



MEETUP ONLINE '21

DEPLOYING AND CONFIGURING ZABBIX 5.4 IN A MULTI-TENANT ENVIRONMENT

ARTŪRS LONTONS
TECHNICAL SUPPORT ENGINEER
ZABBIX

MONITORING REQUIREMENTS OF MULTI-TENANT ENVIRONMENTS

- ✓ Granular role/permission schema
- ✓ Ability to process large amounts of data
- ✓ Redundancy
- ✓ Infrastructure scalability
- ✓ Support of different monitoring approaches

SUPPORTED MONITORING APPROACHES - AGENT

- ✓ Supports passive (polling) and active modes (trapping)

Supports deployment on different platforms (Windows/Unix-like) and provides access to OS level metrics, collecting data from log files and metric collection via custom scripts and commands.



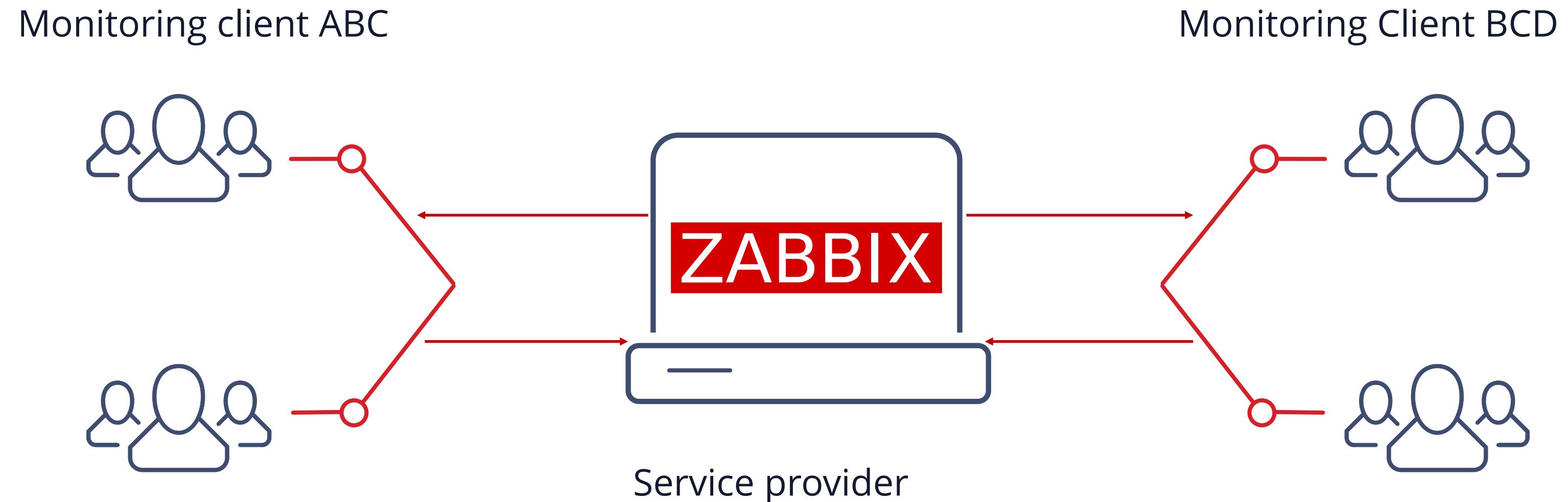
WHAT IF AGENT IS NOT AN OPTION?

- ✓ SNMP, HTTP, IPMI and SSH agentless monitoring
- ✓ Simple checks (ICMP pings, port status)
- ✓ Database and Java monitoring
- ✓ External scripts
- ✓ Data aggregations and calculations
- ✓ VMware monitoring
- ✓ Web monitoring
- ✓ Synthetic monitoring

LATEST IMPROVEMENTS

- ✓ Version 5.2
 - ✓ Permission logic reworked – user roles have been implemented
 - ✓ Support for external vault
 - ✓ Granular frontend configuration
- ✓ Version 5.4
 - ✓ Ability to send scheduled reports
 - ✓ Performance improvements

ZABBIX AND MULTI-TENANT ENVIRONMENTS



ZABBIX PROXY

Collects and preprocesses the data. Supports all of the metric collection methods.

