ZABBIX 5.0

UPGRADING TO 5.0 - BEST PRACTICE AND COMMON PITFALLS.
PREPARING FOR THE UPGRADE

NEW REQUIREMENTS, BACKUPS AND SCHEDULING DOWNTIMES
- Updated requirements for PHP and DB backend
- Backing up your zabbix infrastructure
- Estimating the potential downtime
UPDATED REQUIREMENTS FOR 5.0

- Minimum supported version for PHP is now 7.2 (from 5.4)
- mbedTLS (former polarSSL) is no longer supported for encryption.
- Added support of LIBSSH to support newer platforms like RHEL 8
- MySQL 5.5.62-8.0.x (from 5.0.3)
- Oracle 11.2 (from 10g)
- PostgreSQL 9.2.24 (from 8.1)
- Timescale 1.0 or later
- IBM DB2 support dropped
PREPARING FOR THE BACKUP

☑ Check for any OS updates. If the decision is made to apply the updates for stability/performance reasons, apply the updates and give your environment a few days to detect any new potential issues.

☑ The same rule applies for DB backend upgrades and configuration changes.

☑ This helps us to rule out any performance issues/instabilities caused by any of the performed changes which are unrelated to Zabbix itself.
PREPARING FOR THE BACKUP

✔ Check for any custom solutions used in your Zabbix instance

✔ For upgrading from versions <= 3.0, partitioning will have to be turned off before proceeding with the upgrade

✔ Are there any custom modules or patches applied on your instance?

✔ Are packages available on the underlying OS? Does my policy allow using packages to install Zabbix? Am I able to compile Zabbix or Zabbix packages from source?
BACKING UP YOUR ZABBIX INFRASTRUCTURE

- Perform the Zabbix database backend, server and frontend file backup
- Perform the backup of any custom scripts, modules or any other customizations that are applied to your Zabbix instance
- Back up the configuration files
BACKING UP YOUR YOUR ZABBIX CONFIGURATION

#cp -r /usr/lib/zabbix/externalscripts/ /tmp/zabbix_backup
#cp -r /etc/zabbix/ /tmp/zabbix_backup/
#cp -r /etc/httpd/ /tmp/zabbix_backup/
#cp -r /usr/share/zabbix/ /tmp/zabbix_backup/web/
#cp -r /usr/share/doc/zabbix-* /tmp/zabbix_backup/doc/
BACKING UP YOUR DB CONFIGURATION TABLES

ESTIMATING THE DOWNTIME

✔ Check the size of the database

```sql
SELECT table_schema AS "<zabbix>",
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS "Size in Gb"
FROM information_schema.TABLES
GROUP BY table_schema;
```
ESTIMATING THE DOWNTIME

☐ Check the size of the database tables

```
SELECT table_name, table_rows, data_length, index_length, round(((data_length + index_length) / 1024 / 1024 ),2) "Size in MB" FROM information_schema.tables WHERE table_schema = "zabbix" order by round(((data_length + index_length) / 1024 / 1024 ),2) DESC LIMIT 20;
```
ESTIMATING THE DOWNTIME

