decisionengine

Release 1.1.1

Fermi Research Alliance, LLC.

Mar 04, 2021

CONTENTS

1 Release Notes 3

2 Developer Documentation 9

3 Jenkins CI pipeline 11

4 Source code 17

5 Indices and tables 27

Python Module Index 29

Index 31

The decisionengine_modules package contains provider specific implementations of base classes provided by the decision engine framework.

CHAPTER

ONE

RELEASE NOTES

1.1 Release 1.5.0

In this release:

- New AWS bill calculator
- Bug fixes

1.1.1 Issues fixed in this release

- 271 : Change configuration so the bill-calculator-hep-mapsacosta library doesn't write stuff to stdout (5410c46)
- 283 : Change DEAccountConstants.py to allow more variables through (7a65032)
- 293 : Unit test tests/test_job_q.py is failing (f5006fb)
- 263 : Python3 encodes hex strings differently, resulting in malformed htcondor classads (c16f0)
- 289 : Travis CI fails after decisionenigine transitioned to pybind11 (2bec1da)
- 281 : New exception in NERSC newt get_usage (398dc42)

1.1.2 Full list of commits since version 1.4.2

5410c46 : new bill calculator package (#303)

7a65032 : Resolving issue 283 (#301)

1a2c3e4 : Fix syntax errors and pylint bits (#300)

976c4a4 : Minor linting of actions (#297)

c43f311 : Drop redundant action, pep8speaks is already doing this (#296)

10691e4 : Add support to use custom docker layer as base of the docker image (#295)

f5006fb : fix for unit test test_job_q issue 293 (#294)

0b588ef : Update workflow per changes at github (#292)

7841d19 : Updated Jenkins configuration documentation (#291)

c08ffce : edited release notes

c16f0ac : Another fix to malformed strings (#288)

2bec1da : Fix #289 where TravisCI is missing changes from https://github.com/HEPCloud/decisionengine/commit/ea7ade57ca825afb0afbfa5ebd49820b16786d8b (#290)

8647bb2 : Update to current python test matrix (#287)

234a78e : update release notes

398dc42 : decisionengine_modules/NERSC/util: Added more 5xx codes to retry (#285)

d2df861 : force docker pull when building the docker container to make sure to use an updated base layer (#280)

9a78d22 : added 1.4.0 release notes

5d3b7ec : Updating Dockerfile for unittest (#279)

e16531e : Provide default for util function. (#278)

02f1738 : set RPM version number to 1.5.0rc and drop py3 suffix (#277)

664b9fe : Add timeout support for unit test on Jenkins for DE modules repo (#274)

e392a3c : Added initial unit test for glide_frontend_element.py (#269)

2f0069f : Updated DE modules Jenkins documentation

1.2 Release 1.4.2

In this release:

- Bug fixes to 1.4.1 release

1.2.1 Issues fixed in this release

- 282 : Decision Engine classad encoding problems make multiple exceptions in the factory (c16f0ac)

1.2.2 Full list of commits since version 1.4.1

c16f0ac : Another fix to malformed strings (#288)

1.3 Release 1.4.1

In this release:

- Bug fixes to 1.4.0 release

1.3.1 Issues fixed in this release

- 281 : New exception in NERSC newt get_usage (4d1301e)

1.3.2 Full list of commits since version 1.4.0

4d1301e : decisionengine_modules/NERSC/util: Added more 5xx codes to retry (#285)

1.4 Release 1.4.0

In this release:

- Works with python3 glideinWMS package
- Uses new AWS bill calculator
- Improved Jenkins CI pipeline

1.4.1 Issues fixes in this release

- 59 : Replace AWS Billing and GCE Billing modules with revamped bill Calculator code (71f25ea)
- 246 : Modify NERSCAllocation main() so that it can be run standalone (11ce772)
- 263 : Python3 encodes hex strings differently, resulting in malformed htcondor classads (13a24e4)
- 255 : Add option to config of condor/sources/job_q.py and slots.py (6f65af4)

1.4.2 Full list of commits since version 1.3.0

42f62d3 : remove py3 from package name

6e56c4d : bump version number

6433dba : Add support to test any branch in Jenkins for DE modules repo (#276)

60db1b3 : Accommodate breaking changes to framework's configuration system. (#273)

13a24e4 : Fix263 (#272)

a866e47 : Correct 'file_cache is unavailable when using oauth2client >= 4.0.0' (#267)

2c82064 : Fix requirements.txt for #266 (#268)

6f65af4 : A more generic solution for the issue 255 (#265)

df08cb4 : Speed up tests. (#264)

057e578 : Update Jenkins pipeline to properly test closing PR (#262)

a28f272 : Adding Jenkins pipeline documentation (#261)

71f25ea : T59 (#258)

90b6580 : Updating Jenkins pipeline configuration (#260)

b4bd759 : dealing with 2 columnsindividually (#257)

11ce772 : decisionengine_modules/NERSC: Can run NerscAllocationInfo.py without (#256)

4a70465 : doc: add release notes

1.5 Release 1.3.0

In this release:

- Introduced Jsonnet based configuration system
- Improved error handling

1.5.1 Full list of commits since version 1.2.0

e8c8e24 : Update to split a tuple key into two separate keys, this allows python config file to be JSON compliant config file (#254)

fbb5d28 : add jsonnet dependency in build/scripts/utils.sh for unit tests (#252)

95a0a32 : decisionengine_modules/glideinwms: Added retry to factory queries (#250)

77f894a : Aws (#249)

3be5be1 : Set theme jekyll-theme-slate

a73e7f8 : Doc (#247)

f8aa96d : Adding editor config file, see https://editorconfig.org/ (#248)

1ab1720 : Issue no. 237 (#245)

5edc8f9 : Issue 227 (#243)

e6047f2 : GCE channel goes OFFLINE on a timeout (#242)

63e3627 : Aws billing (#241)

c3626b9 : 1.1.3/fix (#239)

ae7c5a6 : Jenkins pipeline improvements (#236)

310befd : T198 (#235)

a65886d : Fix import as reported in : https://github.com/HEPCloud/decisionengin... (#232)

93711cc : Run coveralls even if tests fail (#229)

03d763a : Jenkins pipeline improvements (#230)

f48d30f : Fix/223 (#228)

c8aa262 : github ticket 199 (#222)

0323bda : Address : https://github.com/HEPCloud/decisionengine_modules/issues/224 (#226)

62e4df6 : Add support to run CI on Jenkins (#221)

5ab1541 : bump master version to 1.2.0 (for now) (#219)

bc19c65 : decisionengine_modules/NERSC: Added retry loop for NERSC API Calls (#220)

