# AUTOMATION FOR INCORPORATING ASSETS INTO MONITORING TOOLS

**A Degree Thesis**

**Submitted to the Faculty of the**

**Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona**

**Universitat Politècnica de Catalunya**

**by**

**Héctor Arroyo Recio**

**In partial fulfilment**

**of the requirements for the degree in**

**TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING**

**Advisor:**

**Israel Martin Escalona**

**Ricardo Marín Vinuesa**

**Barcelona, June 2022**

# Abstract

The project consists of an analysis of the different monitoring tools and automation functions in them to find the best tool for incorporating assets. These tools have been tested in a controlled environment to determine their capabilities.

It all started with a study of automation needs and a search for monitoring tools. Subsequently, I made the choice of the tool according to established criteria and an adjusted result was obtained, so it was decided to incorporate the second-best option. Then, the configuration and implementation of both were carried out in a controlled environment and a test of both was proposed and executed.

Finally, after analysing and testing the two best options, it has been seen that both Nagios Core and Zabbix have offered similar results, but it has been determined that the best option for implementation in the client network is to meet the established needs is Zabbix.

# Resum

El projecte consisteix en una anàlisi de les diferents eines de monitorització i funcions d'automatització per trobar la millor eina per a la incorporació d'actius. Aquestes eines s'han provat en un entorn controlat per determinar-ne les capacitats.

Tot va començar amb un estudi de necessitats d'automatització i una cerca d'eines de monitorització. Posteriorment, vaig fer l'elecció de l'eina segons criteris establerts i es va obtenir un resultat ajustat, per la qual cosa es va decidir incorporar-hi la segona millor opció. Després, es va realitzar la configuració i implementació de totes dues en un ambient controlat i es va proposar i executar un testeig d’ambdues.

Finalment, després d'analitzar i testejar les dues millors opcions, s'ha vist que tant Nagios Core com Zabbix han ofert resultats similars, però s'ha determinat que la millor opció d'implementació a la xarxa del client per cobrir les necessitats establertes és Zabbix.

# Resumen

El proyecto consiste en un análisis de las diferentes herramientas de monitorización y funciones de automatización de las mismas para encontrar la mejor herramienta para la incorporación de activos. Estas herramientas se han probado en un entorno controlado para determinar sus capacidades.

Todo comenzó con un estudio de necesidades de automatización y una búsqueda de herramientas de monitoreo. Posteriormente, realicé la elección de la herramienta según criterios establecidos y se obtuvo un resultado ajustado, por lo que se decidió incorporar la segunda mejor opción. Luego, se realizó la configuración e implementación de ambas en un ambiente controlado y se propuso y ejecutó un testeo para ambas.

Finalmente, tras analizar y testear las dos mejores opciones, se ha visto que tanto Nagios Core como Zabbix han ofrecido resultados similares, pero se ha determinado que la mejor opción de implementación en la red del cliente para cubrir las necesidades establecidas es Zabbix.

# Agradecimientos

A mi familia, pareja y amigos,
por cuidarme, apoyarme y soportarme
en los momentos difíciles.

A mis compañeros de proyecto,
por echarme una mano
a la hora de decidir temática,
por ayudarme cuando lo necesitaba
y por compartir momentos de estrés y alegría juntos.

A mi tutor en la empresa, Ricardo Marín,
por preocuparse siempre por mí,
por ayudarme a decidir temática, en la redacción
y en la toma de decisiones importantes.

A mi tutor de la Universidad, Israel Martin,
por todas las horas dedicadas a reuniones,
por los comentarios y ayudas recibidas,
por responder tan detalladamente a mis correos
y por guiarme hasta el final.

## Revision history and approval record

| Revision | Date | Purpose |
|---|---|---|
| 0 | 21/09/2021 | Document creation |
| 1 | 10/12/2021 | Document revision with Project Supervisor |
| 2 | 23/12/2021 | Document revision with Project Supervisor |
| 3 | 29/04/2022 | Document revision with Project Supervisor |
| 4 | 06/05/2022 | Document revision with Project Supervisor |
| 5 | 13/05/2022 | Document revision with Project Supervisor |
| 6 | 27/05/2022 | Document revision with Project Supervisor |
| 7 | 10/06/2022 | Document revision with Project Supervisor |
| 8 | 17/06/2022 | Document revision with Project Supervisor |
| 9 | 20/06/2022 | Document final revision with Project Supervisor |

DOCUMENT DISTRIBUTION LIST

| Name | e-mail |
|---|---|
| Héctor Arroyo Recio | hector.arroyo@estudiantat.upc.edu |
| Ricardo Marín Vinuesa | rmarin@gmv.com |
| Israel Martin Escalona | israel.martin@upc.edu |

| | | | |
|---|---|---|---|
| Written by: | | Reviewed and approved by: | |
| Date | 21/09/2021 | Date | 20/06/2022 |
| Name | Héctor Arroyo Recio | Name | Israel Martín Escalona |
| Position | Project Author | Position | Project Supervisor |

# Table of contents

## List of Figures

# List of Tables

# 1. Introduction

## 1.1. Current Drawbacks of Open-Source Monitoring Tools

Monitoring a large network is not an easy task. On a small scale, it is relatively easy to add devices manually and study each technology to customize the parameters, but when the network consists of hundreds or thousands of devices, the task becomes tedious and takes a long time to complete. In my case, I work on a client infrastructure with a considerable size and an integration of new devices is coming up. Therefore, it is necessary to develop an automation solution to reduce the hours and resources spent on adding new devices to a monitoring tool that meets a series of requirements.

Before monitoring, we must assess what protocols we want to use and what elements we want to take into account. Normally it is necessary to monitor the hardware of a device, especially its most important elements such as the processor, memory, or hard drive. In some equipment, the bandwidth used or the status of its services must also be taken into account, especially in web servers. It is also vital to monitor the communication status of the various devices, such as checking if the device is still up using Internet Control Message Protocol (ICMP), making sure Simple Network Management Protocol (SNMP) is running to maintain active monitoring or connection status via Secure Shell (SSH) to be able to access devices remotely.

In our environment, it would also be necessary and would reduce the workload to monitor the status of the services and resources at all times. For example in the company infrastructure, we can find some linked devices that will form a cluster, and each cluster has different services and critical resources running. For this reason, to verify their correct operation, they must be monitored and notify us of any anomaly or problem, whether one of them stops or one of the nodes (devices that make up a cluster) is down.

The vast majority of these metrics can be monitored using SNMP, but this protocol poses the problem of searching for their Object Identifiers (OIDs) and seeing which ones we can use, as well as checking that they are valid for our team and that they correctly display the data we are looking for. In addition, conversions of the data obtained from these OIDs are often required to obtain the metrics we are looking for (for example, converting bytes/second to bits/second or converting numerical resource utilization to a percentage).

There are some metrics that cannot be monitored by using SNMP because they are no existing OIDs to monitor them. In this case, we can use SSH protocol to connect with devices, send commands and get command response as a metric. Normally, our team has to connect to every device individually and manually check every resource, wasting around one hour daily to make a check of existing devices. But if we automate this process and incorporate this check into a monitoring tool, we can reduce this time and make it not necessary to do this check daily. These SSH responses have to be transformed to obtain valid values to make a graph or to show some status. These transformations normally need to use Regular Expressions (Regex) or similar software to work.

Finally, we will need a monitoring tool that allows us to carry out all the actions we want, but to avoid additional costs in the project I have decided to use a freeware tool that works on an open-source Operating System (OS). Due to the multitude of options that we will have, a preliminary study will have to be carried out to obtain the tools that best adapt to the project and decide which of them is more appropriate.

Taking these problems into account, my TFG proposes a solution that will simplify the process of incorporating equipment and its different metrics into a monitoring tool appropriate to the objective and requirements that I will present in the following sections.

## 1.2. Project Goals

The main objective of this project is to develop an automation system through scripts or small pieces of software to add network devices and services to an open-source monitoring tool.

To meet the Project goal I will follow this roadmap:

a. *State of the Art* on monitoring tools: Carry out a study of some open-source monitoring tools that meet the established requirements in a certain way.

b. Provide a methodology to select the most suitable monitoring tool: Establishment of selection criteria for the monitoring tool and its subsequent selection for the Proof of Concept.

c. Design of automation system for the incorporation of assets in monitoring tools: Make an initial design, then optimize and check it with the supervisor

d. Proof of Concept implementation: Implement the code, design the validation tests and execute them in the laboratory.

## 1.3. Methods and Procedures

I have based my project methodology on the agile model since this allows a good follow-up and control of the project development. I have used an internal company tool called Kanban (similar to Trello) to plan the milestones. The following columns have been used on the platform to display the status of the tasks:

- To do: Tasks that can be started.
- OnGoing: Tasks that are being carried out.
- Stopped: Tasks that are on hold due to external needs.
- Done: Tasks that are already finished.

Follow-up with the tutor has been through meetings approximately every month and a half. These meetings have been held using Google Meet and in them, the milestones achieved have been reviewed and doubts have been resolved regarding the tasks carried out, the tasks to be carried out or the structure of the document and where to place the different sections. The email has also been used to answer specific questions and schedule meetings.

The Tools, the programs and the code have been stored on a server provided by the company with the aim of facilitating the implementation of the project and maintaining the client's privacy. For security reasons, the project has been developed and tested only using company equipment with the aim of guaranteeing the integrity of the code and of the environments and equipment with which they have been tested.

## 1.4. Stakeholders

The Stakeholders of this Project are:

Project Manager: It will be in charge of supervising the project, guiding and advising the Project Developer to ensure that the tasks and milestones are carried out correctly.

System Administrator: It will be in charge of the administration and maintenance of the customer infrastructure.

Customer: They will be able to use the monitoring system and the automation developed in the project to check the status of the different elements of their infrastructure.

## 1.5. Requirements

The monitoring system that is proposed in the framework of this project should fulfil the following requirements (detailed on next page):

| Type of Requirement | Requirement | Important | ID |
|---|---|---|---|
| **Project Functional Requirements** | Provide dashboards | High | F1 |
| | Provide device discovery and integration tool | High | F2 |
| | Show critical measurements | Medium | F3 |
| | Set up threshold-triggered alarms based on hardware capabilities | Low | F4 |
| | Send email alerts | Low | F5 |
| | Add parameters and graphs by using available OIDs | High | F6 |
| | Have a list of users with different roles | Medium | F7 |
| | Track configuration changes | Medium | F8 |
| **Project Non-Functional Requirements** | Work with agentless monitoring | High | NF1 |
| | Programmable based on scripts or small pieces of software | High | NF2 |
| | Support web interface | Medium | NF3 |
| | Allow basic security methods | High | NF4 |
| | Collect and filter log file entries | High | NF5 |
| | Deployment of the system in a virtual environment based on an open-source operating system | High | NF6 |
| | Development of the system using only freeware software | High | NF7 |
| | Compatible with existing operating systems | High | NF8 |

*Table 1. Requirements*

Project Functional Requirements

- Provide dashboards: It must allow the use of widgets that shows:
  - Advanced Graphs: It must allow lines and bars graphs, axis modification, set custom time period and shows the average and maximum displayed values.
  - Single Item Value: It must show item value with custom text before/after them.
- Provide device discovery and integration tool: It must provide a device discovery tool which can incorporate device interfaces automatically and also include test options for ICMP, SNMP and SSH protocols.
- Show critical measurements: It must show at least Central Processing Unit (CPU) and Memory values or utilization.
- Set up threshold-triggered alarms based on hardware capabilities: It must get the maximum possible value and sets alarms at a certain percentage.
- Send email alerts: It must allow sending alerts by email when some action has occurred.
- Add parameters and graphs by using available OIDs: It must discover the available OIDs and add the desired ones for monitoring and graphing.
- Have a list of users with different roles: It must support at least the admin, editor, and viewer roles, all of which are assigned to a particular project.
- Track configuration changes: It must provide a history of changes identifying which user has made it.

Project Non-Functional Requirements

- Work with agentless monitoring: It must allow agentless monitoring using SNMP (versions 2 or 3), SSH and ICMP protocols.
- Programmable based on scripts or small pieces of software: It must allow script execution in some areas:
  - Frontend Locations: It must allow dashboard creation and manipulation.
  - Events: It must allow event interaction.
  - Specific actions: It must allow action operation and communication with devices.
- Support web interface: It must give a friendly, easy-to-use and aesthetic web interface.
- Allow basic security methods: It must use Secure Sockets Layer (SSL) certificates on the server.
- Collect and filter log file entries: It must offer Security Information and Event Management (SIEM) basic functionalities such as log and incident management. This functionality must be done through the QRadar Community Edition tool at the request of the System Administrator.
- Deployment of the system in a virtual environment based on an open-source operating system: It must be installed in an open-source OS like Linux or other Unix-based OS.
- Development of the system using only freeware software: Monitoring tool software must be freeware software.

- Compatible with existing OS: It must provide monitoring for Windows, CentOS and Debian.

## 1.6. Work Plan

Work Plan is explained in *APPENDIX J.*

## 1.7. Risks and Action Plans

Risks

After introducing the Work Plan and Tasks, I have detected a set of risks, which I will comment on below:

- Infrastructure Knowledge: Initially, I do not know the infrastructure in great detail and this means that the estimated time for the study and subsequent creation of the laboratory environment may fall short at some point and you need to spend more time on it. Risk impact is medium and probability is medium, so the risk is medium.
- Implementation Problems: Problems may arise in the implementation and make some features require more development or configuration time. Risk impact is medium and probability is medium, so the risk is medium.
- Lab Server Issues: Because a customer-provided server will be used to create the lab environment, problems with it could occur and parts of the environment or the entire environment could be lost. Risk impact is low and probability is low, so the risk is low.

Action Plans

The alternatives and action plans for the obtained risks are:

- Infrastructure Knowledge: First, the time to be dedicated to the study task has been overestimated. If it happens, as more time will have to be dedicated to the study of the infrastructure, the generation of the laboratory environment will be simplified to a model that is easier to implement, but which is still sufficient for the development of the project.
- Problems in Implementation: If this happens, you will consider leaving the test in the client environment for future development and focus on getting the implementation working well in the testing phase.
- Lab Server Issues: Weekly backups of the laboratory environment and its virtual machines will be made, so in most cases, a large part of the process carried out could be recovered and if not, it could be redone using the documentation written during the days after the last backup.

# 2. State of the Art of the Technology Used in this Thesis

## 2.1. Introduction

The goal of this *State of the Art* is to investigate some of the main open-source monitoring tools that exist and show their main characteristics. With this, I intend to make known the existing options to, later, make a comparison using the requirements and end up developing the project with the tool that best suits them and the objective of the project.

Below, I will present the possible monitoring tools and then I will evaluate to what level they meet the requirements to rule out the least suitable for the development of the project.

I also want to add that the SNMP, SSH and SYSLOG protocols will be used as mechanisms for obtaining information and the resulting tool will be used to display this information.

## 2.2. Nagios Core

Nagios Core [1] is an open-source system and network monitoring application. It is used to monitor specified hosts and services and alert the network administrator when something goes wrong or if everything is fine again. It has several Application Programming Interfaces (APIs) [2] that are used to make additional tasks. It is also implemented as a daemon and is designed to run natively on Linux and Unix Operating Systems(OS).

Some of the Nagios Core features include are:

- Monitoring of network services (HTTP, SSH, SNMP, ICMP, etc.)
- Monitoring of host resources (ram utilization, processor load, disk usage, etc.)
- Plugin design that allows users to develop service checks
- Parallelized service checks
- Ability to define network host hierarchy, allowing detection of and the distinction between hosts that are down and those that are unreachable
- Contact notifications when service or host problems occur and get resolved
- Ability to define event handlers to be run during events for proactive problem resolution
- Web interface for viewing current network status, notification and problem history, and other features.

## 2.3. Prometheus + Grafana

Prometheus [3] is an open-source system monitoring and alerting toolkit built at SoundCloud. Now is a standalone open-source project and is maintained independently of any company. It is used to collect and store its metrics as time-series data. To show the graphs, we need to use a complementary software called Grafana, which is also open-source and has a lot of applications with graphs and more data visualization options.

