linux-64::libgomp-11.2.0-h1234567_1
  libstdcxx-ng       pkgs/main/linux-64::libstdcxx-ng-11.2.0-h1234567_1
  libuuid            pkgs/main/linux-64::libuuid-1.41.5-h5eee18b_0
  ncurses            pkgs/main/linux-64::ncurses-6.4-h6a678d5_0
  openssl            pkgs/main/linux-64::openssl-1.1.1t-h7f8727e_0
  pip                pkgs/main/linux-64::pip-23.0.1-py310h06a4308_0
  python             pkgs/main/linux-64::python-3.10.11-h7a1cb2a_2
  readline           pkgs/main/linux-64::readline-8.2-h5eee18b_0
  setuptools         pkgs/main/linux-64::setuptools-67.8.0-py310h06a4308_0
  sqlite             pkgs/main/linux-64::sqlite-3.41.2-h5eee18b_0
  tk                 pkgs/main/linux-64::tk-8.6.12-h1ccaba5_0
  tzdata             pkgs/main/noarch::tzdata-2023c-h04d1e81_0
  wheel              pkgs/main/linux-64::wheel-0.38.4-py310h06a4308_0
  xz                 pkgs/main/linux-64::xz-5.4.2-h5eee18b_0
  zlib               pkgs/main/linux-64::zlib-1.2.13-h5eee18b_0


Proceed ([y]/n)? y


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate metadata
#
# To deactivate an active environment, use
#
#     $ conda deactivate

root@ubuntu-desktop:~#
```

List exist miniconda environment

ubuntu_client:~# conda env list

```
root@lam-G550JK:~# conda env list
# conda environments:
#
base                  /root/miniconda3
droidbot              /root/miniconda3/envs/droidbot
metadata              /root/miniconda3/envs/metadata

root@lam-G550JK:~#
```

5.3.2 Active virtual environment

(base) ubuntu_client:~# conda activate metadata

```
(base) root@ubuntu-desktop:~# conda activate metadata
(metadata) root@ubuntu-desktop:~#
```

5.3.3 Install jupyter

Open new terminal

(metadata) ubuntu_client:~# conda install jupyter

```
  qtpy               pkgs/main/linux-64::qtpy-2.2.0-py311h06a4308_0
  qtwebkit           pkgs/main/linux-64::qtwebkit-5.212-h3fafdc1_5
  readline           pkgs/main/linux-64::readline-8.2-h5eee18b_0
  requests           pkgs/main/linux-64::requests-2.29.0-py311h06a4308_0
  rfc3339-validator  pkgs/main/linux-64::rfc3339-validator-0.1.4-py311h06a4308_0
  rfc3986-validator  pkgs/main/linux-64::rfc3986-validator-0.1.1-py311h06a4308_0
  send2trash         pkgs/main/noarch::send2trash-1.8.0-pyhd3eb1b0_1
  setuptools         pkgs/main/linux-64::setuptools-67.8.0-py311h06a4308_0
  sip                pkgs/main/linux-64::sip-6.6.2-py311h6a678d5_0
  six                pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
  sniffio            pkgs/main/linux-64::sniffio-1.2.0-py311h06a4308_1
  soupsieve          pkgs/main/linux-64::soupsieve-2.4-py311h06a4308_0
  sqlite             pkgs/main/linux-64::sqlite-3.41.2-h5eee18b_0
  stack_data         pkgs/main/noarch::stack_data-0.2.0-pyhd3eb1b0_0
  terminado          pkgs/main/linux-64::terminado-0.17.1-py311h06a4308_0
  tinycss2           pkgs/main/linux-64::tinycss2-1.2.1-py311h06a4308_0
  tk                 pkgs/main/linux-64::tk-8.6.12-h1ccaba5_0
  toml               pkgs/main/noarch::toml-0.10.2-pyhd3eb1b0_0
  tomli              pkgs/main/linux-64::tomli-2.0.1-py311h06a4308_0
  tornado            pkgs/main/linux-64::tornado-6.2-py311h5eee18b_0
  traitlets          pkgs/main/linux-64::traitlets-5.7.1-py311h06a4308_0
  typing-extensions  pkgs/main/linux-64::typing-extensions-4.5.0-py311h06a4308_0
  typing_extensions  pkgs/main/linux-64::typing_extensions-4.5.0-py311h06a4308_0
  tzdata             pkgs/main/noarch::tzdata-2023c-h04d1e81_0
  urllib3            pkgs/main/linux-64::urllib3-1.26.15-py311h06a4308_0
  wcwidth            pkgs/main/noarch::wcwidth-0.2.5-pyhd3eb1b0_0
  webencodings       pkgs/main/linux-64::webencodings-0.5.1-py311h06a4308_1
  websocket-client   pkgs/main/linux-64::websocket-client-0.58.0-py311h06a4308_4
  wheel              pkgs/main/linux-64::wheel-0.38.4-py311h06a4308_0
  widgetsnbextension pkgs/main/linux-64::widgetsnbextension-4.0.5-py311h06a4308_0
  xz                 pkgs/main/linux-64::xz-5.4.2-h5eee18b_0
  yaml               pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0
  zeromq             pkgs/main/linux-64::zeromq-4.3.4-h2531618_0
  zlib               pkgs/main/linux-64::zlib-1.2.13-h5eee18b_0
  zstd               pkgs/main/linux-64::zstd-1.5.5-hc292b87_0


