



Webinar

# Upgrade to Zabbix 6.4

all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

## Zabbix 6.4

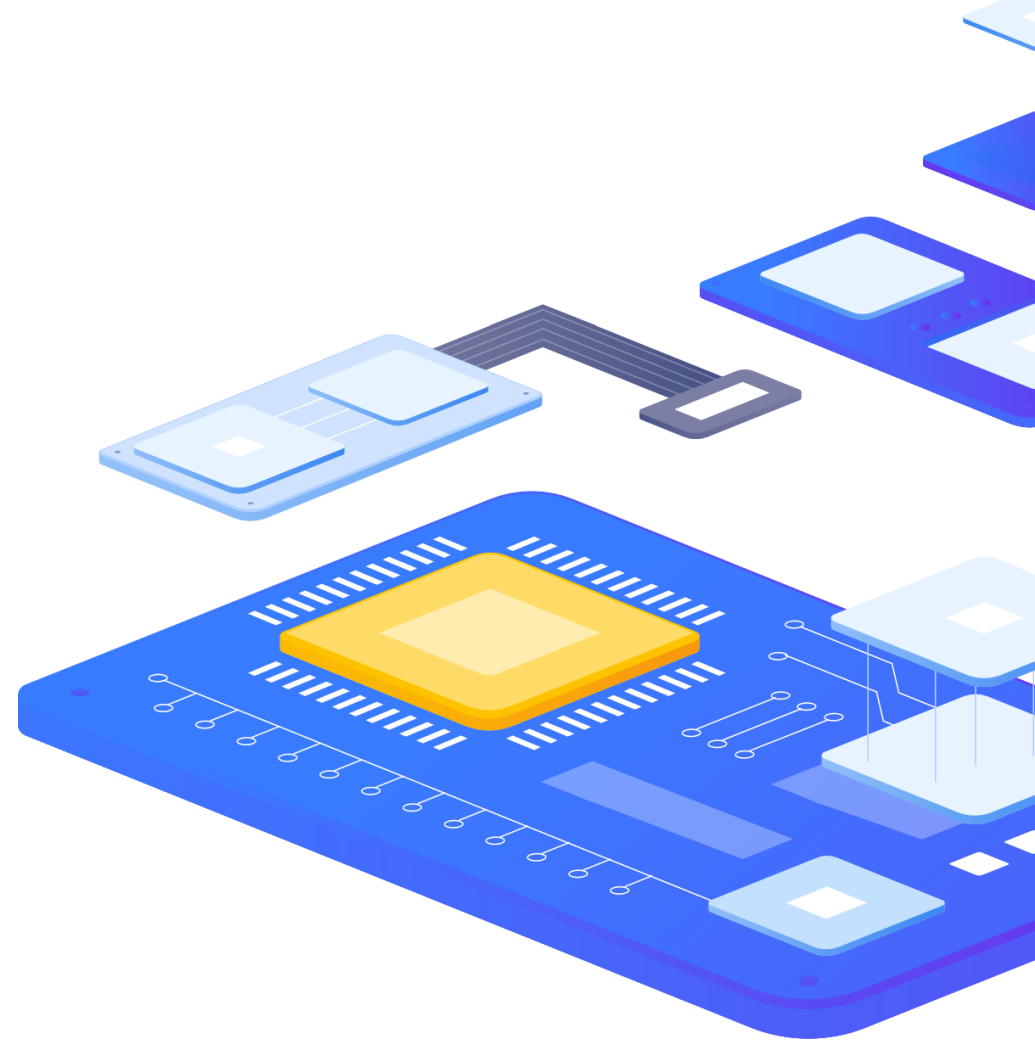
# Zabbix 6.4

Zabbix 6.4 improves the flexibility of multiple existing features, thus supporting many new use cases

Zabbix 6.4 focuses on:

- › User management and provisioning (JIT)
- › Instant propagation of configuration changes across the whole Zabbix infrastructure
- › Performance improvements, especially for large environments
- › Integrating Zabbix with 3rd party systems
- › Improved problem and alert management
- › More streamlined upgrade procedure and more...

Zabbix 6.4 also adds a variety of templates for the most popular vendors and cloud providers.



