



Webinar

Zabbix best practices

Architecture • Installation • Operation • Maintenance

all your microphones are muted

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

use Chat for discussion, networking or applause

Agenda

1 ARCHITECTURE & HIGH AVAILABILITY

2 INSTALLATION & VERSION MANAGEMENT

3 SECURITY & ENCRYPTED COMMUNICATION

4 IDENTITY & ACCESS MANAGEMENT

5 HOST GROUPS, TAGS & TEMPLATES

6 PERFORMANCE & TUNING

7 MAINTENANCE & OPERATIONS

1

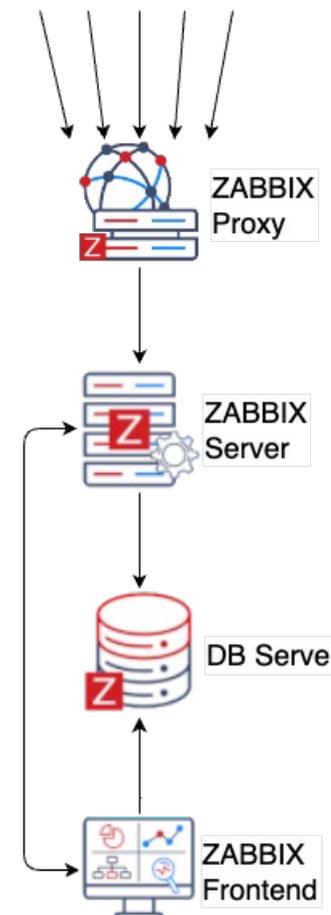


ARCHITECTURE & HIGH AVAILABILITY

Eliminating SPOF • DB Cluster • Proxy Groups • Active-Active

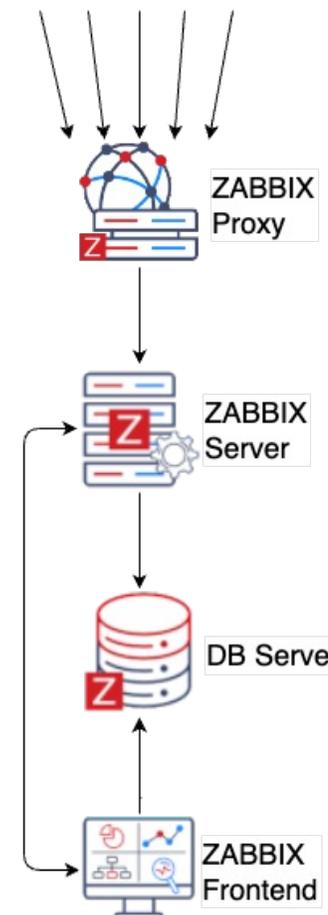
Architecture – Why separate components

- ▶ **Zabbix Server**
 - ▶ CPU&Memory-intensive – evaluation of triggers, calculations, preprocessing
- ▶ **Database**
 - ▶ I/O-intensive – reading/writing history and trends
- ▶ **Frontend**
 - ▶ Memory-intensive – user sessions, dashboard queries against DB
- ▶ **Proxies reduce load on the Server**
 - ▶ Collect data from agents
 - ▶ Run preprocessing before sending results to Server
- ▶ **Philosophy: Server orchestrates. Proxies collect and process data.**



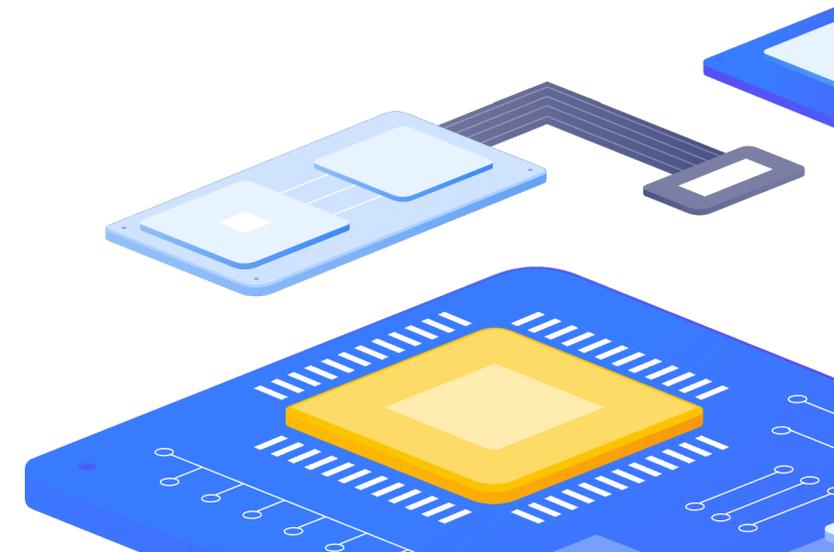
Architecture – When to think about separation

- ▶ Single server is viable for smaller setups (< 1000 NVPS)
- ▶ Separation becomes necessary when environment grows
 - ▶ More hosts and items = more DB I/O and Server CPU pressure
 - ▶ More users pulling history data into dashboards = additional DB read load
- ▶ You don't need to separate everything upfront
 - ▶ Start with DB first – Server and Frontend can stay together
- ▶ Signs it's time
 - ▶ High CPU and memory utilization on the server
 - ▶ Slower DB response times
 - ▶ Slow or unresponsive frontend



High Availability – Overview

- ▶ **Every layer can be made highly available**
- ▶ **Database**
 - ▶ PostgreSQL: Patroni + etcd (2+1 or 3 nodes)
 - ▶ MySQL: InnoDB Cluster or Galera (single-primary mode)
 - ▶ Clients must always connect to current Leader via VIP, DNS failover or load balancer
- ▶ **Zabbix Server**
 - ▶ Active/Passive pair
- ▶ **Proxy**
 - ▶ Proxy Groups (active-active, from Zabbix 7.0)
 - ▶ No external balancer needed, Zabbix Server handles it
- ▶ **Frontend**
 - ▶ Load balancer with traffic splitting (e.g. round-robin)

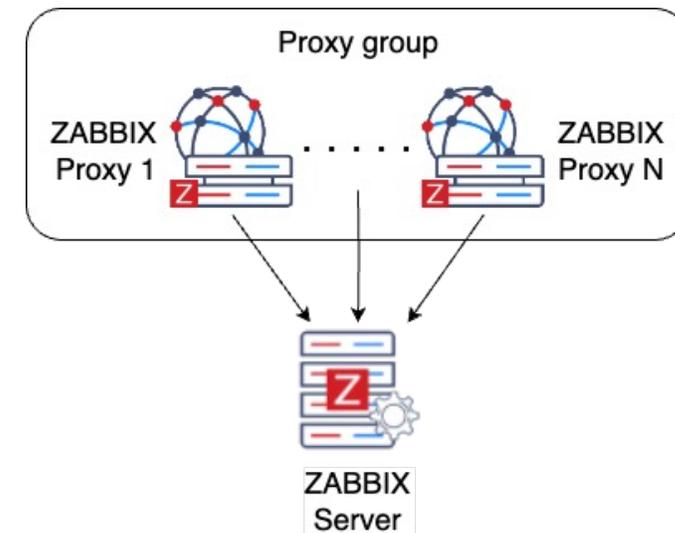


High Availability – Why start with DB

- ▶ **Database is the only stateful component**
 - ▶ All monitoring data lives here
- ▶ **Without DB, Zabbix Server cannot start or work**
- ▶ **If Server goes down**
 - ▶ DB stays readable – history and trends remain accessible through Frontend
 - ▶ Proxy servers can collect data locally in DB until the connection to the server is restored
- ▶ **HA deployment order**
 - ▶ DB → Server → Frontend
- ▶ **Without an HA database, any Zabbix Server HA is pointless**
 - ▶ DB is the only real Single Point Of Failure (SPOF)

