ZABBIX

ZABBIX HOUSEKEEPER AND DATA RETENTION PERIOD

KASPARS MEDNIS
CHIEF TRAINER
ZABBIX

ZABBIX

HOUSEKEEPER SETTINGS

WHY ZABBIX NEEDS HOUSEKEEPER?

Zabbix is generating a lot of data on regular basis:

- ✓ Zabbix is collecting metrics
- ✓ Zabbix is calculating hourly trends
- ✓ Zabbix is detecting problems and creating events
- ✓ Zabbix is calculating IT service status
- ✓ Zabbix is tracing audit records
- ✓ Zabbix is keeping user sessions

Without housekeeper the database size will grow indefinitely

HOW FAST WILL YOUR DATA GROW?

- The bigger the NVPS (New Values per Second), the faster the database will grow

```
CREATE TABLE `history` (
  `itemid` bigint unsigned NOT ULL,
  `clock` int NOT NULL DEFAULT '0',
  `value` double NOT NULL DEFAULT '0',
  `ns` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`itemid`,`clock`,`ns`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
```

```
+--------+------------+--------------------+-----------+
| itemid | clock      | value              | ns        |
+--------+------------+--------------------+-----------+
|  10073 | 1676374673 | 0.5814519861048069 | 536801725 |
+--------+------------+--------------------------------+
```

- bigint (8B) + int (4B) + double (8B) + int (4B) + index (16B) ~ 40 bytes per history entry
- history_str, history_text and history_log entries can use even more space

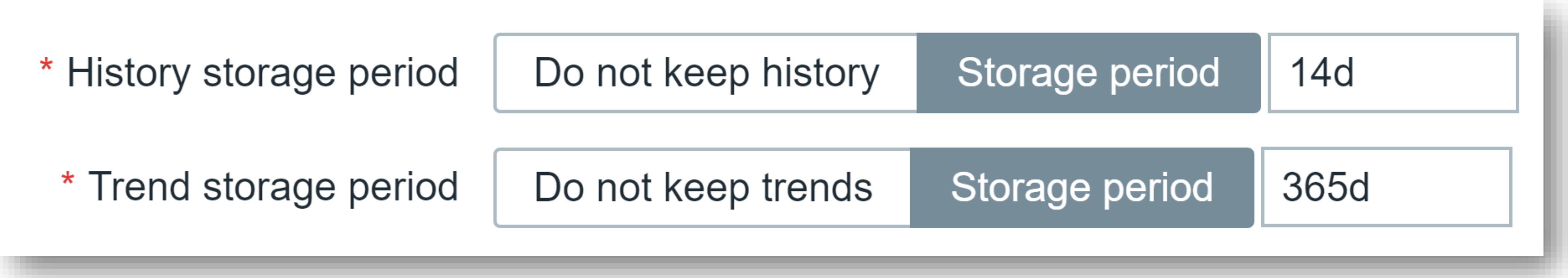
WHAT HOUSEKEEPER IS / IS NOT DOING?

Housekeeper will remove historical data based on the retention period:

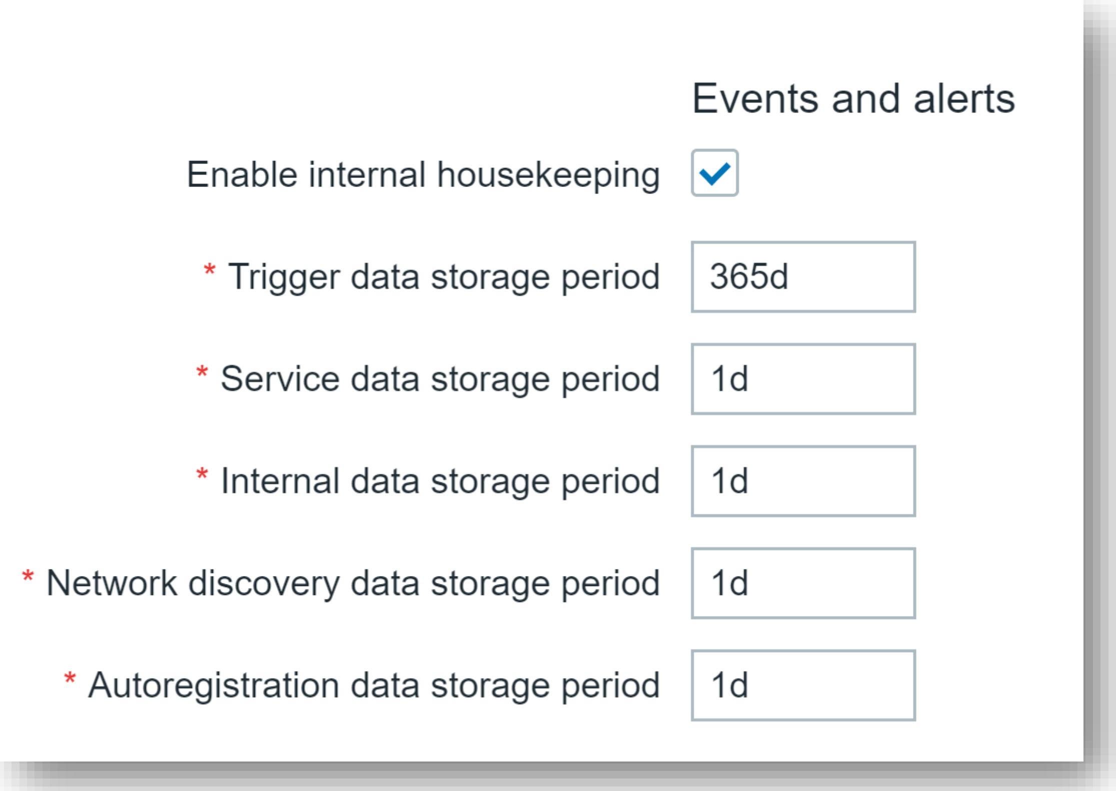
- retention periods can be defined individually for each item
- for each type of data (history, events, etc.) different retention period can be specified

