



OpenStack's subunit2SQL

Open source tooling for converting subunit streams into a SQL DB.

Agenda

Subunit Overview

CI Overview

OpenStack's Use Case

OpenStack's Architecture (for log processing)

subunit2SQL Overview

- Objectives
- Schema
- Usage/Example

OpenStack Health

- Architecture/Overview
- screen captures from dashboard

Demo

- Architecture/Overview
- Creating subunit2SQL runs with jenkins CI





Subunit Overview

Subunit is a streaming protocol for test results.

It has excellent Python support but also supports, C, C++, and Shell.

The general format of a subunit is below.

```
time: 2016-03-24 21:05:38.652075Z
test: mytest.SampleTestCase.runTest
failure: mytest.SampleTestCase.runTest
[
  Traceback (most recent call last): File "/media/windows/dev/java/qaworkspace/pythonnosetests/src/mytest.py", line
  11, in runTest self.assertEqual(len(s), 4, 'Wrong length') AssertionError: Wrong length
]
time: 2011-05-2322:49:38.858163Z
```



Subunit Overview

Subunit is a streaming protocol for test results.

It has **excellent Python support** but also supports, C, C++, and Shell.

Generating a subunit is simple if you use the python testtools package.

- `$ python -m subunit.run jenkins2sql/tests/test_api.py`

A subunit can be generated in a python application as well using the testtools api.



Subunit Overview

```
class APITestCase(testtools.TestCase):
    jenkins_url = 'http://192.168.1.103:8080'
    job_name = 'gate-observer-py27'
    build_number = 42
    build_url = '%s/job/%s/%s' % (jenkins_url,
                                   job_name,
                                   build_number)

    def test_get_url(self):
        json_data = mock.Mock(text=json.dumps({'url': self.build_url}))
        self.addDetail('json-data', content.text_content(str(json_data.text)))
        url = api.get_url(json_data)
        self.assertEqual(self.build_url, url + 'error')
```



Subunit Overview

```
>>> from subunit import v2 as subunit_v2
>>> fd = open('my-subunit.stream', 'w')
>>> output = subunit_v2.StreamResultToBytes(fd)
>>> test_id = 'jenkins2sql.tests.test_api.APITestCase.test_get_url'
>>> file_bytes = 'help help help'.encode('utf-8')
>>> output.startTestRun()
>>> output.status(test_id=test_id, file_name='results',
                  mime_type='text/plain; charset="utf-8"',
                  eof=True, file_bytes=file_bytes)
>>> output.status(test_id=test_id, test_status='fail')
>>> output.stopTestRun()
>>> fd.close()
```

Subunit Overview

Subunit makes it easy to generate reports like this one.