41a50de : Sync up pep8speaks and run_pylint.sh with decisionengine settings (#218)

db4634f : silence pylint error (#217)

1b95141 : Fix whitespace around operator error

746ea38 : ignore W503

8a8b5f4 : remove unused variable

a6668bf : fix PEP8 warnings

13773ee : address pep8 warnings

6bea4ca : silence pylint error

f589895 : Pass sort=True parameter to fix future warning (#215)

a1d0507 : fixing pep8 warning

a10bd17 : debugging one import error

ec501ad : make coveralls.io links work

deab1a7 : T201 (#204)

69f2645 : Add coveragerc

6d8a5f5 : decisionengine_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)

a7e0af9 : Only run Github Actions for python3.6 (#24)

1.6 Release 1.2.0

In this release:

- Switched to python3
- NERSC newt interface adapted to changed NERSC RESTful API
- Improved unit tests
- Added Jenkins CI pipeline

1.7 Patches:

30d928b : clone version 1.2.0 of decisionengine

ae7c5a6 : Jenkins pipeline improvements (#236)

310befd : T198 (#235)

a65886d : Fix import as reported in : https://github.com/HEPCloud/decisionengin... (#232)

93711cc : Run coveralls even if tests fail (#229)

03d763a : Jenkins pipeline improvements (#230)

f48d30f : Fix/223 (#228)

c8aa262 : github ticket 199 (#222)

0323bda : Address : https://github.com/HEPCloud/decisionengine_modules/issues/224 (#226)

62e4df6 : Add support to run CI on Jenkins (#221)

5ab1541 : bump master version to 1.2.0 (for now) (#219)

bc19c65 : decisionengine_modules/NERSC: Added retry loop for NERSC API Calls (#220)

41a50de : Sync up pep8speaks and run_pylint.sh with decisionengine settings (#218)

db4634f : silence pylint error (#217)

1b95141 : Fix whitespace around operator error

746ea38 : ignore W503

8a8b5f4 : remove unused variable

a6668bf : fix PEP8 warnings

13773ee : address pep8 warnings

6bea4ca : silence pylint error

f589895 : Pass sort=True parameter to fix future warning (#215)

a1d0507 : fixing pep8 warning

a10bd17 : debugging one import error

ec501ad : make coveralls.io links work

deab1a7 : T201 (#204)

69f2645 : Add coveragerc

6d8a5f5 : decisionengine_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)

a7e0af9 : Only run Github Actions for python3.6 (#24)

CHAPTER

TWO

DEVELOPER DOCUMENTATION

2.1 Decisionengine_modules

2.1.1 Prerequisites:

```
yum install postgresql11 postgresql11-server
sudo localedef -v -c -i en_US -f UTF-8 C.UTF-8 # (needed for tests)
```

2.1.2 Test

```
git clone https://github.com/HEPCloud/decisionengine_modules
./decisionengine_modules/.github/actions/unittest-in-sl7-docker/entrypoint.sh
```

CHAPTER

THREE

JENKINS CI PIPELINE

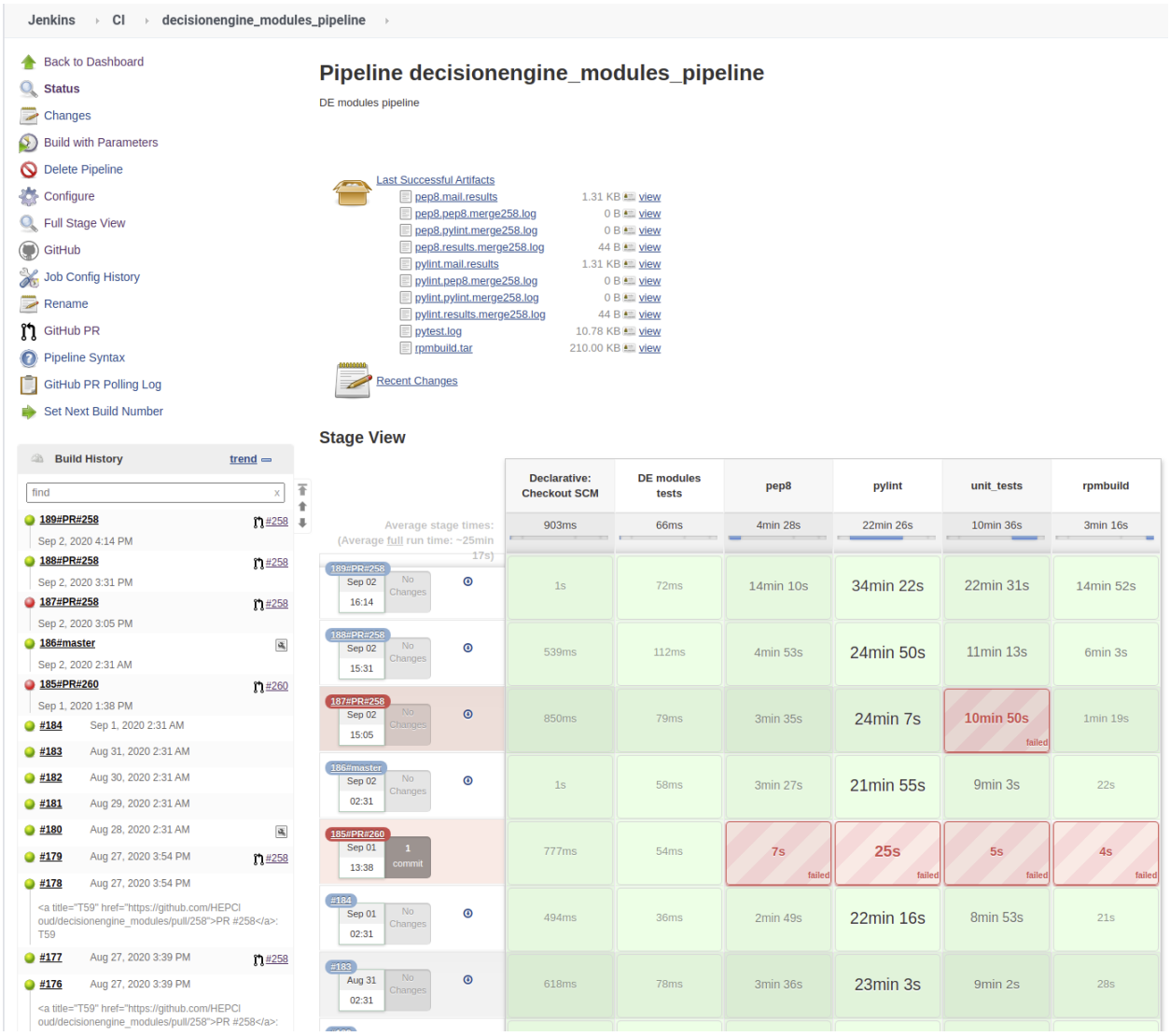
3.1 Decisionengine modules CI with Jenkins pipeline

Jenkins dashboard with Decisionengine modules CI results is available here.