Housekeeper will not:

- remove any configuration data (items, hosts, templates etc.)
- delete trigger events in a problem state
- keep Zabbix database at a predefined size
- delete data when "Clear history" button is pressed



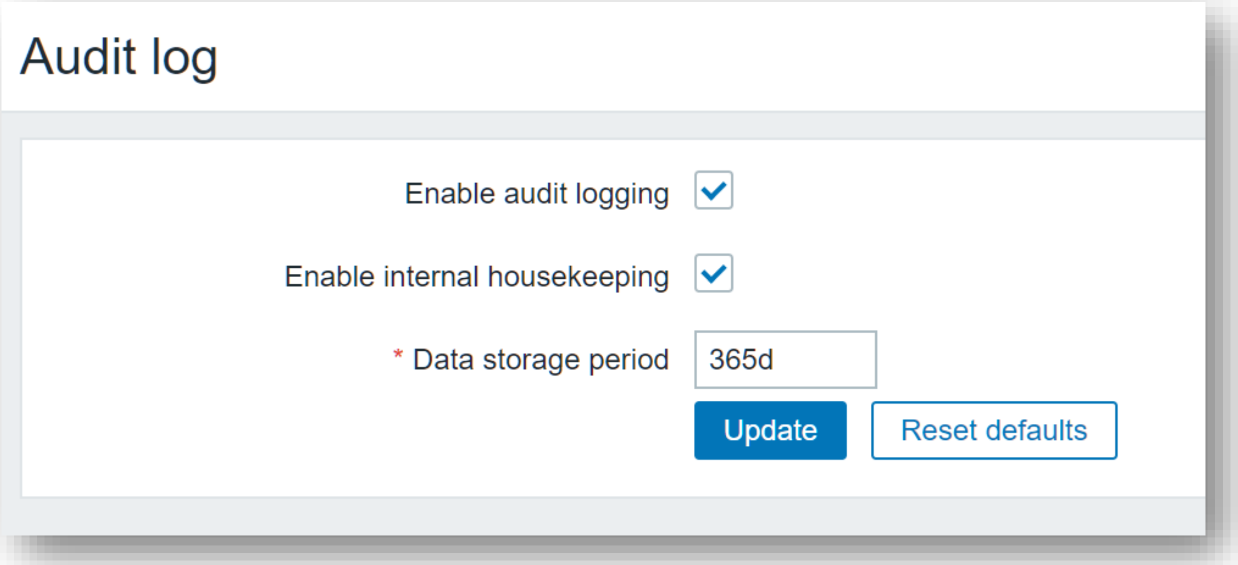
* History storage period	Do not keep history	Storage period	14d
* Trend storage period	Do not keep trends	Storage period	365d



	Events and alerts
Enable internal housekeeping	☑
* Trigger data storage period	365d
* Service data storage period	1d
* Internal data storage period	1d
* Network discovery data storage period	1d
* Autoregistration data storage period	1d