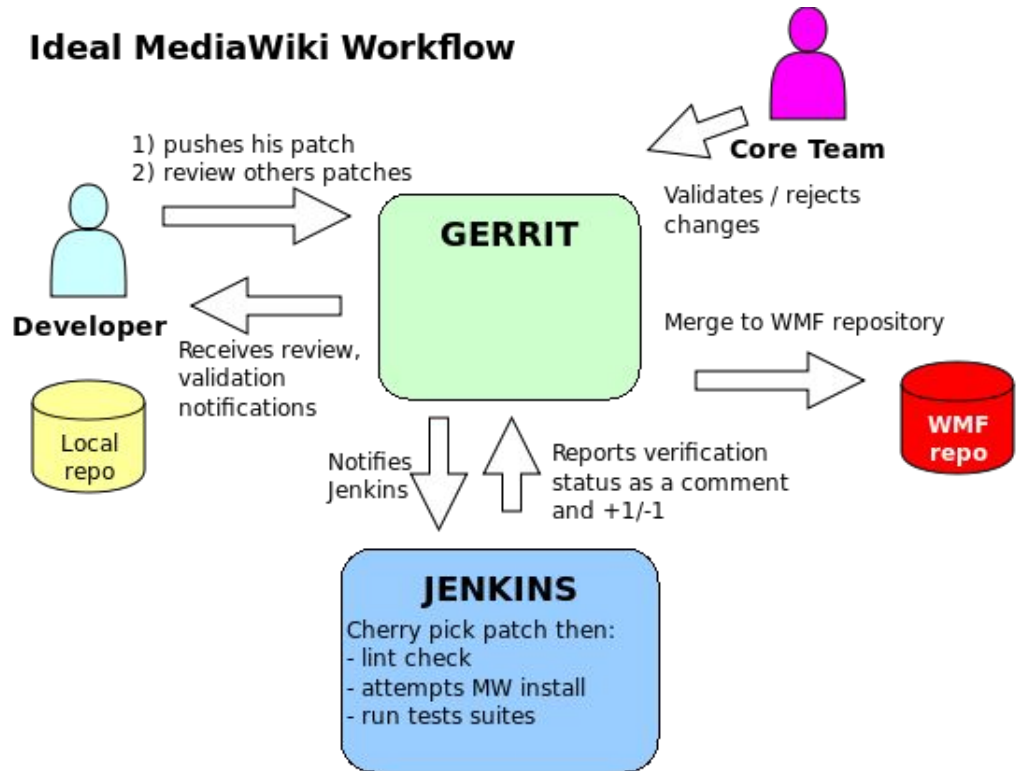
Test Group/Test case	Count	Pass	Fail	Error	Skip	View
tempest.tests.test_list_tests.TestTestList	1	0	1	0	0	Detail
<div><div>test_stestr_list_no_errors</div><div><pre>ftl1: tempest.tests.test_list_tests.TestTestList.test_stestr_list File "tempest/tests/test_list_tests.py", line 44, in test_stestr self.assertEqual('haha a failure', '') File "/home/ubuntu/tempest/.tox/py27/local/lib/python2.7/site-pac self.assertThat(observed, matcher, message) File "/home/ubuntu/tempest/.tox/py27/local/lib/python2.7/site-pac raise mismatch_error testtools.matchers._impl.MismatchError: 'haha a failure' != ''</pre></div></div>						
tempest.tests.api.compute.test_base.TestBaseV2ComputeTest	9	9	0	0	0	Detail
tempest.tests.cmd.test_account_generator.TestAccountGeneratorV2	2	2	0	0	0	Detail

CI Overview

Gerrit is a free web-based code collaboration tool. It allows a team of developers to approve or reject code.

Jenkins is a popular open source automation server. In this case, it's used for testing the code pushed by the developer.

Ideal MediaWiki Workflow





OpenStack's Use Case

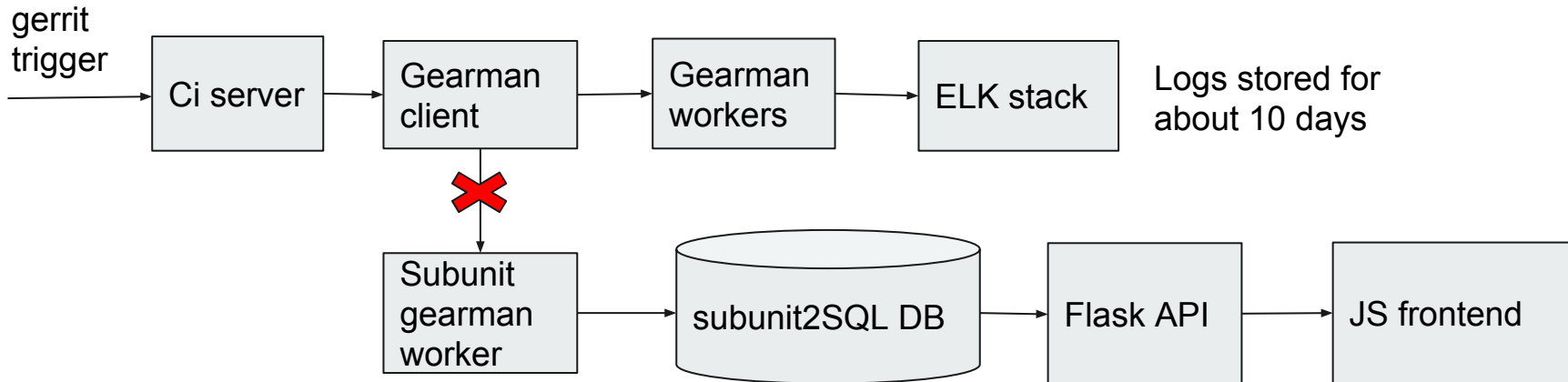
OpenStack's CI system runs the same basic test suites hundreds of times a week.

There was no way for storing test data for a long period of time.

