**Appium Mobile Automation with Raspberry PI**

What is **raspberry PI**? [ <https://www.raspberrypi.org/help/faqs/#introWhatIs> ]

The Raspberry Pi is a credit-card-sized [single-board computer](https://en.wikipedia.org/wiki/Single-board_computer)  that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does.it is a very low cost device.

What is **Appium**?

Appium is an open source test automation tool for mobile applications. It allows you to test all the three types of mobile applications: native, hybrid and mobile web. It also allows you to run the automated tests on actual devices, emulators and simulators.

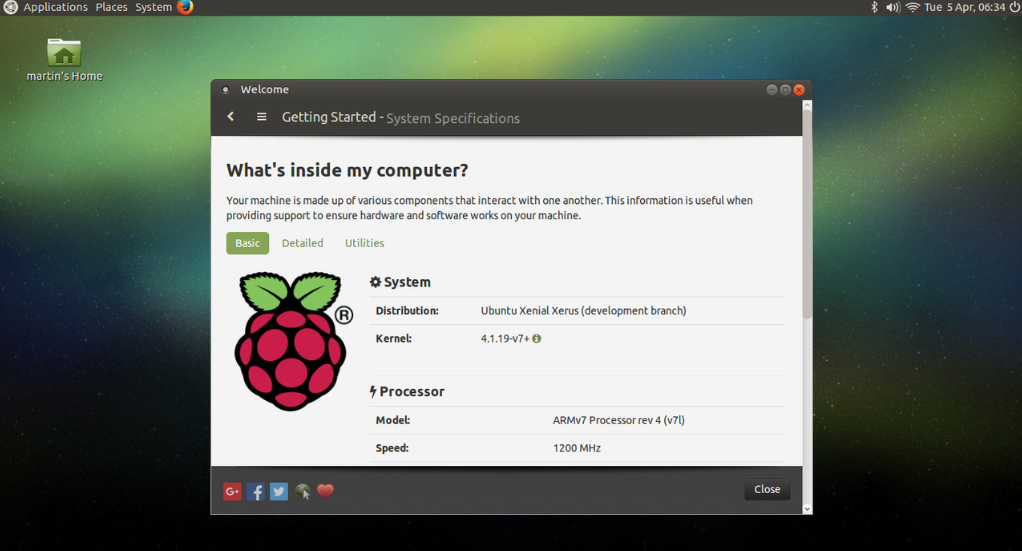
<https://collaboration.wal-mart.com/display/NIMAUT/Mobile+Automation+Testing+with+Appium+and+Android+Emulator>

Configuring **raspberry PI**?

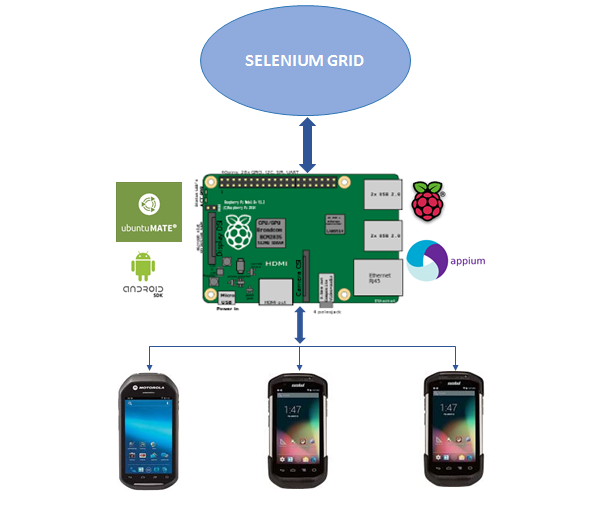
Start by downloading the Ubuntu MATE for Raspberry Pi 2 or the one for Raspberry Pi 3 in your ~/Downloads folder. Eject the SD card physically from your PC and Insert the SD card in your Raspberry Pi. The raspberry PI we have is ARM 32 bit

You’ll need a microSD card which is **8GB** or greater to fit the image. The file system can be resized to occupy the unallocated space of the microSD card.