- ✓ Data gets compressed before forwarding it to the server
- ✓ Data collection continues even in case of network interruptions
- ✓ A single connection is used to send the data to the server
- ✓ Enables remote script execution
- ✓ Can be used to improve scalability

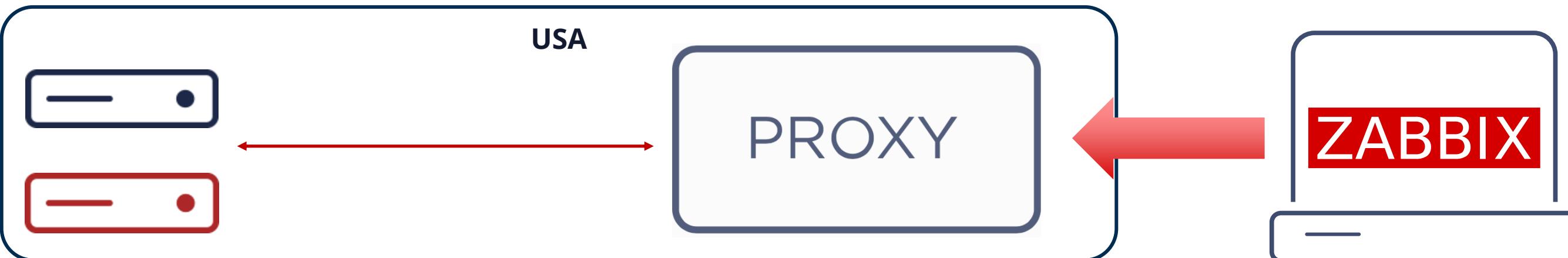
ZABBIX PROXY

- ✓ Select the direction of the connection

Active proxy



Passive proxy



DATA PREPROCESSING - THROTTLING

- ✓ By using preprocessing, we can decrease the amount of stored data by discarding repeating values
- ✓ This preprocessing method is called Throttling

discard unchanged

00:00 value - 1
00:01 value - 2 OK
00:02 value - 2 DROPPED
00:02 value - 2 DROPPED
10:02 value - 2 DROPPED
50:02 value - 2 DROPPED

discard unchanged heartbeat 2minutes

00:00 value - 1
00:05 value - 1 dropped
01:05 value - 1 dropped
01:55 value - 1 dropped
02:00 value - 1 SENT
02:01 value - 1 dropped

PERMISSIONS

- ✓ Permissions are based on User group to host group relations
- ✓ Three user types – User, Admin, Super admin
- ✓ User Roles

Users

User Media Permissions

* Role

User type

Permissions

Host group	Permissions
All groups	None

Permissions can be assigned for user groups only.

Access to UI elements

Monitoring

Inventory

Reports

Access to modules
No enabled modules found.

Access to API

Access to actions

PERMISSIONS - ROLES

- ✓ Roles grant the ability to configure access to specific UI elements, actions and restrict API calls in a granular fashion.

User roles

* Name **NOC Team**

User type **User**

Access to UI elements

Monitoring	<input checked="" type="checkbox"/> Dashboard	<input type="checkbox"/> Problems	<input type="checkbox"/> Hosts
	<input type="checkbox"/> Overview	<input type="checkbox"/> Latest data	<input type="checkbox"/> Screens
	<input checked="" type="checkbox"/> Maps	<input type="checkbox"/> Discovery	<input type="checkbox"/> Services
Inventory	<input type="checkbox"/> Overview	<input type="checkbox"/> Hosts	
Reports	<input type="checkbox"/> System information	<input type="checkbox"/> Availability report	<input type="checkbox"/> Triggers top 100
	<input type="checkbox"/> Audit	<input type="checkbox"/> Action log	<input type="checkbox"/> Notifications
Configuration	<input type="checkbox"/> Host groups	<input type="checkbox"/> Templates	<input type="checkbox"/> Hosts
	<input type="checkbox"/> Maintenance	<input type="checkbox"/> Actions	<input type="checkbox"/> Event correlation
	<input type="checkbox"/> Discovery	<input type="checkbox"/> Services	
Administration	<input type="checkbox"/> General	<input type="checkbox"/> Proxies	<input type="checkbox"/> Authentication
	<input type="checkbox"/> User groups	<input type="checkbox"/> User roles	<input type="checkbox"/> Users
	<input type="checkbox"/> Media types	<input type="checkbox"/> Scripts	<input type="checkbox"/> Queue

Access to actions

- Create and edit dashboards and screens
- Create and edit maps
- Create and edit maintenance
- Acknowledge problems
- Close problems
- Change severity
- Add problem comments
- Execute scripts

PERMISSIONS – USER GROUPS

- ✓ Access to hosts is defined on User group level
- ✓ Users can have Full/Read only/Deny permissions on particular host groups

