Zabbix 6.0 will no longer support PHP versions older than 7.2.5. Newer operating systems support PHP 7.2.5 so you can upgrade to 6.0 without any difficulties on these Linux distributions:

* **Ubuntu 22.04 LTS (Jammy Jellyfish)** **/ Ubuntu 20.04 (Focal)**
* **CentOS 8 or 9 / RHEL 8 or 9 / Oracle Linux 8 or 9 / Alma Linux 8 or 9 / Rocky Linux 8 or 9**
* **Debian 11 (Bullseye) / Debian 10 (Buster)**
* **Rasberry Pi OS (Debian 11 / 10**)

In addition to dropping support for older PHP versions, Zabbix 6.0 has tightened requirements for the database. Here are the **minimum** database versions that new Zabbix supports:

* MySQL/Percona 8.0+
* MariaDB 10.5+
* PostgreSQL 13+
* Oracle 19c+
* TimescaleDB 2.0.1+
* SQLite 3.3.5+

Check your database version and if it does not meet the requirements, you will first need to upgrade it before continuing with this guide or use override by setting **AllowUnsupportedDBVersions=1** in Zabbix server configuration file at your own risk.

Note, this guide is for **upgrading Zabbix** while guides for **installing Zabbix 6.0 or 6.2** from scratch on various Linux distribution can be found on these links: [CentOS, RHEL, Oracle/Alma/Rocky Linux](https://bestmonitoringtools.com/how-to-install-zabbix-server-on-centos-or-rhel/), [Ubuntu](https://bestmonitoringtools.com/how-to-install-zabbix-server-on-ubuntu/), [Debian](https://bestmonitoringtools.com/how-to-install-zabbix-server-on-debian/), [Rasbian (Raspberry Pi)](https://bestmonitoringtools.com/how-to-install-zabbix-on-raspberry-pi-raspbian/).

**Step 1: Stop Zabbix Server**

We must first stop the Zabbix server so that the database does not receive any data as we perform the upgrade:

systemctl stop zabbix-server

**Step 2: Backup Zabbix components**

Zabbix upgrade should go smoothly with no problems but there is that famous saying “*better safe than sorry*” or how I like to put it “better backup than spend all day in a linux terminal”.

Therefore, back up your data and copy all relevant Zabbix files to your backup directory! This will give you the option to restore the old version of Zabbix if something goes wrong with the upgrade. The procedure for restoration is described in [Step 9: Restore procedure (if Zabbix upgrade fails)](https://bestmonitoringtools.com/upgrade-zabbix-to-the-latest-version/#Step_9_Restore_procedure_if_Zabbix_upgrade_fails).

**a) Create directories for backup files**

Create backup directories for binary, configuration, doc, web and database files.

mkdir -p /opt/zabbix\_backup/bin\_files /opt/zabbix\_backup/conf\_files /opt/zabbix\_backup/doc\_files

mkdir -p /opt/zabbix\_backup/web\_files /opt/zabbix\_backup/db\_files

**b) Backup Zabbix binary, doc and conf files**

Next, copy Zabbix binary, doc and configuration files. Some files will not exits, depending on what webserver you are using (Apache or NGINX), so I will use “2>/dev/null” to ignore copy errors.

cp -rp /etc/zabbix/zabbix\_server.conf /opt/zabbix\_backup/conf\_files

cp -rp /usr/sbin/zabbix\_server /opt/zabbix\_backup/bin\_files

cp -rp /usr/share/doc/zabbix-\* /opt/zabbix\_backup/doc\_files

cp -rp /etc/httpd/conf.d/zabbix.conf /opt/zabbix\_backup/conf\_files 2>/dev/null

cp -rp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix\_backup/conf\_files 2>/dev/null

cp -rp /etc/zabbix/php-fpm.conf /opt/zabbix\_backup/conf\_files 2>/dev/null

**c) Backup Zabbix web files (frontend)**

Backup Zabbix frontend files.

cp -rp /usr/share/zabbix/ /opt/zabbix\_backup/web\_files

**d) Backup Zabbix database**

Note: This guide is tailored for Zabbix installation in combination with MySQL / MariaDB, and if you are using PostgreSQL, please google how to backup PostgreSQL!

Make sure you have enough free disk space on “/opt” or use another dictionary path for database backup.

This command will make a full backup of the Zabbix database on MySQL/MariaDB instance (change hostname, user, password and db name to match your environment):

mysqldump -h localhost -u'root' -p'rootDBpass' --single-transaction 'zabbix' | gzip > /opt/zabbix\_backup/db\_files/zabbix\_backup.sql.gz

Note that this backup process can take anywhere from few minutes to hours, depending on the database size.