Once the OS has been loaded the PI device can be connected to monitor with HDMI port and a window like below will be loaded.



**Architecture Diagram**



**Install/configure Android SDK & JAVA**

Download JAVA runtime [JDK] and set the JAVA\_HOME path.

Download the android SDK for Linux**;** right click the file [.tgz] and extract.





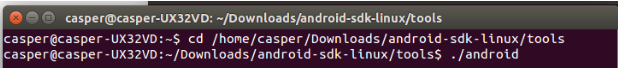
Set the PATH, ANDROID\_HOME, JAVA\_HOME in .bashrc

export PATH=${PATH}:~/android-sdk-linux/tools

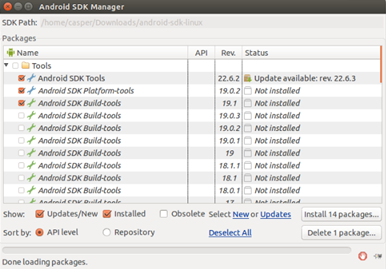
export PATH=${PATH}:~/android-sdk-linux/platform-tools

Press the keys CTRL + ALT + T to open a terminal and then use the following command. Where /home/\*\*\*\*/Downloads/android-sdk-linux/tools is the location I copied previously. You can just type cd and right click to past the location to the terminal.

cd /home/\*\*\*\*/Downloads/android-sdk-linux/tools ./android

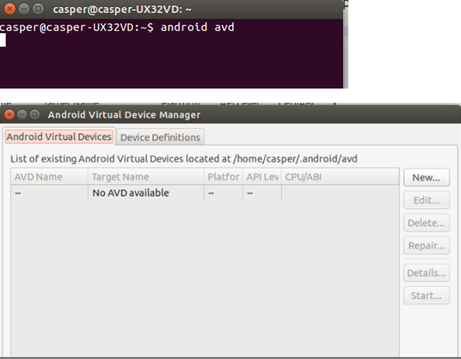


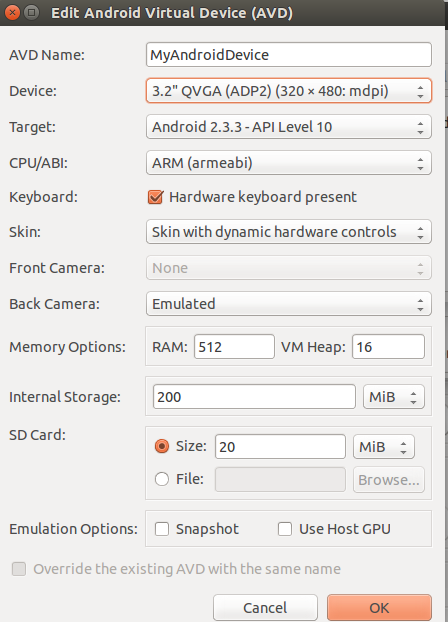
Installing packages can be done in the below window



Use Virtual device or the real hand held devices [MC40, TC70] for the testing with RPI

Press CTRL + ALT + T and type android avd to open the Android Virtual Device Manager.







Launching virtual Device



In case of our Raspberry PI we may not be able to run a virtual device accurately since of low memory and ARM based. We can directly plugin the android devices for the testing.

We can connect the devices [MC40, TC70, etc.] to PI with USB cable and open terminal ‘adb devices’ should give the list of devices connected to RPI.

**Appium Installation/configuration on RPI**

1. Install ruby: Paste the below command at terminal and hit enter

sudo apt-get install build-essential curl git m4 ruby texinfo libbz2-dev libcurl4-openssl-dev libexpat-dev libncurses-dev zlib1g-dev

2.Install linux brew:Paste the below command at terminal and hit enter

ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/linuxbrew/go/install)"

3.set path for brew

Type: gedit.bashrc at terminal and copy paste following into the .bashrc file

export PATH="$HOME/.linuxbrew/bin:$PATH"

export MANPATH="$HOME/.linuxbrew/share/man:$MANPATH"

export INFOPATH="$HOME/.linuxbrew/share/info:$INFOPATH"

4.Install node:Paste the below commands one by one at terminal and hit enter

brew update

brew install node

brew link node

5.Install appium

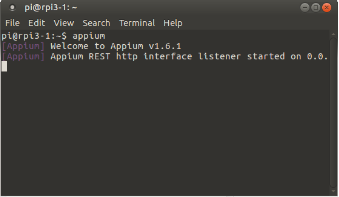
npm install -g appium

npm install wd

To start appium: Paste the below command at terminal and hit enter

appium

We don’t have any user Interface window for appium in Linux so we control appium through the terminal with server arguments.



