

# NETWORK MONITORING PROJECT

ITMOS

DAVID DURAES VALADARES TRISTAN DE LA BORDA

## Contents

Introduction.....	3
1. The Mission for This Project.....	3
2. Group Members.....	3
3. Chosen Monitoring Solution.....	4
4. Project Methodology .....	4
4.1. Continuous Network Monitoring.....	4
4.2. Five Functions of Network Monitoring.....	4
5. Infrastructure Analysis and Inventory .....	5
5.1 Description of the Infrastructure.....	5
5.2 Inventory of Configuration Items (CI) .....	6
5.3 Monitoring Plan .....	6
6. Zabbix.....	7
6.1 License Model and Cost.....	7
6.2 Commercial Offerings (Optional) .....	7
6.3 Comparison with Other Monitoring Solutions.....	7
6.4 Who Uses Zabbix .....	7
7. Architecture and Working Principles of Zabbix.....	7
7.1 Zabbix Architecture Overview.....	7
7.2 Communication Flow.....	7
8. Implementation and Setup .....	8
8.1 Deployment Environment .....	8
8.2 Installation of Zabbix .....	8
8.3 Basic Configuration.....	9
8.4 Adding a Host .....	10
9. Dashboards .....	12
9.1 Creating a Dashboard .....	12
9.2 Dashboard Widgets Used.....	13
10. Our Dashboard's.....	14
10.1 Overview / NOC Dashboard.....	14
10.2 Network & Firewall Dashboard.....	14
10.3 Servers & Virtualization Dashboard.....	15
10.4 Storage (Synology) Dashboard .....	15

10.5	Printers & Consumables Dashboard .....	16
10.6	Zabbix Server / Global View Dashboard.....	16
11.	Advantages and Disadvantages of Zabbix .....	16
11.1	Advantages .....	16
11.2	Disadvantages .....	17
12.	Challenges faced .....	17
13.	Lessons Learned .....	17
14.	Testing and Validation .....	18
14.1	Host Availability Test (Zabbix Agent) .....	18
14.2	CPU Load Test.....	18
14.3	SNMP Monitoring Test .....	19
15.	Conclusion.....	19

## Introduction

Modern IT infrastructures are composed of devices and services that must remain continuously available, secure, and performant. To ensure service reliability and rapid incident response, organizations rely on infrastructure monitoring systems. This project focuses on the planning, installation, configuration, and validation of an infrastructure monitoring system using Zabbix.

### 1. The Mission for This Project

The objective of this project was to design and deploy a complete infrastructure monitoring solution capable of supervising network devices, servers, and services. The monitoring system had to be implemented in a virtualized environment and documented in a way that allows reproducibility.

### 2. Group Members

Due to the relatively small number of students in this BTS class, the project group consisted of:

- David Durães Valadares
- Tristan De La Borda

To ensure transparency and clarity regarding individual contributions, the responsibilities within the project were divided as follows:

#### David Durães Valadares

- Overall design and implementation of the monitoring solution
- Installation and configuration of the Zabbix server, MariaDB database, and web frontend
- Configuration of Zabbix agents on Linux servers
- Creation, customization, and final selection of dashboards used in the project
- Troubleshooting of monitoring issues
- Main author of the project documentation
- PowerPoint

#### Tristan De La Borda

- Provision of a Proxmox virtualization server used in the project environment
- Initial infrastructure information and partial documentation support
- Creation of preliminary dashboards
- Troubleshooting
- Demo

### 3. Chosen Monitoring Solution

After evaluating several network monitoring systems, Zabbix was selected as the monitoring platform for this project.

#### Reasons for Choosing Zabbix

- Enterprise-grade open-source solution
- No licensing limitations on hosts, metrics, or users
- Supports multiple monitoring methods (agent, SNMP, HTTP, API)
- Strong community support and extensive documentation
- Widely used in professional environments

Zabbix offers a complete monitoring stack, making it suitable for educational purposes and real-world infrastructure monitoring.

### 4. Project Methodology

#### 4.1. Continuous Network Monitoring

Continuous network monitoring refers to the automated and uninterrupted observation of network devices, servers, applications, and services to ensure availability, performance, and reliability.

Unlike periodic or manual checks, continuous monitoring collects data in real time, allowing immediate detection of faults, abnormal behavior, or performance degradation.