A CI build is triggered any time a PR is created/closed or a commit is made to an existing PR.
There are also *nightly CI builds* to test a list of predefined branches.

The Jenkins pipeline runs *pep8*, *pylint* and *unit_tests* test suites alongside the *rpmbuild* stage.

The Jenkins dashboard looks like this:



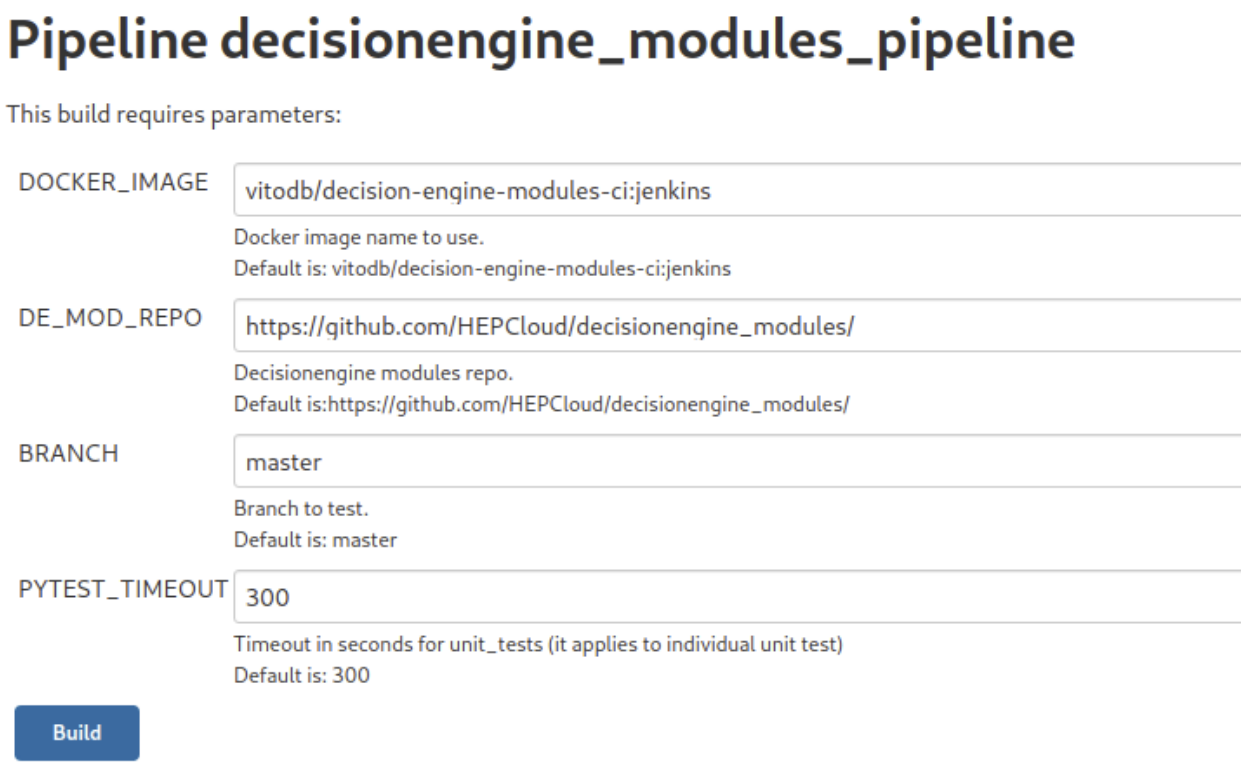
On the bottom left side there is the list of recent CI builds that are named after the PR or the branch tested.
On the bottom right side the dashboard shows for each CI build detailed status for each test suite.

Hovering the mouse over the *status box* for each CI build stage, a tool-tip with a button to access log details shows up.

Next to the build number the symbol gives access to a menu with the list of artifacts stored for that build. Those artifacts include logs and the tarball with RPMs.

From the panel on the left side it is possible to access the PR on GitHub by clicking on the PR icon that looks like this #258.

On occasion it could be useful to trigger a manual CI build to test a branch on the official DE GitHub repository or on the user fork. For this purpose, on the top left panel the user can click on the Build with Parameters button, and this panel shows up



the user can modify these parameters to customize what code to test with the CI build.

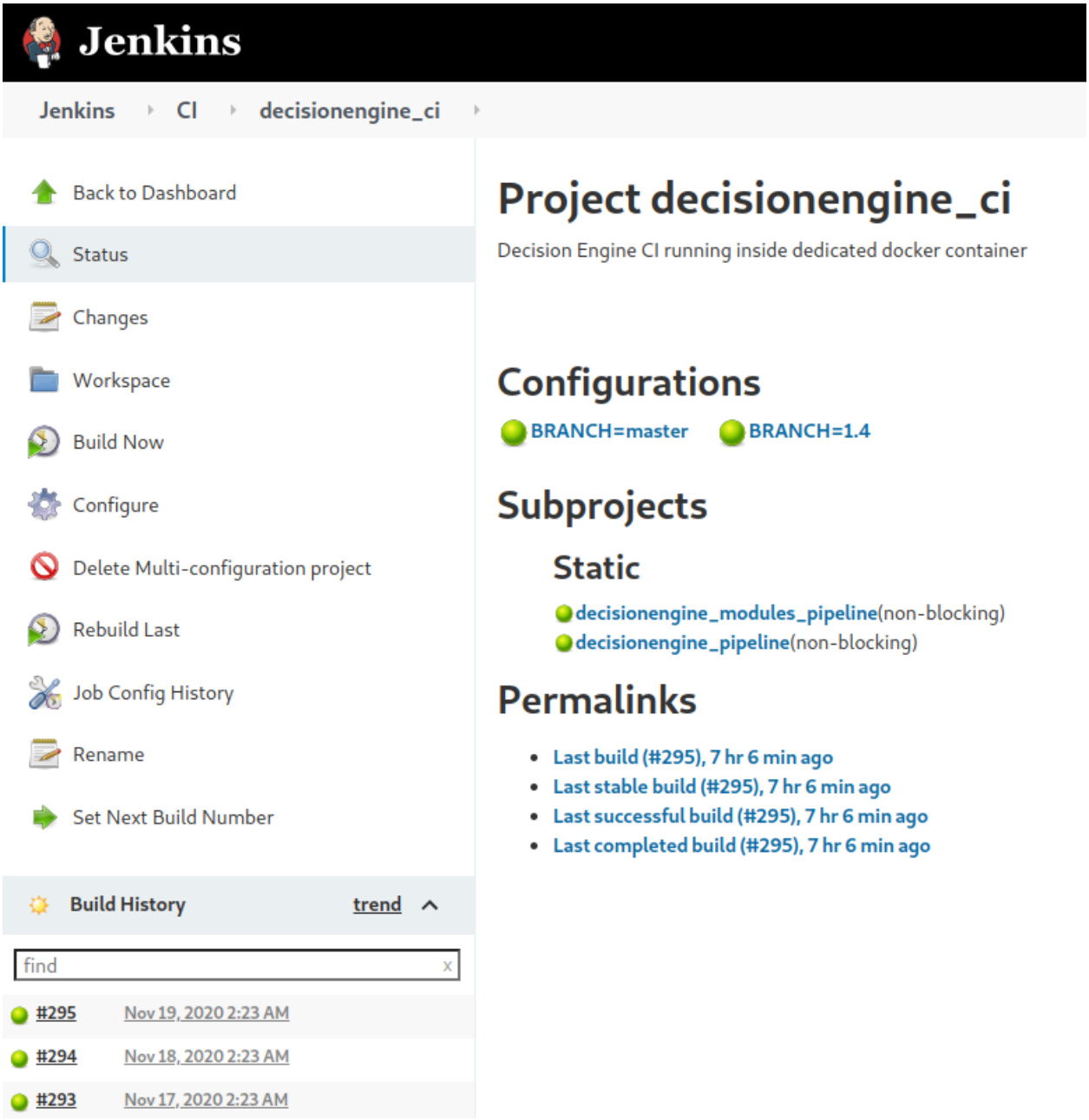
The *DE_MOD_REPO* parameter can point to the user fork or to the main repository.
The *BRANCH* parameter can point to the desired branch to test.
The *PYTEST_TIMEOUT* parameter is the timeout in seconds for *unit_tests*.