1

## WHY UPGRADE (Proč upgradovat)



## WHY UPGRADE

# Why upgrade



Support of new functionality and bugfixes



Improved performance and stability



Improved security

## WHY UPGRADE

# New feature support

- › Zabbix 6.0: High availability cluster for Zabbix server
- › Zabbix 6.0: Services
- › Zabbix 6.0: New widgets
- › Zabbix 6.2: Discovered host customization
- › Zabbix 6.2: Active checks availability icon
- › Zabbix 6.2: Separate groups for templates
- › **Zabbix 6.4: Streaming to external systems**
- › **Zabbix 6.4: New menu layout**
- › **Zabbix 6.4: Interface not required for some checks**
- › **Zabbix 6.4: Cause and symptom problems**
- › **Zabbix 6.4: Action log export to CSV**
- › **Zabbix 6.4: Dashboard widgets**

## WHY UPGRADE

# Improved stability and performance

- › Zabbix 6.0: Primary keys
- › Zabbix 6.0: Bulk processing for Prometheus metrics
- › Zabbix 6.0: Separate processing for ODBC checks
- › Zabbix 6.2: Faster configuration sync
- › Zabbix 6.2: User macro cache
- › Zabbix 6.2: Reload proxy configuration
- › **Zabbix 6.4: Optimized proxy configuration update**
- › **Zabbix 6.4: Thread-based preprocessing workers**
- › **Zabbix 6.4: Instant refresh of active checks**
- › **Zabbix 6.4: Optimized SNMP discovery and collection**
- › **Zabbix 6.4: Zabbix server support for older proxies**
- › **Zabbix 6.4: Automated database upgrade on proxies with SQLite**

## WHY UPGRADE

# Improved security

- › Zabbix 6.0: Password requirements
- › Zabbix 6.0: Audit log
- › Zabbix 6.0: Drop unsupported versions of DBs
- › Zabbix 6.2: CyberArk Vault CV2
- › Zabbix 6.2: Multiple LDAP sources
- › Zabbix 6.2: MD5 cryptography has been dropped completely for user passwords
- › **Zabbix 6.4: LDAP/SAML user provisioning**
- › **Zabbix 6.4: Secure password change**
- › **Zabbix 6.4: CSRF tokens**

# 2

Life Cycle  
(životní cyklus)





# Business service monitoring

## ► Release policy:

- **LTS** (Long Term Support) release: **5.0, 6.0, 7.0** – for large environments, long term support, more thoroughly tested release.
- **Standard** release: **6.2, 6.4** – Earlier access to new features.

## ► Major and minor releases:

- **5.4, 6.0, 6.2, 6.4** – Major releases
- **5.4.7, 6.0.5, 6.2.18, 6.4.1** – Minor releases

## LIFE CYCLE

# Life cycle

- › Stable and unstable releases:
  - Even numbers: 6.0, 6.2, 6.4 – Stable releases
  - Odd numbers: 6.3, 6.5 – Unstable (alpha and beta)
- › All the components contained within a single major release of Zabbix are 100% backwards compatible.
- › Web frontend 6.4.5 + Server 6.4.1 + Proxy 6.4.0 + Agent 3.4.5 – OK (Agent 6.4 is recommended for active checks)
- › Best practice is to have the **same version** on Frontend, Server and Proxy.
- › No database changes within a single major release.
- › Agents are backwards compatible: 1.4 – 6.4 agents compatible with 6.X server.
- › Forward compatibility is not guaranteed: 7.0 – 7.2 agents might not be compatible with 6.4 server.

3

Upgrade types  
(typy upgradu)



## UPGRADE TYPES

# Upgrade types

- ▶ Upgrade between **minor** versions:

3.4.7 → 3.4.15

4.0.0 → 4.0.5

- ▶ Upgrade between **major** versions:

4.0 → 6.4

4.4 → 6.4

## UPGRADE TYPES

# Which components to upgrade?

- ▶ The Zabbix **minor** upgrade procedure is very simple.
- ▶ You can upgrade a single or multiple components of Zabbix (for example, web frontend, proxy, Zabbix server).
- ▶ Zabbix agent upgrade is not mandatory.

