Report:

ZABBIX

Monitoring and Performance Management with Zabbix: Deployment and Configuration of Ubuntu, Windows, PaloAlto, FortiGate, and ESXi Platforms.

ET.TAKI Omar

Table of Contents:

Introduction **4**
Install Zabbix on Ubuntu server 22.04 **5**
Architecture **5**
Install Zabbix server, frontend, and agent 6
Configure database 8
Start Zabbix server and agent processes 12
Configure Zabbix frontend 12
Login to frontend using Zabbix default login credentials 17
The Zabbix dashboard: 18
Creating and configuring Zabbix users, user groups, and user roles 20
Creating User Groups 20
Monitoring Ubuntu machine with Zabbix using agent **25**
Updating our System 25
Installing Zabbix Agent 25
Configuring Zabbix Agent 26
Starting and Enabling Zabbix Agent Service 27
Adding the Zabbix Client to the Zabbix Server 27
Monitoring Windows machine with Zabbix using agent **30**
Installing Zabbix Agent 30
Configuring Zabbix Agent 31
Adding the Zabbix Client to the Zabbix Server 32
Monitoring Palo Alto with Zabbix via SNMPv3 **34**
Configuring and enabling SNMP on Palo Alto 34
Adding PaloAlto to Zabbix server 37
Monitoring FortiGate with Zabbix via SNMPv3 **39**
Configuring and enabling SNMP on FortiGate 39
Adding FortiGate to Zabbix server 41
Monitoring ESXI with Zabbix via SNMPv3 **42**
Configuring and enabling SNMP on ESXI 42
Enabling SSH Access on ESXi 42
ESXi SNMP Configuration 43
Configuring parameters of an SNMP agent 44
Configuring ESXi Firewall 46
Configuring SNMP v3 46
Adding ESXI to Zabbix server 47
Zabbix Maps for Better Visualization **49**
Creating a new map 50
Configuring new map 51
Linking map elements 53
Conclusion **55**

Introduction

The Zabbix project is focused on the deployment and configuration of a comprehensive monitoring and performance management platform. This project aims to implement Zabbix, a powerful tool for surveillance and analysis, to ensure efficient monitoring of various components within the network architecture.

The primary objective of this project is to establish a centralized monitoring system using Zabbix, enabling real-time visibility into the performance and health of critical network elements. By leveraging Zabbix's capabilities, including data storage in a relational database and a user-friendly web interface, the project seeks to empower administrators with the tools necessary to effectively monitor and manage the network infrastructure.

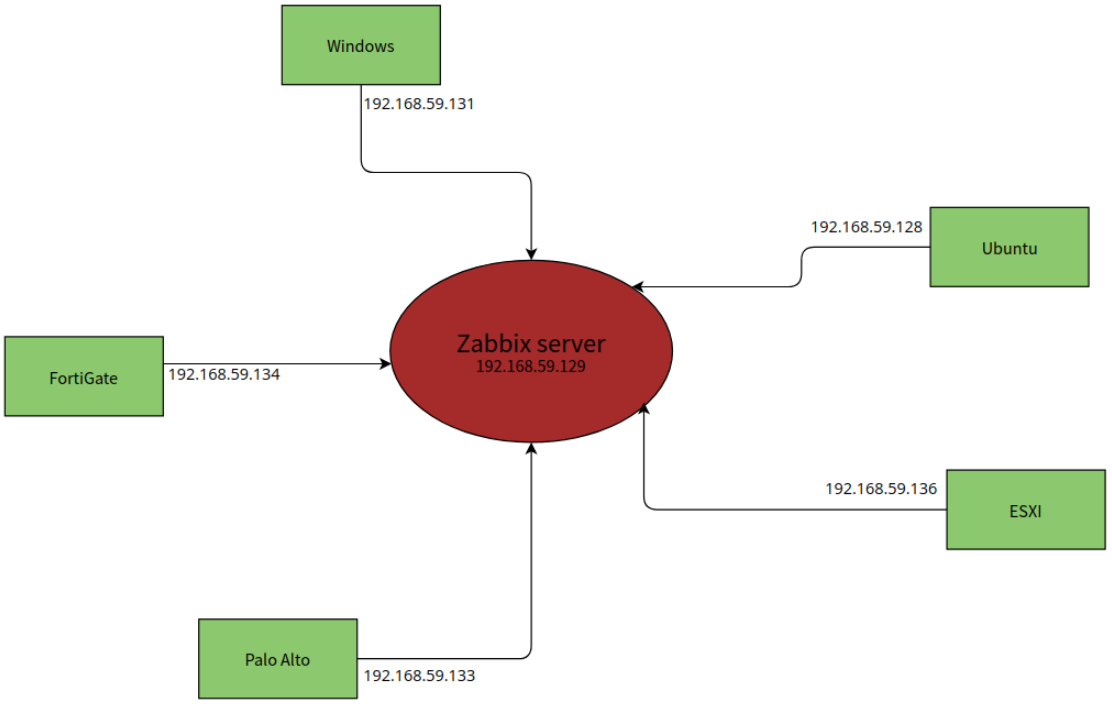
Throughout the project, special attention will be given to deploying Zabbix on an Ubuntu server, as well as configuring client machines and servers, such as Ubuntu, Windows, PaloAlto, FortiGate, and ESXi, to act as monitoring agents. These agents will facilitate the collection and analysis of vital system metrics, ensuring prompt detection of issues and proactive performance optimization.

By implementing Zabbix, this project aims to provide a holistic view of the network, empowering administrators with the insights required to make informed decisions and ensure optimal performance. The following sections will delve into the specific tasks involved in the installation, configuration, and management of Zabbix, as well as highlight the expected benefits and outcomes of this project.

Install Zabbix on Ubuntu server 22.04

Architecture

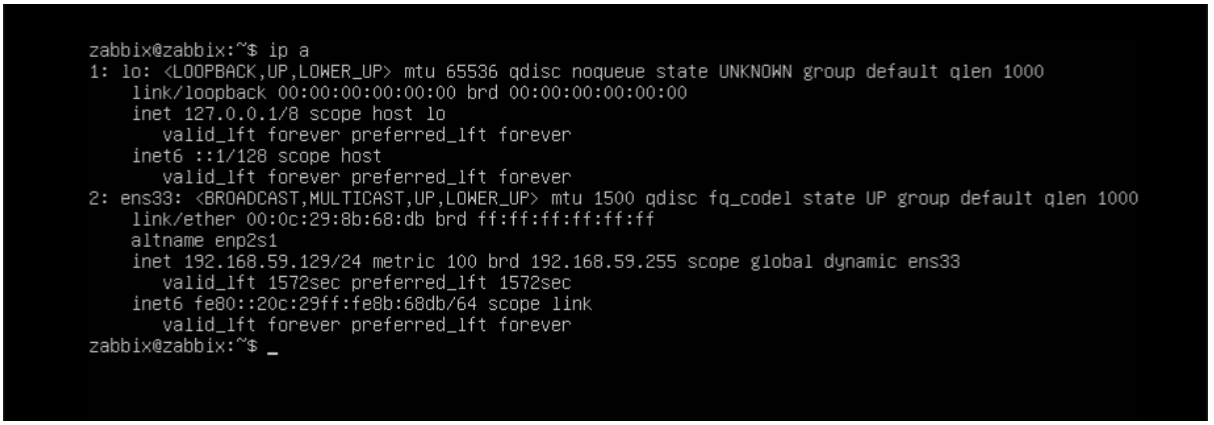
Here’s the architecture of the network :



Install Zabbix server, frontend, and agent

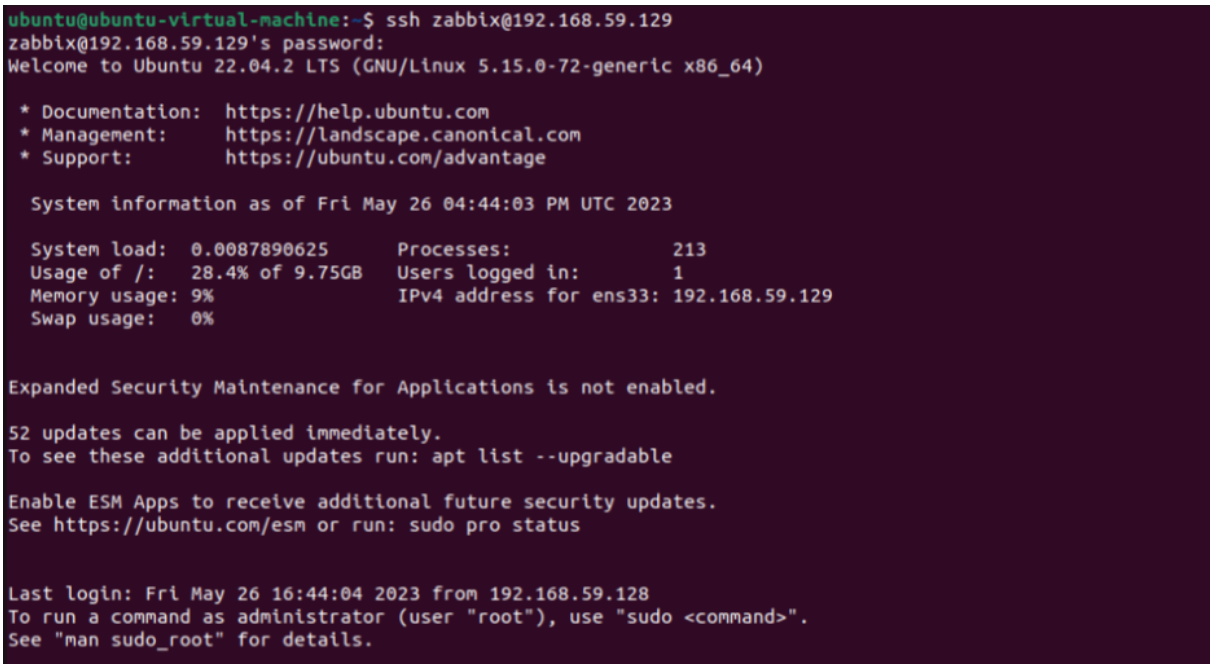
On ubuntu server, to check for the internal IP address, we execute the following command:

```
> ip a
```

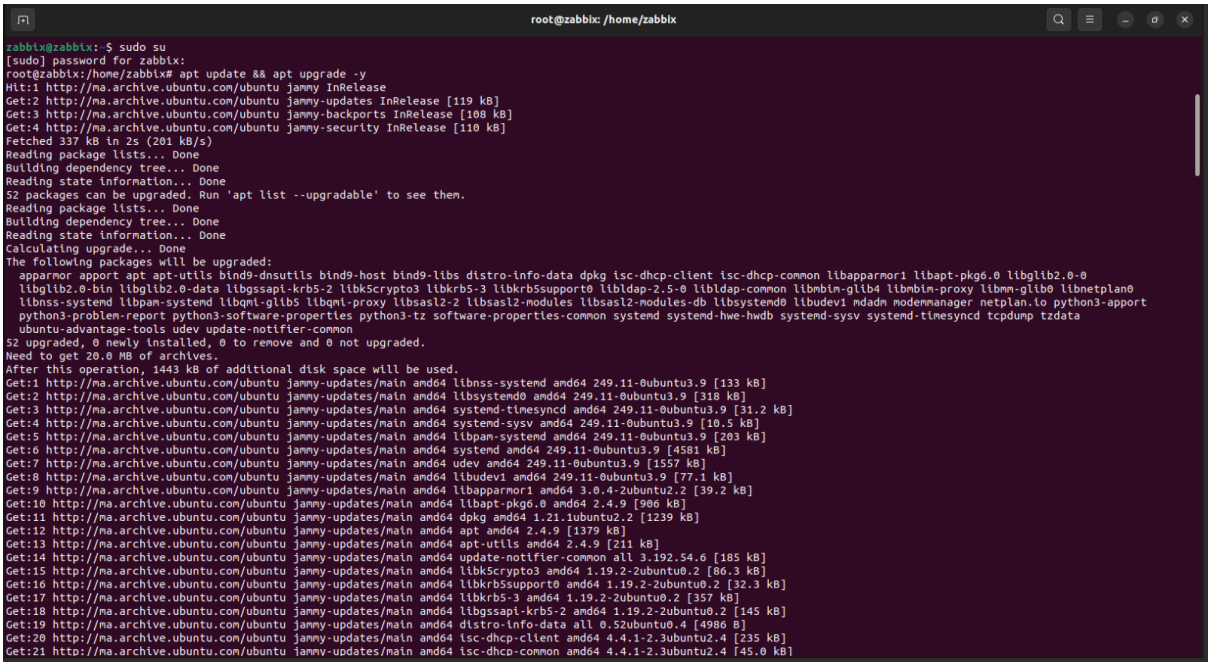


Then, we connect to the Zabbix server via SSH using the following command:

```
> ssh zabbix@192.168.59.129
```

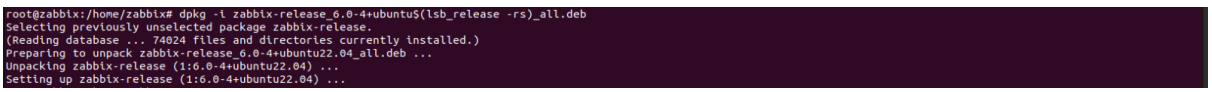


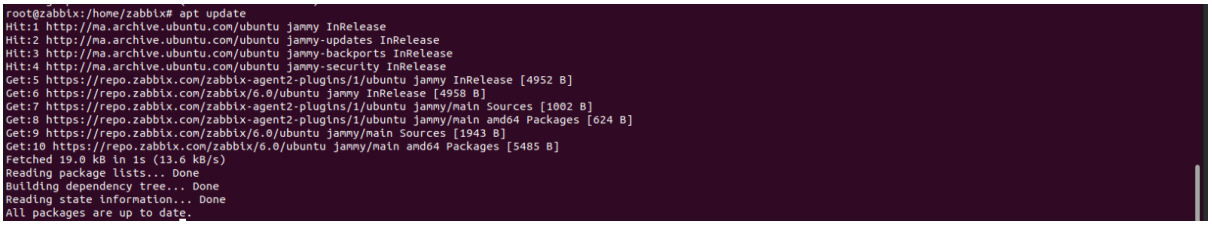
To gain superuser privileges and access elevated permissions on the Ubuntu server, the command `'sudo su'` is executed.