The primary goals of continuous monitoring are to:

- Minimize service downtime
- Detect incidents early
- Support proactive maintenance
- Improve overall infrastructure reliability

#### 4.2. Five Functions of Network Monitoring

##### Fault Management

Detects failures such as device outages or service interruptions and generates alerts.

##### Performance Management

Monitors performance metrics like CPU usage, memory usage, and network traffic.

##### Availability Management

Ensures that systems and services are reachable and operational.

##### Security Monitoring

Observes network activity and logs to detect potential security issues.

### **Capacity Planning**

Analyzes historical data to predict future resource requirements and prevent overload.

## **5. Infrastructure Analysis and Inventory**

### **5.1 Description of the Infrastructure**

The monitoring infrastructure was deployed on a Linux virtual machine hosted on a Proxmox server within the school network. Remote access to the Zabbix server was provided through a VPN connection, allowing configuration and monitoring from outside the school network.

## 5.2 Inventory of Configuration Items (CI)

The following configuration items were identified as critical services and components to be monitored:

- Firewall
- Network storage
- Virtualization hosts
- Application servers
- Network printers

## 5.3 Monitoring Plan

Host name	Device type	IP address	Host group	Monitoring method	Template used	Extra checks
FortiGate Firewall	Firewall / Gateway	10.0.0.1	Network	SNMP	FortiGate by SNMP	Interface traffic
Synology NAS	Network Storage	10.0.0.8	Storage	SNMP	Synology DiskStation by SNMP	RAID & disk health
Proxmox mitnick	Virtualization Host	10.0.0.100	Virtualization	HTTP API	Proxmox VE by HTTP	VM status
Proxmox snowden	Virtualization Host	10.0.0.101	Virtualization	HTTP API	Proxmox VE by HTTP	VM status
Proxmox thomas	Virtualization Host	10.0.0.102	Virtualization	HTTP API	Proxmox VE by HTTP	VM status
Web Server	Application Server	10.0.0.52	Servers	Agent + HTTP	Linux by Zabbix agent	Web scenario
Proxy Server	Proxy Server	10.0.0.54	Servers	Agent	Linux by Zabbix agent	Port check
Mail Server	Mail Server	10.0.0.58	Servers	Agent + SMTP	Linux by Zabbix agent	SMTP check
Pi-hole	DNS Server	10.0.0.62	Servers	Agent + DNS	Linux by Zabbix agent	DNS check
Printers	Network Printers	Various	Printers	SNMP	Generic Printer by SNMP	Toner levels

## 6. Zabbix

### 6.1 License Model and Cost

Zabbix is released under the GNU General Public License v2, making it completely free and open source.

There are no licensing fees for devices, metrics, or users.

### 6.2 Commercial Offerings (Optional)

Zabbix offers optional paid services such as:

- Professional support plans
- Training and certification
- Consulting and implementation services

These services are optional and not required for full functionality.

### 6.3 Comparison with Other Monitoring Solutions

Aspect	Zabbix	PRTG	Nagios XI	SolarWinds
<b>License</b>	Open source	Commercial	Commercial	Commercial
<b>Cost</b>	Free	Paid per sensor	Paid	Paid
<b>Scalability</b>	Excellent	Limited	Excellent	Excellent
<b>Customization</b>	Very high	Limited	High	Limited
<b>Learning curve</b>	Steep	Low	Steep	Moderate

### 6.4 Who Uses Zabbix

Zabbix is used by a wide range of organizations, including internet service providers, hosting companies, enterprises, educational institutions, and public sector organizations. It is commonly deployed to monitor large-scale infrastructures with thousands of devices.

## 7. Architecture and Working Principles of Zabbix

### 7.1 Zabbix Architecture Overview

- Zabbix Server: Central processing unit that evaluates data and triggers alerts
- Database: Stores configuration data, metrics, and events
- Web Frontend: User interface for configuration and visualization
- Zabbix Agent: Collects metrics from monitored hosts
- SNMP & Service Checks: Used for agentless monitoring

### 7.2 Communication Flow

1. Monitoring data is collected from agents, SNMP devices, or HTTP checks
2. Data is sent to the Zabbix server
3. Triggers evaluate thresholds
4. Alerts are generated
5. Notifications are sent if configured

## 8. Implementation and Setup

### 8.1 Deployment Environment

- Virtual machine hosted on Proxmox
- Ubuntu Server operating system
- MariaDB database
- VPN access to BTS network