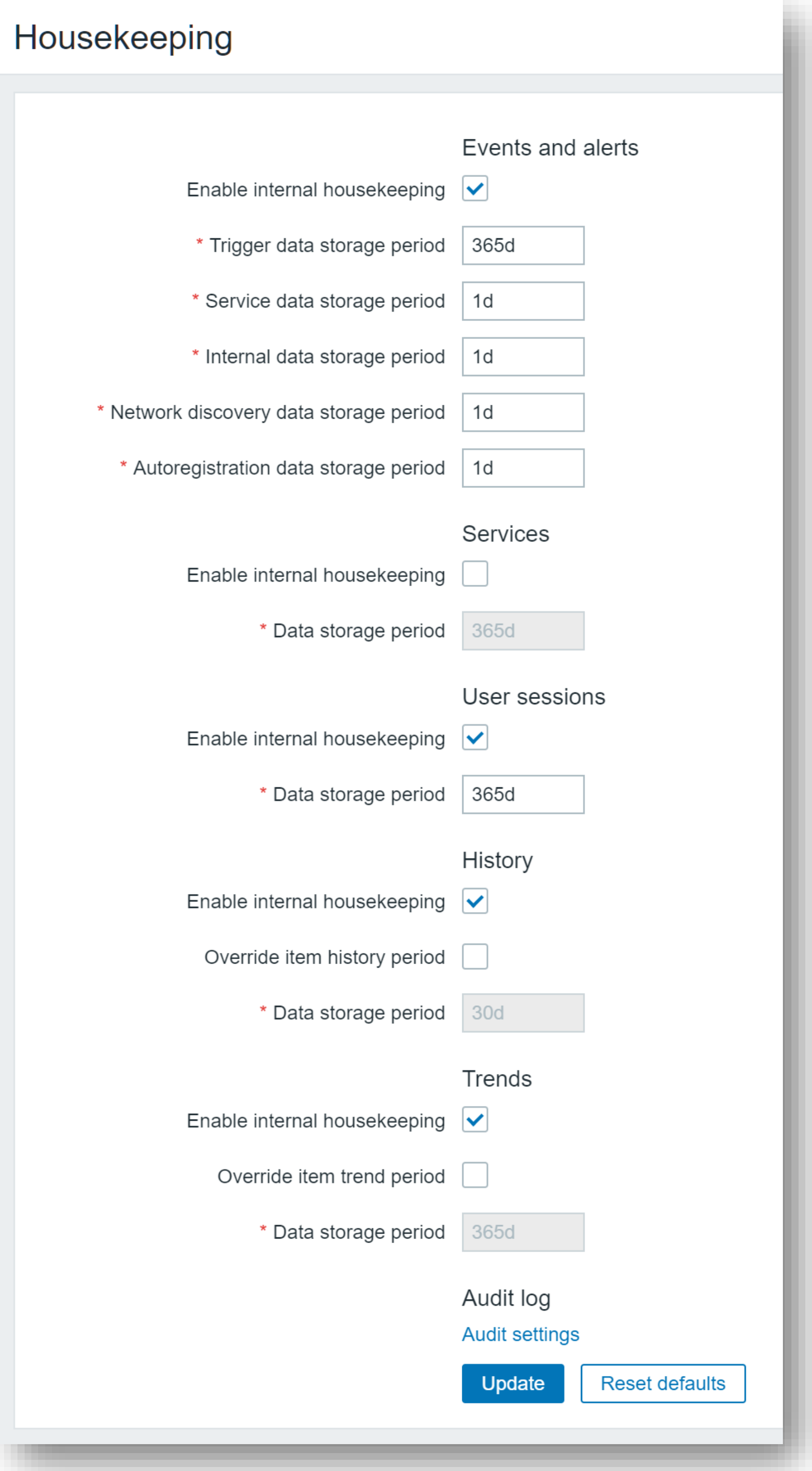
DATA RETENTION PERIODS

- Periods are defined in Administration -> General -> Housekeeping
- Audit log settings are defined in a separate section
- Housekeeper can be individually enabled or disabled for each data type



Audit log

Enable audit logging	☑
Enable internal housekeeping	☑
* Data storage period	365d



Housekeeping

Events and alerts

Enable internal housekeeping	☑
* Trigger data storage period	365d
* Service data storage period	1d
* Internal data storage period	1d
* Network discovery data storage period	1d
* Autoregistration data storage period	1d

Services

Enable internal housekeeping	☐
* Data storage period	365d

User sessions

Enable internal housekeeping	☑
* Data storage period	365d

History

Enable internal housekeeping	☑
Override item history period	☐
* Data storage period	30d

