Test results format

Created by John Spray, last modified by Lee Ochoa on Jan 24, 2020

This page describes the format of test results which is expected by Maloo. If you're developing scripts to output to maloo, consider uploading the test results to the maloo-temp staging environment first to validate your output.

Quick note on maloo terminology: an import tarball contains one **test session** which contains 1 or more **test sets**, each of which contains 0 or more **sub tests**.

Results upload

Test session results may be uploaded in one of three ways:

- Automatically from an Autotest instance when the session is completed
- Using the maloo_upload.sh script which may be downloaded from the *Results...Upload* main menu option in Maloo. This script can in turn then be used to either manually or automatically upload ad-hoc results.
- Manually upload the results tarball from the *Results...Upload* main menu option in Maloo

Results tarball

Regardless of the upload option used above, Maloo expects an import to consist of a single gzipped tarball. The filename can be anything but must have the extension ".tar.gz" and meet the following conditions:

- The tarball should contain only a flat set of files (no directories, if present they will be ignored)
- Tarballs must contain a file called "results.yml"
- Tarballs may contain a number of files named "node.<hostname>.yml"
- Tarballs may contain any number of files named ".log"

Maloo parses these tarballs into a database. Although the original tarball is preserved allowing users to download the results after ingest, **do not** 'piggyback' additional content and/or use it as a log file dumpster, we have a finite amount of storage and will optionally reject large tarballs if necessary.

An imported tarball is referred to in maloo as a 'test session'.

YAML files

Results YAML file

This file is required and without it the entire import will fail, the file **must** be named *results.yml*

In general, if a mandatory field is missing, the file will be judged invalid and the entire session import will be rejected by maloo. In the example below all fields are **mandatory** except as noted here:

- 'user_name' attribute in 'TestGroup' is optional
- 'project' is a required attribute, at present only LUSTRE is supported but others may be added in the future (such as MR or LIPE)
- if 'cumulative_result_id' is missing a unique id will be created, this id is shared by related test sessions so in this case there will not be any related sessions
- 'session_group_id' must be unique, if missing one will be created and if an existing import is found with the same id this import will be rejected
- 'enforcing' is optional, if missing it will be set to *false* and will in turn prevent Maloo from sending +/- reviews (see below)
- 'triggering_job_name' and 'triggering_build_number' are normally added by Autotest, for manual uploads they should be set to the job/build that this set of results tested against
- 'total_enforcing_sessions' is normally appended to the file by Autotest, for manual uploads this may be set to 1
- 'code_review' is optional, if present the test session will include a link to the review patch and maloo will issue +/-1 and/or results upload messages to the code review site
- 'issue_tracker' is optional, if present the test session will include a link to the issues ticket that is related to this test
- 'Tests' is an array of zero or more test sets
- The 'description' attribute in 'Tests' is optional
- 'SubTests' is an array of zero or more sub tests listed within a 'Tests' definition
- 'error' attribute in 'SubTests' definition is optional and will be set to an empty string if missing
- If the 'status' field is missing from a 'Test', then maloo imports the test set with the status 'ABORT'

Notes on status fields:

- Valid status values are 'PASS', 'FAIL', 'SKIP', 'TIMEOUT', 'ABORT', and 'CRASH'. Remember that sub tests have an 'error' value for recording the reason for a failure when you need to be more specific than just FAIL.
- The SKIP status only applies to sub tests.
- The status of a test session is not specified in the input, it's derived by maloo as "did all test sets in this test session PASS?".
- The status of a test set is read in during import independently of the sub test statuses. That means it's possible to have a test set where all the sub tests PASS but the test set FAILs (i.e. something outside of a specific sub test broke, like setup/teardown stuff). It's also possible to have a test set where the sub tests FAIL but the test set PASSes -- maloo will read the file "as-is", so if this happens there's probably something wrong with your test framework/script.

Notes on error fields:

- While Maloo has built-in resiliency to avoid rejecting an upload due to invalid characters in the error description, there is no guarantee it will succeed under all conditions, make sure the test framework follows the guidelines for properly formatting a yaml string