### 8.2 Installation of Zabbix

1. Update the system packages:

```
sudo apt update && sudo apt upgrade -y
```

2. Install required dependencies:

```
sudo apt install -y wget curl gnupg2 software-properties-common
```

3. Install MariaDB database server:

```
sudo apt install -y mariadb-server
```

4. Secure and start MariaDB:

```
sudo systemctl enable mariadb
```

```
sudo systemctl start mariadb
```

5. Add the official Zabbix repository:

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_7.0-1+ubuntu22.04_all.deb
```

```
sudo dpkg -i zabbix-release_7.0-1+ubuntu22.04_all.deb
```

```
sudo apt update
```

6. Install Zabbix server, frontend, and agent:

```
sudo apt install -y zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf  
zabbix-agent
```

7. Create the Zabbix database and user:

```
sudo mysql  
  
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;  
  
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'password';  
  
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
```

```
FLUSH PRIVILEGES;
EXIT;
```

- Configure the Zabbix server database connection:

```
sudo nano /etc/zabbix/zabbix_server.conf
```

SET:

```
DBPassword=password
```

- Start and enable Zabbix services:

```
sudo systemctl restart zabbix-server zabbix-agent apache2
sudo systemctl enable zabbix-server zabbix-agent apache2
```

- Access the web interface:

