SQUAD: an overview of the platform behind qa-reports.linaro.org

Antonio Terceiro
QA Services
The qa-reports.linaro.org service

- Provided by QA Services
- Currently in (open) BETA
- Current in use by LITE, LKFT, ERP, and QA Services
- Features
  - Report test results and metrics
  - Metric history charts
  - CI loop integration (esp. LAVA)
  - Comparison across builds (regressions, changes, etc)
  - Reports by email
<table>
<thead>
<tr>
<th>Project</th>
<th>Total</th>
<th>Passed</th>
<th>Skipped</th>
<th>Failed</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>lite/zephyr-upstream</td>
<td>111</td>
<td>108</td>
<td>3</td>
<td></td>
<td>1 week ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sun, 18, 2017, 9:25 p.m.</td>
</tr>
<tr>
<td>lite/zephyr-upstream-arm</td>
<td>307</td>
<td>304</td>
<td>3</td>
<td></td>
<td>4 days, 6 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wed, 21, 2017, 2:30 p.m.</td>
</tr>
<tr>
<td>lkft/android-hikey-linaro-4.4-aosp</td>
<td>331</td>
<td>319</td>
<td>12</td>
<td></td>
<td>3 days, 17 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Thu, 22, 2017, 4:27 a.m.</td>
</tr>
<tr>
<td>lkft/android-hikey-linaro-4.4-oe</td>
<td>1244</td>
<td>1092</td>
<td>129</td>
<td>23</td>
<td>3 days, 16 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Thu, 22, 2017, 5:08 a.m.</td>
</tr>
<tr>
<td>lkft/android-hikey-linaro-4.9-aosp</td>
<td>333</td>
<td>329</td>
<td>4</td>
<td></td>
<td>3 days, 18 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Thu, 22, 2017, 3:29 a.m.</td>
</tr>
<tr>
<td>lkft/android-hikey-linaro-4.9-oe</td>
<td>1245</td>
<td>1190</td>
<td>130</td>
<td>15</td>
<td>4 days, 3 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fri, 21, 2017, 5:42 p.m.</td>
</tr>
<tr>
<td>lkft/linux-mainline-oe</td>
<td>1272</td>
<td>1072</td>
<td>176</td>
<td>24</td>
<td>1 week, 2 days ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sat, 16, 2017, 8:43 a.m.</td>
</tr>
<tr>
<td>lkft/linux-mainline-oe</td>
<td>1272</td>
<td>1072</td>
<td>176</td>
<td>24</td>
<td>1 week, 2 days ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sat, 16, 2017, 8:43 a.m.</td>
</tr>
<tr>
<td>lkft/linux-mainline-oe</td>
<td>1272</td>
<td>1072</td>
<td>176</td>
<td>24</td>
<td>1 week, 2 days ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sat, 16, 2017, 8:43 a.m.</td>
</tr>
</tbody>
</table>
Last build - 544 Sept. 25, 2017, 8:03 p.m. an hour ago