Proxy Groups – Horizontal Scaling & Resilience

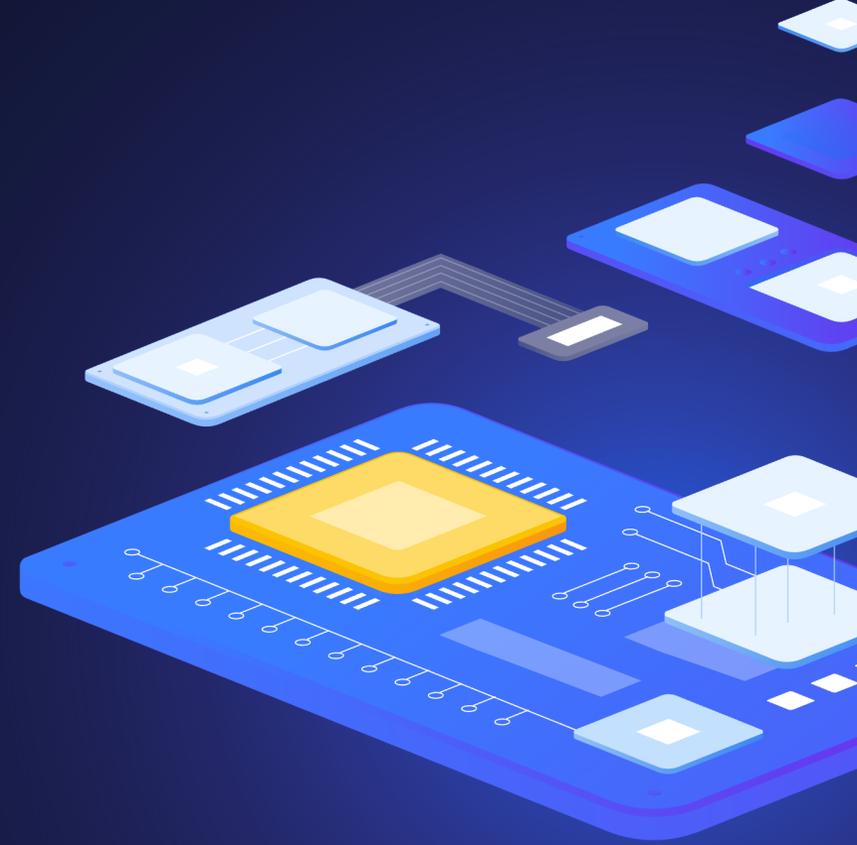
- ▶ **End of static assignment**
 - ▶ Moving from "1 host = 1 proxy" to dynamic groups
- ▶ **Failover**
 - ▶ On proxy failure, hosts are automatically moved to available proxies
- ▶ **Load balancing**
 - ▶ Hosts are rebalanced based on proxy load
- ▶ **Agent 2 dynamically changes its communication target within the group**
- ▶ **Key configuration parameters**
 - ▶ Failover period – time (sec) after which hosts are moved on proxy failure
 - ▶ Min online proxies – minimum number of active proxies for the group to function
 - ▶ Rebalancing delay – wait time before rebalancing after a proxy recovers



2

INSTALLATION & VERSION MANAGEMENT

LTS vs Major • RHEL DNF modules • PHP-FPM



LTS vs. Major versions – When to use which

› Timeline

- › 6.0 LTS → 6.2 → 6.4 → 7.0 LTS → 7.2 → 7.4 → 8.0 (upcoming)

› LTS versions (Long Term Support)

- › Released every ~1.5 years
- › 5 years of support: 3 years Full Support + 2 years Limited Support
- › Recommended for production and critical environments

› Standard (non-LTS) major versions

- › Released every ~6 months
- › 12 months of support: 6 months Full Support + 6 months Limited Support
- › New features available sooner (Proxy Groups, SCIM...)

TIP: Always test the upgrade on staging before applying to production

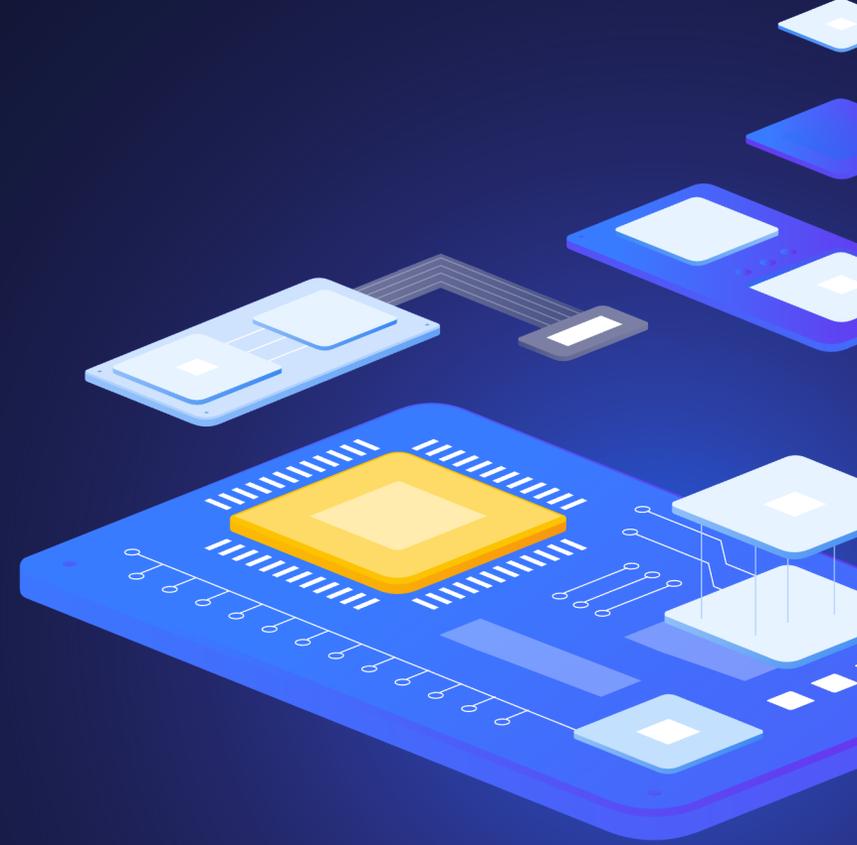
RHEL specifics – DNF modules and PHP-FPM

- › **Keep PHP and Nginx updated**
 - › Always use the latest available version
- › **RHEL 8/9/10 hides newer PHP and Nginx versions inside DNF modules**
 - › Default "dnf install php" installs only the basic OS version of PHP
 - › Correct approach - explicitly enable the module

```
# List available modules
dnf module list php
# Enable the right version
dnf module enable php:8.2
dnf module enable nginx:1.26.0
# Verify
php -v && nginx -v
```

- › **PHP-FPM: default memory_limit is not enough**
 - › Recommended: memory_limit = 512M

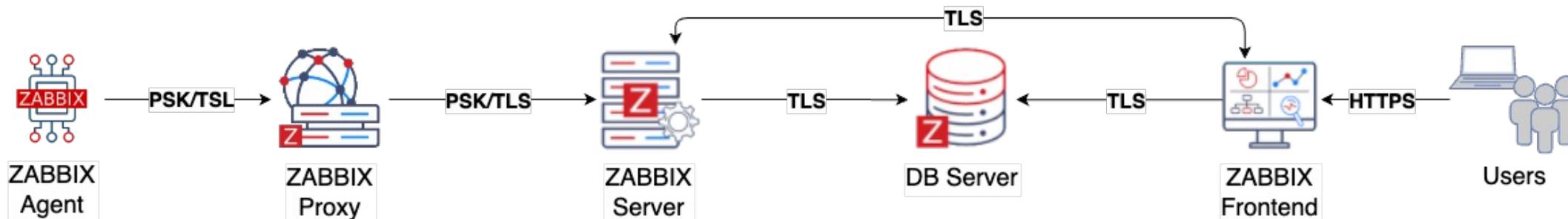
3



SECURITY & ENCRYPTED COMMUNICATION

PSK/SSL • Ansible automation • SELinux • Firewall • Auto-registration

SSL/TLS across all components



› **Agent → Proxy/Server**

- › PSK or certificates

› **Server/Proxy → DB**

- › `sslmode=verify-full` in `pg_hba.conf` and connection string

› **Frontend → DB**

- › SSL parameters in `zabbix.conf.php`

› **User → Frontend**

- › Standard HTTPS (certbot or commercial/internal CA)
- › Always over HTTPS – never use HTTP in production

Firewall Rules & Hardening

› Set firewall rules properly

- › Zabbix Server
 - › Allow 10051/tcp from Proxies and Agents only
- › Proxy
 - › Allow 10051/tcp from Server, 10050/tcp to passive Agents
- › Frontend
 - › Allow 443/tcp (HTTPS)
- › DB
 - › Allow 5432/tcp from Zabbix Server and Frontend only

› Intrusion Prevention (fail2ban)

- › Protect SSH access (sshd)
- › Ban after 5 failed login attempts
- › Use incremental bans for repeated attacks

PSK Security – Automation vs. Auto-registration

› PSK advantages for agent communication

- › Simple configuration – just generate a hex key and write it to both sides
- › Certificates
 - › Risky without mature PKI
 - › Manual lifecycle management at scale is prone to errors (expired certs)

› Critical limitation of Zabbix auto-registration

- › Auto-registration supports **ONLY** 1 global PSK key
- › One compromised agent = all other agents at risk

› Ansible solution (community.zabbix)

- › All agents are installed in the same way
- › Generates a unique PSK per host automatically
- › Keys can be stored in Ansible Vault (encrypted in git)
- › Re-running causes no unintended changes
- › Scales to thousands of hosts with no manual effort

SELinux – Proper policies instead of disabling

› What NOT to do

- › setenforce 0 – forbidden in production
- › SELINUX=disabled in /etc/selinux/config
- › Disabled SELinux = system without Mandatory Access Control protection
- › Security audits (ISO 27001) will always catch this

