

O LAVA Test-Runner é uma ferramenta que lhe permite realizar testes de hardware e firmware.

O LAVA utiliza de scripts escritos em Shell script, yaml, e em alguns casos TCL script.

Esta ferramenta te permite testar o hardware tanto localmente quanto remotamente, via comunicação ssh.

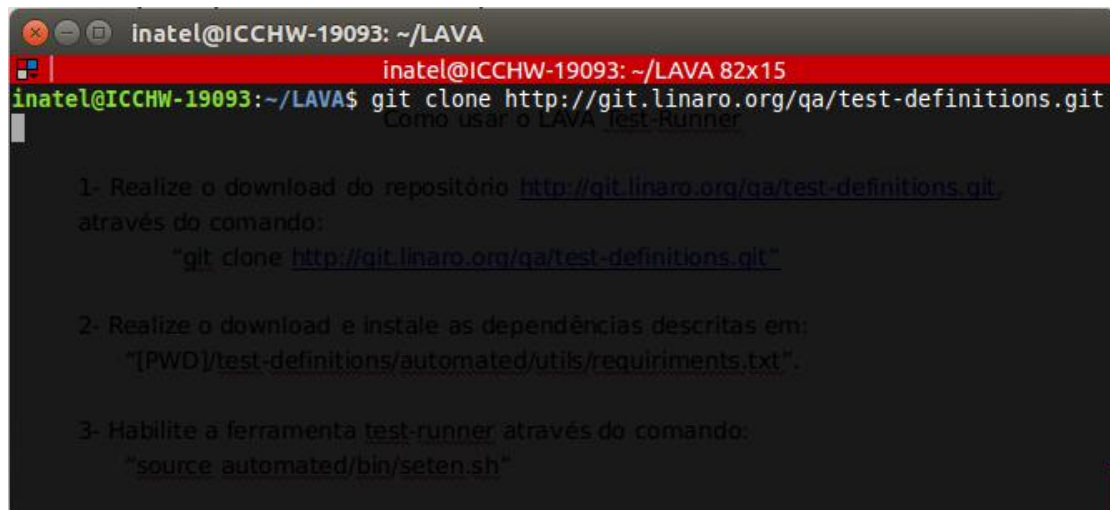
Uma leve documentação se encontra em:

<https://github.com/96boards/documentation/blob/master/ConsumerEdition/DragonBoard-410c/Guides/Testing.md>.

Como usar o LAVA Test-Runner

1- Realize o download do repositório <http://git.linaro.org/qa/test-definitions.git>, através do comando:

"git clone <http://git.linaro.org/qa/test-definitions.git>"



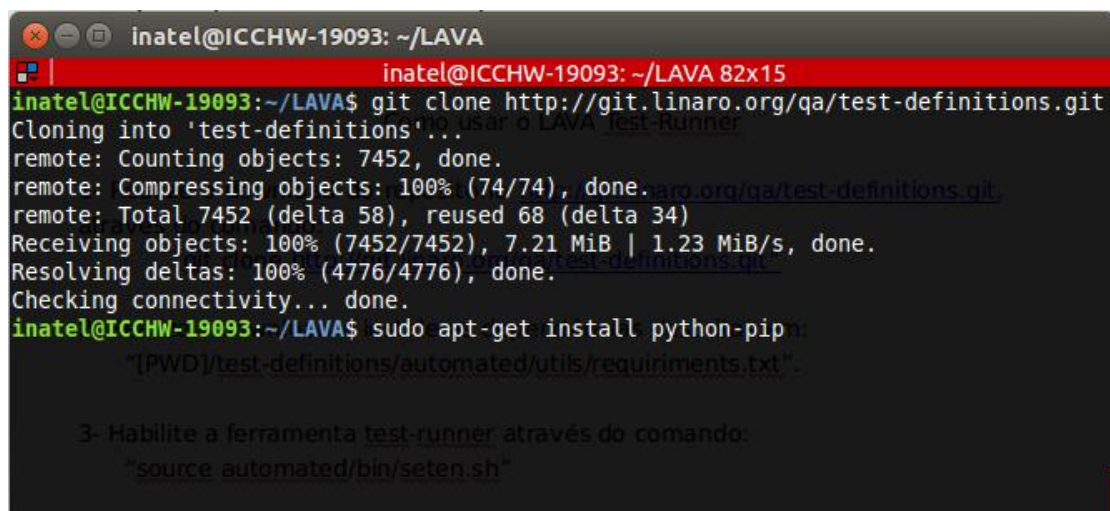
```
inatel@ICCHW-19093: ~/LAVA
inatel@ICCHW-19093: ~/LAVA 82x15
inatel@ICCHW-19093:~/LAVA$ git clone http://git.linaro.org/qa/test-definitions.git
Cloning into 'test-definitions'...
```

