

How to write your own tests

...

(not only for AGL)

Prerequisites

Laptop

- with a Linux distro of your choice
- A text editor
- Python version $\geq 2.7.1$
 - apt-get install python
 - apt-get install python-pip
- python-jinja ≥ 2.9
 - pip install --upgrade jinja2

Internet access

- github (or any publicly accessible git) account
- that you can use to create a publicly visible project and upload to

Local LAN

- WiFi: *baylibre* pwd: *lavabaylibre*
- ping lavabox

Local LAVA

- LAVA instance: <http://lavabox:10080/>
- LAVA user: demo
- LAVA auth token: tokendemo

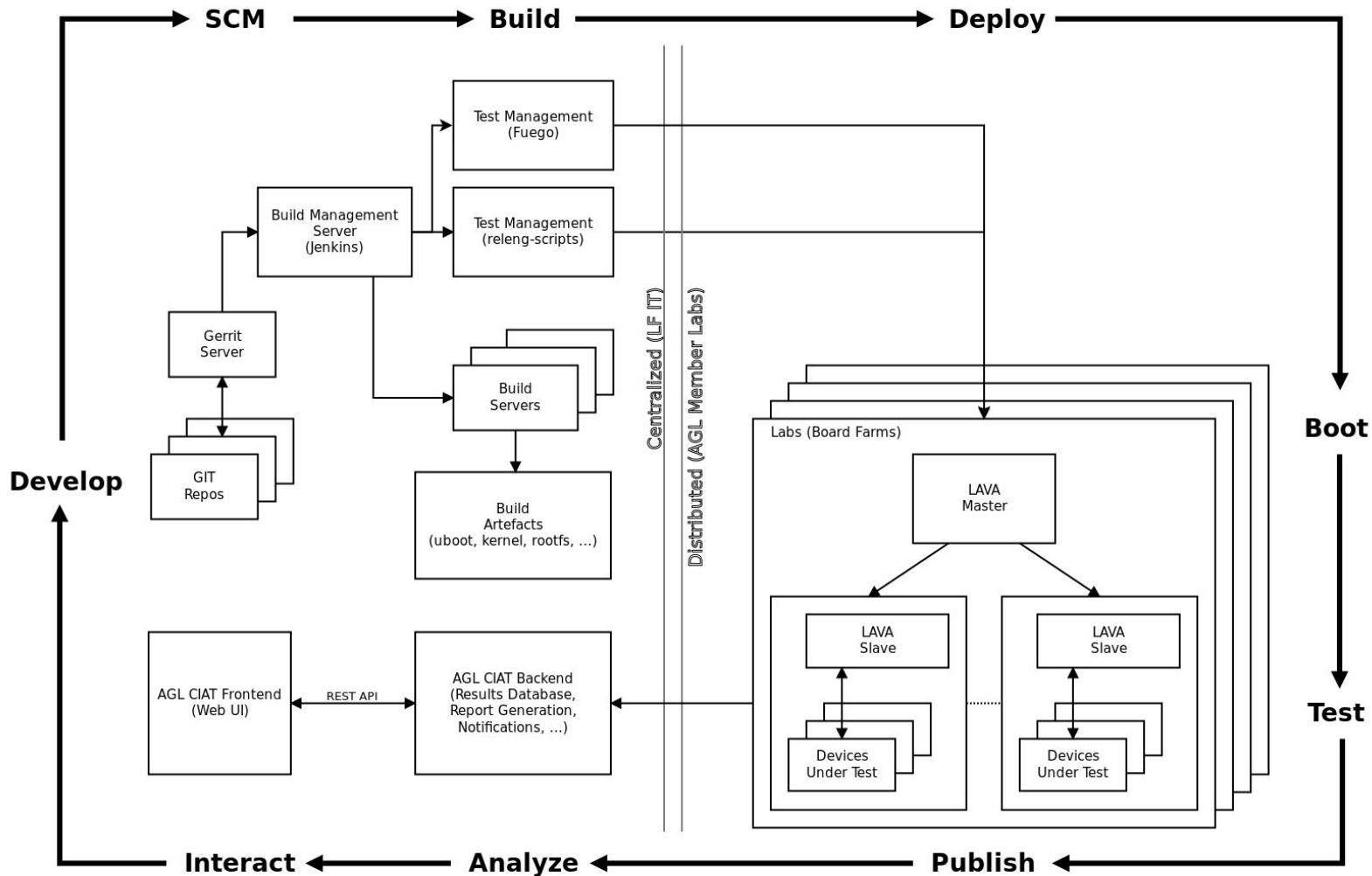
Local kernel CI frontend

- <http://lavabox:8080/>

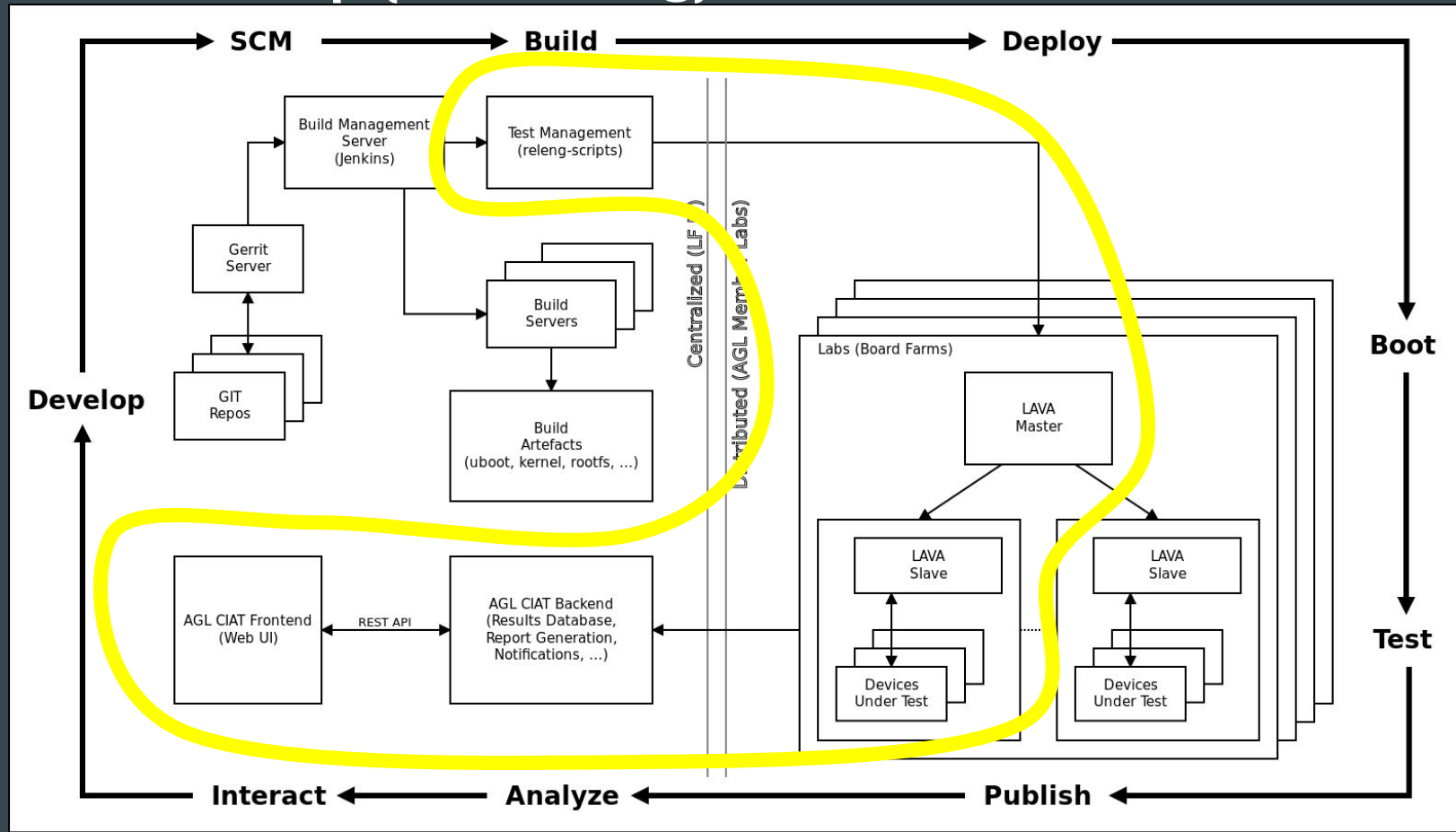
AGL CIAT overview

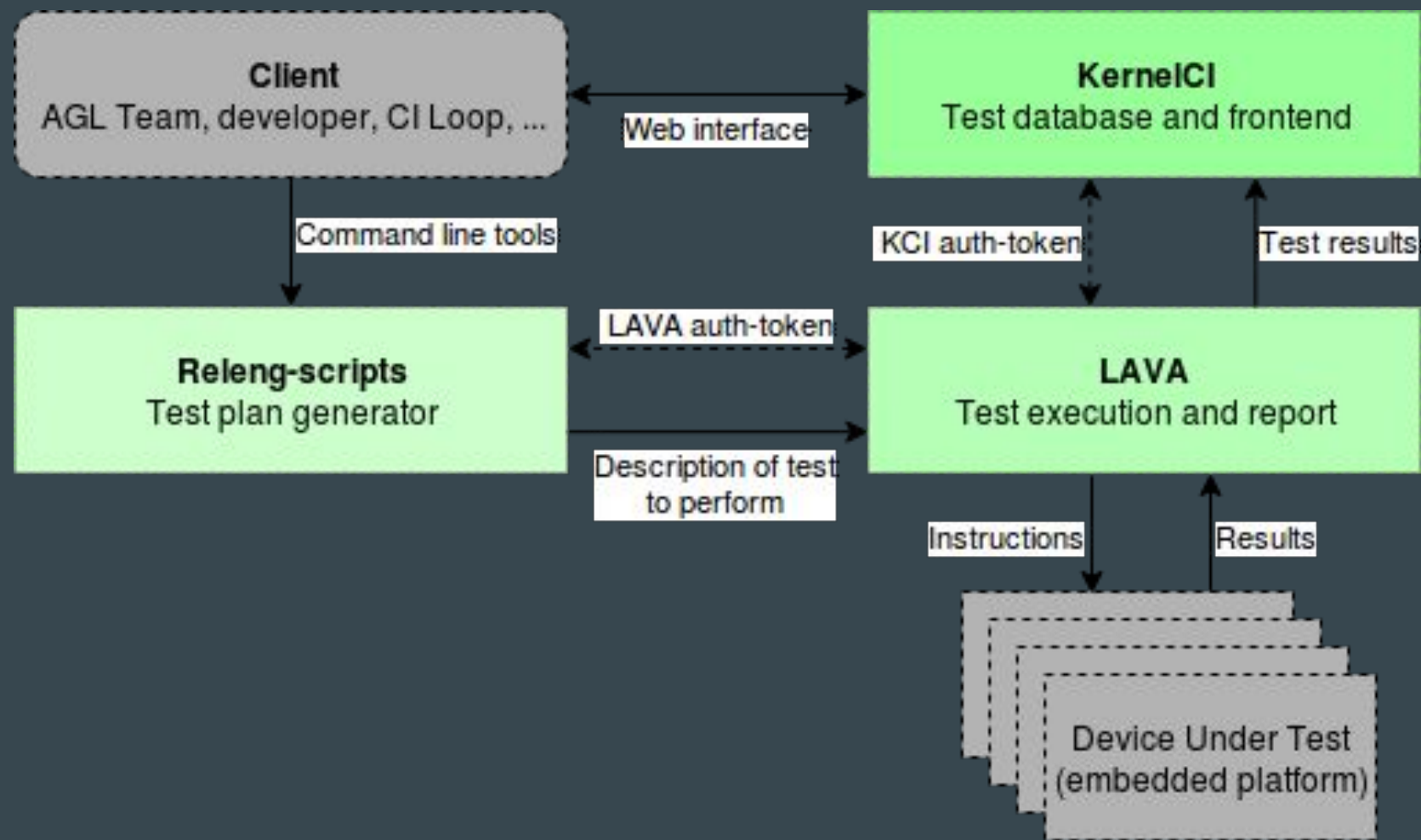


AGL CIAT loop



AGL CIAT loop (for training)





Submitting jobs to LAVA

...

LAVA: test “jobs”

YAML format with multiple sections

- timeouts
- actions (deploy, boot, test)