```
http://<ZABBIX_SERVER_IP>/zabbix
```

### 8.3 Basic Configuration

- Log in to the Zabbix web interface using the default credentials
- Configure the correct timezone in the frontend:
- Create host groups to organize monitored devices:

New host group

\* Group name

Add Cancel

Create groups such as:

- Network
- Servers
- Storage
- Virtualization
- Printers

## 8.4 Adding a Host

### 1. Create a Host:

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
better Fortigate Firewall	Items 400	Triggers 135	Graphs 51	Discovery 9	Web	10.0.0.1:161		FortiGate by SNMP	Enabled	SNMP	None		
better Mail Server	Items 44	Triggers 16	Graphs 8	Discovery 3	Web	10.0.0.58-10050		Linux by Zabbix agent	Enabled	ZBX SNMP	None		
better Pi-hole	Items 44	Triggers 16	Graphs 8	Discovery 3	Web	10.0.0.62-10050		Linux by Zabbix agent	Enabled	ZBX SNMP	None		
better Printer A207	Items 12	Triggers 6	Graphs	Discovery 2	Web	10.0.95.1-161		Network Generic Device by SNMP	Enabled	SNMP	None		
better Printer BT1 BT2	Items 3	Triggers 6	Graphs	Discovery	Web	127.0.0.1:161		brother Printers s	Enabled	SNMP	None		
better Printer C302	Items 12	Triggers 6	Graphs	Discovery 2	Web	10.0.96.2-161		Network Generic Device by SNMP	Enabled	SNMP	None		

## 2. Configure the Host.

The screenshot shows the 'New host' configuration dialog in Zabbix. The 'Host' tab is active. The 'Host name' field is mandatory and highlighted with a red asterisk. Other fields include 'Visible name', 'Templates' (with a search bar and 'Select' button), 'Host groups' (with a search bar and 'Select' button), and 'Interfaces' (which says 'No interfaces are defined'). Below these are sections for 'Description' (with an 'Add' button) and 'Monitoring' (with tabs for 'Server', 'Proxy', and 'Proxy group', and a checked 'Enabled' checkbox). At the bottom are 'Add' and 'Cancel' buttons.

Enter the host name:

- Example: Synology NAS

Assign the host to a host group:

- Example: Storage

Configure the interface depending on the monitoring method:

- SNMP
  - Interface type: SNMP
  - IP address: <device IP>
  - Port: 161
- AGENT
  - Interface type: Agent
  - IP address: <host IP>
  - Port: 10050

Assign a template to the host:

- Examples of build in templates:
  - Linux by Zabbix agent
  - Synology DiskStation by SNMP
  - Generic Network Device by SNMP

Zabbix also allows the import of community templates from GitHub, providing extended monitoring support for devices not covered by the built-in templates.

<https://github.com/zabbix/community-templates>

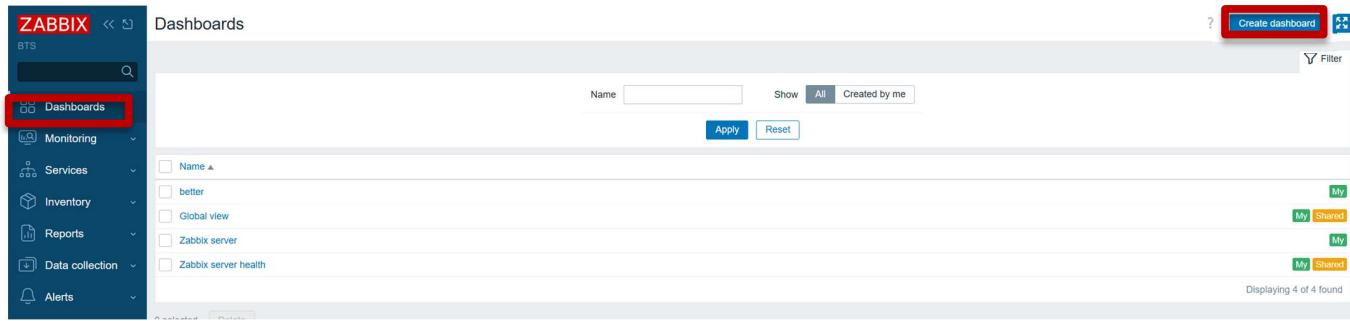
Click Add to create the host.

## 9. Dashboards

Dashboards in Zabbix provide a centralized visual overview of the monitored infrastructure. They allow administrators to quickly assess the overall health, availability, and performance of hosts and services.

### 9.1 Creating a Dashboard

#### 1. Create Dashboard



#### 2. Enter a dashboard name:

##### Dashboard properties

\* Owner

\* Name

Default page display period

Start slideshow automatically

#### 3. Click Add widget to populate the dashboard.



## 9.2 Dashboard Widgets Used

The following widgets were added to provide a clear overview:

- Problems widget  
Displays active problems and alerts across all monitored hosts.
- Host availability widget  
Shows whether hosts are online or unreachable.
- Graph widgets  
Displays CPU usage, memory usage, disk space, and network traffic for selected hosts.