Looking at July (results according to their mysql db)

- 62k jobs were ran
- That makes 110M test runs

OpenStack-like Architecture



* This diagram tries to give a rough idea



subunit2SQL Overview

subunit2SQL is a tool for storing data on test results by processing subunit streams.

It's a python library, a set of utilities, and tools that set up the sql schema for you.

It's written with SQLAlchemy and in theory should be able to support the same databases as SQLAlchemy:

- SQLite, MySQL, PostgreSQL, Microsoft SQL Sever, and more!

Utilities:

- subunit2sql
 - For storing subunits
- sql2subunit
 - For retrieving subunits
- subunit2sql-db-manage
 - For managing the database
- subunit2sql-graph
 - For interacting with the database to produce graphs



Project objectives

To provide a way of observing things like:

- Overall test results from a given release, test server, or any data really
- Success and failure rates of a test over a period of time
 - Helpful with failures related to race conditions
- Performance analysis for...
 - Identifying tests that always pass, fail, or skip
 - Identifying slow tests and verifying their fixes



Project Schema

subunit2SQL has 3 data types: Tests, Runs, and Test Runs. Each of these types have their own key value pair metadata table for aggregating on any data of any specific row.

- 01 **Tests** - A way of identifying a test. This table stores the test id.
- 02 **Runs** - A single execution of a set of tests. This table stores **aggregate** information like run time, number of successes, skips, and failures.
- 03 **Test Runs** - A single test execution in a run. This table stores the **subunit data** itself. It stores the pass, skip, or fail status, start time, and stop time for a given test for a given run.



CLI Usage

```
subunit2sql subunit.stream \  
    --run_meta build_id:tempest-py27-manual-1,build_name:tempest-py27 \  
    --non_subunit_name 'non-subunit' \  
    --database-connection mysql://citest:citest@localhost/citest
```

Project schema: run metadata

```
mysql> select * from run_metadata;
```

id	key	value	run_id
1	build_id	tempest-py27-manual-1	1
2	build_name	tempest-py27	1

```
2 rows in set (0.00 sec)
```



Project schema: runs

```
mysql> select * from runs;
```

uuid	id	skips	fails	passes	run_time	artifacts	run_at
bbcff377-6b99-4a35-84b5-eb01fa119737	1	0	0	1834	13.2054	NULL	2018-08-04 22:52:24

```
1 row in set (0.00 sec)
```




Project schema tests

```
mysql> select * from tests limit 10;
```

id	test_id
1	tempest.tests.lib.services.compute.test_networks_client.TestNetworksClient.test_show_network_with_bytes_body
2	tempest.tests.api.compute.test_base.TestBaseV2ComputeTest.test_create_image_from_server_no_wait
3	tempest.tests.lib.services.identity.v3.test_credentials_client.TestCredentialsClient.test_list_credentials_with_str_body
4	tempest.tests.lib.common.test_validation_resources.TestValidationResources.test_create_validation_resources_neutron
5	tempest.tests.lib.test_auth.TestKeystoneV3AuthProvider.test_request_with_alt_part_without_alt_data_no_change_headers
6	tempest.tests.lib.services.volume.v3.test_groups_client.TestGroupsClient.test_delete_group
7	tempest.tests.lib.services.compute.test_snapshots_client.TestSnapshotsClient.test_list_snapshots_with_params
8	tempest.tests.lib.services.compute.test_aggregates_client.TestAggregatesClient.test_create_aggregate_with_str_body
9	tempest.tests.lib.test_credentials.KeystoneV2CredentialsTests.test_reset_all_attributes
10	tempest.tests.lib.services.identity.v3.test_services_client.TestServicesClient.test_delete_service

```
10 rows in set (0.00 sec)
```



Project schema: test_runs

```
mysql> select * from test_runs limit 10;
```

id	test_id	run_id	status	start_time	stop_time	start_time_microsecond	stop_time_microsecond
1	1	1	success	2018-08-04 22:51:18	2018-08-04 22:51:18	796727	797347
2	2	1	success	2018-08-04 22:51:12	2018-08-04 22:51:12	602854	604468
3	3	1	success	2018-08-04 22:51:18	2018-08-04 22:51:18	980420	981079
4	4	1	success	2018-08-04 22:51:18	2018-08-04 22:51:18	625286	630594
5	5	1	success	2018-08-04 22:51:18	2018-08-04 22:51:18	325803	326690
6	6	1	success	2018-08-04 22:51:16	2018-08-04 22:51:16	177198	178010
7	7	1	success	2018-08-04 22:51:18	2018-08-04 22:51:18	912655	914332
8	8	1	success	2018-08-04 22:51:15	2018-08-04 22:51:15	367657	369568
9	9	1	success	2018-08-04 22:51:16	2018-08-04 22:51:16	239089	239667
10	10	1	success	2018-08-04 22:51:16	2018-08-04 22:51:16	925180	926024