Some of the Prometheus features include are:

- Multi-dimensional data model with time series data identified by metric name and key/value pairs
- PromQL, a specific query language for communication
- Single server nodes are self-contained, so they are not dependent on distributed storage
- Time series collection over HTTP pulls
- Pushing time series using an intermediary gateway

### 2.3.1. Grafana

Grafana [4] is open-source software that allows you to query, visualize and alert your stored data. You can create, explore and share all of your data through dashboards. It will be used to visualize the data stored by Prometheus and OpenNMS Horizon.

Some of the Grafana features include are:

- Explore metrics, logs and traces: Explore data from queries. Split view for comparing time ranges, queries and data sources.
- Alerts: Can use alerts sent through different alert notification options. It can also provide alert rules for metrics.
- Annotations: This feature shows up as a graph marker and it can be used for correlating data in case something goes wrong.
- Dashboard variables: Allow you to create dashboards that can be used for different cases. These dashboards can be shared easily.
- Configuration: Covers both configuration files and environment variables. Gives several configuration options.
- Imports: Get dashboards and plugins from the official library.
- Authentication: Supports different authentication types.
- Provisioning: Allows automation through scripts.
- Permissions: Have different permissions for folders and dashboards depending on users.

## 2.4. Zabbix

Zabbix [5] is an enterprise-class open source distributed monitoring tool created by Alexei Vladishev, and currently developed and supported by Zabbix SIA. It can monitor several network parameters and the health and integrity of elements like servers, virtual machines, databases, cloud and more. It uses a notification mechanism that allows users to configure alerts triggered by events, allowing a fast reaction to server problems. Zabbix offers reporting and data visualization features based on stored data.

It also supports both polling and trapping. All its reports, statistics and configurations are accessed through a web-based frontend, making it easier to access it from any location.

Some of the Zabbix features include are [6]:

- Data gathering: Availability and performance checks, support for SNMP monitoring and data gathering at custom intervals.
- Threshold definitions: Can define triggers referencing values from the database.
- Alert configuration: Custom notifications can be used and simplified by using macro variables. Actions as remote commands can be automatized.
- Graphs: Monitored metrics are graphed in real-time using the built-in graphing functionality.
- Web monitoring: Follow interaction on a website and check for functionality and response time.
- Visualization options: Create custom graphs combining multiple items, network maps, slideshows overview, reports and view of monitored resources.
- Data storage: Data stored in a database and configurable history.
- Configuration: Easily add monitored devices as hosts and picked them up for monitoring once in the database. Also can apply templates to them.

- Network discovery: Automatic device discovery on network and agent autoregistration. Discovery of file systems, network interfaces and SNMP OIDs.
- Web Interface: PHP web-based frontend accessible from anywhere.
- Zabbix API: Provides APIs for additional features.
- Permissions: Secure user login with different assigned roles.
- Binary daemons: Written in C to lighten resources and easily portable.
- Ready for complex environments: Allows remote monitoring by using Zabbix proxy.

## 2.5. OpenNMS Horizon + Grafana

OpenNMS Horizon [7] is an open-source solution to visualize and monitor everything on your networks. It offers comprehensive fault, performance, traffic monitoring, and alarm generation. It is highly customizable, scalable and embeddable on existing infrastructure. Like Prometheus, it is compatible with Grafana to do the data visualization function.

Some of the OpenNMS Horizon features include are [8]:

- Network Monitoring:
  - Fault and performance management
  - Network and route discovery
  - Distributed monitoring
  - Compatible with asynchronous events
  - Supports standard network protocols (SNMP, ICMP, etc.)
  - Flow analysis
  - Thresholding
  - Service hierarchy
- Platform Solution:
  - Highly extensible
  - Configuration through web User Interface (UI) and REpresentational State Transfer (REST) API
- Highly Scalable, Event-Driven Architecture:
  - Postgre Structured Query Language (PostgreSQL) and Rocket-Fast System for Log Processing (RRDtool) persistence

## 2.6. Cacti

Cacti [9] is an open-source performance and fault management and a frontend to RRDTool created by Ian Berry. It is natively based on “Linux, Apache, My Structured Query Language (MySQL) and Hypertext Preprocessor (PHP)” (LAMP) model, but now it has evolved. Cacti is now also supported on Windows, it can use both Engine- ex (NGINX) on Linux and Internet Information Server (IIS) on Windows as an alternative to Apache, and also can use Maria Database (MariaDB) as an alternative to MySQL to store data and then leverages its data collectors to populate RRDTool based Time Series Database (TSDB) with that data. PHP is still used as a code language for web development.

Some of the OpenNMS Horizon features include are [10]:

- Devices:
  - Supports SNMP, ICMP, TCP and UDP checks
  - Provides automatic graph and data source creation through automatization and discovery
  - Supports plugins for adding extra functionalities

- Graphs:
  - Can show average, maximum, minimum and last values
  - Can edit both the left and right axis
  - Graph edition through either graph template or manually.
- Data Sources:
  - Defines RRDtools Round Robin Database File structure in Cacti
  - Each Cacti system can contain multiple Data Source Profiles and use them concurrently depending on the Cacti administrator
- Data Collection:
  - Data can be collected via SNMP OIDs or scripts.
  - Scripts can pull data from databases or run arbitrary code.
  - Can also use other protocols through Plugins.
- Discovery and Automation:
  - Provides automation for network discovery and devices addition
  - Provides the ability to add graphs to devices with predefined creation rules
  - Can automatically create alerts based on rules with some additional plugins
- Users and User Groups:
  - Administrator can create users and assign them different levels of permissions
  - Permissions can be specified for a specific graph and user

## 2.7. QRadar Community Edition SIEM

This tool will be used at the request of the System Administrator since this tool offers advanced parsing features that will be very useful for our project and for future integrations.

System Administrator have chosen QRadar Community Edition because it is the only freeware tool with no time limitation and it also has one of the best positions among SIEM options [11]:

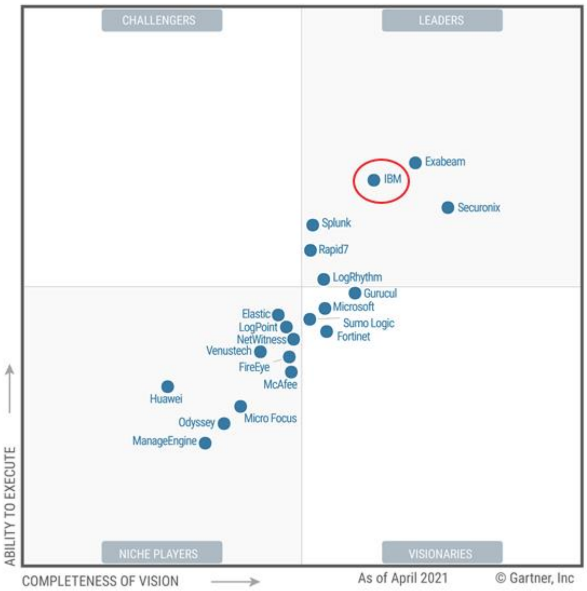


*Figure 1. Gartner Magic Quadrant for SIEM*

It is also the leading solution in the main Security Operations Centers (SoC) in the world [12].

This SIEM is a freeware tool with limited resources, but it will help us to carry out tests with a product similar to the one in the client environment. This version is limited to 50 events per second and 5,000 network flows a minute [13].

A SIEM solution normally offers the same core set of functionality [14]:

- Log Management: SIEM captures event data from different sources across the company network. This data is collected, stored and analysed in real-time, giving IT and security teams the ability to automatically manage their network event log and flow data in one centralised location.
- Event Correlation and Analytics: SIEM uses advanced analytics to identify intricate data patterns and correlation provides insights to quickly locate and mitigate potential threats. It helps to reduce response time to a threat.
- Incident Monitoring and Security Alerts: Centralized model allows SIEM quickly identify abnormal behaviour so network administrators can be alerted immediately and mitigate these security issues.
- Compliance Management and Reporting: SIEM solutions are a frequent choice for organizations subject to regulatory compliance. SIEM solutions can generate real-time compliance reports for most of the compliance standards, reducing the burden of security management and detecting potential violations early so they can be addressed.

QRadar Community Edition GUI has different tabs [15]:

- Dashboard: The Dashboard tab is a workspace environment that provides a summary and detailed information on events occurring in your network. The available dashboards are:
    - Application Overview
    - Compliance Overview
    - Network Overview
    - Risk Monitoring
    - Threat Monitoring
    - System Monitoring
- Offenses: View offenses that occur on your network, which you can locate by using various navigation options or through powerful searches.
- Log Activity: Investigate event logs that are sent to QRadar in real-time, perform powerful searches, and view log activity by using configurable time-series charts.
- Network activity tab: Used to investigate flows that are sent in real-time, perform powerful searches, and view network activity by using configurable time-series charts.
- Assets tab: Discover assets, servers, and hosts that are operating on your network.
- Reports tab: Used to create, distribute, and manage reports for any data.
- Admin: Contains the SIEM management tools. It allows, among other features, user editing, adding event sources and modifying the system configuration.

# 3. Project Development

## 3.1. Introduction

The project development is based on different internal tasks. After studying the *State of the art*, we have seen several tools that could be used in this project and hence, we need to set up a Selection Criteria to choose the best option to develop the project. This selection criterion must account for the functional and non-functional requirements previously raised.

In order to build this criteria, the first step is to search for documentation on the specifications of the infrastructure and the devices that make it up. Then, I will define a selection criteria that awards the fulfilment and penalizes non-fulfilment of the requirements and rank all the available tools so that the best one can be identified and further used in this project.

Once the choice of the tool has been made, we will propose the design of the test environment and review it with the System Administrator. When the design has been validated, we will install the different operating systems on VMs and install the tools on those operating systems.

After having completed the installation and initial configuration of the different devices and tools, I will develop different tests on the chosen monitoring tool to check its validity to automate the proposed functionalities. These tests will be part of the implementation, validation and test phase that will be developed in parallel.

Finally, if we have time and the client gives us their approval, we will execute a proof of concept (PoC) in one of the infrastructure areas.

## 3.2. Definition of Selection Criterion

I have made a selection criteria based on the requirements established for this project. Each requirement has been evaluated in each of the tools.

To do this, a scoring system has been established based on the level of compliance with each requirement. For example, if the tool completely meets one of the evaluated requirements, it will be evaluated as "Yes" and 3 points will be awarded. If it does not meet one of the conditions or depends on additional settings or plugins it will be evaluated as "Partially" and 1 point will be awarded. If you do not meet any of the conditions of the requirement, it will be evaluated as "No" and -3 points will be awarded.

This score is weighted according to the level of importance of the requirement under study. This weights are assigned as Low (x1), Medium (x2) and High (x3).

After the calculation, I will choose the tool with the best scoring results. If any tool has a score very close to that of the best, a superficial study of it is contemplated to determine its validity as an alternative.

## 3.3. Application of Criterion

First, we are going to investigate the level of compliance with each of the requirements for the different tools under study. To do this, we will collect data from the different tools in their documentation and in their specific forums and repositories to find alternatives if they do not offer the required functionalities directly. This study is shown below:

Project Functional Requirements

- F1. Provide dashboards [16-19]: Most of the solutions require additional software or plugin installation to use graph functionality. Nagios Core needs an additional plugin named PNP4Nagios to show graphs. Prometheus and OpenNMS need Grafana to show graphs. Zabbix and Cacti offer by default graph functionality. Most of the solutions offer advanced graphs and single-item value widgets, but some of them require additional software or plugins.

- F2. Provide device discovery and integration tool [20-29]: Allow SNMP interface addition with a discovery rule and basic information like community password. This functionality depends on SNMP protocol and Discovery function. Because of that, Prometheus requires additional configuration to use SNMP and Nagios Core needs a plugin to use Discovery functionality.

- F3. Show critical measurements [20-24]: If I want to monitor CPU and Memory, I can use SNMP Polls or SSH commands. Then, most of the studied options give this functionality by default, but Prometheus needs additional configuration to use both protocols.

- F4. Set up threshold-triggered alarms based on hardware capabilities [30-34]: Most of them offer this functionality by default, but Nagios Core maybe have some problems combining multiple obtained metrics and Cacti needs an additional plugin to use thresholds.

- F5. Send email alerts [30, 34-36]: All of them offer this functionality by default or with little configuration.

- F6. Add parameters and graphs by using available OIDs [20-24]: I can make an SNMP Walk request to display the list of OIDs, then filter by name based on the metrics we want and add them for monitoring. Depending on the operating system we want to monitor, it has a base OID that can make the process easier. Therefore, most of the tools studied have this functionality except Prometheus, which requires additional configurations.

- F7. Have a list of users with different roles [37-40]: All of them offer this functionality by default.

- F8. Track configuration changes [37-40]: Grafana and Zabbix offer this functionality by default. Other tools offer to track changes but without user identification.

Project Non-Functional Requirements

- NF1. Work with agentless monitoring [20-24]: Most of the studied tools offer by default the integration of new systems and the addition of the basic functionality of monitoring metrics, such as monitoring through the use of SNMP, ICMP and SSH protocols. Prometheus needs an additional configuration to use SNMP and SSH. Cacti needs an additional extension to use SSH. Additional monitoring metrics will be studied in later requirements.

- NF2. Programmable based on scripts or small pieces of software:

    - Frontend Locations [41-44]: Nagios Core and Zabbix allow graph manipulation and creation through scripts. Cacti and Grafana allow some script adaptation, but they depend on GUI manipulation.

  - Events [24, 43, 45-47]: Nagios Core, Zabbix, OpenNMS and Cacti offer event manipulation through scripts. Prometheus allows only alerting rules through scripts.
  - Specific actions [24, 43, 48]: Nagios Core, Zabbix and Cacti admits host addition through scripts. I have not found anything about this tool for Prometheus and OpenNMS.

- NF3. Support web interface [49-52]: Zabbix, Grafana and Cacti offer an easy-to-use and aesthetic web interface. Nagios core offers an aesthetic web interface, but it has to be used through CLI.
- NF4. Allows basic security methods [53-56]: Most of them offer SSL certificates for servers, but I have not found anything about SSL certificates for Cacti.
- NF5. Collect and filter log file entries: I will use QRadar Community Edition to do this functionality, so this requirement is not used in the comparison between tools. For this reason, the requirement appears marked in blue in the results table (Table 2).
- NF6. Deployment of the system in a virtual environment based on an open-source operating system [57-61]: All of them can be installed in an open-source Operating System.
- NF7. Development of the system using only freeware software: All of them are freeware software.
- NF8. Compatible with existing OS [62-65]: Nagios Core and Zabbix are compatible with Windows, CentOS and Debian monitoring, but Prometheus needs additional plugins to monitor Linux and Windows. I have not found anything about the compatibility of OpenNMS and Cacti with operating systems.

| Type of Requirement | ID | Important | Nagios Core | Prometheus + Grafana | Zabbix | Open NMS + Grafana | Cacti |
|---|---|---|---|---|---|---|---|
| **Project Functional Requirements** | F1 | High | Partially | Partially | Yes | Partially | Yes |
| | F2 | High | Partially | Partially | Yes | Yes | Yes |
| | F3 | Medium | Yes | Partially | Yes | Yes | Yes |
| | F4 | Low | Partially | Yes | Yes | Yes | Partially |
| | F5 | Low | Yes | Yes | Yes | Yes | Yes |
| | F6 | High | Yes | Partially | Yes | Yes | Yes |
| | F7 | Medium | Yes | Yes | Yes | Yes | Yes |
| | F8 | Medium | Partially | Yes | Yes | Yes | Partially |
| **Project Non-Functional Requirements** | NF1 | High | Yes | Partially | Yes | Yes | Partially |
| | NF2 | High | Yes | Partially | Yes | Partially | Partially |
| | NF3 | Medium | Partially | Yes | Yes | Yes | Yes |
| | NF4 | High | Yes | Yes | Yes | Yes | No |
| | NF5 | High | - | - | - | - | - |
| | NF6 | High | Yes | Yes | Yes | Yes | Yes |
| | NF7 | High | Yes | Yes | Yes | Yes | Yes |
| | NF8 | High | Yes | Partially | Yes | No | No |
| **Total** | | | **89** | **71** | **111** | **81** | **57** |

*Table 2. Tools Basic Check According to Requirements*

Once the comparison and calculations have been made, we see that Zabbix is the best option for the development of the project, closely followed by Nagios Core. This latter is a very well-known product, which can be easily seen in most medium and large network deployments when remote management is required. Therefore, in addition to Zabbix, I carried out a basic installation with Nagios Core to provide a simpler test to assess if the Nagios solution could eventually become a feasible solution, as good as Zabbix. The implementation with Nagios Core is thought of as a way to provide a simple but experimental comparison with Zabbix. Furthermore, we will be able to use Nagios Core as one more device within the testing environment to carry out monitoring tests.

