



Test Data Reporting

# 1 Contents

2	<b>Workflow</b>	<b>2</b>
3	Automated Tests . . . . .	2
4	Manual Tests . . . . .	2

## 5 Reports **3**

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

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

## 12 Workflow

13 The deployment consists of two docker images, one containing the main report  
14 application and the other running the postgresql database. The general work-  
15 flow is as follows:

### 16 Automated Tests

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

### 30 Manual Tests

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

---

<sup>1</sup><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 (19.03, v2020dev0 ..)
  - 38 • Version: The daily build identifier (20190705.0, 20190510.1 ..)
  - 39 • Select Deployment Type (APT, OSTree)
  - 40 • Select Image Type
- 41 4. A new page only showing the valid test cases for the selected image type  
42 is shown.
- 43 5. User selects `PASS` , `FAIL` OR `NOT TESTED` for each test case.
- 44 6. An optional `Notes` text area box is available besides each test case for the  
45 user to add any extra information (e.g tasks links, a brief comment about  
46 any issue with the test, etc).
- 47 7. Once results have been selected for all test cases, user should submit this  
48 data using the `Submit All Results` button at the top of the page.
- 49 8. The application now will save the results into the database and redirect  
50 the user to a page with the following two options:
- 51 • Submit Manual Test Report: To submit tests results for a new image  
52 type.
  - 53 • Go Back to Main Page: To check the recently submitted tests results.
- 54 9. If the user wants to update a report, just repeat the above steps selecting  
55 the specific image type for the existing report and then updating the  
56 results for the necessary test cases.

## 57 Reports

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

---

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