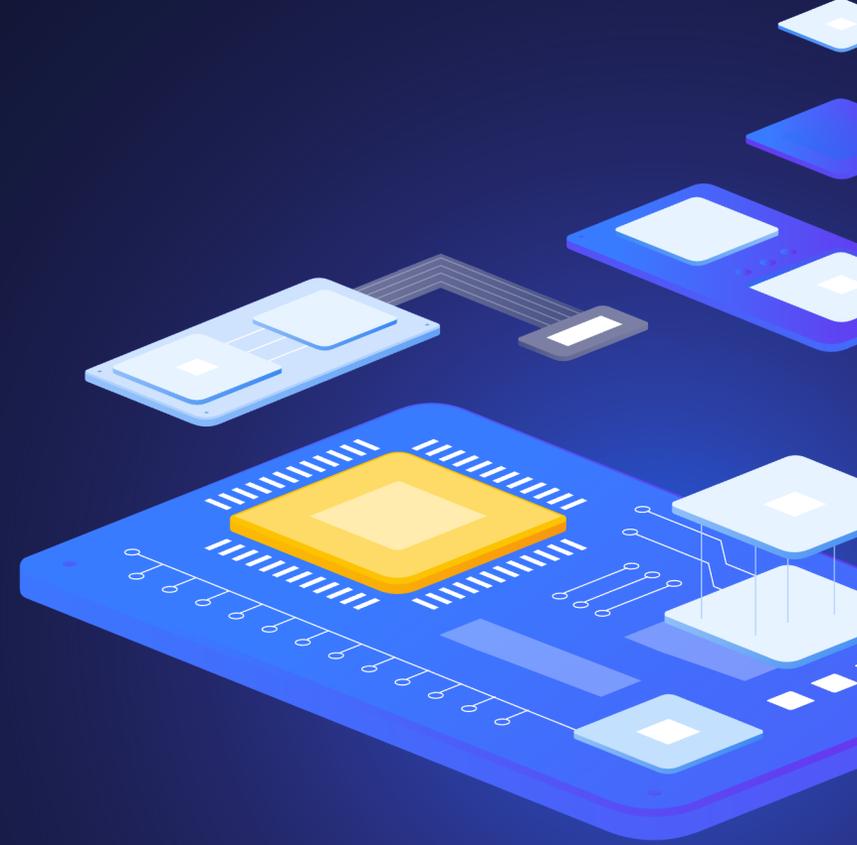
› Correct approach

- › Keep SELinux in Enforcing mode
- › Zabbix ships its own SELinux package – install it
- › Log denied actions "ausearch -m avc"
- › Generate policies from logs: "audit2allow"

4

IDENTITY & ACCESS MANAGEMENT

SAML 2.0 • JIT Provisioning • SCIM • Roles and User Groups



SAML + JIT vs. SCIM – When to use which

› SAML + JIT Provisioning

- › Login via SSO – Azure AD, Okta, ADFS, Keycloak...
- › JIT = user is created automatically on first login
- › DOWNSIDE: Deprovisioning must be handled manually or via scheduled jobs

› SCIM

- › Full automation of user lifecycle management
- › Automatic deprovisioning when an employee leaves the IdP
- › Group sync from IdP to Zabbix User Groups
- › Requires a SCIM-compatible IdP (Okta, Azure AD, Entra ID)

› Deactivate instead of Delete

- › Dashboards, maps and other objects in Zabbix are tied to the user record
- › Deactivated user cannot log in but all their work is preserved
- › Zabbix moves offboarded users to a "Deprovisioned group"

Users, roles and naming conventions

› Zabbix User Roles

- › Super Admin full access
- › Admin manage hosts and templates, no system settings
- › User read data and acknowledge problems
- › Guest read-only rights, see dashboards
- › Custom role granular permissions per UI section

› Naming Conventions & Access Control

› 1:1 Mapping (Simple)

- › User Group name matches Host Group name
- › Example: HG "Prague / Windows" → UG "Prague / Windows"
- › Pros: Immediate clarity during troubleshooting

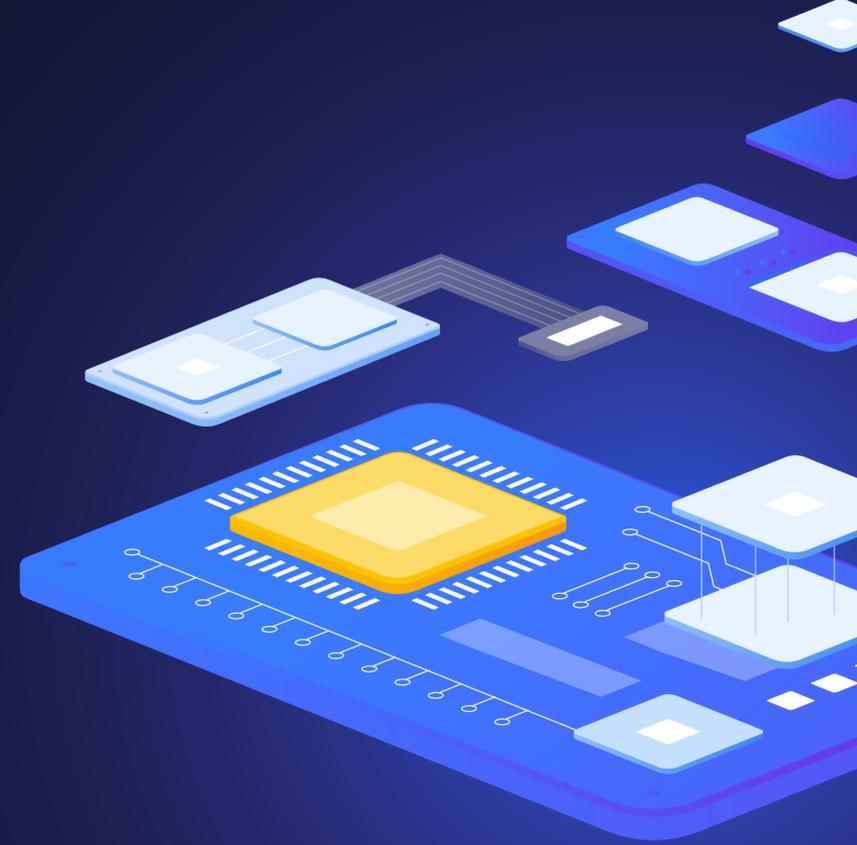
› Functional Mapping (Enterprise)

- › Linking User Groups to technology trees
- › Example: UG "DB Admins" mapped to all "Database/*" Host Groups
- › Note: High-density environments require External Automation (API) to maintain consistency

5

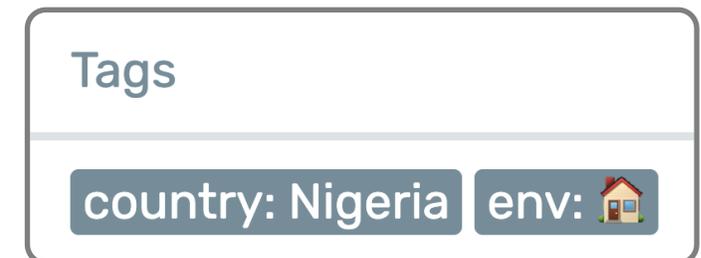
HOST GROUPS, TAGS & TEMPLATES

Structure • Tagging • Clone templates



Host Groups – Two trees + Tags

- ▶ **Structured Host Groups - two parallel trees**
 - ▶ Tree 1 (Technology)
 - ▶ Linux/RHEL, Linux/Ubuntu, Windows/Servers, Network/Switches...
 - ▶ Tree 2 (Location)
 - ▶ CZ/Prague/DC1, CZ/Brno, SK/Bratislava, Cloud/AWS...
 - ▶ A host can belong to groups from both trees simultaneously
- ▶ **Tags as a complement for finer classification**
 - ▶ env:production, env:staging, env:dev, env:🏠
 - ▶ .app:nginx, .app:postgres (.dot prefix = always FIRST in GUI)
 - ▶ tier:web, tier:db, tier:cache
 - ▶ ~owner:ops-team (~tilde prefix = always LAST in GUI)



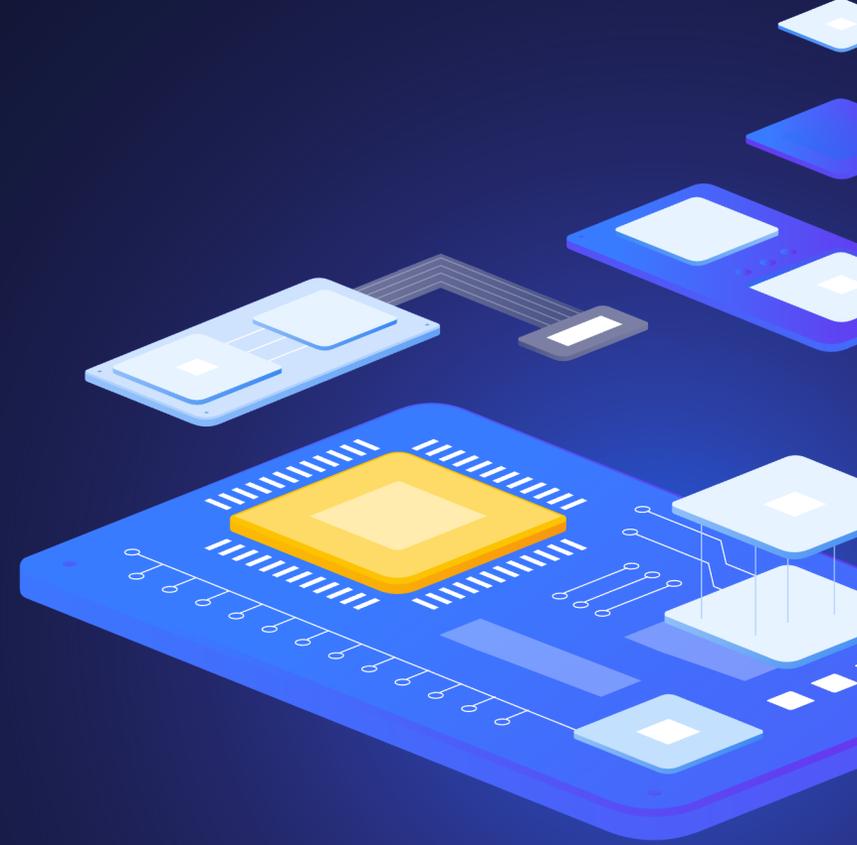
Templates – Clone workflow and GitOps protection