Proceed ([y]/n)? y


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

(metadata) ubuntu_client:~# jupyter notebook --generate-config

```
(base) root@ubuntu-desktop:~# jupyter notebook --generate-config
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
(base) root@ubuntu-desktop:~#
```

(metadata) ubuntu_client:~# nano /root/.jupyter/jupyter_notebook_config.py

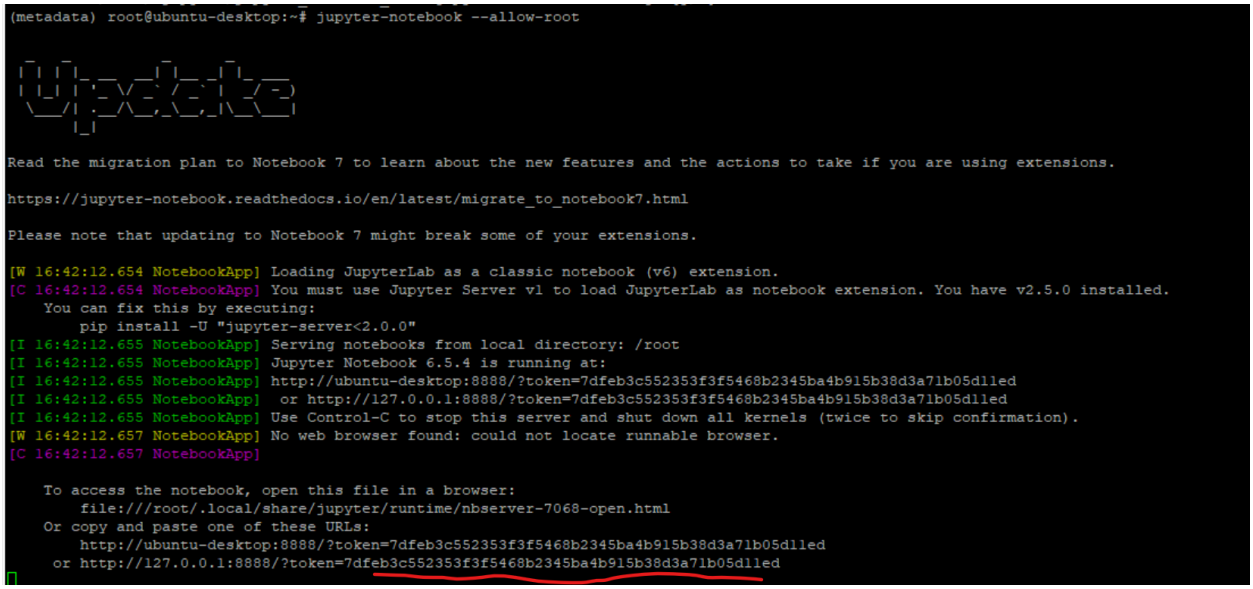
Edit

```
# c.NotebookApp.allow_remote_access = False
c.NotebookApp.allow_remote_access = True
## Whether to allow the user to run the notebook as root.
#  Default: False
c.NotebookApp.allow_root = True
```

```
# c.NotebookApp.ip = 'localhost'
c.NotebookApp.ip = '0.0.0.0'
```