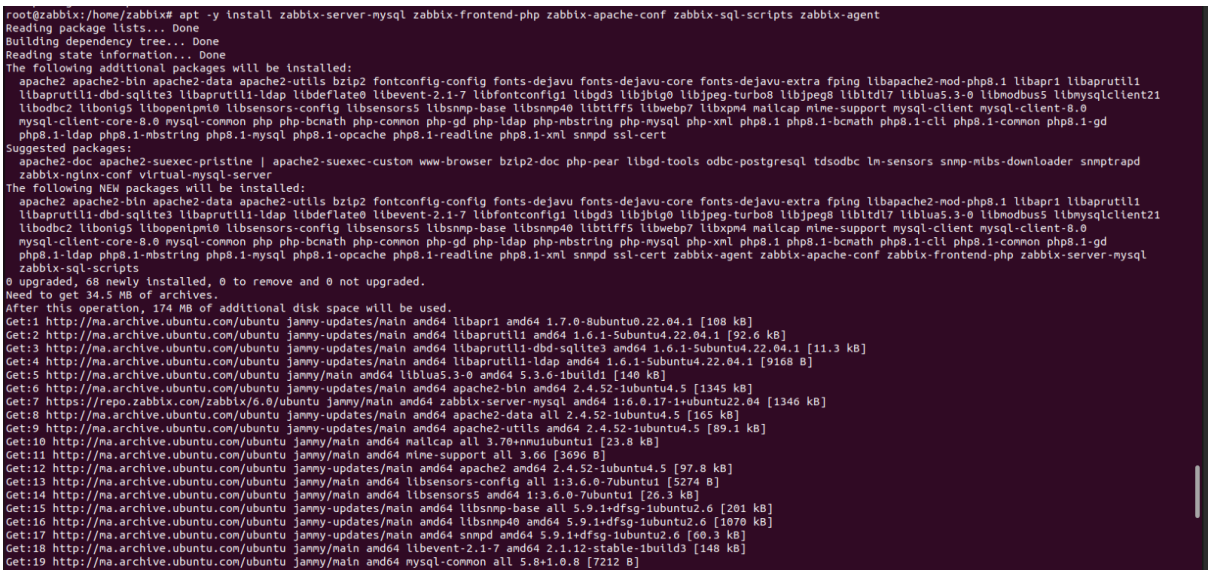


We install Zabbix 6 deb package on the Ubuntu server, using the following commands:

```
> wget
https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-re
lease/zabbix-release_6.4-1+ubuntu$(lsb_release-rs)_all.deb
> sudo dpkg -i zabbix-release_6.4-1+ubuntu$(lsb_release
-rs)_all.deb
> sudo apt update
> sudo apt -y install zabbix-server-mysql zabbix-frontend-php
zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```







Configure database

In this installation, we will use password rootDBpass as root password and zabbixDBpass as Zabbix password for DB.

Install MariaDB 10.6 :

In our terminal, we use the following commands:

```
> sudo apt install software-properties-common -y

> curl -LsS -O
https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
> sudo bash mariadb_repo_setup --mariadb-server-version=10.6

> sudo apt update

> sudo apt -y install mariadb-common mariadb-server-10.6
mariadb-client-10.6
```

```
root@zabbix:/home/zabbix# sudo apt install software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.7).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
root@zabbix:/home/zabbix# curl -LsS -O https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
sudo bash mariadb_repo_setup --mariadb-server-version=10.6
# [info] Checking for script prerequisites.
# [info] MariaDB Server version 10.6 is valid
# [info] Repository file successfully written to /etc/apt/sources.list.d/mariadb.list
# [info] Adding trusted package signing keys...
# [info] Running apt-get update...
# [info] Done adding trusted package signing keys
```

```
root@zabbix:/home/zabbix# sudo apt update
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 https://dlm.mariadb.com/repo/mariadb-server/10.6/repo/ubuntu jammy InRelease [7767 B]
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://ma.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://downloads.mariadb.com/Tools/ubuntu jammy InRelease
Get:7 https://dlm.mariadb.com/repo/maxscale/latest/apt jammy InRelease [9350 B]
Hit:8 https://repo.zabbix.com/zabbix-agent2-plugins/1/ubuntu jammy InRelease
Hit:9 https://repo.zabbix.com/zabbix/6.0/ubuntu jammy InRelease
Fetched 17.1 kB in 1s (16.8 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

```
root@zabbix:/home/zabbix# sudo apt -y install mariadb-common mariadb-server-10.6 mariadb-client-10.6
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 libcgi-fast-perl libcgi-pm-perl libclone-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3 libndctl6 libpmem1
  libtimedate-perl liburi-perl liburing2 mariadb-client-core-10.6 mariadb-server-core-10.6 socat
Suggested packages:
  libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx mariadb-test
The following packages will be REMOVED:
  mysql-client mysql-client-8.0 mysql-client-core-8.0
The following NEW packages will be installed:
  galera-4 libcgi-fast-perl libcgi-pm-perl libclone-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3 libndctl6 libpmem1
  libtimedate-perl liburi-perl liburing2 mariadb-client-10.6 mariadb-client-core-10.6 mariadb-common mariadb-server-10.6 mariadb-server-core-10.6 socat
0 upgraded, 30 newly installed, 3 to remove and 0 not upgraded.
Need to get 26.9 MB of archives.
After this operation, 131 MB of additional disk space will be used.
Get:1 https://dlm.mariadb.com/repo/mariadb-server/10.6/repo/ubuntu jammy/main amd64 mariadb-client-10.6 amd64 1:10.6.13+maria~ubu2204 [2187 kB]
Get:2 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libdbi-perl amd64 1.643-3build3 [741 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libdaxctl1 amd64 72.1-1 [19.8 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libndctl6 amd64 72.1-1 [57.7 kB]
Get:5 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libpmem1 amd64 1.11.1-3build1 [81.4 kB]
Get:6 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 liburing2 amd64 2.1-2build1 [10.3 kB]
Get:7 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 socat amd64 1.7.4.1-3ubuntu4 [349 kB]
Get:8 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-tagset-perl all 3.20-4 [12.5 kB]
Get:9 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 liburi-perl all 5.10-1 [78.8 kB]
Get:10 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-parser-perl amd64 3.76-1build2 [88.4 kB]
Get:11 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libcgi-pm-perl all 4.54-1 [188 kB]
Get:12 https://dlm.mariadb.com/repo/mariadb-server/10.6/repo/ubuntu jammy/main amd64 mariadb-common all 1:10.6.13+maria~ubu2204 [3928 B]
Get:13 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi0ldbl amd64 2.4.2-2build2 [28.0 kB]
Get:14 https://dlm.mariadb.com/repo/mariadb-server/10.6/repo/ubuntu jammy/main amd64 libmariadb3 amd64 1:10.6.13+maria~ubu2204 [172 kB]
Get:15 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-perl amd64 0.82+ds-1build1 [22.8 kB]
Get:16 https://dlm.mariadb.com/repo/mariadb-server/10.6/repo/ubuntu jammy/main amd64 mariadb-client-core-10.6 amd64 1:10.6.13+maria~ubu2204 [995 kB]
Get:17 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libcgi-fast-perl all 1:2.15-1 [10.5 kB]
Get:18 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libclone-perl amd64 0.45-1build3 [11.0 kB]
Get:19 http://ma.archive.ubuntu.com/ubuntu jammy/universe amd64 libdbd-mysql-perl amd64 4.050-5 [87.6 kB]
Get:20 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libencode-locale-perl all 1.05-1.1 [11.8 kB]
Get:21 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-bin amd64 2.4.2-2build2 [11.2 kB]
Get:22 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-template-perl all 2.97-1.1 [59.1 kB]
Get:23 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libtimedate-perl all 2.3300-2 [34.0 kB]
Get:24 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libhttp-date-perl all 6.05-1 [9920 B]
Get:25 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 libio-html-perl all 1.004-2 [15.4 kB]
Get:26 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 liblwp-mediatypes-perl all 6.04-1 [19.5 kB]
```

Once the installation is complete, we start the MariaDB service and we enable it using the following commands:

```
> systemctl start mariadb
> systemctl enable mariadb
> systemctl status mariadb
```

```
root@zabbix:/home/zabbix# systemctl start mariadb
root@zabbix:/home/zabbix# systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mariadb
root@zabbix:/home/zabbix# systemctl status mariadb
● mariadb.service - MariaDB 10.6.13 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/mariadb.service.d
             └─migrated-from-my.cnf-settings.conf
     Active: active (running) since Fri 2023-05-26 17:01:44 UTC; 1min 20s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 31389 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 4530)
     Memory: 61.0M
        CPU: 254ms
     CGroup: /system.slice/mariadb.service
             └─31389 /usr/sbin/mariadbd

May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Note] Plugin 'FEEDBACK' is disabled.
May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Note] Server socket created on IP: '127.0.0.1'.
May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Note] InnoDB: Buffer pool(s) load completed at 230526 17:01:44
May 26 17:01:44 zabbix mariadbd[31389]: 2023-05-26 17:01:44 0 [Note] /usr/sbin/mariadbd: ready for connections.
May 26 17:01:44 zabbix mariadbd[31389]: Version: '10.6.13-MariaDB-1:10.6.13+maria~ubu2204'  socket: '/run/mysqld/mysqld.sock'  port: 3306  mariadb.org binary distribution
May 26 17:01:44 zabbix systemd[1]: Started MariaDB 10.6.13 database server.
May 26 17:01:44 zabbix /etc/mysql/debian-start[31419]: Checking for insecure root accounts.
May 26 17:01:44 zabbix /etc/mysql/debian-start[31423]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
```

Reset root password for database :

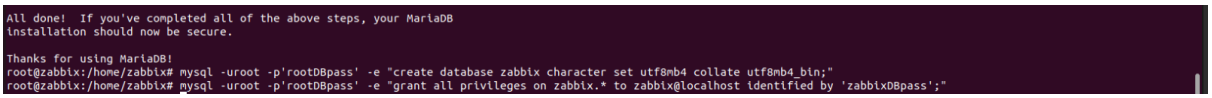
We secure MySQL/MariaDB by changing the default password for MySQL root:

```
> mysql_secure_installation

Enter current password for root (enter for none): Press Enter
Switch to unix_socket authentication [Y/n] y
Change the root password? [Y/n] y
New password: <Enter root DB password>
Re-enter new password: <Repeat root DB password>
Remove anonymous users? [Y/n]: Y
Disallow root login remotely? [Y/n]: Y
Remove test database and access to it? [Y/n]:  Y
Reload privilege tables now? [Y/n]:  Y
```

```
root@zabbix:/home/zabbix# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.
```

Create database :

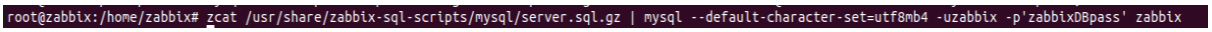
```
> mysql -uroot -p'rootDBpass' -e "create database zabbix
character set utf8mb4 collate utf8mb4_bin;"
> mysql -uroot -p'rootDBpass' -e "grant all privileges on
zabbix.* to zabbix@localhost identified by 'zabbixDBpass';"
```



Import initial schema and data

We import database shema for Zabbix server (could last up to 5 minutes):

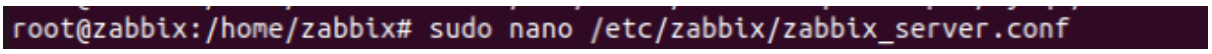
```
> zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz |
mysql --default-character-set=utf8mb4 -uzabbix
-p'zabbixDBpass'zabbix
```



Enter database password in Zabbix configuration file :

We open zabbix_server.conf file with command:

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



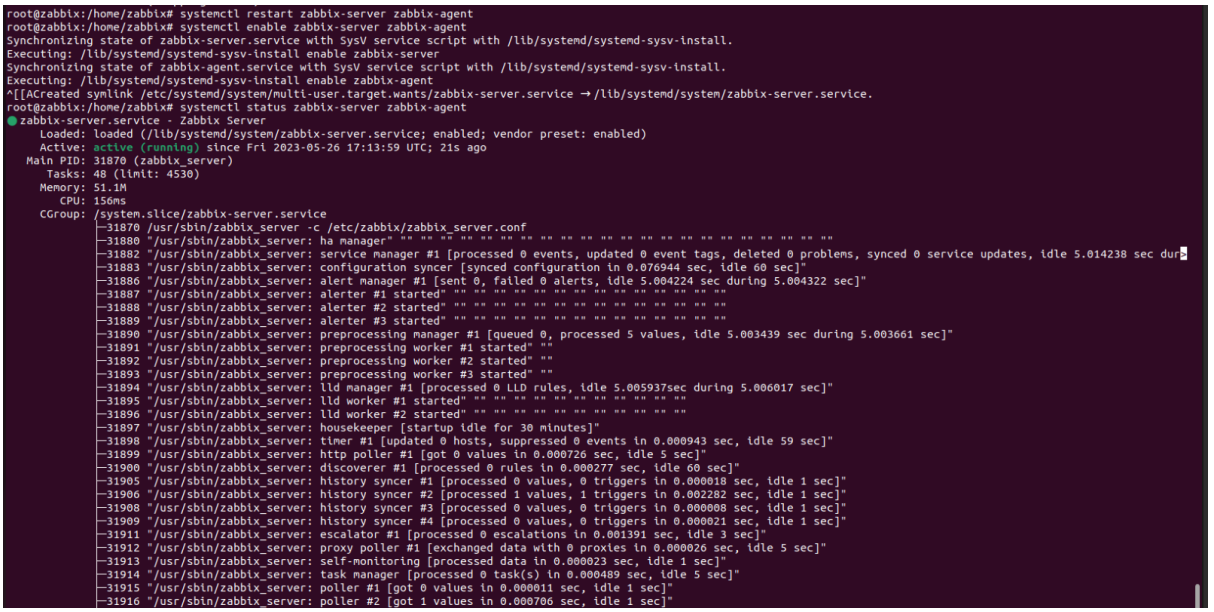
and we add database password in this format anywhere in file:

DBPassword=zabbixDBpass

We save and exit file (ctrl+x, followed by y and enter).

Start Zabbix server and agent processes

```
> sudo systemctl restart zabbix-server zabbix-agent
> sudo systemctl enable zabbix-server zabbix-agent
> sudo systemctl status zabbix-server zabbix-agent
```



Configure Zabbix frontend

Configure PHP for Zabbix frontend :

We edit file /etc/zabbix/apache.conf:

```
> sudo nano /etc/zabbix/apache.conf
```

We uncomment “# php_value date.timezone Europe/Riga” by removing symbol # and we set the right timezone for our country:

php_value date.timezone Africa/Casablanca

We save and exit file (ctrl+x, followed by y and enter)

```
 GNU nano 6.2                                        /etc/zabbix/apache.conf *
# Define /zabbix alias, this is the default
<IfModule mod_alias.c>
    Alias /zabbix /usr/share/zabbix
</IfModule>

<Directory "/usr/share/zabbix">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all

    <IfModule mod_php.c>
        php_value max_execution_time 300
        php_value memory_limit 128M
        php_value post_max_size 16M
        php_value upload_max_filesize 2M
        php_value max_input_time 300
        php_value max_input_vars 10000
        php_value always_populate_raw_post_data -1
    </IfModule>

    <IfModule mod_php7.c>
        php_value max_execution_time 300
        php_value memory_limit 128M
        php_value post_max_size 16M
        php_value upload_max_filesize 2M
        php_value max_input_time 300
        php_value max_input_vars 10000
        php_value always_populate_raw_post_data -1
        php_value date.timezone Africa/Casablanca
    </IfModule>
</Directory>

<Directory "/usr/share/zabbix/conf">
    Order deny,allow
    Deny from all
    <files *.php>
        Order deny,allow
        Deny from all
    </files>
</Directory>

File Name to Write: /etc/zabbix/apache.conf
^G Help                    M-D DOS Format             M-A Append                 M-B Backup File
```

Restart Apache web server and make it start at system boot :

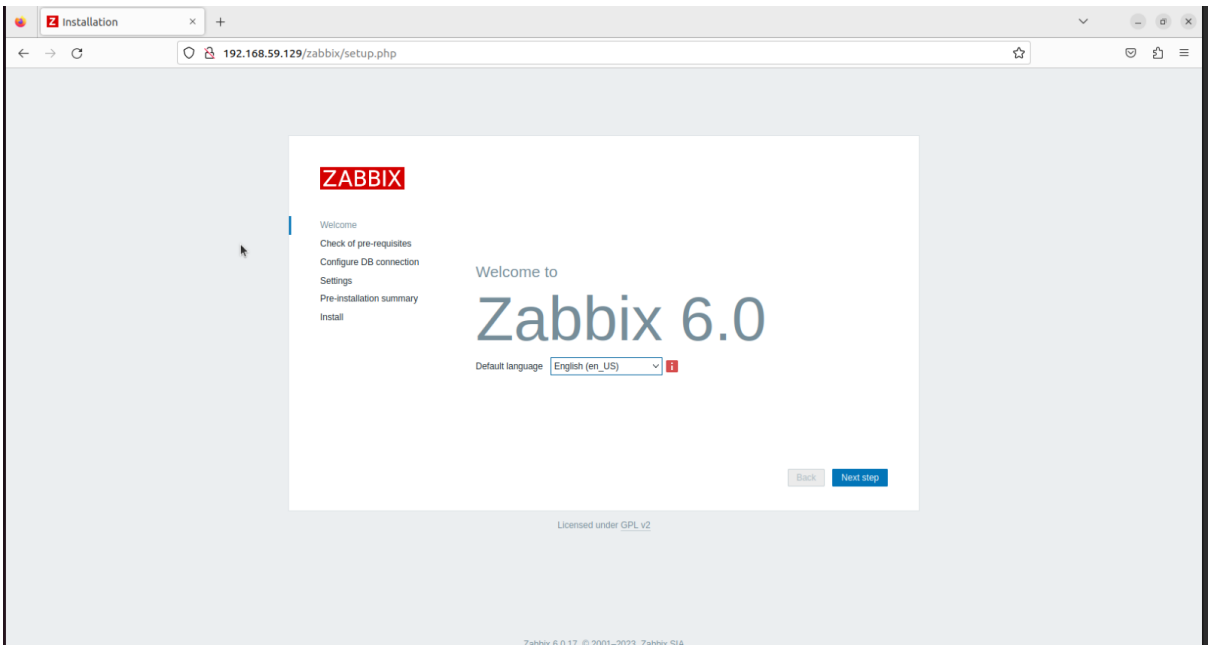
```
> sudo systemctl restart apache2
> sudo systemctl enable apache2
>sudo systemctl status apache2
```

```
root@zabbix:/home/zabbix# systemctl restart apache2
root@zabbix:/home/zabbix# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@zabbix:/home/zabbix# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-05-26 17:17:25 UTC; 17s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 32165 (apache2)
      Tasks: 6 (limit: 4530)
     Memory: 12.3M
        CPU: 32ms
     CGroup: /system.slice/apache2.service
             ├─32165 /usr/sbin/apache2 -k start
             ├─32167 /usr/sbin/apache2 -k start
             ├─32168 /usr/sbin/apache2 -k start
             ├─32169 /usr/sbin/apache2 -k start
             ├─32170 /usr/sbin/apache2 -k start
             └─32171 /usr/sbin/apache2 -k start

May 26 17:17:25 zabbix systemd[1]: Starting The Apache HTTP Server...
May 26 17:17:25 zabbix apachectl[32164]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directiv
May 26 17:17:25 zabbix systemd[1]: Started The Apache HTTP Server.
```