Trends

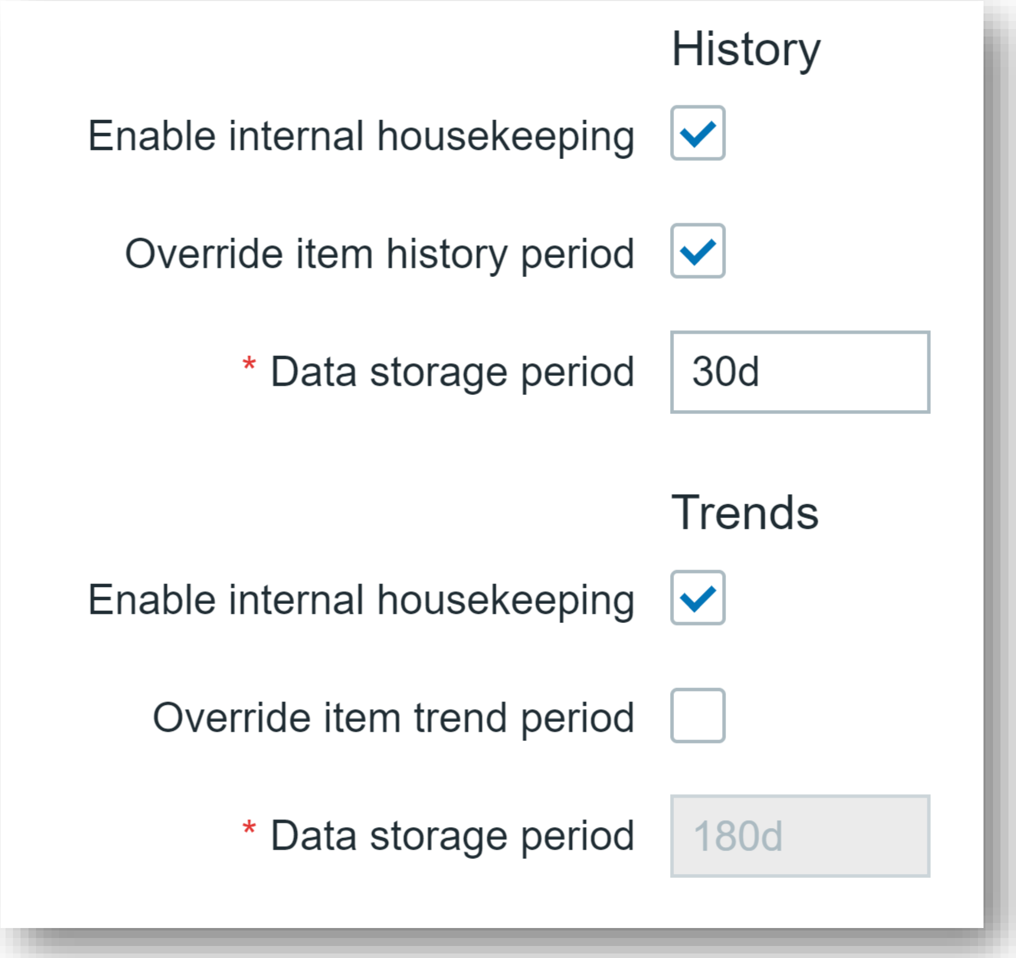
Enable internal housekeeping	☑
Override item trend period	☐
* Data storage period	365d

Audit log

Audit settings

DATA RETENTION PERIOD OVERRIDES

- History and trends periods can be overridden globally for all items
- Override is indicated by a yellow ! in the item configuration form
- Overrides are also used for TimescaleDB configuration



