Elasticsearch Curator Documentation

Release 0.10.10

Aaron Mildenstein

Oct 30, 2018

Contents

CHAPTER 1

Contents

1.1 Installation

The initial installation requires Python 2.7.9 or better, though Python 3.5+ are preferred.

1.1.1 Using setup.py

Clone the GitHub repository or download the source.

Ensure you have pip and virtualenv installed before proceeding.

```
virtualenv /my/chosen/path
```

```
source /my/chosen/path/bin/activate
```

Go to the directory where you cloned the GitHub repository, or expanded the source archive and run the following:

```
pip install -r requirements.txt
```

```
pip install .
```

This will install the necessary python entry points to /my/chosen/path/bin/

1.2 Configuration

In order to prevent a high number of individual calls to the Elasticsearch API, es_stats_zabbix uses a caching client backend. The TTL on the cache can be tuned, with a default of 60 seconds.

1.2.1 Configuring es_stats_zabbix

Backend Configuration

The config.yml file has multiple blocks, or subsections. Each is described in the following.

elasticsearch

This is an example configuration block:

```yaml
elasticsearch:
  master_only: false
  skip_version_test: false
  client:
    hosts:
      - 127.0.0.1
    port: 9200
    url_prefix:
    timeout: 30
    username:
    password:
    verify_certs: false
    use_ssl: false
    ca_certs:
    client_cert:
    client_key:
    ssl_version:
    ssl_assert_hostname:
    http_compress: true
  aws:
    sign_requests: false
    aws_region: us_east
```

master_only

master_only should be true or false. This setting specifies that Elasticsearch is permitted to establish a client connection *only* with the elected master node in the cluster.

The default value is false.

Tip: This option only works if there is one, and only one host provided for hosts.

skip_version_test

skip_version_test may permit use with an unsupported version of Elasticsearch. The value should be true or false.

The default value is false.

client

client is a sub-section that contains further options for establishing a client connection to Elasticsearch. The options are as follows:

hosts

hosts is an array of potential hosts to connect to. If multiple are specified, requests will be performed in a round robin approach across all *available* hosts. Any acceptable YAML syntax for lists/arrays will work.

The default value is 127.0.0.1

url_prefix

Set this if you have Elasticsearch behind a proxy or some other end point, and have configured an additional endpoint before the API portion of the URL.

For example, if you have to type https://www.example.com/myprefix/_cat to reach the _cat API, then your url_prefix would be myprefix.

There is no default value, or the default value is empty.

timeout

How long a client connection should wait (in seconds) for a response before timing out.

The default value is *30*, meaning 30 seconds.

username

If Elasticsearch requires a username and password to authenticate, specify the username here.

There is no default value, or the default value is empty.

password

If Elasticsearch requires a username and password to authenticate, specify the password here.

There is no default value, or the default value is empty.

verify_certs

The value of this should be true or false, and determines whether SSL certificates will be verified.

The default value is false, though you should create your certificates in a way that authenticity can be verified.

use_ssl

The value of this should be `true` or `false`, and determines whether SSL/TLS will be used to connect to Elasticsearch.

The default value is `false`

ca_certs

The value of this can be a single value, or an array of values. Each value must be a filesystem path to a certificate authority file.

There is no default value, but if not specified, and `use_ssl` is `true`, it will try to use the default certificates provided by `certifi` for verification purposes.

client_cert

This setting should point to a filesystem path to a client public key (or certificate) file to be used for two-way SSL authentication/verification.

There is no default value, or the default value is empty.

client_key

This setting should point to a filesystem path to a client private key file to be used for two-way SSL authentication/verification.

There is no default value, or the default value is empty.

ssl_version

You can specify what versions of SSL/TLS to permit here.

Acceptable choices are:

- `SSLv23`
- `SSLv2`
- `SSLv3`
- `TLSv1`

There may be other potential values for different environments.

The default value is `SSLv23`

ssl_assert_hostname

If the value of this setting is `true`, hostname verification will take place, otherwise it will not.

The default value is `false`.

http_compress

The value must be `true` or `false` and determines whether gzip compression will be used for client traffic to Elasticsearch.

The default value is `false`.

aws

If you are trying to monitor an Elasticsearch cluster in AWS with Zabbix, you can!

sign_requests

This value should be `true` if you want your requests to be signed with AWS IAM credentials retrieved from your environment.

The order in which credentials will be searched for is:

1. Environment variables
2. Shared credential file (`=/.aws/credentials`)
3. AWS config file (`=/.aws/config`)
4. Boto2 config file (`/etc/boto.cfg` and `=/.boto`)
5. Instance metadata service on an Amazon EC2 instance that has an IAM role configured.

The default value is `false`.

aws_region

This should be an AWS region, such as `us-east`, or left empty.

There is no default value, or the default value is empty.

logging

This is an example configuration block:

```
logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist:
    - 'elasticsearch'
    - 'urllib3'
    - 'werkzeug'
```

loglevel

Must be one of:

- `CRITICAL`

- ERROR
- WARNING
- INFO
- DEBUG

The default value is INFO

logfile

This should be a path to a file writable by the user running the backend.