When ready, by clicking on the *Build* button, the CI build will start.

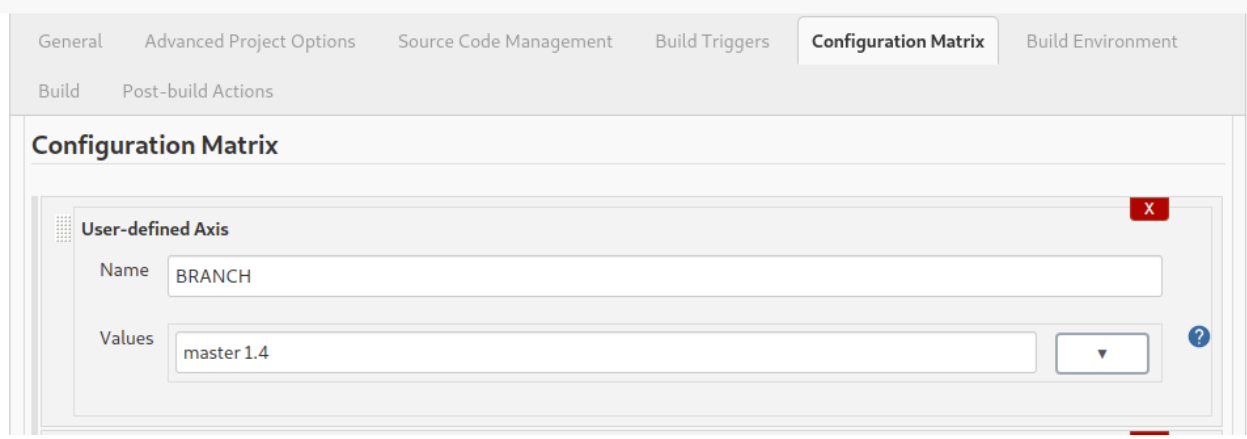
The pipeline configuration is part of the decisionengine_modules repo.

3.1.1 Nightly CI build configuration

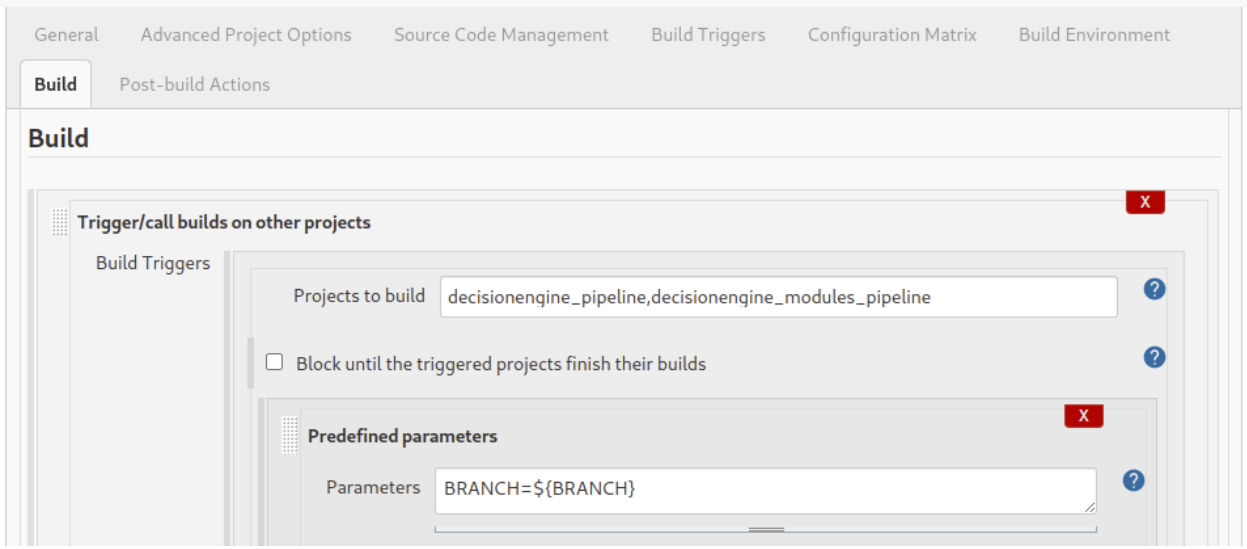
The nightly CI build for Decisionengine framework uses this Jenkins project that triggers a CI build using the Jenkins pipeline described above to test a list of predefined branches.



