



# Zabbix and Elastic

Elastic as history storage back-end

ZabConf BeNeLux 01-02-2019

**ZABBIX**

# \$ whoami

```
gecos:      Robert Hoekstra  
jobtitle:   Linux and Zabbix Consultant  
Org:        Xifeo ICT  
Special1:   Zabbix Certified Professional (6+ yrs)  
Special2:   Red Hat Certified Architect (9+ yrs)
```



# Why This Topic

- Unfamiliar with Elastic
- Intrigued by the support for it since 3.4
- Experimenting
- Sharing my experiences so far

# Elastic Content

- Current Database Back-End Setup
- About Elastic
- Zabbix and Elastic as History Back-End
- Current Install - Migrate or Not
- How to set up
- What's Next
- Questions

# Current Database Back-End Setup

- Structured databases, they do have their limits
  - History Housekeeping needed, or
  - Partitioning
- Typically a single server storing and servicing the data
  - Scales vertically (more CPU / Memory)
- Replication / back-up must be arranged for HA/DR
  - Often relying on 3<sup>rd</sup> party solutions like VM- or volume snapshots
- Database needs tweaking to increase performance/efficiency

# About Elastic

- NoSQL
- Unstructured data store
- JSON RESTful API
- Scalable – Distributed by nature
  - Hardware fault tolerant because of replication
  - No single node to query
- Free, but with limitations
  - Mainly concerning security (RBAC, Auditing, Alerting) and some more advanced features like Machine Learning

# Zabbix and Elastic as History Back-End

- Separate config database (still SQL based)
  - Easier to back-up
  - Holds information about all your hosts/items, just no historical information
- No housekeeping for history
  - History does not get purged by Zabbix
  - Just delete – date based – indices to purge
- Scalable history back-end
  - Just add nodes to Elastic cluster
- Not to be confused with Zabbix MONITORING content in Elastic !!

**ZABBIX**



# Current Install - Migrate or Not

- Only one history back-end possible per item type
- Either
  - Drop history and start new, in Elastic
  - Migrate data from DB to Elastic
    - Possible with some scripting – not out-of-the-box
- You need scale to gain performance in Elastic (no numbers, sorry)
- Querying history needs knowledge of data, but is possible
  - Itemid and data is in Elastic, meaning of id is in SQL database



# How to set up - Preparations

- Read documentation 😊
- Install, like ‘yum install elasticsearch’
- Define mappings and possibly templates
  - Found in database/elasticsearch/elasticsearch.map
  - Templates for automatic index creation – date based
- Make sure SELinux and Firewall are configured to allow communications
- Watch log files to find any errors that need fixing to get it working

# How to set up - Zabbix Server

- zabbix\_server.conf

```
### Option: HistoryStorageURL
#   History storage HTTP[S] URL.
#
# Mandatory: no
# Default:
# HistoryStorageURL=
HistoryStorageURL=http://localhost:9200

### Option: HistoryStorageTypes
#   Comma separated list of value types to be sent to the history storage.
#
# Mandatory: no
# Default:
# HistoryStorageTypes=uint,dbl,str,log,text
HistoryStorageTypes=uint,dbl,str,log,text

### Option: HistoryStorageDateIndex
#   Enable preprocessing of history values in history storage to store values in different indices based on date.
#   0 - disable
#   1 - enable
#
# Mandatory: no
# Default:
# HistoryStorageDateIndex=0
HistoryStorageDateIndex=0
```

# How to set up - Zabbix Web Front-end

- zabbix.conf.php

```
// Zabbix GUI configuration file.  
global $DB, $HISTORY;  
  
// Elasticsearch url (can be string if same url is used for all types).  
$HISTORY['url'] = 'http://localhost:9200';  
//$HISTORY['url'] = [  
//     'uint' => 'http://localhost:9200',  
//     'dbl' => 'http://localhost:9200',  
//     'str' => 'http://localhost:9200',  
//     'log' => 'http://localhost:9200',  
//     'text' => 'http://localhost:9200'  
//];  
// Value types stored in Elasticsearch.  
$HISTORY['types'] = ['uint', 'dbl', 'str', 'log', 'text'];
```

# How to set up – Verify Functionality

- `curl http://localhost:9200/_cat/indices?v`

```
> curl http://localhost:9200/_cat/indices?v
health status index      uuid                                pri rep docs.count docs.deleted store.size pri.store.size
yellow open   text      8VPXpHQiR7qHd8loGfVMzw           5  1         0             0         1.2kb         1.2kb
green  open   .kibana_1 opXGk7kITGe2GXoT3RTxyQ           1  0          6             0         15.9kb        15.9kb
yellow open   str       zmaLR_xjQVifqVQwrRywhw           5  1        588             0        195.5kb        195.5kb
yellow open   log       9Xqw7TEzToy73Xr2Fwr5yg           5  1          0             0          1.2kb          1.2kb
yellow open   db1       f_raSFa9R1mcDSxL-A3FgA           5  1       340954           0        34.5mb         34.5mb
green  open   .kibana_2 IRnARzGiRheo9Z2APsRx7w           1  0          7             0         19.2kb         19.2kb
green  open   .tasks    wxaCdhVVSkiMC3iliNeuEw           1  0          1             0          6.2kb          6.2kb
yellow open   uint     UUM_R80qRm6aN4zv0sJbMg           5  1       144666           0        14.1mb         14.1mb
```

# What's Next – My Personal Observations

- Still experimental – too soon to jump into it for production?
- What other uses for having history in Elastic?
- Difficult – for me – to find real use case

# Questions?

Thank you, now it's time for lunch . . .