- ▶ **Default templates are EXAMPLES, not production configurations**
 - ▶ Thresholds in default templates are generic – not suited to your environment
- ▶ **Never modify default templates directly**
 - ▶ Importing an updated default template will overwrite your changes
- ▶ **Correct workflow**
 - ▶ Find a suitable default template as your starting point
 - ▶ Create a CLONE – name it clearly, e.g. "Custom: Linux by Zabbix agent"
 - ▶ Modify the clone: thresholds, disabled items, custom triggers
 - ▶ Default template = reference only, never link directly to hosts
 - ▶ After every Zabbix upgrade: diff default vs. your custom templates

6

PERFORMANCE & TUNING

Triggers • Caches • Internal Processes • TimescaleDB



History vs. Trend functions in triggers

▶ History functions (work with raw values)

- ▶ last(), avg(1h), min(5m), max(10m), count(1h), ...
- ▶ Results cached in Value Cache (RAM) - fast for short periods
- ▶ On cache miss: read from DB - I/O-intensive
- ▶ Avoid for long periods
 - ▶ Holding many raw values exhausts Value Cache quickly

▶ Trend functions (work with hourly aggregates)

- ▶ trendavg(7d), trendmax(30d), trendmin(90d)
- ▶ Work with hourly aggregates instead of raw values - far less data to hold
- ▶ Extremely fast even for long periods
 - ▶ Works with hourly aggregates instead of thousands of raw values
- ▶ Use for: calculated items, long-term triggers and anomaly detection

Zabbix internal caches – overview

Cache	Config param	Default	Purpose	Monitor key
Value Cache	ValueCacheSize	8MB	Latest item values for trigger evaluation	zabbix[vcache,buffer,pused]
Configuration Cache	CacheSize	8MB	Host/item/trigger configuration	zabbix[rcache,buffer,pused]
History Cache	HistoryCacheSize	16MB	Buffer incoming values before DB write	zabbix[wcache,history,pused]
Trend Cache	TrendCacheSize	4MB	Buffer hourly aggregates before DB write	zabbix[wcache,trend,pused]

- ▶ **Keep all caches below 60% utilization – increase size if any cache exceeds this threshold.**

Debug and Internal Processes

› Runtime diagnostics – without restarting Zabbix Server

```
# Dump internal process statistics
zabbix_server -R diaginfo

# Increase log level at runtime (debug without restart)
zabbix_server -R log_level_increase
zabbix_server -R log_level_increase=poller # specific process only
```

› Scaling internal processes (zabbix_server.conf)

- › Poller utilization > 60% → increase number of pollers
- › History syncer busy → increase StartDBSyncers
- › Preprocessor busy → increase StartPreprocessors
- › Monitor: zabbix[process,<type>,avg,busy]

TimescaleDB – Compression and data retention

› Why TimescaleDB for Zabbix

- › PostgreSQL extension optimized for time-series data
- › Automatic compression of older data – typically 60–80% storage savings
- › Hypertables: transparent automatic partitioning by time
- › Continuous aggregates for fast trend queries
- › Native integration in the Zabbix
- › Data retention simplified – entire chunks are dropped at once instead of row-by-row deletion
- › Reduces Housekeeper process load significantly

› Real-world example

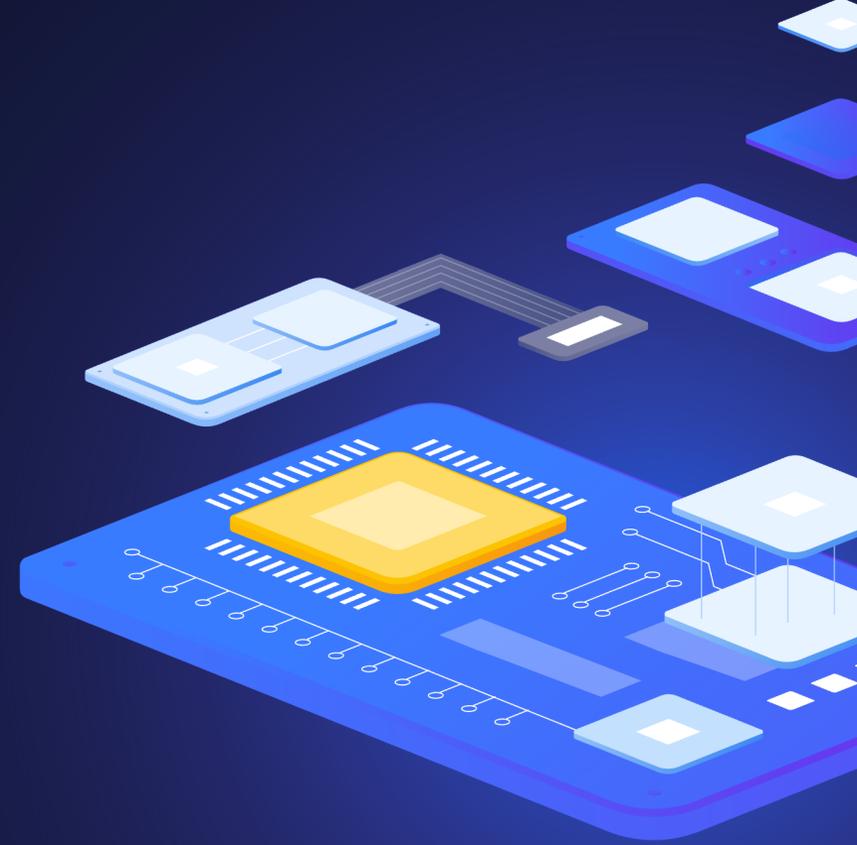
- › 200 GB uncompressed history → ~60 GB with TimescaleDB compression

- › Deploying TimescaleDB from the start is much easier/faster than retrofitting it later.

7

MAINTENANCE & OPERATIONS

Self-monitoring • Housekeeping • Escalation • End-user Training



Watcher of the Watcher – Self-monitoring

› Zabbix provides built-in templates for self-monitoring

- › Template "Zabbix server health" – covers caches, queues, process utilization, unsupported items
- › Template "Zabbix proxy health" – same metrics for proxy instances
- › Apply both immediately after installation

› Recommended self-monitoring triggers

- › Queue > 100 items for 5 min → HIGH (server can't keep up with collection)
- › Unsupported items > 50 → WARNING (review templates and agents)
- › Poller utilization > 60% → WARNING (increase StartPollers)
- › Value cache hits < 90% → WARNING (increase ValueCacheSize)