## 3.4. Documentation about Infrastructure Specifications

### 3.4.1. Infrastructure Components

The infrastructure has the following main components:

- SIEM Cluster: It is SIEM brain. They receive all events from the RSYSLOG Cluster and from some specific devices. These events are stored for years to be able to analyse data later if necessary. Real-time rules are also applied to them, providing information to detect problems instantly. If there is a problem, the SIEM will send SNMP Traps to the monitoring tool reporting the problem so that system administrators can act immediately.
- RSYSLOG Clusters: They are used to avoid problems between some event senders and SIEM. They are located between the general event sources and the SIEM and are used to discard unwanted events, analyse some fields and drop events to lighten the SIEM processing load.
- Monitoring Server: Server for environment monitoring. It will use SNMP, SSH and ICMP protocols for monitoring tasks. It will be in charge of monitoring the status of the different devices and the resources and services they use.
- Event Sources: These are the components that are used on a daily basis by the customer and its partners. They are the sources of the various events and communicate directly with the RSYSLOG Clusters. In some special cases, some of the sources can directly send the events to the SIEM if it is considered necessary.

### 3.4.2. Connections Scheme

After introducing the different elements of the infrastructure, I want to show the traffic flow diagram and connections scheme between them.

Normally, event sources generate events and send them to RSYSLOG Cluster assigned to its zone. Then, RSYSLOG Clusters are in charge of receiving, processing according to the arrival port and redirecting the events to the SIEM cluster, where they are filtered and collected. We can also see in blue lines the monitoring traffic flows (ICMP, SSH and SNMP) between the RSYSLOG Clusters and the monitoring server. Finally, we see the flow of SNMP Traps that the SIEM Cluster sends to the monitoring server. This diagram is shown in Figure 2.

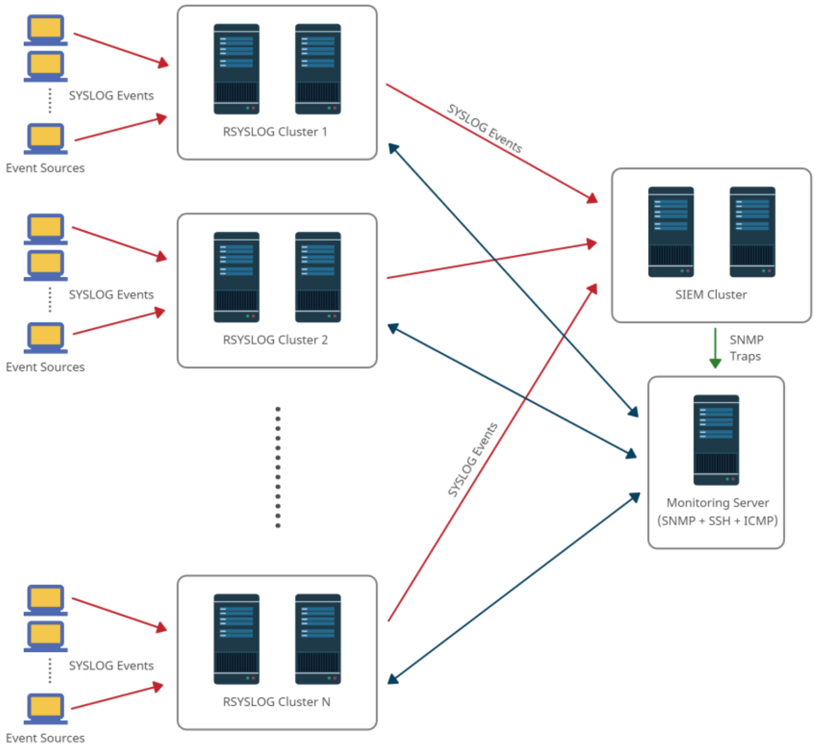


*Figure 2. General Flow Diagram*

For some special sources such as Windows devices, the events flow diagram is simplified, because SYSLOG Events are directly sent to SIEM Cluster. This diagram is shown in Figure 3.

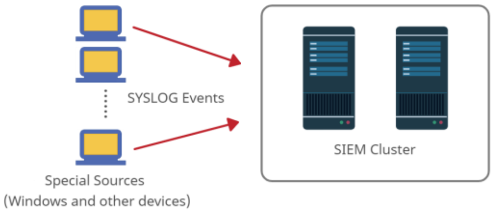


*Figure 3. Specific Event Flow Diagram*

### 3.5. Laboratory Environment Creation

The lab environment is a scaled-down replica of one of the areas that make up the client infrastructure. The lab environment I created has the following devices:

- QRadar Community Edition SIEM: This element represents the figure of the SIEM Cluster of the client infrastructure. It receives, filters and collects events from the other devices in the lab environment.
- RSYSLOG Cluster: This cluster is configured the same as those of the client network. It receives and forwards events from event sources.
- Zabbix Server: This device represents the client network monitoring server. It is used for monitoring the environment and also as an event source.

- Nagios Core Server: This device represents a second monitoring server to make a comparison with Zabbix. It is also used for monitoring the environment and as an event source.
- Windows 10 Host: This host represents a Windows device such as one that could be used as an event source on the customer network.

These virtual devices are allocated on two different servers:

- VMware vSphere [66]: Server virtualization software used to allocate Virtual Machines (VM) used on Nagios Core, Zabbix and RSYSLOG Cluster devices. VMs Specifications are explained on *APPENDIX A*.
- Proxmox Virtual Environment [67]: Open-source platform for virtualization to allocate QRadar Community Edition SIEM because of compatibility problems with vSphere Secure Hash Algorithm (SHA) SHA-256 checksum version.

As the proposed monitoring tools are freeware and are installed on VMs with Debian or CentOS as OS, the NF6 and NF7 requirements are validated.

Finally, we will use a physical device located in the company lab with a working version of Windows 10 to send SYSLOG events using WinCollect.

### 3.5.1. QRadar Community Edition SIEM

QRadar Community Edition has been installed on Proxmox server because it has some compatibility issues with vSphere tool. Installation and configuration processes are shown on *APPENDIX E*, section "*Installation Process*".

Nagios Core, Zabbix and RSYSLOG Cluster nodes VMs configuration to send SYSLOG events is explained on *APPENDIX F*, section "*Debian and CentOS Configuration*".

Windows 10 Host configuration to send WinCollect events is explained on *APPENDIX F*, section "*Windows Configuration*".

SNMP Traps configuration is explained on *APPENDIX E*, section "*SNMP Traps Configuration*".

### 3.5.2. RYSLOG Cluster

The cluster is made up of two or more devices called nodes that work together to perform a specific task. In this case, I have used a high-availability cluster the same as the one used in the client infrastructure. This type of cluster provides high availability services that guarantee that the services will always be available, since if one node fails there will be another one that will take over the task and the services will continue to work [68].

I am going to use 2 devices called rsyslog1 and rsyslog2 and their Virtual Internet Protocol Address (VIPA) to centralize the reception of event logs and send them to the SIEM. Its installation is explained on *APPENDIX D*.

### 3.5.3. Zabbix Server

Zabbix VMs installation is explained on *APPENDIX C*, section "*Installation Process*".

### 3.5.4. Nagios Core Server

Nagios Core VMs installation is explained on *APPENDIX B*, section "*Installation Process*".

### 3.5.5. Windows 10 Host

Windows Server 2016 Standard is allocated in a company Data Center from which you have access to the rest of the devices directly or by creating an SSH tunnel to the Zabbix server. I can access to it by using Remote Desktop (IP: 10.0.3.199).

### 3.5.6. Connections Scheme

After introducing the different elements, I want to show the traffic flow diagram and the connections scheme between them.

Debian VMs (Nagios Core and Zabbix servers) are used as event sources and they generate events and send them to RSYSLOG Cluster. Then, RSYSLOG Cluster is in charge of receiving, processing according to the arrival port and redirecting the events to QRadar Community Edition SIEM, where they are filtered and collected. Windows 10 Host sends events directly to QRadar Community Edition SIEM. These event traffic flows are represented with red lines.

We can also see in blue lines the monitoring traffic flows (ICMP, SSH and SNMP) between the RSYSLOG Cluster and the monitoring servers and also between QRadar Community Edition and Nagios Core server.

Finally, Zabbix server monitoring flow (ICMP, SSH and SNMP Polls and Traps) with QRadar Community is represented with a green line and ICMP Windows host monitoring is represented with a purple line.

This diagram is shown in Figure 4.

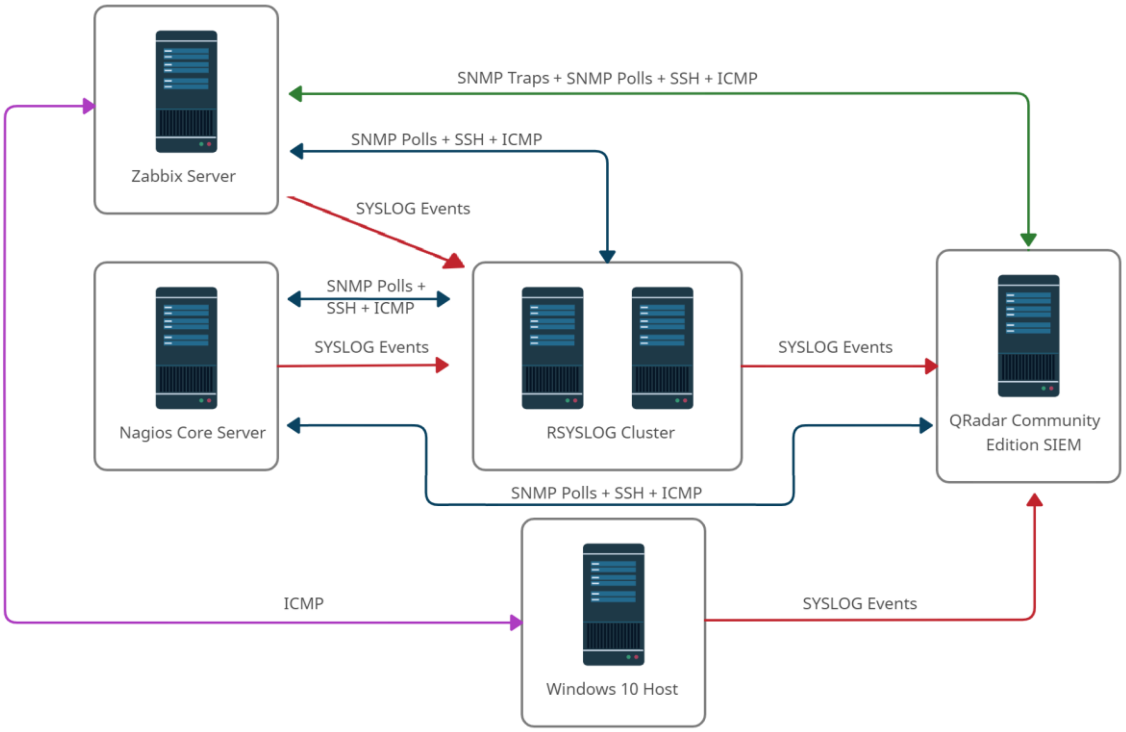


*Figure 4. Laboratory Environment Connections Scheme*

### 3.6. Implementation, Validation and Test

Once the tools and environment are introduced, I have planned a test phase to check requirements one by one and some of them probably can be tested together. This test phase is applied on Zabbix and Nagios Core and is defined in the following table:

| |
|---|
| **Name:** Zabbix and Nagios Core Automation Test |
| **Description:** This test intends to validate Zabbix and Nagios Core automation and integration capabilities in the Laboratory Environment introduced in last section. |
| **Input and Preconditions:**<br>• Zabbix and Nagios Core VMs with ICMP and SSH enabled and configured and with SNMP Load, CPU, Memory and Disk monitoring stats available.<br>• QRadar Community Edition with SNMP Traps enabled and configured to send to Zabbix and has the Wincollect plugin installed to monitor the Windows device. SSH enabled and SNMP Polls available to be monitored by Zabbix and Nagios Core.<br>• RSYSLOG Cluster created and configured and VMs (rsyslog1 and rsyslog2) configured as forwarding devices for RSYSLOG events. ICMP and SSH enabled and configured and with SNMP Load, CPU, Memory, Disk and RSYSLOG Resources and Services monitoring stats available.<br>• Windows 10 VM configured as event source by using Wincollect. |
| **Expected Output:**<br>• Zabbix monitoring tool with added SNMP, SSH and ICMP items and graphs from RSYSLOG Cluster devices, Nagios Core and QRadar Community Edition and ICMP monitoring from Windows.<br>• Nagios Core monitoring tool with added SNMP, SSH and ICMP items and graphs from RSYSLOG Cluster devices, Zabbix and QRadar Community Edition and ICMP monitoring from Windows.<br>• QRadar Community Edition with collected and filtered logs and with SNMP Traps configurated on Zabbix. |
| **Applicable Requirements (pass/fail criterion):**<br>NF1, NF2, NF3, NF4, NF5, NF8, F1, F2, F3, F4, F5, F6, F7 and F8. |
| **Followed Steps and Requirements Tested:**<br>1. Configure HTTPS use on web interfaces<br>2. Create and configure new user and track its changes<br>3. Create and configure device discovery and integration<br>4. Add items, graphs and triggers by using available OIDs<br>5. Configure email alerts and check they work<br>6. Show events on SIEM |

*Table 3. Test Definition Table*

In the following subsections you can find some information about the followed steps:

### 3.6.1. Configure HTTPS use on web interfaces

There was the option to use both MariaDB and PostGRESQL databases on monitoring servers, but I have decided to use MariaDB because is lighter and has less memory impact on the devices. Both tools can use SSL certificates and their creation and MariaDB Configuration are explained on *APPENDIX G*.

I have installed both tools based on Apache2 servers, although there was also the NGINX alternative, but I preferred to use Apache2 because I had prior knowledge of its operation. Then, I need to configure both devices to use SSL Certificates on their web interfaces. Steps for configuring both devices are described below [69-70]:

1. Edit this lines on Default SSL file:

   ```
   localhost: nano /etc/apache2/sites-available/default-ssl.conf
   ```

   ```
   ServerAdmin hector.arroyo@estudiantat.upc.edu
   SSLCertificateFile      /etc/mysql/certs/server-cert.pem
   SSLCertificateKeyFile /etc/mysql/certs/server-key.pem
   ```

2. Enable some modules and default-ssl site:

   ```
   localhost: sudo a2enmod rewrite
   localhost: sudo a2enmod ssl
   localhost: sudo a2ensite default-ssl
   ```

3. Modify default server file to redirect HTTP connection to HTTPS:

   ```
   localhost: nano /etc/apache2/sites-available/000-default.conf
   ```

   ```
   RewriteEngine On
   RewriteCond %{HTTPS} off
   RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
   ```

   Also add this lines to Zabbix Configuration file on Apache2 for Zabbix Server:

   ```
   localhost: nano /etc/apache2/conf-enabled/zabbix.conf
   ```

4. Finally, restart Apache2:

   ```
   localhost: systemctl restart apache2
   ```

Finally, I have tried to access to both websites:

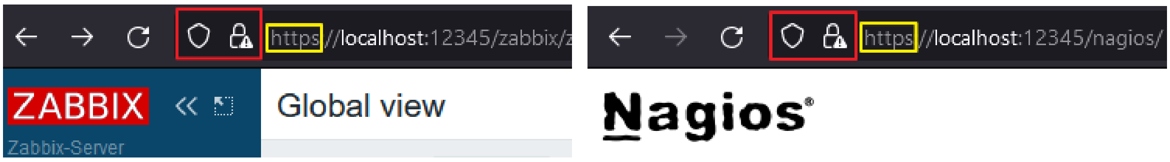

https://localhost:12345/zabbix/z
ZABBIX
Zabbix-Server
Global view
https://localhost:12345/nagios/
Nagios

*Figure 5. Zabbix and Nagios Core HTTPS Validation*

You can see a warning in the red box because self-signed certificates have been used.

