Elasticsearch Curator Documentation

Release 4.3.0

Aaron Mildenstein

Apr 28, 2017

Contents

The Elasticsearch Curator Python API helps you manage your indices and snapshots.

Note: This documentation is for the Elasticsearch Curator Python API. Documentation for the Elasticsearch Curator *CLI* – which uses this API and is installed as an entry_point as part of the package – is available in the Elastic guide.

Compatibility

The Elasticsearch Curator Python API is compatible with Elasticsearch versions 2.x through 5.0, and supports Python versions 2.6 and later.

Example Usage

```python
import elasticsearch
import curator

client = elasticsearch.Elasticsearch()

ilo = curator.IndexList(client)
ilo.filter_by_regex(kind='prefix', value='logstash-')
ilo.filter_by_age(source='name', direction='older', timestring='%Y.%m.%d', unit='days
→', unit_count=30)
delete_indices = curator.DeleteIndices(ilo)
delete_indices.do_action()
```

Tip: See more examples in the *Examples* page.

Features

The API methods fall into the following categories:

- *Object Classes* build and filter index list or snapshot list objects.

- *Action Classes* act on object classes.

- *Utilities* are helper methods.

Logging

The Elasticsearch Curator Python API uses the standard logging library from Python. It inherits two loggers from `elasticsearch-py`: `elasticsearch` and `elasticsearch.trace`. Clients use the `elasticsearch` logger to log standard activity, depending on the log level. The `elasticsearch.trace` logger logs requests to the server in JSON format as pretty-printed `curl` commands that you can execute from the command line. The `elasticsearch.trace` logger is not inherited from the base logger and must be activated separately.

Contents

Object Classes

- *IndexList*
- *SnapshotList*

IndexList

class curator.indexlist.**IndexList**(*client*)

> **all_indices** = **None**
>   Instance variable. All indices in the cluster at instance creation time. **Type:** list()
>
> **client** = **None**
>   An Elasticsearch Client object Also accessible as an instance variable.
>
> **empty_list_check**()
>   Raise exception if *indices* is empty
>
> **filter_allocated**(*key=None*, *value=None*, *allocation_type='require'*, *exclude=True*)
>   Match indices that have the routing allocation rule of *key=value* from *indices*
>
> > **Parameters**
> >
> > - **key** – The allocation attribute to check for
> >
> > - **value** – The value to check for
> >
> > - **allocation_type** – Type of allocation to apply
> >
> > - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

filter_by_age(*source='name'*, *direction=None*, *timestring=None*, *unit=None*, *unit_count=None*, *field=None*, *stats_result='min_value'*, *epoch=None*, *exclude=False*)

Match *indices* by relative age calculations.

> **Parameters**
>
> - **source** – Source of index age. Can be one of 'name', 'creation_date', or 'field_stats'
>
> - **direction** – Time to filter, either `older` or `younger`
>
> - **timestring** – An strftime string to match the datestamp in an index name. Only used for index filtering by `name`.
>
> - **unit** – One of `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`.
>
> - **unit_count** – The number of `unit``s. ``unit_count * unit` will be calculated out to the relative number of seconds.
>
> - **field** – A timestamp field name. Only used for `field_stats` based calculations.
>
> - **stats_result** – Either *min_value* or *max_value*. Only used in conjunction with *source*'=``field_stats`` to choose whether to reference the minimum or maximum result value.
>
> - **epoch** – An epoch timestamp used in conjunction with `unit` and `unit_count` to establish a point of reference for calculations. If not provided, the current time will be used.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

filter_by_alias(*aliases=None*, *exclude=False*)

Match indices which are associated with the alias identified by *name*

> **Parameters**
>
> - **aliases** (*[list](#)*) – A list of alias names.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

filter_by_count(*count=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *field=None*, *stats_result='min_value'*, *exclude=True*)

Remove indices from the actionable list beyond the number *count*, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.

The default is usually what you will want. If only one kind of index is provided–for example, indices matching `logstash-%Y.%m.%d`–then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older indices.

By setting *reverse* to *False*, then `index3` will be deleted before `index2`, which will be deleted before `index1`

use_age allows ordering indices by age. Age is determined by the index creation date by default, but you can specify an *source* of `name`, `max_value`, or `min_value`. The `name` *source* requires the timestring argument.

> **Parameters**
>
> - **count** – Filter indices beyond *count*.
>
> - **reverse** – The filtering direction. (default: *True*).
>
> - **use_age** – Sort indices by age. `source` is required in this case.
>
> - **source** – Source of index age. Can be one of `name`, `creation_date`, or `field_stats`. Default: `creation_date`

- **timestring** – An strftime string to match the datestamp in an index name. Only used if *source* name is selected.

- **field** – A timestamp field name. Only used if *source* field_stats is selected.

- **stats_result** – Either *min_value* or *max_value*. Only used if *source* field_stats is selected. It determines whether to reference the minimum or maximum value of *field* in each index.

- **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

filter_by_regex(*kind=None*, *value=None*, *exclude=False*)

Match indices by regular expression (pattern).

> **Parameters**
>
> - **kind** – Can be one of: suffix, prefix, regex, or timestring. This option defines what kind of filter you will be building.
>
> - **value** – Depends on *kind*. It is the strftime string if *kind* is timestring. It's used to build the regular expression for other kinds.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

filter_by_space(*disk_space=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *field=None*, *stats_result='min_value'*, *exclude=False*)

Remove indices from the actionable list based on space consumed, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.

The default is usually what you will want. If only one kind of index is provided–for example, indices matching logstash-%Y.%m.%d–then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older indices.

By setting *reverse* to *False*, then index3 will be deleted before index2, which will be deleted before index1

use_age allows ordering indices by age. Age is determined by the index creation date by default, but you can specify an *source* of name, max_value, or min_value. The name *source* requires the timestring argument.

> **Parameters**
>
> - **disk_space** – Filter indices over *n* gigabytes
>
> - **reverse** – The filtering direction. (default: *True*). Ignored if *use_age* is *True*
>
> - **use_age** – Sort indices by age. source is required in this case.
>
> - **source** – Source of index age. Can be one of name, creation_date, or field_stats. Default: creation_date
>
> - **timestring** – An strftime string to match the datestamp in an index name. Only used if *source* name is selected.
>
> - **field** – A timestamp field name. Only used if *source* field_stats is selected.
>
> - **stats_result** – Either *min_value* or *max_value*. Only used if *source* field_stats is selected. It determines whether to reference the minimum or maximum value of *field* in each index.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

---

filter_closed(*exclude=True*)
>   Filter out closed indices from *indices*

>> **Parameters exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

filter_forceMerged(*max_num_segments=None*, *exclude=True*)
>   Match any index which has *max_num_segments* per shard or fewer in the actionable list.

>> **Parameters**

>>> - **max_num_segments** – Cutoff number of segments per shard.

>>> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

filter_kibana(*exclude=True*)
>   Match any index named `.kibana`, `kibana-int`, `.marvel-kibana`, or `.marvel-es-data` in *indices*.

>> **Parameters exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

filter_opened(*exclude=True*)
>   Filter out opened indices from *indices*

>> **Parameters exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

index_info = None
>   Instance variable. Information extracted from indices, such as segment count, age, etc. Populated at instance creation time, and by other private helper methods, as needed. **Type:** `dict()`

indices = None
>   Instance variable. The running list of indices which will be used by an Action class. Populated at instance creation time. **Type:** `list()`

iterate_filters(*filter_dict*)
>   Iterate over the filters defined in *config* and execute them.

>> **Parameters filter_dict** – The configuration dictionary

Note: *filter_dict* should be a dictionary with the following form:

```
{ 'filters' : [
        {
            'filtertype': 'the_filter_type',
            'key1' : 'value1',
            ...
            'keyN' : 'valueN'
        }
    ]
}
```

working_list()
>   Return the current value of *indices* as copy-by-value to prevent list stomping during iterations

---

SnapshotList

class curator.snapshotlist.**SnapshotList**(*client*, *repository=None*)

>
> **client = None**
>   An Elasticsearch Client object. Also accessible as an instance variable.
>
> **empty_list_check**()
>   Raise exception if *snapshots* is empty
>
> **filter_by_age**(*source='creation_date'*, *direction=None*, *timestring=None*, *unit=None*, *unit_count=None*, *epoch=None*, *exclude=False*)
>   Remove snapshots from *snapshots* by relative age calculations.
>
>> **Parameters**
>>
>> - **source** – Source of snapshot age. Can be 'name', or 'creation_date'.
>>
>> - **direction** – Time to filter, either `older` or `younger`
>>
>> - **timestring** – An strftime string to match the datestamp in an snapshot name. Only used for snapshot filtering by `name`.
>>
>> - **unit** – One of `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`.
>>
>> - **unit_count** – The number of unit``s. ``unit_count * unit will be calculated out to the relative number of seconds.
>>
>> - **epoch** – An epoch timestamp used in conjunction with `unit` and `unit_count` to establish a point of reference for calculations. If not provided, the current time will be used.
>>
>> - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*
>
> **filter_by_count**(*count=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *exclude=True*)
>   Remove snapshots from the actionable list beyond the number *count*, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.
>
>   The default is usually what you will want. If only one kind of snapshot is provided–for example, snapshots matching `curator-%Y%m%d%H%M%S`– then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older snapshots.
>
>   By setting *reverse* to *False*, then `snapshot3` will be acted on before `snapshot2`, which will be acted on before `snapshot1`
>
>   *use_age* allows ordering snapshots by age. Age is determined by the snapshot creation date (as identified by `start_time_in_millis`) by default, but you can also specify a *source* of `name`. The `name` *source* requires the timestring argument.
>
>> **Parameters**
>>
>> - **count** – Filter snapshots beyond *count*.
>>
>> - **reverse** – The filtering direction. (default: *True*).
>>
>> - **use_age** – Sort snapshots by age. `source` is required in this case.
>>
>> - **source** – Source of snapshot age. Can be one of `name`, or `creation_date`. Default: `creation_date`
>>
>> - **timestring** – An strftime string to match the datestamp in a snapshot name. Only used if *source* `name` is selected.

- **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *True*

filter_by_regex(*kind=None*, *value=None*, *exclude=False*)
Filter out snapshots not matching the pattern, or in the case of exclude, filter those matching the pattern.

   Parameters

   - **kind** – Can be one of: `suffix`, `prefix`, `regex`, or `timestring`. This option defines what kind of filter you will be building.

   - **value** – Depends on *kind*. It is the strftime string if *kind* is *timestring*. It's used to build the regular expression for other kinds.

   - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*

filter_by_state(*state=None*, *exclude=False*)
Filter out snapshots not matching `state`, or in the case of exclude, filter those matching `state`.

   Parameters

   - **state** – The snapshot state to filter for. Must be one of `SUCCESS`, `PARTIAL`, `FAILED`, or `IN_PROGRESS`.

   - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*

iterate_filters(*config*)
Iterate over the filters defined in *config* and execute them.

   Parameters config – A dictionary of filters, as extracted from the YAML configuration file.

Note: *config* should be a dictionary with the following form:

```
{ 'filters' : [
        {
            'filtertype': 'the_filter_type',
            'key1' : 'value1',
            ...
            'keyN' : 'valueN'
        }
    ]
}
```

most_recent()
Return the most recent snapshot based on *start_time_in_millis*.

repository = None
An Elasticsearch repository. Also accessible as an instance variable.

snapshot_info = None
Instance variable. Information extracted from snapshots, such as age, etc. Populated by internal method *__get_snapshots* at instance creation time. **Type:** `dict()`

snapshots = None
Instance variable. The running list of snapshots which will be used by an Action class. Populated by internal methods *__get_snapshots* at instance creation time. **Type:** `list()`

---

working_list()
> Return the current value of *snapshots* as copy-by-value to prevent list stomping during iterations

Action Classes

See also:

It is important to note that each action has a *do_action()* method, which accepts no arguments. This is the means by which all actions are executed.