**Latest builds**

<table>
<thead>
<tr>
<th>Build</th>
<th>Date</th>
<th>Test jobs</th>
<th>Test results summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>544</td>
<td>Sept. 25, 2017, 8:03 p.m. an hour ago</td>
<td>4 total 4 completed</td>
<td>816 total 816 passed</td>
</tr>
<tr>
<td>541</td>
<td>Sept. 21, 2017, 9:41 p.m. 3 days, 23 hours ago</td>
<td>4 total 4 completed</td>
<td>544 total 544 passed</td>
</tr>
<tr>
<td>539</td>
<td>Sept. 21, 2017, 7:36 p.m. 4 days, 1 hour ago</td>
<td>4 total 4 completed</td>
<td>816 total 816 passed</td>
</tr>
<tr>
<td>536</td>
<td>Sept. 21, 2017, 5:41 p.m. 4 days, 3 hours ago</td>
<td>4 total 4 completed</td>
<td>816 total 816 passed</td>
</tr>
<tr>
<td>534</td>
<td>Sept. 21, 2017, 11:40 a.m. 4 days, 9 hours ago</td>
<td>4 total 4 completed</td>
<td>816 total 816 passed</td>
</tr>
<tr>
<td>532</td>
<td>Sept. 20, 2017, 9:31 p.m. 5 days ago</td>
<td>4 total 4 completed</td>
<td>816 total 816 passed</td>
</tr>
<tr>
<td>529</td>
<td>Sept. 20, 2017, 5:42 p.m. 5 days, 3 hours ago</td>
<td>4 total 4 completed</td>
<td>813 total 813 passed</td>
</tr>
<tr>
<td>525</td>
<td>Sept. 19, 2017, 6:01 p.m. 6 days, 3 hours ago</td>
<td>4 total 4 completed</td>
<td>1080 total 1080 passed</td>
</tr>
<tr>
<td>522</td>
<td>Sept. 19, 2017, 3:12 p.m. 6 days, 6 hours ago</td>
<td>4 total 4 completed</td>
<td>1080 total 1080 passed</td>
</tr>
<tr>
<td>517</td>
<td>Sept. 19, 2017, 12:26 p.m. 6 days, 9 hours ago</td>
<td>4 total 4 completed</td>
<td>1080 total 1080 passed</td>
</tr>
<tr>
<td>516</td>
<td>Sept. 18, 2017, 8:14 p.m. 1 week ago</td>
<td>4 total 4 completed</td>
<td>1080 total 1080 passed</td>
</tr>
</tbody>
</table>
SQUAD initial requirements

- Support for multiple projects
- Support for both test results (pass/fail) and metrics (measurements)
- Support upload of manual testing results
- Decoupled from test execution platforms (Jenkins, LAVA, etc)
## SQUAD data model

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Namespace for projects</td>
</tr>
<tr>
<td>Project</td>
<td>Represents a software project (e.g. SQUAD, or &quot;The 4.9 stable kernel&quot;). Has many builds</td>
</tr>
<tr>
<td>Build</td>
<td>Represents a snapshot of the project, at the source level. Has many test runs</td>
</tr>
<tr>
<td>Test Run</td>
<td>Represents the execution of a given build on given environment (e.g. a test job on LAVA, or a build on travis-ci.org) test. Contains test results and metrics.</td>
</tr>
<tr>
<td>Environment</td>
<td>A hardware/software context where tests are run</td>
</tr>
<tr>
<td>Suite</td>
<td>Group of tests or metrics; e.g. &quot;LTP&quot;, &quot;kselftests&quot;, &quot;libhugetlbfs&quot;. Tests in a test run can belong to a suite.</td>
</tr>
</tbody>
</table>
SQUAD data model: a "picture"
SQUAD data model: example (SQUAD itself)

- Project: qa/squad
- Builds: one for each commit
- Test Runs: one for each environment
- Environments: travis-sqlite3, travis-postgresql, travis-sqlite3-django1.10, travis-postgresql-django1.10
- Tests: the test suite in the SQUAD source tree
- Metrics: basic source metrics (# of Python lines, # of Python modules, etc)
Getting data into SQUAD: API

```bash
$ curl \
  --header "Auth-Token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \ 
  --form tests=@/path/to/test-results.json \ 
  --form metrics=@/path/to/metrics.json \ 
  --form metadata=@/path/to/metadata.json \ 
  --form log=@/path/to/log.txt \ 
  --form attachment=@/path/to/screenshot.png \ 
  --form attachment=@/path/to/extra-info.txt \ 
  https://host/api/submit/my-team/my-project/x.y.z/my-env
```
Getting data into SQUAD: API (2)