User group Permissions ● Tag filter

Permissions	Host group	Permissions		
All groups		None		
Application Servers	Read-write	Read	Deny	None
Linux servers	Read-write	Read	Deny	None
Network devices	Read-write	Read	Deny	None
Riga Servers	Read-write	Read	Deny	None

type here to search Select Read-write Read Deny None
 Include subgroups
[Add](#)

Add Cancel

PERMISSIONS – HOST GROUPS

- ✓ Using group hierarchy can be very beneficial in such scenarios

Name ▲	Hosts	Templates	Members	Info
Customer ABC	Hosts	Templates		
Customer ABC/Network Devices	Hosts	Templates		
Customer ABC/Servers	Hosts	Templates		
Customer DEF	Hosts	Templates		
Customer DEF/Network Devices/Cisco	Hosts	Templates		

User group Permissions Tag filter

Permissions Host group Permissions

All groups	None
------------	------

Customer ABC X
type here to search

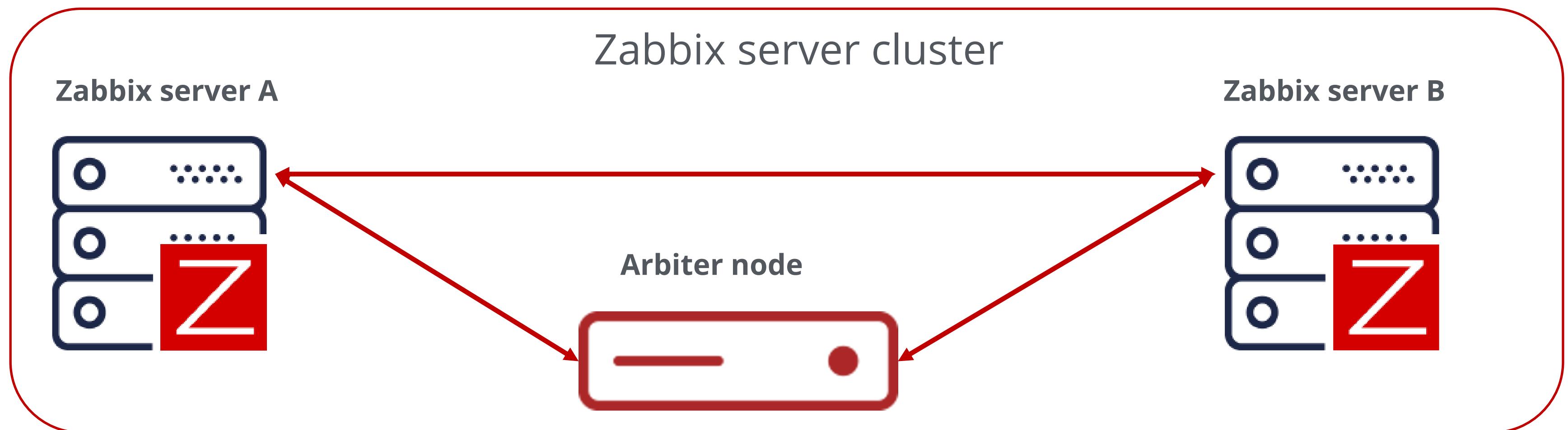
Include subgroups

Select Read-write Read Deny None

Add

HIGH AVAILABILITY

- ✓ Having near 100% uptime can be especially critical for such environments
- ✓ HA can be used to minimize downtime and add redundancy
- ✓ Tools such as PCS, Corosync, Pacemaker are used to enable the HA
- ✓ Out of the box HA planned for Zabbix 6.0



HIGH AVAILABILITY - NOTES

- ✓ Odd number of nodes should be used to achieve quorum
- ✓ Only Active/Passive cluster architecture is supported
- ✓ Two nodes should never be active at the same time
- ✓ STONITH mechanism can prevent such a scenario (recommended)
- ✓ Failure to abide by these restrictions can result in database consistency issues