- *Alias*
- *Allocation*
- *Close*
- *ClusterRouting*
- *DeleteIndices*
- *DeleteSnapshots*
- *ForceMerge*
- *Open*
- *Replicas*
- *Snapshot*

Alias

class curator.actions.**Alias**(*name=None*, *extra_settings={}*, ***kwargs*)
> Define the Alias object.

> > **Parameters**

> > > - **name** – The alias name

> > > - **extra_settings** (*dict, representing the settings.*) – Extra settings, including filters and routing. For more information see https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html

> **actions = None**
> > The list of actions to perform. Populated by *curator.actions.Alias.add* and *curator.actions.Alias.remove*

> **add**(*ilo*, *warn_if_no_indices=False*)
> > Create *add* statements for each index in *ilo* for *alias*, then append them to *actions*. Add any *extras* that may be there.

> > > **Parameters ilo** – A *curator.indexlist.IndexList* object

> **body**()
> > Return a *body* string suitable for use with the *update_aliases* API call.

> **client = None**
> > Instance variable. The Elasticsearch Client object derived from *ilo*

> **do_action**()
> > Run the API call *update_aliases* with the results of *body()*

---

do_dry_run()
> Log what the output would be, but take no action.

extra_settings = None
> Instance variable. Any extra things to add to the alias, like filters, or routing.

name = None
> Instance variable The strftime parsed version of *name*.

remove(*ilo*, *warn_if_no_indices=False*)
> Create *remove* statements for each index in *ilo* for *alias*, then append them to *actions*.
>
> > **Parameters ilo** – A *curator.indexlist.IndexList* object

Allocation

class curator.actions.**Allocation**(*ilo*, *key=None*, *value=None*, *allocation_type='require'*, *wait_for_completion=False*, *timeout=30*)

> **Parameters**
>
> - **ilo** – A *curator.indexlist.IndexList* object
> - **key** – An arbitrary metadata attribute key. Must match the key assigned to at least some of your nodes to have any effect.
> - **value** – An arbitrary metadata attribute value. Must correspond to values associated with *key* assigned to at least some of your nodes to have any effect.
> - **allocation_type** – Type of allocation to apply. Default is *require*
> - **wait_for_completion** (*bool*) – Wait (or not) for the operation to complete before returning. (default: *False*)
> - **timeout** – Number of seconds to *wait_for_completion*

Note: See: https://www.elastic.co/guide/en/elasticsearch/reference/current/shard-allocation-filtering.html

---

bkey = None
> Instance variable. Populated at instance creation time. Value is index.routing.allocation. *allocation_type* . *key* . *value*

client = None
> Instance variable. The Elasticsearch Client object derived from *ilo*

do_action()
> Change allocation settings for indices in *index_list.indices* with the settings in *body*.

do_dry_run()
> Log what the output would be, but take no action.

index_list = None
> Instance variable. Internal reference to *ilo*

timeout = None
> Instance variable. How long in seconds to *wait_for_completion* before returning with an exception

wfc = None
> Instance variable. Internal reference to *wait_for_completion*

Close

class curator.actions.**Close**(*ilo*, *delete_aliases=False*)

> **Parameters**
>
> - **ilo** – A *curator.indexlist.IndexList* object
>
> - **delete_aliases** (*bool*) – If *True*, will delete any associated aliases before closing indices.

> **client = None**
> Instance variable. The Elasticsearch Client object derived from *ilo*

> **delete_aliases = None**
> Instance variable. Internal reference to *delete_aliases*

> **do_action**()
> Close open indices in *index_list.indices*

> **do_dry_run**()
> Log what the output would be, but take no action.

> **index_list = None**
> Instance variable. Internal reference to *ilo*

ClusterRouting

class curator.actions.**ClusterRouting**(*client*, *routing_type=None*, *setting=None*, *value=None*, *wait_for_completion=False*, *timeout=30*)

For now, the cluster routing settings are hardcoded to be transient

> **Parameters**
>
> - **client** – An *elasticsearch.Elasticsearch* client object
>
> - **routing_type** – Type of routing to apply. Either *allocation* or *rebalance*
>
> - **setting** – Currently, the only acceptable value for *setting* is enable. This is here in case that changes.
>
> - **value** – Used only if *setting* is *enable*. Semi-dependent on *routing_type*. Acceptable values for *allocation* and *rebalance* are all, primaries, and none (string, not *NoneType*). If *routing_type* is *allocation*, this can also be new_primaries, and if *rebalance*, it can be replicas.
>
> - **wait_for_completion** (*bool*) – Wait (or not) for the operation to complete before returning. (default: *False*)
>
> - **timeout** – Number of seconds to *wait_for_completion*

> **client = None**
> Instance variable. An *elasticsearch.Elasticsearch* client object

> **do_action**()
> Change cluster routing settings with the settings in *body*.

> **do_dry_run**()
> Log what the output would be, but take no action.

> **timeout = None**
> Instance variable. How long in seconds to *wait_for_completion* before returning with an exception

wfc = None
> Instance variable. Internal reference to *wait_for_completion*

DeleteIndices

class curator.actions.**DeleteIndices**(*ilo*, *master_timeout=30*)

> **Parameters**
>> • **ilo** – A *curator.indexlist.IndexList* object
>>
>> • **master_timeout** – Number of seconds to wait for master node response

> **client** = None
>> Instance variable. The Elasticsearch Client object derived from *ilo*

> **do_action**()
>> Delete indices in *index_list.indices*

> **do_dry_run**()
>> Log what the output would be, but take no action.

> **index_list** = None
>> Instance variable. Internal reference to *ilo*

> **master_timeout** = None
>> Instance variable. String value of *master_timeout* + 's', for seconds.

DeleteSnapshots

class curator.actions.**DeleteSnapshots**(*slo*, *retry_interval=120*, *retry_count=3*)

> **Parameters**
>> • **slo** – A *curator.snapshotlist.SnapshotList* object
>>
>> • **retry_interval** – Number of seconds to delay betwen retries. Default: 120 (seconds)
>>
>> • **retry_count** – Number of attempts to make. Default: 3

> **client** = None
>> Instance variable. The Elasticsearch Client object derived from *slo*

> **do_action**()
>> Delete snapshots in *slo* Retry up to *retry_count* times, pausing *retry_interval* seconds between retries.

> **do_dry_run**()
>> Log what the output would be, but take no action.

> **repository** = None
>> Instance variable. The repository name derived from *slo*

> **retry_count** = None
>> Instance variable. Internally accessible copy of *retry_count*

> **retry_interval** = None
>> Instance variable. Internally accessible copy of *retry_interval*

> **snapshot_list** = None
>> Instance variable. Internal reference to *slo*

ForceMerge

class curator.actions.**ForceMerge**(*ilo*, *max_num_segments=None*, *delay=0*)

> **Parameters**
>
> - **ilo** – A *curator.indexlist.IndexList* object
> - **max_num_segments** – Number of segments per shard to forceMerge
> - **delay** – Number of seconds to delay between forceMerge operations

client = None
> Instance variable. The Elasticsearch Client object derived from *ilo*

delay = None
> Instance variable. Internally accessible copy of *delay*

do_action()
> forcemerge indices in *index_list.indices*

do_dry_run()
> Log what the output would be, but take no action.

index_list = None
> Instance variable. Internal reference to *ilo*

max_num_segments = None
> Instance variable. Internally accessible copy of *max_num_segments*

Open

class curator.actions.**Open**(*ilo*)

> **Parameters ilo** – A *curator.indexlist.IndexList* object

client = None
> Instance variable. The Elasticsearch Client object derived from *ilo*

do_action()
> Open closed indices in *index_list.indices*

do_dry_run()
> Log what the output would be, but take no action.

index_list = None
> Instance variable. Internal reference to *ilo*

Replicas

class curator.actions.**Replicas**(*ilo*, *count=None*, *wait_for_completion=False*, *timeout=30*)

> **Parameters**
>
> - **ilo** – A *curator.indexlist.IndexList* object
> - **count** – The count of replicas per shard
> - **wait_for_completion** (*bool*) – Wait (or not) for the operation to complete before returning. (default: *False*)

client = None
> Instance variable. The Elasticsearch Client object derived from *ilo*

count = None
> Instance variable. Internally accessible copy of *count*

do_action()
> Update the replica count of indices in *index_list.indices*

do_dry_run()
> Log what the output would be, but take no action.

index_list = None
> Instance variable. Internal reference to *ilo*

timeout = None
> Instance variable. How long in seconds to *wait_for_completion* before returning with an exception

wfc = None
> Instance variable. Internal reference to *wait_for_completion*

Snapshot

class curator.actions.**Snapshot**(*ilo*, *repository=None*, *name=None*, *ignore_unavailable=False*, *include_global_state=True*, *partial=False*, *wait_for_completion=True*, *skip_repo_fs_check=False*)

> **Parameters**
>
> - **ilo** – A *curator.indexlist.IndexList* object
>
> - **repository** – The Elasticsearch snapshot repository to use
>
> - **name** – What to name the snapshot.
>
> - **wait_for_completion** (*bool*) – Wait (or not) for the operation to complete before returning. (default: *True*)
>
> - **ignore_unavailable** (*bool*) – Ignore unavailable shards/indices. (default: *False*)
>
> - **include_global_state** (*bool*) – Store cluster global state with snapshot. (default: *True*)
>
> - **partial** (*bool*) – Do not fail if primary shard is unavailable. (default: *False*)
>
> - **skip_repo_fs_check** (*bool*) – Do not validate write access to repository on all cluster nodes before proceeding. (default: *False*). Useful for shared filesystems where intermittent timeouts can affect validation, but won't likely affect snapshot success.

body = None
> Instance variable. Populated at instance creation time by calling *curator.utils.create_snapshot_body* with *ilo.indices* and the provided arguments: *ignore_unavailable*, *include_global_state*, *partial*

client = None
> Instance variable. The Elasticsearch Client object derived from *ilo*

do_action()
> Snapshot indices in *index_list.indices*, with options passed.

do_dry_run()
> Log what the output would be, but take no action.

get_state()
> Get the state of the snapshot

index_list = None
> Instance variable. Internal reference to *ilo*

name = None
> Instance variable. The parsed version of *name*

report_state()
> Log the state of the snapshot

repository = None
> Instance variable. Internally accessible copy of *repository*

skip_repo_fs_check = None
> Instance variable. Internally accessible copy of *skip_repo_fs_check*

wait_for_completion = None
> Instance variable. Internally accessible copy of *wait_for_completion*

Restore

class curator.actions.**Restore**(*slo*, *name=None*, *indices=None*, *include_aliases=False*, *ignore_unavailable=False*, *include_global_state=True*, *partial=False*, *rename_pattern=None*, *rename_replacement=None*, *extra_settings={}*, *wait_for_completion=True*, *skip_repo_fs_check=False*)

> **Parameters**
>
> * **slo** – A *curator.snapshotlist.SnapshotList* object
>
> * **name** (*str*) – Name of the snapshot to restore. If no name is provided, it will restore the most recent snapshot by age.
>
> * **indices** (*list*) – A list of indices to restore. If no indices are provided, it will restore all indices in the snapshot.
>
> * **include_aliases** (*bool*) – If set to *True*, restore aliases with the indices. (default: *False*)
>
> * **ignore_unavailable** (*bool*) – Ignore unavailable shards/indices. (default: *False*)
>
> * **include_global_state** (*bool*) – Store cluster global state with snapshot. (default: *True*)
>
> * **partial** (*bool*) – Do not fail if primary shard is unavailable. (default: *False*)
>
> * **rename_pattern** (*str*) – A regular expression pattern with one or more captures, e.g. `index_(.+)`
>
> * **rename_replacement** (*str*) – A target index name pattern with *$#* numbered references to the captures in `rename_pattern`, e.g. `restored_index_$1`
>
> * **extra_settings** (*dict, representing the settings.*) – Extra settings, including shard count and settings to omit. For more information see https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html#_changing_index_settings_during_restore
>
> * **wait_for_completion** (*bool*) – Wait (or not) for the operation to complete before returning. (default: *True*)

- **skip_repo_fs_check** (`bool`) – Do not validate write access to repository on all cluster nodes before proceeding. (default: *False*). Useful for shared filesystems where intermittent timeouts can affect validation, but won't likely affect snapshot success.

body = None
> Instance variable. Populated at instance creation time from the other options

client = None
> Instance variable. The Elasticsearch Client object derived from *slo*

do_action()
> Restore indices with options passed.

do_dry_run()
> Log what the output would be, but take no action.

name = None
> Instance variable. Will use a provided snapshot name, or the most recent snapshot in slo

py_rename_replacement = None
> Also an instance variable version of `rename_replacement` but with Java regex group designations of `$#` converted to Python's `\\#` style.

rename_pattern = None
> Instance variable version of `rename_pattern`

rename_replacement = None
> Instance variable version of `rename_replacement`

report_state()
> Log the state of the restore This should only be done if `wait_for_completion` is *True*, and only after completing the restore.

repository = None
> Instance variable. *repository* derived from *slo*

skip_repo_fs_check = None
> Instance variable. Internally accessible copy of *skip_repo_fs_check*

snapshot_list = None
> Instance variable. Internal reference to *slo*

Filter Methods

- *IndexList*

- *SnapshotList*

IndexList

IndexList.**filter_allocated**(*key=None*, *value=None*, *allocation_type='require'*, *exclude=True*)
> Match indices that have the routing allocation rule of *key=value* from *indices*

> **Parameters**

> - **key** – The allocation attribute to check for

> - **value** – The value to check for

> - **allocation_type** – Type of allocation to apply

- **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

`IndexList.filter_by_age`(*source='name'*, *direction=None*, *timestring=None*, *unit=None*, *unit_count=None*, *field=None*, *stats_result='min_value'*, *epoch=None*, *exclude=False*)

> Match *indices* by relative age calculations.