Project schema: attachments.. but it's not pretty.

```
> select * from attachments;
```

id	test_run_id	label	attachment
1	2	traceback	<Content type=text/x-traceback; charset="utf8'
2	2	json-data	<Content type=text/plain; charset="utf8", val



Python Usage

subunit2SQL is importable. The arguments for the CLI can be overwritten using the shell.CONF object.

This example shows:

- Setting the database connection
- Setting run metadata and artifacts path
- Parsing and storing the results

```
from subunit2sql import shell
from subunit2sql import read_subunit

subunit_file = open('subunit_file', 'r')
# Load default config
shell.cli_opts()
shell.parse_args([])
# Set database connection
db_uri = 'mysql://subunit:subunit@localhost/subunit'
shell.CONF.set_override('connection', db_uri, group='database')
# Set run metadata and artifact path
artifacts = 'http://fake_url.com'
metadata = {
    'job_type': 'full-run',
    'job_queue': 'gate',
    'build_id': 'fun_hash'
}
shell.CONF.set_override('artifacts', artifacts)
shell.CONF.set_override('run_meta', metadata)
# Parse results and write to DB
stream = read_subunit.ReadSubunit(subunit_file)
shell.process_results(stream.get_results())
```



Python Usage

subunit2SQL has a database API

This example shows:

- Creating a session
- Getting all test_runs for a given test
- Getting all runs that failed because of this test

```
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from subunit2sql.db import api

# Create engine with db url for session generation
engine=create_engine('mysql://subunit:subunit@localhost/subunit')
Session = sessionmaker(bind=engine)

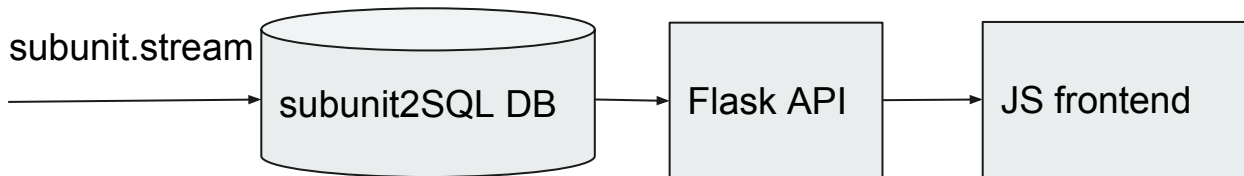
# Create a new session to pass to API calls
# EX: api.get_run_metadata(session=session)
session = Session()
db_test_runs = api.get_test_runs_by_test_test_id(
    test_id, session=session,
    start_date=start_date, stop_date=stop_date)

failed_run_ids = [
    x.run_id for x in db_test_runs if x.status == 'fail']

failed_runs = api.get_runs_by_ids(failed_run_ids, session=session)
```

OpenStack Health

OpenStack Health is a dashboard for visualizing test results of Openstack CI jobs.



displays

- aggregate information with subunit2SQL's DB API
- Uses **numpy/pandas** underneath for modifying numerical data and time series information:
 - Smooths out time series for job/test mean run time graphs





