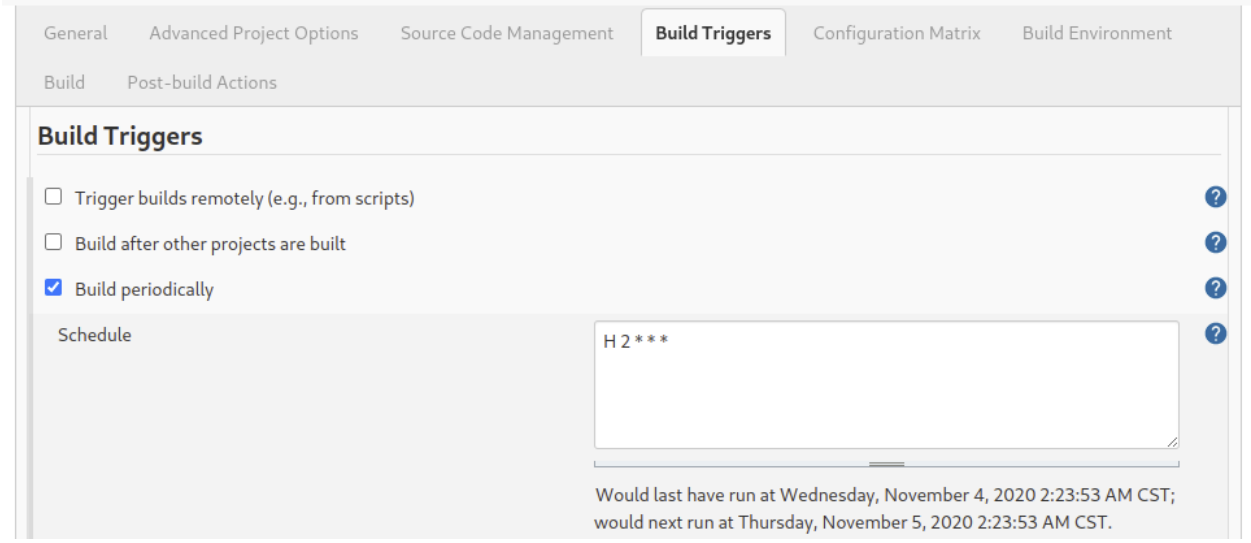
Branches to test are defined using the project matrix as shown in the picture below.
Each branch in the list (here *master* and *1.4*) spawns an independent CI build.



In the *Build* section of the configuration it is set the list of Jenkins subprojects to be triggered, in this case we have *decisionengine_pipeline* and *decisionengine_modules_pipeline*.
The *Parameters* text box is used to override parameters of each Jenkins subproject with a custom value.
In total this Jenkins project triggers 4 CI builds, i.e. 2 branches X 2 Jenkins subprojects.



Finally the *Build Triggers* section is used to setup the schedule for the periodic build,
in this case it is scheduled to run at about 2 AM.
Jenkins will choose the actual time depending on the actual load on the system.


General
Advanced Project Options
Source Code Management
Build Triggers
Configuration Matrix
Build Environment
Build
Post-build Actions
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
Schedule
H 2 * * *
Would last have run at Wednesday, November 4, 2020 2:23:53 AM CST;
would next run at Thursday, November 5, 2020 2:23:53 AM CST.

CHAPTER

FOUR

SOURCE CODE

4.1 Welcome to decisionengine_modules's documentation!

4.1.1 decisionengine_modules package

Subpackages

decisionengine_modules.AWS package

Subpackages

decisionengine_modules.AWS.publishers package

Submodules

decisionengine_modules.AWS.publishers.AWS_burn_rate module

decisionengine_modules.AWS.publishers.AWS_figure_of_merit module

decisionengine_modules.AWS.publishers.AWS_generic_publisher module

decisionengine_modules.AWS.publishers.AWS_price_performance module

Module contents

decisionengine_modules.AWS.sources package

Submodules

decisionengine_modules.AWS.sources.AWSBurnRateSourceProxy module

decisionengine_modules.AWS.sources.AWSInstancePerformance module

decisionengine_modules.AWS.sources.AWSJobLimits module

decisionengine_modules.AWS.sources.AWSOccupancy module

decisionengine_modules.AWS.sources.AWSOccupancyWithSourceProxy module

decisionengine_modules.AWS.sources.AWSSpotPrice module

decisionengine_modules.AWS.sources.AWSSpotPriceWithSourceProxy module

decisionengine_modules.AWS.sources.BillingInfo module

decisionengine_modules.AWS.sources.BillingInfoSourceProxy module

decisionengine_modules.AWS.sources.DEAccountContants module

This data is taken from billing and arranged as dictionary of AccountConstants instead of separate class for each billing class. This way is more covenient to use in programms.

class decisionengine_modules.AWS.sources.DEAccountContants.**AccountConstants**(*constants={}*)
Bases: object

info()

decisionengine_modules.AWS.sources.DEAccountContants.**load_constants**(*constants_file*)
Load constants from file.

Parameters **constants_file** (file) – configuration file as python dict

Return type *AccountConstants*

decisionengine_modules.AWS.sources.FigureOfMeritSourceProxy module

decisionengine_modules.AWS.sources.FinancialParameters module

decisionengine_modules.AWS.sources.occupancy_config_sample module

decisionengine_modules.AWS.sources.spot_price_config_sample module

Module contents

decisionengine_modules.AWS.transforms package

Submodules

decisionengine_modules.AWS.transforms.AwsBurnRate module

decisionengine_modules.AWS.transforms.FigureOfMerit module

Module contents

Submodules

decisionengine_modules.AWS.AWSSpotPrice module

decisionengine_modules.AWS.CloudBudget module

decisionengine_modules.AWS.CloudRequests module

decisionengine_modules.AWS.HPCAllocation module

decisionengine_modules.AWS.HTCondorJobQ module

decisionengine_modules.AWS.LocalCapacity module

decisionengine_modules.AWS.ProvisionerResourceList module

decisionengine_modules.AWS.pandas_scratch_pad module

decisionengine_modules.AWS.pandas_scratch_pad.**load_data_frame**(*list_of_dicts*)