HIGH AVAILABILITY - PROXY

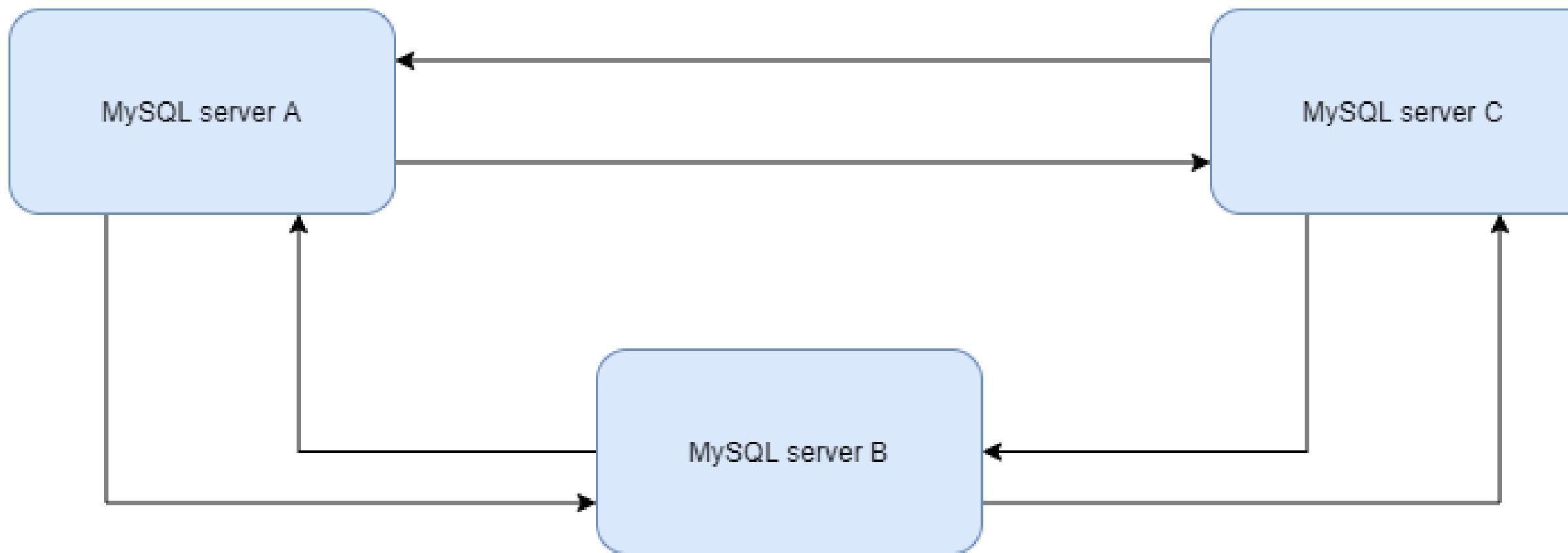
HA solution can also be implemented for Zabbix proxies, though it can add a significant HA configuration management overhead

HA for proxies can also be implemented by using Zabbix API scripts

```
{  
    "jsonrpc": "2.0",  
    "method": "host.massupdate",  
    "params": {  
        "hosts": [  
            {  
                "hostid": "10396"  
            },  
            {  
                "hostid": "10397"  
            }  
        ],  
        "proxy_hostid": 10398  
    },  
    "auth": "d9040f24a183638e00940fe64059420b",  
    "id": 1}
```

DATABASE REPLICATION

- ✓ Database replication can be used as a form of redundancy for the Zabbix DB
- ✓ Multiple replication approaches are supported (Master/Slave, Master/Master, Multi-master)



DATABASE PERFORMANCE TUNING

- ✓ In most scenarios using the default DB settings is not sufficient in such environments
- ✓ Hardware requirements should also be taken into account. The DB hosts should have sufficient memory and storage should be selected according to the I/O requirements (SSD/RAID setup)
- ✓ Ideally, consult a DBA for optimal DB configuration

The screenshot shows the PGtune website interface. At the top, there are navigation links: Home, How it works, and light. Below that is the PGtune logo, which is a green elephant icon followed by the text "PGTune". On the left, there's a section titled "Parameters of your system" with input fields for DB version (12), OS Type (Linux), DB Type (Mixed type of applications), Total Memory (RAM) (32 GB), Number of CPUs (8), Number of Connections (200), and Data Storage (SSD storage). Each input field has a "what is this?" link. On the right, there are two tabs: "postgresql.conf" (selected) and "ALTER SYSTEM". Under "postgresql.conf", there is explanatory text: "Add/modify this settings in **postgresql.conf** and restart database". Below this is a large block of PostgreSQL configuration code:

```
# DB Version: 12
# OS Type: linux
# DB Type: mixed
# Total Memory (RAM): 32 GB
# CPUs num: 8
# Connections num: 200
# Data Storage: ssd

max_connections = 200
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2GB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 200
work_mem = 5242kB
min_wal_size = 1GB
max_wal_size = 4GB
max_worker_processes = 8
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
max_parallel_maintenance_workers = 4
```