> **Parameters**

>> - **source** – Source of index age. Can be one of 'name', 'creation_date', or 'field_stats'

>> - **direction** – Time to filter, either `older` or `younger`

>> - **timestring** – An strftime string to match the datestamp in an index name. Only used for index filtering by `name`.

>> - **unit** – One of `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`.

>> - **unit_count** – The number of `unit``s. ``unit_count * unit` will be calculated out to the relative number of seconds.

>> - **field** – A timestamp field name. Only used for `field_stats` based calculations.

>> - **stats_result** – Either *min_value* or *max_value*. Only used in conjunction with *source`=``field_stats`* to choose whether to reference the minimum or maximum result value.

>> - **epoch** – An epoch timestamp used in conjunction with `unit` and `unit_count` to establish a point of reference for calculations. If not provided, the current time will be used.

>> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

`IndexList.filter_by_regex`(*kind=None*, *value=None*, *exclude=False*)

> Match indices by regular expression (pattern).

> **Parameters**

>> - **kind** – Can be one of: `suffix`, `prefix`, `regex`, or `timestring`. This option defines what kind of filter you will be building.

>> - **value** – Depends on *kind*. It is the strftime string if *kind* is `timestring`. It's used to build the regular expression for other kinds.

>> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

`IndexList.filter_by_space`(*disk_space=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *field=None*, *stats_result='min_value'*, *exclude=False*)

> Remove indices from the actionable list based on space consumed, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.

> The default is usually what you will want. If only one kind of index is provided–for example, indices matching `logstash-%Y.%m.%d`–then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older indices.

> By setting *reverse* to *False*, then `index3` will be deleted before `index2`, which will be deleted before `index1`

> *use_age* allows ordering indices by age. Age is determined by the index creation date by default, but you can specify an *source* of `name`, `max_value`, or `min_value`. The `name` *source* requires the timestring argument.

> **Parameters**

- **disk_space** – Filter indices over *n* gigabytes

- **reverse** – The filtering direction. (default: *True*). Ignored if *use_age* is *True*

- **use_age** – Sort indices by age. `source` is required in this case.

- **source** – Source of index age. Can be one of `name`, `creation_date`, or `field_stats`. Default: `creation_date`

- **timestring** – An strftime string to match the datestamp in an index name. Only used if *source* `name` is selected.

- **field** – A timestamp field name. Only used if *source* `field_stats` is selected.

- **stats_result** – Either *min_value* or *max_value*. Only used if *source* `field_stats` is selected. It determines whether to reference the minimum or maximum value of *field* in each index.

- **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

IndexList.**filter_closed**(*exclude=True*)
　　Filter out closed indices from *indices*

　　　　Parameters exclude – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

IndexList.**filter_forceMerged**(*max_num_segments=None*, *exclude=True*)
　　Match any index which has *max_num_segments* per shard or fewer in the actionable list.

　　　　Parameters

- **max_num_segments** – Cutoff number of segments per shard.

- **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

IndexList.**filter_kibana**(*exclude=True*)
　　Match any index named `.kibana`, `kibana-int`, `.marvel-kibana`, or `.marvel-es-data` in *indices*.

　　　　Parameters exclude – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

IndexList.**filter_opened**(*exclude=True*)
　　Filter out opened indices from *indices*

　　　　Parameters exclude – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

IndexList.**filter_none**()

IndexList.**filter_by_alias**(*aliases=None*, *exclude=False*)
　　Match indices which are associated with the alias identified by *name*

　　　　Parameters

- **aliases** (*[list](#)*) – A list of alias names.

- **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *False*

IndexList.**filter_by_count**(*count=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *field=None*, *stats_result='min_value'*, *exclude=True*)
　　Remove indices from the actionable list beyond the number *count*, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.

The default is usually what you will want. If only one kind of index is provided–for example, indices matching `logstash-%Y.%m.%d`–then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older indices.

By setting *reverse* to *False*, then `index3` will be deleted before `index2`, which will be deleted before `index1`

use_age allows ordering indices by age. Age is determined by the index creation date by default, but you can specify an *source* of `name`, `max_value`, or `min_value`. The `name` *source* requires the timestring argument.

> **Parameters**
>
> - **count** – Filter indices beyond *count*.
>
> - **reverse** – The filtering direction. (default: *True*).
>
> - **use_age** – Sort indices by age. `source` is required in this case.
>
> - **source** – Source of index age. Can be one of `name`, `creation_date`, or `field_stats`. Default: `creation_date`
>
> - **timestring** – An strftime string to match the datestamp in an index name. Only used if *source* `name` is selected.
>
> - **field** – A timestamp field name. Only used if *source* `field_stats` is selected.
>
> - **stats_result** – Either *min_value* or *max_value*. Only used if *source* `field_stats` is selected. It determines whether to reference the minimum or maximum value of *field* in each index.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching indices from *indices*. If *exclude* is *False*, then only matching indices will be kept in *indices*. Default is *True*

SnapshotList

SnapshotList.**filter_by_age**(*source='creation_date'*, *direction=None*, *timestring=None*, *unit=None*, *unit_count=None*, *epoch=None*, *exclude=False*)
    Remove snapshots from *snapshots* by relative age calculations.

> **Parameters**
>
> - **source** – Source of snapshot age. Can be 'name', or 'creation_date'.
>
> - **direction** – Time to filter, either `older` or `younger`
>
> - **timestring** – An strftime string to match the datestamp in an snapshot name. Only used for snapshot filtering by `name`.
>
> - **unit** – One of `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`.
>
> - **unit_count** – The number of unit``s. ``unit_count * unit will be calculated out to the relative number of seconds.
>
> - **epoch** – An epoch timestamp used in conjunction with `unit` and `unit_count` to establish a point of reference for calculations. If not provided, the current time will be used.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*

SnapshotList.**filter_by_regex**(*kind=None*, *value=None*, *exclude=False*)
    Filter out snapshots not matching the pattern, or in the case of exclude, filter those matching the pattern.

> **Parameters**

- **kind** – Can be one of: `suffix`, `prefix`, `regex`, or `timestring`. This option defines what kind of filter you will be building.

- **value** – Depends on *kind*. It is the strftime string if *kind* is *timestring*. It's used to build the regular expression for other kinds.

- **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*

SnapshotList.**filter_by_state**(*state=None*, *exclude=False*)

Filter out snapshots not matching `state`, or in the case of exclude, filter those matching `state`.

> **Parameters**
>
> - **state** – The snapshot state to filter for. Must be one of `SUCCESS`, `PARTIAL`, `FAILED`, or `IN_PROGRESS`.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *False*

SnapshotList.**filter_none**()

SnapshotList.**filter_by_count**(*count=None*, *reverse=True*, *use_age=False*, *source='creation_date'*, *timestring=None*, *exclude=True*)

Remove snapshots from the actionable list beyond the number *count*, sorted reverse-alphabetically by default. If you set *reverse* to *False*, it will be sorted alphabetically.

The default is usually what you will want. If only one kind of snapshot is provided–for example, snapshots matching `curator-%Y%m%d%H%M%S`– then reverse alphabetical sorting will mean the oldest will remain in the list, because lower numbers in the dates mean older snapshots.

By setting *reverse* to *False*, then `snapshot3` will be acted on before `snapshot2`, which will be acted on before `snapshot1`

use_age allows ordering snapshots by age. Age is determined by the snapshot creation date (as identified by `start_time_in_millis`) by default, but you can also specify a *source* of `name`. The `name` *source* requires the timestring argument.

> **Parameters**
>
> - **count** – Filter snapshots beyond *count*.
>
> - **reverse** – The filtering direction. (default: *True*).
>
> - **use_age** – Sort snapshots by age. `source` is required in this case.
>
> - **source** – Source of snapshot age. Can be one of `name`, or `creation_date`. Default: `creation_date`
>
> - **timestring** – An strftime string to match the datestamp in a snapshot name. Only used if *source* `name` is selected.
>
> - **exclude** – If *exclude* is *True*, this filter will remove matching snapshots from *snapshots*. If *exclude* is *False*, then only matching snapshots will be kept in *snapshots*. Default is *True*

Utility & Helper Methods

class curator.utils.**TimestringSearch**(*timestring*)

An object to allow repetitive search against a string, *searchme*, without having to repeatedly recreate the regex.

> **Parameters timestring** – An strftime pattern

get_epoch(*searchme*)

Return the epoch timestamp extracted from the *timestring* appearing in *searchme*.

> **Parameters** **searchme** – A string to be searched for a date pattern that matches *timestring*
>
> **Return type** int

curator.utils.**byte_size**(*num*, *suffix='B'*)

Return a formatted string indicating the size in bytes, with the proper unit, e.g. KB, MB, GB, TB, etc.

> **Parameters**
>
> - **num** – The number of byte
>
> - **suffix** – An arbitrary suffix, like *Bytes*
>
> **Return type** float

curator.utils.**check_csv**(*value*)

Some of the curator methods should not operate against multiple indices at once. This method can be used to check if a list or csv has been sent.

> **Parameters** **value** – The value to test, if list or csv string
>
> **Return type** bool

curator.utils.**check_master**(*client*, *master_only=False*)

Check if connected client is the elected master node of the cluster. If not, cleanly exit with a log message.

> **Parameters** **client** – An elasticsearch.Elasticsearch client object
>
> **Return type** None

curator.utils.**check_version**(*client*)

Verify version is within acceptable range. Raise an exception if it is not.

> **Parameters** **client** – An elasticsearch.Elasticsearch client object
>
> **Return type** None

curator.utils.**chunk_index_list**(*indices*)

This utility chunks very large index lists into 3KB chunks It measures the size as a csv string, then converts back into a list for the return value.

> **Parameters** **indices** – A list of indices to act on.
>
> **Return type** list

curator.utils.**create_repo_body**(*repo_type=None*, *compress=True*, *chunk_size=None*, *max_restore_bytes_per_sec=None*, *max_snapshot_bytes_per_sec=None*, *location=None*, *bucket=None*, *region=None*, *base_path=None*, *access_key=None*, *secret_key=None*, ***kwargs*)

Build the 'body' portion for use in creating a repository.