## UPGRADE TYPES

# Which components to upgrade?

- ▶ Zabbix major upgrade procedure is more complicated and requires prior preparation.
- ▶ Upgrade of **all** Zabbix components is required (Zabbix server, database, web frontend, proxy, Java Gateway, Web Service).
- ▶ Zabbix agent upgrade is not mandatory. (Agent 6.4 is recommended for active checks)

## UPGRADE TYPES

# Minor upgrade procedure

### Read the release notes!

#### Minor version upgrade:

- › Make a backup copy (just in case)
- › Upgrade the binary files and the web frontend files, restart the daemons

```
$ dnf upgrade 'zabbix-*
```

No changes in the database (6.4.0 → 6.4.x) - not true from 6.0.11 when is backported some functionality from 6.2

## UPGRADE TYPES

# Minor upgrade procedure

Fix the performance issues (if you have any).

Otherwise the upgrade could take a considerable amount of time.

**Read the release notes!**



## UPGRADE TYPES

# Major upgrade procedure

### Major version upgrade:

- › Make a backup copy
- › Stop the Zabbix server
- › Upgrade the binary files and the web frontend
- › Start the Zabbix server

Automatic database upgrade (2.2.0)

Cumulative updates (2.2 → 6.4)

**Update rollback is not supported**

# 4

## Upgrade Notes (poznámky k verzím)



## UPGRADE NOTES

# Upgrade notes 4.0 > 4.2

- ▶ Preprocessing support on Zabbix proxy.
- ▶ PHP GD library minimum version has been upped from 2.0 to 2.0.28 to support animated GIFs in maps.
- ▶ JSON returned by low-level discovery rules has been changed.

## UPGRADE NOTES

# Upgrade notes 4.2 > 4.4

- › Jabber, Ez Texting media types removed.
- › Real-time export protocol changed.

## UPGRADE NOTES

# Upgrade notes 4.4 > 5.0

- › Minimum required PHP version has been upped from 5.4 to 7.2
- › Check minimal DB version.
- › Support of IBM DB2 dropped.
- › Support of Internet Explorer 11 dropped.
- › Fix all issue in **System information**

## UPGRADE NOTES

# Upgrade notes 5.0 > 5.2

- ▶ Minimum required PHP version has been upped from 7.2 to 7.2.5
- ▶ Centos 6,7 and RedHat 6,7 - Only Agent packages (!!!NOW!!!)

## UPGRADE NOTES

# Upgrade notes 5.2 > 5.4

- › New package for scheduled PDF reports
- › Check minimal DB version.
- › API changes
- › New expression syntax for triggers and calculated items
- › Centos 6,7 and RedHat 6,7 - Only Agent packages (!!!NOW!!!)

## UPGRADE NOTES

# Upgrade notes 5.4 > 6.0

- › Check minimal DB version!
- › API changes
- › Primary keys
- › Centos 8 – please upgrade to supported OS
- › Support for PCRE2 has been added
- › Old audit records will not be preserved
- › Simple macros replaced by expression macros
- › Positional macros no longer supported
- › User macros in item name no longer supported
- › Separate processing for ODBC checks
- › Check items scripts for procedures renaming



## UPGRADE NOTES

# Upgrade notes 6.0 > 6.2

- › Check minimal DB version
- › Check minimal PHP version
- › DB triggers
- › Primary keys
- › User passwords in MD5 are not supported anymore
- › Zabbix agent 6.2 – Active check icon
- › CurlHttpRequest removed
- › Templates

## UPGRADE NOTES

# Upgrade notes 6.2 > 6.4

- › Check minimal DB version
- › Check minimal PHP version
- › It is recommended to use PHP 8.0 or newer, because PHP 7.4 is no longer supported by the vendor.
- › Zabbix agent 6.4 – Active check
- › The minimum required libssh version has been raised from 0.6.0 to 0.9.0
- › For SUSE Linux Enterprise Server (SLES) 15 the minimum supported service pack version is now SP4
- › DB triggers
- › Automated database upgrade on proxies with SQLite
- › **Old numeric (float) value type is deprecated**
- › **You can use outdated proxy in same LTS version for example 6.0 and 6.2**
- › ConfigFrequency parameter for active proxies is now deprecated

instead the ProxyConfigFrequency parameter must be used on both server (for passive proxies) and proxy (for active proxies). Note that if both ProxyConfigFrequency and ConfigFrequency are used, the proxy will log an error and terminate

## UPGRADE NOTES

# Major upgrade: how NOT to perform an upgrade

- › DO NOT read the release notes.
- › DO NOT test the new version.
- › DO NOT make a backup copy.

```
$ dnf upgrade 'zabbix-*
```

## UPGRADE NOTES

# Why is "yum upgrade" bad?

*Zabbix User:* We have an issue, the database size is growing really fast after upgrading from 1.8 to 3.0.

