



Webinar

Upgrade to Zabbix 6.0

all our microphones are muted

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

use Chat for discussion, networking or applause

Zabbix 6.0 LTS

Many of Zabbix 6.0 LTS features are focused on solving complex enterprise-level requirements.

Zabbix 6.0 focuses on:

- › Solving enterprise level security and redundancy requirements
- › Improving performance for large Zabbix instances
- › Providing additional value to different types of Zabbix users – DevOps and ITOps teams, Business process owner, Managers
- › Further extending Zabbix monitoring and data collection capabilities
- › Continued delivery of official integrations with 3rd party systems





WHY UPGRADE
(Proč upgradovat)



Why upgrade

New feature support

- › Zabbix 5.2: Script item type
- › Zabbix 5.2: MQTT
- › Zabbix 5.2: Active agent checks from multiple hosts
- › Zabbix 5.4: Scheduled PDF reports
- › Zabbix 5.4: Pre-defined views for Monitoring->Problems
- › Zabbix 5.4: Tags for items
- › Zabbix 6.0: High availability cluster for Zabbix server
- › Zabbix 6.0: Services
- › Zabbix 6.0: New widgets

Why upgrade

Improved stability and performance

- › Zabbix 5.2: Refreshing unsupported items
- › Zabbix 5.2: Template screens converted to dashboards
- › Zabbix 5.2: Custom handling for item error
- › Zabbix 5.4: Direct connections to database removed from pollers
- › Zabbix 5.4: Host availability on the interface level
- › Zabbix 5.4: Trend function cache
- › Zabbix 6.0: Primary keys
- › Zabbix 6.0: Bulk processing for Prometheus metrics
- › Zabbix 6.0: Separate processing for ODBC checks

Why upgrade

Improved security

- › Zabbix 5.2: User roles
- › Zabbix 5.2: Storage of secrets (HashiCorp)
- › Zabbix 5.2: Digest authentication for HTTP agent
- › Zabbix 5.4: API tokens
- › Zabbix 5.4: Hidden PSK data for hosts and proxies
- › Zabbix 5.4: Strong encryption protocols for SNMPv3
- › Zabbix 6.0: Password requirements
- › Zabbix 6.0: Audit log
- › Zabbix 6.0: Drop unsupported versions of DBs



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:

- **5.2, 5.4, 6.0, 6.2** - Major releases
- **5.2.7, 5.4.5, 6.0.18, 6.2.1** - Minor releases

Life cycle

Life cycle

- › Stable and unstable releases:
 - Even numbers: 5.4, 6.0, 6.2 – 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 6.0.5 + Server 6.0.1 + Proxy 6.0.0 + Agent 3.4.5 – OK
- › 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 - 6.0 agents compatible with 6.X server.
- › **Forward compatibility not guaranteed:** 6.2 – 6.4 agents might not be compatible with 6.0 server.



Which version to choose?

6.0 LTS





upgrade types
(typy upgradu)



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

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!

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 → 6.0)

Update rollback is **not supported**



upgrade Notes
(poznámky k verzím)



upgrade Notes

Upgrade notes 2.0 > 2.2

- › Housekeeper is disabled by default.
- › Column/Index changes in the events table.
- › Case-sensitive database.

upgrade Notes

Upgrade notes 2.2 > 2.4

- › Node-based distributed monitoring removed

upgrade Notes

Upgrade notes 2.4 > 3.0

- › Node-based distributed monitoring removed

Upgrade notes 3.0 > 3.2

- › The *history_text.id* and *history_log.id* fields will be removed from the corresponding history tables during database upgrade.

upgrade Notes

Upgrade notes 3.2 > 3.4

- › Command/script execution changes.
- › Java gateway protocol changes.
- › PCRE (Perl Compatible Regular Expressions).
- › Exit code check for user parameters, remote commands and `system.run[]` as well as Zabbix server executed scripts.

upgrade Notes

Upgrade notes 3.4 > 4.0

- › Column changes in the events table.
- › Web-browser cookie/cache cleanup.
- › Agents older than version 1.4 are no longer supported.