> **Parameters**
>
> - **repo_type** – The type of repository (presently only *fs* and *s3*)
>
> - **compress** – Turn on compression of the snapshot files. Compression is applied only to metadata files (index mapping and settings). Data files are not compressed. (Default: *True*)
>
> - **chunk_size** – The chunk size can be specified in bytes or by using size value notation, i.e. 1g, 10m, 5k. Defaults to *null* (unlimited chunk size).
>
> - **max_restore_bytes_per_sec** – Throttles per node restore rate. Defaults to `20mb` per second.

- **max_snapshot_bytes_per_sec** – Throttles per node snapshot rate. Defaults to `20mb` per second.

- **location** – Location of the snapshots. Required.

- **bucket** – *S3 only.* The name of the bucket to be used for snapshots. Required.

- **region** – *S3 only.* The region where bucket is located. Defaults to *US Standard*

- **base_path** – *S3 only.* Specifies the path within bucket to repository data. Defaults to value of `repositories.s3.base_path` or to root directory if not set.

- **access_key** – *S3 only.* The access key to use for authentication. Defaults to value of `cloud.aws.access_key`.

- **secret_key** – *S3 only.* The secret key to use for authentication. Defaults to value of `cloud.aws.secret_key`.

  Returns A dictionary suitable for creating a repository from the provided arguments.

  Return type [dict]

curator.utils.**create_repository**(*client*, ***kwargs*)

Create repository with repository and body settings

  Parameters

- **client** – An `elasticsearch.Elasticsearch` client object

- **repository** – The Elasticsearch snapshot repository to use

- **repo_type** – The type of repository (presently only *fs* and *s3*)

- **compress** – Turn on compression of the snapshot files. Compression is applied only to metadata files (index mapping and settings). Data files are not compressed. (Default: *True*)

- **chunk_size** – The chunk size can be specified in bytes or by using size value notation, i.e. 1g, 10m, 5k. Defaults to *null* (unlimited chunk size).

- **max_restore_bytes_per_sec** – Throttles per node restore rate. Defaults to `20mb` per second.

- **max_snapshot_bytes_per_sec** – Throttles per node snapshot rate. Defaults to `20mb` per second.

- **location** – Location of the snapshots. Required.

- **bucket** – *S3 only.* The name of the bucket to be used for snapshots. Required.

- **region** – *S3 only.* The region where bucket is located. Defaults to *US Standard*

- **base_path** – *S3 only.* Specifies the path within bucket to repository data. Defaults to value of `repositories.s3.base_path` or to root directory if not set.

- **access_key** – *S3 only.* The access key to use for authentication. Defaults to value of `cloud.aws.access_key`.

- **secret_key** – *S3 only.* The secret key to use for authentication. Defaults to value of `cloud.aws.secret_key`.

  Returns A boolean value indicating success or failure.

  Return type [bool]

curator.utils.**create_snapshot_body**(*indices*, *ignore_unavailable=False*, *include_global_state=True*, *partial=False*)

Create the request body for creating a snapshot from the provided arguments.

> **Parameters**
>
> - **indices** – A single index, or list of indices to snapshot.
>
> - **ignore_unavailable** (*bool*) – Ignore unavailable shards/indices. (default: *False*)
>
> - **include_global_state** (*bool*) – Store cluster global state with snapshot. (default: *True*)
>
> - **partial** (*bool*) – Do not fail if primary shard is unavailable. (default: *False*)
>
> **Return type** dict

curator.utils.**ensure_list**(*indices*)

> Return a list, even if indices is a single value
>
> **Parameters indices** – A list of indices to act upon
>
> **Return type** list

curator.utils.**fix_epoch**(*epoch*)

> Fix value of *epoch* to be epoch, which should be 10 or fewer digits long.
>
> **Parameters epoch** – An epoch timestamp, in epoch + milliseconds, or microsecond, or even nanoseconds.
>
> **Return type** int

curator.utils.**get_client**(***kwargs*)

> NOTE: AWS IAM parameters *aws_key*, *aws_secret_key*, and *aws_region* are provided for future compatibility, should AWS ES support the /_cluster/state/metadata endpoint. So long as this endpoint does not function in AWS ES, the client will not be able to use *curator.indexlist.IndexList*, which is the backbone of Curator 4
>
> Return an elasticsearch.Elasticsearch client object using the provided parameters. Any of the keyword arguments the elasticsearch.Elasticsearch client object can receive are valid, such as:
>
> **Parameters**
>
> - **hosts** (*list*) – A list of one or more Elasticsearch client hostnames or IP addresses to connect to. Can send a single host.
>
> - **port** (*int*) – The Elasticsearch client port to connect to.
>
> - **url_prefix** (*str*) – *Optional* url prefix, if needed to reach the Elasticsearch API (i.e., it's not at the root level)
>
> - **use_ssl** (*bool*) – Whether to connect to the client via SSL/TLS
>
> - **certificate** – Path to SSL/TLS certificate
>
> - **client_cert** – Path to SSL/TLS client certificate (public key)
>
> - **client_key** – Path to SSL/TLS private key
>
> - **aws_key** – AWS IAM Access Key (Only used if the requests-aws4auth python module is installed)
>
> - **aws_secret_key** – AWS IAM Secret Access Key (Only used if the requests-aws4auth python module is installed)
>
> - **aws_region** – AWS Region (Only used if the requests-aws4auth python module is installed)
>
> - **ssl_no_validate** (*bool*) – If *True*, do not validate the certificate chain. This is an insecure option and you will see warnings in the log output.

- **http_auth** (`str`) – Authentication credentials in *user:pass* format.

- **timeout** (`int`) – Number of seconds before the client will timeout.

- **master_only** (`bool`) – If *True*, the client will *only* connect if the endpoint is the elected master node of the cluster. **This option does not work if 'hosts' has more than one value.** It will raise an Exception in that case.

  Return type `elasticsearch.Elasticsearch`

curator.utils.**get_date_regex**(*timestring*)
    Return a regex string based on a provided strftime timestring.

        Parameters timestring – An strftime pattern

        Return type str

curator.utils.**get_datetime**(*index_timestamp*, *timestring*)
    Return the datetime extracted from the index name, which is the index creation time.

        Parameters

- **index_timestamp** – The timestamp extracted from an index name

- **timestring** – An strftime pattern

        Return type `datetime.datetime`

curator.utils.**get_indices**(*client*)
    Get the current list of indices from the cluster.

        Parameters client – An `elasticsearch.Elasticsearch` client object

        Return type list

curator.utils.**get_point_of_reference**(*unit*, *count*, *epoch=None*)
    Get a point-of-reference timestamp in epoch + milliseconds by deriving from a *unit* and a *count*, and an optional reference timestamp, *epoch*

        Parameters

- **unit** – One of `seconds`, `minutes`, `hours`, `days`, `weeks`, `months`, or `years`.

- **unit_count** – The number of `units`. `unit_count * unit` will be calculated out to the relative number of seconds.

- **epoch** – An epoch timestamp used in conjunction with `unit` and `unit_count` to establish a point of reference for calculations.

        Return type int

curator.utils.**get_repository**(*client*, *repository=''*)
    Return configuration information for the indicated repository.

        Parameters

- **client** – An `elasticsearch.Elasticsearch` client object

- **repository** – The Elasticsearch snapshot repository to use

        Return type dict

curator.utils.**get_snapshot**(*client*, *repository=None*, *snapshot=''*)
    Return information about a snapshot (or a comma-separated list of snapshots) If no snapshot specified, it will return all snapshots. If none exist, an empty dictionary will be returned.

        Parameters

- **client** – An `elasticsearch.Elasticsearch` client object
- **repository** – The Elasticsearch snapshot repository to use
- **snapshot** – The snapshot name, or a comma-separated list of snapshots

> **Return type** dict

curator.utils.**get_snapshot_data**(*client*, *repository=None*)
> Get `_all` snapshots from repository and return a list.

> **Parameters**

- **client** – An `elasticsearch.Elasticsearch` client object
- **repository** – The Elasticsearch snapshot repository to use

> **Return type** list

curator.utils.**get_version**(*client*)
> Return the ES version number as a tuple. Omits trailing tags like -dev, or Beta

> **Parameters client** – An `elasticsearch.Elasticsearch` client object

> **Return type** tuple

curator.utils.**get_yaml**(*path*)
> Read the file identified by *path* and import its YAML contents.

> **Parameters path** – The path to a YAML configuration file.

> **Return type** dict

curator.utils.**is_master_node**(*client*)
> Return *True* if the connected client node is the elected master node in the Elasticsearch cluster, otherwise return *False*.

> **Parameters client** – An `elasticsearch.Elasticsearch` client object

> **Return type** bool

curator.utils.**override_timeout**(*timeout*, *action*)
> Override the default timeout for *forcemerge*, *snapshot*, and *sync_flush* operations if the default value of `30` is provided.

> **Parameters**

- **timeout** – Number of seconds before the client will timeout.
- **action** – The *action* to be performed.

curator.utils.**parse_date_pattern**(*name*)
> Scan and parse *name* for `time.strftime()` strings, replacing them with the associated value when found, but otherwise returning lowercase values, as uppercase snapshot names are not allowed.

> The `time.strftime()` identifiers that Curator currently recognizes as acceptable include:

> - Y: A 4 digit year
> - y: A 2 digit year
> - m: The 2 digit month
> - W: The 2 digit week of the year
> - d: The 2 digit day of the month
> - H: The 2 digit hour of the day, in 24 hour notation

- `M`: The 2 digit minute of the hour

- `S`: The 2 digit number of second of the minute

- `j`: The 3 digit day of the year

> **Parameters** `name` – A name, which can contain `time.strftime()` strings

curator.utils.**prune_nones**(*mydict*)
> Remove keys from *mydict* whose values are *None*

> **Parameters** `mydict` – The dictionary to act on

> **Return type** dict

curator.utils.**read_file**(*myfile*)
> Read a file and return the resulting data.

> **Parameters** `myfile` – A file to read.

> **Return type** str

curator.utils.**report_failure**(*exception*)
> Raise a *FailedExecution* exception and include the original error message.

> **Parameters** `exception` – The upstream exception.

> **Return type** None

curator.utils.**repository_exists**(*client*, *repository=None*)
> Verify the existence of a repository

> **Parameters**

>> - `client` – An `elasticsearch.Elasticsearch` client object

>> - `repository` – The Elasticsearch snapshot repository to use

> **Return type** bool

curator.utils.**safe_to_snap**(*client*, *repository=None*, *retry_interval=120*, *retry_count=3*)
> Ensure there are no snapshots in progress. Pause and retry accordingly

> **Parameters**

>> - `client` – An `elasticsearch.Elasticsearch` client object

>> - `repository` – The Elasticsearch snapshot repository to use

>> - `retry_interval` – Number of seconds to delay betwen retries. Default: 120 (seconds)

>> - `retry_count` – Number of attempts to make. Default: 3

> **Return type** bool

curator.utils.**show_dry_run**(*ilo*, *action*, ***kwargs*)
> Log dry run output with the action which would have been executed.

> **Parameters**

>> - `ilo` – A *curator.indexlist.IndexList*

>> - `action` – The *action* to be performed.

>> - `kwargs` – Any other args to show in the log output

curator.utils.**snapshot_in_progress**(*client*, *repository=None*, *snapshot=None*)
>   Determine whether the provided snapshot in *repository* is IN_PROGRESS. If no value is provided for *snapshot*, then check all of them. Return *snapshot* if it is found to be in progress, or *False*

>   **Parameters**

>   >   • **client** – An `elasticsearch.Elasticsearch` client object

>   >   • **repository** – The Elasticsearch snapshot repository to use

>   >   • **snapshot** – The snapshot name

curator.utils.**snapshot_running**(*client*)
>   Return *True* if a snapshot is in progress, and *False* if not

>   **Parameters client** – An `elasticsearch.Elasticsearch` client object

>   **Return type** bool

curator.utils.**test_client_options**(*config*)
>   Test whether a SSL/TLS files exist. Will raise an exception if the files cannot be read.