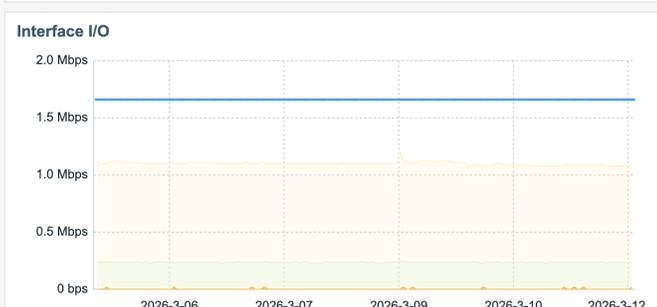
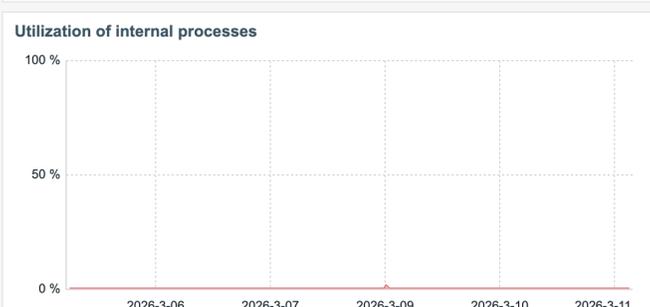
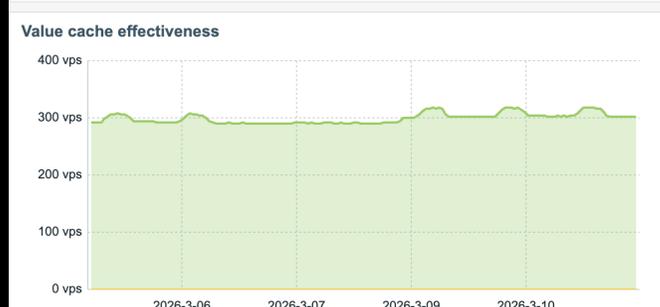
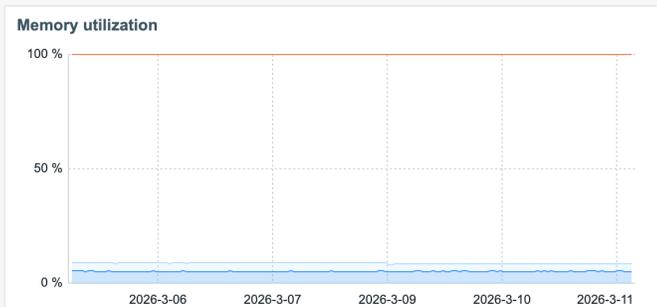
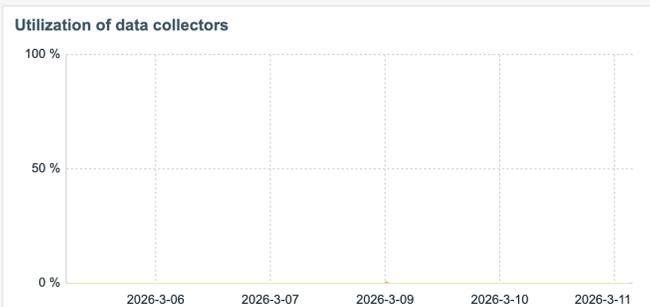
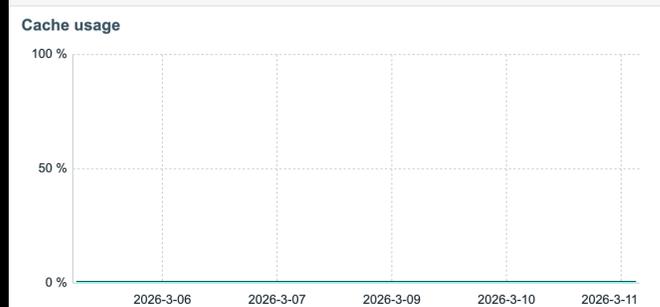
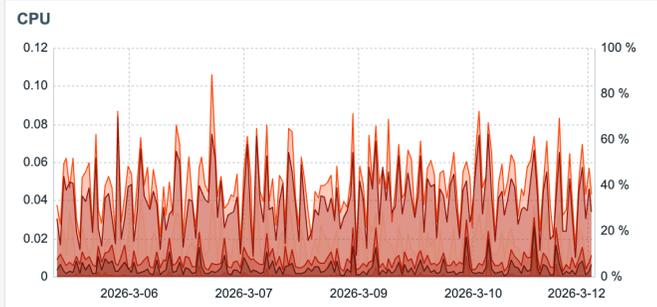
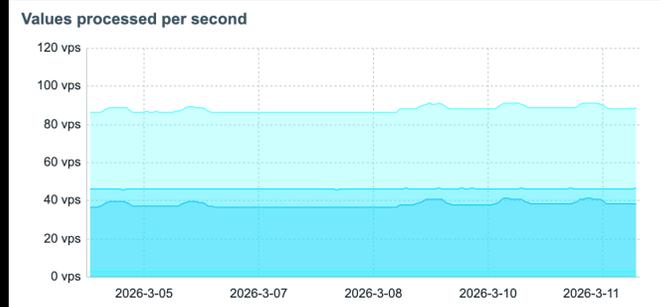
› Who watches the watcher?

- › Without an external watcher you won't know Zabbix is down
- › Exactly when you need it most

Zabbix server

Server version 7.4.6	Queue 147.00 ↑	LLD Queue 0.00	VPS 88.52 vps ↓	VPS 89.13
03:08:05	Queue over 10m 1.00	Preproc. Queue 0.00	VPS (unsupported) 0.33 vps ↓	
zabbix-be-01 10.1.2.1	Number of CPUs 8 vCPU	Total memory 11.42 GB	Agent version 7.4.6	System uptime 79 days, 16:21:20
zabbix-be-02 10.1.2.2	Number of CPUs 8 vCPU	Total memory 11.42 GB	Agent version 7.4.6	System uptime 79 days, 16:20:29

Time	Recovery time	Status	Info	Host	Problem • Severity	Duration	Update	Actions
No data found								

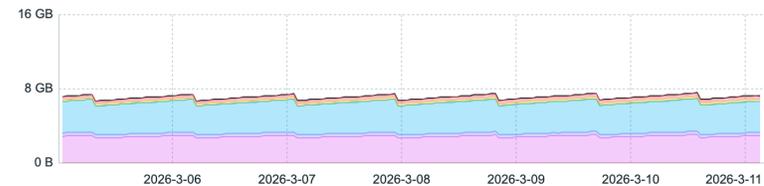


PostgreSQL DB Cluster

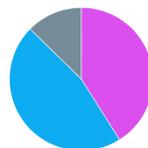
Top hosts

Host	Agent version	Number of CPUs	Total memory	Version	Role	Lag in bytes	DB space	Utilization	Version	Is leader?	Has leader?	Node health	DB size		
zabbix-db-node-01	7.4.6	16	31.09 GB	PostgreSQL	18.0	Up (1)	0.00 B	459.77 GB	<input type="text" value="10.43 %"/> 10.43 %	ETCD	3.6.5	No (0)	Yes (1)	Ok (1)	1.45 MB
zabbix-db-node-02	7.4.6	16	31.09 GB	PostgreSQL	18.0	Master (2)	0.00 B	459.77 GB	<input type="text" value="10.62 %"/> 10.62 %	ETCD	3.6.5	Yes (1)	Yes (1)	Ok (1)	1.45 MB
zabbix-db-witn-03	7.4.6	4	7.50 GB	PostgreSQL					ETCD	3.6.5	No (0)	Yes (1)	Ok (1)	1.45 MB	

Zabbix DB size



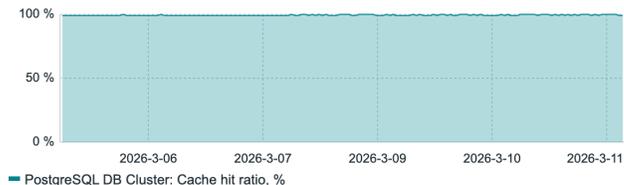
Zabbix table...



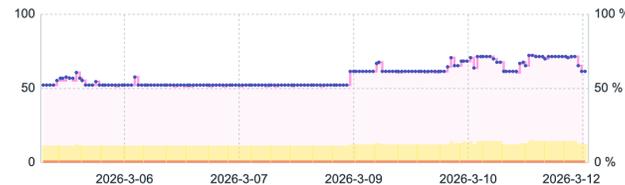
Problems

Time	Recovery time	Status	Info	Host	Problem • Severity	Operational data	Duration	Update	Actions
No data found									

Cache hit



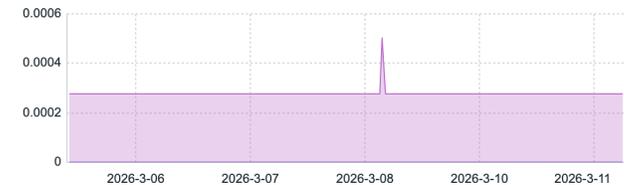
DB connections summary



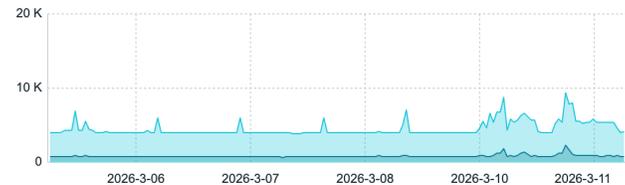
Transactions (TPS)