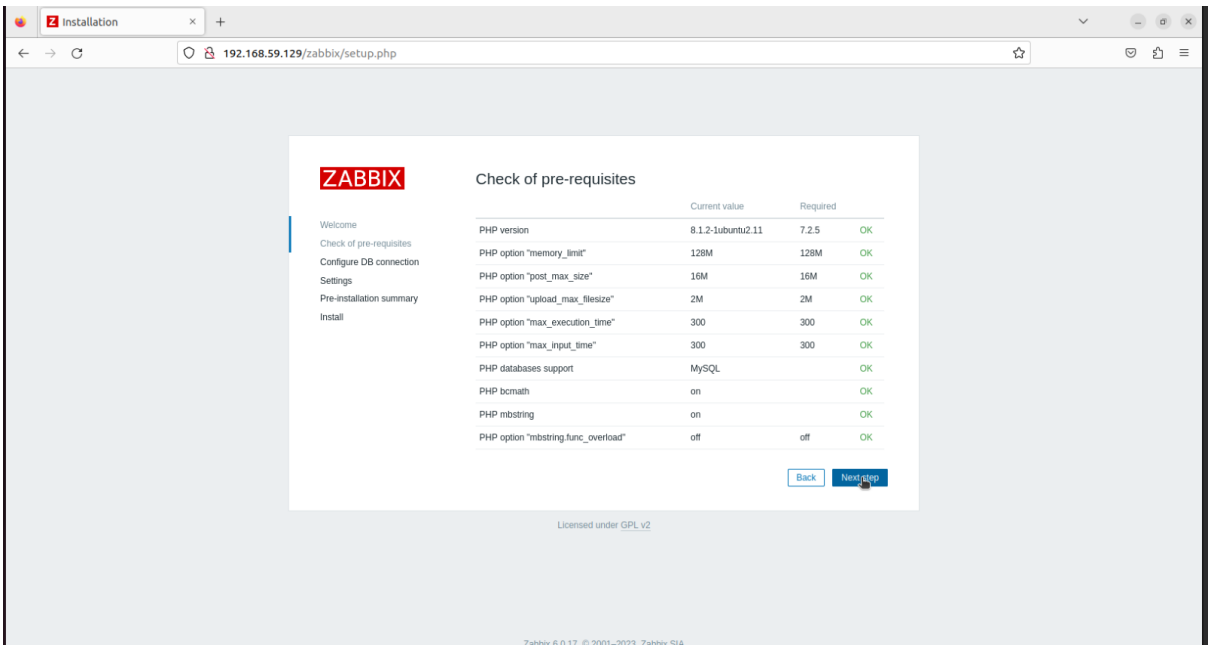
Configure web frontend :

We connect to our newly installed Zabbix frontend using URL “http://192.168.59.129/zabbix” to initiate the Zabbix installation wizard.

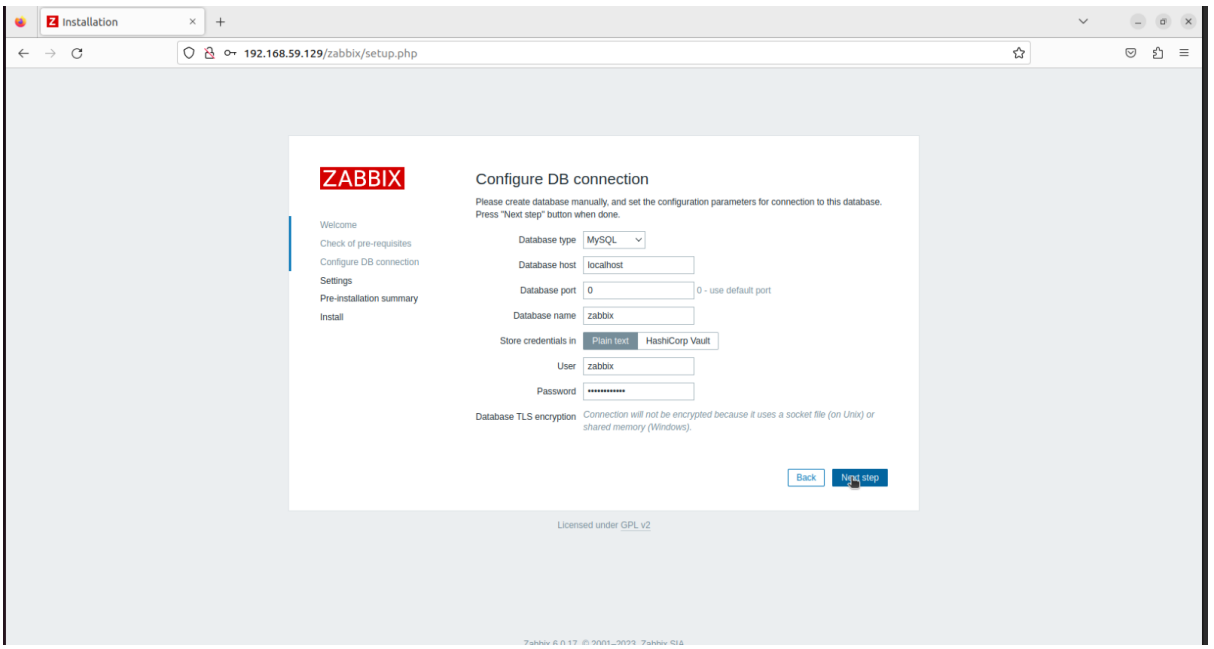
Basically, in this wizard we only need to enter a password for Zabbix DB user and we just click “Next step” for everything else.



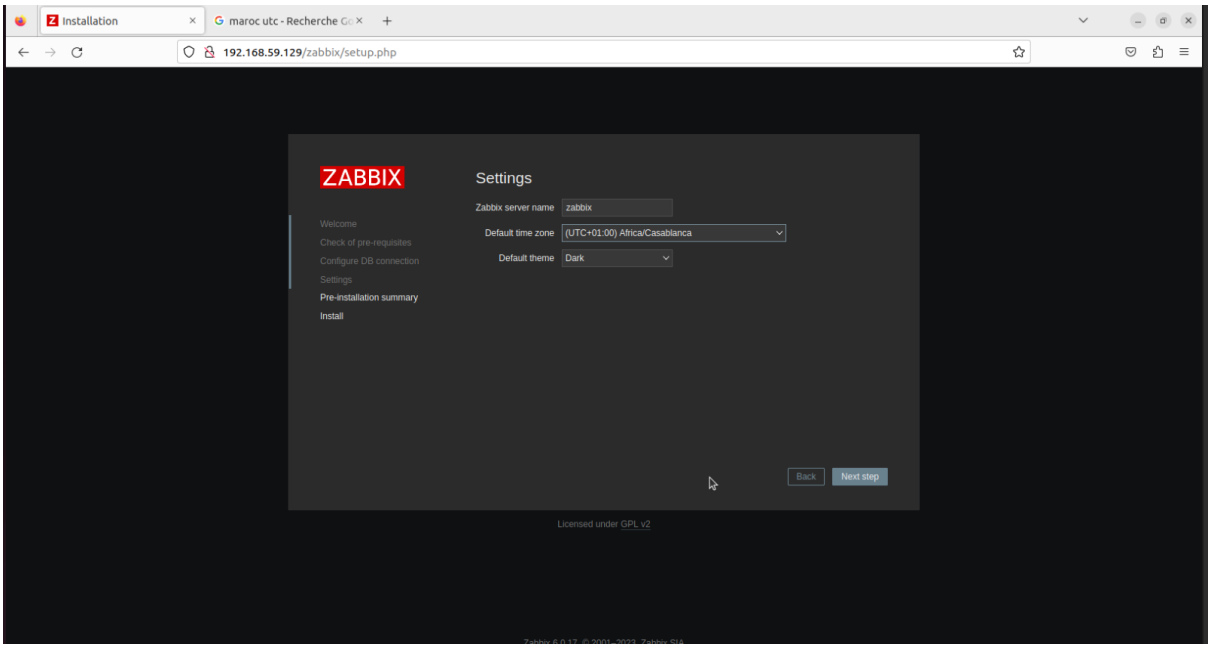
1. Installation step: Welcome screen



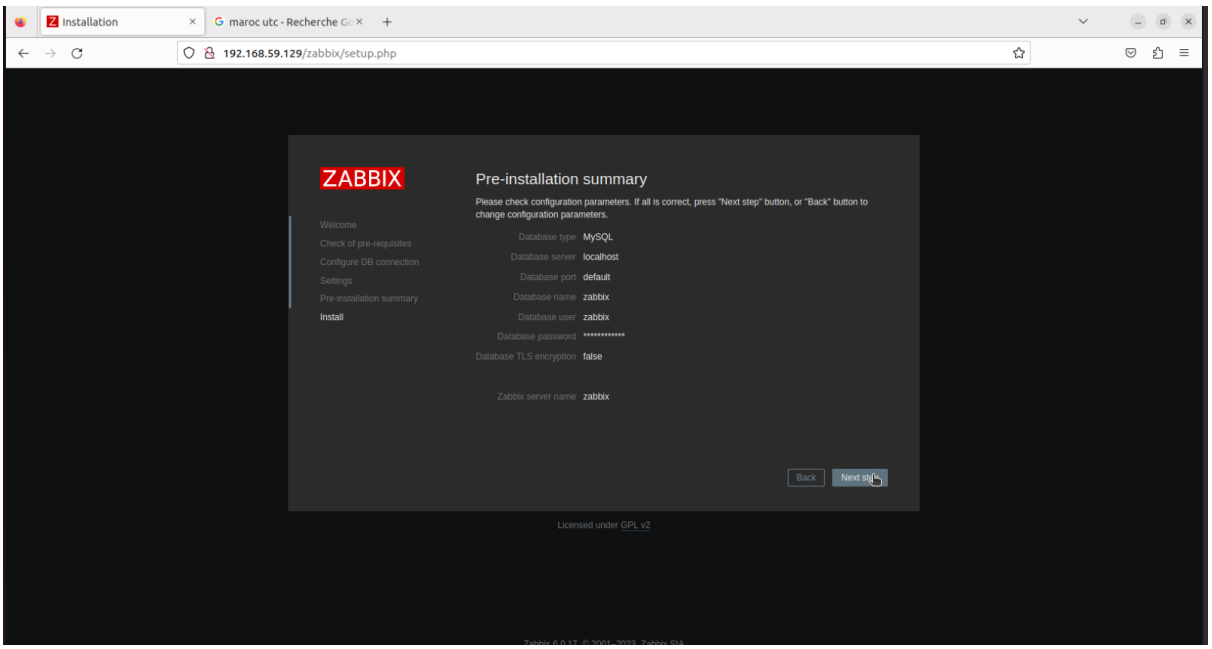
2. Installation step: Prerequisites check



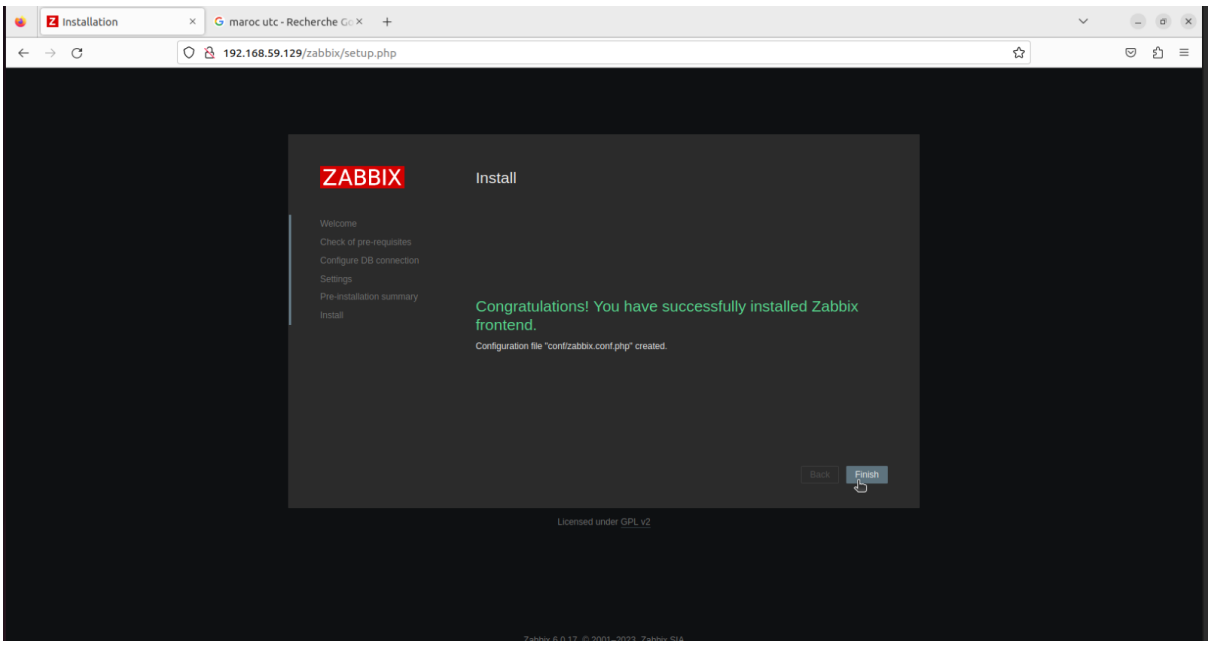
3. Installation step: Configure DB connection