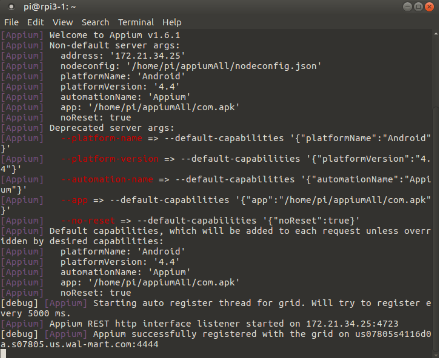
We register the hand held devices to the selenium grid to perform the testing from Jenkins CD Pipeline

We need to define the device capabilities and configuration of the device the machine where we host the appium server and then start the appium along with required attributes.

sudo appium -a 172.21.34.25 -p 4723 --app /home/pi/appiumAll/com.apk --app-pkg com.openstream.cueme.services.workbench --app-activity com.openstream.mmi.gui.CuemeActivityDelegate --platform-name Android --automation-name Appium --platform-version 4.4 --nodeconfig /home/pi/appiumAll/nodeconfig.json

………………………………………………………………………………………………………………………………………………………….

appium -a 172.21.34.25 -p 4723 --no-reset --app /home/pi/appiumAll/com.apk --platform-name Android --automation-name Appium --platform-version 4.4 --chromedriver-executable /home/pi/node\_modules/appium-chromedriver/chromedriver/linux/chromedriver\_64 --nodeconfig /home/pi/appiumAll/nodeconfig.json

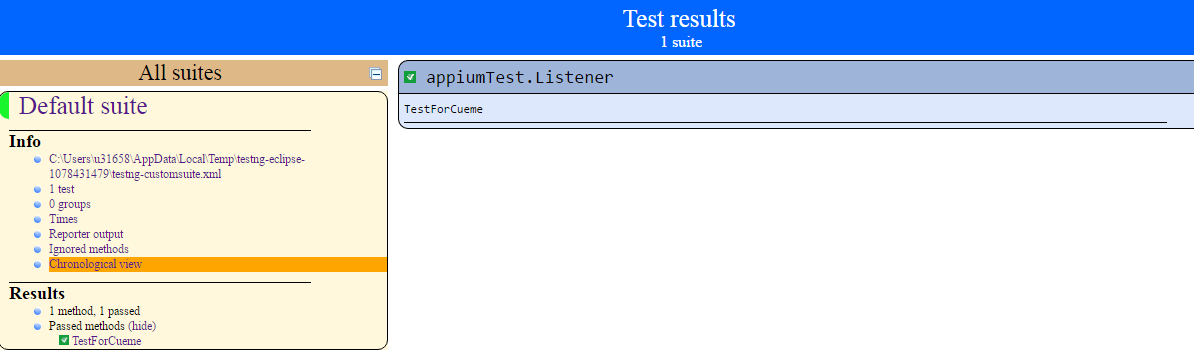


After the device registered to the selenium grid we can run the automation testing job from Jenkins. We can specify the target of the application deployed in various regions like DEV > TEST etc.

We can create multiple sessions with appium and can be run the tests on multiple devices in parallel.

Job executes the test scripts and uploads the testNG report to the corresponding boxes.

The application is capable of taking the screen-prints in case of any test failure and attach along with reports.



Sample test script code base [maven project] can be cloned here

<https://vcm.wal-mart.com/users/jamee1/repos/appiumtest/browse>

Known Issue

Chrome driver support issue in Linux ARM 32bit environment.