Run Jupyter notebook

(metadata) ubuntu_client:~# jupyter-notebook --allow-root

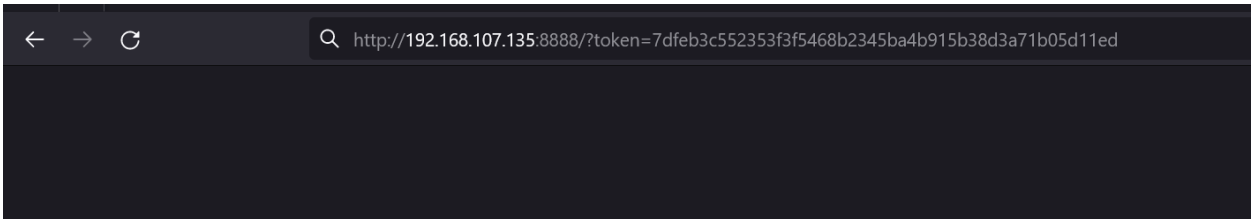


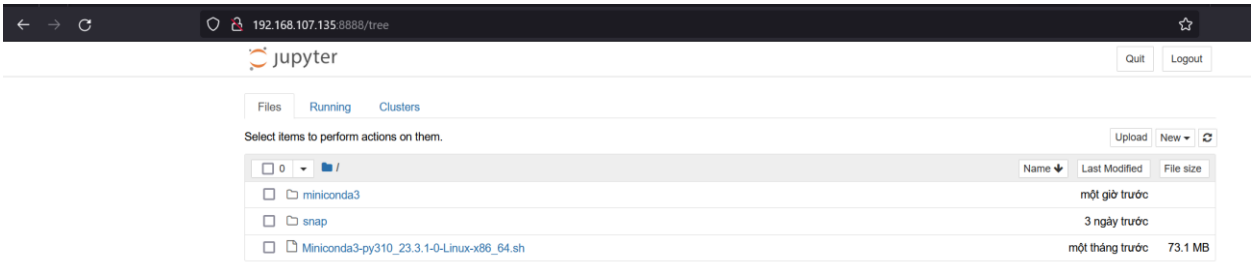
To access the notebook, open this file in a browser:
file:///root/.local/share/jupyter/runtime/nbserver-7068-open.html
Or copy and paste one of these URLs:
http://ubuntu-desktop:8888/?token=7dfeb3c552353f3f5468b2345ba4b915b38d3a71b05d11ed
or http://127.0.0.1:8888/?token=7dfeb3c552353f3f5468b2345ba4b915b38d3a71b05d11ed

Remember the token for login

Access Jupyter notebook

http://**CLIENT_IP**:8888/?token=7dfeb3c552353f3f5468b2345ba4b915b38d3a71b05d11ed





5.3.4 Install python library

5.3.4.1 Pillow

(metadata) ubuntu_client:~# conda install -c anaconda pillow

```
(metadata) root@ubuntu-desktop:~# conda install -c anaconda pillow
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /root/miniconda3/envs/metadata

  added / updated specs:
    - pillow


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2023.01.10 |       h06a4308_0         127 KB  anaconda
    certifi-2020.6.20          |     pyhd3eb1b0_3         159 KB  anaconda
    lcms2-2.12                 |       h3be6417_0         396 KB  anaconda
    pillow-9.4.0               |  py311h6a678d5_0         815 KB
    ------------------------------------------------------------
                                           Total:         1.5 MB

The following NEW packages will be INSTALLED:

  lcms2              anaconda/linux-64::lcms2-2.12-h3be6417_0
  pillow             pkgs/main/linux-64::pillow-9.4.0-py311h6a678d5_0

The following packages will be SUPERSEDED by a higher-priority channel:

  ca-certificates                                pkgs/main --> anaconda
  certifi            pkgs/main/linux-64::certifi-2023.5.7-~ --> anaconda/noarch::certifi-2020.6.20-pyhd3eb1b0_3


Proceed ([y]/n)? y


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

(metadata) ubuntu_client:~# pip show pillow