4. Installation step: Configure Zabbix server



5. Installation step: Pre-installation summary

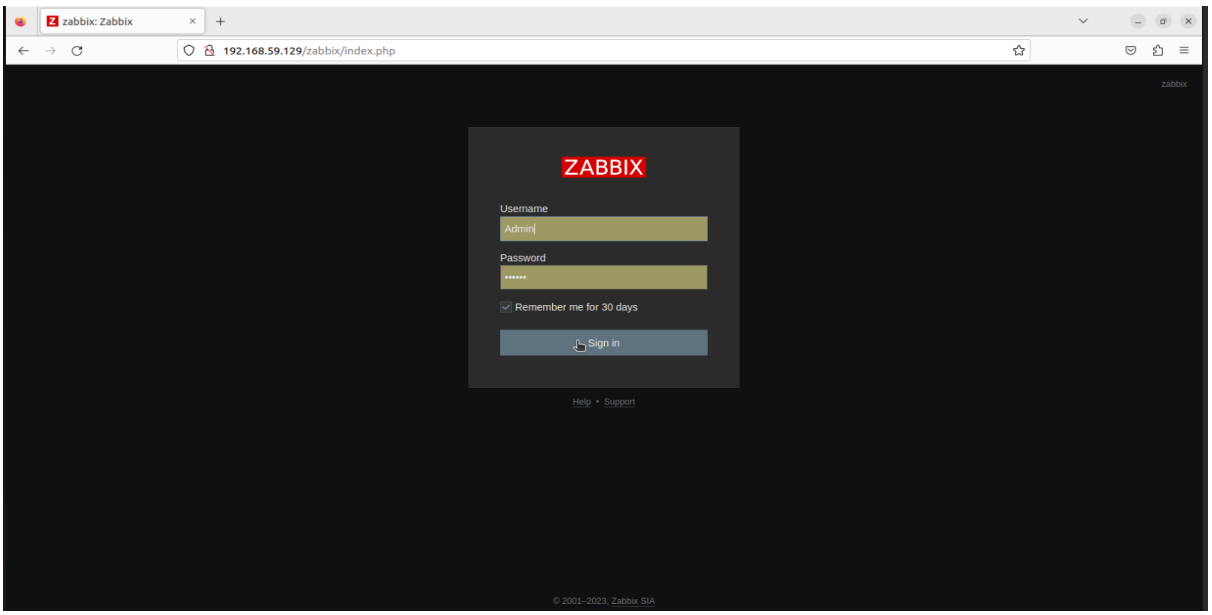


6. Installation step: Finish

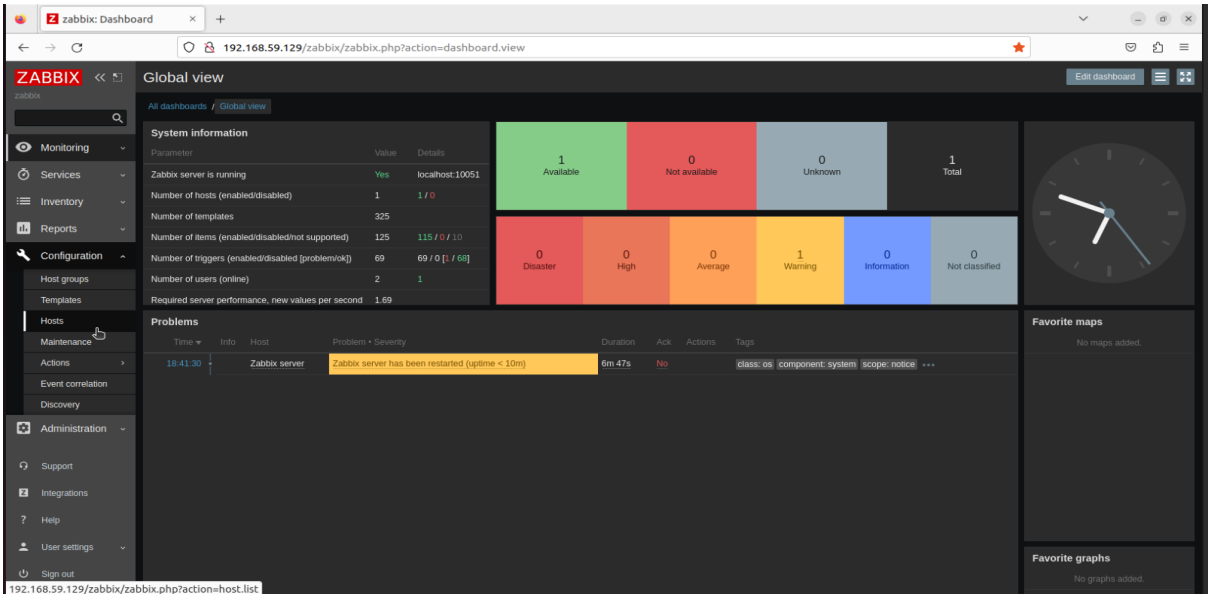
That’s it, we have installed the Zabbix monitoring system!

Login to frontend using Zabbix default login credentials

We use Zabbix default admin username “Admin” and password “zabbix” (without quotes) to login to Zabbix frontend at URL “http://192.168.59.129/zabbix” via our browser.



ZABBIX LOGIN PAGE



Zabbix dashboard

The Zabbix dashboard:

In the realm of IT infrastructure monitoring and management, having a comprehensive and intuitive dashboard is paramount to staying on top of network health and performance. Zabbix offers a robust dashboard feature that enables organizations to visualize critical metrics, monitor network elements, and streamline operations. The capabilities and benefits of Zabbix dashboards, shedding light on how they empower administrators to efficiently monitor their networks and optimize performance.

- An Overview of Zabbix Dashboards:

Zabbix dashboards serve as a centralized hub for monitoring key performance indicators (KPIs), system events, and network components. These dashboards provide administrators with a bird's-eye view of their infrastructure, allowing them to detect issues, track performance trends, and make data-driven decisions. Zabbix's customizable dashboard interface enables users to tailor the display of data to suit their specific monitoring needs.

- Real-Time Monitoring and Alerting:

One of the primary strengths of Zabbix dashboards lies in their real-time monitoring capabilities. Administrators can configure widgets and visual elements to display live data streams, such as graphs, charts, maps, and status indicators. This real-time visibility empowers IT teams to proactively identify and address anomalies or performance bottlenecks before they escalate. Zabbix's alerting system complements the dashboards by providing instant notifications via various channels, including email, SMS, or integrations with popular collaboration tools.

- Customization and Personalization:

Zabbix dashboards offer extensive customization options, allowing administrators to create tailored views that align with their monitoring goals and requirements. Users can select from a rich library of pre-built widgets or design custom ones. These widgets can be arranged and organized within the dashboard to provide a cohesive and intuitive monitoring experience. With the ability to resize, reposition, and adjust

the granularity of displayed data, Zabbix ensures that users have full control over their monitoring interface.

- Drilling Down for Granular Insights:

In complex network environments, it is essential to have the ability to drill down into specific elements to gain deeper insights. Zabbix dashboards support multi-level drill-down, enabling administrators to navigate from high-level overviews to granular details of individual hosts, services, or applications. This functionality facilitates efficient troubleshooting and root cause analysis, saving valuable time and resources.

- Historical Data Analysis:

Zabbix dashboards are not limited to real-time monitoring but also provide access to historical data. The system captures and stores metrics over time, allowing administrators to analyze trends, compare performance across different time periods, and identify long-term patterns or recurring issues. Historical data analysis plays a vital role in capacity planning, performance optimization, and informed decision-making.

- Collaboration and Reporting:

Collaboration among IT teams is essential for effective network management. Zabbix dashboards enable seamless collaboration by providing shared views of monitoring data. Multiple stakeholders can access and interact with the dashboard, fostering collaboration, knowledge sharing, and cross-functional problem-solving. Additionally, Zabbix offers reporting capabilities, allowing administrators to generate detailed reports based on dashboard data, providing stakeholders with comprehensive insights into network performance and trends.

Zabbix dashboards are an asset in the arsenal of network administrators and IT teams. With real-time monitoring, customizable visualizations, multi-level drill-down, historical data analysis, and collaboration features, Zabbix dashboards empower

organizations to streamline operations, optimize network performance, and ensure the health and reliability of their IT infrastructure. By leveraging the power of Zabbix dashboards, administrators can efficiently monitor, troubleshoot, and make informed decisions, ultimately enhancing their network management capabilities.

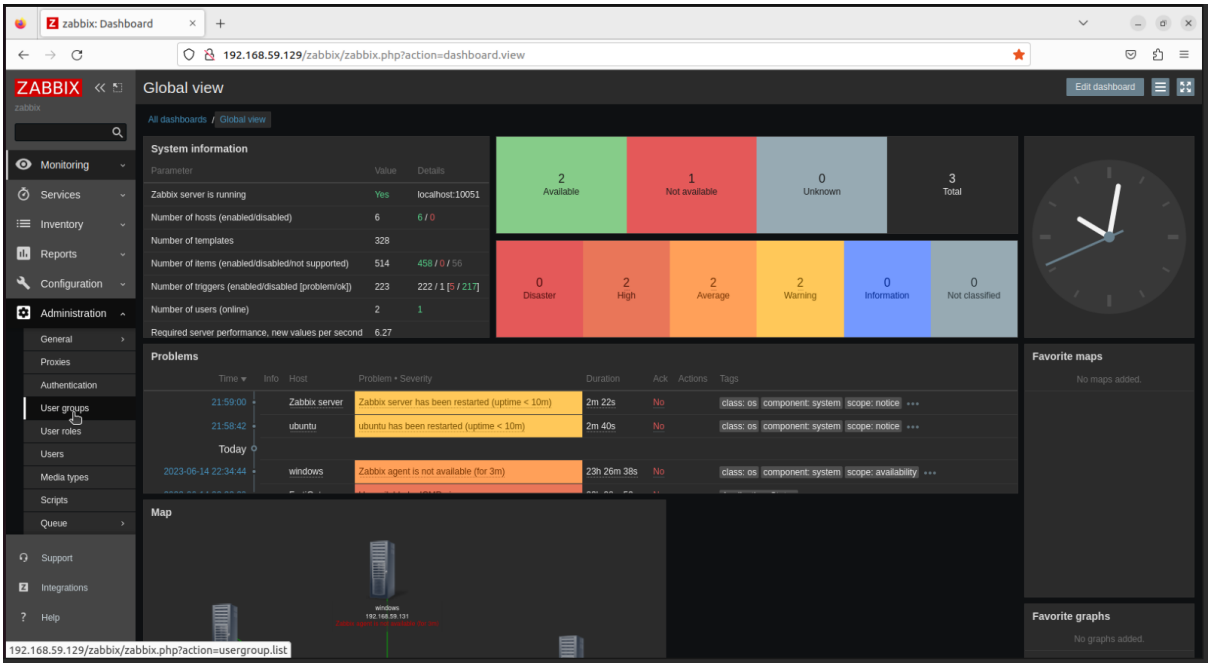
Creating and configuring Zabbix users, user groups, and user roles

To make the most of Zabbix's features, it's important to create users, user groups, and permissions. This will allow us to control who can access and modify different parts of the monitoring system, ensuring that our data remains secure and that only authorized personnel can make changes. In this report, we'll go through step-by-step instructions on how to create users, user groups, and permissions.

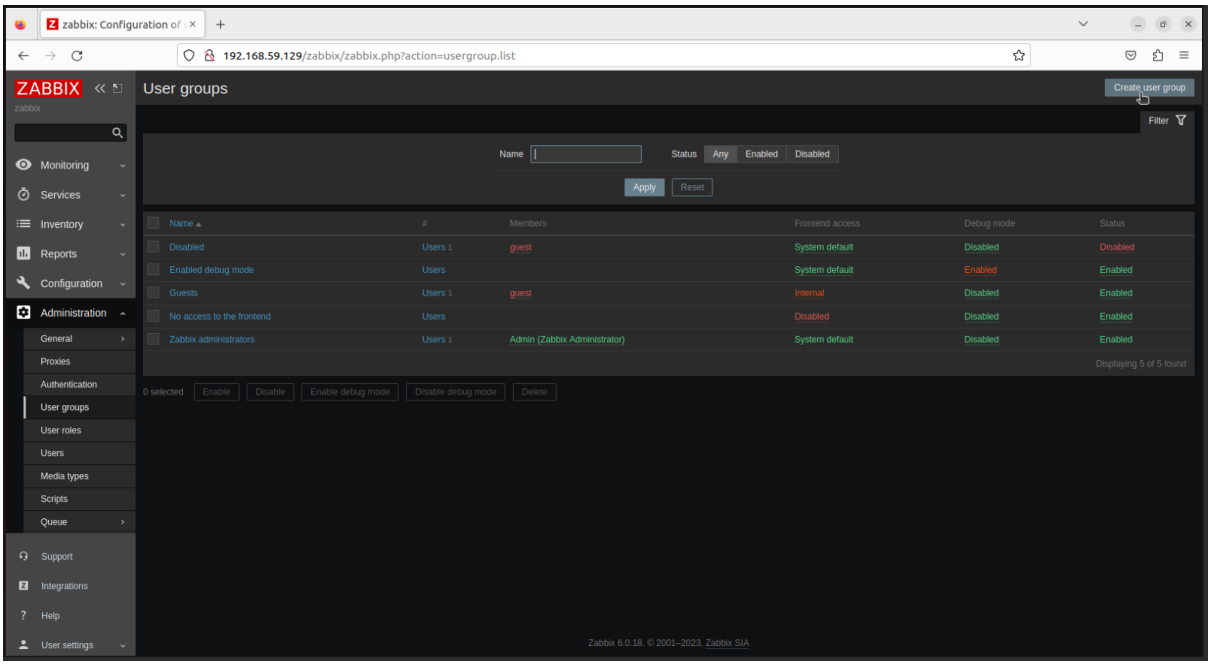
Creating User Groups

User groups in Zabbix serve to manage and organize users who share similar responsibilities within the monitoring system. User groups allow administrators to assign permissions to a group of users, rather than having to assign permissions to each user individually. This helps to simplify the management of user access and makes it easier to ensure that all users with similar responsibilities have the appropriate level of access.

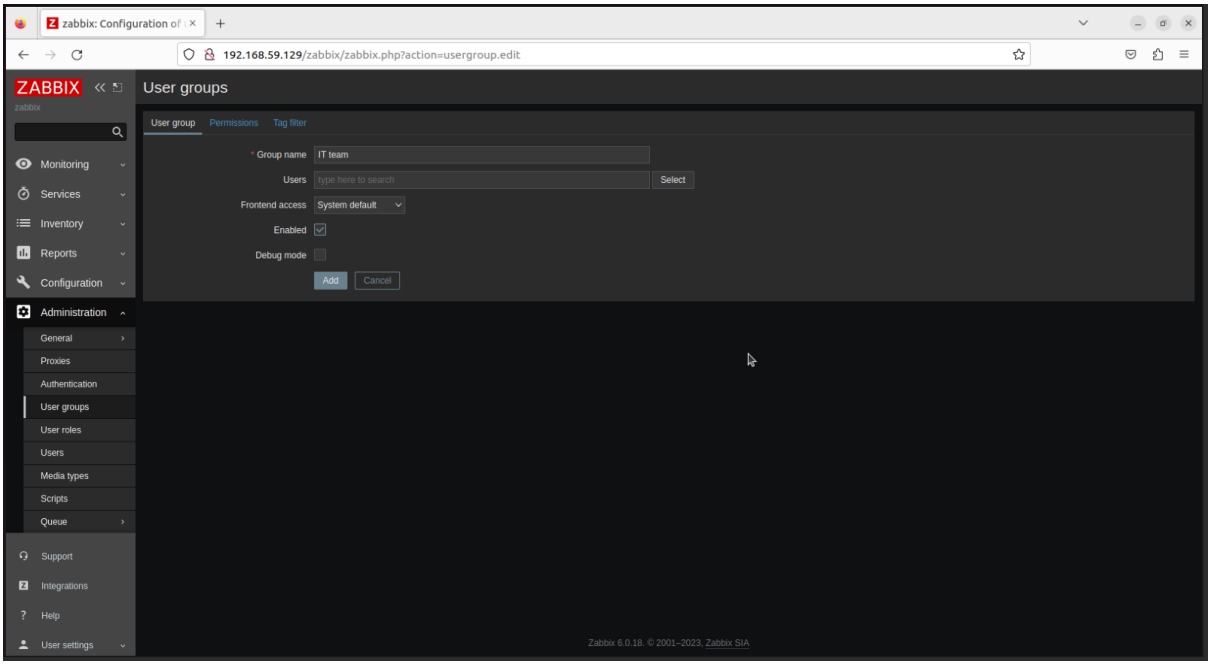
Go to the Administration tab in the navigation menu, and then click on User groups.