*Zabbix Team:* The housekeeper has been disabled after the upgrade!

*Zabbix User:* Java monitoring has stopped working after upgrading from 3.2 to 4.0.

*Zabbix Team:* Java Gateway protocol has been changed, but Java Gateway has not been upgraded.

*Zabbix User:* Zabbix restart takes too much time(Large amount of slow queries "select clock,ns,value from history").

*Zabbix Team:* MySQL parameter "index\_condition\_pushdown is enabled" (ZBX-10652).

*Zabbix User:* Upgrade to the latest version took us 6 hours to perform (Downtime).

*Zabbix Team:* There have been changes to large table structure.

## UPGRADE NOTES

# Why do such situations occur?

### **No knowledge about:**

- › changes,
- › required upgrade steps.

Larger Zabbix instance => more time to upgrade.

### **Depends on:**

- › Zabbix version,
- › database size,
- › underlying hardware.
- › Optional DB patches

## UPGRADE NOTES

# How to avoid such situations?

- › Read the release notes.
- › Test new version before upgrade.
- › Make a backup copy.
- › Keep custom changes to minimum.
- › Don't make changes to database schema (partitioning).

## UPGRADE NOTES

# How to plan the upgrade?

- › Type of upgrade (minor – major).
- › Changes in the structure of large tables (history, trends, events).
- › Zabbix database size and performance.
- › Complexity of the infrastructure (partitioning, High Availability cluster, replication).
- › Customization (DB schema, custom PHP pages, patches, integrations, scripts, modules).

## UPGRADE NOTES

# What causes errors?

### **Changes in the DB schema:**

- › Removed or additional indexes (for example, with the goal to increase the performance).
- › Partitioning misconfiguration.
- › DB config misconfiguration.



## UPGRADE NOTES

# What affects the upgrade duration?

- ▶ Zabbix DB size and performance.
- ▶ Changes in the structure of large tables (history\*, trends\*, events).

## UPGRADE NOTES

# How to identify changes in large DB tables?

Read the release notes.

[Click here for example](#)

*history\_text.id and history\_log.id fields will be removed from the corresponding history tables during database upgrade. Depending on the history table size this process can be slow.*

## UPGRADE NOTES

# How to accelerate the upgrade in case of a large DB?

### **Deletion of indexes for large tables can take hours (*history\_text/log*).**

To accelerate Zabbix upgrade from hours to minutes - use temporary tables. Example for *history\_text*:

- › Create a temporary table (for example, *history\_text\_tmp*).
- › Rename the existing table to *history\_text\_old*.
- › Rename the new table *history\_text\_tmp* to *history\_text*.
- › Upgrade Zabbix (All the changes will be applied to empty table).
- › Copy the data from the old table *history\_text\_old* to the new table *history\_text*.
- › Delete the old table *history\_text\_old*.

5

Upgrade



## UPGRADE

# Major upgrade: correct upgrade procedure

### Example N° 1:

- › Zabbix Upgrade: 4.0 → 6.4 (without proxy)
- › DB size: 70GB (No customization or complexity)
- › NVPS: 200
- › Acceptable downtime: Up to 2 hours

## UPGRADE

# Major upgrade: example №1

1. Read the [release notes](#) and take note of the important changes:
2. Make a backup copy of the configuration files, PHP files and scripts, etc. Example:

```
$ cp /etc/zabbix/zabbix_server.conf /<backup directory>/ # Zabbix server config file
$ cp /etc/zabbix/zabbix_agentd.conf /<backup directory>/ # Zabbix agent config file
$ cp /usr/share/zabbix/alertscripts/* /<backup directory>/ # Alert scripts
$ cp /usr/share/zabbix/externalscripts/* /<backup directory>/ # External scripts
$ cp -R /usr/share/zabbix/ /<backup directory>/ # Web frontend PHP files
$ cp /etc/httpd/conf/httpd.conf /<backup directory>/ # Apache config. files
$ cp /etc/httpd/conf.d/zabbix.conf /<backup directory>/ # Zabbix/PHP parameters
$ cp /etc/zabbix/web/zabbix.conf.php /<backup directory>/ # Zabbix frontend parameters.
```