>   **Parameters config** – A client configuration file data dictionary

>   **Return type** None

curator.utils.**test_repo_fs**(*client*, *repository=None*)
>   Test whether all nodes have write access to the repository

>   **Parameters**

>   >   • **client** – An `elasticsearch.Elasticsearch` client object

>   >   • **repository** – The Elasticsearch snapshot repository to use

curator.utils.**to_csv**(*indices*)
>   Return a csv string from a list of indices, or a single value if only one value is present

>   **Parameters indices** – A list of indices to act on, or a single value, which could be in the format of a csv string already.

>   **Return type** str

curator.utils.**validate_actions**(*data*)
>   Validate an Action configuration dictionary, as imported from actions.yml, for example.

>   The method returns a validated and sanitized configuration dictionary.

>   **Parameters data** – The configuration dictionary

>   **Return type** dict

curator.utils.**validate_filters**(*action*, *filters*)
>   Validate that the filters are appropriate for the action type, e.g. no index filters applied to a snapshot list.

>   **Parameters**

>   >   • **action** – An action name

>   >   • **filters** – A list of filters to test.

curator.utils.**verify_client_object**(*test*)
>   Test if *test* is a proper `elasticsearch.Elasticsearch` client object and raise an exception if it is not.

>   **Parameters test** – The variable or object to test

>   **Return type** None

4.4. Utility & Helper Methods

curator.utils.**verify_index_list**(*test*)

Test if *test* is a proper *curator.indexlist.IndexList* object and raise an exception if it is not.

> **Parameters test** – The variable or object to test
>
> **Return type** None

curator.utils.**verify_snapshot_list**(*test*)

Test if *test* is a proper *curator.snapshotlist.SnapshotList* object and raise an exception if it is not.

> **Parameters test** – The variable or object to test
>
> **Return type** None

class curator.**SchemaCheck**(*config*, *schema*, *test_what*, *location*)

Validate `config` with the provided voluptuous `schema`. `test_what` and `location` are for reporting the results, in case of failure. If validation is successful, the method returns `config` as valid.

> **Parameters**
>
> - **config** (*dict*) – A configuration dictionary.
> - **schema** (`voluptuous.Schema`) – A voluptuous schema definition
> - **test_what** (*str*) – which configuration block is being validated
> - **location** (*str*) – An string to report which configuration sub-block is being tested.

Examples

Each of these examples presupposes that the requisite modules have been imported and an instance of the Elasticsearch client object has been created:

```python
import elasticsearch
import curator

client = elasticsearch.Elasticsearch()
```

Filter indices by prefix

```python
ilo = curator.IndexList(client)
ilo.filter_by_regex(kind='prefix', value='logstash-')
```

The contents of *ilo.indices* would then only be indices matching the *prefix*.

Filter indices by suffix

```python
ilo = curator.IndexList(client)
ilo.filter_by_regex(kind='suffix', value='-prod')
```

The contents of *ilo.indices* would then only be indices matching the *suffix*.

Filter indices by age (name)

This example will match indices with the following criteria:

- Have a date string of `%Y.%m.%d`

- Use *days* as the unit of time measurement

- Filter indices *older* than 5 *days*

```
ilo = curator.IndexList(client)
ilo.filter_by_age(source='name', direction='older', timestring='%Y.%m.%d',
    unit='days', unit_count=5
)
```

The contents of *ilo.indices* would then only be indices matching these criteria.

Filter indices by age (creation_date)

This example will match indices with the following criteria:

- Use *months* as the unit of time measurement

- Filter indices where the index creation date is *older* than 2 *months* from this moment.

```
ilo = curator.IndexList(client)
ilo.filter_by_age(source='creation_date', direction='older',
    unit='months', unit_count=2
)
```

The contents of *ilo.indices* would then only be indices matching these criteria.

Filter indices by age (field_stats)

This example will match indices with the following criteria:

- Use *days* as the unit of time measurement

- Filter indices where the *timestamp* field's *min_value* is a date *older* than 3 *weeks* from this moment.

```
ilo = curator.IndexList(client)
ilo.filter_by_age(source='field_stats', direction='older',
    unit='weeks', unit_count=3, field='timestamp', stats_result='min_value'
)
```

The contents of *ilo.indices* would then only be indices matching these criteria.

Changelog

4.3.0 (10 April 2017)

New Features

- Added `warn_if_no_indices` option for `alias` action in response to #883. Using this option will permit the `alias` add or remove to continue with a logged warning, even if the filters result in a NoIndices condition. Use with care.

Bug Fixes

- Check if an index is in an alias before attempting to delete it from the alias. Issue raised in #887. (untergeek)
- Fix allocation issues when using Elasticsearch 5.1+. Issue raised in #871 (untergeek)

Documentation

- Add missing repository arg to auto-gen API docs. Reported in #888 (untergeek)

4.2.6 (27 January 2016)

General

- Change max supported version of Elasticsearch so Curator will work with elasticsearch 5.2
- Use `isinstance()` to verify client object. #862 (cp2587)
- Update `certifi` dependency to latest version

Documentation

- Add version compatibility section to official documentation.

4.2.5 (22 December 2016)

General

- Add and increment test versions for Travis CI. #839 (untergeek)
- Make *filter_list* optional in snapshot, show_snapshot and show_indices singleton actions. #853 (alexef)

Bug Fixes

- Fix cli integration test when different host/port are specified. Reported in #843 (untergeek)
- Catch empty list condition during filter iteration in singleton actions. Reported in #848 (untergeek)

Documentation

- Add docs regarding how filters are ANDed together, and how to do an OR with the regex pattern filter type. Requested in #842 (untergeek)
- Fix typo in Click version in docs. #850 (breml)
- Where applicable, replace *[source,text]* with *[source,yaml]* for better formatting in the resulting docs.

4.2.4 (7 December 2016)

Bug Fixes

- `--wait_for_completion` should be *True* by default for Snapshot singleton action. Reported in #829 (untergeek)
- Increase *version_max* to 5.1.99. Prematurely reported in #832 (untergeek)
- Make the '.security' index visible for snapshots so long as proper credentials are used. Reported in #826 (untergeek)

4.2.3.post1 (22 November 2016)

This fix is *only* going in for `pip`-based installs. There are no other code changes.

Bug Fixes

- Fixed incorrect assumption of PyPI picking up dependency for certifi. It is still a dependency, but should not affect `pip` installs with an error any more. Reported in #821 (untergeek)

4.2.3 (21 November 2016)

4.2.2 was pulled immediately after release after it was discovered that the Windows binary distributions were still not including the certifi-provided certificates. This has now been remedied.

General

- `certifi` is now officially a requirement.

- `setup.py` now forcibly includes the `certifi` certificate PEM file in the "frozen" distributions (i.e., the compiled versions). The `get_client` method was updated to reflect this and catch it for both the Linux and Windows binary distributions. This should *finally* put to rest #810

4.2.2 (21 November 2016)

Bug Fixes

- The certifi-provided certificates were not propagating to the compiled RPM/DEB packages. This has been corrected. Reported in #810 (untergeek)

General

- Added missing `--ignore_empty_list` option to singleton actions. Requested in #812 (untergeek)

Documentation

- Add a FAQ entry regarding the click module's need for Unicode when using Python 3. Kind of a bug fix too, as the entry_points were altered to catch this omission and report a potential solution on the command-line. Reported in #814 (untergeek)

- Change the "Command-Line" documentation header to be "Running Curator"

4.2.1 (8 November 2016)

Bug Fixes

- In the course of package release testing, an undesirable scenario was caught where boolean flags default values for `curator_cli` were improperly overriding values from a yaml config file.

General

- Adding in direct download URLs for the RPM, DEB, tarball and zip packages.

4.2.0 (4 November 2016)

New Features

- Shard routing allocation enable/disable. This will allow you to disable shard allocation routing before performing one or more actions, and then re-enable after it is complete. Requested in #446 (untergeek)

- Curator 3.x-style command-line. This is now `curator_cli`, to differentiate between the current binary. Not all actions are available, but the most commonly used ones are. With the addition in 4.1.0 of schema and configuration validation, there's even a way to still do filter chaining on the command-line! Requested in #767, and by many other users (untergeek)

General

- Update testing to the most recent versions.

- Lock elasticsearch-py module version at >= 2.4.0 and <= 3.0.0. There are API changes in the 5.0 release that cause tests to fail.

Bug Fixes

- Guarantee that binary packages are built from the latest Python + libraries. This ensures that SSL/TLS will work without warning messages about insecure connections, unless they actually are insecure. Reported in #780, though the reported problem isn't what was fixed. The fix is needed based on what was discovered while troubleshooting the problem. (untergeek)

4.1.2 (6 October 2016)

This release does not actually add any new code to Curator, but instead improves documentation and includes new linux binary packages.

General

- New Curator binary packages for common Linux systems! These will be found in the same repositories that the python-based packages are in, but have no dependencies. All necessary libraries/modules are bundled with the binary, so everything should work out of the box. This feature doesn't change any other behavior, so it's not a major release.

  These binaries have been tested in:

  - CentOS 6 & 7

  - Ubuntu 12.04, 14.04, 16.04

  - Debian 8

  They do not work in Debian 7 (library mismatch). They may work in other systems, but that is untested.

  The script used is in the unix_packages directory. The Vagrantfiles for the various build systems are in the Vagrant directory.

Bug Fixes

- The only bug that can be called a bug is actually a stray `.exe` suffix in the binary package creation section (cx_freeze) of `setup.py`. The Windows binaries should have `.exe` extensions, but not unix variants.

- Elasticsearch 5.0.0-beta1 testing revealed that a document ID is required during document creation in tests. This has been fixed, and a redundant bit of code in the forcemerge integration test was removed.

Documentation

- The documentation has been updated and improved. Examples and installation are now top-level events, with the sub-sections each having their own link. They also now show how to install and use the binary packages, and the section on installation from source has been improved. The missing section on installing the voluptuous schema verification module has been written and included. #776 (untergeek)

4.1.1 (27 September 2016)

Bug Fixes

- String-based booleans are now properly coerced. This fixes an issue where *True*/*False* were used in environment variables, but not recognized. #765 (untergeek)
- Fix missing *count* method in `__map_method` in SnapshotList. Reported in #766 (untergeek)

General

- Update es_repo_mgr to use the same client/logging YAML config file. Requested in #752 (untergeek)

Schema Validation

- Cases where `source` was not defined in a filter (but should have been) were informing users that a *timestring* field was there that shouldn't have been. This edge case has been corrected.

Documentation

- Added notifications and FAQ entry to explain that AWS ES is not supported.

4.1.0 (6 September 2016)

New Features

- Configuration and Action file schema validation. Requested in #674 (untergeek)
- Alias filtertype! With this filter, you can select indices based on whether they are part of an alias. Merged in #748 (untergeek)
- Count filtertype! With this filter, you can now configure Curator to only keep the most recent _n_ indices (or snapshots!). Merged in #749 (untergeek)
- Experimental! Use environment variables in your YAML configuration files. This was a popular request, #697. (untergeek)

General

- New requirement! `voluptuous` Python schema validation module
- Requirement version bump: Now requires `elasticsearch-py` 2.4.0

Bug Fixes

- `delete_aliases` option in `close` action no longer results in an error if not all selected indices have an alias. Add test to confirm expected behavior. Reported in #736 (untergeek)

Documentation

- Add information to FAQ regarding indices created before Elasticsearch 1.4. Merged in #747

4.0.6 (15 August 2016)

Bug Fixes

- Update old calls used with ES 1.x to reflect changes in 2.x+. This was necessary to work with Elasticsearch 5.0.0-alpha5. Fixed in #728 (untergeek)

Doc Fixes

- Add section detailing that the value of a `value` filter element should be encapsulated in single quotes. Reported in #726. (untergeek)

4.0.5 (3 August 2016)

Bug Fixes

- Fix incorrect variable name for AWS Region reported in #679 (basex)
- Fix `filter_by_space()` to not fail when index age metadata is not present. Indices without the appropriate age metadata will instead be excluded, with a debug-level message. Reported in #724 (untergeek)

Doc Fixes

- Fix documentation for the space filter and the source filter element.