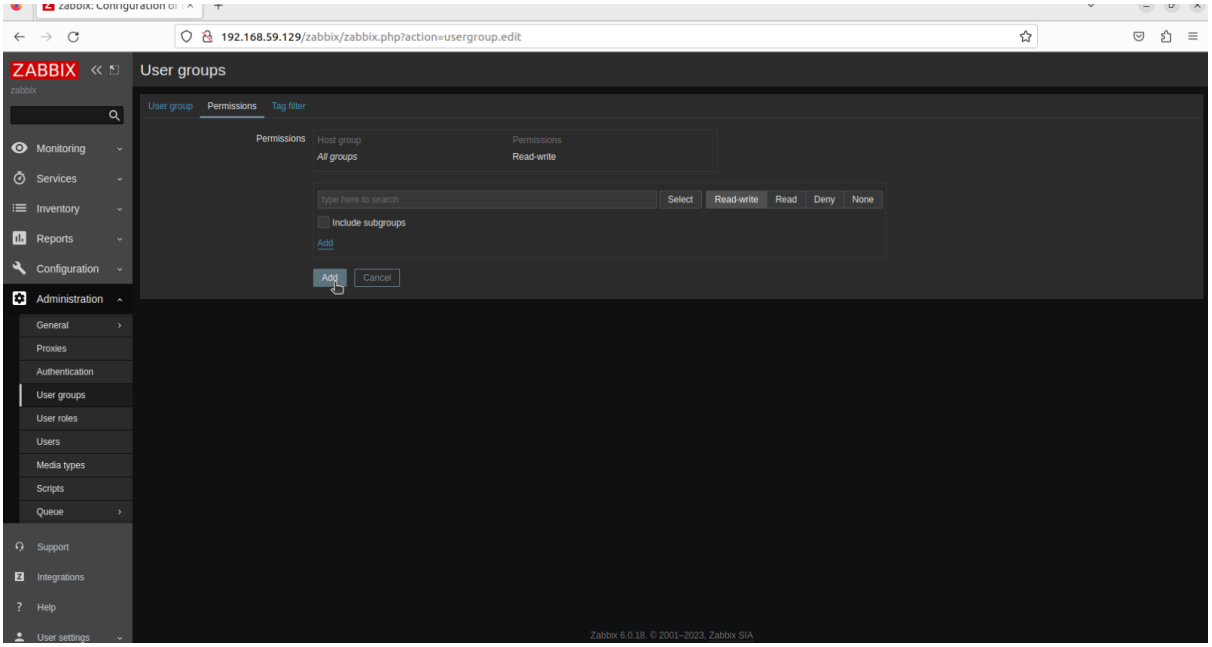
Click on the Create user group button, or the group we need to change.



In the User group details section, fill in the group name.



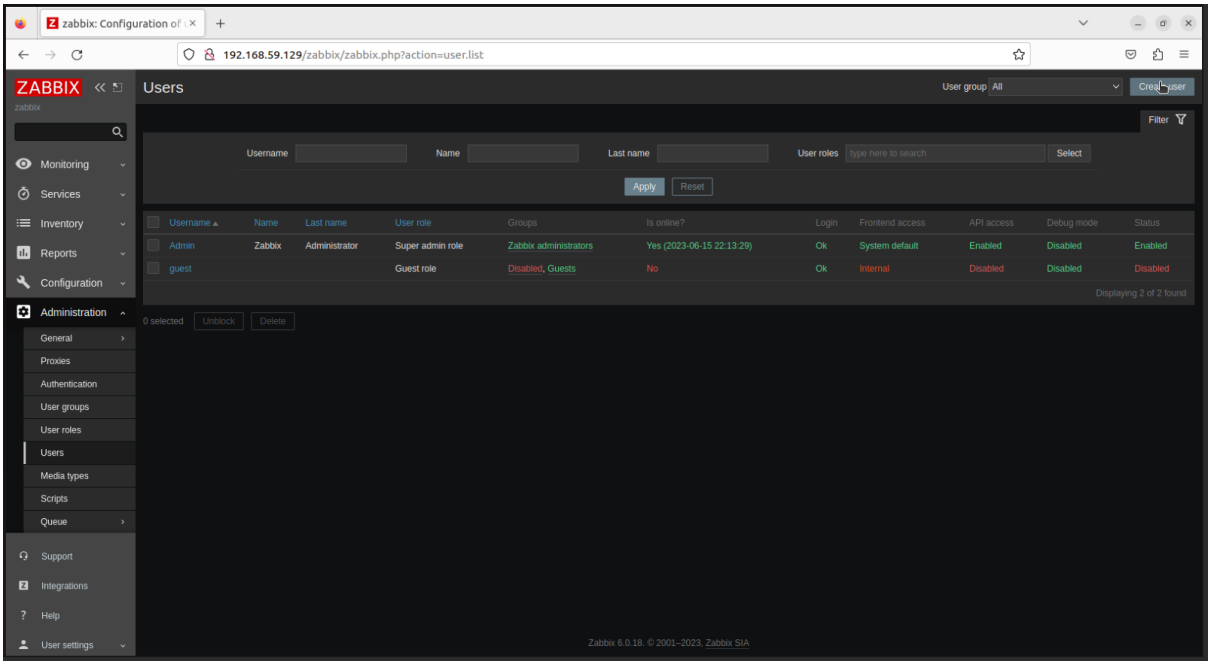
In the Permissions section, we can specify which host groups and templates, and other objects linked to them the user group can access and read and/or write.



Creating Users

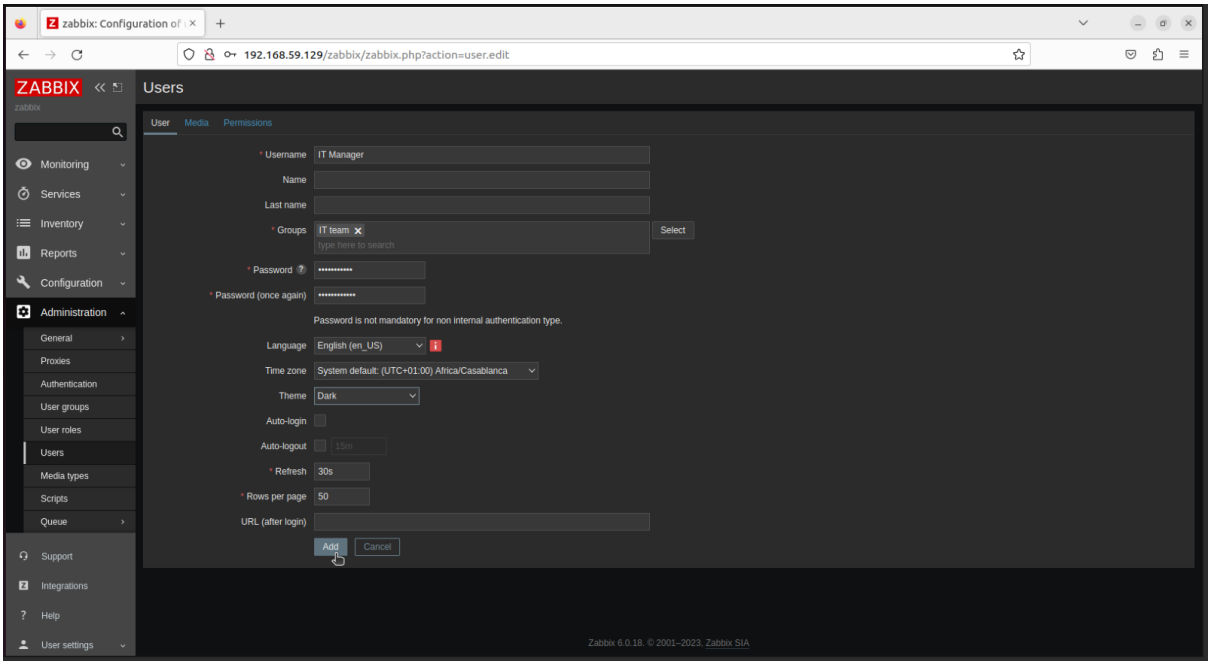
Click on the Administration tab in the top navigation menu, and then click on Users.

Click on the Create user button, or a user we want to edit.

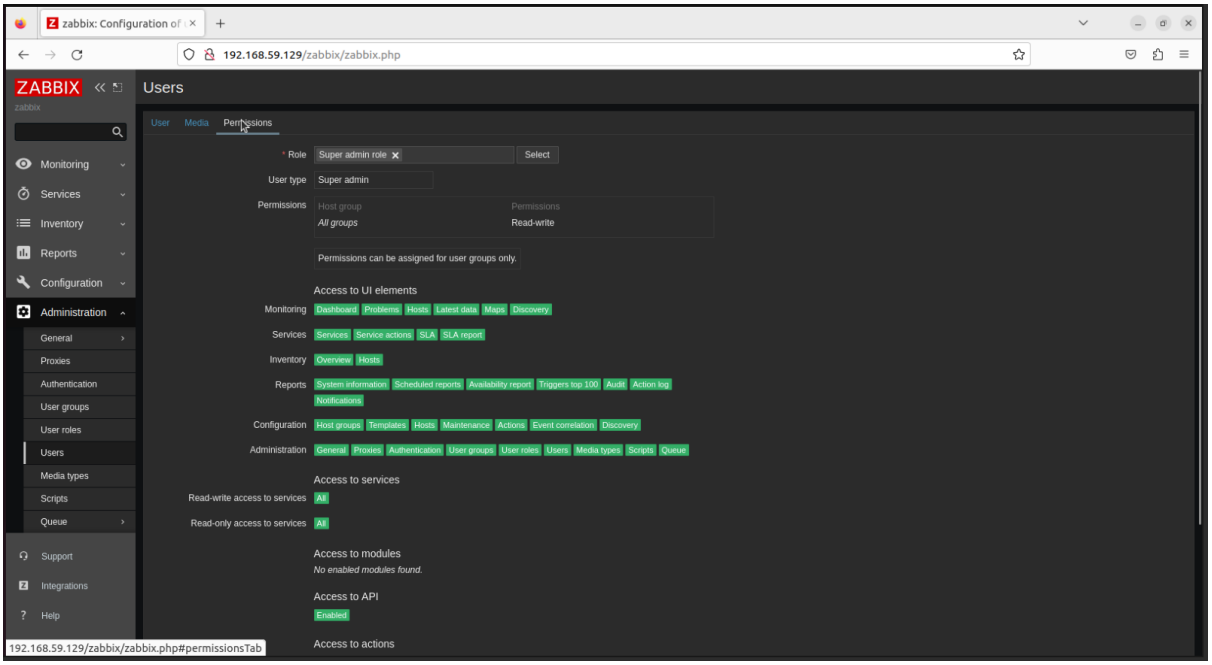


In the User details section, fill in the username and password fields.

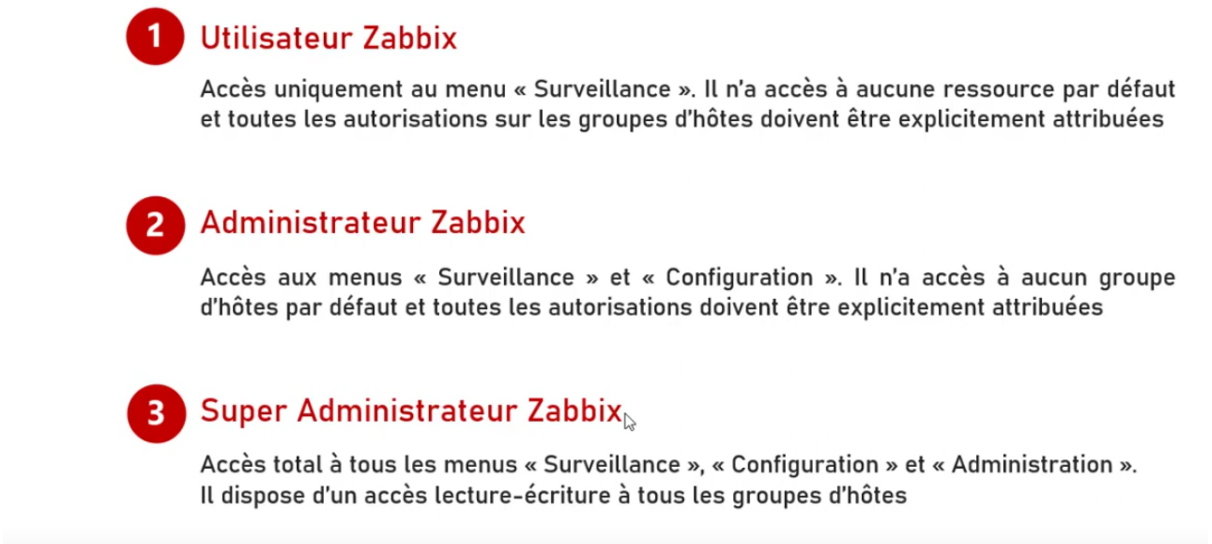
In the User group section, select the user group to which the user should belong. If the user group does not exist, we can create it.



In the Permissions section, we should choose a pre-created role for our user which specifies what UI elements are accessible for this user. It also sets permissions for specific actions, such as acknowledging problems or deleting events.



Here's a simple explanation how to choose Role:

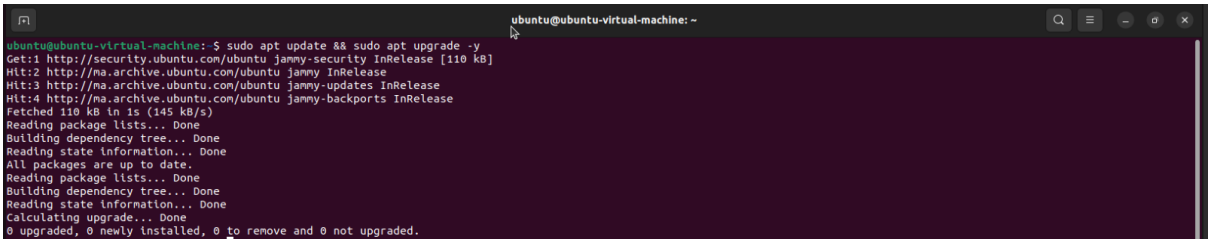


Monitoring Ubuntu machine with Zabbix using agent

Updating our System

Before starting the installation process, it is always recommended to update our system. Open a terminal and execute the following command:

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



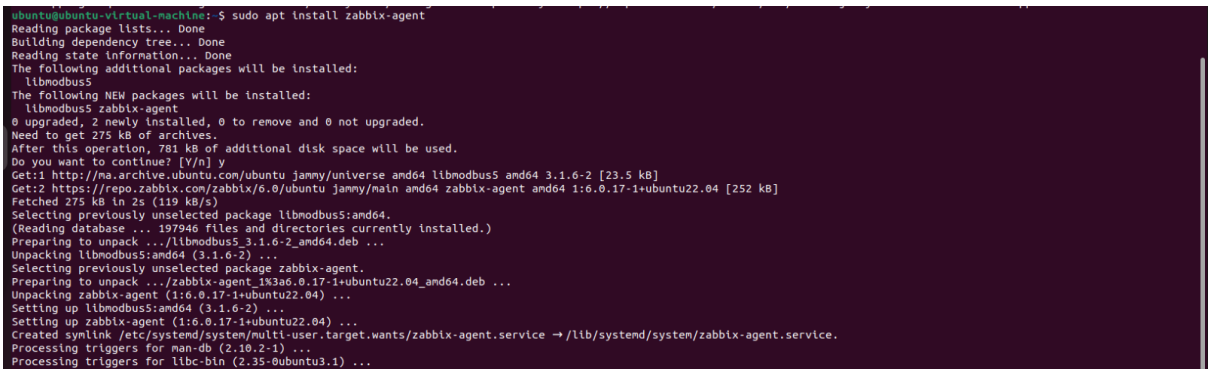
Installing Zabbix Agent

Now, we will install the Zabbix Agent from the official Zabbix repository. First, add the Zabbix repository to our system:

```
>wget
https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release
/zabbix-release_6.0-4+ubuntu22.04_all.deb
> sudo dpkg -i zabbix-release_6.0-4+ubuntu22.04_all.deb
> sudo apt update
```