## UPGRADE

# Major upgrade: example №1

### 3. Make a backup copy of Zabbix DB

```
$ screen -S backup  
$ mysqldump -uroot -p zabbix --extended-insert --single-transaction | bzip2 -9 > /<backup  
directory>/zabbix_db_backup-$(date "+%Y-%m-%d_%H.%M.%S").sql.bz2
```

### 4. Stop Zabbix processes and update the packages

```
$ systemctl stop zabbix-server  
$ rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/8/x86\_64/zabbix-release-6.4-1.el8.noarch.rpm  
$ dnf upgrade zabbix*
```

## UPGRADE

# Major upgrade: example №1

5. Start Zabbix server 6.4 process

```
$ systemctl start zabbix-server
```

6. Stop Zabbix processes and update the packages

```
$ tail -f /var/log/zabbix/zabbix_server.log  
3152:20190308:175043.966 completed 100% of database upgrade  
3152:20190308:175043.966 database upgrade fully completed
```



## UPGRADE

# Major upgrade: correct upgrade procedure

### Example N° 2:

- › Zabbix Upgrade: 5.0 → 6.4 with 20 proxy servers
- › DB size: 7 TB (partitioning)
- › NVPS: 10000
- › Acceptable downtime: Up to 10 minutes
- › Customized PHP pages, scripts and modules

## UPGRADE

# Major upgrade: example №2

1. Read the [release notes](#) and take note of the important changes:
2. Create test environment to test compatibility of:

Type of customization	problem	Source of problems
Integrations and PHP pages	Incompatible	API changes
Scripts	Incompatible Errors	exit code checks (data collection, alerts, user parameters, etc.)
modules	won't work cause errors	header changes and data size in request/response messages between zabbix components

## UPGRADE

# Major upgrade: example №2

3. Check the performance and optimize the DB settings (If necessary). Slow DB = Slow Upgrade!

4. Removal of internal events to accelerate the upgrade (changes in the events table 3.4>6.4)

```
mysql> DELETE from events where source > 0 limit 10000;
```

5. Compare the Zabbix DB schema with the official one (for example, by using MySQL compare or a similar application). You need to understand if there are any customizations and if they can cause errors during the upgrade.

6. Create a backup copy of the configuration tables

```
$ mysqldump -u<user> -p <database> --ignore-table=zabbix.history \  
--ignore-table=zabbix.history_uint --ignore-table=zabbix.history_str \  
--ignore-table=zabbix.history_log --ignore-table=zabbix.history_text \  
--ignore-table=zabbix.trends --ignore-table=zabbix.trends_uint > zabbix_config_dump.sql
```

## UPGRADE

# Major upgrade: example №2

7. Create a test database and import the initial schema/data into the Zabbix test environment.

8. Import the backup configuration table copy to the test environment

```
$ mysql -uzabbix -p zabbix_test < config_dump.sql
```

9. Perform the upgrade on test environment to obtain information regarding the upgrade duration and possible errors during the upgrade

```
$ systemctl stop zabbix-server  
$ rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/8/x86\_64/zabbix-release-6.4-1.el8.noarch.rpm  
$ dnf upgrade zabbix-*  
$ systemctl start zabbix-server
```

10. Solve the issues detected during the test upgrade (if any were encountered)

## UPGRADE

# Major upgrade: example №2

Continue with the production environment upgrade only if the test upgrade was performed successfully!

11. Make a backup copy of the configuration files, PHP files and scripts, etc. Example:

```
$ cp /etc/zabbix/zabbix_server.conf /<backup directory>/ # Zabbix server config file
$ cp /etc/zabbix/zabbix_agentd.conf /<backup directory>/ # Zabbix agent config file
$ cp /usr/share/zabbix/alertscripts/* /<backup directory>/ # Alert scripts
$ cp /usr/share/zabbix/externalscripts/* /<backup directory>/ # External scripts
$ cp -R /usr/share/zabbix/ /<backup directory>/ # Web frontend PHP files
$ cp /etc/httpd/conf/httpd.conf /<backup directory>/ # Apache config. files
$ cp /etc/httpd/conf.d/zabbix.conf /<backup directory>/ # Zabbix/PHP parameters
$ cp /etc/zabbix/web/zabbix.conf.php /<backup directory>/ # Zabbix frontend config file
```

## UPGRADE

# Major upgrade: example №2

12. Make a backup copy of the Zabbix database

```
$ screen -S backup  
$ innobackupex -p<password> /<backup directory>/  
$ innobackupex -p<password> --apply-log --use-memory=4G /<backup directory>/`date`/
```

13. Temporarily disable all of the defined Actions, to avoid false notifications

```
$ systemctl stop zabbix-server  
$ rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/8/x86\_64/zabbix-release-6.4-1.el8.noarch.rpm  
$ dnf upgrade zabbix-*
```

14. Stop Zabbix processes and update the packages

## UPGRADE

# Major upgrade: example №2

15. Create temporary tables for history\_text and history\_log to avoid prolonged downtimes in the case of upgrading from version 3.