**Step 3: Upgrade Zabbix Server and Frontend**

Select the appropriate OS repository and upgrade the Zabbix server and frontend using the provided instructions.

Note: If using PostgreSQL, replace “mysql” with “pgsql” in the upgrade command.

**a) Ubuntu 22.04 & Ubuntu 20.04**

Please delete old Zabbix repository so that we can install the new one:

dpkg --purge zabbix-release

Upgrade Zabbix and when prompted what to do with Zabbix configuration file, just press enter to keep the current version of “zabbix\_server.conf“:

wget <https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release\_6.0-4+ubuntu$>(lsb\_release -rs)\_all.deb

sudo dpkg -i zabbix-release\_6.0-4+ubuntu$(lsb\_release -rs)\_all.deb

apt update

apt install -y --only-upgrade zabbix-server-mysql zabbix-frontend-php

And if you are using Apache server (httpd) install “zabbix-apache-conf“:

apt-get install -y zabbix-apache-conf

**b) RHEL 8/9 & CentOS 8/9 & Oracle/Alma/Rocky Linux 8/9**

Upgrade Zabbix and when prompted what to do with Zabbix configuration file, just press enter to keep the current version of “zabbix\_server.conf“

rpm -Uvh <https://repo.zabbix.com/zabbix/6.0/rhel/$>(rpm -E %{rhel})/x86\_64/zabbix-release-6.0-4.el$(rpm -E %{rhel}).noarch.rpm

dnf clean all

dnf upgrade -y zabbix-server-mysql zabbix-web-mysql

And if you are using Apache server (httpd) install “zabbix-apache-conf“:

dnf install -y zabbix-apache-conf

**c) Debian 11 & Debian 10**

Please delete old Zabbix repository so that we can install the new one:

dpkg --purge zabbix-release

Upgrade Zabbix and when prompted what to do with Zabbix configuration file, just press enter to keep the current version of “zabbix\_server.conf“:

wget <https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release\_6.0-4+debian$>(cut -d"." -f1 /etc/debian\_version)\_all.deb

dpkg -i zabbix-release\_6.0-4+debian$(cut -d"." -f1 /etc/debian\_version)\_all.deb

apt update

apt install -y --only-upgrade zabbix-server-mysql zabbix-frontend-php

And if you are using Apache server (httpd) install “zabbix-apache-conf“:

apt-get install -y zabbix-apache-conf

**d) Rasbian 11 & Rasbian 10**

Please delete old Zabbix repository so that we can install the new one:

dpkg --purge zabbix-release

Upgrade Zabbix and when prompted for Zabbix configuration file, just press enter to keep the current version of “zabbix\_server.conf“:

wget <https://repo.zabbix.com/zabbix/6.0/raspbian/pool/main/z/zabbix-release/zabbix-release\_6.0-4+debian$>(cut -d"." -f1 /etc/debian\_version)\_all.deb

dpkg -i zabbix-release\_6.0-4+debian$(cut -d"." -f1 /etc/debian\_version)\_all.deb

apt update

apt install -y --only-upgrade zabbix-server-mysql zabbix-frontend-php

And if you are using Apache server (httpd) install “zabbix-apache-conf“:

apt-get install -y zabbix-apache-conf

**Step 4: Start Zabbix service and database upgrade**

Great job! Zabbix server and frontend are successfully upgraded! However, the Zabbix database is still on the old version because we didn’t upgrade it. This is especially evident when connecting to a frontend because you would be greeted with the message “[*The frontend does not match Zabbix database*](https://bestmonitoringtools.com/upgrade-zabbix-to-the-latest-version/#Step_11_Learn_about_common_upgrade_errors)“.

Luckily, upgrade of the Zabbix database is the easy part, just start the Zabbix service and it will automatically do the upgrade:

systemctl start zabbix-server

Database upgrade can last from 1 minute to hours depending on database size. Check the upgrade progress with the command “\*c\*at /var/log/zabbix/zabbix\_server.log | grep database“:

#cat /var/log/zabbix/zabbix\_server.log | grep database

1794:20200408:200607.700 current database version (mandatory/optional): 05040000/05040002

1794:20200408:200607.700 starting automatic database upgrade

1794:20200408:200607.706 completed 1% of database upgrade

1794:20200408:200608.804 completed 10% of database upgrade

.....