Next, install the Zabbix Agent:

```
> sudo apt install zabbix-agent
```



Configuring Zabbix Agent

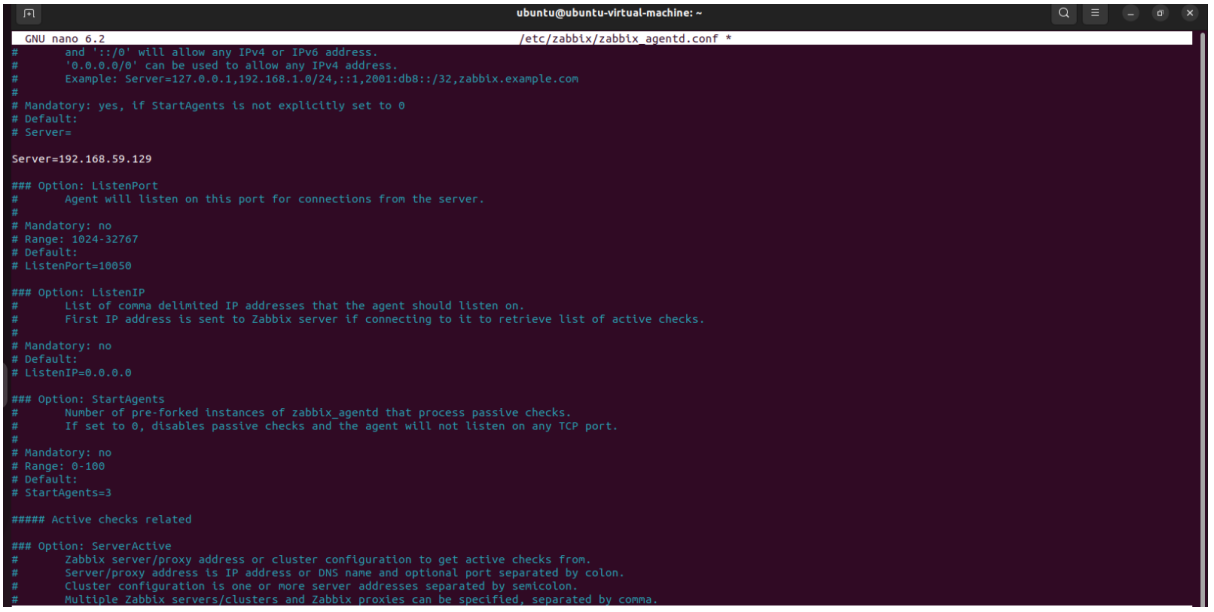
After the installation is complete, we configure the Zabbix Agent to communicate with the Zabbix server. We edit the Zabbix Agent configuration file using nano:

```
> sudo nano /etc/zabbix/zabbix_agentd.conf
```

We find the following lines and we replace them with the correct information:

- Server=<Zabbix_Server_IP>
- ServerActive=<Zabbix_Server_IP>
- Hostname=<Hostname_Of_Ubuntu_Client>

We replace <Zabbix_Server_IP> with the IP address of our Zabbix server, and <Hostname_Of_Ubuntu_Client> with the hostname of our Ubuntu client. We save the changes and close the editor.



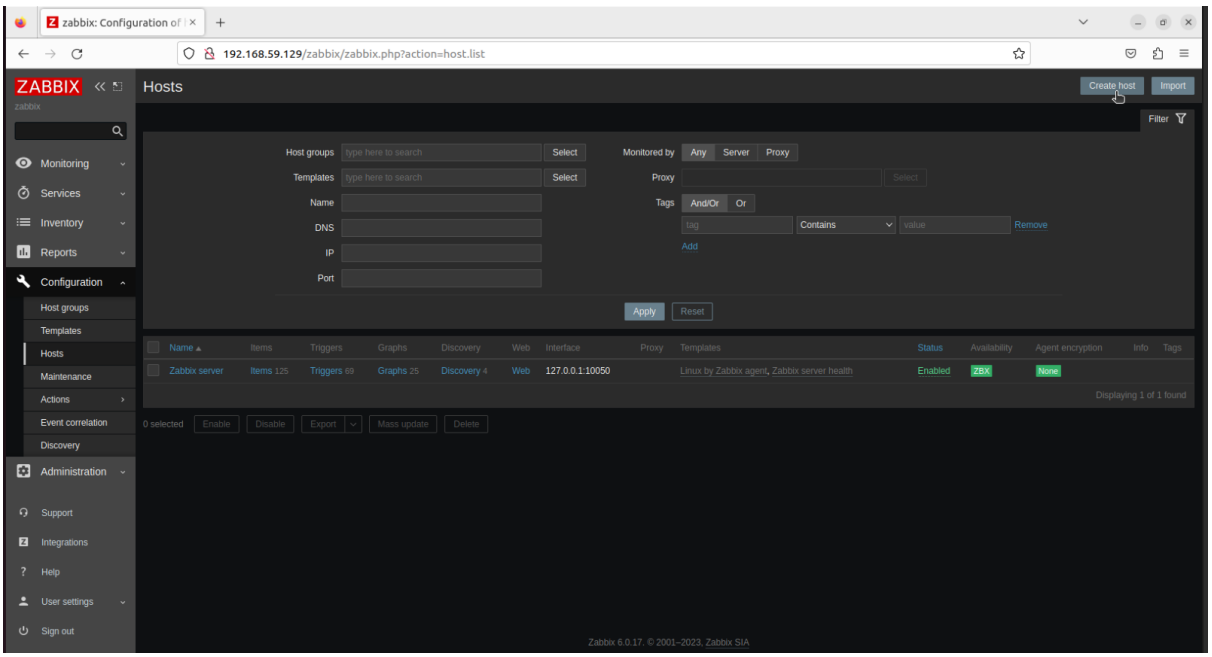
Starting and Enabling Zabbix Agent Service

Now that the configuration is done, we start the Zabbix Agent service and enable it to start automatically at boot, we check the status of the Zabbix Agent service with the following commands:

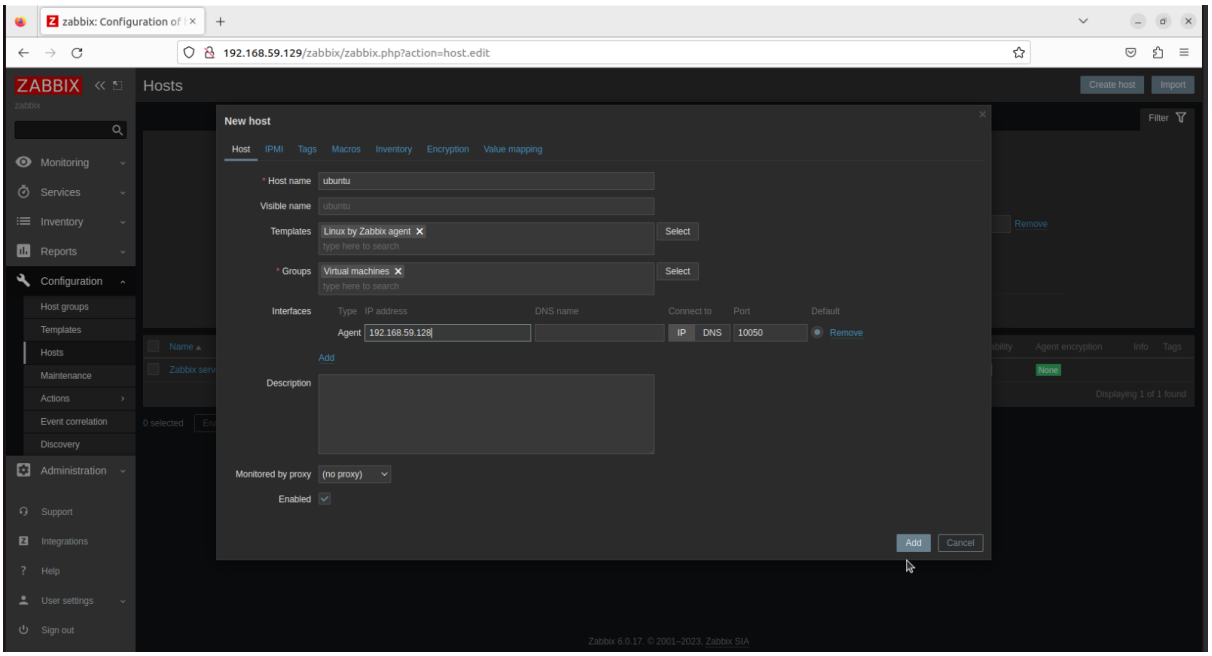
```
> sudo systemctl start zabbix-agent
> sudo systemctl enable zabbix-agent
> sudo systemctl status zabbix-agent
```

Adding the Zabbix Client to the Zabbix Server

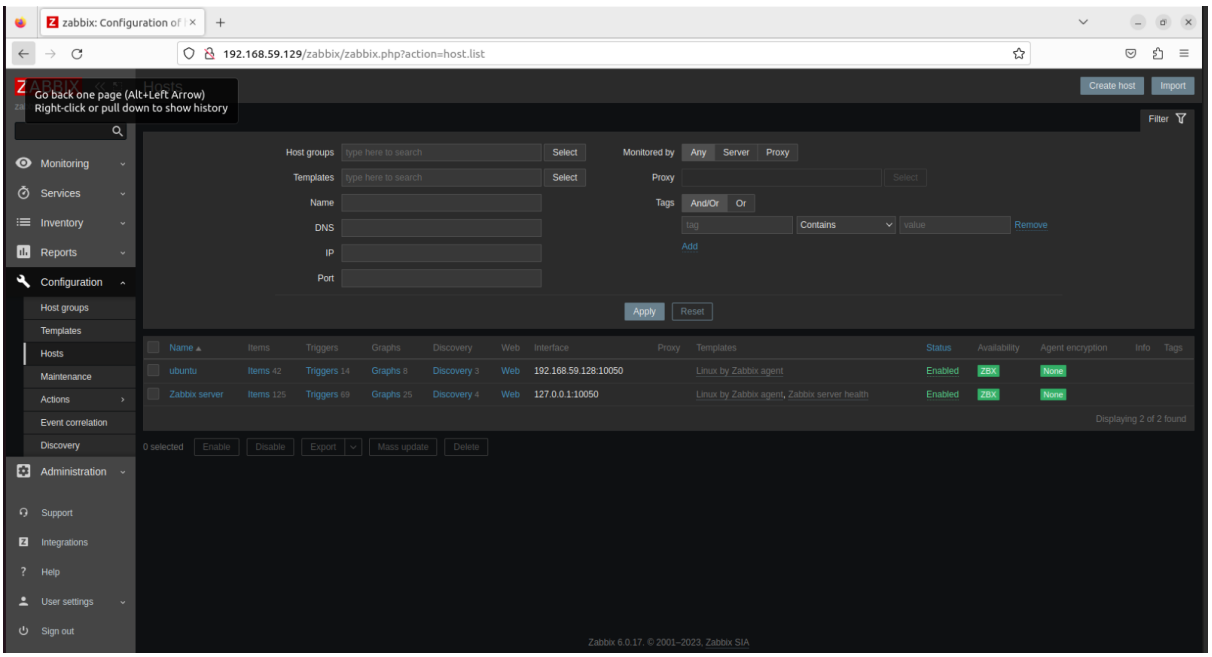
Lastly, we need to add the newly installed Zabbix Agent to the Zabbix server. Log in to the Zabbix web interface, navigate to “Configuration” > “Hosts” > “Create Host”, and fill in the necessary details.



Adding Zabbix Host to Server



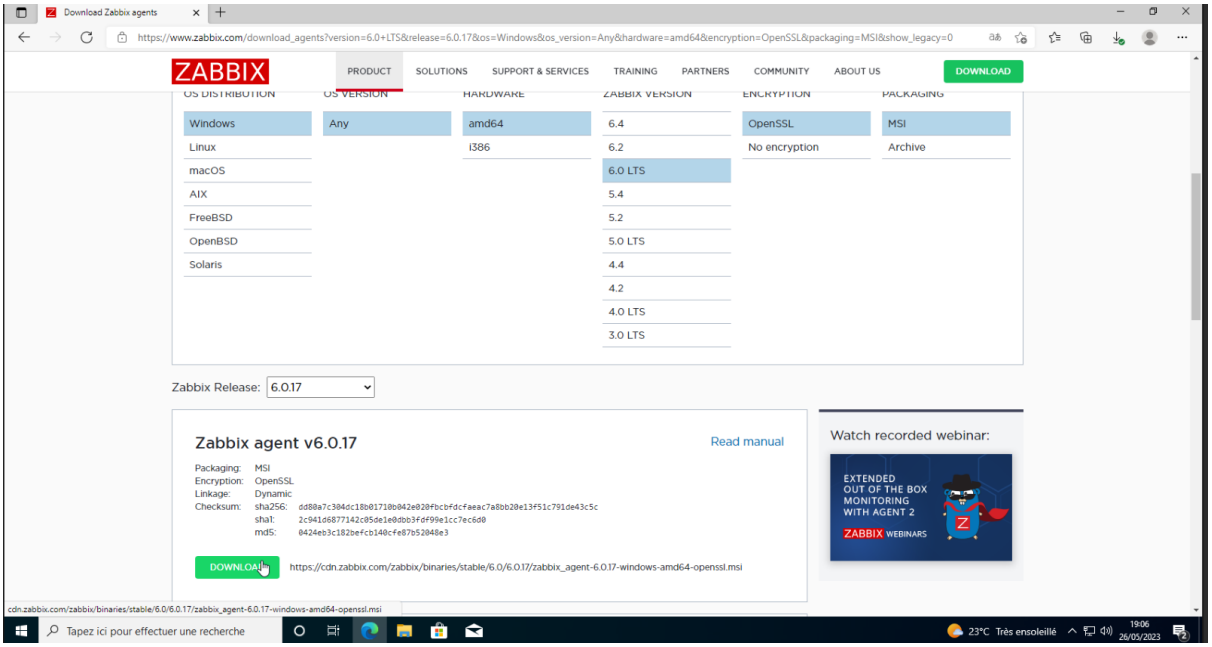
After adding the host, we should see the host status as “Enabled”. Within a few minutes, the Zabbix server will start receiving data from the Zabbix Agent installed on our Ubuntu 22.04 system


zabbix: Configuration of
192.168.59.129/zabbix/zabbix.php?action=host.list
Go back one page (Alt+Left Arrow)
Right-click or pull down to show history
Hosts
Create host
Import
Filter
Monitoring
Services
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Administration
Support
Integrations
Help
User settings
Sign out
Host groups
type here to search
Select
Templates
type here to search
Select
Name
DNS
IP
Port
Monitored by
Any
Server
Proxy
Proxy
Select
Tags
And/Or
Or
tag
Contains
value
Remove
Add
Apply
Reset
Name
Items
Triggers
Graphs
Discovery
Web
Interface
Proxy
Templates
Status
Availability
Agent encryption
Info
Tags
ubuntu
Items 42
Triggers 14
Graphs 8
Discovery 3
Web
192.168.59.128:10050
Linux by Zabbix agent
Enabled
ZBX
None
Zabbix server
Items 125
Triggers 69
Graphs 25
Discovery 4
Web
127.0.0.1:10050
Linux by Zabbix agent, Zabbix server health
Enabled
ZBX
None
Displaying 2 of 2 found
0 selected
Enable
Disable
Export
Mass update
Delete
Zabbix 6.0.17. © 2001–2023, Zabbix SIA

