

This is a walkthrough on how to build an android test farm on ubuntu desktop that will work with an existing Jenkins installation.

Dependencies:

Ubuntu 16.04 Desktop (Server is probably fine too)

Installing OpenSTF (Smartphone Test Farm)

Install OpenSTF Dependencies (Note this was written on 10/31/2016)

1. Node (We will be installing NVM but feel free to install node directly from the repository)
 - In Terminal execute the following
 1. `sudo apt-get update`
 2. `sudo apt-get install -y build-essential libssl-dev`
 3. `curl -sL https://raw.githubusercontent.com/creationix/nvm/v0.31.0/install.sh | bash`
 4. `nvm ls-remote` (Optional step but it lists all available versions)
 5. `nvm install 6.9.1`
 - feel free to use latest here
 6. `nvm use 6.9.1`
 7. `nvm alias default 6.9.1`
 8. `nvm use default`
2. Android SDK
 1. Download the latest SDK <https://developer.android.com/studio/index.html>
 2. Install the required libs <https://developer.android.com/studio/install.html>
 1. `sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32bz2-1.0`
 3. Extract the sdk where ever you would like, I prefer home `/home/USER`
 4. Launch android studio, `sh ~/android-studio/bin/studio.sh`
 5. Walk through the installer, the sdk should be created at `/home/USER/Android/Sdk`
 - Note you will need this path later ^_^
3. This resource was great for setting up rethinkdb <https://www.rethinkdb.com/docs/install/ubuntu/>
 1. `source /etc/lsb-release && echo "deb http://download.rethinkdb.com/apt $DISTRIB_CODENAME main" | sudo tee /etc/apt/sources.list.d/rethinkdb.list`
 2. `wget -qO- https://download.rethinkdb.com/apt/pubkey.gpg | sudo apt-key add -`
 3. `sudo apt-get update`
 4. `sudo apt-get install rethinkdb`
4. ZeroMQ, another great install guide <https://tuananh.org/2015/06/16/how-to-install-zeromq-on-ubuntu/>
 1. `sudo apt-get install libtool pkg-config build-essential autoconf automake`
5. Install Libsodium
 1. `sudo apt-get install libzmq-dev`
 2. `git clone git://github.com/jedisct1/libsodium.git`
 3. `cd libsodium`
 4. `./autogen.sh`
 5. `./configure && make check`
 6. `sudo make install`
 7. `sudo ldconfig`
6. Install Zeromq
 1. `wget http://download.zeromq.org/zeromq-4.1.2.tar.gz`
 2. `tar -xvf zeromq-4.1.2.tar.gz`
 3. `cd zeromq-4.1.2`
 4. `./autogen.sh`

- If you see “autogen.sh: error: could not find libtool. libtool is required to run autogen.sh.” run the below command

- `sudo ln -s /usr/bin/libtoolize /usr/bin/libtool`

5. `./configure && make check`

6. `sudo make install`

7. `sudo ldconfig`

7. Now for the remaining dependencies

1. `sudo apt-get install -y graphicsmagick yasm pkg-config`

8. Path Time!

1. Edit your `~/.bash_profile` (it's ok if it doesn't exist yet) add the following lines

- `export ANDROID_HOME="/home/ubuntu/Android/Sdk" PATH=/home/ubuntu/Android/Sdk/tools:$PATH`
`PATH=/home/ubuntu/Android/Sdk/platform-tools:$PATH`

- (Optional, but recommended) Some devices may require an updated adbusini

- Link to the github <https://github.com/apkudo/adbusini>

1. `curl https://raw.githubusercontent.com/apkudo/adbusini/master/adb_usb.ini > ~/.android/adb_usb.ini`

9. And for the finally install open stf

1. `npm install -g stf`

- Note You might need to run rethinkdb in another terminal `rethinkdb --bind all`
- Note we will be setting up a service later, this is just to make sure everything is in working order

2. Check to see if it is all working in a browser at <http://localhost:7100/>

- Please make sure this step works before setting up the systemd services

Autostart

Technically you are all done, but if you would like to set up the tasks to autostart then this is the place to look.

Rethinkdb Service

1. Please create a file at this path `/lib/systemd/system/rethinkdb.service`

- The file should read similar to below

2. After the file is saved, let's start the service

1. `sudo systemctl start rethinkdb`

2. `sudo systemctl status rethinkdb`

- Just to make sure everything is working well
- `sudo systemctl enable rethinkdb`
- (Optional) this will start the service after restarts

```
[Unit]
Description=rethinkdb

[Service]
Restart=always
ExecStart=/usr/bin/rethinkdb --bind all

[Install]
WantedBy=default.target
```

OpenStf Service

1. Please create a file at this path `/lib/systemd/system/rethinkdb.service`

- The file should read similar to below, **please update User**
- After the file is saved, let's start the service

1. `sudo systemctl start openstf`

2. `sudo systemctl status openstf`

- Just to make sure everything is working well
- `sudo systemctl enable openstf`
- (Optional) this will start the service on restarts

```
[Unit]
Description=openstf

[Service]
User=<ACTUAL_USER>
Restart=always
Environment="PATH=/home/ubuntu/.nvm/versions/node/v6.7.0/bin:$PATH"
ExecStart=/home/ubuntu/.nvm/versions/node/v6.7.0/bin/stf local --public-ip <ACTUAL_IP>

[Install]
WantedBy=default.target
```

Jenkins Plugin

If you have jenkins and would like to use your fancy new device farm to make builds or to test, then this is the place. Thankfully all the hard work is done.

Install the Jenkins Plugin

1. install the plugin <https://wiki.jenkins-ci.org/display/JENKINS/Open+STF+Plugin>
2. Configure the Jenkins Plugin
 1. Go to Jenkins -> Configure
 2. Go to the Open STF Configuration
 3. Enter in the Open STF url
 4. Enter in the Open STF token (you can generate one in the open stf ui under Settings -> Keys)
 - If everything went well you should see no errors, go ahead and save the configuration
 5. Go to an existing job
 6. You should now see "Use a STF device during build" under "Build Environment"
 - Note I personally like to track Android jobs by the min sdk of the project, and unfortunately the jenkins plugin does not support that other than a fun regex
 - Example Regex `/1[6-9]|2[0-9]/`
 - This regex means api 16+
 - Feel free to use this to check your min api <http://www.regexpal.com/?fam=96044>
 7. And if everything is great you should see
 - ☒ Use a STF device during build

Conditions of the device you want to use

ConditionName

Value

Device list that might be used



Advanced Options