- Latest data widget  
Shows recent metric values for critical services and systems.
- Web monitoring widget  
Displays the status of HTTP/HTTPS web scenarios.

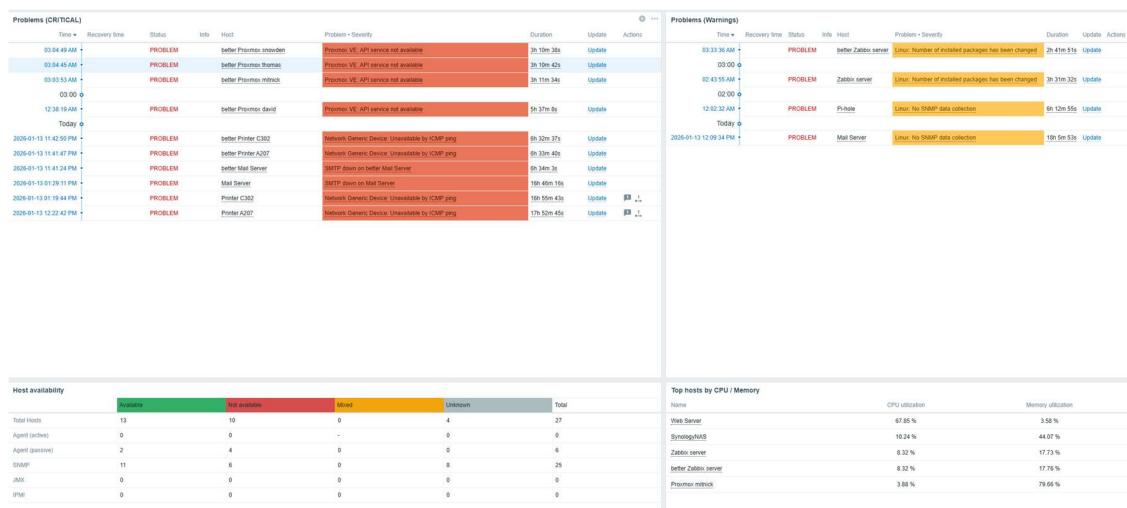
## 10. Our Dashboard's

### 10.1 Overview / NOC Dashboard

This dashboard provides a global real-time view of the entire infrastructure.

It shows critical and warning problems, overall host availability, and the top hosts by CPU and memory usage.

It is mainly used by administrators to quickly detect outages and prioritize incidents.

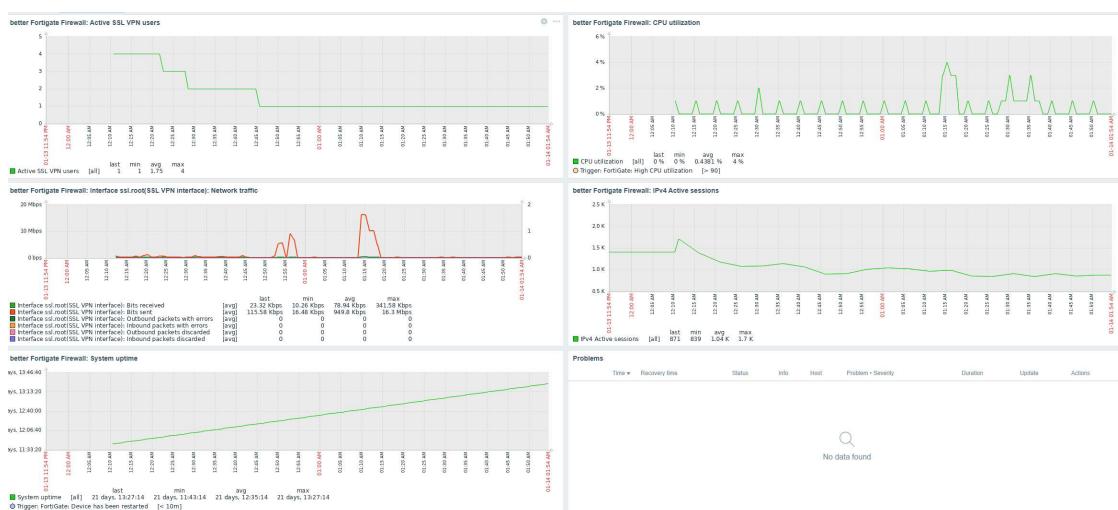


### 10.2 Network & Firewall Dashboard

This dashboard focuses on the FortiGate firewall.

It displays CPU utilization, active SSL VPN users, network traffic on the VPN interface, active IPv4 sessions, and system uptime.

Its purpose is to monitor network load, VPN usage, and firewall health.



### 10.3 Servers & Virtualization Dashboard

This dashboard monitors Linux servers and Proxmox virtualization hosts.

It highlights Zabbix agent availability, Proxmox API status, and server-related problems such as CPU, memory, network, and service failures.

It helps detect server outages and virtualization issues quickly.

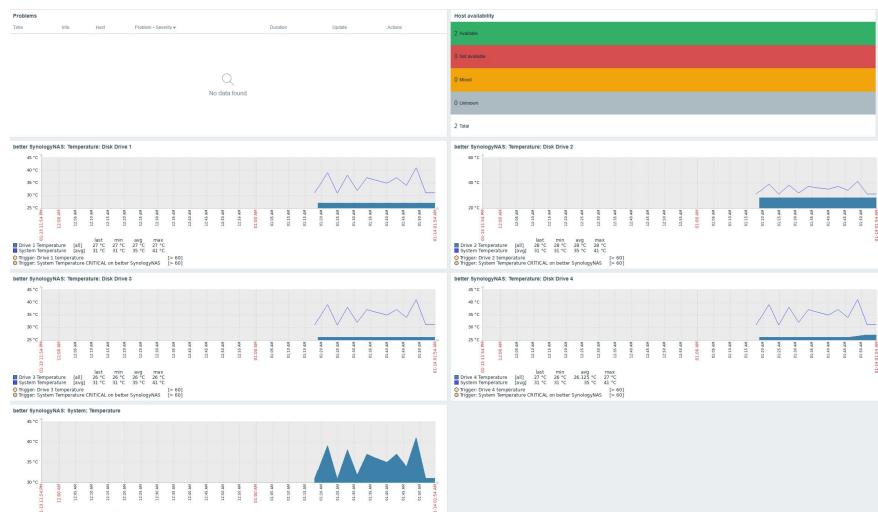
Problems (Servers + Virtualization)		Status	Info	Host	Problem + Severity	Duration	Update	Actions
03:47:43 AM	•	PROBLEM		better Proxy Server	Linux: Zabbix agent is not available (for 3m)	2h 35m 11s	Update	
03:47:32 AM	•	PROBLEM		better Pi-hole	Linux: Zabbix agent is not available (for 3m)	2h 35m 22s	Update	
03:47:21 AM	•	PROBLEM		better Mail Server	Linux: Zabbix agent is not available (for 3m)	2h 35m 33s	Update	
03:45:54 AM	•	PROBLEM		better Web Server	Linux: Zabbix agent is not available (for 3m)	2h 37m	Update	
03:44:49 AM	•	PROBLEM		better Proxmox snowden	Proxmox VE: API service not available	3h 18m 5s	Update	
03:44:49 AM	•	PROBLEM		better Proxmox thomas	Proxmox VE: API service not available	3h 18m 9s	Update	
03:33:51 AM	•	PROBLEM		better Proxmox mitnick	Proxmox VE: API service not available	3h 19m 1s	Update	
03:00	•							
12:38:19 AM	•	PROBLEM		better Proxmox david	Proxmox VE: API service not available	5h 44m 35s	Update	
12:02:32 AM	•	PROBLEM		Pi-hole	Linux: No SNMP data collection	6h 20m 22s	Update	
TODAY	•							
2020-01-13 04:41:24 PM	•	PROBLEM		better Mail Server	SMTP down on better Mail Server	6h 41m 30s	Update	
2020-01-13 04:48:45 PM	•	PROBLEM		Proxmox mitnick	Linux: interface lan125910 Link down	13h 56m 9s	Update	
2020-01-13 01:29:11 PM	•	PROBLEM		Mail Server	SMTP down on Mail Server	16h 53m 43s	Update	
2020-01-13 12:09:34 PM	•	PROBLEM		Mail Server	Linux: No SNMP data collection	18h 13m 20s	Update	

### 10.4 Storage (Synology) Dashboard

This dashboard is dedicated to the Synology NAS.

It shows disk temperatures, system temperature, and host availability using SNMP.

The goal is to detect hardware risks, such as overheating disks, before failures occur.



## 10.5 Printers & Consumables Dashboard

This dashboard monitors network printers using SNMP and ICMP.

It shows printer availability, connectivity issues, and provides a base for toner and page counter monitoring.

It ensures printers remain reachable for users.

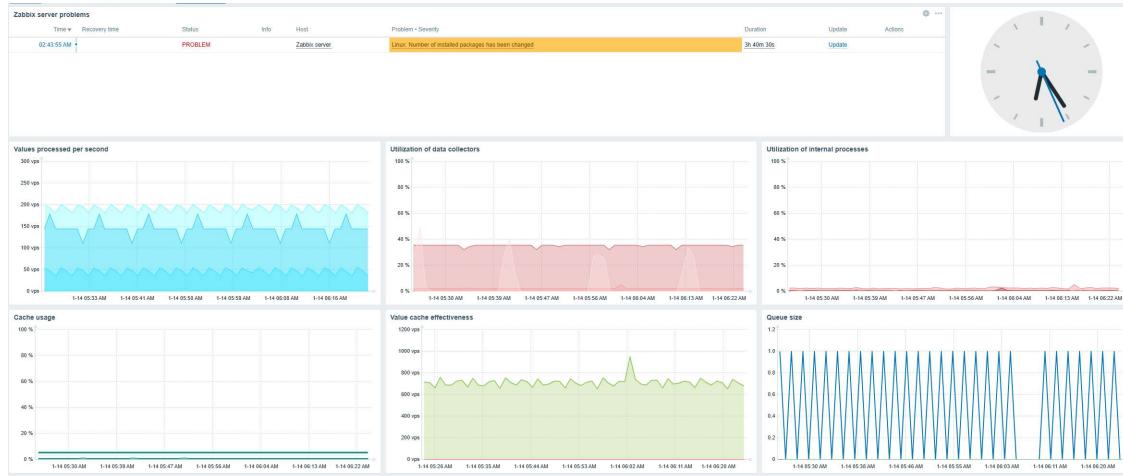
Time	Recovery time	Status	Info	Host	Problem + Severity	Duration	Update	Actions
2026-01-13 11:42:50 PM		PROBLEM	better Printer C302		Network Generic Device: Unavailable by ICMP ping	6h 40m 58s	Update	