Module contents

decisionengine_modules.GCE package

Subpackages

decisionengine_modules.GCE.publishers package

Submodules

decisionengine_modules.GCE.publishers.GCEBurnRatePublisher module

decisionengine_modules.GCE.publishers.GCEFigureOfMerit_publisher module

decisionengine_modules.GCE.publishers.GCEPricePerformance_publisher module

Module contents

decisionengine_modules.GCE.sources package

Submodules

decisionengine_modules.GCE.sources.GCEBillingInfo module

decisionengine_modules.GCE.sources.GCEBillingInfoSourceProxy module

decisionengine_modules.GCE.sources.GCEInstancePerformance module

decisionengine_modules.GCE.sources.GCEResourceLimits module

decisionengine_modules.GCE.sources.GceBurnRateSourceProxy module

decisionengine_modules.GCE.sources.GceFigureOfMeritSourceProxy module

decisionengine_modules.GCE.sources.GceOccupancy module

Module contents

decisionengine_modules.GCE.transforms package

Submodules

decisionengine_modules.GCE.transforms.GceBurnRate module

decisionengine_modules.GCE.transforms.GceFigureOfMerit module

Module contents

Module contents

decisionengine_modules.NERSC package

Subpackages

decisionengine_modules.NERSC.publishers package

Submodules

decisionengine_modules.NERSC.publishers.NerscFigureOfMerit_publisher module

Module contents

decisionengine_modules.NERSC.sources package

Submodules

decisionengine_modules.NERSC.sources.NerscAllocationInfo module

decisionengine_modules.NERSC.sources.NerscAllocationInfoSourceProxy module

decisionengine_modules.NERSC.sources.NerscFigureOfMeritSourceProxy module

decisionengine_modules.NERSC.sources.NerscInstancePerformance module

decisionengine_modules.NERSC.sources.NerscJobInfo module

Module contents

decisionengine_modules.NERSC.transforms package

Submodules

decisionengine_modules.NERSC.transforms.CompareNerscFactoryJobs module

decisionengine_modules.NERSC.transforms.CompareNerscUserpoolSlots module

decisionengine_modules.NERSC.transforms.NerscFigureOfMerit module

Module contents

decisionengine_modules.NERSC.util package

Submodules

decisionengine_modules.NERSC.util.newt module

Newt API interface

class decisionengine_modules.NERSC.util.newt.**Newt**(*password_file*, *newt_base_url=None*, *num_retries=0*, *retry_backoff_factor=0*)

Bases: object

_add_retries_to_session()
Adds retries to requests Session for requests to NEWT URLs :return: void

_login()
Establishes newt Session :return: void

get_queue(*system*, *query=None*)
Returns information about queues controlled by optional query name parameter :param system: string system name (like 'cori', 'edison' ..) :param query: string query (optional) :return: json containing queue information

get_status(*system=None*)
Returns system status for a given system (if provided) or all systems :param system: string name of the system (optional) :return: json containing system status

get_usage(*username*)
Returns allocation and usage for the given user using NEWT IRIS API :param username: string username :return: json containing allocation and usage information for a given user

Module contents

Module contents

decisionengine_modules.graphite package

Subpackages

decisionengine_modules.graphite.publishers package

Submodules

decisionengine_modules.graphite.publishers.generic_publisher module

Module contents

Module contents

decisionengine_modules.htcondor package

Subpackages

decisionengine_modules.htcondor.publishers package

Submodules

decisionengine_modules.htcondor.publishers.publisher module

Module contents

decisionengine_modules.htcondor.sources package

Submodules

decisionengine_modules.htcondor.sources.job_q module

decisionengine_modules.htcondor.sources.job_q_source_proxy module

decisionengine_modules.htcondor.sources.slots module

decisionengine_modules.htcondor.sources.slots_source_proxy module

decisionengine_modules.htcondor.sources.source module

Module contents

Submodules

decisionengine_modules.htcondor.htcondor_query module

class decisionengine_modules.htcondor.htcondor_query.**CondorQ**(*schedd_name=None*, *pool_name=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.CondorQuery*

Class to implement condor_q

_abc_cache = <_weakrefset.WeakSet object>

_abc_negative_cache = <_weakrefset.WeakSet object>

_abc_negative_cache_version = 192

_abc_registry = <_weakrefset.WeakSet object>

fetch(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch job classads

class decisionengine_modules.htcondor.htcondor_query.**CondorQuery**(*resource_str*, *group_attr*, *pool_name=None*, *env=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.Query*

Fully implemented class for CondorQuery

_abc_cache = <_weakrefset.WeakSet object>

_abc_negative_cache = <_weakrefset.WeakSet object>

_abc_negative_cache_version = 192

_abc_registry = <_weakrefset.WeakSet object>

class decisionengine_modules.htcondor.htcondor_query.**CondorStatus**(*subsystem_name=None*, *pool_name=None*, *group_attr=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.CondorQuery*

Class to implement condor_status

_abc_cache = <_weakrefset.WeakSet object>

_abc_negative_cache = <_weakrefset.WeakSet object>

_abc_negative_cache_version = 192

_abc_registry = <_weakrefset.WeakSet object>

fetch(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch resource classads and return a list of evaluated classads

class decisionengine_modules.htcondor.htcondor_query.**Query**

Bases: object

Pure virtual class to have a minimum set of methods defined

_abc_cache = <_weakrefset.WeakSet object>

_abc_negative_cache = <_weakrefset.WeakSet object>

_abc_negative_cache_version = 192

_abc_registry = <_weakrefset.WeakSet object>

abstract fetch(*constraint=None*, *format_list=None*, *condor_config=None*)
Fetch the classad attributes specified in the format_list matching the constraint

fetch_stored(*constraint_func=None*)

Parameters **constraint_func** – A boolean function, with only one argument (data el). If None, return all the data.

@rtype: dict @return: Same as fetch(), but limited to constraint_func(el)==True

load(*constraint=None*, *format_list=None*, *condor_config=None*)
Fetch the data and store it in self.stored_data

exception decisionengine_modules.htcondor.htcondor_query.**QueryError**(*err_str*)
Bases: RuntimeError

Thrown when there are exceptions using htcondor python bindings or commands

decisionengine_modules.htcondor.htcondor_query.**apply_constraint**(*data*, *constraint_func*)
Return a subset of data that satisfies constraint_function If constraint_func is None, return back entire data

decisionengine_modules.htcondor.htcondor_query.**bindings_friendly_attrs**(*format_list*)
Convert the format_list into attrs that can be used with python bindings Python bindings should take care of the typing

decisionengine_modules.htcondor.htcondor_query.**bindings_friendly_constraint**(*constraint*)
Convert the constraint to format that can be used with python bindings

decisionengine_modules.htcondor.htcondor_query.**eval_classad_expr**(*classads*, *format_list=None*)
Convert the classads into list of classads Guarantees any attribute in format_list to be added with default value None

decisionengine_modules.htcondor.htcondor_query.**list2dict**(*list_data*, *attr_name*)
Convert a list to a dictionary and group the results based on attributes specified by attr_name

decisionengine_modules.htcondor.htcondor_query.**resource_str_to_py_adtype**(*resource_str*)
Given the resource string return equivalent classad type

decisionengine_modules.htcondor.htcondor_query.**split_collector_host**(*collector_host*)
Given the collector_host string extract primary collector_host which is the first collector host in the list and sort the secondary collectors. Return a tuple of primary collector and CSV secondary collector.