16. Start Zabbix server 6.4 process

```
$ systemctl start zabbix-server
```

17. Follow the database upgrade log entries

```
$ tail -f /var/log/zabbix/zabbix_server.log
```

## UPGRADE

# Major upgrade: example №2

18. Check the Zabbix server performance after the upgrade (log file, unsupported items, queue, scripts)

19. Stop the proxy and delete the DB file in the case when SQLite DB is used (SQLite DB file upgrade is not supported)

```
$ systemctl stop zabbix-proxy  
$ rm /tmp/zabbix_proxy.db
```

20. Upgrade all of the proxy servers

```
$ rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/8/x86\_64/zabbix-release-6.4-1.el8.noarch.rpm  
$ dnf upgrade zabbix-proxy*  
$ systemctl start zabbix-proxy
```



## UPGRADE

# Major upgrade: example №2

21. Confirm that the proxies are running:

- “Last seen” in Administration > Proxies
- values and graphs from: “Template App Zabbix Proxy”

22. Enable Actions

23. Make sure [latest «health» templates](#) have been assigned to server/proxy

(check the major version number at the end of link to fit with your version)

24. Upgrade Zabbix agents to obtain new functionality

## UPGRADE

# What to do if "database upgrade failed"?

- ▶ DON'T be hasty with restoring the backup DB copy!
- ▶ Find the error message in the Zabbix server log file:

```
6111:20190308:162806.987 starting automatic database upgrade
6111:20190308:162806.987 [Z3005] query failed: [1091] Can't DROP 'dservices_1'; check that
column/key exists [drop index dservices_1 on dservices]
6111:20190308:162806.987 database upgrade failed
```

- ▶ Find and eliminate root cause of the error (for example, recreate the missing index).
- ▶ Run the Zabbix Server again to continue the upgrade process

6

Demonstration  
(ukázka)



## DEMONSTRATION

# Demonstration

```
› systemctl stop zabbix-server
› rpm -Uvh https://repo.zabbix.com/zabbix/6.4/rhel/8/x86_64/zabbix-release-6.4-1.el8.noarch.rpm
› dnf clean all
› dnf upgrade zabbix* -y
› dnf install zabbix-web-service zabbix-apache-conf -y
› grep -v "^#" /etc/zabbix/zabbix_server.conf | grep -v "^$"
› cd /etc/zabbix && mv zabbix_server.conf zabbix_server.conf_backup && mv
  zabbix_server.conf.rpmnew zabbix_server.conf
› nano /etc/zabbix/zabbix_server.conf
› ...
› DBHost=127.0.0.1
› DBPassword=password
› StartReportWriters=1
› WebServiceURL=http://localhost:10053/report
› ...
› systemctl enable zabbix-web-service --now
› systemctl start zabbix-server.service
› tail -f /var/log/zabbix/zabbix_server.log
› systemctl stop zabbix-server.service
```

## DEMONSTRATION

# Demonstration

- › `systemctl stop postgresql-11`
- › `dnf install -y postgresql15-server`
- › `/usr/pgsql-15/bin/postgresql-15-setup initdb`
- › `su - postgres`
- › `/usr/pgsql-15/bin/pg_upgrade -b /usr/pgsql-11/bin -B /usr/pgsql-15/bin -d /var/lib/pgsql/11/data -D /var/lib/pgsql/15/data -k`
- › `logout`
- › `systemctl disable postgresql-11.service`
- › `cat /var/lib/pgsql/11/data/pg_hba.conf > /var/lib/pgsql/15/data/pg_hba.conf`
- › **#It is also recommended also /var/lib/pgsql/15/data/postgresql.conf**
- › `systemctl enable postgresql-15.service --now`
- › `systemctl start zabbix-server.service && tail -f /var/log/zabbix/zabbix_server.log`
- › `su - postgres`
- › `/usr/pgsql-15/bin/vacuumdb --all --analyze-in-stages`
- › `reindexdb`
- › `./delete_old_cluster.sh`
- › `rm -rf 11 delete_old_cluster.sh`