Example:

```
TestGroup:
    test_group: review-zfs-part-2
    testhost: onyx-35vm3
    submission: Wed Nov 08 15:50:13 UTC 2017
    user_name: root
project: LUSTRE
cumulative_result_id: 8f363c28-7e9d-46e9-8e9f-84a9411abddb
test_sequence: 1
test_index: 9
session_group_id: fd1780d5-55ec-4996-af84-267cee45a1a1
enforcing: true
triggering_job_name: lustre-reviews
triggering_build_number: 51951
total_enforcing_sessions: 5
code_review:
    type: Gerrit
    url: review.hpdd.intel.com
    project: fs/lustre-release
    branch: master
    identifiers:
    -   id: 4e6f7a6246f63fa802d87b02ceeb51a984de6f2a
issue_tracker:
    type: Jira
    url: jira.hpdd.intel.com
    identifiers:
    -   id: LU-10190
Tests:
-   name: node-provisioning-1
    description: node-provisioning logs
    submission: Wed Nov 08 15:50:13 UTC 2017
    report_version: 2
    SubTests:
    -   name: node-provisioning
        status: PASS
        duration: 1263
        return_code: 0
        error:
    duration: 1263
    status: PASS
-   name: replay-single
    description: auster replay-single
    submission: Wed Nov  8 16:12:26 UTC 2017
    report_version: 2
    SubTests:
    -   name: test_0a
        status: PASS
        duration: 29
        return_code: 0
        error:
    -   name: test_0b
        status: PASS
        duration: 22
        return_code: 0
        error:
    -   name: test_0c
        status: PASS
        duration: 97
        return_code: 0
        error: layout_lock needs MDS connection for IO
```

Node YAML files

Generally speaking, every test node used in a test session should create a node yaml file of the form "node.<node_name>.yml". While these files are optional, a lot of Maloo functionality will be missing and/or incorrect if they're not present.

In the example below, all fields are mandatory except:

- 'lbats_build_id' and 'lbats_build_name' are optional
- 'NodeEntities' may be an empty list (although it's required)
- 'networks' may be an empty list (although it's required)

Example:

```
Build:
    lbats_build_id:
    lbats_build_name:
    architecture: x86_64
    os: GNU/Linux
    os_distribution: CentOS Linux 7
    version: 2.10.54.57
    build: https://build.hpdd.intel.com/job/lustre-reviews/51951
    branch: master
    revision: 4e6f7a6246f63fa802d87b02ceeb51a984de6f2a
    kernel_version: 3.10.0-693.2.2.el7_lustre.x86_64
    file_system: zfs

Node:
    node_name: onyx-35vm11
    mem_size: 1720652 kB
    architecture: x86_64
    networks:
        - tcp

NodeEntities:
-
    node_type: OST 1
    node_name: onyx-35vm11
-
    node_type: OST 2
    node_name: onyx-35vm11
```

Log file names

Maloo attempts to set up relationships between test sets/sub tests and their corresponding log files. It also looks for and associates console log files. Any files that are not automatically associated are gathered into a "session" log files bucket where they're displayed for review. Any file that matches the following description will be automatically associated to an existing test set or sub test, all others will be collected as session log files:

- Filename parts are dot-separated
- Part 1 - Test set name
- Part 2 - Sub test name
- Part 3 - Type of file (e.g. dmesg, test-complete, etc)
- Part 4 - Test host
- Parts 5..n - Any other user-relevant information

Notes of filenames:

- If the hostname in a log filename does not correspond to a host defined in the accompanying node YAML, then the log is not associated
- Filenames may not contain periods other than to separate the various filename parts

Here are some example filenames:

These log files would associate to the test session

- autotest_auster.log
- autotest_execution.log
- console.onyx-35vm11.log

These log files would associate to a sub test assuming both the test set and sub test are defined in the *results.yml* file

- conf-sanity.test_35b.test_log.onyx-35vm3.log
- conf-sanity.test_36.test_log.onyx-35vm3.log
- lustre-rsync-test.test_8.lrsync_log.onyx-35vm3.log
- lustre-rsync-test.test_9.test_log.onyx-35vm3.log
- mds-survey.suite_log.onyx-35vm3.log
- mds-survey.test_1.test_log.onyx-35vm3.log
- mds-survey.test_2.test_log.onyx-35vm3.log
- node-provisioning-1.autotest.onyx-35vm3.log
- recovery-small.suite_log.onyx-35vm3.log
- recovery-small.test_1.test_log.onyx-35vm3.log
- replay-single.test_119.test_log.onyx-35vm3.log

Like   Be the first to like this   No labels

2 Comments

Richard Henwood

Looks good John, thanks!

Can you expand the example around Nodes - I'm specifically interested in:

- An example that includes multiple Nodes.
- The rules surrounding Node names.

Reply Edit Delete Like May 06, 2011

Li Wei

> The filenames are then parsed as <test set name>.<sub test name>.<log name>.<host name>.<junk>.log
>
> The 'junk' part of the filename may only be present when the log name is 'dmesg' or 'debug_log': this is a special case to deal with how the lustre tests name those particular log types.

Experiments show that the "junk" part is actually mandatory for "debug_log" and "dmesg"; without it Maloo doesn't take the logs.

Reply Edit Delete Like Sep 03, 2011

Write a comment...