Module contents

decisionengine_modules.util package

Submodules

decisionengine_modules.util.figure_of_merit module

Calculate figure of merit

decisionengine_modules.util.figure_of_merit.**figure_of_merit**(*performance*, *running*, *allowed*, *idle=None*, *max_idle=None*)

decisionengine_modules.util.retry_function module

decisionengine_modules.util.retry_function.**retry_on_error**(*nretries=1*, *retry_interval=2*, *backoff=True*)

decisionengine_modules.util.retry_function.**retry_wrapper**(*f*, *nretries=1*, *retry_interval=2*, *backoff=True*)

Retry on error with parameters of how many times (nretries) and interval in seconds (retry_interval). If the function to be decorated is an instance method and the values come from a configuration, then this decorator can retrieve these two parameters from instance attributes with the same name. Otherwise, use the default values or pass new values to the decorator.

decisionengine_modules.util.test_retry_function module

class decisionengine_modules.util.test_retry_function.**Dummy**(*name='Dummy'*)

Bases: object

_func_failure(*foo*, *input2=3*)

_func_success(*foo*, *input2=3*)

func_failure(*foo*, *input2=3*)

func_success(*foo*, *input2=3*)

decisionengine_modules.util.test_retry_function.**test_all**()

decisionengine_modules.util.testutils module

Utils to simplify testing

decisionengine_modules.util.testutils.**compare_dfs**(*df1*, *df2*)

for some reason df.equals does not work here but if I compare cell by cell it works :type df1: pd.DataFrame :arg df1: data frame instance :type df2: pd.DataFrame :arg df2: data frame instance :rtype: bool - True if equal

decisionengine_modules.util.testutils.**input_from_file**(*fname*)

decisionengine_modules.util.testutils.**raw_input_from_file**(*fname*)

Module contents

Submodules

decisionengine_modules.about module

PEP-0396 provides instructions for providing module versions While we are at it, add a few other useful bits

decisionengine_modules.graphite_client module

class decisionengine_modules.graphite_client.**Graphite**(*host='fifemondata.fnal.gov'*, *pickle_port=2004*)

Bases: object

send_dict(*namespace*, *data*, *debug_print=True*, *send_data=True*)

send data contained in dictionary as {k: v} to graphite dataset $namespace.k with current timestamp

decisionengine_modules.graphite_client.**sanitize_key**(*key*)

decisionengine_modules.load_config module

decisionengine_modules.load_config.**load**(*python_file*, *retries=0*, *timeout=0*)

Load constants from file.

Parameters **constants_file** (file) – configuration file as python dict

Return type object

Module contents

4.2 Indices and tables

- genindex
- modindex
- search

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

decisionengine_modules, 26
decisionengine_modules.about, 25
decisionengine_modules.AWS, 19
decisionengine_modules.AWS.pandas_scratch_pad, 19
decisionengine_modules.AWS.publishers, 17
decisionengine_modules.AWS.sources, 18
decisionengine_modules.AWS.sources.DEAccountContants, 18
decisionengine_modules.AWS.sources.occupancy_config_sample, 18
decisionengine_modules.AWS.sources.spot_price_config_sample, 18
decisionengine_modules.AWS.transforms, 18
decisionengine_modules.GCE, 20
decisionengine_modules.GCE.publishers, 19
decisionengine_modules.GCE.sources, 20
decisionengine_modules.GCE.transforms, 20
decisionengine_modules.graphite, 22
decisionengine_modules.graphite.publishers, 22
decisionengine_modules.graphite_client, 26
decisionengine_modules.htcondor, 24
decisionengine_modules.htcondor.htcondor_query, 23
decisionengine_modules.htcondor.publishers, 22
decisionengine_modules.htcondor.sources, 22
decisionengine_modules.load_config, 26
decisionengine_modules.NERSC, 22
decisionengine_modules.NERSC.publishers, 20
decisionengine_modules.NERSC.sources, 21
decisionengine_modules.NERSC.transforms, 21
decisionengine_modules.NERSC.util, 22
decisionengine_modules.NERSC.util.newt, 21
decisionengine_modules.util, 25
decisionengine_modules.util.figure_of_merit, 24
decisionengine_modules.util.retry_function, 25
decisionengine_modules.util.test_retry_function, 25
decisionengine_modules.util.testutils, 25

INDEX

Symbols

_abc_cache (*decisionengine_modules.htcondor.htcondor_query.CondorQ attribute*), 23
_abc_cache (*decisionengine_modules.htcondor.htcondor_query.CondorQuery attribute*), 23
_abc_cache (*decisionengine_modules.htcondor.htcondor_query.CondorStatus attribute*), 23
_abc_cache (*decisionengine_modules.htcondor.htcondor_query.Query attribute*), 23
_abc_negative_cache (*decisionengine_modules.htcondor.htcondor_query.CondorQ attribute*), 23
_abc_negative_cache (*decisionengine_modules.htcondor.htcondor_query.CondorQuery attribute*), 23
_abc_negative_cache (*decisionengine_modules.htcondor.htcondor_query.CondorStatus attribute*), 23
_abc_negative_cache (*decisionengine_modules.htcondor.htcondor_query.Query attribute*), 23
_abc_negative_cache_version (*decisionengine_modules.htcondor.htcondor_query.CondorQ attribute*), 23
_abc_negative_cache_version (*decisionengine_modules.htcondor.htcondor_query.CondorQuery attribute*), 23
_abc_negative_cache_version (*decisionengine_modules.htcondor.htcondor_query.CondorStatus attribute*), 23
_abc_negative_cache_version (*decisionengine_modules.htcondor.htcondor_query.Query attribute*), 23
_abc_registry (*decisionengine_modules.htcondor.htcondor_query.CondorQ attribute*), 23
_abc_registry (*decisionengine_modules.htcondor.htcondor_query.CondorQuery attribute*), 23
_abc_registry (*decisionengine_modules.htcondor.htcondor_query.CondorStatus attribute*), 23
_abc_registry (*decisionengine_modules.htcondor.htcondor_query.Query attribute*), 23
_add_retries_to_session() (*decisionengine_modules.NERSC.util.newt.Newt method*), 21
_func_failure() (*decisionengine_modules.util.test_retry_function.Dummy method*), 25
_func_success() (*decisionengine_modules.util.test_retry_function.Dummy method*), 25
_login() (*decisionengine_modules.NERSC.util.newt.Newt method*), 21

A

AccountConstants (*class in decisionengine_modules.AWS.sources.DEAccountContants*), 18
apply_constraint() (*in module decisionengine_modules.htcondor.htcondor_query*), 24