This checks that HTTPS is enabled and must be used on Zabbix and Nagios Core, so the NF4 requirement is validated.

### 3.6.2. Create and configure new user and track its changes

**Zabbix**

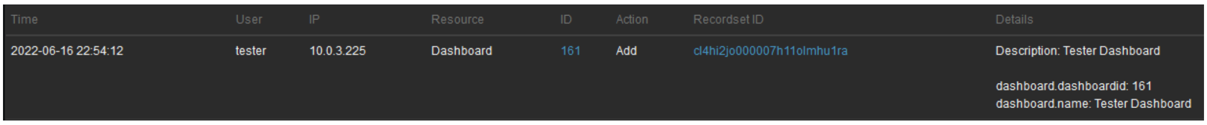
Zabbix allows users with different roles assigned to different projects and also track configuration changes. User roles can be configured in “Administration > User roles” tab.

To make it easy, I will use the role called “User role” that is created by default and give permissions similar to and “Editor” role. Then, I need to create a new user called “tester”. To make this, go to “Administration > Users” tab and click on “Create user”:

- Username: tester
- Groups: Guests
- Password: xxxxxxxx
- Permissions>Role: User role

Then, I log out from the “Admin” user and log in with the “tester” user. With that users, we have no visibility to the “Administrator” tab, but with the “Admin” user we can see it because it has a “super administrator” role.

Then, I go to “Monitoring>Dashboard” tab and create a new dashboard named “Tester Dashboard” with “tester” user. Finally, go to “Reports>Audit” with “Admin” user to see change tracking:



| Time | User | IP | Resource | ID | Action | Recordset ID | Details |
|---|---|---|---|---|---|---|---|
| 2022-06-16 22:54:12 | tester | 10.0.3.225 | Dashboard | 161 | Add | cl4hi2jo000007h11olmhu1ra | Description: Tester Dashboard<br>dashboard.dashboardid: 161<br>dashboard.name: Tester Dashboard |

*Figure 6. Change Tracking on Zabbix*

With Zabbix, I can create a new user and give them a “viewer”, “editor” or “admin” role. It also has track changes functionality and shows which user has made the change. It also gives a friendly and easy-to-use web interface. So, fulfils F7, F8 and NF3 requirements.

**Nagios Core**

Nagios Core supports different permissions assigned to different users (explained in *APPENDIX B*, section “*User Configuration Options*”).

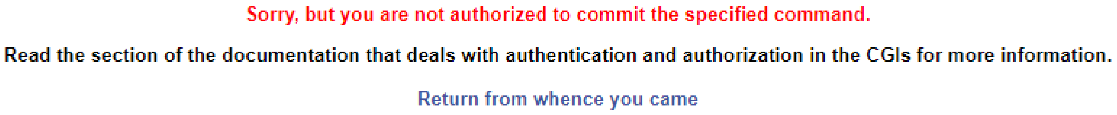
First, we need to create a new user called “tester”:

localhost: htpasswd /usr/local/nagios/etc/htpasswd.users tester → Ask for password.

Then, I have to modify the Computer Generated Imagery (CGI) configuration file. To configure “tester” user, I have to modify the following variables using this format:

*authorized_for_XXXXX=nagiosadmin,**tester*** → Add tester to lists to give the authorization.

I have given “tester” user authorization for “Read-only”. Then, I have to restart Nagios’ service. Next, I go to the web interface, login with tester credentials, go to the “Services” tab and select one service. When I try to use service commands, the web interface shows the message from Figure 7.



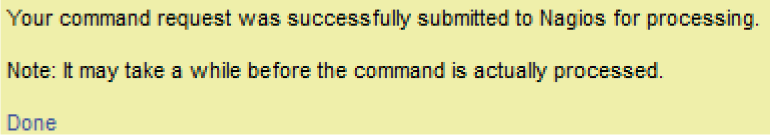
Sorry, but you are not authorized to commit the specified command.

Read the section of the documentation that deals with authentication and authorization in the CGIs for more information.

Return from whence you came

*Figure 7. No Authorized Message for Nagios Core*

But when I try to make the same with “nagiosadmin” user, it works and shows the message from Figure 8.



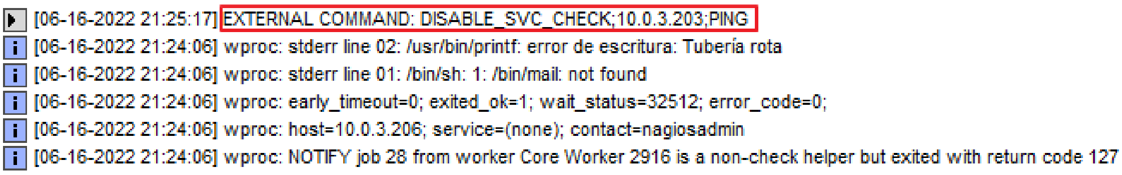
Your command request was successfully submitted to Nagios for processing.

Note: It may take a while before the command is actually processed.

Done

*Figure 8. Authorized Message for Nagios Core*

Finally, go to “Event log” tab to see the change tracking shown in Figure 9.



```
[06-16-2022 21:25:17] EXTERNAL COMMAND: DISABLE_SVC_CHECK;10.0.3.203;PING
[06-16-2022 21:24:06] wproc: stderr line 02: /usr/bin/printf: error de escritura: Tubería rota
[06-16-2022 21:24:06] wproc: stderr line 01: /bin/sh: 1: /bin/mail: not found
[06-16-2022 21:24:06] wproc: early_timeout=0; exited_ok=1; wait_status=32512; error_code=0;
[06-16-2022 21:24:06] wproc: host=10.0.3.206; service=(none); contact=nagiosadmin
[06-16-2022 21:24:06] wproc: NOTIFY job 28 from worker Core Worker 2916 is a non-check helper but exited with return code 127
```

*Figure 9. Change Tracking on Nagios Core*

With Nagios Core, I can create a new user and give them a “viewer” role, but I can also give more attributes to make them an “editor” or “admin” role. It has track changes functionality, but it does not show which user has made the change. It also gives a friendly web interface, but it is not easy to use because it is configured on the terminal in Nagios Core VM. So, F7 is achieved, but F8 and NF3 are not.

### 3.6.3. Create and configure device discovery and integration

**Zabbix:**

I need to create a Discovery Rule for ICMP, SSH and SNMP protocols scanning and then create a Discovery Action to incorporate them as devices. I have done this with these steps:

1. First, go to “Configuration > Discovery” tab and click on “Create discovery rule”:

   This discovery rule has the following configuration:

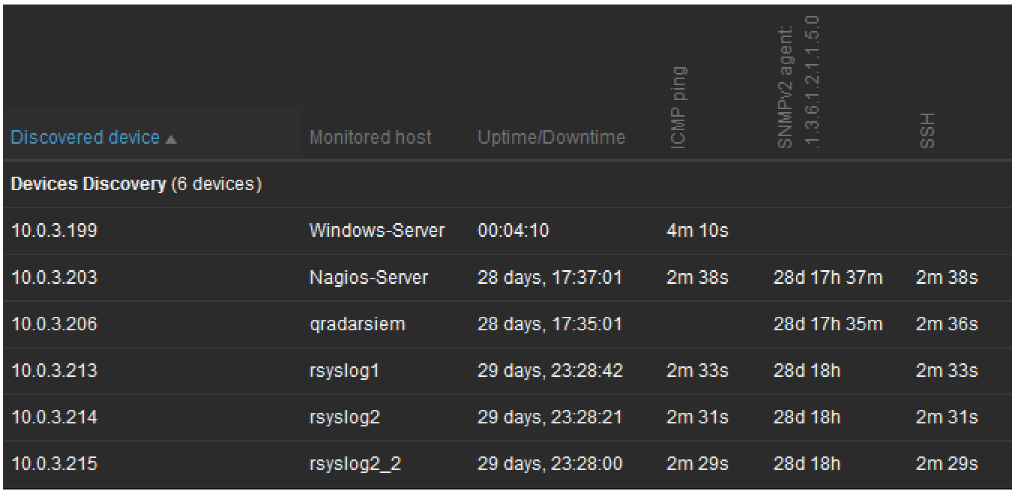
   - Name: “Devices Discovery”
   - Update interval: 1h (to not overuse “Discovery” resources)
   - Internet Protocol (IP) range: 10.0.3.199-215 (where relevant devices are allocated).
   - Checks: SSH, ICMP and SNMP.

     To check SNMP an OID has to be specified, so I will use this OID to get the hostname from a device:

     Name: sysName OID: .1.3.6.1.2.1.1.5.0 Type: STRING

   - Hostname/Visible name: Use “sysName” OID value.

     I have been waiting some minutes, and I can see all devices and services discovered, including Windows 10 Host ICMP service, in Figure 10.



| Discovered device ▲ | Monitored host | Uptime/Downtime | ICMP ping | SNMPv2 agent: .1.3.6.1.2.1.1.5.0 | SSH |
|---|---|---|---|---|---|
| **Devices Discovery** (6 devices) | | | | | |
| 10.0.3.199 | Windows-Server | 00:04:10 | 4m 10s | | |
| 10.0.3.203 | Nagios-Server | 28 days, 17:37:01 | 2m 38s | 28d 17h 37m | 2m 38s |
| 10.0.3.206 | qradarsiem | 28 days, 17:35:01 | | 28d 17h 35m | 2m 36s |
| 10.0.3.213 | rsyslog1 | 29 days, 23:28:42 | 2m 33s | 28d 18h | 2m 33s |
| 10.0.3.214 | rsyslog2 | 29 days, 23:28:21 | 2m 31s | 28d 18h | 2m 31s |
| 10.0.3.215 | rsyslog2_2 | 29 days, 23:28:00 | 2m 29s | 28d 18h | 2m 29s |

*Figure 10. Devices Discovery Results*

2. <u>Now, go to “Configuration>Actions>Discovery actions” tab and click on “Create action”:</u>

I need to create two discovery actions, one for RSYSLOG Devices and a second one more generic for Linux Devices:

- <u>RSYSLOG Devices Integration</u>

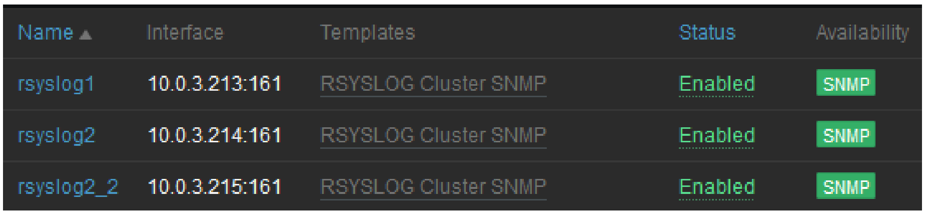
<u>Actions:</u> (Accomplishment of A and B are required)

A. Discovery check equals Devices Name: SNMPv2 agent ".1.3.6.1.2.1.1.5.0".
B. Received value contains *rsyslog*

<u>Operations:</u>

A. Add host
B. Add to host groups: RSYSLOG Devices
C. Link to templates: RSYSLOG Cluster SNMP

<u>Results (on “Configuration>Hosts” tab):</u>



| Name ▲ | Interface | Templates | Status | Availability |
|---|---|---|---|---|
| rsyslog1 | 10.0.3.213:161 | RSYSLOG Cluster SNMP | Enabled | SNMP |
| rsyslog2 | 10.0.3.214:161 | RSYSLOG Cluster SNMP | Enabled | SNMP |
| rsyslog2_2 | 10.0.3.215:161 | RSYSLOG Cluster SNMP | Enabled | SNMP |

*Figure 11. RSYSLOG Devices Integration on Zabbix*

- <u>Linux Devices Integration</u>

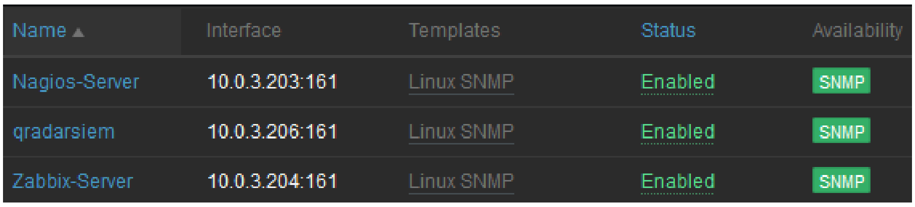
<u>Actions:</u> Accomplishment of (A and B) and (C and D) are required.

A. Discovery check equals Devices Name: SNMPv2 agent ".1.3.6.1.2.1.1.5.0".
B. Received value does not contain *rsyslog*

<u>Operations:</u>

A. Add host
B. Add to host groups: Linux Devices
C. Link to templates: Linux SNMP

Results (on “Configuration>Hosts” tab):



| Name ▲ | Interface | Templates | Status | Availability |
|---|---|---|---|---|
| Nagios-Server | 10.0.3.203:161 | Linux SNMP | Enabled | SNMP |
| qradarsiem | 10.0.3.206:161 | Linux SNMP | Enabled | SNMP |
| Zabbix-Server | 10.0.3.204:161 | Linux SNMP | Enabled | SNMP |

*Figure 12. Linux Devices Integration on Zabbix*

Zabbix provides a Device discovery tool with ICMP, SSH and SNMP scan options. It also has compatibility with the existing OS and works with agentless monitoring. So, it fulfils F2, NF1 and NF8 requirements.

**Nagios Core:**

In Nagios Core, I need to use the Nagios Bulk Import (NBI) script to use the discovery and integration tool on Nagios Core. I have followed the steps below:

1) Get nbi.pl script from Nagios Core add-ons repository and edit it for what I want:

```
cd /usr/local/nagios/etc/hosts
```

```
wget
https://exchange.nagios.org/components/com_mtree/attachment.php?link_id=3158&cf_id=24
```

2) Generate a file with Network Map(NMAP) output:

nmap -sS -O -oG discovery.txt 10.0.3.203-215 → Gets all hosts (except Windows).

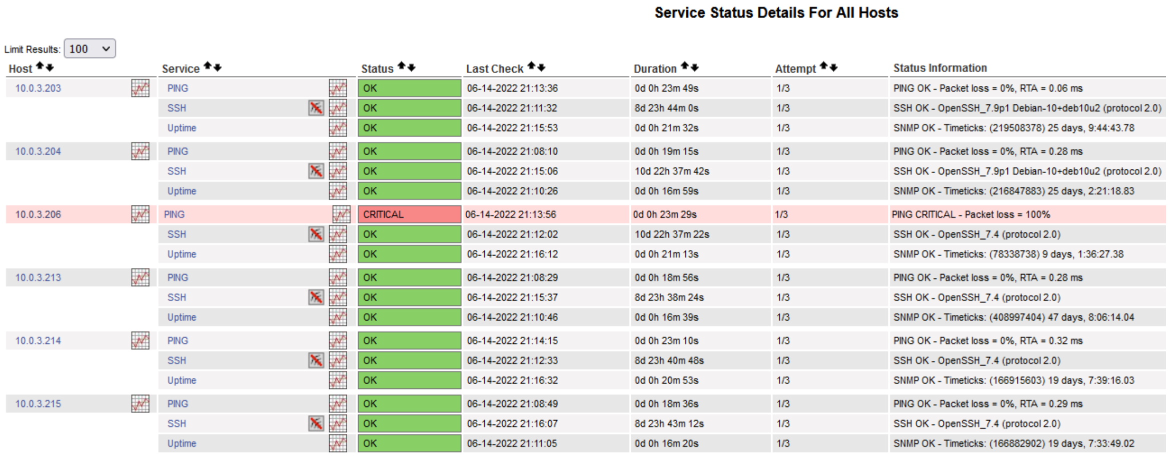
3) Edit nbi.pl file to adapt it to our needs (ICMP and SSH checks and SNMP check by using “Uptime OID”):

nano nbi.pl → Shown in *APPENDIX I*, “*nbi.pl*” section.