OpenStack Health Overview

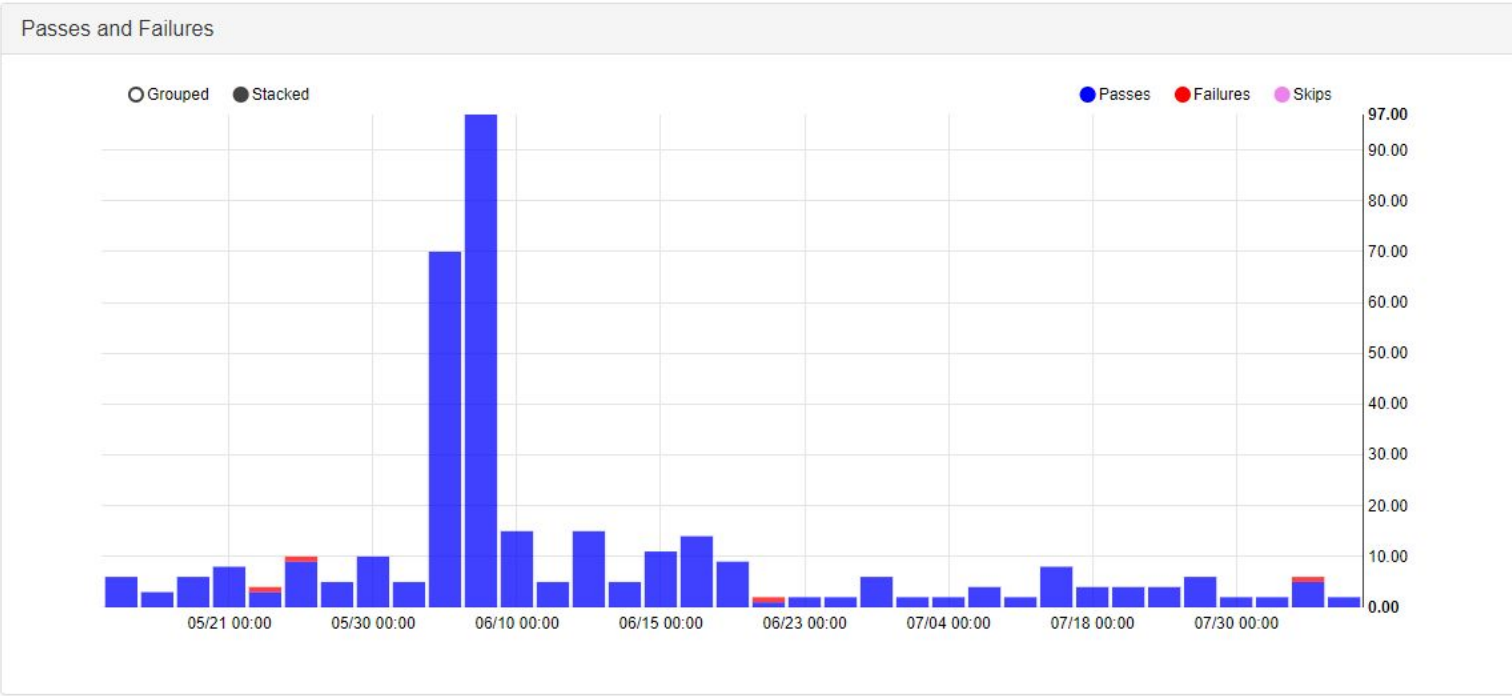
<http://status.openstack.org/openstack-health/#/>
<http://zuul.openstack.org/>

* The following slides use OpenStack Health as an example for viewing this data.

Overall test results from a given release or test server

Build Branch Status		<input type="text" value="(stable master)"/>				
#	Name	Passes	Failures	% Passes	% Failures ▲	Bar Graph
1	stable/ocata 	3,646	93	97.51	2.49	
2	stable/pike 	10,000	164	98.39	1.61	
3	master 	114,740	1,646	98.59	1.41	
4	stable/queens 	20,535	158	99.24	0.76	
5	stable/newton 	523	3	99.43	0.57	
6	stable/17.11 	7	0	100.00	0.00	
7	stable/0.10 	6	0	100.00	0.00	
8	stable/jewel 	4	0	100.00	0.00	
9	stable/18.02 	83	0	100.00	0.00	