## DEMONSTRATION

# Demonstration

- › `logout`
- › `dnf remove postgresql11-*`
- › `rm -rf /usr/pgsql-11/`
- › `dnf install zabbix-sql-scripts`
- › `systemctl stop zabbix-server.service`
- › `cat /usr/share/zabbix-sql-scripts/postgresql/double.sql | sudo -u zabbix psql zabbix`
- › `cat /usr/share/zabbix-sql-scripts/postgresql/history_pk_prepare.sql | sudo -u zabbix psql zabbix`
- › [https://www.zabbix.com/documentation/6.4/en/manual/appendix/install/db\\_primary\\_keys](https://www.zabbix.com/documentation/6.4/en/manual/appendix/install/db_primary_keys)
- › #Recommendation is use offline version
- › `sudo -u zabbix psql zabbix`
- › `systemctl start zabbix-server && less /var/log/zabbix/zabbix_server.log`

## DEMONSTRATION

# Demonstration

- › `dnf module switch-to php:8.0 -y`
- › `sudo -u zabbix psql zabbix`
- › `UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBwPmrS'`  
`WHERE username = 'Admin';`
- › `mv /etc/zabbix/web/zabbix.conf.php /etc/zabbix/web/zabbix.conf.php_backup`
- › `touch /etc/zabbix/web/zabbix.conf.php`
- › `chown apache:apache /etc/zabbix/web/zabbix.conf.php`
- › `rm -rf /etc/zabbix/web/zabbix.conf.php_backup`
- › `sudo -u zabbix psql zabbix`
- › `DROP TABLE history_old;`
- › `DROP TABLE history_uint_old;`
- › `DROP TABLE history_str_old;`
- › `DROP TABLE history_log_old;`
- › `DROP TABLE history_text_old;`