1- Realize o download do repositório <http://git.linaro.org/qa/test-definitions.git>, através do comando:

"git clone <http://git.linaro.org/qa/test-definitions.git>"

2- Realize o download e instale as dependências descritas em:

"[PWD]/test-definitions/automated/utlis/requiriments.txt".



```
inatel@ICCHW-19093: ~/LAVA
inatel@ICCHW-19093: ~/LAVA 82x15
inatel@ICCHW-19093:~/LAVA$ git clone http://git.linaro.org/qa/test-definitions.git
Cloning into 'test-definitions'...
remote: Counting objects: 7452, done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 7452 (delta 58), reused 68 (delta 34)
Receiving objects: 100% (7452/7452), 7.21 MiB | 1.23 MiB/s, done.
Resolving deltas: 100% (4776/4776), done.
Checking connectivity... done.
inatel@ICCHW-19093:~/LAVA$ sudo apt-get install python-pip
Reading package lists... Done
Building dependency tree... Done
The following packages will be installed:
  python-pip
The following NEW packages will be installed:
  python-pip
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 19.5 kB of archives.
After this operation, 56.1 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2- Realize o download e instale as dependências descritas em:

"[PWD]/test-definitions/automated/utlis/requiriments.txt".

3- Habilite a ferramenta `test-runner` através do comando:

"source automated/bin/seten.sh"

```
inatel@ICCHW-19093: ~/LAVA/test-definitions
inatel@ICCHW-19093: ~/LAVA/test-definitions 82x15
1.5.4-lubuntu4 [97,3 kB] para trabalhar com um projeto com as dependências
Baixados 140 kB em 0s (205 kB/s)
A seleccionar pacote anteriormente não seleccionado python-requests.
(Lendo banco de dados ... 289972 ficheiros e directórios actualmente instalados.)
A preparar para desempacotar .../python-requests_2.2.1-lubuntu0.3_all.deb ...
A descompactar python-requests (2.2.1-lubuntu0.3) ...
A seleccionar pacote anteriormente não seleccionado python-pip.
A preparar para desempacotar .../python-pip_1.5.4-lubuntu4_all.deb ...
A descompactar python-pip (1.5.4-lubuntu4) ...
A processar 'triggers' para man-db (2.6.7.1-lubuntu1) ...
Configurando python-requests (2.2.1-lubuntu0.3) ...
Configurando python-pip (1.5.4-lubuntu4) ...
inatel@ICCHW-19093:~/LAVA$ cd test-definitions/
inatel@ICCHW-19093:~/LAVA/test-definitions$ pip install -r automated/utls/requirements.txt
```

3- Habilite a ferramenta test-runner através do comando:

“source automated/bin/seten.sh”