There is no default value, which means that logs will be written to STDOUT

logformat

Must be one of:

- default
- json
- logstash (same as json)

json and logstash will output logs formatted in JSON suitable for immediate ingest by Elasticsearch.

The default value is default.

blacklist

The log blacklist permits filtering log lines based on the module's root logger. The default values of:

- elasticsearch
- urllib3
- werkzeug

. . . will effectively leave only the logs generated by es_stats_zabbix, es_stats, and protobix.

For full debugging, providing an empty array for this setting will override these defaults:

```
blacklist: []
```

backend

This is an example configuration block:

```
backend:
  host: 127.0.0.1
  port: 7600
  cache_timeout: 60
  debug: false
```

host

host should *always* be localhost or 127.0.0.1.

The backend should not be available to outside queries by itself, so the value of this should never permit outside access.

The default value is 127.0.0.1

port

The port to listen on.

The default value is 7600

cache_timeout

How long to cache the values from an API call.

The default value is 60, meaning 60 seconds.

debug

If true, turn on debug logging for the Flask backend. This results in pretty-printing the command-line output for easier reading.

The default value is false

zabbix

This is an example configuration block:

```
zabbix:
  ServerActive: zabbixserver.example.com
  ServerPort:
  LogType: 'file'
  LogFile: '/tmp/zabbix_agentd.log'
  DebugLevel: 3
  Timeout: 3
  Hostname:
  TLSConnect: 'unencrypted'
  TLSCAFile:
  TLSCertFile:
  TLSCRLFile:
  TLSKeyFile:
  TLSServerCertIssuer:
  TLSServerCertSubject:
  TLSPSKIdentity:
  TLSPSKFile:
```

Values and defaults are described at https://www.zabbix.com/documentation/3.4/manual/appendix/config/zabbix_agentd except as follows:

`ServerPort`

`ServerPort` refers to the listening port on the Zabbix Server.

The default value is 10051.

`endpoints`

This is an example configuration block:

```
endpoints:
  cluster:
    60s:
      clusterstats:
        - ...
  coordinating:
    60s:
      nodestats:
        - ...
  data:
    60s:
      nodestats:
        - ...
```

`cluster`

This block contains endpoints exclusively used for cluster-level monitoring. These include the cluster `health`, `clusterstate`, and `clusterstats` metrics.

The default values included are probably sufficient, but you can remove any items that you do not want discovered and monitored.

`coordinating`

In Elasticsearch terminology, all nodes can act as *coordinating* nodes, so these metrics will be monitored for every node in the cluster. The stats are broken down first by interval (`60s`, `300s`, `900s`), and by API endpoint (`nodestats`, `nodeinfo`).

Everything included here will be discovered, and monitored.

`data`

Differing from `coordinating` node, these stats are specific to data nodes. All endpoints specified here will be discovered and monitored.

Other Node Types

While no endpoints are provided out of the box, you can also monitor:

- `ingest`
- `master`

- ml - Machine Learning

Zabbix Configuration

Configuring Zabbix comes in multiple parts.

zabbix_agentd.conf

It is important to note where es_stats_zabbix installs the entry points for this part. If you installed to a virtualenv, then the path will be /my/chosen/path/bin/. If you installed to the system Python by being root or using sudo, then the path can be determined with which esz_get_stat. The response will be something like /usr/bin/esz_get_stat, so the path would be /usr/bin. If you installed via pip to your username, e.g. pip install --user ., then the path will be $HOME/.local/bin.

Creating es_stats_zbx.conf

A sample of the necessary Zabbix config file entries is included in the source repository at https://github.com/untergeek/es_stats_zabbix/tree/master/configuration/zabbix

```
# $1 is api, $2 is endpoint, $3 is node
UserParameter=es_stat[*],/usr/bin/esz_get_stat --node="$3" $1 $2

# $1 is node, $2 is show_all
UserParameter=es_stats_discovery[*],/usr/bin/esz_discovery --node="$1" --show_all="$2"

# $1 is either 'cluster' or 'nodes'
# $2 is any arbitrary value (allows for multiple keys to use this script)
UserParameter=es_cluster_discovery[*],/usr/bin/esz_cluster_discovery --flag="$2" --
↪value="$1"

# $1 is node, $2 is any arbitrary value (allows for multiple keys to use this script)
UserParameter=es_node_discovery[*],/usr/bin/esz_node_discovery --flag="$2" "$1"

# $1 is node, $2 is any arbitrary value (allows for multiple keys to use this script)
UserParameter=es_trapper_discovery[*],/usr/bin/esz_trapper_discovery --flag="$2" --
↪node="$1"

# $1 is node, $2 is nodetype, $3 is interval
UserParameter=es_trapper_stats[*],/usr/bin/esz_trapper_stats --interval="$3" $1 $2
```

In this file, the PATH is /usr/bin. Replace /usr/bin with your path, e.g. /my/chosen/path/bin/esz_discovery, for each line.