- Configuration tables usually take a relatively small amount of space

<table>
<thead>
<tr>
<th>table_name</th>
<th>table_rows</th>
<th>data_length</th>
<th>index_length</th>
<th>Size in GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>alerts</td>
<td>2049491</td>
<td>1431306240</td>
<td>324812800</td>
<td>1.64</td>
</tr>
<tr>
<td>items</td>
<td>2199915</td>
<td>1116225536</td>
<td>448479232</td>
<td>1.46</td>
</tr>
<tr>
<td>triggers</td>
<td>1277344</td>
<td>298336256</td>
<td>119996416</td>
<td>0.39</td>
</tr>
<tr>
<td>item_discovery</td>
<td>1921170</td>
<td>245071872</td>
<td>161660928</td>
<td>0.38</td>
</tr>
<tr>
<td>history_text</td>
<td>2623617</td>
<td>187858944</td>
<td>199557120</td>
<td>0.36</td>
</tr>
<tr>
<td>items_applications</td>
<td>2115352</td>
<td>147439616</td>
<td>188497920</td>
<td>0.31</td>
</tr>
<tr>
<td>auditlog</td>
<td>1588337</td>
<td>203145216</td>
<td>71483392</td>
<td>0.26</td>
</tr>
<tr>
<td>trigger_discovery</td>
<td>1057864</td>
<td>63160320</td>
<td>43188224</td>
<td>0.10</td>
</tr>
<tr>
<td>graphs</td>
<td>277365</td>
<td>61440000</td>
<td>42909696</td>
<td>0.10</td>
</tr>
<tr>
<td>functions</td>
<td>723781</td>
<td>41500672</td>
<td>54525952</td>
<td>0.09</td>
</tr>
</tbody>
</table>
ESTIMATING THE DOWNTIME

On the other hand, history*, trend* and events tables tend to rapidly grow in size on large scale instances

<table>
<thead>
<tr>
<th>table_name</th>
<th>table_rows</th>
<th>data_length</th>
<th>index_length</th>
<th>Size in GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>history</td>
<td>2383675122</td>
<td>143353069568</td>
<td>92952838144</td>
<td>220.08</td>
</tr>
<tr>
<td>events</td>
<td>1079791931</td>
<td>68071456768</td>
<td>99840163840</td>
<td>156.38</td>
</tr>
<tr>
<td>trends</td>
<td>1397366123</td>
<td>136157478912</td>
<td>0</td>
<td>126.81</td>
</tr>
<tr>
<td>trends_uint</td>
<td>1114431630</td>
<td>110791983104</td>
<td>0</td>
<td>103.18</td>
</tr>
<tr>
<td>history_uint</td>
<td>415753897</td>
<td>25146097664</td>
<td>16716546048</td>
<td>38.99</td>
</tr>
<tr>
<td>history_str</td>
<td>161111152</td>
<td>11504795648</td>
<td>6574358528</td>
<td>16.84</td>
</tr>
</tbody>
</table>
CLEANING UP THE EVENTS TABLE (1)

☑ Set Event storage period to 1 day
☑ Manually execute the housekeeper process until the tables are cleared

#zabbix_server -R housekeeper_execute

housekeeper [deleted 20764 hist/trends, 0 items/triggers, 41934 events, 2301 problems, 0 sessions, 0 alarms, 0 audit, 0 records in 0.646578 sec, idle for 1 hour(s)]
CLEANING UP THE EVENTS TABLE (2)

- Use a basic for loop to execute DELETE statements with a LIMIT clause
- Can take a long time if the events table has grown extremely large over time

```bash
#!/bin/bash
for i in {1..50}
  do
    mysql -uroot -ppassword -e "DELETE FROM zabbix.events where source in (1,2,3) limit 100000;"
  done

This is a workaround, use only if you’re having issues with the first method!
CLEANING UP THE EVENTS TABLE (3)

- Copy events with source 0 (trigger events) to a new table

```sql
create table events_new like events;
insert into events_new select * from events where source=0;
RENAME TABLE events to events_old;
RENAME TABLE events_new TO events;
```

- You will have to drop and recreate all of the constraints referencing events on other tables – since they will be linked to the renamed events_old table!

```sql
CONSTRAINT `c_acknowledges_2` FOREIGN KEY (`eventid`) REFERENCES `events_old` (`eventid`) ON DELETE CASCADE
```

This is a workaround, use only if you’re having issues with the first method!

**NO changes to events table when upgrading from 4.0 – 5.0!**
CREATING THE TEMPORARY HISTORY TABLES

RENAME TABLE history_text TO history_text_old;
RENAME TABLE history_log TO history_log_old;

CREATE TABLE history_text like history_text_old;
CREATE TABLE history_log like history_log_old;

NO changes to history tables when upgrading from 4.0 – 5.0!
STEP BY STEP UPGRADE
EXAMPLE USE CASE

- CentOS 7 with Zabbix 3.0 server, frontend and 3 proxies
- One of the proxies uses Amazon Linux AMI OS
- Zabbix server uses MariaDB 10.2 as the DB backend
- Zabbix Proxies use a mix of MariaDB 5.5 and SQLite
# UPGRADE THE SERVER