4.0.4 (1 August 2016)

Bug Fixes

- Fix incorrect variable name in Allocation action. #706 (lukewaite)
- Incorrect error message in `create_snapshot_body` reported in #711 (untergeek)
- Test for empty index list object should happen in action initialization for snapshot action. Discovered in #711. (untergeek)

Doc Fixes

- Add menus to asciidoc chapters #704 (untergeek)
- Add pyyaml dependency #710 (dtrv)

4.0.3 (22 July 2016)

General

- 4.0.2 didn't work for `pip` installs due to an omission in the MANIFEST.in file. This came up during release testing, but before the release was fully published. As the release was never fully published, this should not have actually affected anyone.

Bug Fixes

- These are the same as 4.0.2, but it was never fully released.
- All default settings are now values returned from functions instead of constants. This was resulting in settings getting stomped on. New test addresses the original complaint. This removes the need for `deepcopy`. See issue #687 (untergeek)
- Fix `host` vs. `hosts` issue in `get_client()` rather than the non-functional function in `repomgrcli.py`.
- Update versions being tested.
- Community contributed doc fixes.
- Reduced logging verbosity by making most messages debug level. #684 (untergeek)
- Fixed log whitelist behavior (and switched to blacklisting instead). Default behavior will now filter traffic from the `elasticsearch` and `urllib3` modules.
- Fix Travis CI testing to accept some skipped tests, as needed. #695 (untergeek)
- Fix missing empty index test in snapshot action. #682 (sherzberg)

4.0.2 (22 July 2016)

Bug Fixes

- All default settings are now values returned from functions instead of constants. This was resulting in settings getting stomped on. New test addresses the original complaint. This removes the need for `deepcopy`. See issue #687 (untergeek)

- Fix `host` vs. `hosts` issue in `get_client()` rather than the non-functional function in `repomgrcli.py`.

- Update versions being tested.

- Community contributed doc fixes.

- Reduced logging verbosity by making most messages debug level. #684 (untergeek)

- Fixed log whitelist behavior (and switched to blacklisting instead). Default behavior will now filter traffic from the `elasticsearch` and `urllib3` modules.

- Fix Travis CI testing to accept some skipped tests, as needed. #695 (untergeek)

- Fix missing empty index test in snapshot action. #682 (sherzberg)

4.0.1 (1 July 2016)

Bug Fixes

- Coerce Logstash/JSON logformat type timestamp value to always use UTC. #661 (untergeek)

- Catch and remove indices from the actionable list if they do not have a *creation_date* field in settings. This field was introduced in ES v1.4, so that indicates a rather old index. #663 (untergeek)

- Replace missing `state` filter for `snapshotlist`. #665 (untergeek)

- Restore `es_repo_mgr` as a stopgap until other CLI scripts are added. It will remain undocumented for now, as I am debating whether to make repository creation its own action in the API. #668 (untergeek)

- Fix dry run results for snapshot action. #673 (untergeek)

4.0.0 (24 June 2016)

It's official! Curator 4.0.0 is released!

Breaking Changes

- New and improved API!

- Command-line changes. No more command-line args, except for `--config`, `--actions`, and `--dry-run`:
  - `--config` points to a YAML client and logging configuration file. The default location is `~/.curator/curator.yml`
  - `--actions` arg points to a YAML action configuration file
  - `--dry-run` will simulate the action(s) which would have taken place, but not actually make any changes to the cluster or its indices.

New Features

- Snapshot restore is here!

- YAML configuration files. Now a single file can define an entire batch of commands, each with their own filters, to be performed in sequence.

- Sort by index age not only by index name (as with previous versions of Curator), but also by index *creation_date*, or by calculations from the Field Stats API on a timestamp field.

- Atomically add/remove indices from aliases! This is possible by way of the new *IndexList* class and YAML configuration files.

- State of indices pulled and stored in *IndexList* instance. Fewer API calls required to serially test for open/close, *size_in_bytes*, etc.

- Filter by space now allows sorting by age!

- Experimental! Use AWS IAM credentials to sign requests to Elasticsearch. This requires the end user to *manually* install the *requests_aws4auth* python module.

- Optionally delete aliases from indices before closing.

- An empty index or snapshot list no longer results in an error if you set `ignore_empty_list` to *True*. If *True* it will still log that the action was not performed, but will continue to the next action. If 'False' it will log an ERROR and exit with code 1.

API

- Updated API documentation

- Class: *IndexList*. This pulls all indices at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- Class: *SnapshotList*. This pulls all snapshots from the given repository at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- Add *wait_for_completion* to Allocation and Replicas actions. These will use the client timeout, as set by default or *timeout_override*, to determine how long to wait for timeout. These are handled in batches of indices for now.

- Allow *timeout_override* option for all actions. This allows for different timeout values per action.

- Improve API by giving each action its own *do_dry_run()* method.

General

- Updated use documentation for Elastic main site.

- Include example files for `--config` and `--actions`.

4.0.0b2 (16 June 2016)

Second beta release of the 4.0 branch

New Feature

- An empty index or snapshot list no longer results in an error if you set `ignore_empty_list` to *True*. If *True* it will still log that the action was not performed, but will continue to the next action. If 'False' it will log an ERROR and exit with code 1. (untergeek)

4.0.0b1 (13 June 2016)

First beta release of the 4.0 branch!

The release notes will be rehashing the new features in 4.0, rather than the bug fixes done during the alphas.

Breaking Changes

- New and improved API!

- Command-line changes. No more command-line args, except for `--config`, `--actions`, and `--dry-run`:

  - `--config` points to a YAML client and logging configuration file. The default location is `~/.curator/curator.yml`

  - `--actions` arg points to a YAML action configuration file

  - `--dry-run` will simulate the action(s) which would have taken place, but not actually make any changes to the cluster or its indices.

New Features

- Snapshot restore is here!

- YAML configuration files. Now a single file can define an entire batch of commands, each with their own filters, to be performed in sequence.

- Sort by index age not only by index name (as with previous versions of Curator), but also by index *creation_date*, or by calculations from the Field Stats API on a timestamp field.

- Atomically add/remove indices from aliases! This is possible by way of the new *IndexList* class and YAML configuration files.

- State of indices pulled and stored in *IndexList* instance. Fewer API calls required to serially test for open/close, *size_in_bytes*, etc.

- Filter by space now allows sorting by age!

- Experimental! Use AWS IAM credentials to sign requests to Elasticsearch. This requires the end user to *manually* install the *requests_aws4auth* python module.

- Optionally delete aliases from indices before closing.

API

- Updated API documentation

- Class: *IndexList*. This pulls all indices at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- Class: *SnapshotList*. This pulls all snapshots from the given repository at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- Add *wait_for_completion* to Allocation and Replicas actions. These will use the client timeout, as set by default or *timeout_override*, to determine how long to wait for timeout. These are handled in batches of indices for now.

- Allow *timeout_override* option for all actions. This allows for different timeout values per action.

- Improve API by giving each action its own *do_dry_run()* method.

General

- Updated use documentation for Elastic main site.

- Include example files for `--config` and `--actions`.

4.0.0a10 (10 June 2016)

New Features

- Snapshot restore is here!

- Optionally delete aliases from indices before closing. Fixes #644 (untergeek)

General

- Add *wait_for_completion* to Allocation and Replicas actions. These will use the client timeout, as set by default or *timeout_override*, to determine how long to wait for timeout. These are handled in batches of indices for now.

- Allow *timeout_override* option for all actions. This allows for different timeout values per action.

Bug Fixes

- Disallow use of *master_only* if multiple hosts are used. Fixes #615 (untergeek)

- Fix an issue where arguments weren't being properly passed and populated.

- ForceMerge replaced Optimize in ES 2.1.0.

- Fix prune_nones to work with Python 2.6. Fixes #619 (untergeek)

- Fix TimestringSearch to work with Python 2.6. Fixes #622 (untergeek)

- Add language classifiers to `setup.py`. Fixes #640 (untergeek)

- Changed references to readthedocs.org to be readthedocs.io.

4.0.0a9 (27 Apr 2016)

General

- Changed *create_index* API to use kwarg *extra_settings* instead of *body*

- Normalized Alias action to use *name* instead of *alias*. This simplifies documentation by reducing the number of option elements.

- Streamlined some code

- Made *exclude* a filter element setting for all filters. Updated all examples to show this.

- Improved documentation

New Features

- Alias action can now accept *extra_settings* to allow adding filters, and/or routing.

4.0.0a8 (26 Apr 2016)

Bug Fixes

- Fix to use *optimize* with versions of Elasticsearch < 5.0

- Fix missing setting in testvars

4.0.0a7 (25 Apr 2016)

Bug Fixes

- Fix AWS4Auth error.

4.0.0a6 (25 Apr 2016)

General

- Documentation updates.

- Improve API by giving each action its own *do_dry_run()* method.

Bug Fixes

- Do not escape characters other than `.` and `-` in timestrings. Fixes #602 (untergeek)

** New Features**

- Added *CreateIndex* action.

4.0.0a4 (21 Apr 2016)

Bug Fixes

- Require *pyyaml* 3.10 or better.

- In the case that no *options* are in an action, apply the defaults.

4.0.0a3 (21 Apr 2016)

It's time for Curator 4.0 alpha!

Breaking Changes

- New API! (again?!)

- Command-line changes. No more command-line args, except for `--config`, `--actions`, and `--dry-run`:

    - `--config` points to a YAML client and logging configuration file. The default location is `~/.curator/curator.yml`

    - `--actions` arg points to a YAML action configuration file

    - `--dry-run` will simulate the action(s) which would have taken place, but not actually make any changes to the cluster or its indices.

General

- Updated API documentation

- Updated use documentation for Elastic main site.

- Include example files for `--config` and `--actions`.

New Features

- Sort by index age not only by index name (as with previous versions of Curator), but also by index *creation_date*, or by calculations from the Field Stats API on a timestamp field.

- Class: *IndexList*. This pulls all indices at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- Class: *SnapshotList*. This pulls all snapshots from the given repository at instantiation, and you apply filters, which are class methods. You can iterate over as many filters as you like, in fact, due to the YAML config file.

- YAML configuration files. Now a single file can define an entire batch of commands, each with their own filters, to be performed in sequence.

- Atomically add/remove indices from aliases! This is possible by way of the new *IndexList* class and YAML configuration files.

- State of indices pulled and stored in *IndexList* instance. Fewer API calls required to serially test for open/close, *size_in_bytes*, etc.

- Filter by space now allows sorting by age!

- Experimental! Use AWS IAM credentials to sign requests to Elasticsearch. This requires the end user to *manually* install the *requests_aws4auth* python module.