- protocols
- device-type
- notify

NOTE: we won't be writing these from scratch, we'll be using tools based on templates.

```
2 priority: medium
3 visibility: public
4
5 notify:
6   criteria:
7     status: finished
8   callback:
9     url: http://api.dev.baylibre.com/callback/lava/test?lab_name=lab-baylibre-dev&status={STATUS}
10    method: POST
11    dataset: all
12    token: 12345678-XXXX-XXXX-XXXX-123456789012
13    content-type: json
14
15 device_type: r8a7796-m3ulcb
16
17 timeouts:
18   job:
19     minutes: 30
20   action:
21     minutes: 15
22   connection:
23     minutes: 5
24
25 protocols:
26   lava-xnbd:
27     port: auto
28
29 actions:
30 - deploy:
31   timeout:
32     minutes: 15
33   to: nbd
34   os: oe
35   failure_retry: 2
36   kernel:
37     url: http://www.baylibre.com/pub/agl/ci/m3ulcb-nogfx/Image
38   initrd:
39     url: http://www.baylibre.com/pub/agl/ci/m3ulcb-nogfx/initramfs-netboot-image-m3ulcb.ext4.gz
40     allow_modify: false
41   nbdroot:
42     url: http://www.baylibre.com/pub/agl/ci/m3ulcb-nogfx/core-image-minimal-m3ulcb.ext4.xz
43     compression: xz
44   dtb:
45     url: http://www.baylibre.com/pub/agl/ci/m3ulcb-nogfx/Image-r8a7796-m3ulcb.dtb
46
47 - boot:
48   timeout:
49     minutes: 10
50   method: u-boot
51   prompts: ["root@m3ulcb:~"]
52   auto_login:
53     login_prompt: "login:"
54     username: root
55     type: booti
56     commands: nbd
57     transfer_overlay:
58       download_command: wget
59       unpack_command: tar -C / -xvpf
60
61 - test:
62   definitions:
63     - repository: git://git.linaro.org/qa/test-definitions.git
64     from: git
65     path: automated/linux/smoke/smoke.yaml
66     name: smoke-tests
67
```


Generating LAVA test jobs

Get the tools:

```
$ git clone https://git.automotivelinux.org/AGL/releng-scripts/  
$ cd releng-scripts  
$ git checkout -b lavabox origin/lavabox
```

Generate example jobs:

```
$ ./utils/create-jobs.py --machine m3ulcb --url http://lavabox/ -o  
myjob.yaml
```

Take a look !

This time with some tests:

```
$ ./utils/create-jobs.py --machine m3ulcb --url http://lavabox/ -o  
myjob.yaml --test all
```

Take another look !

Submit jobs to LAVA

Authenticate with LAVA server (one-time setup)

```
$ apt-get install lava-tool
```

```
$ lava-tool auth-add http://demo@lavabox:10080/RPC2/  
Enter token for http://demo@lavabox:10080/RPC2/:  
Token added successfully for user demo.
```

Enter this
token: **tokendemo**

Submit

```
$ lava-tool submit-job http://demo@lavabox:10080/RPC2/ myjob.yaml  
submitted as job: http://lavabox:10080/scheduler/job/16
```

Keyring problems?

```
cat ~/.local/share/python_keyring/keyringrc.cfg  
[backend]  
default-keyring=keyring.backends.file.PlaintextKeyring
```

Test plans

Written as YAML templates, included into LAVA job.