History

Enable internal housekeeping ☑

Override item history period ☑

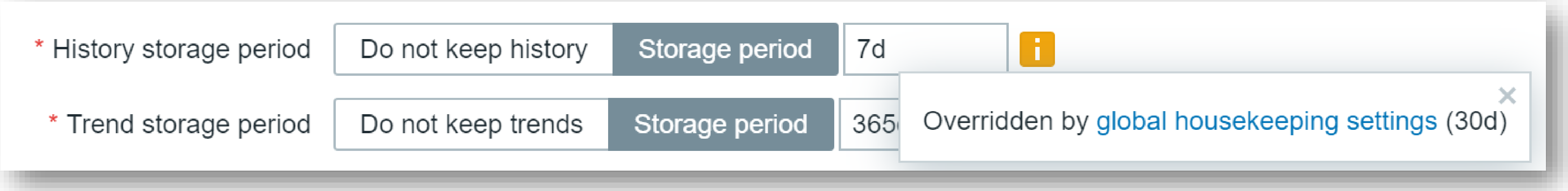
* Data storage period 30d

Trends

Enable internal housekeeping ☑

Override item trend period ☐

* Data storage period 180d



* History storage period | Do not keep history | Storage period | 7d | i

* Trend storage period | Do not keep trends | Storage period | 365

Overridden by global housekeeping settings (30d) ✕

ZABBIX

HOUSEKEEPER CONFIGURATION

HOUSEKEEPER CONFIGURATION

- Housekeeper process is configured in the Zabbix server configuration file

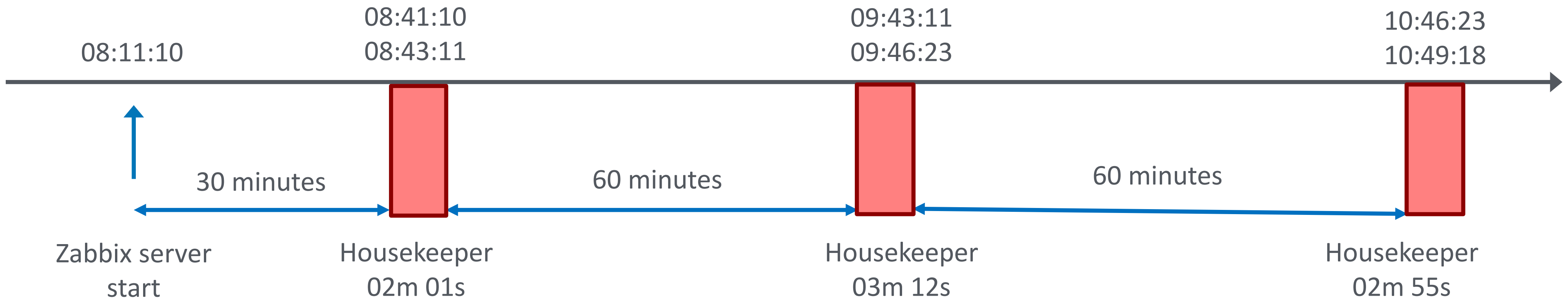
```
### Option: HousekeepingFrequency
#       How often Zabbix will perform housekeeping procedure (in hours).
#       Housekeeping is removing outdated information from the database.
#       To prevent Housekeeper from being overloaded, no more than 4 times HousekeepingFrequency
#       hours of outdated information are deleted in one housekeeping cycle, for each item.
#       To lower load on server startup housekeeping is postponed for 30 minutes after server start.
#       With HousekeepingFrequency=0 the housekeeper can be only executed using the runtime control option.
#       In this case the period of outdated information deleted in one housekeeping cycle is 4 times the
#       period since the last housekeeping cycle, but not less than 4 hours and not greater than 4 days.
# Range: 0-24
HousekeepingFrequency=1

### Option: MaxHousekeeperDelete
#       The table "housekeeper" contains "tasks" for housekeeping procedure in the format:
#       [housekeeperid], [tablename], [field], [value].
#       No more than 'MaxHousekeeperDelete' rows (corresponding to [tablename], [field], [value])
#       will be deleted per one task in one housekeeping cycle.
#       If set to 0 then no limit is used at all. In this case you must know what you are doing!
# Range: 0-1000000
MaxHousekeeperDelete=5000
```

HOUSEKEEPER FREQUENCY SETTING

- Housekeeper is executed once per hour by default (HousekeepingFrequency=1)
- 60 minutes is measured after the previous run has finished
- Housekeeper execution is postponed 30 minutes from Zabbix server/proxy start

```
081110.110 server #2 started [housekeeper #1]
084110.112 executing housekeeper
084311.228 housekeeper [deleted 24456 hist/trends, 25431 items/triggers, ... in 181.017761 sec, idle for 1 hour(s)]
094311.633 executing housekeeper
094623.706 housekeeper [deleted 24430 hist/trends, 25514 items/triggers, ... in 192.047072 sec, idle for 1 hour(s)]
104623.819 executing housekeeper
104918.583 housekeeper [deleted 24102 hist/trends, 24570 items/triggers, ... in 175.026543 sec, idle for 1 hour(s)]
```



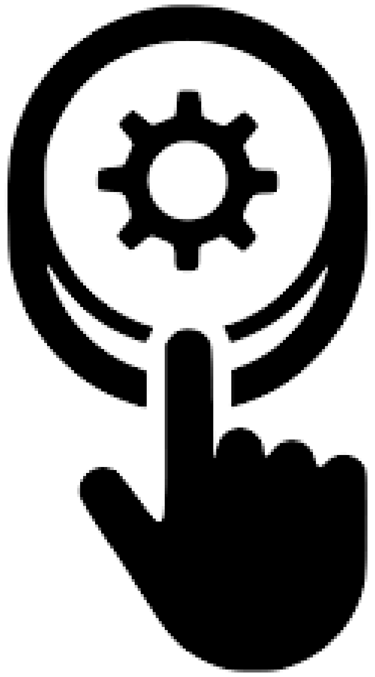
MANUAL HOUSEKEEPER EXECUTION

- Housekeeper process can be completely disabled by setting HousekeepingFrequency = 0
- Runtime command can be used to execute housekeeper manually

```
# zabbix_server -R housekeeper_execute
```

```
 10071:20230215:123734.183 forced execution of the housekeeper
 10071:20230215:123734.183 executing housekeeper
 10071:20230215:123734.231 housekeeper [deleted 41253 hist/trends, 243 items/triggers, 45 events, 13 problems, 5
sessions, 10 alarms, 125 audit, 0 records in 3.046571 sec, idle for 1 hour(s)]
```

- Sometimes repeated manual execution is useful to speed up cleanup
- A custom housekeeping schedule can also be defined using cron jobs



HOUSEKEEPING IN DETAILS

```
 20230215:123734.183 forced execution of the housekeeper
 20230215:123734.183 executing housekeeper
 20230215:123734.231 housekeeper [deleted 41253 hist/trends, 243 items/triggers, 45 events, 13
problems, 5 sessions, 10 alarms, 125 audit, 0 records in 3.046571 sec, idle for 1 hour(s)]
```