4) Give permissions, execute script and restart Nagios service:

```
chmod +rwx nbi.pl
./nbi.pl discovery.txt
systemctl restart Nagios
```

5) <u>Finally, check the results on the web interface on “Current Status>Services” tab:</u>



Service Status Details For All Hosts

Limit Results: 100

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|---|---|---|---|---|---|---|
| 10.0.3.203 | PING | OK | 06-14-2022 21:13:36 | 0d 0h 23m 49s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.06 ms |
| | SSH | OK | 06-14-2022 21:11:32 | 8d 23h 44m 0s | 1/3 | SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:15:53 | 0d 0h 21m 32s | 1/3 | SNMP OK - Timeticks: (219508378) 25 days, 9:44:43.78 |
| 10.0.3.204 | PING | OK | 06-14-2022 21:08:10 | 0d 0h 19m 15s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.28 ms |
| | SSH | OK | 06-14-2022 21:15:06 | 10d 22h 37m 42s | 1/3 | SSH OK - OpenSSH_7.9p1 Debian-10+deb10u2 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:10:26 | 0d 0h 16m 59s | 1/3 | SNMP OK - Timeticks: (216847883) 25 days, 2:21:18.83 |
| 10.0.3.206 | PING | CRITICAL | 06-14-2022 21:13:56 | 0d 0h 23m 29s | 1/3 | PING CRITICAL - Packet loss = 100% |
| | SSH | OK | 06-14-2022 21:12:02 | 10d 22h 37m 22s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:16:12 | 0d 0h 21m 13s | 1/3 | SNMP OK - Timeticks: (78338738) 9 days, 1:36:27.38 |
| 10.0.3.213 | PING | OK | 06-14-2022 21:08:29 | 0d 0h 18m 56s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.28 ms |
| | SSH | OK | 06-14-2022 21:15:37 | 8d 23h 38m 24s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:10:46 | 0d 0h 16m 39s | 1/3 | SNMP OK - Timeticks: (408997404) 47 days, 8:06:14.04 |
| 10.0.3.214 | PING | OK | 06-14-2022 21:14:15 | 0d 0h 23m 10s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.32 ms |
| | SSH | OK | 06-14-2022 21:12:33 | 8d 23h 40m 48s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:16:32 | 0d 0h 20m 53s | 1/3 | SNMP OK - Timeticks: (166915603) 19 days, 7:39:16.03 |
| 10.0.3.215 | PING | OK | 06-14-2022 21:08:49 | 0d 0h 18m 36s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.29 ms |
| | SSH | OK | 06-14-2022 21:16:07 | 8d 23h 43m 12s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | Uptime | OK | 06-14-2022 21:11:05 | 0d 0h 16m 20s | 1/3 | SNMP OK - Timeticks: (166882902) 19 days, 7:33:49.02 |

*Figure 13. Device Integration on Nagios Core*

I can see that the “PING” service status in QRadar Community Edition (10.0.3.206) device is “CRITICAL” and this is because this device blocks ICMP messages to avoid a Denial of Service (DoS) attack.

Nagios Core provides a device discovery and integration script with ICMP, SSH and SNMP scan options. It also has compatibility with the existing OS and works with agentless monitoring. So, it fulfils F2, NF1, NF2 and NF8 requirements.

### 3.6.4. Add items, graphs and triggers by using available OIDs

To focus on the metrics we are interested in, I will limit the discovery tool to the following metrics:

- <u>CPU Stats Discovery:</u> CPU Load average in 1, 5 or 15 minutes and time statistics for the user, system and idle.
- <u>Memory Stats Discovery:</u> Swap size and available space. Total, Used and Free RAM.
- <u>Disk Stats Discovery:</u> Total size and space used in disk
- <u>System Uptime</u>
- <u>RSYSLOG Resources and Services:</u> Cluster name, nodes number, Corosync and Pacemaker online nodes, and resources number and running.

Discovered OIDs details are shown on *APPENDIX H*. I have created items, graphs and triggers for the metrics specified and used the *rsyslog1* device to make the test.

**<u>Zabbix:</u>**

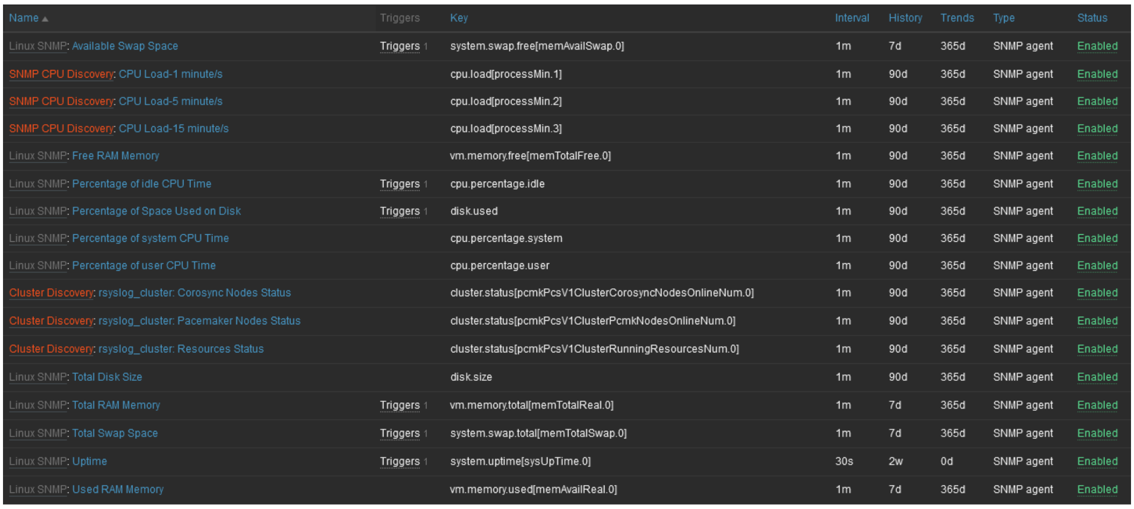
Zabbix has Discovery rules options associated with given Templates, but these options have to be configured to obtain desired results. These Discovery rules offer SNMP OIDs compatibility and I am going to configure them to discover available OIDs, create items and graphs and add triggers based on these elements' prototypes.

Zabbix also offers Templates with predefined items and all existing items are discovered when a host is associated with a template. In this case, I have created two Templates to incorporate items:

- Linux SNMP: Used in rsyslog1, rsyslog2, qradarsiem and Nagios-Server devices.
- RSYSLOG Cluster SNMP: Used in rsyslog1 and rsyslog2 devices.

Steps for this test are explained on *APPENDIX C*, section "*Items, Graphs and Triggers creation*".

Discovered Items:

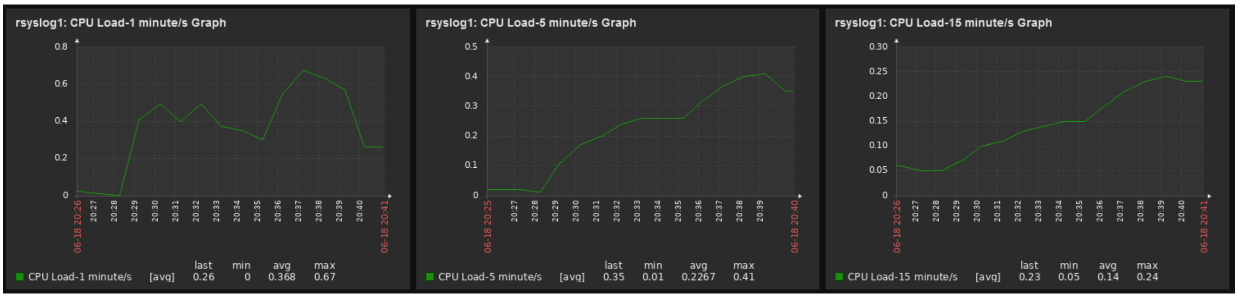


| Name ▲ | Triggers | Key | Interval | History | Trends | Type | Status |
|---|---|---|---|---|---|---|---|
| Linux SNMP: Available Swap Space | Triggers 1 | system.swap.free[memAvailSwap.0] | 1m | 7d | 365d | SNMP agent | Enabled |
| SNMP CPU Discovery: CPU Load-1 minute/s | | cpu.load[processMin.1] | 1m | 90d | 365d | SNMP agent | Enabled |
| SNMP CPU Discovery: CPU Load-5 minute/s | | cpu.load[processMin.2] | 1m | 90d | 365d | SNMP agent | Enabled |
| SNMP CPU Discovery: CPU Load-15 minute/s | | cpu.load[processMin.3] | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Free RAM Memory | | vm.memory.free[memTotalFree.0] | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Percentage of idle CPU Time | Triggers 1 | cpu.percentage.idle | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Percentage of Space Used on Disk | Triggers 1 | disk.used | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Percentage of system CPU Time | | cpu.percentage.system | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Percentage of user CPU Time | | cpu.percentage.user | 1m | 90d | 365d | SNMP agent | Enabled |
| Cluster Discovery: rsyslog_cluster: Corosync Nodes Status | | cluster.status[pcmkPcsV1ClusterCorosyncNodesOnlineNum.0] | 1m | 90d | 365d | SNMP agent | Enabled |
| Cluster Discovery: rsyslog_cluster: Pacemaker Nodes Status | | cluster.status[pcmkPcsV1ClusterPcmkNodesOnlineNum.0] | 1m | 90d | 365d | SNMP agent | Enabled |
| Cluster Discovery: rsyslog_cluster: Resources Status | | cluster.status[pcmkPcsV1ClusterRunningResourcesNum.0] | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Total Disk Size | | disk.size | 1m | 90d | 365d | SNMP agent | Enabled |
| Linux SNMP: Total RAM Memory | Triggers 1 | vm.memory.total[memTotalReal.0] | 1m | 7d | 365d | SNMP agent | Enabled |
| Linux SNMP: Total Swap Space | Triggers 1 | system.swap.total[memTotalSwap.0] | 1m | 7d | 365d | SNMP agent | Enabled |
| Linux SNMP: Uptime | Triggers 1 | system.uptime[sysUpTime.0] | 30s | 2w | 0d | SNMP agent | Enabled |
| Linux SNMP: Used RAM Memory | | vm.memory.used[memAvailReal.0] | 1m | 7d | 365d | SNMP agent | Enabled |

Figure 14. Zabbix Discovered Items

Graphs Discovered:

- Load Graphs:



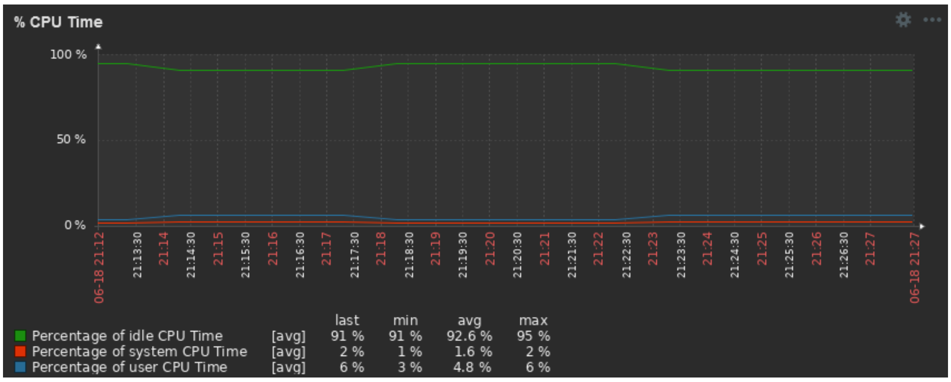
| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 CPU Load-1 minute/s | [avg] | 0.26 | 0 | 0.368 | 0.67 |

| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 CPU Load-5 minute/s | [avg] | 0.35 | 0.01 | 0.2267 | 0.41 |

| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 CPU Load-15 minute/s | [avg] | 0.23 | 0.05 | 0.14 | 0.24 |

Figure 15. CPU Load Graphs

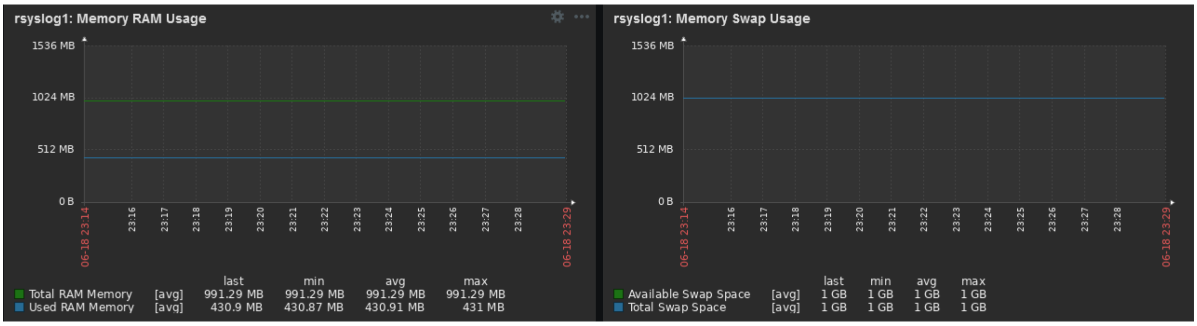
- CPU Percentage Graph:



| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Percentage of idle CPU Time | [avg] | 91 % | 91 % | 92.6 % | 95 % |
| 🟥 Percentage of system CPU Time | [avg] | 2 % | 1 % | 1.6 % | 2 % |
| 🟦 Percentage of user CPU Time | [avg] | 6 % | 3 % | 4.8 % | 6 % |

Figure 16. CPU Percentages Graph

- <u>Memory Graph:</u>

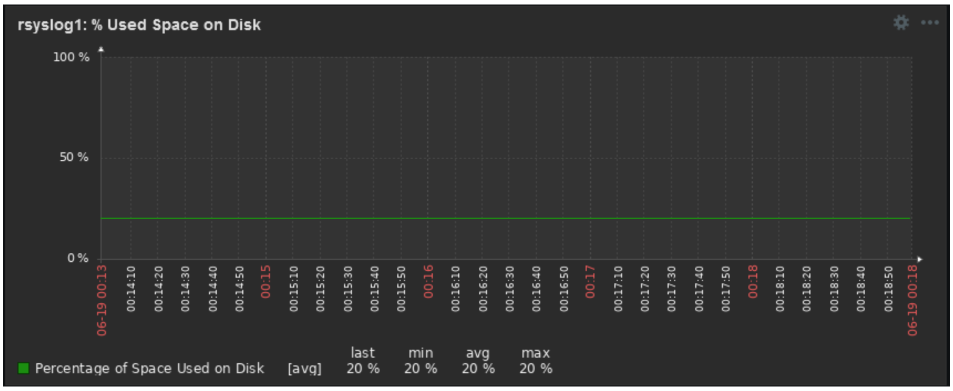


| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Total RAM Memory | [avg] | 991.29 MB | 991.29 MB | 991.29 MB | 991.29 MB |
| 🟦 Used RAM Memory | [avg] | 430.9 MB | 430.87 MB | 430.91 MB | 431 MB |

| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Available Swap Space | [avg] | 1 GB | 1 GB | 1 GB | 1 GB |
| 🟦 Total Swap Space | [avg] | 1 GB | 1 GB | 1 GB | 1 GB |

*Figure 17. RAM and Swap Memory Graphs*

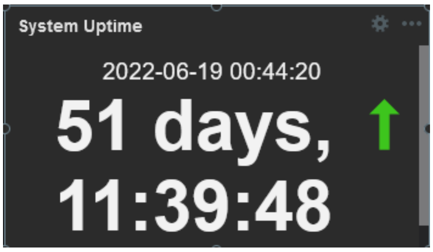
- <u>Disk Utilization Graph:</u>



| | | last | min | avg | max |
|---|---|---|---|---|---|
| 🟩 Percentage of Space Used on Disk | [avg] | 20 % | 20 % | 20 % | 20 % |

