



LAVA External Device Monitoring

1	Contents	
2	Test Cases	2
3	LAVA Features	2
4	LXC	3
5	MultiNode	3
6	Secondary Connections	3
7	Approach Overview	3
8	LAVA Job Connection Layout	3
9	Test Job	4
10	Job File Example	5
11	QA Report	10
12	This document describes how to execute automated LAVA tests controlling re-	
13	sources external to the DUT across a network implementing a LAVA parallel	
14	pipeline job.	

15 **Test Cases**

16 The approach proposed in this document will help to address test cases like:

- 17 • Executing a test in the DUT where certain power states are simulated (for
- 18 example a power loss) during specific test actions using a programmable
- 19 PSU external to the DUT.
- 20 • Executing a test in the DUT simulating SD card insertion and removal
- 21 using an external device.

22 The only assumption, in both scenario, proposed in this document is that the

23 external device (either a programmable PSU or SD-card simulator) can be ac-

24 cessed through the network using SSH.

25 **LAVA Features**

26 LAVA offers the following features that can be combined to implement a solution

27 for the test cases mentioned in this document:

- 28 • LXC to deploy required software and tools to access the external device.
- 29 • MultiNode to communicate data between jobs actions.
- 30 • Secondary connections for executing tests through SSH.

31 LXC

32 LAVA supports LXC containers both as a standalone device type and as dynamic
33 transparent environments in order to interact with external devices. In either
34 case the [LXC Protocol](#)¹ is used.

35 MultiNode

36 The [MultiNode Protocol](#)² allows data to be shared between actions, including
37 data generated in one test shell definition being made available over the protocol
38 to a deploy or boot action of jobs with a different role.

39 Synchronisation is done using the MultiNode API, specifically the `lava-send` and
40 `lava-wait` calls.

41 Secondary Connections

42 LAVA allows [Secondary Connections](#)³ to open network connections to external
43 devices using MultiNode submissions.

44 Approach Overview

45 The main idea is to create a LXC container device associated to the DUT
46 responsible to execute the automated test, then opens a SSH connection to
47 an external device, and use the MultiNode API in order to synchronize both
48 devices and pass data between them with the LXC container serving like a
49 coordinator of the different LAVA tests actions.

50 In this way, a server-client layout is setup that will help to execute tests in
51 a board attached to LAVA (server side) with intervention of external devices
52 (client side).

53 LAVA Job Connection Layout

54 The LXC container is deployed directly from the LAVA dispatcher and coordi-
55 nate the execution of the parallel pipeline between the DUT and the external
56 device (secondary connection) from there.

57 The layout model would be something like:

```
58         ----- DUT
59         /  MultiNode
60 LAVA (LXC)
61         \
```

¹<https://lava.collabora.dev/static/docs/v2/actions-protocols.html#lxc-protocol-reference>

²<https://lava.collabora.dev/static/docs/v2/actions-protocols.html#multinode-protocol>

³<https://lava.collabora.dev/static/docs/v2/pipeline-writer-secondary.html>

62 ----- Secondary Connection (PSU, SD-Card HW)
63 MultiNode

64 Test Job

65 This section shows the basics proposed in this document using a LAVA job file
66 example.

67 The following steps describe the main flow of the job:

68 1 - Create two types of roles `host` and `guest`. The `host` role will contain the LXC
69 container and the DUT, the `guest` role will label the SSH connection for the
70 external device. This creates two groups (`host` and `guest`) that can communicate
71 using the MultiNode API, so messages can be sent between the LXC and Device
72 as the server and the secondary connection as the client.

73 2 - Label both types of roles in the `protocols` section of the job.

74 3 - Deploy and boot the LXC container (`host`).

75 4 - Execute a test in the LXC container using the MultiNode API to send the
76 `lava_start` message, so the `deploy` action for the external device can start, and
77 waits for remaining clients to start using the `lava-sync` call.

78 5 - Deploy the DUT (`host`).

79 6 - Deploy the external device (`guest`) , which is waiting for the LXC `lava_start`
80 message to start deployment. Once this message is received, the guest device is
81 deployed.

82 7 - Boot DUT.

83 8 - Boot external device.

84 9 - Execute a test in the DUT sending the `lava-sync` call.