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

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.

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.

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).

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).

what causes errors?

Changes in the DB schema:

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

What affects the upgrade duration?

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

How to identify changes in large DB tables?

Read the release notes. For example:

https://www.zabbix.com/documentation/3.2/manual/installation/upgrade_notes_320

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.

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*.



upgrade



Upgrade

Major upgrade: correct upgrade procedure

Example N° 1:

- › Zabbix Upgrade: 4.0 → 6.0 (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:

https://www.zabbix.com/documentation/current/en/manual/installation/upgrade/packages/rhel_centos

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/6.0/rhel/8/x86\_64/zabbix-release-6.0-1.el8.noarch.rpm  
$ dnf upgrade zabbix*
```

Upgrade

Major upgrade: example №1

5. Start Zabbix server 6.0 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 → 6.0 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:

https://www.zabbix.com/documentation/current/en/manual/installation/upgrade/packages/rhel_centos

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 No2

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>5.2)

```
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 No2

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/6.0/rhel/8/x86\_64/zabbix-release-6.0-1.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 No2

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/6.0/rhel/8/x86\_64/zabbix-release-6.0-1.el8.noarch.rpm  
$ dnf upgrade zabbix-*
```

Upgrade

Major upgrade: example No2

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 5.4 process

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

17. Follow the database upgrade log entries

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

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
```

Upgrade

Major upgrade: example №2

20. Upgrade all of the proxy servers

```
$ rpm -Uvh https://repo.zabbix.com/zabbix/6.0/rhel/8/x86\_64/zabbix-release-6.0-1.el8.noarch.rpm
$ dnf upgrade zabbix-proxy*
$ systemctl start zabbix-proxy
```

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

<https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates>

(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



Demonstration
(ukázka)



Demonstration

- › `php-fpm -v`
- › `mysql -V`
- › `cat /etc/centos-release`
- › `curl https://raw.githubusercontent.com/rocky-linux/rocky-tools/main/migrate2rocky/migrate2rocky.sh -o migrate2rocky.sh`
- › `chmod u+x migrate2rocky.sh`
- › `./migrate2rocky.sh -r`

Demonstration

- › `systemctl stop zabbix-server`
- › `rpm -Uvh https://repo.zabbix.com/zabbix/6.0/rhel/8/x86_64/zabbix-release-6.0-1.el8.noarch.rpm`
- › `dnf clean all`
- › `dnf install zabbix-web-service`
- › `nano /etc/zabbix/zabbix_server.conf`
- › ...
- › `StartReportWriters=1`
- › `WebServiceURL=http://localhost:10053/report`
- › ...
- › `systemctl enable zabbix-web-service --now`
- › `dnf upgrade zabbix*`
- › `systemctl start zabbix-server.service && tail -f /var/log/zabbix/zabbix_server.log`
- › `systemctl stop zabbix-server.service`
- › `rpm -i https://repo.mysql.com//mysql80-community-release-el8-3.noarch.rpm`
- › `dnf clean all`
- › `cat /etc/yum.repos.d/mysql-community.repo.rpmnew > /etc/yum.repos.d/mysql-community.repo`
- › `systemctl stop mysqld.service`
- › `dnf upgrade`
- › `systemctl start mysqld.service`
- › `mysql -V`
- › `systemctl start zabbix-server.service && tail -f /var/log/zabbix/zabbix_server.log`
- › `systemctl stop zabbix-server.service`

Demonstration

- › https://www.zabbix.com/documentation/6.0/en/manual/appendix/install/db_primary_keys
- › `dnf install zabbix-sql-scripts`
- › `mysql -uzabbix -p zabbix < /usr/share/doc/zabbix-sql-scripts/mysql/history_pk_prepare.sql`
- › `yum install mysql-shell`
- › `mysql -p -e "set global local_infile=ON;"`
- › `mysqlsh --mysql -u root -h localhost -Dzabbix`
- › `systemctl start zabbix-server.service`
- › `less /var/log/zabbix/zabbix_server.log`