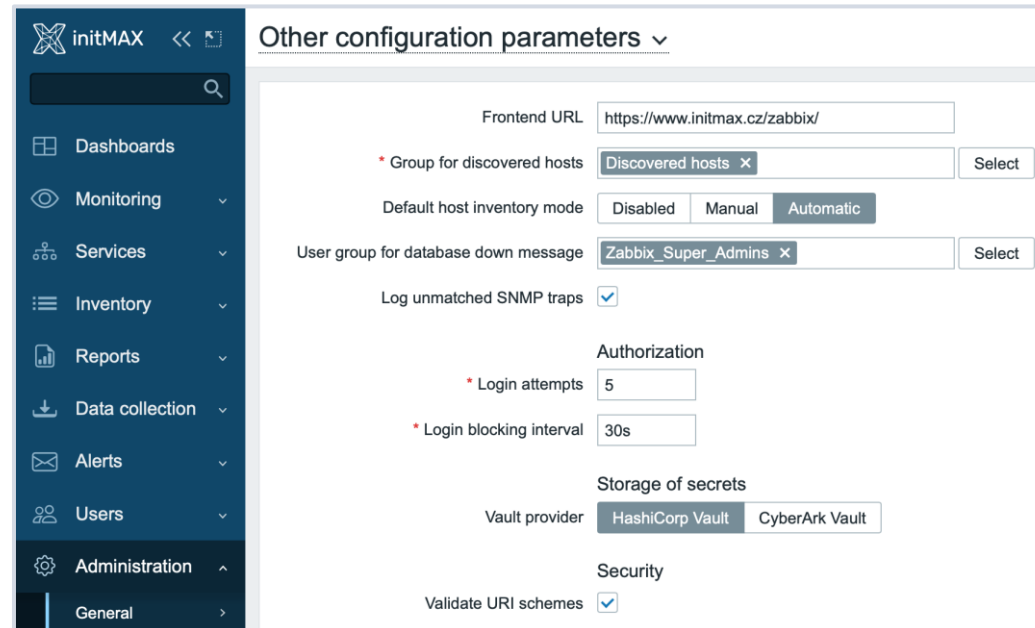
## DEMONSTRATION

# Demonstration commands

## Frontend settings

► Fill your: **Frontend URL**

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm
dnf localinstall google-chrome-stable_current_x86_64.rpm
```



The screenshot shows the 'Other configuration parameters' section of the initMAX web interface. The left sidebar contains a navigation menu with items: Dashboards, Monitoring, Services, Inventory, Reports, Data collection, Alerts, Users, and Administration (expanded). The main content area is titled 'Other configuration parameters' and contains the following settings:

- Frontend URL:
- \* Group for discovered hosts:
- Default host inventory mode: ☐ Disabled ☐ Manual ☒ Automatic
- User group for database down message:
- Log unmatched SNMP traps: ☒
- Authorization:
  - \* Login attempts:
  - \* Login blocking interval:
- Storage of secrets:
  - Vault provider: ☒ HashiCorp Vault ☐ CyberArk Vault
- Security:
  - Validate URI schemes: ☒



## DEMONSTRATION

# Demonstration – optional steps

Install timescaleDB

- › `systemctl stop zabbix-server.service`  
<https://docs.timescale.com/install/latest/self-hosted/installation-linux/>
- › `yum install timescaledb-2-postgresql-15`
- › `timescaledb-tune --pg-config=/usr/postgresql-15/bin/pg_config --max-conns=100`
- › `systemctl restart postgresql-15.service`
- › Setup TimescaleDB in Zabbix  
<https://www.zabbix.com/documentation/6.4/en/manual/appendix/install/timescaledb>
- › `echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix`
- › `cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb.sql | sudo -u zabbix psql zabbix`
- › #Depend on Zabbix server version
- › `nano /etc/zabbix/zabbix_server.conf`
- › ...
- › `AllowUnsupportedDBVersions=1`
- › ...
- › `systemctl start zabbix-server.service && tail -f /var/log/zabbix/zabbix_server.log`

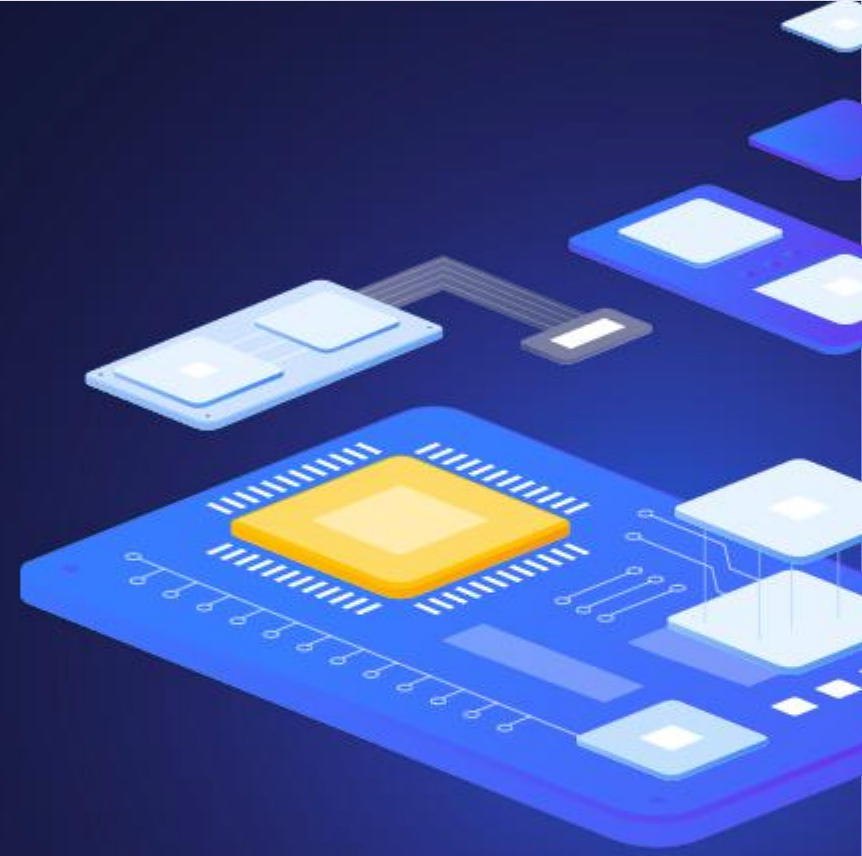
7

DEMO



8

Questions?



UPGRADE



# Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



[tomas.hermanek@initmax.cz](mailto:tomas.hermanek@initmax.cz)

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184