Checkpoints



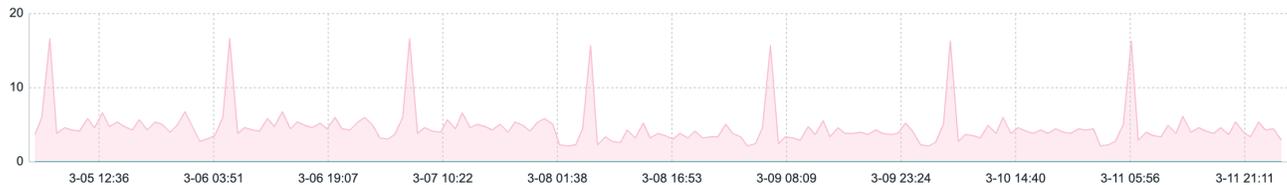
Rows



DB exclusive locks

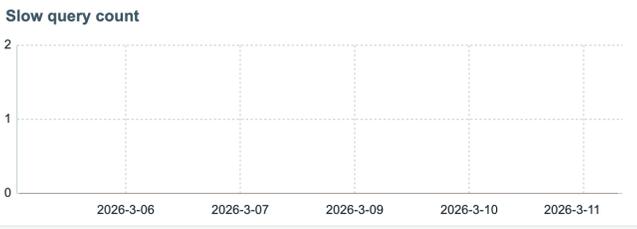
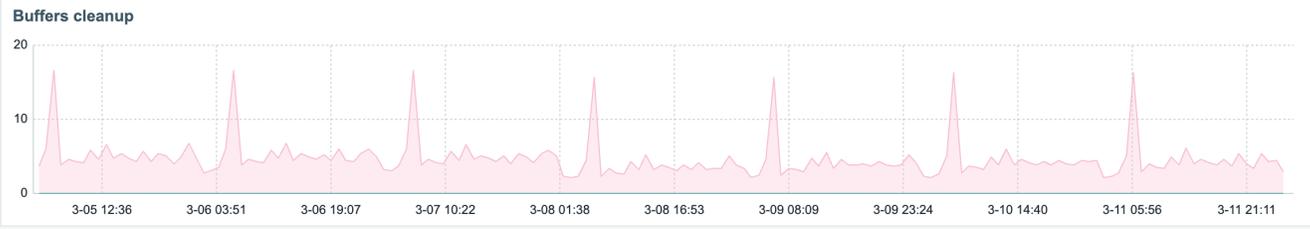
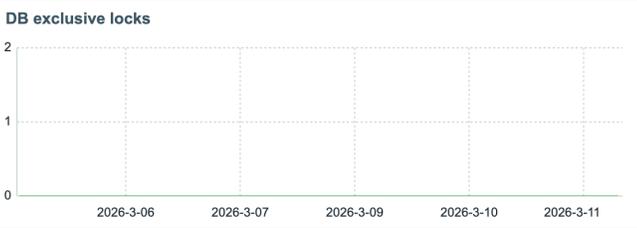
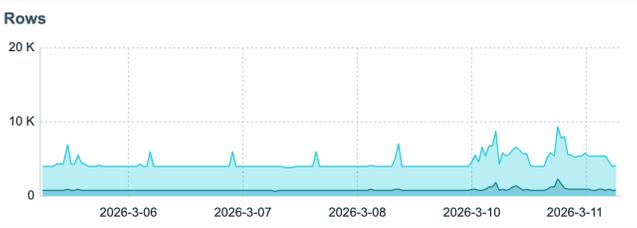
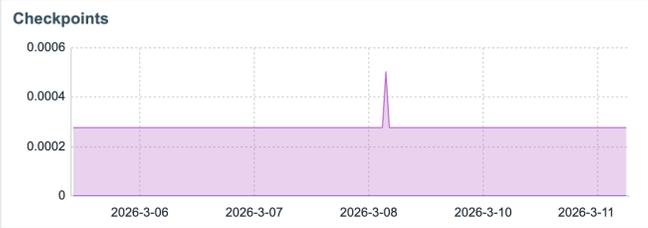
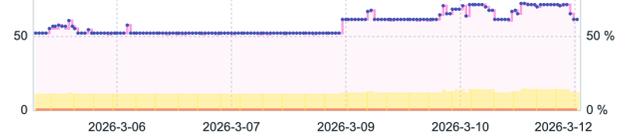
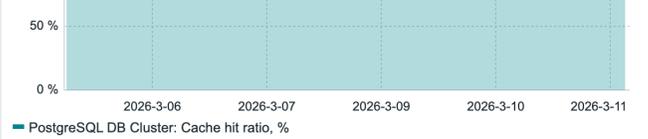


Buffers cleanup



Slow query count

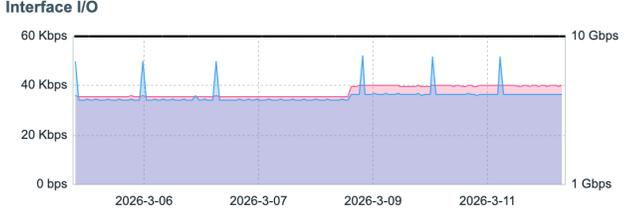
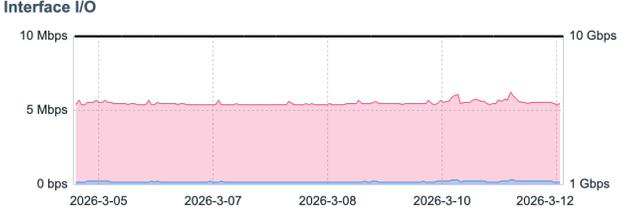
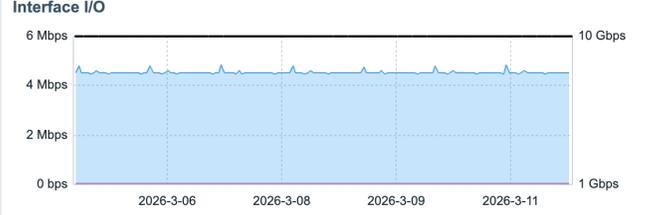
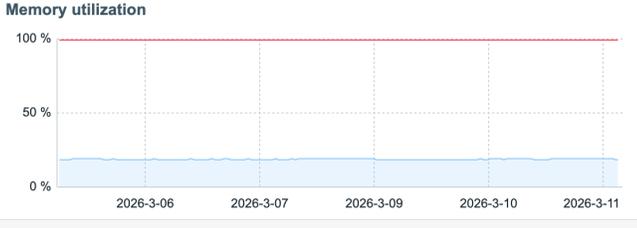
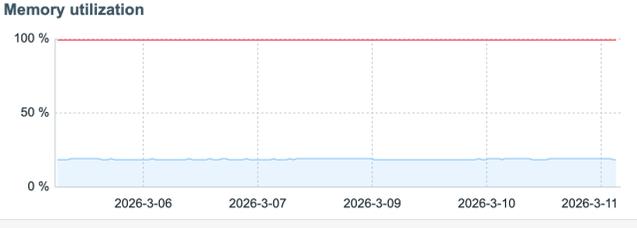
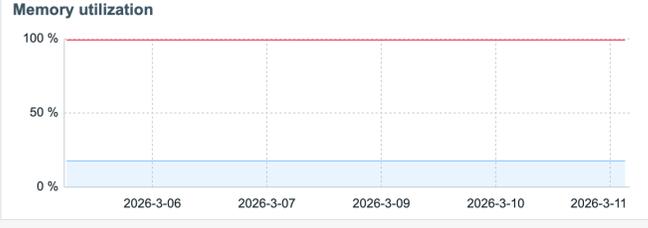
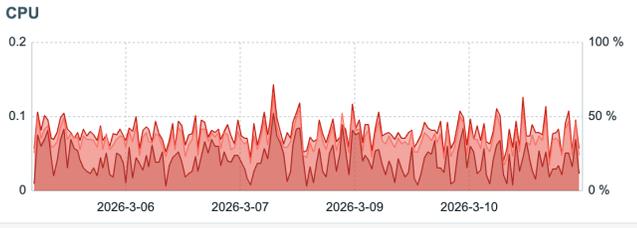
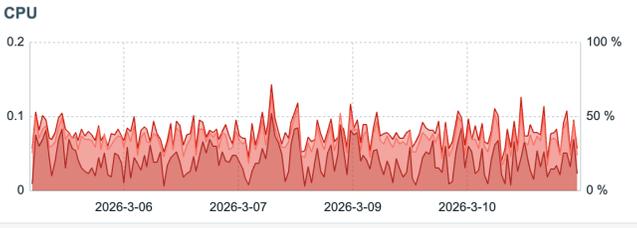
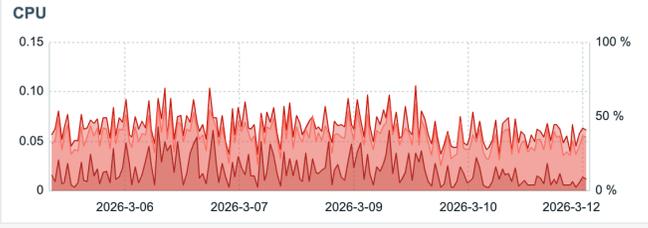