Success and failure rate of a test over a period of time

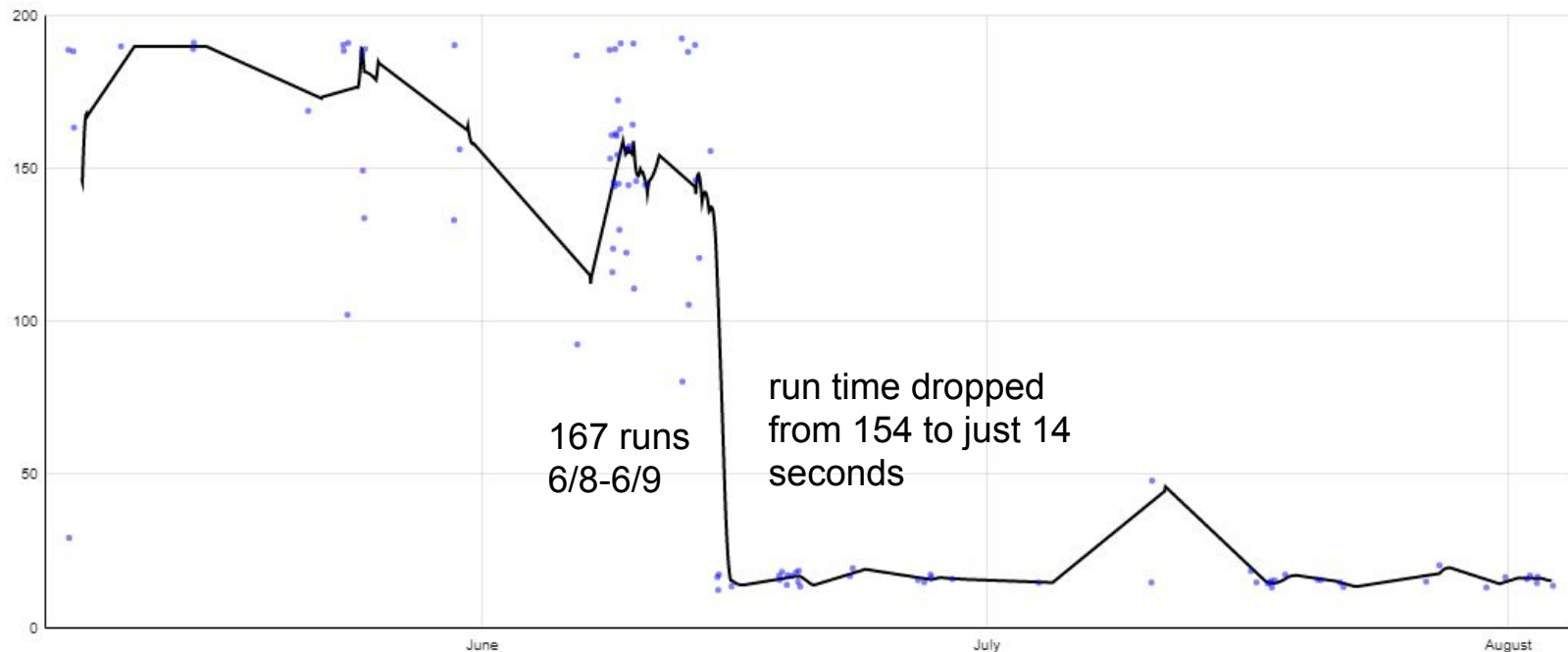


Total Test Status Counts		
Status	Count	Percentage
Successes	354	98.88%
Failures	4	1.12%
Skips	0	0.00%

Identifying slow tests and fixes for them

VolumesActionsV3RbacTest.test_force_detach_volume_from_instance

Run Time



Identify tests that always pass, fail, or skip

Tests					
<div><div></div><div>Search for test with regex</div></div>					
Test Name	Passes ▾	Failures	Skips	% Failures	Mean Runtime
kuryr_tempest_plugin.tests.scenario.test_service.TestServiceScenario	0	5	4	100.00	0.00
kuryr_tempest_plugin.tests.scenario.test_port_pool.TestPortPoolScenario	0	0	20	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_kuryr_restart.TestKuryrRestartScenario	0	0	20	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_kuryr_restart.TestKuryrRestartScenario.test_kuryr_pod_delete	0	0	23	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_service.TestLoadBalancerServiceScenario	0	0	4	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_namespace.TestNamespaceScenario	0	0	90	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_port_pool.TestPortPoolScenario.test_port_pool	0	0	106	0.00	0.00
kuryr_tempest_plugin.tests.scenario.test_service.TestLoadBalancerServiceScenario.test_lb_service_curl	106	0	0	0.00	0.09