```
(metadata) root@ubuntu-desktop:~# pip show pillow
Name: Pillow
Version: 9.4.0
Summary: Python Imaging Library (Fork)
Home-page: https://python-pillow.org
Author: Alex Clark (PIL Fork Author)
Author-email: aclark@python-pillow.org
License: HPND
Location: /root/miniconda3/envs/metadata/lib/python3.11/site-packages
Requires:
Required-by:
(metadata) root@ubuntu-desktop:~#
```

5.3.4.2 pure-python-adb

(metadata) ubuntu_client:~# pip install pure-python-adb

```
(metadata) root@lam-G550JK:~# pip install pure-python-adb
Collecting pure-python-adb
  Using cached pure_python_adb-0.3.0.dev0-py3-none-any.whl
Installing collected packages: pure-python-adb
Successfully installed pure-python-adb-0.3.0.dev0
```

(metadata) ubuntu_client:~# pip show pure-python-adb

```
(metadata) root@lam-G550JK:~# pip show pure-python-adb
Name: pure-python-adb
Version: 0.3.0.dev0
Summary: Pure python implementation of the adb client
Home-page: https://github.com/Swind/pure-python-adb
Author: Swind Ou
Author-email: swind@cloudmosa.com
License: MIT license
Location: /root/miniconda3/envs/metadata/lib/python3.10/site-packages
Requires:
Required-by:
(metadata) root@lam-G550JK:~#
```

5.3.4.3 numpy

(metadata) ubuntu_client:~# conda install numpy=1.24.3

```
(metadata) root@lam-G550JK:~# conda install numpy=1.24.3
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /root/miniconda3/envs/metadata

  added / updated specs:
    - numpy=1.24.3


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    openssl-1.1.1u             |       h7f8727e_0         3.7 MB
    ------------------------------------------------------------
                                           Total:         3.7 MB

The following NEW packages will be INSTALLED:

  blas               pkgs/main/linux-64::blas-1.0-mkl
  intel-openmp       pkgs/main/linux-64::intel-openmp-2023.1.0-hdb19cb5_46305
  mkl                pkgs/main/linux-64::mkl-2023.1.0-h6d00ec8_46342
  mkl-service        pkgs/main/linux-64::mkl-service-2.4.0-py310h5eee18b_1
  mkl_fft            pkgs/main/linux-64::mkl_fft-1.3.6-py310h1128e8f_1
  mkl_random         pkgs/main/linux-64::mkl_random-1.2.2-py310h1128e8f_1
  numpy              pkgs/main/linux-64::numpy-1.24.3-py310h5f9d8c6_1
  numpy-base         pkgs/main/linux-64::numpy-base-1.24.3-py310hb5e798b_1
  tbb                pkgs/main/linux-64::tbb-2021.8.0-hdb19cb5_0

The following packages will be UPDATED:

  ca-certificates    anaconda::ca-certificates-2023.01.10-~ --> pkgs/main::ca-certificates-2023.05.30-h06a4308_0
  certifi            anaconda::certifi-2022.12.7-py310h06a~ --> pkgs/main::certifi-2023.5.7-py310h06a4308_0
  openssl                                1.1.1t-h7f8727e_0 --> 1.1.1u-h7f8727e_0
```

(metadata) ubuntu_client:~# pip show numpy

```
(metadata) root@lam-G550JK:~# pip show numpy
Name: numpy
Version: 1.24.3
Summary: Fundamental package for array computing in Python
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD-3-Clause
Location: /root/miniconda3/envs/metadata/lib/python3.10/site-packages
Requires:
Required-by: mkl-fft, mkl-random
(metadata) root@lam-G550JK:~#
```

5.3.4.4 pandas

(metadata) ubuntu_client:~# conda install pandas=1.4.4

```
(metadata) root@lam-G550JK:~# conda install pandas=1.4.4
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /root/miniconda3/envs/metadata

  added / updated specs:
    - pandas=1.4.4


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    pandas-1.4.4               |  py310h6a678d5_0        25.5 MB
    ------------------------------------------------------------
                                           Total:        25.5 MB