*Figure 18. % Used Space on Disk Graph*

- <u>System Uptime:</u>



System Uptime

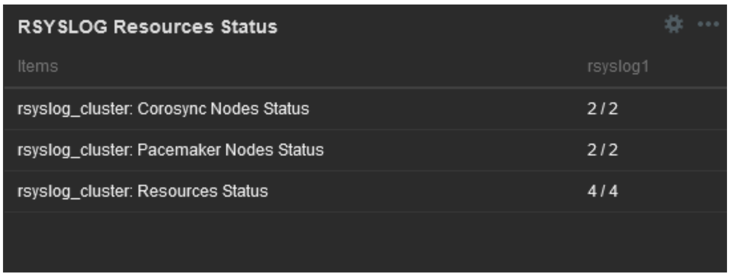
2022-06-19 00:44:20

51 days, ↑

11:39:48

*Figure 19. System Uptime Single Item*

- <u>RSYSLOG Resources and Services:</u>

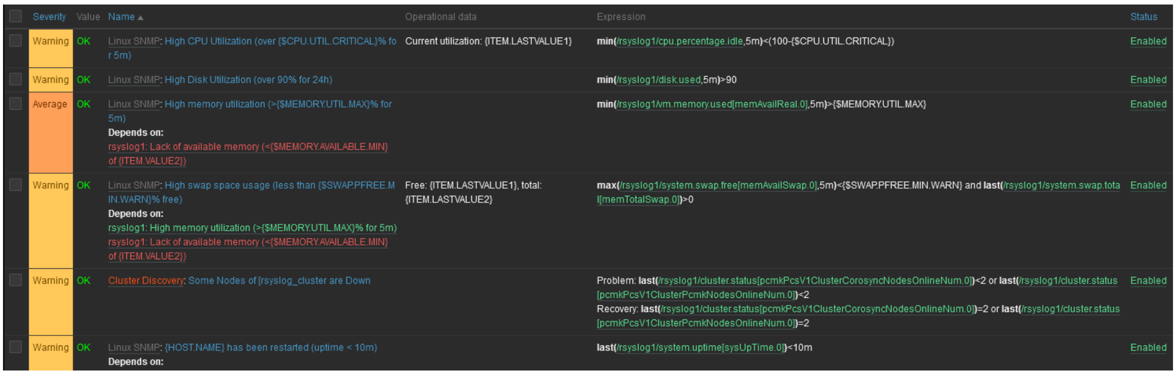


RSYSLOG Resources Status

| Items | rsyslog1 |
|---|---|
| rsyslog_cluster: Corosync Nodes Status | 2 / 2 |
| rsyslog_cluster: Pacemaker Nodes Status | 2 / 2 |
| rsyslog_cluster: Resources Status | 4 / 4 |

*Figure 20. RSYSLOG Resources and Services Status View*

## Created Triggers:



| □ | Severity | Value | Name ▲ | Operational data | Expression | Status |
|---|---|---|---|---|---|---|
| □ | Warning | OK | Linux SNMP: High CPU Utilization (over {$CPU.UTIL.CRITICAL}% for 5m) | Current utilization: {ITEM.LASTVALUE1} | min(/rsyslog1/cpu.percentage.idle,5m)<(100-{$CPU.UTIL.CRITICAL}) | Enabled |
| □ | Warning | OK | Linux SNMP: High Disk Utilization (over 90% for 24h) | | min(/rsyslog1/disk.used,5m)>90 | Enabled |
| □ | Average | OK | Linux SNMP: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)<br>**Depends on:**<br>rsyslog1: Lack of available memory (<{$MEMORY.AVAILABLE.MIN} of {ITEM.VALUE2}) | | min(/rsyslog1/vm.memory.used[memAvailReal.0],5m)>{$MEMORY.UTIL.MAX} | Enabled |
| □ | Warning | OK | Linux SNMP: High swap space usage (less than {$SWAP.PFREE.MIN.WARN}% free)<br>**Depends on:**<br>rsyslog1: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)<br>rsyslog1: Lack of available memory (<{$MEMORY.AVAILABLE.MIN} of {ITEM.VALUE2}) | Free: {ITEM.LASTVALUE1}, total: {ITEM.LASTVALUE2} | max(/rsyslog1/system.swap.free[memAvailSwap.0],5m)<{$SWAP.PFREE.MIN.WARN} and last(/rsyslog1/system.swap.total[memTotalSwap.0])>0 | Enabled |
| □ | Warning | OK | Cluster Discovery: Some Nodes of [rsyslog_cluster are Down | | Problem: last(/rsyslog1/cluster.status[pcmkPcsV1ClusterCorosyncNodesOnlineNum.0])<2 or last(/rsyslog1/cluster.status[pcmkPcsV1ClusterPcmkNodesOnlineNum.0])<2<br>Recovery: last(/rsyslog1/cluster.status[pcmkPcsV1ClusterCorosyncNodesOnlineNum.0])=2 or last(/rsyslog1/cluster.status[pcmkPcsV1ClusterPcmkNodesOnlineNum.0])=2 | Enabled |
| □ | Warning | OK | Linux SNMP: {HOST.NAME} has been restarted (uptime < 10m)<br>**Depends on:** | | last(/rsyslog1/system.uptime[sysUpTime.0])<10m | Enabled |

*Figure 21. Created Triggers*

Zabbix offers parameters and graphs discovery by using its Discovery rules and Templates. Also, advanced graphs and single item values can be shown in dashboards. It also offers trigger alarms based on hardware capabilities to keep critical services under control. Discovered services include critical measurements like CPU and Memory utilization. Furthermore, it can be programmable with small pieces of software in triggers configuration and prototypes, and also with JavaScript for pre-processing. So, it fulfils F1, F3, F4, F6 and NF2 requirements.

## Nagios Core:

On Nagios Core, thresholds and triggers can be predefined on the command definition file (commands.cfg) or on the service definition. In this case, I will create thresholds in the service definition to make it easy to use scripts. This threshold and trigger configuration is explained on *APPENDIX B*, section "*Trigger Configuration Options*".

To add available SNMP services to Nagios Core, I have implemented a Perl script called Nagios OID Discovery (nod.pl) to do this functionality. You can see this script on *APPENDIX I* in "*nod.pl*". To use it, we need to do this steps:

1) On terminal, go to Nagios files and generate a file with snmpwalk output with only OIDs for rsyslog1:

   ```
   cd /usr/local/nagios/etc/hosts
   snmpwalk -v2c -c public -Onq 10.0.3.213 .1 | awk '{print $1}' > oidlist.txt
   ```

2) Give permissions and execute script:

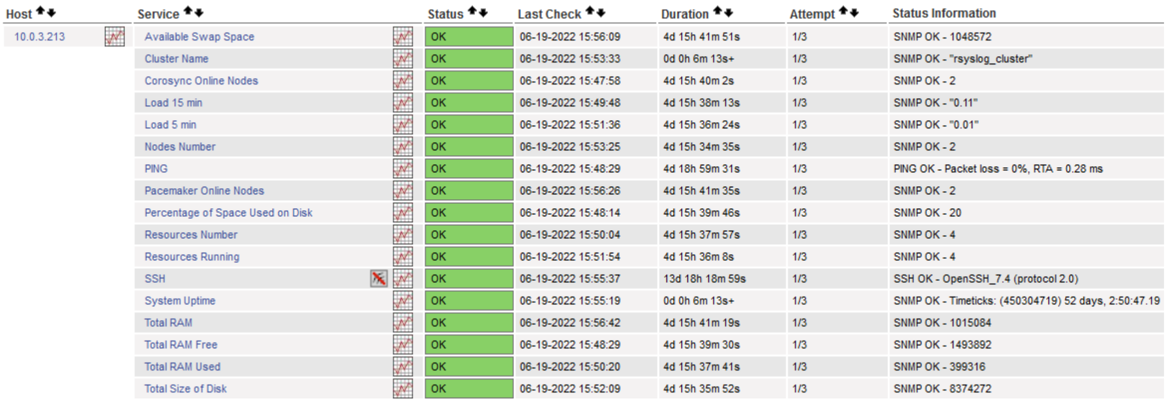
   ```
   chmod +rwx nod.pl
   ./nod.pl oidlist.txt 10.0.3.213 snmp_services_rsyslog1.cfg
   ```

3) Check that created file is correct and restart Nagios service:

   ```
   /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg →OK
   systemctl restart Nagios
   ```

4) Check results on web interface:

Discovered Items:



| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|---|---|---|---|---|---|---|
| 10.0.3.213 | Available Swap Space | OK | 06-19-2022 15:56:09 | 4d 15h 41m 51s | 1/3 | SNMP OK - 1048572 |
| | Cluster Name | OK | 06-19-2022 15:53:33 | 0d 0h 6m 13s+ | 1/3 | SNMP OK - "rsyslog_cluster" |
| | Corosync Online Nodes | OK | 06-19-2022 15:47:58 | 4d 15h 40m 2s | 1/3 | SNMP OK - 2 |
| | Load 15 min | OK | 06-19-2022 15:49:48 | 4d 15h 38m 13s | 1/3 | SNMP OK - "0.11" |
| | Load 5 min | OK | 06-19-2022 15:51:36 | 4d 15h 36m 24s | 1/3 | SNMP OK - "0.01" |
| | Nodes Number | OK | 06-19-2022 15:53:25 | 4d 15h 34m 35s | 1/3 | SNMP OK - 2 |
| | PING | OK | 06-19-2022 15:48:29 | 4d 18h 59m 31s | 1/3 | PING OK - Packet loss = 0%, RTA = 0.28 ms |
| | Pacemaker Online Nodes | OK | 06-19-2022 15:56:26 | 4d 15h 41m 35s | 1/3 | SNMP OK - 2 |
| | Percentage of Space Used on Disk | OK | 06-19-2022 15:48:14 | 4d 15h 39m 46s | 1/3 | SNMP OK - 20 |
| | Resources Number | OK | 06-19-2022 15:50:04 | 4d 15h 37m 57s | 1/3 | SNMP OK - 4 |
| | Resources Running | OK | 06-19-2022 15:51:54 | 4d 15h 36m 8s | 1/3 | SNMP OK - 4 |
| | SSH | OK | 06-19-2022 15:55:37 | 13d 18h 18m 59s | 1/3 | SSH OK - OpenSSH_7.4 (protocol 2.0) |
| | System Uptime | OK | 06-19-2022 15:55:19 | 0d 0h 6m 13s+ | 1/3 | SNMP OK - Timeticks: (450304719) 52 days, 2:50:47.19 |
| | Total RAM | OK | 06-19-2022 15:56:42 | 4d 15h 41m 19s | 1/3 | SNMP OK - 1015084 |
| | Total RAM Free | OK | 06-19-2022 15:48:29 | 4d 15h 39m 30s | 1/3 | SNMP OK - 1493892 |
| | Total RAM Used | OK | 06-19-2022 15:50:20 | 4d 15h 37m 41s | 1/3 | SNMP OK - 399316 |
| | Total Size of Disk | OK | 06-19-2022 15:52:09 | 4d 15h 35m 52s | 1/3 | SNMP OK - 8374272 |

*Figure 22. Nagios Core Discovered Items*

Discovered Graphs: Show only some examples

- Memory Graph:

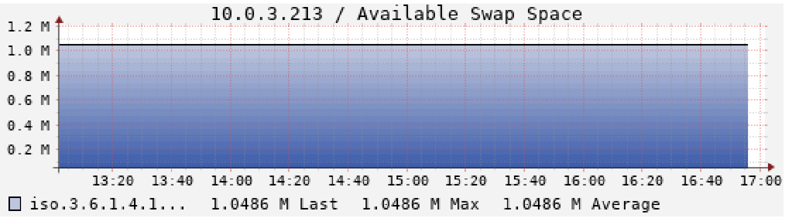


*Figure 23. Swap Memory Graph*

- Disk Utilization Graph:

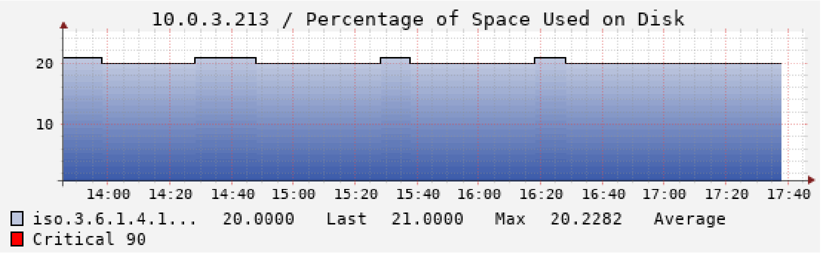


*Figure 24. CPU Utilization Graph*

Nagios Core offers parameters and graphs discovery by using a script, but Single Item Value cannot be shown in dashboards. It also offers triggering alarms, but they are not based on hardware capabilities because they scan all OIDs and create services at the same time, but their values are not scanned until the service is operative. Discovered services include critical measurements like CPU and Memory utilization. Furthermore, it can be programmed with a script written in Perl. So F3, F6 and NF2 requirements are achieved, but F1 and F4 are not.

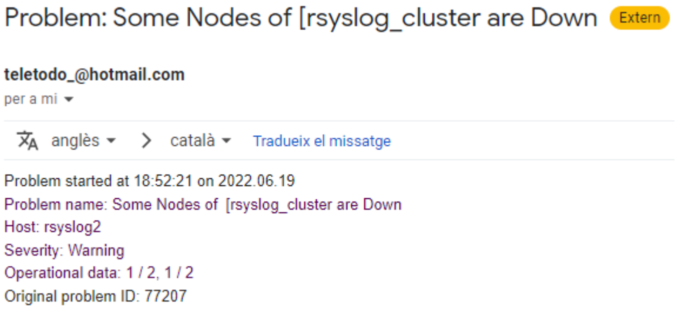
### 3.6.5. Configure email alerts and check they work

I will check email alerts work by shutdown rsyslog1 VM and see email responses from both monitoring tools:

**Zabbix:**

Zabbix email configuration is explained in *APPENDIX* C, section "*Email alerts configuration*".

After shutting down *rsyslog1*, *rsyslog2* raises an alert because node *rsyslog1* is down. So, I have received the mail shown in Figure 25.



Problem: Some Nodes of [rsyslog_cluster are Down Extern

teletodo_@hotmail.com
per a mi

anglès > català Tradueix el missatge

Problem started at 18:52:21 on 2022.06.19
Problem name: Some Nodes of [rsyslog_cluster are Down
Host: rsyslog2
Severity: Warning
Operational data: 1 / 2, 1 / 2
Original problem ID: 77207

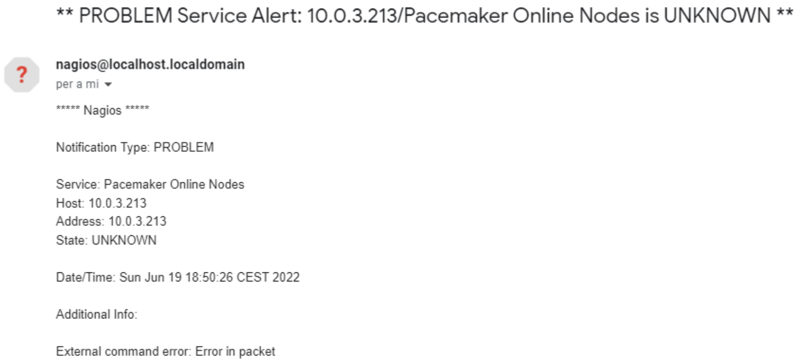
*Figure 25. Zabbix Mail Notification*

Zabbix is sending email alerts, so it fulfils the F5 requirement.

**Nagios Core:**

Nagios Core email configuration is explained in *APPENDIX B*, section "*Email alerts configuration*".

After shutting down *rsyslog1*, Nagios Core detects that the active node monitoring metric for this device is not being received and raises the alert. Then, I received the mail shown in Figure 26.