Job Failure Scenario

periodic-tempest-dsvm-full-test-accounts-master job started failing



Recent Runs

Link	Status	Run At ▲
http://logs.openstack.org/periodic/git.openstack.org/openstack/tempest/master/legacy-periodic-tempest-dsvm-full-test-accounts-master/657682f	fail	8/5/2018 06:14
http://logs.openstack.org/periodic/git.openstack.org/openstack/tempest/master/legacy-periodic-tempest-dsvm-full-test-accounts-master/0e6cf54	fail	8/4/2018 06:10
	fail	8/3/2018 06:08
	fail	8/2/2018 06:08
s_negative.BaseVolumeQuotasNegativeTestJSON	fail	8/1/2018 06:19
hot_quotas_negative.VolumeSnapshotQuotasNegativeTestJ	fail	7/31/2018 06:09
ces.AccountTest.test_list_containers_with_end_marker	fail	7/30/2018 06:16
hot_quotas_negative.VolumeSnapshotQuotasNegativeTestJ	fail	7/29/2018 06:13
http://logs.openstack.org/periodic/git.openstack.org/openstack/tempest/master/legacy-periodic-tempest-dsvm-full-test-accounts-master/9df49dc	success	7/28/2018 06:09
http://logs.openstack.org/periodic/git.openstack.org/openstack/tempest/master/legacy-periodic-tempest-dsvm-full-test-accounts-master/54df9ab	success	7/27/2018 06:21

Q

Search for test with regex

	Passes	Failures	Skips	% Failures ▲	Mean Runtime
s_negative.BaseVolumeQuotasNegativeTestJSON	0	4	0	100.00	0.00
hot_quotas_negative.VolumeSnapshotQuotasNegativeTestJ	0	3	0	100.00	0.00
ces.AccountTest.test_list_containers_with_end_marker	3	4	0	57.14	0.03
hot_quotas_negative.VolumeSnapshotQuotasNegativeTestJ	3	1	0	25.00	-0.40

Job Failure Scenario

```
setUpClass (tempest.api.volume.admin.test_volume_quotas_negative.BaseVolumeQuotasNegativeTestJSON)
```