zabbix-db-node-01

zabbix-db-node-02

zabbix-db-witn-03



Zabbix proxies

Proxies

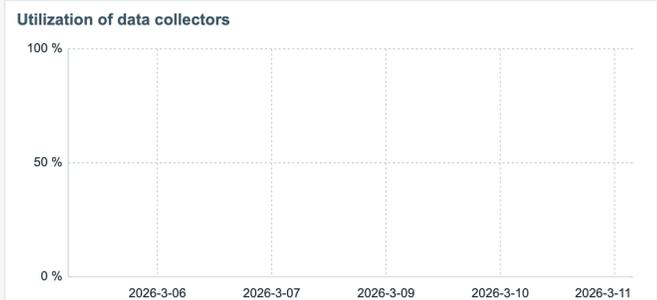
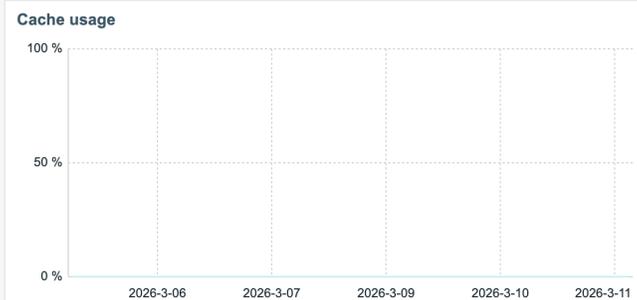
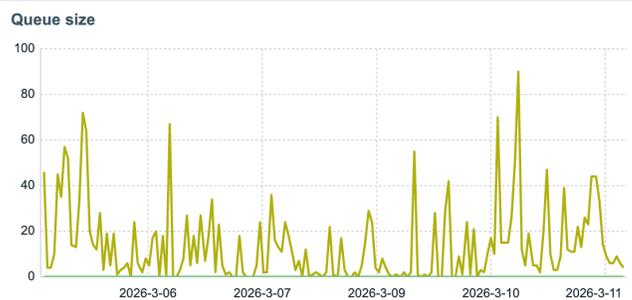
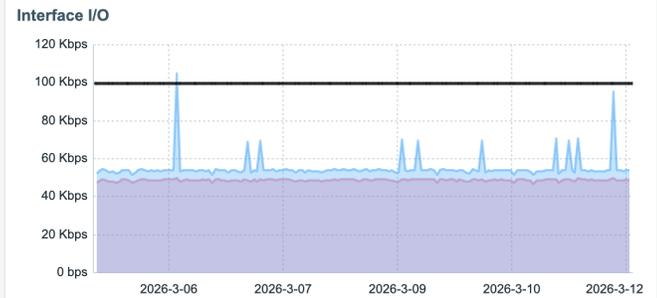
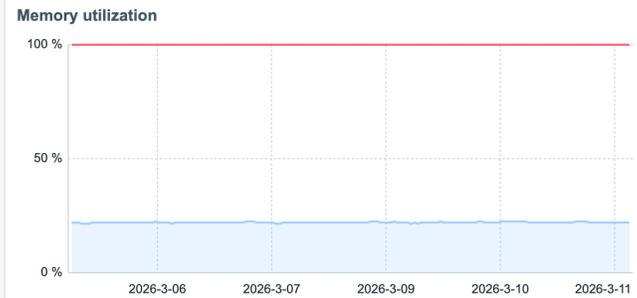
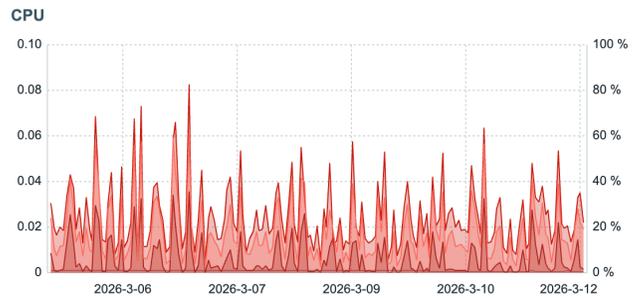
- ▼ Zabbix/Proxies 2
- zabbix-dmz-proxy-01
- zabbix-dmz-proxy-02
- zabbix-proxy-01
- zabbix-proxy-02 1
- zabbix-proxy-03
- zabbix-proxy-04 1

Host	Proxy v.	Agent v.	VPS (Req)	VPS	VPS (N/S)	Hosts	Items	Conf cache%	CPU util.	Memory %	Buffer %	Buffer state	Poller %	Pinger %	SNMP %	HTTP %
zabbix-dmz-proxy-01	7.4.6	7.4.6	26.15 vps	24.51 vps	0.00 vps	12 hosts	1659 items	0.11 %	1.04 %	22.38 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.00 %
zabbix-dmz-proxy-02	7.4.6	7.4.6	31.89 vps	30.64 vps	0.00 vps	13 hosts	2014 items	0.14 %	1.08 %	24.33 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.00 %
zabbix-proxy-01	7.4.6	7.4.6	0.91 vps	0.83 vps	0.00 vps	1 hosts	75 items	0.02 %	0.69 %	18.90 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.01 %
zabbix-proxy-02	7.4.6	7.4.6	0.72 vps	0.70 vps	0.00 vps	1 hosts	53 items	0.02 %	0.67 %	17.85 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.01 %
zabbix-proxy-03	7.4.6	7.4.6	0.00 vps	0.00 vps	0.00 vps	0 hosts	0 items	0.01 %	0.48 %	16.55 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.00 %
zabbix-proxy-04	7.4.6	7.4.6	0.57 vps	0.02 vps	0.00 vps	1 hosts	43 items	0.02 %	0.21 %	8.60 %	0.01 %	Memory (1)	0.00 %	0.01 %	0.00 %	0.01 %

zabbix-dmz-proxy-01 10.1.9.1	Proxy version 7.4.6	Agent version 7.4.6	Number of CPUs 2 vCPU	Total memory 3.57 GB	Proxy uptime 63 days, 18:58:37	System uptime 63 days, 18:58:26
--	-------------------------------	-------------------------------	---------------------------------	--------------------------------	--	---

Problems

Time ▼	Recovery time	Status	Info	Host	Problem • Severity	Operational data	Duration	Update	Actions
<div style="text-align: center; font-size: 2em;">🔍</div> <p>No data found</p>									



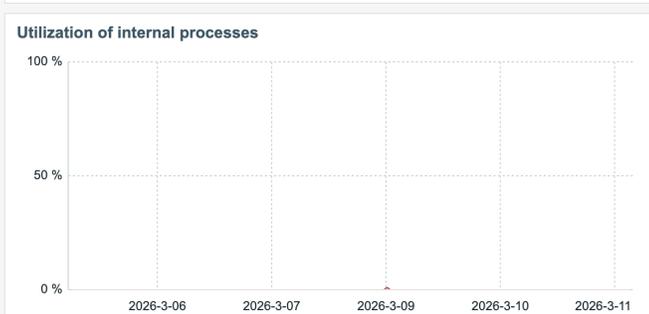
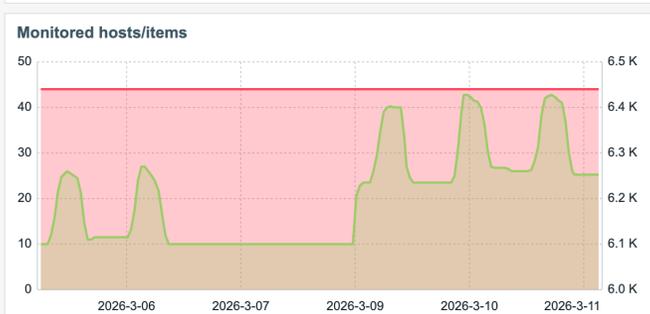
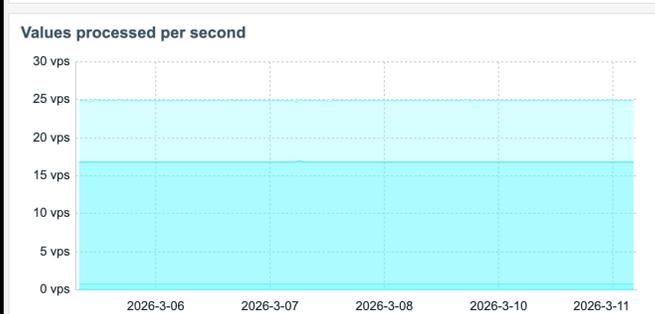
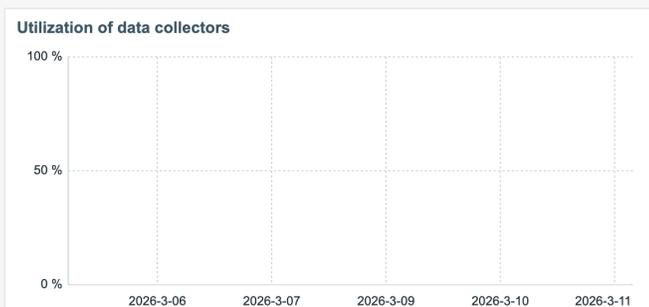
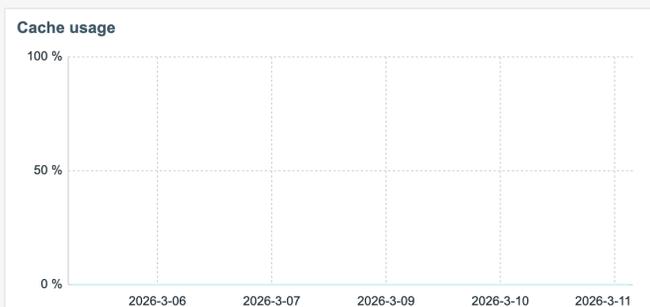
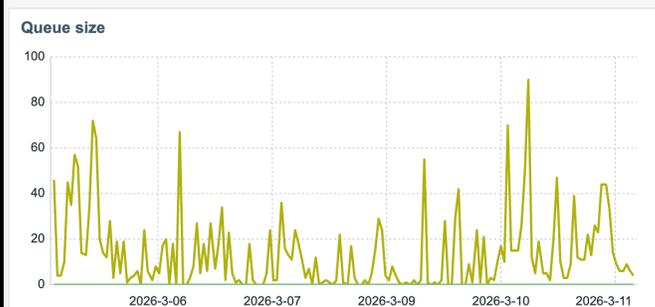
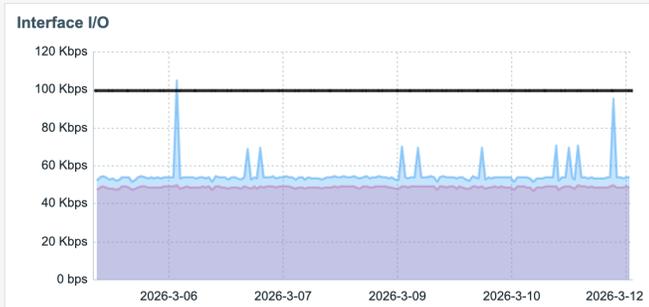
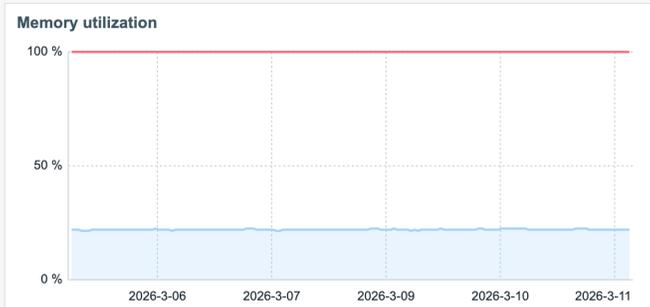
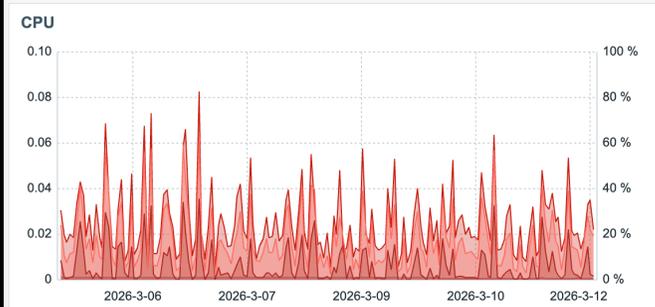
zabbix-proxy-01	
zabbix-proxy-02	1
zabbix-proxy-03	
zabbix-proxy-04	1