A collection of “test suites”

Basic structure of a test plan (c.f. `releeng-scripts/templates/tests`)

```
- test:
  definitions:
    # The test suite definition that will be parsed and executed goes here
- test:
  definitions:
    # A second test suite
- test:
  definitions:
    # A third test suite, all part of the same test plan
```

Example test plan: remote git repo

Inside the test plan template
(local: releng-scripts/tests/templates)

```
- test:
  timeout:
    minutes: 2
  definitions:
  - repository: git://github.com/baylibre/agl-test-definitions.git
    from: git
    path: examples/basic-inline.yaml
    name: test-example-basic-inline
```

Inside the remote git repo:

```
metadata:
  name: metadataname-basictest
  format: "Lava-Test-Shell Test Definition 1.0"
  description: "A basic test definition."
run:
  steps:
  - lava-test-set start constant
  - echo "Hello"
  - lava-test-case always-pass --result pass
  - lava-test-set stop constant
```

Example test plan: simple, inline commands

```
- test:
  timeout:
    minutes: 2
  definitions:
  - repository:
    metadata:
      name: metadataname-basictest
      format: "Lava-Test-Shell Test Definition 1.0"
      description: "A basic test definition."
    run:
      steps:
        - lava-test-set start basic-test-set
        - echo "Hello"
        - lava-test-case always-pass --result pass
        - lava-test-set stop basic-test-set
  from: inline
  name: test-example-basictest
  path: inline/basictest.yaml
```

Starts a "test suite"

Start a "test set"

Test case

Stop a "test set"

Viewing test results

...

LAVA callbacks

LAVA v2 allows any job to send results via HTTP GET/PUT

Allows publishing / pushing results as to external tools/service as soon as job is finished.

External services can require authentication

```
notify:
  criteria:
    status: finished
  callback:
    url: <URL>
    method: POST
    dataset: all
    token: <auth token>
    content-type: json
```

Example job with callbacks

This time with callbacks:

```
$ ./utils/create-jobs.py m3ulcb --urlbase demo -o myjob.yaml --test all  
--callback lab-baylibre-lavabox
```

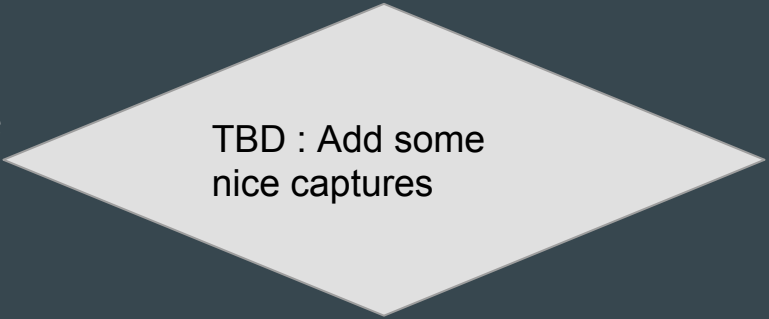


Look for 'notify' section

- Jobs w/callbacks
- kernelCI backend
- kerneCI front-end

Next steps

- Updated, modern web UI
 - Flexible access to data results
- Better access to logs
 - e.g. jump directly to the relevant part from a test case
- Searchable logs
- Add yocto “ptest”



TBD : Add some nice captures

When LAVAbbox is gone...

...

Moving from LAVAbbox to AGL core lab

AGL LAVA server: <http://lava.automotivelinux.org/>

- Need account (and auth-token) on LAVA server for today
 - User: AMMDemo
 - Token: <ask instructor>

Detailed instructions for hands-on part:

<https://goo.gl/R8ZUVJ>