```bash
#rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/7/x86_64/zabbix-release-5.0-1.el7.noarch.rpm
#yum clean all
#yum -y upgrade zabbix-server-mysql zabbix-agent
```
UPGRADE THE FRONTEND

- Install the Zabbix 5.0 repository and the CentOS SCL repository

```
#rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/7/x86_64/zabbix-release-5.0-1.el7.noarch.rpm
#yum clean all
#yum -y install centos-release-scl
```

- Enable the frontend repo

```
#vi /etc/yum.repos.d/zabbix.repo

[zabbix-frontend]
enabled=1
```
UPGRADE THE FRONTEND

- Reinstall the web server with the apache configuration
- Make sure that you have backed up your php and web server configuration!

```
#yum remove zabbix-web-3.*
#yum -y install zabbix-web-mysql-scl zabbix-apache-conf-scl
```
UPGRADE THE FRONTEND

- Change the default php configuration

```
php_value[max_execution_time] = 300
php_value[memory_limit] = 128M
php_value[post_max_size] = 16M
php_value[upload_max_filesize] = 2M
php_value[max_input_time] = 300
php_value[max_input-vars] = 10000
php_value[date.timezone] = Europe/Riga
```
IMPORT BACK THE OLD HISTORY DATA

insert into history_log select itemid, clock, timestamp, source, severity, value, logeventid, ns from history_log_old;

insert into history_text select itemid, clock, value, ns from history_text_old;

✓ Can be done with the Zabbix server process running
UPGRADE THE PROXIES (1)

- Upgrade the CentOS proxies

#rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/7/x86_64/zabbix-release-5.0-1.el7.noarch.rpm
#yum clean all
#yum -y upgrade zabbix-proxy-mysql

- Since the proxy uses MariaDB backend DB, the database schema upgrade process is performed automatically on the Zabbix proxy startup.
UPGRADE THE PROXIES (1)

For SQLite3 – remove the DB file and it will be recreated after the upgrade.


19892:20200626:160201.397 Zabbix does not support SQLite3 database upgrade.
UPGRADE THE PROXIES (2)

☑ Proxy runs on the Amazon Linux AMI which uses CentOS 6 packages
☑ No server or proxy Zabbix 5.0 packages are available for CentOS 6

What is the best course of action?

☑ Compile the Zabbix proxy from source
☑ Bring up a new VM with the up to date supported OS and install the proxy from official packages
UPGRADE OR REPLACE THE AGENTS

☑️ Remember, that agents are backwards compatible!
☑️ In 5.0 you have an option to choose between the GO agent or the C agent