```
inatel@ICCHW-19093: ~/LAVA/test-definitions
inatel@ICCHW-19093: ~/LAVA/test-definitions 82x15
Requirement already satisfied (use --upgrade to upgrade): requests in /usr/lib/python2.7/dist-packages (from -r automated/utls/requirements.txt (line 3))
Downloading/unpacking ptyprocess>=0.5 (from pexpect->-r automated/utls/requirements.txt (line 1))
  Downloading ptyprocess-0.5.2-py2.py3-none-any.whl
Installing collected packages: pexpect, ptyprocess
Compiling /tmp/pip_build_root/pexpect/pexpect/async.py ...
File "/tmp/pip_build_root/pexpect/pexpect/async.py", line 16
    transport, pw = yield from asyncio.get_event_loop()\
                    ^
SyntaxError: invalid syntax
Successfully installed pexpect ptyprocess
Cleaning up...
inatel@ICCHW-19093:~/LAVA/test-definitions$ source automated/bin/setenv.sh
```

4- Agora é hora de executar nosso teste, para isso podemos realizar de duas formas.

Lembrando que para realizar a execução é aconselhável estar no diretório “....../test-definitions/”

4.1- Executar o teste localmente, para executar desta forma devemos utilizar o padrão a seguir:

“test-runner [token] PATH/arquivo.yaml”

Onde o token descrito deverá ser substituído por -d (realiza a execução de um único teste) ou por -p (realiza a execução de um plano de testes).

O PATH é o caminho até chegar aonde está o arquivo.


```
inatel@ICCHW-19093: ~/LAVA/test-definitions
inatel@ICCHW-19093: ~/LAVA/test-definitions 82x15
Compiling /tmp/pip_build_root/pexpect/pexpect/async.py ...
File "/tmp/pip_build_root/pexpect/pexpect/async.py", line 16
    transport, pw = yield from asyncio.get_event_loop()\
                        ^
SyntaxError: invalid syntax Como usar o LAVA Test-Runner

Successfully installed pexpect ptyprocess
Cleaning up...
inatel@ICCHW-19093:~/LAVA/test-definitions$ source automated/bin/setenv.sh
REPO PATH: /home/inatel/LAVA/test-definitions/definitions.git
BIN PATH: /home/inatel/LAVA/test-definitions/automated/bin:/home/inatel/LAVA/test-definitions/automated/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
inatel@ICCHW-19093:~/LAVA/test-definitions$ test-runner -d automated/linux/smoke/s
smoke.yaml
```

4.2- Executar o teste remotamente (em outra plataforma a partir da plataforma atual), para executar desta forma devemos utilizar o a padrão a seguir:

“test-runner -g root@[IP] [token] PATH/arquivo.yaml”

Onde o IP é o endereço ip da plataforma destino.

O token descrito deverá ser substituído por -d (realiza a execução de um único teste) ou por -p (realiza a execução de um plano de testes).

O PATH é o caminho até chegar aonde está o arquivo.

```
inatel@ICCHW-19093: ~/LAVA/test-definitions
inatel@ICCHW-19093: ~/LAVA/test-definitions 82x15
Compiling /tmp/pip_build_root/pexpect/pexpect/async.py ...
File "/tmp/pip_build_root/pexpect/pexpect/async.py", line 16
    transport, pw = yield from asyncio.get_event_loop()\
                        ^
SyntaxError: invalid syntax Como usar o LAVA Test-Runner

Successfully installed pexpect ptyprocess
Cleaning up...
inatel@ICCHW-19093:~/LAVA/test-definitions$ source automated/bin/setenv.sh
REPO PATH: /home/inatel/LAVA/test-definitions/definitions.git
BIN PATH: /home/inatel/LAVA/test-definitions/automated/bin:/home/inatel/LAVA/test-definitions/automated/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
inatel@ICCHW-19093:~/LAVA/test-definitions$ test-runner -g root@192.168.28.45 -d
automated/linux/smoke/smile.yaml
```

5- Todos os teste escritos devem estar separados por pastas, e quando for executar o teste deve-se colocar a pasta dentro do diretório:

“...../test-definitions/automated/linux/”

OBS.: Estará disponível um código exemplo no endereço (colocarei um link para uma página do github ou algo do tipo) para que seja de fácil compreensão a forma adequada de se escrever o arquivo yaml, e uma forma de se criar os arquivos shell script e tcl script.