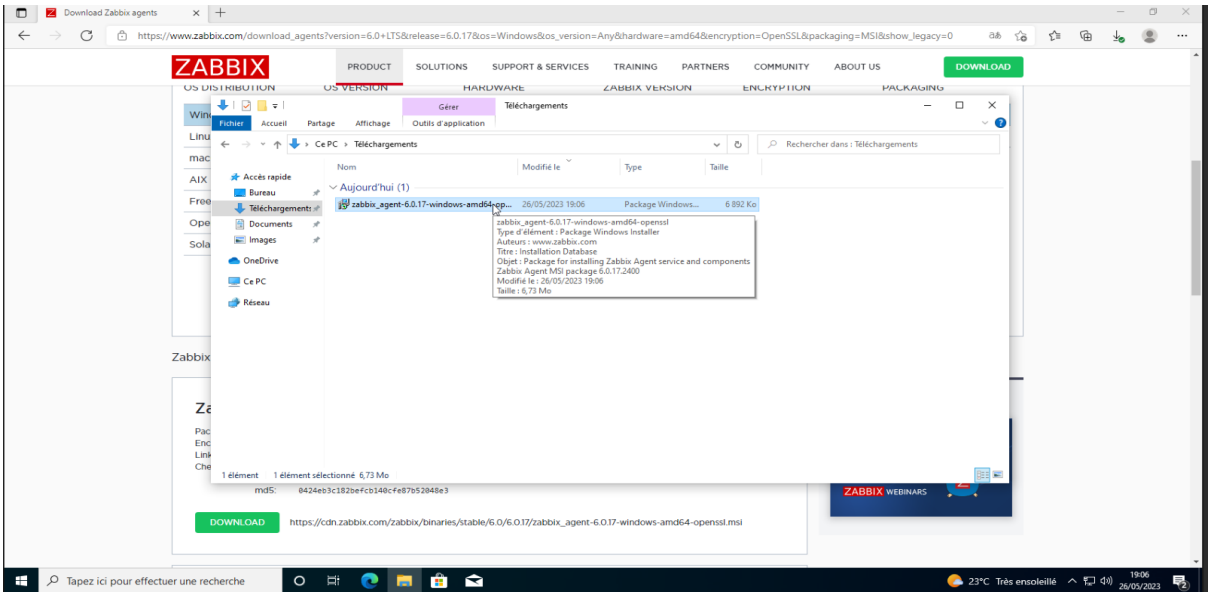
Monitoring Windows machine with Zabbix using agent

Installing Zabbix Agent

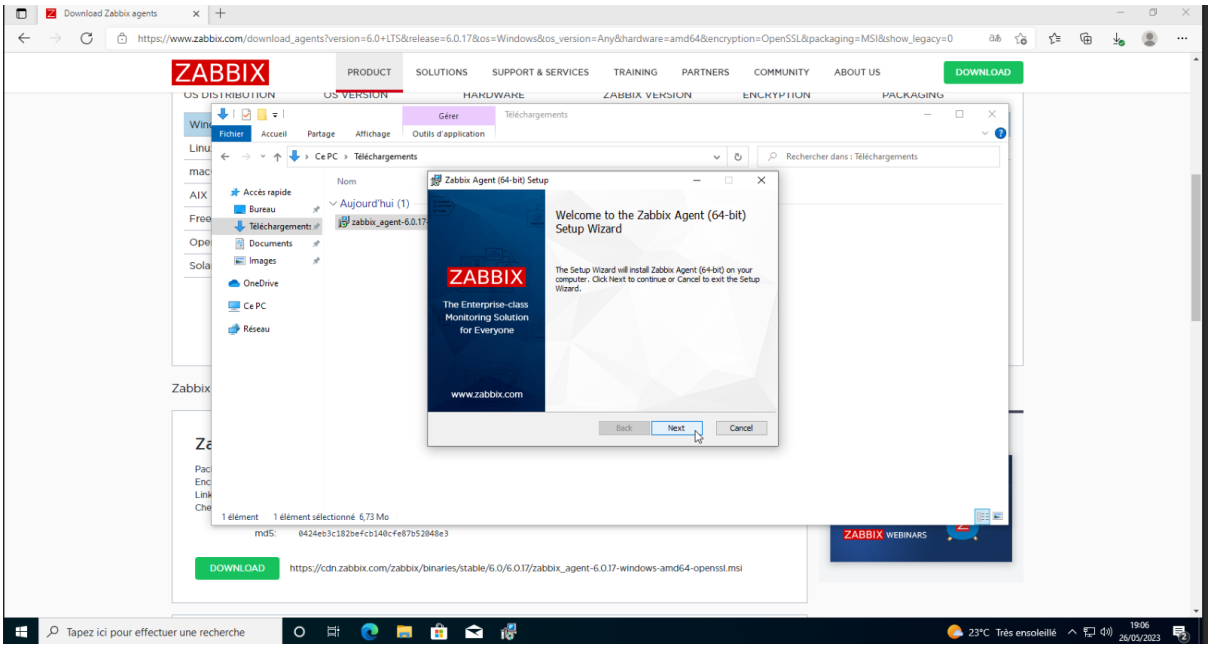
We download Zabbix Windows agent install from the https://www.zabbix.com/download_agents



Launch the installation Wizard

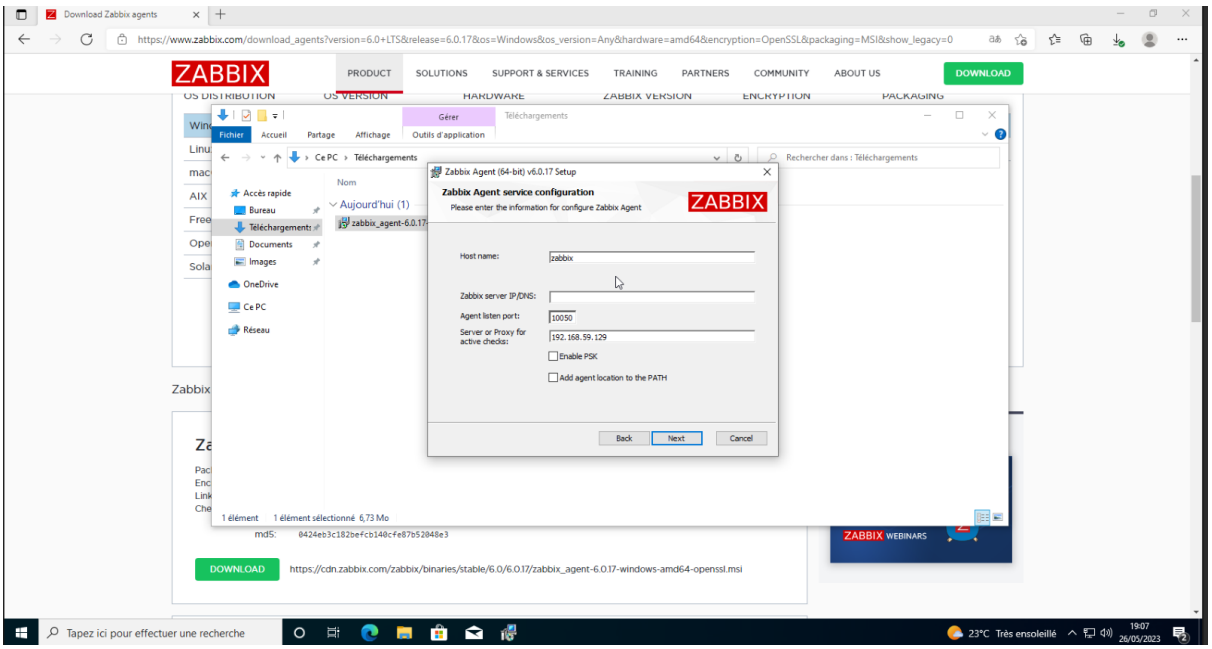


Click Next and accept the End User License Agreement.



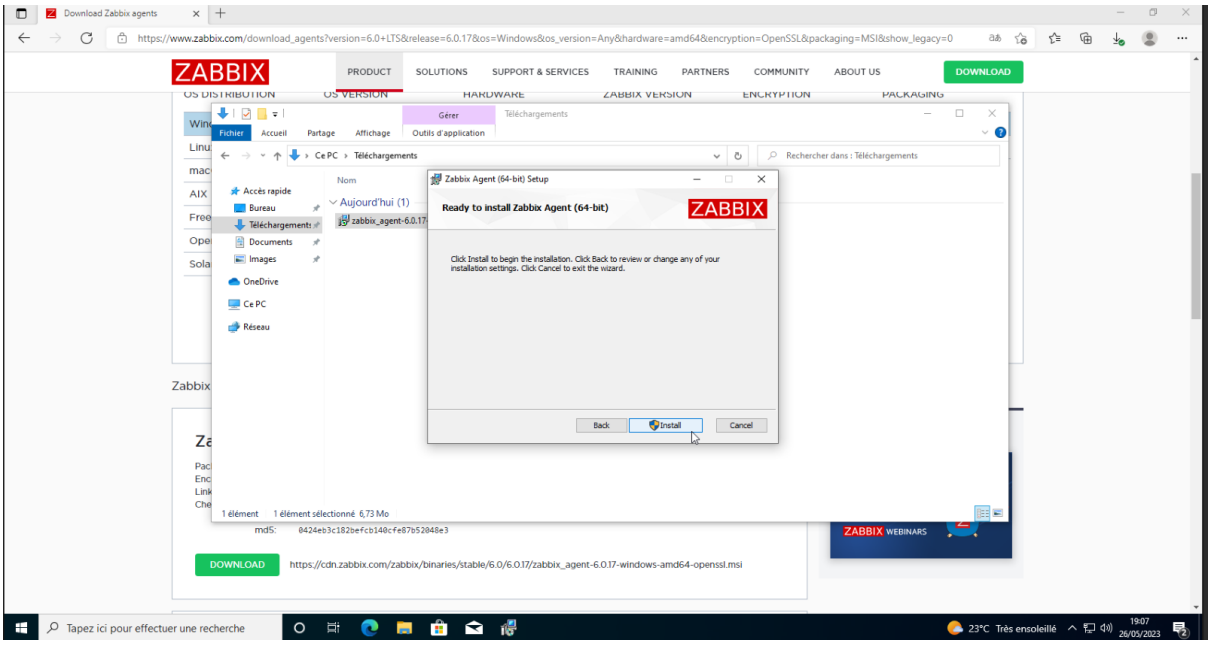
Configuring Zabbix Agent

Next, Set the Zabbix agent hostname, the Zabbix server IP. Define the correct hostname as this must match the hostname while adding the host to Zabbix server for monitoring.



On Custom setup, just click Next.

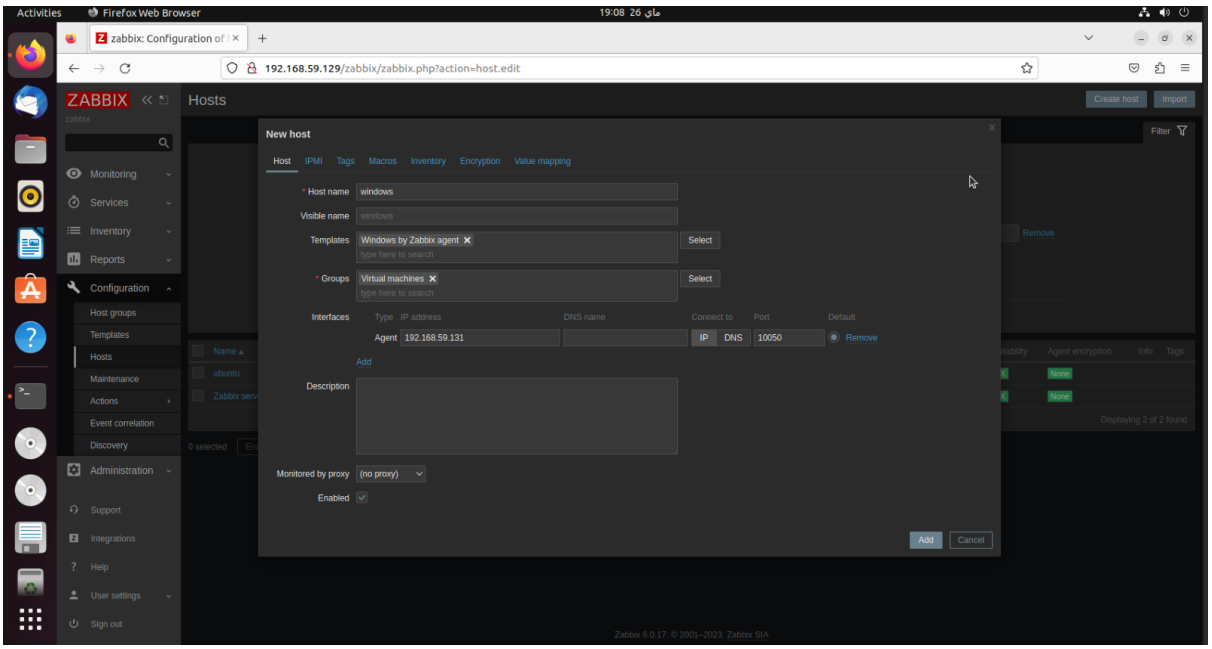
Click Install to install Zabbix agent on Windows.



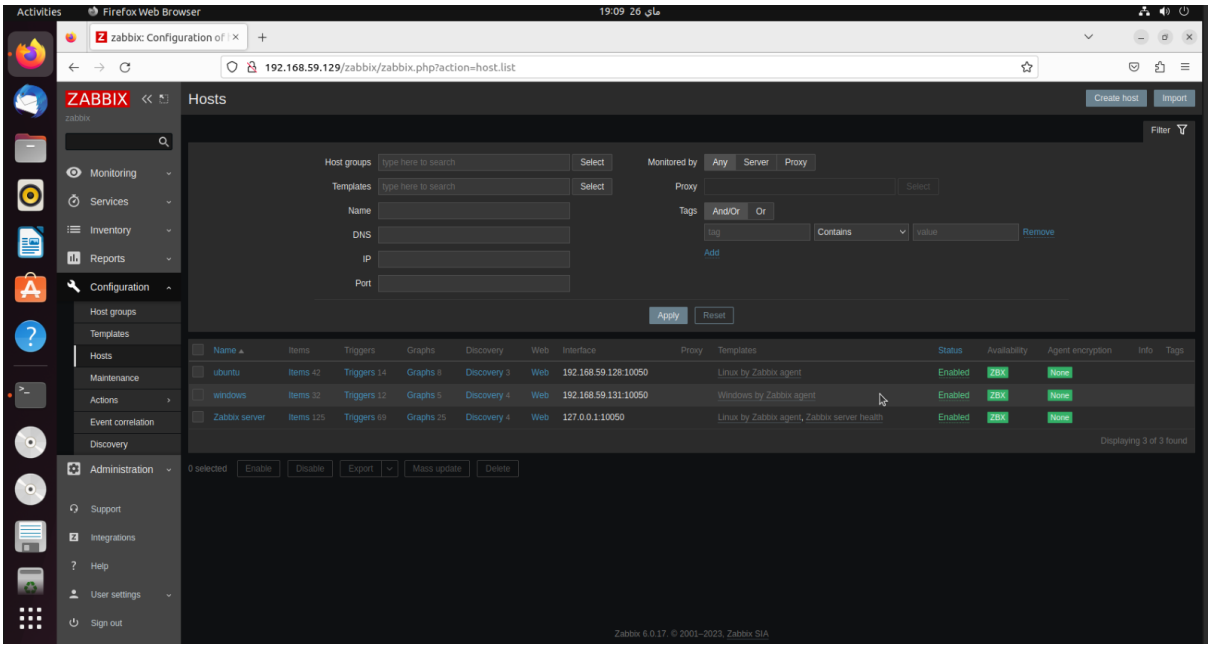
When the installation is done, we click Finish to exit the Zabbix agent installation wizard.

Adding the Zabbix Client to the Zabbix Server

Lastly, we add the newly installed Zabbix Agent to the Zabbix server. We log into the Zabbix web interface, navigate to “Configuration” > “Hosts” > “Create Host”, and we fill in the necessary details.



After adding the host, we should see the host status as “Enabled”.