85 10 - Execute a test in the external device sending the `lava-sync` call.

86 11 - Once all clients are synchronized (the LXC, DUT and external device), start
87 executing tests.

88 12 - Tests executed in the DUT and external device needs to use the [MultiN-](#)
89 [odeAPI](#)⁴ in order to pass data between them.

90 As the LXC is deployed and booted first, the LXC can run a test shell before
91 deploying the device, before booting the device, before the test shell action on
92 the device which starts the secondary connection guests or at any later point
93 ([AddingTestsActions](#)⁵).

⁴<https://lava.collabora.dev/static/docs/v2/multinodeapi.html#multinode-api>

⁵<https://lava.collabora.dev/static/docs/v2/writing-multinode.html#adding-test-actions>

94 Job File Example

```
95 job_name: LXC and Secondary connection with a Device
96
97 timeouts:
98   job:
99     minutes: 30
100   action:
101     minutes: 3
102   connection:
103     minutes: 5
104 priority: medium
105 visibility: public
106
107 protocols:
108   lava-lxc:
109     host:
110       name: lxc-ssh-test
111       template: debian
112       distribution: debian
113       release: stretch
114   lava-multinode:
115     # expect_role is used by the dispatcher and is part of delay_start
116     # host_role is used by the scheduler, unrelated to delay_start.
117     roles:
118       host:
119         device_type: beaglebone-black
120       # This makes this role essential in order to execute the test.
121       essential: True
122         count: 1
123         timeout:
124           minutes: 10
125       guest:
126         # protocol API call to make during protocol setup
127         request: lava-start
128         # set the role for which this role will wait
129         expect_role: host
130         timeout:
131           minutes: 15
132         # no device_type, just a connection
133         connection: ssh
134         count: 3
135         # each ssh connection will attempt to connect to the device of role 'host'
136         host_role: host
137
138 actions:
```

```

139 - deploy:
140     role:
141     - host
142     namespace: probe
143     timeout:
144         minutes: 5
145     to: lxc
146     # authorize for ssh adds the ssh public key to authorized_keys
147     authorize: ssh
148     packages:
149     - usbutils
150     - procps
151     - lsb-release
152     - util-linux
153     - ntpdate
154     - openssh-server
155     - net-tools
156
157 - boot:
158     role:
159     - host
160     namespace: probe
161     prompts:
162     - 'root@(.*):/#'
163     timeout:
164         minutes: 5
165     method: lxc
166
167 - test:
168     role:
169     - host
170     namespace: probe
171     timeout:
172         minutes: 5
173     definitions:
174     - repository:
175         metadata:
176             format: Lava-Test Test Definition 1.0
177             name: network
178             description: "Send message ID"
179         run:
180             steps:
181             - lava-test-case ntpdate --shell ntpdate-debian
182             - lava-echo-ipv4 eth0
183             - lava-send ipv4 ipaddr=$(lava-echo-ipv4 eth0)
184             - lava-send lava_start

```

```

185         - lava-sync clients
186         from: inline
187         name: lxc-test
188         path: inline/lxc-test.yaml
189
190 # DUT actions
191 - deploy:
192     role:
193     - host
194     namespace: device
195     timeout:
196         minutes: 5
197     to: tftp
198
199     kernel:
200         url: https://files.lavasoftware.org/components/lava/standard/debian/stretch/armhf/3/vmlinuz-
201 4.9.0-4-armmp
202         sha256sum: b6043cc5a07e2cead3f7f098018e7706ea7840eece2a456ba5fcfaddaf98a21e
203         type: zimage
204     ramdisk:
205         url: https://files.lavasoftware.org/components/lava/standard/debian/stretch/armhf/3/initrd.img-
206 4.9.0-4-armmp
207         sha256sum: 4cc25f499ae74e72b5d74c9c5e65e143de8c2e3b019f5d1781abbf519479b843
208         compression: gz
209     modules:
210         url: https://files.lavasoftware.org/components/lava/standard/debian/stretch/armhf/3/modules.tar.gz
211         sha256sum: 10e6930e9282dd44905cfd3f3a2d5a5058a1d400374afb2619412554e1067d58
212         compression: gz
213     nfsrootfs:
214         url: https://files.lavasoftware.org/components/lava/standard/debian/stretch/armhf/3/stretch-
215 armhf-nfs.tar.gz
216         sha256sum: 46d18f339ac973359e8ac507e5258b620709add94cf5e09a858d936ace38f698
217         compression: gz
218     dtb:
219         url: https://files.lavasoftware.org/components/lava/standard/debian/stretch/armhf/3/dtbs/am335x-
220 boneblack.dtb
221         sha256sum: c4c461712bf52af7d020e78678e20fc946f1d9b9552ef26fd07ae85c5373ece9
222
223 - deploy:
224     role:
225     - guest
226     namespace: guest
227     # Timeout for the ssh connection attempt
228     timeout:
229         seconds: 30
230     to: ssh