B

bindings_friendly_attrs() (*in module decisionengine_modules.htcondor.htcondor_query*), 24
bindings_friendly_constraint() (*in module decisionengine_modules.htcondor.htcondor_query*), 24

C

compare_dfs() (*in module decisionengine_modules.util.testutils*), 25
CondorQ (*class in decisionengine_modules.htcondor.htcondor_query*), 23

CondorQuery (*class in decisionengine_modules.htcondor.htcondor_query*), 23
CondorStatus (*class in decisionengine_modules.htcondor.htcondor_query*), 23

D

decisionengine_modules
    module, 26
decisionengine_modules.about
    module, 25
decisionengine_modules.AWS
    module, 19
decisionengine_modules.AWS.pandas_scratchpad
    module, 19
decisionengine_modules.AWS.publishers
    module, 17
decisionengine_modules.AWS.sources
    module, 18
decisionengine_modules.AWS.sources.DEAccountContants
    module, 18
decisionengine_modules.AWS.sources.occupancy_config_sample
    module, 18
decisionengine_modules.AWS.sources.spot_price_config_sample
    module, 18
decisionengine_modules.AWS.transforms
    module, 18
decisionengine_modules.GCE
    module, 20
decisionengine_modules.GCE.publishers
    module, 19
decisionengine_modules.GCE.sources
    module, 20
decisionengine_modules.GCE.transforms
    module, 20
decisionengine_modules.graphite
    module, 22
decisionengine_modules.graphite.publishers
    module, 22
decisionengine_modules.graphite_client
    module, 26
decisionengine_modules.htcondor
    module, 24
decisionengine_modules.htcondor.htcondor_query
    module, 23
decisionengine_modules.htcondor.publishers
    module, 22
decisionengine_modules.htcondor.sources
    module, 22
decisionengine_modules.load_config
    module, 26
decisionengine_modules.NERSC
    module, 22
decisionengine_modules.NERSC.publishers
    module, 20
decisionengine_modules.NERSC.sources
    module, 21
decisionengine_modules.NERSC.transforms
    module, 21
decisionengine_modules.NERSC.util
    module, 22
decisionengine_modules.NERSC.util.newt
    module, 21
decisionengine_modules.util
    module, 25
decisionengine_modules.util.figure_of_merit
    module, 24
decisionengine_modules.util.retry_function
    module, 25
decisionengine_modules.util.test_retry_function
    module, 25
decisionengine_modules.util.testutils
    module, 25
DummyContants (*class in decisionengine_modules.util.test_retry_function*), 25

E

eval_classad_expr() (*in module decisionengine_modules.htcondor.htcondor_query*), 24

F

fetch() (*decisionengine_modules.htcondor.htcondor_query.CondorQ method*), 23
fetch() (*decisionengine_modules.htcondor.htcondor_query.CondorStatu method*), 23
fetch() (*decisionengine_modules.htcondor.htcondor_query.Query method*), 24
fetch_stored() (*decisionengine_modules.htcondor.htcondor_query.Query method*), 24
figure_of_merit() (*in module decisionengine_modules.util.figure_of_merit*), 24
func_failure() (*decisionengine_modules.util.test_retry_function.Dummy method*), 25
func_success() (*decisionengine_modules.util.test_retry_function.Dummy method*), 25

G

get_queue() (*decisionengine_modules.NERSC.util.newt.Newt method*), 21

get_status() (*decisionengine_modules.NERSC.util.newt.Newt method*), 21
get_usage() (*decisionengine_modules.NERSC.util.newt.Newt method*), 21
Graphite (*class in decisionengine_modules.graphite_client*), 26

I

info() (*decisionengine_modules.AWS.sources.DEAccountContants.AccountConstants method*), 18
input_from_file() (*in module decisionengine_modules.util.testutils*), 25

L

list2dict() (*in module decisionengine_modules.htcondor.htcondor_query*), 24
load() (*decisionengine_modules.htcondor.htcondor_query.Query method*), 24
load() (*in module decisionengine_modules.load_config*), 26
load_constants() (*in module decisionengine_modules.AWS.sources.DEAccountContants*), 18
load_data_frame() (*in module decisionengine_modules.AWS.pandas_scratch_pad*), 19

M

module
decisionengine_modules, 26
decisionengine_modules.about, 25
decisionengine_modules.AWS, 19
decisionengine_modules.AWS.pandas_scratch_pad, 19
decisionengine_modules.AWS.publishers, 17
decisionengine_modules.AWS.sources, 18
decisionengine_modules.AWS.sources.DEAccountContants, 18
decisionengine_modules.AWS.sources.occupancy_config_sample, 18
decisionengine_modules.AWS.sources.spot_price_config_sample, 18
decisionengine_modules.AWS.transforms, 18
decisionengine_modules.GCE, 20
decisionengine_modules.GCE.publishers, 19
decisionengine_modules.GCE.sources, 20
decisionengine_modules.GCE.transforms, 20
decisionengine_modules.graphite, 22
decisionengine_modules.graphite.publishers, 22
decisionengine_modules.graphite_client, 26
decisionengine_modules.htcondor, 24
decisionengine_modules.htcondor.htcondor_query, 23
decisionengine_modules.htcondor.publishers, 22
decisionengine_modules.htcondor.sources, 22
decisionengine_modules.load_config, 26
decisionengine_modules.NERSC, 22
decisionengine_modules.NERSC.publishers, 20
decisionengine_modules.NERSC.sources, 21
decisionengine_modules.NERSC.transforms, 21
decisionengine_modules.NERSC.util, 22
decisionengine_modules.NERSC.util.newt, 21
decisionengine_modules.util, 25
decisionengine_modules.util.figure_of_merit, 24
decisionengine_modules.util.retry_function, 25
decisionengine_modules.util.test_retry_function, 25
decisionengine_modules.util.testutils, 25

N

Newt (*class in decisionengine_modules.NERSC.util.newt*), 21

Q

Query (*class in decisionengine_modules.htcondor.htcondor_query*), 23
QueryError, 24

R

raw_input_from_file() (*in module decisionengine_modules.util.testutils*), 25
resource_str_to_py_adtype() (*in module decisionengine_modules.htcondor.htcondor_query*), 24

retry_on_error() (*in module decisionengine_modules.util.retry_function*), 25
retry_wrapper() (*in module decisionengine_modules.util.retry_function*), 25

S

sanitize_key() (*in module decisionengine_modules.graphite_client*), 26
send_dict() (*decisionengine_modules.graphite_client.Graphite method*), 26
split_collector_host() (*in module decisionengine_modules.htcondor.htcondor_query*), 24

T

test_all() (*in module decisionengine_modules.util.test_retry_function*), 25