3.5.1 (21 March 2016)

Bug fixes

- Add more logging information to snapshot delete method #582 (untergeek)
- Improve default timeout, logging, and exception handling for *seal* command #583 (untergeek)
- Fix use of default snapshot name. #584 (untergeek)

3.5.0 (16 March 2016)

General

- Add support for the *–client-cert* and *–client-key* command line parameters and client_cert and client_key parameters to the get_client() call. #520 (richm)

Bug fixes

- Disallow users from creating snapshots with upper-case letters, which is not permitted by Elasticsearch. #562 (untergeek)
- Remove *print()* command from `setup.py` as it causes issues with command- line retrieval of `--url`, etc. #568 (thib-ack)
- Remove unnecessary argument from *build_filter()* #530 (zzugg)
- Allow day of year filter to be made up with 1, 2 or 3 digits #578 (petitout)

3.4.1 (10 February 2016)

General

- Update license copyright to 2016
- Use slim python version with Docker #527 (xaka)
- Changed `--master-only` exit code to 0 when connected to non-master node #540 (wkruse)
- Add `cx_Freeze` capability to `setup.py`, plus a `binary_release.py` script to simplify binary package creation. #554 (untergeek)
- Set Elastic as author. #555 (untergeek)
- Put repository creation methods into API and document them. Requested in #550 (untergeek)

Bug fixes

- Fix sphinx documentation build error #506 (hydrapolic)
- Ensure snapshots are found before iterating #507 (garyelephant)
- Fix a doc inconsistency #509 (pmoust)
- Fix a typo in *show* documentation #513 (pbamba)
- Default to trying the cluster state for checking whether indices are closed, and then fall back to using the _cat API (for Amazon ES instances). #519 (untergeek)

- Improve logging to show time delay between optimize runs, if selected. #525 (untergeek)

- Allow elasticsearch-py module versions through 2.3.0 (a presumption at this point) #524 (untergeek)

- Improve logging in snapshot api method to reveal when a repository appears to be missing. Reported in #551 (untergeek)

- Test that `--timestring` has the correct variable for `--time-unit`. Reported in #544 (untergeek)

- Allocation will exit with exit_code 0 now when there are no indices to work on. Reported in #531 (untergeek)

3.4.0 (28 October 2015)

General

- API change in elasticsearch-py 1.7.0 prevented alias operations. Fixed in #486 (HonzaKral)

- During index selection you can now select only closed indices with `--closed-only`. Does not impact `--all-indices` Reported in #476. Fixed in #487 (Basster)

- API Changes in Elasticsearch 2.0.0 required some refactoring. All tests pass for ES versions 1.0.3 through 2.0.0-rc1. Fixed in #488 (untergeek)

- es_repo_mgr now has access to the same SSL options from #462. #489 (untergeek)

- Logging improvements requested in #475. (untergeek)

- Added `--quiet` flag. #494 (untergeek)

- Fixed `index_closed` to work with AWS Elasticsearch. #499 (univerio)

- Acceptable versions of Elasticsearch-py module are 1.8.0 up to 2.1.0 (untergeek)

3.3.0 (31 August 2015)

Announcement

- Curator is tested in Jenkins. Each commit to the master branch is tested with both Python versions 2.7.6 and 3.4.0 against each of the following Elasticsearch versions: * 1.7_nightly * 1.6_nightly * 1.7.0 * 1.6.1 * 1.5.1 * 1.4.4 * 1.3.9 * 1.2.4 * 1.1.2 * 1.0.3

- If you are using a version different from this, your results may vary.

General

- Allocation type can now also be `include` or `exclude`, in addition to the the existing default `require` type. Add `--type` to the allocation command to specify the type. #443 (steffo)

- Bump elasticsearch python module dependency to 1.6.0+ to enable synced_flush API call. Reported in #447 (untergeek)

- Add SSL features, `--ssl-no-validate` and `certificate` to provide other ways to validate SSL connections to Elasticsearch. #436 (untergeek)

Bug fixes

- Delete by space was only reporting space used by primary shards. Fixed to show all space consumed. Reported in #455 (untergeek)

- Update exit codes and messages for snapshot selection. Reported in #452 (untergeek)

- Fix potential int/float casting issues. Reported in #465 (untergeek)

3.2.3 (16 July 2015)

Bug fix

- In order to address customer and community issues with bulk deletes, the `master_timeout` is now invoked for delete operations. This should address 503s with 30s timeouts in the debug log, even when `--timeout` is set to a much higher value. The `master_timeout` is tied to the `--timeout` flag value, but will not exceed 300 seconds. #420 (untergeek)

General

- Mixing it up a bit here by putting *General* second! The only other changes are that logging has been improved for deletes so you won't need to have the `--debug` flag to see if you have error codes >= 400, and some code documentation improvements.

3.2.2 (13 July 2015)

General

- This is a very minor change. The `mock` library recently removed support for Python 2.6. As many Curator users are using RHEL/CentOS 6, which is pinned to Python 2.6, this requires the mock version referenced by Curator to also be pinned to a supported version (`mock==1.0.1`).

3.2.1 (10 July 2015)

General

- Added delete verification & retry (fixed at 3x) to potentially cover an edge case in #420 (untergeek)
- Since GitHub allows rST (reStructuredText) README documents, and that's what PyPI wants also, the README has been rebuilt in rST. (untergeek)

Bug fixes

- If closing indices with ES 1.6+, and all indices are closed, ensure that the seal command does not try to seal all indices. Reported in #426 (untergeek)
- Capture AttributeError when sealing indices if a non-TransportError occurs. Reported in #429 (untergeek)

3.2.0 (25 June 2015)

New!

- Added support to manually seal, or perform a [synced flush](http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-synced-flush.html) on indices with the `seal` command. #394 (untergeek)
- Added *experimental* support for SSL certificate validation. In order for this to work, you must install the `certifi` python module: `pip install certifi` This feature *should* automatically work if the `certifi` module is installed. Please report any issues.

General

- Changed logging to go to stdout rather than stderr. Reopened #121 and figured they were right. This is better. (untergeek)
- Exit code 99 was unpopular. It has been removed. Reported in #371 and #391 (untergeek)

- Add `--skip-repo-validation` flag for snapshots. Do not validate write access to repository on all cluster nodes before proceeding. Useful for shared filesystems where intermittent timeouts can affect validation, but won't likely affect snapshot success. Requested in #396 (untergeek)

- An alias no longer needs to be pre-existent in order to use the alias command. #317 (untergeek)

- es_repo_mgr now passes through upstream errors in the event a repository fails to be created. Requested in #405 (untergeek)

Bug fixes

- In rare cases, `*` wildcard would not expand. Replaced with _all. Reported in #399 (untergeek)

- Beginning with Elasticsearch 1.6, closed indices cannot have their replica count altered. Attempting to do so results in this error: `org.elasticsearch.ElasticsearchIllegalArgumentException:` `Can't update [index.number_of_replicas] on closed indices [[test_index]] -` `can leave index in an unopenable state` As a result, the `change_replicas` method has been updated to prune closed indices. This change will apply to all versions of Elasticsearch. Reported in #400 (untergeek)

- Fixed es_repo_mgr repository creation verification error. Reported in #389 (untergeek)

3.1.0 (21 May 2015)

General

- If `wait_for_completion` is true, snapshot success is now tested and logged. Reported in #253 (untergeek)

- Log & return false if a snapshot is already in progress (untergeek)

- Logs individual deletes per index, even though they happen in batch mode. Also log individual snapshot deletions. Reported in #372 (untergeek)

- Moved `chunk_index_list` from cli to api utils as it's now also used by `filter.py`

- Added a warning and 10 second timer countdown if you use `--timestring` to filter indices, but do not use `--older-than` or `--newer-than` in conjunction with it. This is to address #348, which behavior isn't a bug, but prevents accidental action against all of your time-series indices. The warning and timer are not displayed for `show` and `--dry-run` operations.

- Added tests for `es_repo_mgr` in #350

- Doc fixes

Bug fixes

- delete-by-space needed the same fix used for #245. Fixed in #353 (untergeek)

- Increase default client timeout for `es_repo_mgr` as node discovery and availability checks for S3 repositories can take a bit. Fixed in #352 (untergeek)

- If an index is closed, indicate in `show` and `--dry-run` output. Reported in #327. (untergeek)

- Fix issue where CLI parameters were not being passed to the `es_repo_mgr` create sub-command. Reported in #337. (feltnerm)

3.0.3 (27 Mar 2015)

Announcement

This is a bug fix release. #319 and #320 are affecting a few users, so this release is being expedited.

Test count: 228 Code coverage: 99%

General

- Documentation for the CLI converted to Asciidoc and moved to http://www.elastic.co/guide/en/elasticsearch/client/curator/current/index.html

- Improved logging, and refactored a few methods to help with this.

- Dry-run output is now more like v2, with the index or snapshot in the log line, along with the command. Several tests needed refactoring with this change, along with a bit of documentation.

Bug fixes

- Fix links to repository in setup.py. Reported in #318 (untergeek)

- No more `--delay` with optimized indices. Reported in #319 (untergeek)

- `--request_timeout` not working as expected. Reinstate the version 2 timeout override feature to prevent default timeouts for `optimize` and `snapshot` operations. Reported in #320 (untergeek)

- Reduce index count to 200 for test.integration.test_cli_commands.TestCLISnapshot.test_cli_snapshot_huge_list in order to reduce or eliminate Jenkins CI test timeouts. Reported in #324 (untergeek)

- `--dry-run` no longer calls `show`, but will show output in the log, as in v2. This was a recurring complaint. See #328 (untergeek)

3.0.2 (23 Mar 2015)

Announcement

This is a bug fix release. #307 and #309 were big enough to warrant an expedited release.

Bug fixes

- Purge unneeded constants, and clean up config options for snapshot. Reported in #303 (untergeek)

- Don't split large index list if performing snapshots. Reported in #307 (untergeek)

- Act correctly if a zero value for *–older-than* or *–newer-than* is provided. #309 (untergeek)

3.0.1 (16 Mar 2015)

Announcement

The `regex_iterate` method was horribly named. It has been renamed to `apply_filter`. Methods have been added to allow API users to build a filtered list of indices similarly to how the CLI does. This was an oversight. Props to @SegFaultAX for pointing this out.

General

- In conjunction with the rebrand to Elastic, URLs and documentation were updated.

- Renamed horribly named *regex_iterate* method to *apply_filter* #298 (untergeek)

- Added *build_filter* method to mimic CLI calls. #298 (untergeek)

- Added Examples page in the API documentation. #298 (untergeek)

Bug fixes

- Refactored to show *–dry-run* info for *–disk-space* calls. Reported in #290 (untergeek)

- Added list chunking so acting on huge lists of indices won't result in a URL bigger than 4096 bytes (Elasticsearch's default limit.) Reported in https://github.com/elastic/curator/issues/245#issuecomment-77916081

- Refactored *to_csv()* method to be simpler.

- Added and removed tests according to changes. Code coverage still at 99%

3.0.0 (9 March 2015)

Release Notes

The full release of Curator 3.0 is out! Check out all of the changes here!

Note: This release is _not_ reverse compatible with any previous version.

Because 3.0 is a major point release, there have been some major changes to both the API as well as the CLI arguments and structure.

Be sure to read the updated command-line specific docs in the [wiki](https://github.com/elasticsearch/curator/wiki) and change your command-line arguments accordingly.

The API docs are still at http://curator.readthedocs.io. Be sure to read the latest docs, or select the docs for 3.0.0.

General

- **Breaking changes to the API.** Because this is a major point revision, changes to the API have been made which are non-reverse compatible. Before upgrading, be sure to update your scripts and test them thoroughly.

- **Python 3 support** Somewhere along the line, Curator would no longer work with curator. All tests now pass for both Python2 and Python3, with 99% code coverage in both environments.

- **New CLI library.** Using Click now. http://click.pocoo.org/3/ This change is especially important as it allows very easy CLI integration testing.

- **Pipelined filtering!** You can now use `--older-than` & `--newer-than` in the same command! You can also provide your own regex via the `--regex` parameter. You can use multiple instances of the `--exclude` flag.

- **Manually include indices!** With the `--index` paramter, you can add an index to the working list. You can provide multiple instances of the `--index` parameter as well!

- **Tests!** So many tests now. Test coverage of the API methods is at 100% now, and at 99% for the CLI methods. This doesn't mean that all of the tests are perfect, or that I haven't missed some scenarios. It does mean, however, that it will be much easier to write tests if something turns up missed. It also means that any new functionality will now need to have tests.

- **Iteration changes** Methods now only iterate through each index when appropriate! In fact, the only commands that iterate are *alias* and *optimize*. The *bloom* command will iterate, but only if you have added the *–delay* flag with a value greater than zero.

- **Improved packaging!** Methods have been moved into categories of `api` and `cli`, and further broken out into individual modules to help them be easier to find and read.

- Check for allocation before potentially re-applying an allocation rule. #273 (ferki)

- Assigning replica count and routing allocation rules _can_ be done to closed indices. #283 (ferki)

Bug fixes

- Don't accidentally delete `.kibana` index. #261 (malagoli)

- Fix segment count for empty indices. #265 (untergeek)

- Change bloom filter cutoff Elasticsearch version to 1.4. Reported in #267 (untergeek)

3.0.0rc1 (5 March 2015)

Release Notes

RC1 is here! I'm re-releasing the Changes from all betas here, minus the intra-beta code fixes. Barring any show stoppers, the official release will be soon.

General

- **Breaking changes to the API.** Because this is a major point revision, changes to the API have been made which are non-reverse compatible. Before upgrading, be sure to update your scripts and test them thoroughly.