```bash
#yum install zabbix-agent
#yum install zabbix-agent2
```
TO DO POST UPGRADE

ENABLING THE FULL POTENTIAL OF THE 5.0
- Verifying the instance integrity
- Performance and configuration tuning
- Implementing the new features
CHECK FOR ANY ERROR MESSAGES - LOGS

Error on rename of './zabbix/items' to './zabbix/#sql2-caf-2f' (errno: 152) [alter table items drop foreign key c_items_1]
3801:20200130:144010.260 database upgrade failed

- Any errors and issues encountered during the upgrade process will be logged to the Zabbix server log file.
- You need to fix the issues and restart the server for the upgrade to continue!
- Most of the time these are caused by implementing custom changes on Zabbix DB tables
CHECK FOR ANY ERROR MESSAGES - LOGS

6448:20200625:175048.726 Zabbix supports only "utf8_bin" collation. Database "zabbix" has default collation "utf8_general_ci"

6448:20200625:175048.734 character set name or collation name that is not supported by Zabbix found in 29 column(s) of database "zabbix"

- The log file will point out the DB schema component which needs to have its collation changed. E.g. – database default collation or column collation (ZBX-17357)
CHECK FOR ANY ERROR MESSAGES - LOGS

6448:20200625:175048.735 database is not upgraded to use double precision values

✔ Upgrade the history tables – Float64 support (ZBXNEXT-5691)
CHECK FOR ANY ERROR MESSAGES - FRONTEHD

System information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zabbix server is running</td>
<td>Yes</td>
<td>localhost:10051</td>
</tr>
<tr>
<td>Number of hosts (enabled/disabled/templates)</td>
<td>138</td>
<td>7 / 0 / 131</td>
</tr>
<tr>
<td>Number of items (enabled/disabled/not supported)</td>
<td>122</td>
<td>117 / 0 / 5</td>
</tr>
<tr>
<td>Number of triggers (enabled/disabled [problem/ok])</td>
<td>61</td>
<td>61 / 0 [3 / 58]</td>
</tr>
<tr>
<td>Number of users (online)</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Required server performance, new values per second</td>
<td>2.64</td>
<td></td>
</tr>
</tbody>
</table>

- Incorrect default charset for Zabbix database: 'latin1' instead 'UTF8'.
- Database history tables upgraded: No

- Fix the collation for the corresponding DB schema component (ZBX-17357)
- Upgrade the history tables – Float64 support (ZBXNEXT-5691)
TEST YOUR **SCRIPTS AND INTEGRATIONS**

- Confirm that all of your global scripts and alert scripts are working
- Verify that your existing integrations are properly sending out notifications

### Test media type "Script"

- **Send to**: test
- **Subject**: Test subject
- **Message**: This is the test message from Zabbix

Test media type "Script"
TEST YOUR **SCRIPTS AND INTEGRATIONS**

- Make sure that your script based items are receiving data

<table>
<thead>
<tr>
<th>Test item</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Get value from host</td>
<td>✅</td>
</tr>
<tr>
<td>Host address</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td></td>
</tr>
<tr>
<td>Proxy</td>
<td>(no proxy)</td>
</tr>
<tr>
<td>Value</td>
<td>1001</td>
</tr>
<tr>
<td>Time</td>
<td>now</td>
</tr>
<tr>
<td>Previous value</td>
<td></td>
</tr>
<tr>
<td>Prev. time</td>
<td></td>
</tr>
<tr>
<td>End of line sequence</td>
<td>LF, CRLF</td>
</tr>
<tr>
<td>Result</td>
<td>Result converted to Numeric (unsigned)</td>
</tr>
<tr>
<td></td>
<td>1001</td>
</tr>
</tbody>
</table>
VERIFY PERFORMANCE AND CONFIGURATION

- Confirm that there’s no significant queue increase post-upgrade
- Make sure that Zabbix server and proxy performance graphs are not showing any performance anomalies
- Check for any slow queries or unexpected error message in the server or proxy log files

Queue overview by proxy

<table>
<thead>
<tr>
<th>Proxy</th>
<th>5 seconds</th>
<th>10 seconds</th>
<th>30 seconds</th>
<th>1 minute</th>
<th>5 minutes</th>
<th>More than 10 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin proxy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Riga proxy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>78</td>
<td>0</td>
</tr>
<tr>
<td>Server</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Total: 3
Optimize the data collection logic

- Apply the new preprocessing rules such as throttling or data validation
- Modify existing items to use the new features, such as ODBC connection string
- Double check your API scripts – some changes were made to API syntax! (For example – details property is now required for SNMP interface type. Might break legacy host.create scripts)
SWITCH FROM SCRIPTS TO WEBHOOKS

- Many new webhook integrations added starting from 4.2
- Official integrations developed and maintained by Zabbix
- No more need for external scripting – just import the XML file and you’re good to go!
IMPLEMENT THE ADDED SECURITY FEATURES

- Communication with the Zabbix database backend can now be encrypted
- Mask your macros!
- Migrate to out of the box SAML support