```

```

231     connection: ssh
232     protocols:
233         lava-multinode:
234             - action: prepare-scp-overlay
235               request: lava-wait
236               messageID: ipv4
237               message:
238                   ipaddr: $ipaddr
239               timeout: # delay_start timeout
240               minutes: 5
241
242 - boot:
243     role:
244         - host
245     namespace: device
246     timeout:
247         minutes: 15
248     method: u-boot
249     commands: nfs
250     auto_login:
251         login_prompt: 'login:'
252         username: root
253     prompts:
254         - 'root@stretch:'
255     parameters:
256         shutdown-message: "reboot: Restarting system"
257
258 - boot:
259     role:
260         - guest
261     namespace: guest
262     timeout:
263         minutes: 3
264     prompts:
265         - 'root@stretch:'
266     parameters:
267         hostID: ipv4
268         host_key: ipaddr
269     method: ssh
270     connection: ssh
271
272 - test:
273     role:
274         - host
275     namespace: device
276     timeout:

```

```

277         minutes: 30
278     definitions:
279     - repository:
280         metadata:
281             format: Lava-Test Test Definition 1.0
282             name: install-ssh
283             description: "install step"
284         run:
285             steps:
286                 - df -h
287                 - free
288                 - lava-sync clients
289         from: inline
290         name: ssh-inline
291         path: inline/ssh-install.yaml
292     - repository: http://git.linaro.org/lava-team/lava-functional-tests.git
293         from: git
294         path: lava-test-shell/smoke-tests-basic.yaml
295         name: smoke-tests
296     - repository: http://git.linaro.org/lava-team/lava-functional-tests.git
297         from: git
298         path: lava-test-shell/single-node/singlenode02.yaml
299         name: singlenode-intermediate
300
301     - test:
302         role:
303             - guest
304         namespace: guest
305         timeout:
306             minutes: 5
307         definitions:
308             - repository: http://git.linaro.org/lava-team/lava-functional-tests.git
309                 from: git
310                 path: lava-test-shell/smoke-tests-basic.yaml
311                 name: smoke-tests
312             # run the inline last as the host is waiting for this final sync.
313             - repository:
314                 metadata:
315                     format: Lava-Test Test Definition 1.0
316                     name: client-ssh
317                     description: "client complete"
318                 run:
319                     steps:
320                         - df -h
321                         - free
322                         - lava-sync clients

```

```

323         from: inline
324         name: ssh-client
325         path: inline/ssh-client.yaml
326
327     #
328     # Tests executed in the external device and DUT can be added here.
329     # They all need to use the MultiNode API.
330     #
331
332     # Execute test in the DUT
333     - test:
334         role:
335             - host
336         namespace: device
337         timeout:
338             minutes: 10
339         definitions:
340             - repository: https://gitlab.apertis.org/tests/apertis-test-cases/
341               from: git
342               path: lava-test-shell/single-node/singlenode03.yaml
343               name: singlenode-advanced
344
345     # Execute test in the external device (PSU, SD-card device)
346     - test:
347         role:
348             - guest
349         namespace: guest
350         timeout:
351             minutes: 10
352         definitions:
353             - repository: https://gitlab.apertis.org/tests/apertis-test-cases/
354               from: git
355               path: lava-test-shell/single-node/singlenode03.yaml
356               name: singlenode-advanced

```

357 QA Report

358 Once tests results are available at LAVA , and the test cases are enabled for the
359 specific images from the test case repository, the results will be available from
360 the QA Report App automatically.