A closer look at housekeeper results:

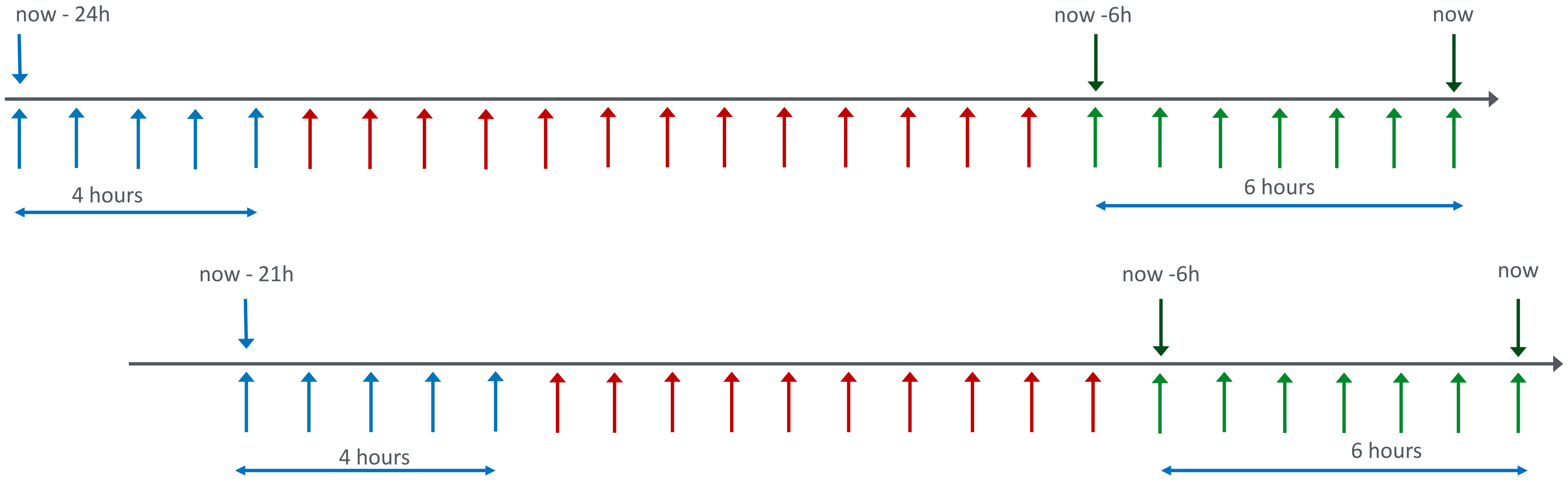
deleted 41253 hist/trends	expired data entries from history*/trends* tables
243 items/triggers	data entries and events for deleted(!!!) items and triggers
45 events	expired events event* from events tables
13 problems	expired problems from problem table
5 sessions	expired user sessions from sessions table
10 alarms	expired services alarms from service_alarms table
125 audit	expired audit entries from audit table
0 records	discovery data from proxies

HOW HOUSEKEEPER REMOVES EXPIRED DATA

- Let's take a look at the configuration file again:

```
#       To prevent Housekeeper from being overloaded, no more than 4 times HousekeepingFrequency
#       hours of outdated information are deleted in one housekeeping cycle, for each item.
HousekeepingFrequency=1
```

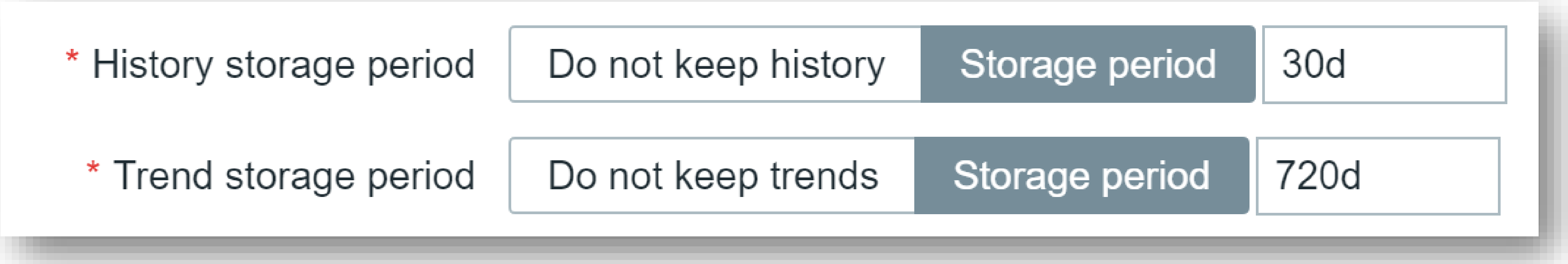
- Normally housekeeper each hour removes one hour of expired information (sliding window)
- When reducing history/trends storage, old data will be removed in a 4-hour portions
- Example - history storage reduced from 24 hours to 6 hours