** PROBLEM Service Alert: 10.0.3.213/Pacemaker Online Nodes is UNKNOWN **

nagios@localhost.localdomain
per a mi

***** Nagios *****

Notification Type: PROBLEM

Service: Pacemaker Online Nodes
Host: 10.0.3.213
Address: 10.0.3.213
State: UNKNOWN

Date/Time: Sun Jun 19 18:50:26 CEST 2022

Additional Info:

External command error: Error in packet

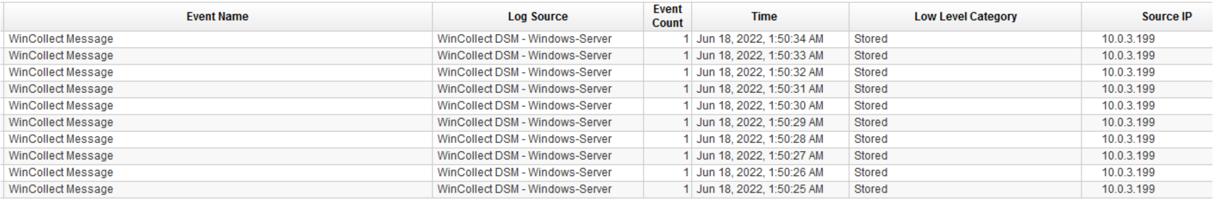
*Figure 26. Nagios Core Mail Notification*

Nagios Core is sending email alerts, so it fulfils the F5 requirement.

### 3.6.6. Show events on SIEM

On QRadar Community Edition, I can configure filters on the “Log activity” tab to show a specific Source IP or Port. I have tested this functionality for Lab Environment hosts and I obtained results shown in Figures 27, 28 and 29.

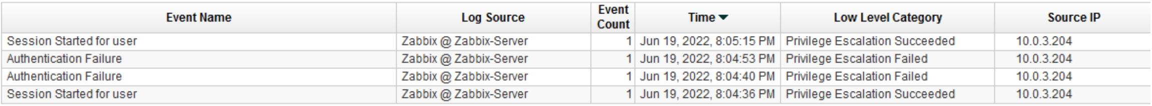
Windows 10 Host (IP Source→10.0.3.199):



| Event Name | Log Source | Event Count | Time | Low Level Category | Source IP |
|---|---|---|---|---|---|
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:34 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:33 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:32 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:31 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:30 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:29 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:28 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:27 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:26 AM | Stored | 10.0.3.199 |
| WinCollect Message | WinCollect DSM - Windows-Server | 1 | Jun 18, 2022, 1:50:25 AM | Stored | 10.0.3.199 |

*Figure 27. Windows 10 Host Events*

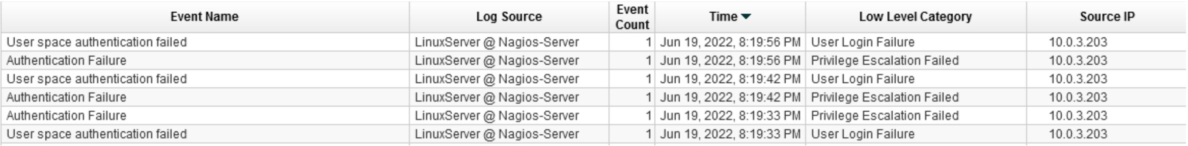
Zabbix Server (IP Source→10.0.3.204):



| Event Name | Log Source | Event Count | Time ▾ | Low Level Category | Source IP |
|---|---|---|---|---|---|
| Session Started for user | Zabbix @ Zabbix-Server | 1 | Jun 19, 2022, 8:05:15 PM | Privilege Escalation Succeeded | 10.0.3.204 |
| Authentication Failure | Zabbix @ Zabbix-Server | 1 | Jun 19, 2022, 8:04:53 PM | Privilege Escalation Failed | 10.0.3.204 |
| Authentication Failure | Zabbix @ Zabbix-Server | 1 | Jun 19, 2022, 8:04:40 PM | Privilege Escalation Failed | 10.0.3.204 |
| Session Started for user | Zabbix @ Zabbix-Server | 1 | Jun 19, 2022, 8:04:36 PM | Privilege Escalation Succeeded | 10.0.3.204 |

*Figure 28. Zabbix Server Events*

Nagios Core Server (IP Source→10.0.3.203):



| Event Name | Log Source | Event Count | Time ▾ | Low Level Category | Source IP |
|---|---|---|---|---|---|
| User space authentication failed | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:56 PM | User Login Failure | 10.0.3.203 |
| Authentication Failure | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:56 PM | Privilege Escalation Failed | 10.0.3.203 |
| User space authentication failed | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:42 PM | User Login Failure | 10.0.3.203 |
| Authentication Failure | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:42 PM | Privilege Escalation Failed | 10.0.3.203 |
| Authentication Failure | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:33 PM | Privilege Escalation Failed | 10.0.3.203 |
| User space authentication failed | LinuxServer @ Nagios-Server | 1 | Jun 19, 2022, 8:19:33 PM | User Login Failure | 10.0.3.203 |

*Figure 29. Nagios Core Server Events*

RSYSLOG Devices are configured only to forward events, so their events are not displayed on QRadar.

It can collect and filter log events from lab environment devices, so it fulfils requirement NF5.

# 4. Validation of Requirement Fulfilment

Before commenting on the test results, I want to say that requirement for the *Development of the system using only freeware software (NF7)* is met as a premise from the approach of the State of the Art. *Deployment of the system in a virtual environment based on an open-source operating system (NF6)* has been met during the design and creation of the laboratory environment.

After having tested both tools, I have been able to make a functional comparison of both to determine if Zabbix meets all the requirements and if Nagios Core could become an option as a monitoring tool. The validation results have been:

- <u>Zabbix:</u> It has managed to meet all the requirements demanded.
- <u>Nagios Core:</u> It meets most of the planned requirements, but it has failed to *provide Single Item Value (from F1)*, *set threshold based on hardware capabilities (from F4)*, *identify which user makes changes (from F8)* and *not give an easy-to-use web interface (from NF3)*.

The data obtained regarding the operation of Zabbix and Nagios Core are relevant to making the decision about which tool to use for monitoring the infrastructure.

On the one hand, we see that Zabbix offers many tools that make it easy to meet the needs of the project, although some additional configuration is required to obtain the expected results. The requirements of low importance as *Setting up threshold-triggered alarms based on hardware capabilities (F4)* and *Sending email alerts (F5)* and medium importance as *Show critical measurements (F3)*, *Having a list of users with different roles (F7)* and *Supporting web interface (NF3)* have been fulfilled without having to make much change. Some of the high-importance requirements have been met by doing little additional configuration as *Providing dashboards (F1)*, *Working with agentless monitoring (NF1)*, *Allowing basic security methods (NF4)* and *Compatible with existing operating systems (NF8)*. But, other important requirements such as *Providing device discovery and integration tool (F2)* and *Adding parameters and graphs by using available OIDs (F6)* have required a lot of dedication for their correct operation and *Programmable based on scripts or small pieces of software (NF2)* has turned out to be useful and are been used in triggers definitions, item and trigger prototypes and for pre-processing by using JavaScript.

On the other hand, we see that Nagios Core is a more limited tool, but thanks to its configuration by terminal and configuration files, it makes necessary and helpful the use of *Programmable based on scripts or small pieces of software (NF2)* to accomplish some of the requirements. In comparison with Zabbix, *Setting up threshold-triggered alarms based on hardware capabilities (F4)* is much more intuitive to implement, but *Sending email alerts (F5)* requires additional configuration which slows down the process a little bit. Also, *Show critical measurements (F3)* and *Having a list of users with different roles (F7)* are more intuitive and require a similar configuration, but the *Web interface (NF3)* is much more complicated to use, despite providing other features. *Providing dashboards (F1)* has required some configuration and the results are much lower than those offered by default in Zabbix. *Work with agentless monitoring (NF1)*, *Allowing basic security methods (NF4)* and *Compatible with existing operating systems (NF8)* have been fulfilled with little additional configuration. But, other important requirements such as *Providing device discovery and integration tool (F2)* and *Adding parameters and graphs by using available OIDs (F6)* have required a lot of dedication for their correct operation and *Programmable* *based on scripts or small pieces of software (NF2)* has turned out to be very useful for their development.

Finally, I want to emphasize that both tools have had their difficulties in some key points when developing the project. Even so, Zabbix has managed to meet expectations and Nagios Core has given a comparative point of view when designing, implementing and using the required features.

# 5. Future Work

Starting from the existing project, the following steps are proposed for its implementation and improvement in the client infrastructure:

First of all, it would be worthwhile to include more specific metrics to monitor, such as monitoring incoming and outgoing traffic on the different interfaces of the RSYSLOG Clusters.

Secondly, the current RSYSLOG dashboard should be improved to perform the daily check more efficiently.

Third, you should review which Windows devices could be ICMP polled to keep them under control and be able to act instantly if the device stops being polled.

To finish, it would be interesting to add more SNMP Traps and customize them for the existing SIEM Cluster for better monitoring of it from the monitoring server.

# 6. Budget

## 6.1. Direct Costs

The project has been developed, implemented and tested only by one person (Project Developer) and the Project Manager has been supervising the project, guiding and advising the Project Developer to ensure that the tasks and milestones are carried out correctly.

Looking at the Gantt Diagram in section 1.6.4, the project was scheduled to be done in 460 hours, but due to deviations from the initial plan, I have added 410 more hours (870 hours). In addition, we have held 9 online follow-up meetings with the Project Manager of 1.5 hours each.

Annual Net Salary is based on Glassdoor information and Proportional Salary is calculated proportionally with worked time (9 months):

| Developed Task | Anual Gross Salary | Proportional Gross Salary | Anual Net Salary | Proportional Net Salary | Social Security Contribution |
|---|---|---|---|---|---|
| Analyst[71] | 38.834 € | 29.126 € | 27.184 € | 20.388 € | 6.116 € |
| Designer[72] | 34.286 € | 25.714 € | 24.000 € | 18.000 € | 5.400 € |
| Project Developer[73] | 43.314 € | 32.486 € | 30.320 € | 22.740 € | 6.822 € |
| Project Manager[74] | 57.143 € | 42.857 € | 40.000 € | 30.000 € | 9.000 € |
| Tester[75] | 38.667 € | 29.000 € | 27.067 € | 20.300 € | 6.090 € |
| **TOTAL** | **212.244 €** | **159.183 €** | **148.571 €** | **111.428 €** | **33.428 €** |

*Table 4. Annual and Proportional Salary and Social Security Contribution*

Price/Hour calculation is made using working days in these 9 months (184 days) and using a working day of 8 hours (1472 hours):

| Developed Task | Estimated Hours | Price/Hour (Net) | Final Cost |
|---|---|---|---|
| Analyst | 180h | 13,85 €/h | 2.493,10 € |
| Designer | 30h | 12,23 €/h | 366,85 € |
| Project Developer | 145h | 15,45 €/h | 2.240,01 € |
| Project Manager | 13,5h | 20,38 €/h | 275,14 € |
| Tester | 125h | 13,79 €/h | 1.723,87 € |
| **TOTAL** | **493,5h** | **-** | **7.098,96 €** |

*Table 5. Direct Costs Calculation*

## 6.2. Indirect Costs

Indirect costs are relevant to a Company Project. They are always in use on this project and are essential tools for its correct development. Some of them are the rent of the office, the cost of water, electricity and internet services, the licenses used and, finally, the equipment used and its amortization. The impact of teleworking has also been taken into account.

First, I have used is Microsoft 365 Business Standard license for Word, Excel, PowerPoint, OneNote, Teams and Outlook use. The monthly price for this license cost 10,50 € and I have been using it for around 200 hours. I also use MobaXTerm software for tunnelling to the company server from home and Kanban tool for organization and methodology.

Second, the cost of hardware has been calculated based on the costs that appear in the company's invoices for specific equipment. Mostly I have been working with a laptop, a company-provided server and virtual devices working on the client's net (clusters, SIEM system, etc). SIEM used is a free version with limited resources and capacities. I have also taken into account a year of 250 working days an 8-hour day. Below you can see the costs associated with the hardware used:

| Product | Market Price | Shelf Life | Hours Used | Depreciation |
|---|---|---|---|---|
| Lenovo ThinkPad T490[76] | 1.580 € | 4 years | 493,5 hours | 97,47 € |
| Company Server | 800 € | 4 years | 313,5 hours | 31,35 € |
| MobaXTerm[77] | 49 € | 1 year | 180 hours | 4,41 € |
| Kanban[78] | 5 € | 1 month | 30 hours | 0,90 € |
| Microsoft License[79] | 10,50 € | 1 month | 200 hours | 12,60 € |
| **TOTAL** | | | | **146,73 €** |

*Table 6. Hardware Costs*

Some free-to-use tools are also been used:

- Creately (Diagrams)
- Google Services (Meet, Drive and Gmail)
- VPN
- IBM QRadar Community Edition

Fourth, the rent of the office is calculated and the cost of water, electricity and internet services is calculated with a cost of approximately 6.000 €/month. The office has a capacity of about 60 people, so the monthly cost per person is about 100 €/month. To simplify, I consider 9 months working in the office because telework costs are approximately the same as working at the office. So, the office and its services final and teleworking costs for this 9 months are 900 €.

Fifth, the last cost to take into account is transport cost. For most of this time, I have been working in a hybrid work model, with 3 days at work and 2 at home. I used a T-Jove from September to November (80 €). We have about 2 months of teleworking between December and January caused by the high incidence of Covid. Then in February, the hybrid model was reinstated and it went back to normal until the project was finished. In this last period, I have used five T-Usual (40 €/month). So, the final transport cost is 280 €.

Finally, to develop this project we have to take into account overhead or structural expenses to cover the expenses of human resources and administration. I have allocated a 10% of Indirect Costs to cover it.

| Indirect Costs | Estimated Cost |
|---|---|
| Transport | 280 € |
| Office Rent + Services | 900 € |
| Hardware | 146,73 € |
| **AMOUNT** | **1.326,73 €** |
| Overhead Expenses | 132,67 € |
| **TOTAL** | **1.459,40 €** |

*Table 7. Indirect Costs*

### 6.3. Contingencies

I have saved 20% of project direct and indirect costs to cover deviations from the initial plan. Once I have calculated these costs and applied this percentage, contingencies are 1.711,67 €.

### 6.4. Unexpected Costs

Estimated costs for hypothetical unforeseen events. They have to take into account to avoid budget problems if something wrong happens. In this project, hardware malfunction is the most relevant issue. It cost about 500 € to repair the laptop for several issues. The company server has a backup server, so it has no extra costs because it is very difficult to lose data. So, the unexpected costs are 500 €.

### 6.5. Total Costs

Finally, the total costs associated with this project are:

| Type of Costs | Costs |
|---|---|
| Direct Costs | 7.098,96 € |
| Indirect Costs | 1.459,40 € |
| Contingencies | 1.711,67 € |
| Unexpected Costs | 500 € |
| **TOTAL COSTS** | **10.770,03 €** |

*Table 8. Total Costs*

# 7. Conclusions

The project began with the approach of covering a need within the tasks developed in the company. After a needs analysis, it was decided that the project should be focused on finding a freeware monitoring tool that would cover a series of requirements. Then, I analysed the possible risks and contingencies for them and made a State of the Art of the most used and recognized freeware monitoring tools. Subsequently, I proposed a selection criterion and applied it to determine which tools were most suitable for the development of the project. The result was a bit tight, and after consulting with the System Administrator, I decided to add the tool with the second-best result to the implementation and development of the project so that I could use it as a comparison. Later, I made the design of the lab environment and, after the approval of the System Administrator, I started the creation of it. Once created and made some of the initial implementations, I proposed and developed a test of both tools that would help me to evaluate their characteristics and their performance with tasks focused on meeting the requirements. Finally, I have presented the results obtained.

In conclusion, I have determined that Zabbix is the best option among the freeware monitoring tools studied to carry out the monitoring process and automate the process of adding devices to it. The comparison with Nagios Core has helped me to see another development and implementation option and to see what points could be improved in the solution obtained in Zabbix. Despite not having obtained as good results in the test as Zabbix and not fully meeting 4 of the 16 requirements, I think that someone with more knowledge of programming and SNMP protocols could get a better result and make Nagios Core out one option to consider.

As for recommendations, I can say that a good organization, planning and determination of the scope of the project can greatly facilitate its development. From my point of view, the project has been very enriching for the development of my daily work within the project. It has revealed many possibilities and has given me knowledge on a subject that I did not dominate.