Installing es_stats_zbx.conf

The es_stats_zbx.conf file can then be placed in /etc/zabbix/zabbix_agentd.d as es_stats_zbx.conf. A corresponding Include line should be present in the zabbix_agentd.conf file:

```
Include=/etc/zabbix/zabbix_agentd.d/*.conf
```

This will ensure that anything ending with .conf will be read from /etc/zabbix/zabbix_agentd.d. If your Zabbix agent has been installed to a different path, you may need to adapt to suit the target location.

Adding the discovery template to your Zabbix host

The Zabbix templates can be found at https://github.com/untergeek/es_stats_zabbix/tree/master/configuration/ES_VERSION where `ES_VERSION` will be the major.minor release of Elasticsearch you plan to monitor. Your version may or may not exist in the repository yet.

Instructions for importing templates into Zabbix can be found at https://www.zabbix.com/documentation/3.4/manual/xml_export_import/templates

Tip: The Zabbix host that is running the `es_stats_zabbix` backend should be the one that has the template assigned.

1.2.2 Launching the Backend

The backend must be running for the `esz_*` scripts to work.

Install the configuration file

The backend configuration file should be installed at `/etc/es_stats_zabbix/config.yml`.

This will be overridden by `$HOME/.es_stats_zabbix/config.yml`, if it exists.

`systemd`

Install unit file

A sample systemd unit file is provided at https://github.com/untergeek/es_stats_zabbix/blob/master/configuration/systemd/esz_backend.service

This file can be placed at `/etc/systemd/system/esz_backend.service`. After installing the file there, run `systemctl daemon-reload` to have the system recognize the new unit file.

Launching via `systemd`

```
systemctl start esz_backend.service
```

Manually launching the backend

The path to the entry points must be known. If the scripts were installed to a virtualenv, you might be able to launch via:

```
/my/chosen/path/bin/esz_backend
```

1.3 Changelog

1.3.1 0.10.10 (30 October 2018)

Beta-release of pending 1.0 version

- Basically a complete rewrite. This version is not yet ready for mass consumption. Still pending are: - Zabbix 3.x template for 5.x ES versions. - Installation scripts/config files - Init scripts for the backend (SysV, Upstart) - More documentation - Make different templates for different Zabbix versions, if necessary.
- Testing this pre-release version requires a `git clone` and `python setup.py install` or `pip install .` from the source directory.

New

- Flask-based backend listener that enables cached reads of the cluster stats. This prevents repeated calls from completely hammering the cluster with repeated stats calls. Currently intended to listen on localhost port 7600, so that the Zabbix UserParameter script can simply read localhost. Cache timeout defaults to 60 seconds, and is configurable.
- Use YAML based configuration. - Elasticsearch configuration block - Logging configuration block - Backend configuration block - *do_not_discover* block
- Uses update *es_stats* and new *es_client* modules. - *es_client* allows monitoring of X-Pack secured Elasticsearch clusters
- Discovery based on value (`bool` for true/false, `unsigned` for Zabbix's *unsigned* integer values, `float`, and `character`). This allows Zabbix to automatically discover items for the correct item type.
- Block individual endpoints from LLD by enumerating them in the `do_not_discover` block in the configuration file.

Other

- Fixed issues with the config_override function not behaving as expected.
- Pruned unnecessary code.
- Added unit and integration tests.
- Added endpoints
- Switched to using `setup.cfg` instead of everything in `setup.py`
- Add some backward compatibility.
- Compatibility testing with different versions of Elasticsearch (all 5.x and 6.x minor releases) and Python client versions (2.7, 3.5, 3.6)
- Add `esz_nodes_discovery` for node related macros for discovery.
- Add systemd service file.
- Full test of prototype template with discovery of cluster and nodes.
- Added `esz_cluster_discovery` for cluster related macros for LLD.
- Added `run_display_endpoints.py` to show all endpoints for a given node. This works better with `--apidebug` enabled in the backend, as it pretty prints the results.
- Reworked the launch scripts to use similar code as much as possible.
- Moved a lot of classes and modules around.
- Template for ES 6.3 (may work for any 6.x).

- Default `config.yml` for Elasticsearch 6.x
- Initial documentation created
- Absence of `protobix` corrected (jameskirsop) #8
- Removed superfluous `api` prefixes in `config.yml` (untergeek) #10
- Pinned `click` version to be less than 7.0 for now

1.3.2 0.1.4 (22 June 2016)

Bug Fixes

- Fix `es_stats` dependency to be 0.2.1, which fixes some reported bugs.

General

- Fix docs in their as yet incomplete state to at least not have incorrect information

1.3.3 0.1.1 (7 October 2015)

New

- Batch now reports count of items which failed on command-line. This enables you to call batch as a Zabbix agent item, and report a result.

Bug Fixes

- Prevent empty lists in batches from generating an error.

1.3.4 0.1.0 (7 October 2015)

New

- Refactor all Zabbix key parsing. Much simpler now.
- Improved logging.

1.3.5 0.0.2 (6 October 2015)

Bug fixes

- Put kaptan dependency in place

1.3.6 0.0.1 (6 October 2015)

Announcement

- Initial release

CHAPTER 2

License

CHAPTER 3

Indices and tables

- genindex
- search