



Test Data Reporting

Contents

2	Workflow	2
3	Automated Tests	2
4	Manual Tests	2

3 Reports

Testing is a fundamental part of the project, but it is not so useful unless it goes along with an accurate and convenient model to report the results of such a testing. The [QA Test Report](#)¹ is an application that has been developed to save and report the test results for the Apertis images.

It supports both automated tests results executed by LAVA and manual tests results submitted by a tester.

Workflow

The deployment consists of two docker images, one containing the main report application and the other running the postgresql database. The general workflow is as follows:

Automated Tests

1. The QA Report Application is executed and it opens HTTP interfaces to receive HTTP requests calls and serve HTML pages in specific HTTP routes.
2. GitLab CI/CD builds the images and they are pushed to the image server.
3. GitLab CI/CD triggers the LAVA jobs to execute the automated tests in the published images.
4. GitLab CI/CD, when triggering the LAVA jobs, also registers these jobs with the QA Report Application using its specific HTTP interface.
5. The QA Report application adds these jobs in its internal queue and waits for the LAVA tests jobs results to be submitted via HTTP.
6. Once LAVA finishes executing the tests jobs, it triggers the configured HTTP callback sending all the test data to the QA Report application.
7. Test data for the respective job is saved into the database.

Manual Tests

1. User authenticate with GitLab credentials from the `Login` button in the main page.

¹<https://gitlab.apertis.org/infrastructure/lava-phab-bridge/>

- 33 2. Once logged in, the user can click on the `Submit Manual Test Report` button
34 that is now available from the main page.
- 35 3. Tester needs to enter the following information in the `Select Image Report`
36 page:
- 37 • Release: Image release (`v2023`, `v2024`, `v2025`, etc)
 - 38 • Version: The daily build identifier (`20240814.0215`, `20240828.0117`,
39 `20240828.0018`, etc)
 - 40 • Select Deployment Type (`APT`, `OSTree`)
 - 41 • Select Image Type (`Fixed function ARM64`, `HMI ARM64`, `SDK AMD64`, etc)
- 42 4. A new page only showing the valid test cases for the selected image type
43 is shown.
- 44 5. User selects `PASS` , `FAIL` or `NOT TESTED` for each test case.
- 45 6. An optional `Notes` text area box is available besides each test case for the
46 user to add any extra information (e.g tasks links, a brief comment about
47 any issue with the test, etc).
- 48 7. Once results have been selected for all test cases, user should submit this
49 data using the `Submit All Results` button at the top of the page.
- 50 8. The application now will save the results into the database and redirect
51 the user to a page with the following two options:
- 52 • Submit Manual Test Report: To submit tests results for a new image
53 type.
 - 54 • Go Back to Main Page: To check the recently submitted tests results.
- 55 9. If the user wants to update a report, just repeat the above steps selecting
56 the specific image type for the existing report and then updating the
57 results for the necessary test cases.

58 Reports

59 Reports for the stored test results (both manual and automated) are generated
60 on the fly by the QA report application, for example as done for the [v2020.3](#)
61 [release](#)².

²<https://qa.apertis.org/report/v2020/20201126.0>