$ curl \
--header "Auth-Token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \
--form tests='{"test1": "pass", "test2": "fail"}/' \
--form metrics='{"metric1": 21, "metric2": 4}/' \
--form metadata'{"foo": "bar", "baz": "qux"}/' \
--form log='log text ...' \
--form attachment=@/path/to/screenshot.png \
--form attachment=@/path/to/extra-info.txt \
https://host/api/submit/my-team/my-project/x.y.z/my-env
Input format: test results (JSON)

```json
{
  "test1": "pass",
  "test2": "pass",
  "testsuite1/test1": "pass",
  "testsuite1/test2": "fail",
  "testsuite2/subgroup1/testA": "pass",
  "testsuite2/subgroup2/testA": null
}
```
Input format: metrics (JSON)

```json
{
    "v1": 1,
    "v2": 2.5,
    "group1/v1": [1.2, 2.1, 3.03],
    "group1/subgroup/v1": [1, 2, 3, 2, 3, 1]
}
```
Input format: metadata (JSON)

```json
{
    "build_url": "http://example.com/buils/XXXXXX/",
    "datetime": "2017-09-12T10:31:47-03:00",
    "job_id": "YYYYYY",
    "job_url": "http://example.com/jobs/YYYYYYY",
    "job_status": "Complete",
    "resubmit_url": "http://example.com//1/resubmit",
    "xxx": "yyy",
    "aaa": "abcdef"
}
```
CI Loop integration

- Modular support for integration with arbitrary test execution platforms (called "backends"). Currently only a LAVA backend is available.
- Test submission:
  - Test job can be submitted through SQUAD
  - Test job submitted by another tool (or even manually), and SQUAD can be told to "listen" for the results of that test job (by its ID)
- Obtaining test results
  - SQUAD can listen for live test results (e.g. LAVA ZMQ support)
  - SQUAD will also poll for results from time to time
  - TODO: using LAVA callback support to let LAVA notify SQUAD that a test job is done
Getting data into SQUAD: CI Loop integration

```
$ curl \\
   --header "Auth-Token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \\
   --form backend=lava \\
   --form definition=@/path/to/definition.yaml \\
   https://host/api/submitjob/my-team/my-project/x.y.z/my-env
```
Getting data into SQUAD: CI Loop integration (2)

$ curl \
   --header "Auth-Token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \
   --form backend=lava \
   --form testjob_id=123456 \
   https://host/api/watchjob/my-team/my-project/x.y.z/my-env
Running your own instance of SQUAD

- License: AGPLv3+
- Written in Python 3 (Python 2 not supported)
- Uses the Django web framework
- You will most probably want to use with:
  - apache2/nginx
  - PostgreSQL
- Tested/known to work with:
  - Debian 9 (stretch)
  - Ubuntu 16.04 (xenial)
- Running on Docker is supported
Running your own instance of SQUAD (2)

- See `INSTALL.rst` for production deployment instructions
- `pip install squad` will do a lot of what's needed
- Most configuration required can be done via environment variables
  - `$DATABASE, $SQUAD_EXTRA_SETTINGS, $SQUAD_SITE_NAME, $XDG_DATA_HOME, $SECRET_KEY_FILE, $DJANGO_LOG_LEVEL, $SQUAD_LOG_LEVEL, $SQUAD_HOSTNAME, $SQUAD_BASE_URL, $SQUAD_EMAIL_FROM, $SQUAD_LOGIN_MESSAGE, $SQUAD_ADMINS`
  - Documented in `INSTALL.rst`
- Linaro's production setup (using ansible) is public:
  - See [https://github.com/Linaro/qa-reports.linaro.org](https://github.com/Linaro/qa-reports.linaro.org)
Contributing

- [https://github.com/Linaro/squad](https://github.com/Linaro/squad)
- See README.rst and HACKING.rst for basic development instructions
- Standard Github workflow
- SQUAD is under active/heavy development

```
$ git log --format='%aN <%aE>' | sort -u
Antonio Terceiro <antonio.terceiro@linaro.org>
Dan Rue <dan.rue@linaro.org>
Luis Araujo <luis.araujo@collabora.co.uk>
Milosz Wasilewski <milosz.wasilewski@linaro.org>
```
Get in touch about SQUAD

There is no dedicated infrastructure yet. For now you can get in touch with us through:

- Github (issues/pull requests)
- IRC: #linaro-qa on Freenode
- email: qa-team@linaro.org
Thank You

#SFO17
BUD17 keynotes and videos on: connect.linaro.org
For further information: www.linaro.org