Captured traceback:

```
~~~~~
Traceback (most recent call last):
  File "tempest/test.py", line 172, in setUpClass
    six.reraise(etype, value, trace)
  File "tempest/test.py", line 165, in setUpClass
    cls.resource_setup()
  File "tempest/api/volume/admin/test_volume_quotas_negative.py", line 59, in resource_setup
    cls.volume = cls.create_volume()
  File "tempest/api/volume/base.py", line 132, in create_volume
    volume = cls.volumes_client.create_volume(**kwargs)['volume']
  File "tempest/lib/services/volume/v3/volumes_client.py", line 75, in create_volume
    resp, body = self.post('volumes', post_body)
  File "tempest/lib/common/rest_client.py", line 279, in post
    return self.request('POST', url, extra_headers, headers, body, chunked)
  File "tempest/lib/services/volume/base_client.py", line 38, in request
    method, url, extra_headers, headers, body, chunked)
  File "tempest/lib/common/rest_client.py", line 670, in request
    self._error_checker(resp, resp_body)
  File "tempest/lib/common/rest_client.py", line 802, in _error_checker
    raise exceptions.OverLimit(resp_body, resp=resp)
tempest.lib.exceptions.OverLimit: Request entity is too large
Details: {u'retryAfter': u'0', u'message': u'VolumeSizeExceedsAvailableQuota: Requested volume or snapshot exceeds allowed gigabyte
s quota. Requested 1G, quota is 2G and 4G has been consumed.', u'code': 413}
```

Perhaps concurrency is an issue for the quota tests in the test accounts job..?



AT&T's dashboard

Extends OpenStack Health's API to create runs from jenkins JSON build data.

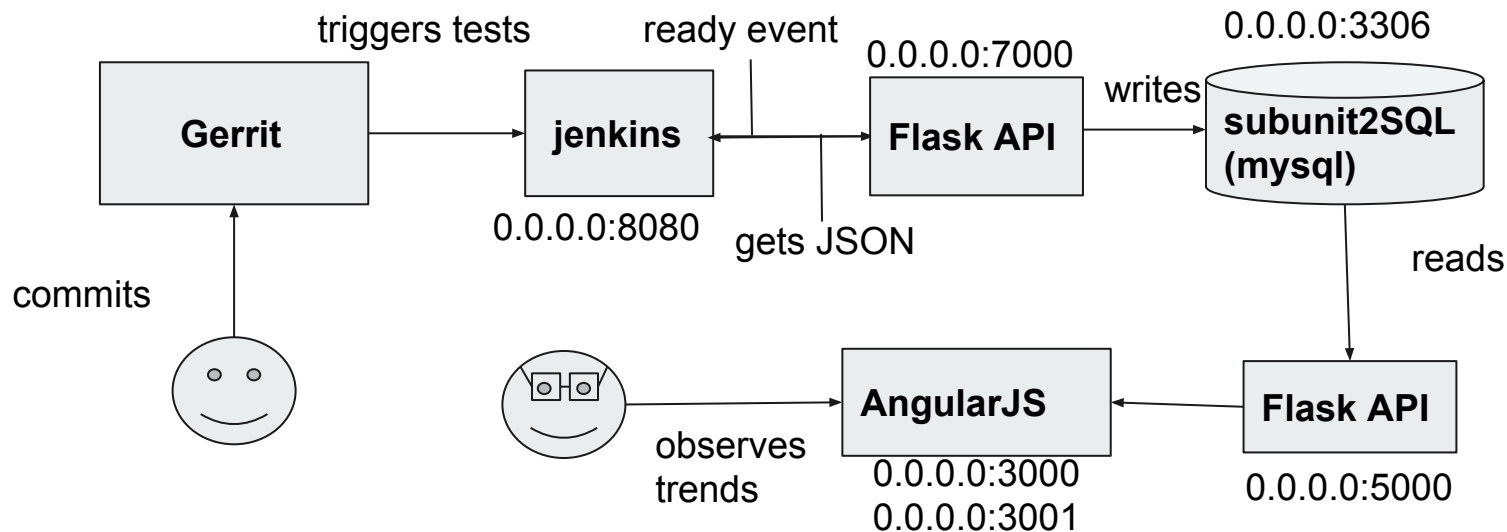
In beta for about 6 months, collects data on “on-demand” jobs.

Soon will be testing on scheduled/periodic deployments.

Still considering an alternative to OpenStack Health because:

- AngularJS is dead/dying
- Runs on unsupported version of node

Demo architecture - jenkins2sql



- * This architecture drops the requirement for installing subunit2SQL on your CI server
- * For simpler demo purposes, instead of using Gerrit, jenkins will poll for changes on github

Target audience

subunit2SQL may be helpful to...

A test automation team who manage and maintain hundreds of tests and test suites like...

OpenStack, the team I work on, and... anyone else?





Target audience

subunit2SQL may be helpful to...

a developer with some free time on his hands?

The project's "backlog" can be found here:

<https://storyboard.openstack.org/#!/project/747>

openstack-infra/subunit2sql

Tooling for converting subunit streams into a SQL DB

Active

Merged

Invalid

▶ 2003229

Infrastructure: Run under Python 3 by default

▶ 2003064

subunit2sql-graph failures throws exception when



Where to get more information?

Freenode IRC channel #openstack-qa

- Author's nickname: mtreinish
- My nickname: trevormc

documentation...

- subunit2SQL <https://docs.openstack.org/subunit2sql/latest/index.html>
- OpenStack Health documentation...
<https://github.com/openstack/openstack-health/blob/master/README.rst>
- jenkins2SQL documentation... <https://github.com/trevormccasland/jenkins2sql/blob/master/README.rst>



Where to get the data?

Want to just play around with the data?

- It's hosted on a public mysql server anyone can login to

```
$ mysql -u query -pquery -hlogstash.openstack.org -Dsubunit2sql;
```



Some recent works

<https://review.openstack.org/#/q/status:open+project:openstack-infra/subunit2sql>

Subject	Status	Owner
docs: add info about subunit2SQL usage	Merged	Trevor McCasland
fix prepare for numeric data		Trevor McCasland
Add attachments flag to get_test_runs_by_status		Trevor McCasland
fix get_numeric_data	Merged	Trevor McCasland
Add run time graph for jobs	Merged	Trevor McCasland
Add load_attachments for fail data		Trevor McCasland
Fix gulp-util deprecation	Merged	Trevor McCasland
Add support for HeadlessChrome in karma	Merged	Trevor McCasland
Fix npm-test: write after end error	Merged	Trevor McCasland
Add subunit2sql CLI option to use non_subunit_name	Merged	Trevor McCasland
Fix mysql db api	Merged	Trevor McCasland
Handle 'I' in build_name for getting test_runs	Merged	Trevor McCasland



Thank you!