1794:20200408:200613.111 completed 98% of database upgrade

1794:20200408:200613.123 completed 100% of database upgrade

1794:20200408:200613.123database upgrade fully completed

1794:20200408:200613.136 database could be upgraded to use primary keys in history tables

Database upgrade is completed successfully when you receive the message “*database upgrade fully completed*” in the Zabbix server log file.

Check out section “[**Learn about common upgrade errors**](https://bestmonitoringtools.com/upgrade-zabbix-to-the-latest-version/#Step_11_Learn_about_common_upgrade_errors)” if you receive error “[Z3005] query failed: [1118] Row size too large“.

**Step 5: Clear browser cache and check Zabbix version**

Zabbix frontend may look pretty weird after the upgrade because you haven’t deleted the cache on your browser! Clear the browser cache and log in to the Zabbix.

And if you still receive the error “*The frontend does not match Zabbix database*” check out [step 11: Learn about common upgrade errors](https://bestmonitoringtools.com/upgrade-zabbix-to-the-latest-version/#Step_11_Learn_about_common_upgrade_errors)

One more thing to do! On the server, check if the upgrade was successful with the command “zabbix\_server -V“:

# zabbix\_server -V

zabbix\_server (Zabbix) 6.0.0

And on the frontend, check if the correct Zabbix version (6.0.x) is displayed on the bottom of the page:

Checking Zabbix version on the frontend

!<https://bestmonitoringtools.com/wp-content/uploads/zabbix_version_check.png>

**Step 6: Database upgrade to primary keys**

As of Zabbix 6.0, primary keys are used for all tables in new installations. Unfortunately there is no automation here, we have to manually upgrade the history tables in existing installations to primary keys.

Before proceeding, make sure to backup your database using [Step 2](https://bestmonitoringtools.com/upgrade-zabbix-to-the-latest-version/#Step_2_Backup_Zabbix_components)!

I will run the demonstration on MariaDB 10.6, and if you are using another type of database (MySQL, PostGreSQL, TimeScaleDB, Oracle) then check out the [official Zabbix guide](https://www.zabbix.com/documentation/current/en/manual/appendix/install/db_primary_keys#mysql).

**a. Create new history tables and rename old ones**

First, install “*zabbix-sql-scripts*” package:

CentOS, REHEL, Oracle/Alma/Rocky Linux

dnf install -y zabbix-sql-scripts

ORUbuntu/Debian/Rasperry Pi OS

apt-get install -y zabbix-sql-scripts

Then run “history\_pk\_prepare.sql” script to rename old history tables and create new ones with the new primary key:

mysql -uroot -prootDBpasszabbix < /usr/share/zabbix-sql-scripts/mysql/history\_pk\_prepare.sql

**b. Migrate data from old history tables to new ones**

Note that this step is optional if you do not need the old data from the history tables (you always have data from the trends tables).

Create a temporary directory to use to export and import history data (delete it after the import):

mkdir /var/lib/mysql-files && chmod 777 /var/lib/mysql-files

Connect to the MySQL/MariaDB server:

mysql -uroot -prootDBpasszabbix

Tweak DB performance to speed up data migration and avoid timeout errors:

SET SESSION SQL\_LOG\_BIN=0;SET SESSION bulk\_insert\_buffer\_size= 1024 \* 1024 \* 256;SET MAX\_STATEMENT\_TIME=0;

Migrate history data from old tables to new ones:

SELECT \* INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n' FROM history\_old;

LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n';

SELECT \* INTO OUTFILE '/var/lib/mysql-files/history\_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n' FROM history\_uint\_old;

LOAD DATA INFILE '/var/lib/mysql-files/history\_uint.csv' IGNORE INTO TABLE history\_uint FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n';

SELECT \* INTO OUTFILE '/var/lib/mysql-files/history\_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n' FROM history\_str\_old;

LOAD DATA INFILE '/var/lib/mysql-files/history\_str.csv' IGNORE INTO TABLE history\_str FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n';

SELECT \* INTO OUTFILE '/var/lib/mysql-files/history\_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n' FROM history\_log\_old;

LOAD DATA INFILE '/var/lib/mysql-files/history\_log.csv' IGNORE INTO TABLE history\_log FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n';

SELECT \* INTO OUTFILE '/var/lib/mysql-files/history\_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n' FROM history\_text\_old;

LOAD DATA INFILE '/var/lib/mysql-files/history\_text.csv' IGNORE INTO TABLE history\_text FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TERMINATED BY '\\n';

If everything works correctly, then delete the old history tables from the Zabbix database:

DROP TABLE history\_old;

DROP TABLE history\_uint\_old;

DROP TABLE history\_str\_old;

DROP TABLE history\_log\_old;

DROP TABLE history\_text\_old;