- **Python 3 support** Somewhere along the line, Curator would no longer work with curator. All tests now pass for both Python2 and Python3, with 99% code coverage in both environments.

- **New CLI library.** Using Click now. http://click.pocoo.org/3/ This change is especially important as it allows very easy CLI integration testing.

- **Pipelined filtering!** You can now use `--older-than` & `--newer-than` in the same command! You can also provide your own regex via the `--regex` parameter. You can use multiple instances of the `--exclude` flag.

- **Manually include indices!** With the `--index` paramter, you can add an index to the working list. You can provide multiple instances of the `--index` parameter as well!

- **Tests!** So many tests now. Test coverage of the API methods is at 100% now, and at 99% for the CLI methods. This doesn't mean that all of the tests are perfect, or that I haven't missed some scenarios. It does mean, however, that it will be much easier to write tests if something turns up missed. It also means that any new functionality will now need to have tests.

- Methods now only iterate through each index when appropriate!

- Improved packaging! Hopefully the `entry_point` issues some users have had will be addressed by this. Methods have been moved into categories of `api` and `cli`, and further broken out into individual modules to help them be easier to find and read.

- Check for allocation before potentially re-applying an allocation rule. #273 (ferki)

- Assigning replica count and routing allocation rules _can_ be done to closed indices. #283 (ferki)

Bug fixes

- Don't accidentally delete `.kibana` index. #261 (malagoli)

- Fix segment count for empty indices. #265 (untergeek)

- Change bloom filter cutoff Elasticsearch version to 1.4. Reported in #267 (untergeek)

3.0.0b4 (5 March 2015)

Notes

Integration testing! Because I finally figured out how to use the Click Testing API, I now have a good collection of command-line simulations, complete with a real back-end. This testing found a few bugs (this is why testing exists, right?), and fixed a few of them.

Bug fixes

- HUGE! *curator show snapshots* would _delete_ snapshots. This is fixed.

- Return values are now being sent from the commands.

- *scripttest* is no longer necessary (click.Test works!)

- Calling *get_snapshot* without a snapshot name returns all snapshots

3.0.0b3 (4 March 2015)

Bug fixes

- setup.py was lacking the new packages "curator.api" and "curator.cli" The package works now.

- Python3 suggested I had to normalize the beta tag to just b3, so that's also changed.

- Cleaned out superfluous imports and logger references from the __init__.py files.

3.0.0-beta2 (3 March 2015)

Bug fixes

- Python3 issues resolved. Tests now pass on both Python2 and Python3

3.0.0-beta1 (3 March 2015)

General

- **Breaking changes to the API.** Because this is a major point revision, changes to the API have been made which are non-reverse compatible. Before upgrading, be sure to update your scripts and test them thoroughly.

- **New CLI library.** Using Click now. http://click.pocoo.org/3/

- **Pipelined filtering!** You can now use `--older-than` & `--newer-than` in the same command! You can also provide your own regex via the `--regex` parameter. You can use multiple instances of the `--exclude` flag.

- **Manually include indices!** With the `--index` paramter, you can add an index to the working list. You can provide multiple instances of the `--index` parameter as well!

- **Tests!** So many tests now. Unit test coverage of the API methods is at 100% now. This doesn't mean that all of the tests are perfect, or that I haven't missed some scenarios. It does mean that any new functionality will need to also have tests, now.

- Methods now only iterate through each index when appropriate!

- Improved packaging! Hopefully the `entry_point` issues some users have had will be addressed by this. Methods have been moved into categories of `api` and `cli`, and further broken out into individual modules to help them be easier to find and read.

- Check for allocation before potentially re-applying an allocation rule. #273 (ferki)

Bug fixes

- Don't accidentally delete `.kibana` index. #261 (malagoli)

- Fix segment count for empty indices. #265 (untergeek)

- Change bloom filter cutoff Elasticsearch version to 1.4. Reported in #267 (untergeek)

2.1.2 (22 January 2015)

Bug fixes

- Do not try to set replica count if count matches provided argument. #247 (bobrik)
- Fix JSON logging (Logstash format). #250 (magnusbaeck)
- Fix bug in *filter_by_space()* which would match all indices if the provided patterns found no matches. Reported in #254 (untergeek)

2.1.1 (30 December 2014)

Bug fixes

- Renamed unnecessarily redundant `--replicas` to `--count` in args for `curator_script.py`

2.1.0 (30 December 2014)

General

- Snapshot name now appears in log output or STDOUT. #178 (untergeek)
- Replicas! You can now change the replica count of indices. Requested in #175 (untergeek)
- Delay option added to Bloom Filter functionality. #206 (untergeek)
- Add 2-digit years as acceptable pattern (y vs. Y). Reported in #209 (untergeek)
- Add Docker container definition #226 (christianvozar)
- Allow the use of 0 with –older-than, –most-recent and –delete-older-than. See #208. #211 (bobrik)

Bug fixes

- Edge case where 1.4.0.Beta1-SNAPSHOT would break version check. Reported in #183 (untergeek)
- Typo fixed. #193 (ferki)
- Type fixed. #204 (gheppner)
- Shows proper error in the event of concurrent snapshots. #177 (untergeek)
- Fixes erroneous index display of `_`, `a`, `l`, `l` when –all-indices selected. Reported in #222 (untergeek)
- Use json.dumps() to escape exceptions. Reported in #210 (untergeek)
- Check if index is closed before adding to alias. Reported in #214 (bt5e)
- No longer force-install argparse if pre-installed #216 (whyscream)
- Bloom filters have been removed from Elasticsearch 1.5.0. Update methods and tests to act accordingly. #233 (untergeek)

2.0.2 (8 October 2014)

Bug fixes

- Snapshot name not displayed in log or STDOUT #185 (untergeek)
- Variable name collision in delete_snapshot() #186 (untergeek)

2.0.1 (1 October 2014)

Bug fix

- Override default timeout when snapshotting –all-indices #179 (untergeek)

2.0.0 (25 September 2014)

General

- New! Separation of Elasticsearch Curator Python API and curator_script.py (untergeek)
- New! `--delay` after optimize to allow cluster to quiesce #131 (untergeek)
- New! `--suffix` option in addition to `--prefix` #136 (untergeek)
- New! Support for wildcards in prefix & suffix #136 (untergeek)
- Complete refactor of snapshots. Now supporting incrementals! (untergeek)

Bug fix

- Incorrect error msg if no indices sent to create_snapshot (untergeek)
- Correct for API change coming in ES 1.4 #168 (untergeek)
- Missing `"` in Logstash log format #143 (cassianoleal)
- Change non-master node test to exit code 0, log as `INFO`. #145 (untergeek)
- *months* option missing from validate_timestring() (untergeek)

1.2.2 (29 July 2014)

Bug fix

- Updated `README.md` to briefly explain what curator does #117 (untergeek)
- Fixed `es_repo_mgr` logging whitelist #119 (untergeek)
- Fixed absent `months` time-unit #120 (untergeek)
- Filter out `.marvel-kibana` when prefix is `.marvel-` #120 (untergeek)
- Clean up arg parsing code where redundancy exists #123 (untergeek)
- Properly divide debug from non-debug logging #125 (untergeek)
- Fixed `show` command bug caused by changes to command structure #126 (michaelweiser)

1.2.1 (24 July 2014)

Bug fix

- Fixed the new logging when called by `curator` entrypoint.

1.2.0 (24 July 2014)

General

- New! Allow user-specified date patterns: `--timestring` #111 (untergeek)
- New! Curate weekly indices (must use week of year) #111 (untergeek)
- New! Log output in logstash format `--logformat logstash` #111 (untergeek)
- Updated! Cleaner default logs (debug still shows everything) (untergeek)
- Improved! Dry runs are more visible in log output (untergeek)

Errata

- The `--separator` option was removed in lieu of user-specified date patterns.
- Default `--timestring` for days: `%Y.%m.%d` (Same as before)
- Default `--timestring` for hours: `%Y.%m.%d.%H` (Same as before)
- Default `--timestring` for weeks: `%Y.%W`

1.1.3 (18 July 2014)

Bug fix

- Prefix not passed in `get_object_list()` #106 (untergeek)
- Use `os.devnull` instead of `/dev/null` for Windows #102 (untergeek)
- The http auth feature was erroneously omitted #100 (bbuchacher)

1.1.2 (13 June 2014)

Bug fix

- This was a showstopper bug for anyone using RHEL/CentOS with a Python 2.6 dependency for yum
- Python 2.6 does not like format calls without an index. #96 via #95 (untergeek)
- We won't talk about what happened to 1.1.1. No really. I hate git today :(

1.1.0 (12 June 2014)

General

- Updated! New command structure
- New! Snapshot to fs or s3 #82 (untergeek)
- New! Add/Remove indices to alias #82 via #86 (cschellenger)
- New! `--exclude-pattern` #80 (ekamil)
- New! (sort of) Restored `--log-level` support #73 (xavier-calland)
- New! show command-line options #82 via #68 (untergeek)
- New! Shard Allocation Routing #82 via #62 (nickethier)

Bug fix

- Fix `--max_num_segments` not being passed correctly #74 (untergeek)
- Change `BUILD_NUMBER` to `CURATOR_BUILD_NUMBER` in `setup.py` #60 (mohabusama)
- Fix off-by-one error in time calculations #66 (untergeek)
- Fix testing with python3 #92 (untergeek)

Errata

- Removed `optparse` compatibility. Now requires `argparse`.

1.0.0 (25 Mar 2014)

General

- compatible with `elasticsearch-py` 1.0 and Elasticsearch 1.0 (honzakral)
- Lots of tests! (honzakral)
- Streamline code for 1.0 ES versions (honzakral)

Bug fix

- Fix `find_expired_indices()` to not skip closed indices (honzakral)

0.6.2 (18 Feb 2014)

General

- Documentation fixes #38 (dharrigan)
- Add support for HTTPS URI scheme and `optparse` compatibility for Python 2.6 (gelim)
- Add elasticsearch module version checking for future compatibility checks (untergeek)

0.6.1 (08 Feb 2014)

General

- Added tarball versioning to `setup.py` (untergeek)

Bug fix

- Fix `long_description` by including `README.md` in `MANIFEST.in` (untergeek)
- Incorrect version number in `curator.py` (untergeek)

0.6.0 (08 Feb 2014)

General

- Restructured repository to a be a proper python package. (arieb)
- Added `setup.py` file. (arieb)
- Removed the deprecated file `logstash_index_cleaner.py` (arieb)
- Updated `README.md` to fit the new package, most importantly the usage and installation. (arieb)
- Fixes and package push to PyPI (untergeek)

0.5.2 (26 Jan 2014)

General

- Fix boolean logic determining hours or days for time selection (untergeek)

0.5.1 (20 Jan 2014)

General

- Fix `can_bloom` to compare numbers (HonzaKral)

- Switched `find_expired_indices()` to use `datetime` and `timedelta`

- Do not try and catch unrecoverable exceptions. (HonzaKral)

- Future proofing the use of the elasticsearch client (i.e. work with version 1.0+ of Elasticsearch) (HonzaKral) Needs more testing, but should work.

- Add tests for these scenarios (HonzaKral)

0.5.0 (17 Jan 2014)

General

- Deprecated `logstash_index_cleaner.py` Use new `curator.py` instead (untergeek)

- new script change: `curator.py` (untergeek)

- new add index optimization (Lucene forceMerge) to reduce segments and therefore memory usage. (untergeek)

- update refactor of args and several functions to streamline operation and make it more readable (untergeek)

- update refactor further to clean up and allow immediate (and future) portability (HonzaKral)

0.4.0

General

- First version logged in `CHANGELOG`

- new `--disable-bloom-days` feature requires 0.90.9+

  http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/index-modules-codec.html#bloom-postings

  This can save a lot of heap space on cold indexes (i.e. not actively indexing documents)

CHAPTER 5

License

Copyright (c) 2012–2016 Elasticsearch <http://www.elastic.co>

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

> http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Indices and tables

- genindex
- search

Python Module Index

C

Index

T

V

W