The following NEW packages will be INSTALLED:

  bottleneck         pkgs/main/linux-64::bottleneck-1.3.5-py310ha9d4c09_0
  numexpr            pkgs/main/linux-64::numexpr-2.8.4-py310h85018f9_1
  pandas             pkgs/main/linux-64::pandas-1.4.4-py310h6a678d5_0
```

(metadata) ubuntu_client:~# pip show pandas

```
(metadata) root@lam-G550JK:~# pip show pandas
Name: pandas
Version: 1.4.4
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: https://pandas.pydata.org
Author: The Pandas Development Team
Author-email: pandas-dev@python.org
License: BSD-3-Clause
Location: /root/miniconda3/envs/metadata/lib/python3.10/site-packages
Requires: numpy, python-dateutil, pytz
Required-by:
(metadata) root@lam-G550JK:~#
```

5.3.4.5 kafka

(metadata) ubuntu_client:~# pip install confluent-kafka

```
(metadata) root@lam-G550JK:~# pip show pure-python-adb
Name: pure-python-adb
Version: 0.3.0.dev0
Summary: Pure python implementation of the adb client
Home-page: https://github.com/Swind/pure-python-adb
Author: Swind Ou
Author-email: swind@cloudmosa.com
License: MIT license
Location: /root/miniconda3/envs/metadata/lib/python3.10/site-packages
Requires:
Required-by:
(metadata) root@lam-G550JK:~# pip install confluent-kafka
Collecting confluent-kafka
  Downloading confluent_kafka-2.1.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.9 MB)
     ---------------------------------------- 3.9/3.9 MB 4.6 MB/s eta 0:00:00
Installing collected packages: confluent-kafka
Successfully installed confluent-kafka-2.1.1
```

(metadata) ubuntu_client:~# pip show confluent-kafka

```
(base) root@distarossi-gs:/home/sirigu/personalization-seperate-0-to-1000# pip show confluent-kafka
Name: confluent-kafka
Version: 2.2.0
Summary: Confluent's Python client for Apache Kafka
Home-page: https://github.com/confluentinc/confluent-kafka-python
Author: Confluent Inc
Author-email: support@confluent.io
License:
Location: /root/miniconda3/lib/python3.10/site-packages
Requires:
Required-by:
(base) root@distarossi-gs:/home/sirigu/personalization-seperate-0-to-1000#
```

1.1.1.1 exif

(metadata) ubuntu_client:~# pip install exif

```
(base) root@distarossi-gs:/home/sirigu/personalization-seperate-0-to-1000# pip install exif
Collecting exif
  Downloading exif-1.6.0-py3-none-any.whl (30 kB)
Collecting plum-py<2.0.0,>=0.5.0 (from exif)
  Downloading plum_py-0.8.6-py3-none-any.whl (69 kB)
     ---------------------------------------- 69.9/69.9 kB 5.7 MB/s eta 0:00:00
Installing collected packages: plum-py, exif
Successfully installed exif-1.6.0 plum-py-0.8.6
```

(metadata) ubuntu_client:~# pip show exif

```
(base) root@distarossi-gs:/home/sirigu/personalization-seperate-0-to-1000# pip show exif
Name: exif
Version: 1.6.0
Summary: Read and modify image EXIF metadata using Python.
Home-page: https://gitlab.com/TNThieding/exif
Author: Tyler N. Thieding
Author-email: python@thieding.com
License: UNKNOWN
Location: /root/miniconda3/lib/python3.10/site-packages
Requires: plum-py
Required-by:
```

1.1.1.2 selenium

pip install selenium

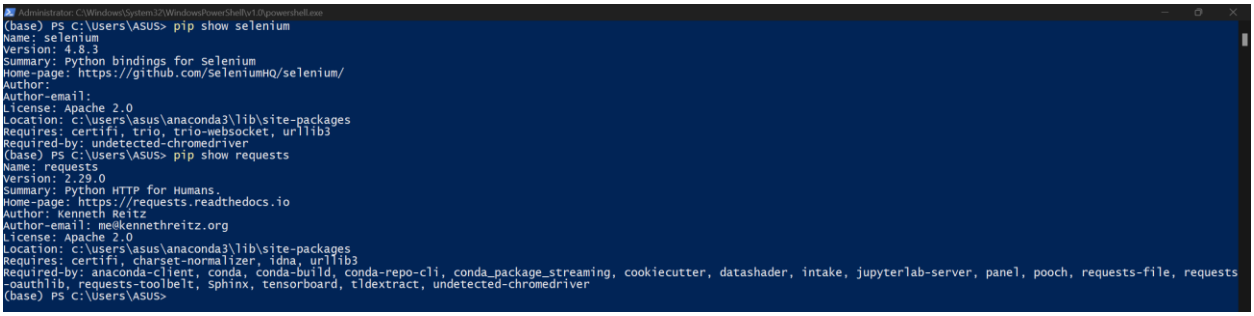
pip show selenium