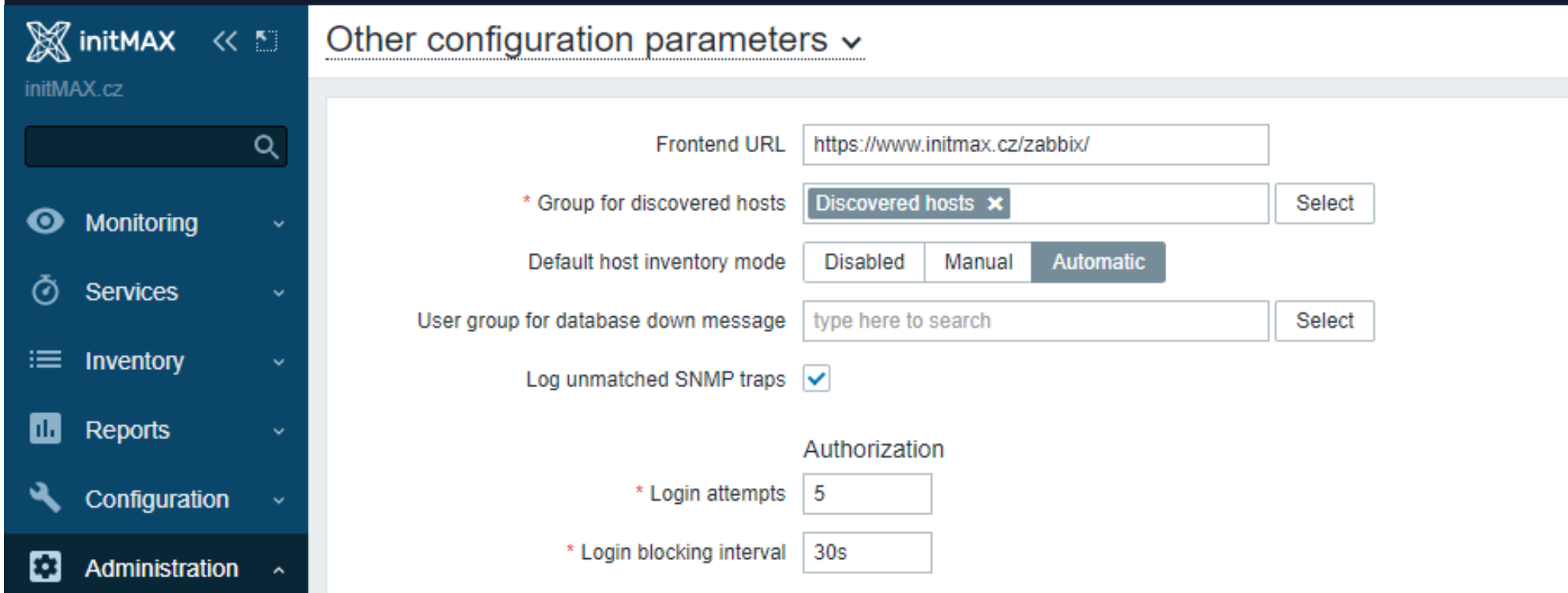
demonstration

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 initMAX web interface. On the left is a dark blue sidebar with the initMAX logo and navigation menu items: Monitoring, Services, Inventory, Reports, Configuration, and Administration. The main content area is titled 'Other configuration parameters' and contains several 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 section:
 - * Login attempts:
 - * Login blocking interval:



DEMO





Questions?



CONTACT US:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



tomas.hermanek@initmax.cz

LinkedIn:



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

Twitter:



<https://twitter.com/initmax1>

Tomáš Heřmánek:



+420 732 447 184