EXPRIED DATA EXAMPLE

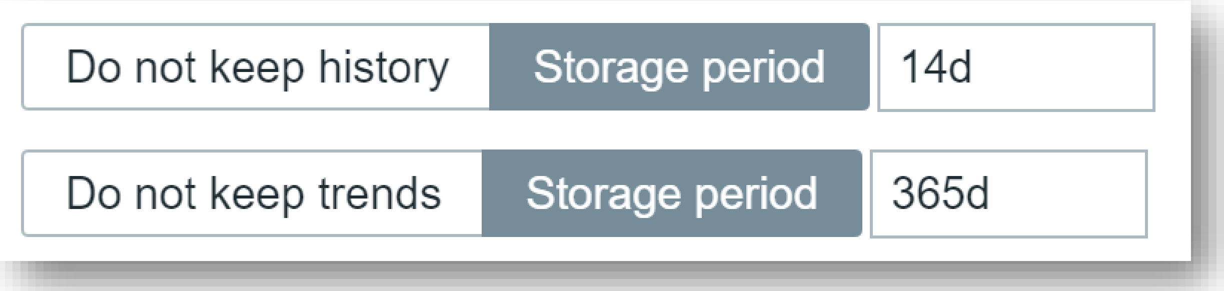
In this example history and trends storage period was reduced for item with itemid 28912

Housekeeper runs every hour and removes 4 hours of expired data each time



* History storage period: Do not keep history | Storage period | 30d

* Trend storage period: Do not keep trends | Storage period | 720d



Do not keep history | Storage period | 14d

Do not keep trends | Storage period | 365d

```
102408.294 query [txnlev:0] [delete from history where itemid=28912 and clock<1627302592]
102408.657 query [txnlev:0] [delete from trends where itemid=28912 and clock<1627300800]
112434.228 query [txnlev:0] [delete from history where itemid=28912 and clock<1627316992]
112434.557 query [txnlev:0] [delete from trends where itemid=28912 and clock<1627315200]
.....
```

history table entries	1627316992 - 1627302592 = 14400 seconds = 4 hours
trends table entries	1627315200 - 1627300800 = 14400 seconds = 4 hours

HOW HOUSEKEEPER REMOVES DATA FROM DELETED ITEMS

- MaxHousekeeperDelete parameter is used for data from deleted items only

```
#       No more than 'MaxHousekeeperDelete' rows (corresponding to [tablename], [field], [value])
#       will be deleted per one task in one housekeeping cycle.
#       If set to 0 then no limit is used at all. In this case you must know what you are doing!
MaxHousekeeperDelete=5000
```

- When item is deleted, corresponding entries in the housekeeper table are created

```
+---------------+--------------+-----------+-------+
| housekeeperid | tablename    | field     | value |
+---------------+--------------+-----------+-------+
|          1408 | events       | triggerid | 16199 |
|          1422 | trends       | itemid    | 29076 |
|          1423 | trends_uint  | itemid    | 29076 |
|          1424 | history_text | itemid    | 29076 |
|          1425 | history_log  | itemid    | 29076 |
|          1426 | history_uint | itemid    | 29076 |
|          1427 | history_str  | itemid    | 29076 |
|          1428 | history      | itemid    | 29076 |
|          1429 | events       | itemid    | 29076 |
+---------------+--------------+-----------+-------+
```

HOUSEKEEPER TABLE

- Housekeeper will delete MaxHousekeeperDelete records from each table specified in one cycle
- Entry is removed from this table when there is no more records left
- Deleting rows manually from housekeeper table may leave some historical data forever in DB
- In the example some data still must be deleted from some tables for deleted items with itemids 29076 and 35476

```
+---------------+---------------+------------+-------+
| housekeeperid | tablename     | field      | value |
+---------------+---------------+------------+-------+
|          1428 | history       | itemid     | 29076 |
|          1429 | events        | itemid     | 29076 |
|          1430 | history_uint  | itemid     | 35476 |
|          1431 | trends_uint   | itemid     | 35476 |
+---------------+---------------+------------+-------+
```

WHY HOUSEKEEPER MAY BE SLOW?