2026-01-13 11:41:47 PM		PROBLEM	better Printer A207		Network Generic Device: Unavailable by ICMP ping	6h 42m 1s	Update	
2026-01-13 01:19:44 AM		PROBLEM	Printer C302		Network Generic Device: Unavailable by ICMP ping	17h 4m 4s	Update	
2026-01-13 12:22:42 PM		PROBLEM	Printer A207		Network Generic Device: Unavailable by ICMP ping	18h 1m 6s	Update	

## 10.6 Zabbix Server / Global View Dashboard

This dashboard monitors Zabbix itself.

It displays values processed per second, data collector usage, queue size, cache usage, and current Zabbix problems.

Its purpose is to verify that the monitoring system is healthy and not overloaded.



## 11. Advantages and Disadvantages of Zabbix

### 11.1 Advantages

- Fully open-source and free
- Highly scalable and customizable
- Supports SNMP, agents, and service checks
- Large community and extensive documentation

## 11.2 Disadvantages

- Steep learning curve for beginners
- Initial setup can be complex
- Interface may feel overwhelming for small environments

## 12. Challenges faced

During the implementation of the monitoring infrastructure, several challenges were encountered that required troubleshooting and adaptation.

One of the main challenges was SNMP configuration. Some devices had SNMP enabled but did not respond to queries due to incorrect community strings, access restrictions, or firewall rules. This required additional testing using tools such as snmpwalk and snmpget to verify connectivity and permissions.

Another difficulty was network access via VPN. Because the monitoring environment was accessed remotely through a VPN, latency and routing issues occasionally affected connectivity and delayed monitoring checks. This was especially noticeable when testing service availability and response times.

Proxmox API integration also presented challenges. Access to the Proxmox API required proper user roles and API tokens. Misconfigured permissions initially caused API checks to fail, resulting in “API service not available” alerts until the correct privileges were assigned.

The initial configuration of Zabbix itself was another challenge. The platform has a steep learning curve, and understanding the relationship between hosts, templates, items, triggers, and dashboards required time and experimentation.

Additionally, some built-in templates did not perfectly match all devices, particularly printers and network equipment. This led to the exploration and import of community templates from GitHub, as well as manual adjustments to monitoring items.

Despite these challenges, each issue contributed to a better understanding of real-world monitoring environments and strengthened troubleshooting skills.

## 13. Lessons Learned

This project improved our understanding of:

- Network monitoring concepts and best practices
- SNMP and agent-based monitoring
- Infrastructure planning and fault detection
- Real-world troubleshooting and alert handling

## 14. Testing and Validation

The monitoring system was tested by deliberately generating incidents in order to validate detection, alerting, and recovery mechanisms. The following tests were performed on monitored hosts and services.

### 14.1 Host Availability Test (Zabbix Agent)

**Objective:** Verify that Zabbix detects host unavailability.

**Action:** The Zabbix agent service was stopped on a monitored Linux server using the following command:

```
sudo systemctl stop zabbix-agent
```

**Expected Result:** The host should become unreachable, and a problem should be generated by Zabbix.

**Observed Result:** Within a short delay, Zabbix detected the agent as unavailable and raised an alert visible in the Problems view and dashboards.

**Recovery:** The agent was restarted using:

```
sudo systemctl start zabbix-agent
```

The host status returned automatically to an OK state.

### 14.2 CPU Load Test

**Objective:** Validate performance monitoring and trigger thresholds.

**Action:** Artificial CPU load was generated on a Linux server using a stress command:

```
yes > /dev/null &
```

This command was executed multiple times to increase CPU usage.

**Expected Result:** CPU utilization should exceed the defined trigger threshold, generating a performance warning.

**Observed Result:** Zabbix detected the high CPU usage and displayed a warning on the server dashboard and Problems view.

**Recovery:** The stress processes were stopped, and CPU usage returned to normal. The alert was automatically cleared.

The stress processes were stopped, and CPU usage returned to normal. The alert was automatically cleared.

#### 14.3 SNMP Monitoring Test

**Objective:** Verify SNMP-based monitoring and error detection.

**Action:** The SNMP community string on a monitored device was temporarily modified to an incorrect value.

**Expected Result:** SNMP checks should fail and generate monitoring errors.

**Observed Result:** Zabbix reported SNMP item failures and raised alerts indicating communication issues with the device.

**Recovery:** The correct community string was restored, and SNMP monitoring resumed normally.

These tests confirmed that the monitoring system correctly detects incidents, generates alerts, and automatically recovers once normal operating conditions are restored.

#### 15. Conclusion

Zabbix proved to be a powerful and flexible monitoring solution suitable for enterprise environments.

Despite initial complexity, it offers deep visibility into infrastructure health and performance.

The project successfully demonstrated how continuous monitoring improves availability, reliability, and incident response.