Regarding the impact of my project, I have been able to develop a task that will greatly facilitate the decision-making of a monitoring tool for the client's infrastructure and the work carried out will serve as a guide for its installation in the client's environment. The System Administrator is very happy with the work done and is expected to present it to the customer for approval and start the installation of the infrastructure.

To finish, I want to highlight that the development of the work has involved some 493.5 hours divided into analysis, design, testing, development and consultations with the project manager. I also want to highlight that the project has required 260 lines of code (Perl and JavaScript), 285 lines of configuration code added and 163 installation and configuration steps.

# Bibliography

[1] "Nagios Core". [Online] Available: https://www.nagios.org/projects/nagios-core/ [Accessed: 21 November 2021].

[2] “About Nagios Core” [Online] Available: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/about.html#whatis [Accessed: 21 November 2021].

[3] “What is Prometheus?”. Prometheus Authors 2014-2021 [Online] Available: https://prometheus.io/docs/introduction/overview/ [Accessed 28 November 2021].

[4] "What is Grafana OSS”. The Cacti Group, Inc. 2004-2021 [Online] Available: https://grafana.com/docs/grafana/latest/introduction/oss-details/ [Accessed: 19 December 2021].

[5] "2 What is Zabbix”. Zabbix SIA 2001-2022 [Online] Available: https://www.zabbix.com/documentation/current/en/manual/introduction/about [Accessed: 28 November 2021].

[6] "3 Zabbix features”. Zabbix SIA 2001-2022 [Online] Available: https://www.zabbix.com/documentation/current/en/manual/introduction/features [Accessed: 28 November 2021].

[7] "OpenNMS Horizon”. [Online] Available: https://www.opennms.com/horizon/ [Accessed: 5 December 2021].

[8] "Feature Overview”. [Online] Available: https://www.opennms.com/opennms-feature-list/ [Accessed: 5 December 2021].

[9] "What is Cacti?”. The Cacti Group, Inc. 2004-2021 [Online] Available: https://www.cacti.net/info/cacti [Accessed: 12 December 2021].

[10] "Features”. The Cacti Group, Inc. 2004-2021 [Online] Available: https://www.cacti.net/info/features [Accessed: 12 December 2021].

[11] “Security Information and Event Management (SIEM) Reviews and Ratings”. [Online] Available: https://www.gartner.com/reviews/market/security-information-event-management [Accessed: 19 December 2021]

[12] “QRadar SIEM”. [Online] Available: https://www.incibe.es/protege-tu-empresa/catalogo-de-ciberseguridad/listado-soluciones/qradar-siem [Accessed: 28 December 2021]

[13] “IBM Security QRadar Community Edition”. [Online] Available: https://www.ibm.com/community/qradar/ce/ [Accessed 28 December 2021]

[14] “What is SIEM?”. [Online] Available: https://www.ibm.com/topics/siem [Accessed: 14 January 2022]

[15] “User interface tabs”. [Online] Available: https://www.ibm.com/docs/en/qradar-on-cloud?topic=SSKMKU/com.ibm.qradar.doc/c_qradar_ui_tabs.html [Accessed: 14 January 2022]

[16] "Nagios Core - Performance Graphs Using PNP4Nagios”. [Online] Available: https://support.nagios.com/kb/article/nagios-core-performance-graphs-using-pnp4nagios-801.html [Accessed: 7 February 2022].

[17] “Grafana Supports for Prometheus”. [Online] Available: https://prometheus.io/docs/visualization/grafana/ [Accessed: 7 February 2022].

[18] “Visualization”. [Online] Available: https://www.zabbix.com/documentation/current/en/manual/config/visualization [Accessed: 7 February 2022].

[19] “Getting started with HELM”. [Online] Available: https://docs.opennms.com/helm/7.0.0/getting_started/index.html [Accessed: 7 February 2022].

[20] “Protocol Monitoring with Nagios”. [Online] Available: https://www.nagios.com/solutions/protocol-monitoring/ [Accessed: 8 February 2022].

[21] “Exporters and Integrations”. [Online] Available: https://prometheus.io/docs/instrumenting/exporters/ [Accessed: 8 February 2022].

[22] “Item Types”. [Online] Available: https://www.zabbix.com/documentation/current/en/manual/config/items/itemtypes [Accessed: 8 February 2022].

[23] “Service Monitors”. [Online] Available: https://docs.opennms.com/horizon/29/reference/service-assurance/introduction.html [Accessed: 8 February 2022].

[24] Ian Berry, Tony Roman, Larry Adams, J.P. Pasnak, Jimmy Conner, Reinhard Scheck, and Andreas Braun. The Cacti Manual, 2017.

[25] "Nagios Bulk Import (nbi.pl)". [Online] Available: https://exchange.nagios.org/directory/Addons/Configuration/Nagios-Bulk-Import-%28nbi-2Epl%29/details [Accessed: 9 February 2022].

[26] "Configuration". [Online] Available: https://prometheus.io/docs/prometheus/latest/configuration/configuration/ [Accessed: 9 February 2022].

[27] "Low-level discovery". [Online] Available: https://www.zabbix.com/documentation/current/it/manual/discovery/low_level_discovery#discovery-rule [Accessed: 9 February 2022].

[28] "Auto Discovery". [Online] Available: https://docs.opennms.com/horizon/29/operation/provisioning/auto-discovery.html [Accessed: 9 February 2022].

[29] "Automation Discovered Devices". [Online] Available: https://github.com/Cacti/documentation/blob/develop/Discovered-Devices.md [Accessed: 9 February 2022].

[30] "Nagios: Set Threshold". [Online] Available: https://www.server-world.info/en/note?os=CentOS_6&p=nagios&f=3 [Accessed: 10 February 2022].

[31] "Alerting Rules". [Online] Available: https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/ [Accessed: 10 February 2022].

[32] "Problem Detection". [Online] Available: https://www.zabbix.com/problem_detection [Accessed: 10 February 2022].

[33] "Add a Threshold". [Online] Available: https://grafana.com/docs/grafana/latest/panels/specify-thresholds/add-a-threshold/ [Accessed: 10 February 2022].

[34] "Cacti/plugin_hold". [Online] Available: https://github.com/Cacti/plugin_thold [Accessed: 10 February 2022].

[35] "Alert notifications". [Online] Available: https://grafana.com/docs/grafana/latest/alerting/old-alerting/notifications/ [Accessed: 11 February 2022].

[36] "Receiving problem notification". [Online] Available: https://www.zabbix.com/documentation/current/en/manual/quickstart/notification [Accessed: 11 February 2022].

[37] "Authentication And Authorization In The CGIs". [Online] Available: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/cgiauth.html [Accessed: 14 February 2022].

[38] "About users and permissions". [Online] Available: https://grafana.com/docs/grafana/next/administration/manage-users-and-permissions/about-users-and-permissions/ [Accessed: 14 February 2022].

[39] "Permissions". [Online] Available: https://www.zabbix.com/documentation/6.0/en/manual/config/users_and_usergroups/permissions [Accessed: 14 February 2022].

[40] "Chapter 9. User Management". [Online] Available: https://files.cacti.net/docs/html/user_management.html [Accessed: 14 February 2022].

[41] "Nagios Command Configuration". [Online] Available: https://support.nagios.com/kb/article/nagios-core-performance-graphs-using-pnp4nagios-801.html#Nagios_Command_Config [Accessed: 15 February 2022].

[42] "How to configure Grafana as code". [Online] Available: https://grafana.com/blog/2020/02/26/how-to-configure-grafana-as-code/ [Accessed: 15 February 2022].

[43] "Scripts". [Online] Available: https://www.zabbix.com/documentation/current/en/manual/web_interface/frontend_sections/administration/scripts [Accessed: 15 February 2022].

[44] Kevin der Kinderen, "Chapter 16. How To Simplest Method of Going from Script to Graph (Walkthrough)". [Online] Available: https://files.cacti.net/docs/html/how_to.html [Accessed: 15 February 2022].

[45] "Event Handlers". [Online] Available: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/eventhandlers.html [Accessed: 16 February 2022].

[46] "Alerting Rules". [Online] Available: https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/ [Accessed: 16 February 2022].

[47] "Event Configuration". [Online] Available: https://docs.opennms.com/horizon/29/operation/events/event-configuration.html [Accessed: 16 February 2022].

[48] "Object Definition". [Online] Available: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/objectdefinitions.html#host [Accessed: 17 February 2022].

[49] "Configuration Overview". [Online] Available: https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/config.html [Accessed: 17 February 2022].

[50] "Grafana". Grafana Labs 2022. [Online] Available: https://grafana.com/grafana/?plcmt=footer [Accessed: 17 February 2022].

[51] "Zabbix Web Frontend". Zabbix LLC 2001-2022. [Online] Available: https://www.zabbix.com/zabbix_web_frontend [Accessed: 17 February 2022].

[52] Tobias Oetiker. "About RRDtool". OETIKER+PARTNER AG 20/02/2017 [Online] Available: https://oss.oetiker.ch/rrdtool/ [Accessed: 17 February 2022].

[53] "Nagios Core - Configuring SSL/TLS". [Online] Available: https://support.nagios.com/kb/article/nagios-core-configuring-ssl-tls-595.html [Accessed: 28 February 2022].

[54] "SECURING PROMETHEUS API AND UI ENDPOINTS USING TLS ENCRYPTION". Prometheus Authors, 2014-2022. [Online] Available: https://prometheus.io/docs/guides/tls-encryption/ [Accessed: 28 February 2022].

[55] "17. Encryption". Zabbix SIA, 2001-2022 [Online] Available: https://www.zabbix.com/documentation/current/en/manual/encryption [Accessed: 28 February 2022].

[56] "SSLCertMonitor". [Online] Available: https://docs.opennms.com/horizon/29/reference/service-assurance/monitors/SSLCertMonitor.html [Accessed: 28 February 2022].

[57] "Nagios Core - Installing Nagios Core From Source". [Online] Available: https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html [Accessed: 1 March 2022].

[58] "FIRST STEPS WITH PROMETHEUS". Prometheus Authors, 2014-2022. [Online] Available: https://prometheus.io/docs/introduction/first_steps/#downloading-prometheus [Accessed: 1 March 2022].

[59] "Download and install Zabbix". Zabbix LLC 2001-2022. [Online] Available: https://www.zabbix.com/download [Accessed: 1 March 2022].

[60] "Operating systems". The OpenNMS Group. [Online] Available: https://docs.opennms.com/horizon/29/deployment/core/system-requirements.html#operating-systems-core [Accessed: 1 March 2022].

[61] "Chapter 2. Installing Under Unix". [Online] Available: https://files.cacti.net/docs/html/install_unix.html [Accessed: 1 March 2022].

[62] "Operating System (OS) Monitoring with Nagios". Nagios Enterprises, 2009-2022 [Online] Available: https://www.nagios.com/solutions/operating-system-monitoring/ [Accessed: 1 March 2022].

[63] "MONITORING LINUX HOST METRICS WITH THE NODE EXPORTER". Prometheus Authors, 2014-2022. [Online] Available: https://prometheus.io/docs/guides/node-exporter/ [Accessed: 2 March 2022].

[64] "Zabbix + Linux". Zabbix LLC 2001-2022. [Online] Available: https://www.zabbix.com/integrations/linux [Accessed: 2 March 2022].

[65] "Zabbix + Windows". Zabbix LLC 2001-2022. [Online] Available: https://www.zabbix.com/integrations/windows [Accessed: 2 March 2022].

[66] "vSphere". VMWare, Inc. 2022. [Online] Available: https://www.vmware.com/products/vsphere.html [Accessed: 2 March 2022].

[67] "Proxmox Virtual Environment". Proxmox Server Solutions GmbH 2004-2022 [Online] Available: https://www.proxmox.com/en/proxmox-ve [Accessed: 2 March 2022].

[68] "Chapter 1. High Availability Add-On Overview", Red Hat, Inc. 2022. [Online] Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_overview/ch-introduction-haao#s1-clstr-basics-HAAO [Accessed: 7 March 2022].

[69] "Creating Self-Signed Certificates and Keys with OpenSSL". MariaDB, 2022. [Online] Available: https://mariadb.com/docs/security/data-in-transit-encryption/create-self-signed-certificates-keys-openssl/ [Accessed: 11 March 2022]

[70] "Apache 2 : SSL/TLS Setting". Server World, 2007-2022. [Online] Available: https://www.server-world.info/en/note?os=Debian_10&p=httpd&f=8 [Accessed: 11 March 2022]

[71] "Average Base Salary for Analyst". [Online] Available: https://www.glassdoor.es/Salaries/analyst-salary-SRCH_KO0,7.htm?countryRedirect=true [Accessed: 21 Abril 2022]

[72] "Average Base Salary for Web Designer". [Online] Available: https://es.talent.com/salary?job=dise%C3%B1ador+web [Accessed: 21 Abril 2022]

[73] "Average Base Salary for Web Developer". [Online] Available: https://www.glassdoor.es/Sueldos/barcelona-desarrollador-web-sueldo-SRCH_IL.0,9_IM1015_KO10,27.htm?clickSource=searchBtn [Accessed: 21 Abril 2022]

[74] "Average Base Salary for Project Manager". [Online] Available: https://www.glassdoor.es/Sueldos/barcelona-project-manager-sueldo-SRCH_IL.0,9_IM1015_KO10,25.htm?clickSource=searchBtn [Accessed: 21 Abril 2022]

[75] "Average Base Salary for Tester". [Online] Available: https://www.glassdoor.es/Sueldos/barcelona-tester-sueldo-SRCH_IL.0,9_IM1015_KO10,16.htm?clickSource=searchBtn [Accessed: 21 Abril 2022]

[76] "Amazon Price History". [Online] Available: https://camelcamelcamel.com/product/B07QBQZSYY [Accessed: 22 Abril 2022]

[77] "MobaXterm Professional registration". [Online] Available: https://mobaxterm.mobatek.net/subscription.html [Accessed: 22 Abril 2022]

[78] Kanban Tool Pricing. [Online] Available: https://kanbantool.com/pricing [Accessed: 22 Abril 2022]

[79] Microsoft 365 Pricing. [Online] Available: https://www.microsoft.com/es-es/microsoft-365/business/compare-all-microsoft-365-business-products?activetab=tab:primaryr2 [Accessed: 22 Abril 2022]

# Glossary

A list of all acronyms and the meaning they stand for.

| | |
|---|---|
| **API** | Application Programming Interface |
| **CA** | Certification Authority |
| **CPU** | Central Processing Unit |
| **CGI** | Computer Generated Imagery |
| **DoS** | Denial of Service |
| **ESM** | Enterprise Security Manager (for SIEMs) |
| **HA** | High Availability |
| **HP-UX** | Hewlett Packard Unix |
| **HTTPS** | HyperText Transfer Protocol Secure |
| **ICMP** | Internet Control Message Protocol |
| **IIS** | Internet Information Server |
| **IP** | Internet Protocol |
| **LAMP** | Linux Apache MySQL PHP |
| **LDAP** | Lightweight Directory Access Protocol |
| **MariaDB** | Maria Database |
| **MIB** | Management Information Base |
| **MySQL** | My Structured Query Language |
| **NBI** | Nagios Bulk Import |
| **NGINX** | Engine-ex |
| **NMAP** | Network Map |
| **OS** | Operative System |
| **OID** | Object Identifiers |
| **PHP** | Hypertext Preprocessor |
| **POC** | Proof of Concept |
| **PostgreSQL** | Postgre Structured Query Language |
| **Regex** | Regular Expressions |
| **REST** | REpresentational State Transfer |
| **RRDtool** | Round-Robin Database Tool |
| **RSYSLOG** | Rocket-Fast System for Log Processing |
| **SAN** | Storage Area Network |
| **SHA** | Secure Hash Algorithm |
| **SIEM** | Security Information and Event Management |

| | |
|---|---|
| **SNMP** | Simple Network Management Protocol |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **TSDB** | Time Series Database |
| **UI** | User Interface |
| **VIPA** | Virtual Internet Protocol Adress |
| **VM** | Virtual Machine |