- Large number of items - two complex queries are executed to collect all items and history periods from configuration

```
select i.itemid, i.value_type, i.history, i.trends, h.hostid from items i, hosts h
select itemid, min(clock) from history group by itemid
select itemid, min(clock) from history_uint group by itemid
.....
```

- Large number of NVPS means a lot of values to delete

```
102408.294 query [txnlev:0] [delete from history where itemid=28912 and clock<1627302592]
102408.657 query [txnlev:0] [delete from trends where itemid=28912 and clock<1627300800]
.....
```

- Recently decreased history storage period - more data to delete
- Lots of items deleted recently with large MaxHousekeeperDelete setting

PROBLEM HOUSEKEEPER PROCESS

- There is a separate housekeeper process which removes only problems for deleted triggers
- This process runs every minute by default
- This is required by IT services to quickly update service status if trigger is deleted

```
### Option: ProblemHousekeepingFrequency
#       How often Zabbix will delete problems for deleted triggers (in seconds).
#
# Mandatory: no
# Range: 1-3600
# Default:
ProblemHousekeepingFrequency=60
```

```
zbx_setproctitle() title:'trigger housekeeper [removing deleted triggers problems]'
query select eventid from problem where source=0 and object=0 and not exists
(select NULL from triggers where triggerid=objectid)
```

ZABBIX

HOUSEKEEPER ON PROXIES

HOUSEKEEPER ON PROXIES

- Housekeeper process runs on Zabbix proxies and removes data which are sent to Zabbix server
- Only HousekeepingFrequency parameter can be defined on proxies
- Same logic with 4 times HousekeepingFrequency is used

```
### Option: HousekeepingFrequency
#       How often Zabbix will perform housekeeping procedure (in hours).
#       Housekeeping is removing outdated information from the database.
#       To prevent Housekeeper from being overloaded, no more than 4 times HousekeepingFrequency
#       hours of outdated information are deleted in one housekeeping cycle.
#       To lower load on proxy startup housekeeping is postponed for 30 minutes after proxy start.
#       With HousekeepingFrequency=0 the housekeeper can be only executed using the runtime control
#       option. In this case the period of outdated information deleted in one housekeeping cycle
#        is 4 times the period since the last housekeeping cycle, but not less than 4 hours and not
#        greater than 4 days.
# Mandatory: no
# Range: 0-24
# Default:
HousekeepingFrequency=1
```

PROXY DATABASE BUFFER SETTINGS

- Two parameters on proxies are used to define "expired data"
- If ProxyLocalBuffer is > 0, data may be kept in DB even if already sent to Zabbix server
- All data which are older than specified in these settings is removed by proxy housekeeper

```
### Option: ProxyLocalBuffer
#       Proxy will keep data locally for N hours, even if the data have already been synced
with the server.
#       This parameter may be used if local data will be used by third party applications.
# Mandatory: no
# Range: 0-720
ProxyLocalBuffer=0

### Option: ProxyOfflineBuffer
#       Proxy will keep data for N hours in case if no connectivity with Zabbix Server.
#       Older data will be lost.
# Mandatory: no
# Range: 1-720
ProxyOfflineBuffer=1
```

PROXY HISTORY TABLES

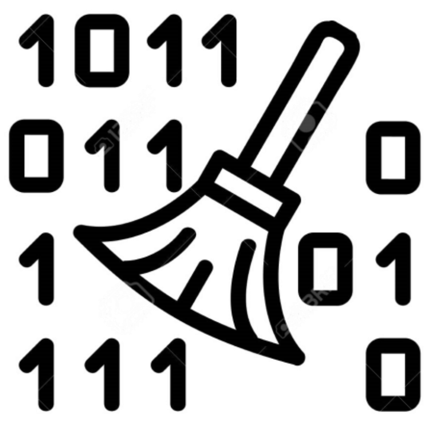
- The only table on proxy which is cleaned by housekeeper is proxy_history
- This table stores all numerical and textual values which must be sent to Zabbix server
- proxy_history is the only table which uses DB autoincrement
- At the same time, ids table has nextid entry for proxy_history…
- Confused?

```
proxy_history | CREATE TABLE `proxy_history` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT,
  `itemid` bigint unsigned NOT NULL,
  `clock` int NOT NULL DEFAULT '0',
  `timestamp` int NOT NULL DEFAULT '0',
.....
```

```
table_name     field_name      nextid
-------------  --------------  ----------
proxy_history  history_lastid  19883
```

PROXY HISTORY CLEANUP

- proxy_history table uses autoincrement for the id column
- nextid field in the ids table keeps the last sent record id
- data sender sends records from proxy_history with id > nextid and keeps nextid up-to-date
- Housekeeper is removing expired information based on the buffer settings, timestamp and nextid value



```
table_name     field_name      nextid
-------------  --------------  ----------
proxy_history  history_lastid  19883
```

```
id          itemid      value       write_clock
----------  ----------  ----------  -----------
19881       28983       0.456       1614778803
19882       29003       2           1614778803
19883       28998       5           1614778803
19884       28994       Centos 8.1  1614778804
19885       28999       0.22165462  1614778804
```

QUESTIONS?

KASPARS MEDNIS
CHIEF TRAINER
ZABBIX

ZABBIX

THANK YOU!

KASPARS MEDNIS
CHIEF TRAINER
ZABBIX