HISTORY TABLE PARTITIONING

- ✓ On large instances the Housekeeper cannot keep up with cleaning out history and trend tables in a timely fashion
- ✓ This has a negative effect on overall DB performance
- ✓ Partitioning is recommended for history/trend tables
- ✓ This is supported out of the box with PostgreSQL + TimescaleDB

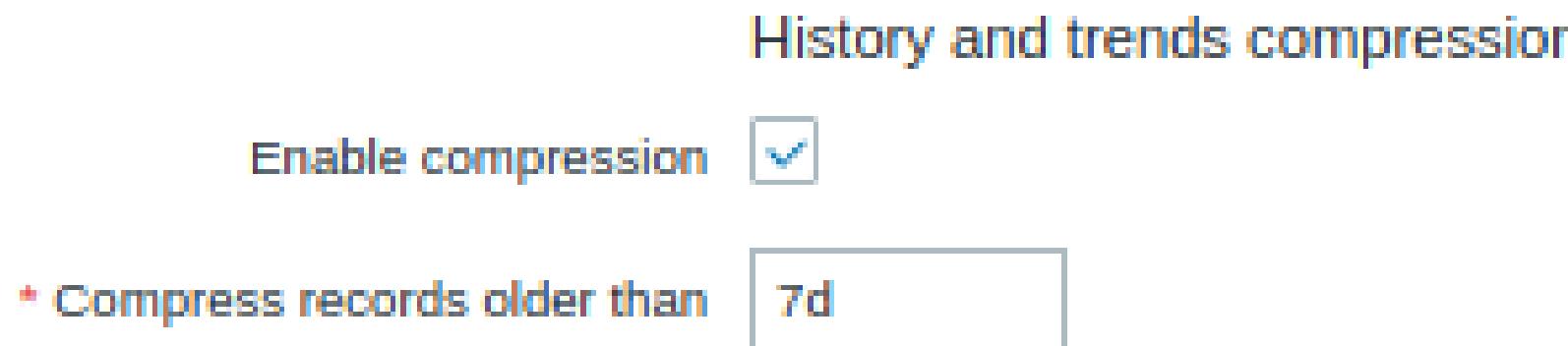
```
ALTER TABLE trends_uint PARTITION BY RANGE ( clock )
(PARTITION p2020_10 VALUES LESS THAN (UNIX_TIMESTAMP("2020-11-01 00:00:00")) ENGINE = InnoDB,
PARTITION p2020_11 VALUES LESS THAN (UNIX_TIMESTAMP("2020-12-01 00:00:00")) ENGINE = InnoDB,
PARTITION p2020_12 VALUES LESS THAN (UNIX_TIMESTAMP("2021-01-01 00:00:00")) ENGINE = InnoDB,
PARTITION p2021_01 VALUES LESS THAN (UNIX_TIMESTAMP("2021-02-01 00:00:00")) ENGINE = InnoDB,
PARTITION p2021_02 VALUES LESS THAN (UNIX_TIMESTAMP("2021-03-01 00:00:00")) ENGINE = InnoDB,
PARTITION p2021_03 VALUES LESS THAN (UNIX_TIMESTAMP("2021-04-01 00:00:00")) ENGINE = InnoDB);
```

- ✓ Community provided partitioning scripts are publicly available

<https://github.com/OpensourceICTSolutions/zabbix-mysql-partitioning-perl>

HISTORY TABLE PARTITIONING - TIMESCALEDB

- ✓ TimescaleDB plugin for PostgreSQL DB backends supports out of the box partitioning
- ✓ No need to rely on community scripts
- ✓ TimescaleDB parameter *chunk_time_interval* defines the partition chunk size.
- ✓ In addition, TimescaleDB provides the compression of history/trends
- ✓ Compressed data becomes read-only and cannot be changed post compression



TO-DO LIST

- ✓ Zabbix version with role support is deployed (>=5.2)
- ✓ Proxy servers are deployed in the client data centers
- ✓ Application and DB servers use HA/Replication solutions
- ✓ Partitioning has been implemented to improve the DB performance
- ✓ Throttling has been implemented to reduce the amount of incoming data
- ✓ Data base is tuned in accordance with the available hardware resources



MEETUP ONLINE '21

QUESTIONS?

ARTŪRS LONTONS
TECHNICAL SUPPORT ENGINEER
ZABBIX

ZABBIX



MEETUP ONLINE '21

THANK YOU!

ARTŪRS LONTONS

TECHNICAL SUPPORT ENGINEER

ZABBIX