zabbix-proxy-03	7.4.6	7.4.6	0.00 vps	0.00 vps	0.00 vps	0 hosts	0 items	0.01 %	0.48 %	16.55 %	0.01 %	Memory (1)	0.00 %	0.00 %	0.00 %	0.00 %
zabbix-proxy-04	7.4.6	7.4.6	0.57 vps	0.02 vps	0.00 vps	1 hosts	43 items	0.02 %	0.21 %	8.60 %	0.01 %	Memory (1)	0.00 %	0.01 %	0.00 %	0.01 %

zabbix-dmz-proxy-01 10.1.9.1	Proxy version 7.4.6	Agent version 7.4.6	Number of CPUs 2 vCPU	Total memory 3.57 GB	Proxy uptime 63 days, 18:58:37	System uptime 63 days, 18:58:26
--	-------------------------------	-------------------------------	---------------------------------	--------------------------------	--	---

Problems

Time	Recovery time	Status	Info	Host	Problem • Severity	Operational data	Duration	Update	Actions
 No data found									



Regular housekeeping and notifications

› Monthly operational housekeeping

- › Unsupported items
 - › Data collection → Hosts → Items → clear Host filter → filter by State: Not supported → review and fix
- › Unused templates
 - › Review templates with 0 linked hosts – remove or archive custom ones based on your team's workflow
- › Disabled hosts, triggers,...:
 - › Why are they disabled? Still relevant?
- › DB vacuum: configure PostgreSQL autovacuum for Zabbix database

› Notification best practices

- › Increase alert repetition: 1× is not enough in prod environment
- › Escalation
 - › 5 min → admin, 30 min → team lead, 60 min → management
- › Recovery message
 - The team needs to know when the problem is resolved
- › Separate media types per environment
 - › Prod = PagerDuty
 - › Dev = Slack/email

Regular housekeeping and notifications

› End-user training

- › Most common issue
 - › Users don't know how to filter problems using tags
- › Teach
 - › Difference between Acknowledge and Close problem (Close ≠ problem resolved!)
- › Maintenance windows
 - › When and how to set them correctly (otherwise spurious alerts)



SUMMARY

Key takeaways

› Architecture

- › Separate Zabbix server and DB
- › Start with HA at DB
- › Use Proxies

› Installation

- › LTS for production
- › DNF modules on RHEL
- › Increase PHP memory_limit

› Security

- › Generate unique PSK with automation
- › SSL everywhere
- › Set SELinux enforcing and Firewall rules

› Identity

- › JIT for quick SSO onboarding
- › SCIM for lifecycle – deactivates, never deletes
- › Consistent group naming conventions

› Templates

- › Never edit defaults, clone them
- › Monitor template updates in release notes
- › Merge template changes after upgrade

› Performance

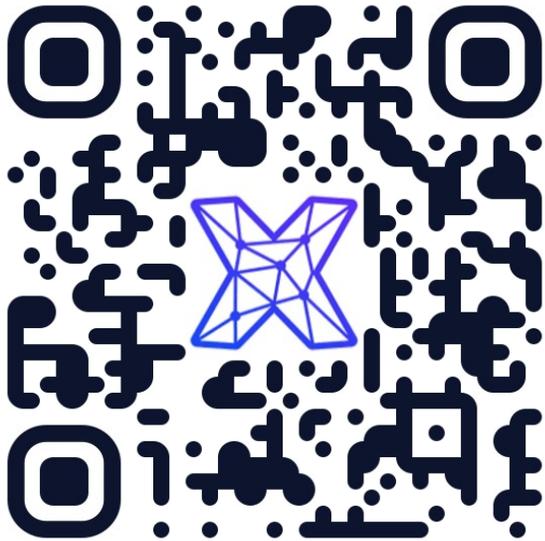
- › Size all caches properly
- › Use TimescaleDB
- › Throttling in preprocessing

› Operations

- › Self-monitoring + external watcher
- › Monthly housekeeping
- › Train your users

Tips and tricks

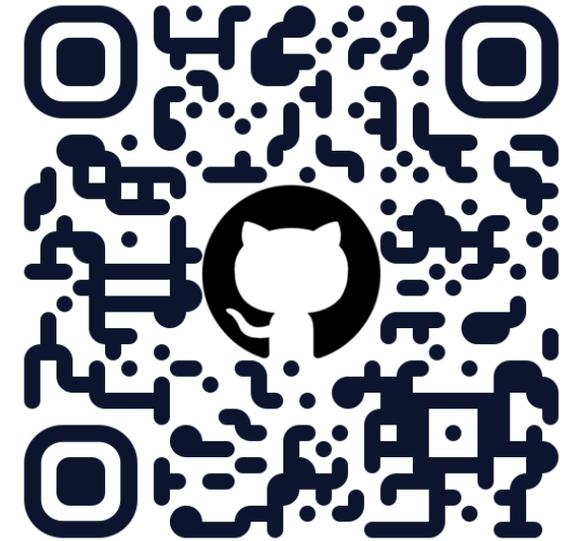
From time to time check our wiki, social networks or GitHub:



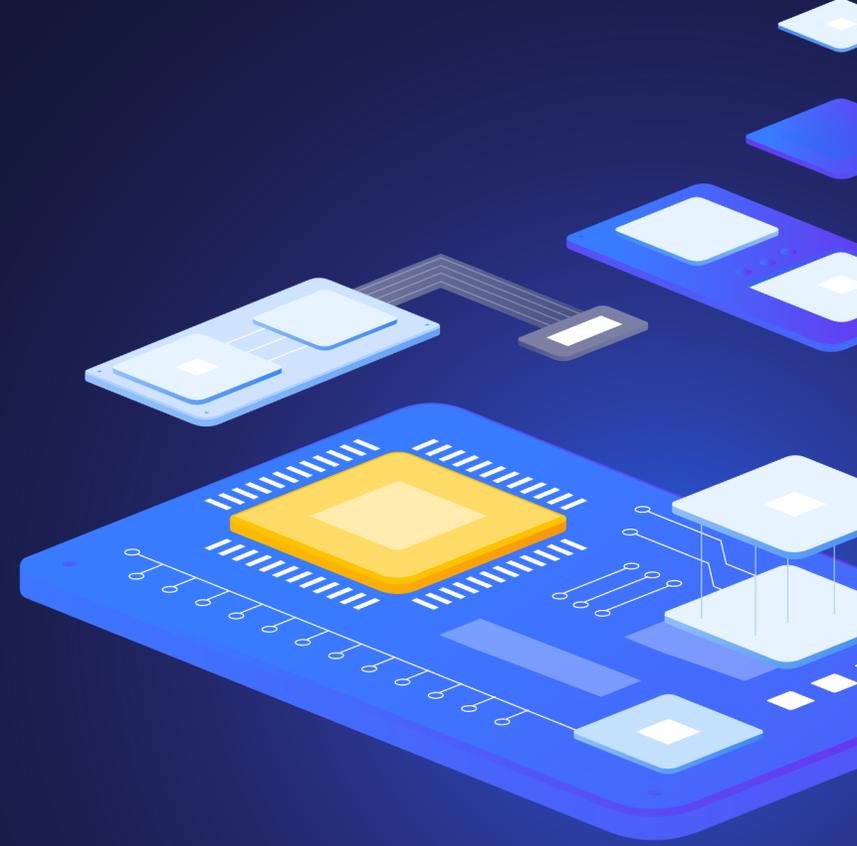
Wiki



LinkedIn



GitHub



QUESTIONS ?

Webinars

- ▶ CZ: <https://www.initmax.cz/webinare/>
- ▶ EN: <https://www.initmax.com/webinars/>

- ▶ **Oprávnění v Zabbixu: role, skupiny a best practices**
 - ▶ 26.3.2026

- ▶ **Nested LLD v praxi: praktické použití pro monitoring Proxmoxu v Zabbixu**
 - ▶ 16.4.2026

- ▶ **Korelace událostí v Zabbixu pomocí n8n: workflow a automatizace**
 - ▶ 23.4.2026

Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.com>

Email:



marek.rysavy@initmax.com

LinkedIn:



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

Twitter:



<https://twitter.com/initmax>

Marek Ryšavý:



+420 732 218 870