```
Select Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
(base) PS C:\Users\ASUS> pip show selenium
Name: selenium
Version: 4.8.3
Summary: Python bindings for Selenium
Home-page: https://github.com/SeleniumHQ/selenium/
Author:
Author-email:
License: Apache 2.0
Location: c:\users\asus\anaconda3\lib\site-packages
Requires: certifi, trio, trio-websocket, urllib3
Required-by: undetected-chromedriver
(base) PS C:\Users\ASUS>
```

1.1.1.3 request

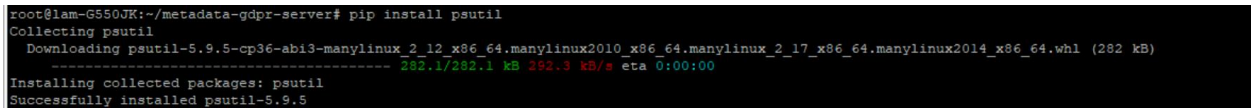
pip install requests

pip show requests



1.1.1.4 psutil

pip install psutil



pip show psutil

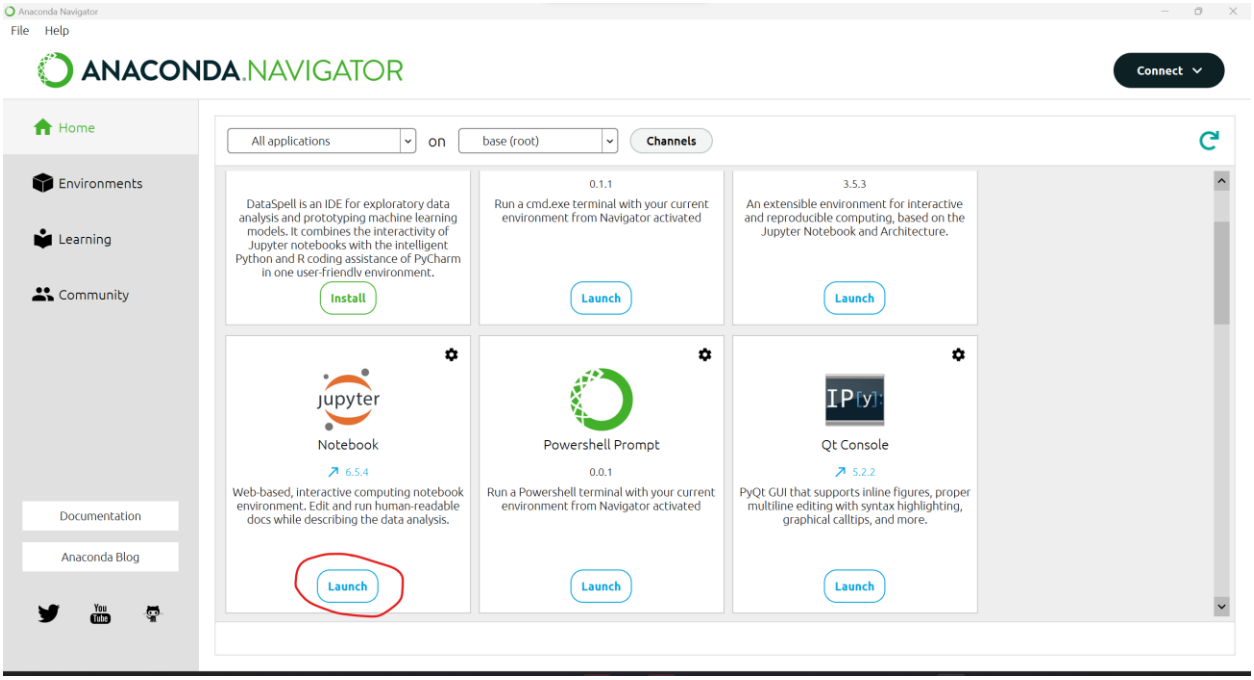
6 Install Running-Script Environment on Client Windows OS

6.1 Install Anaconda

https://docs.anaconda.com/free/anaconda/install/windows/

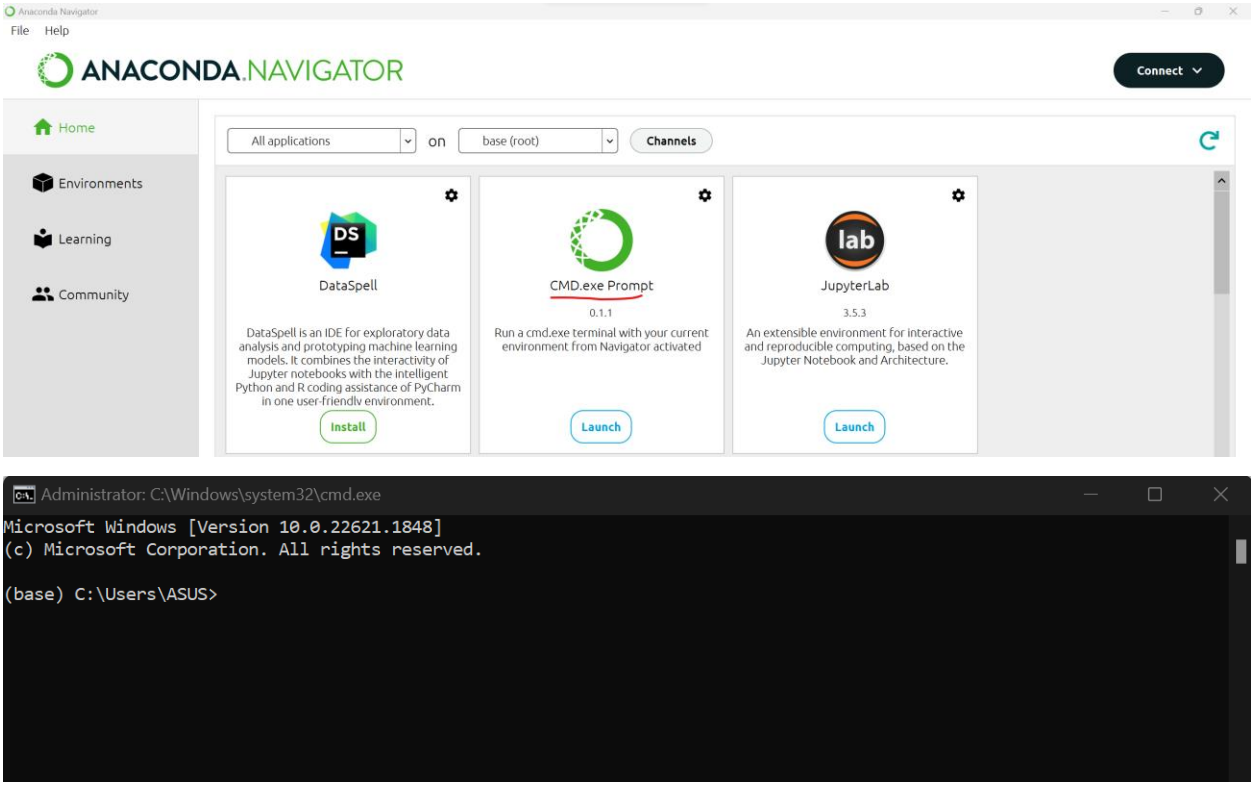
6.2 Install Jupyter

Anaconda already has Jupyter, you don't need to install just launch it.



6.3 Install the Python library

Using **CMD.exe Prompt** to install the library



Install the same Python library as Client Ubuntu OS