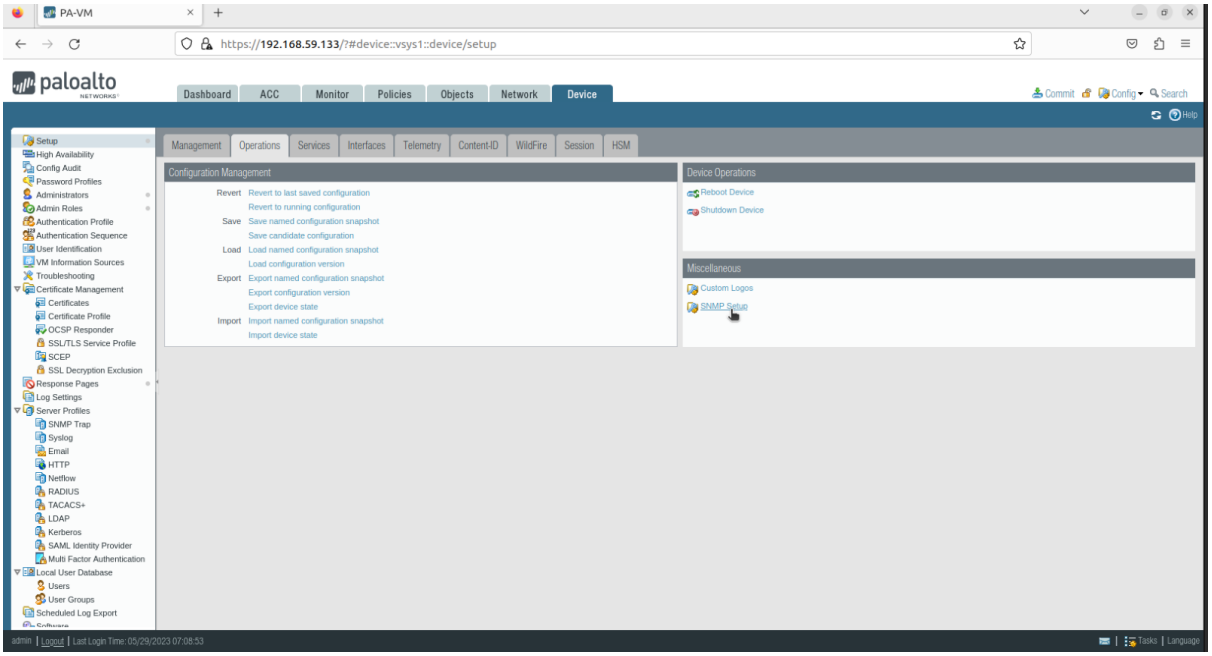
Monitoring Palo Alto with Zabbix via SNMPv3

Configuring and enabling SNMP on Palo Alto

First, we should configure and enable SNMP on Palo Alto.

We navigate to Device > Setup > Operations.

In the lower right corner, click SNMP Setup.



We choose the version SNMPv3. In the View section, we click Add. We enter name for the group, then we configure the following for each view we add to the group:

View: Specify a name for the view. The name can have up to 31 characters that are alphanumeric, periods, underscores, or hyphens.

OID: Specify the OID of the MIB.

Option: Select the matching logic to apply to the MIB.

Mask: Specify the mask in hexadecimal format.

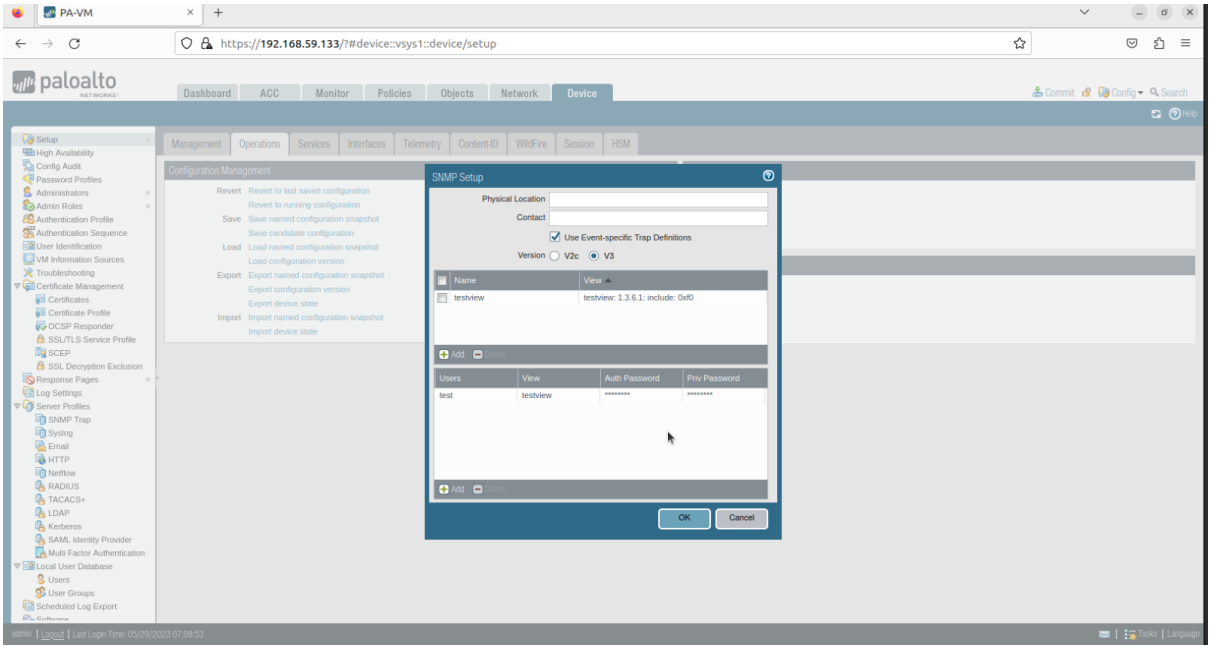
In the User section, we click Add. We enter a name for the user, then we configure the following fields for each view we add to the group:

User Name: Specify a username to identify the SNMP user account. The username we configure on the firewall must match the username configured on the SNMP manager. The username can have up to 31 characters.

View: Assign a group of views to the user.

Authentication Password: Type and confirm the authentication password. The firewall uses the secure hash algorithm (SHA-1 160) to encrypt the password. The password must be between 8 and 256 characters long. All characters are allowed.

Privacy Password: Type and confirm privacy password. The firewall uses the password and Advanced Encryption Standard 128 (AES-128) to encrypt SNMP traps and responses to statistics requests. The password must be between 8 and 256 characters long. All characters are allowed.

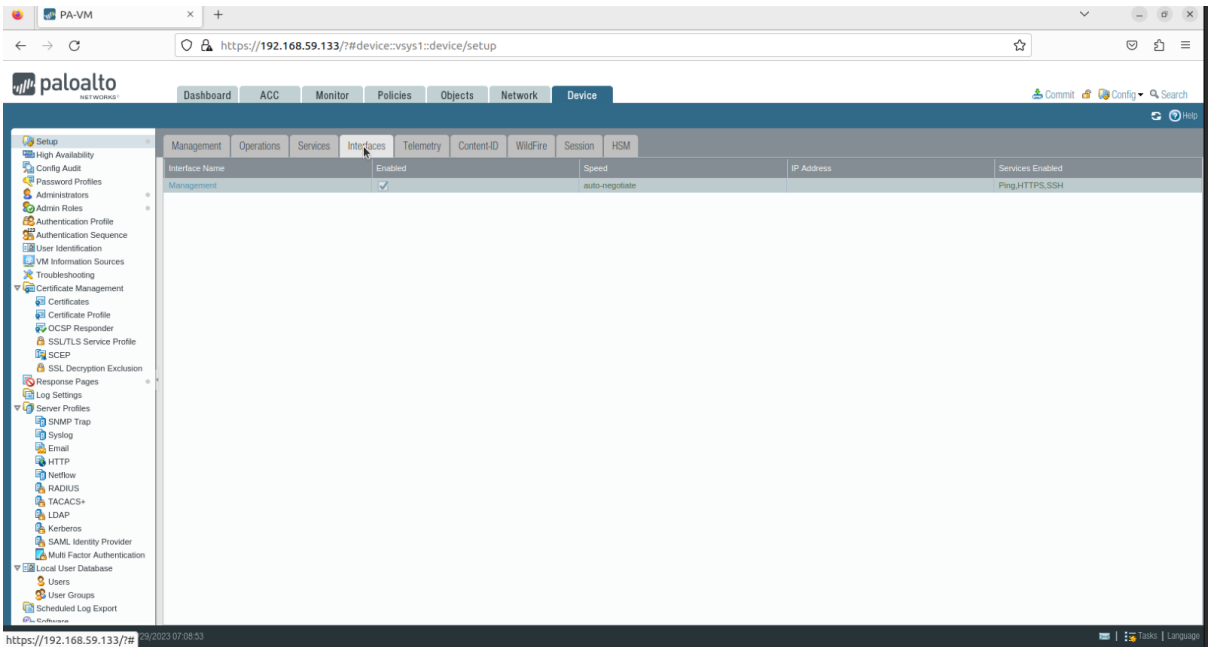


Enabling SNMP on the Management interface

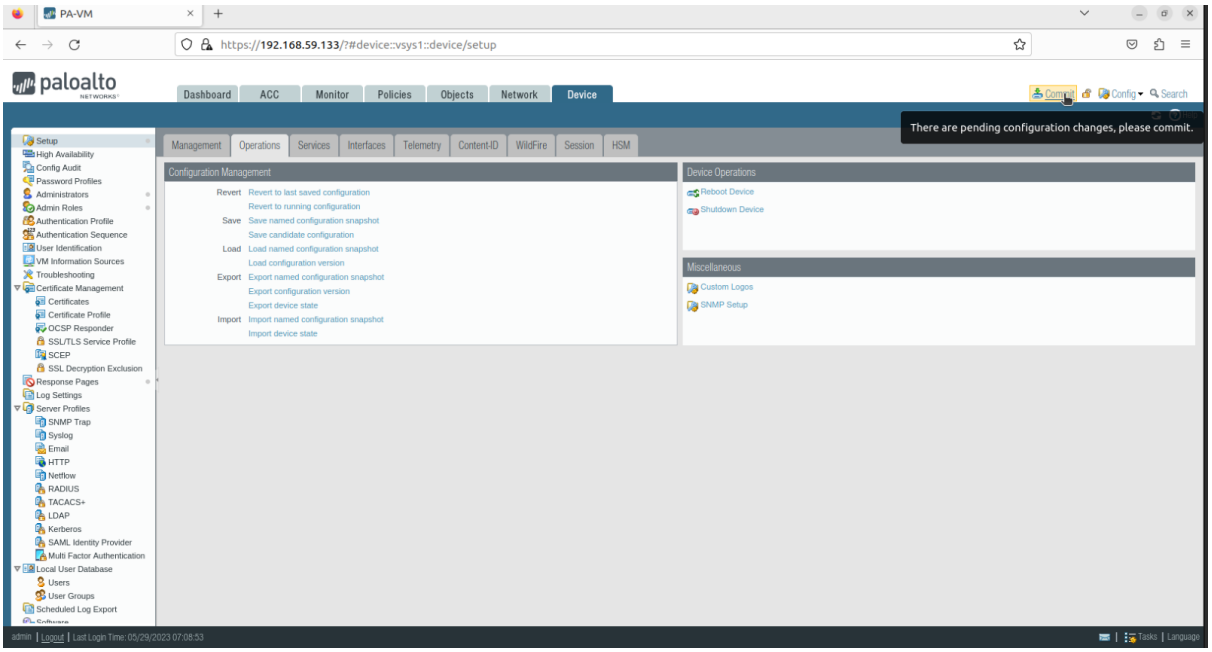
Navigate to Device > Setup > Interfaces.

Click the Management button.

Tick the SNMP box.

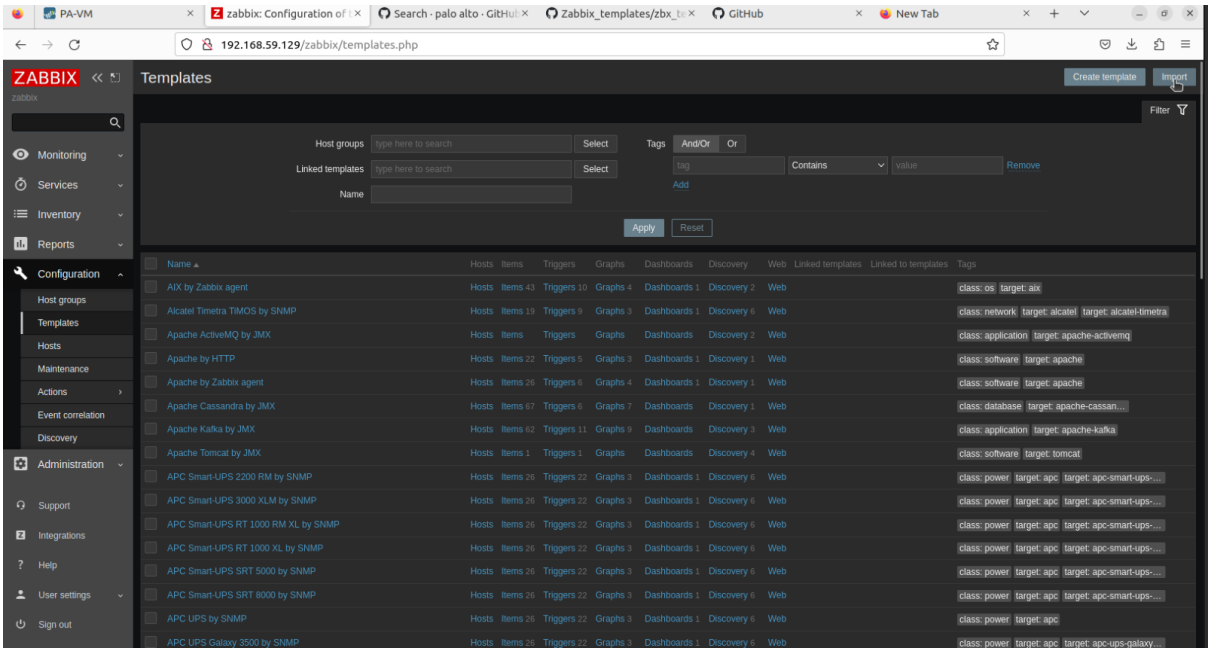


And we click on Commit to valid my configuration

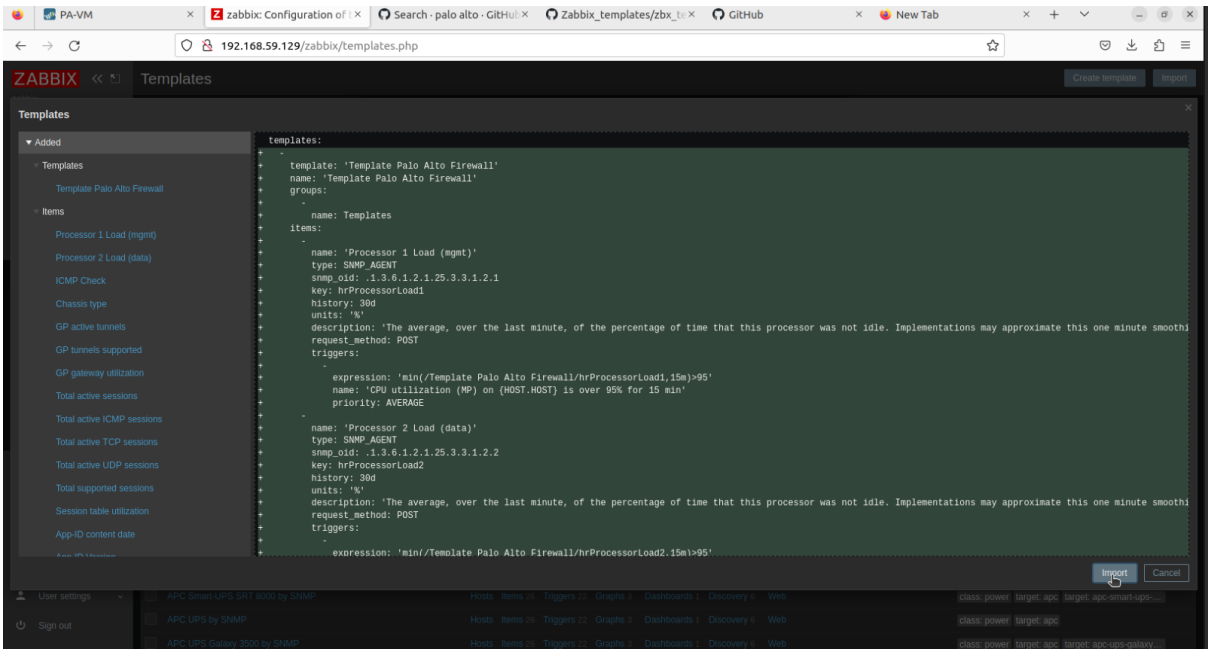


Adding PaloAlto to Zabbix server

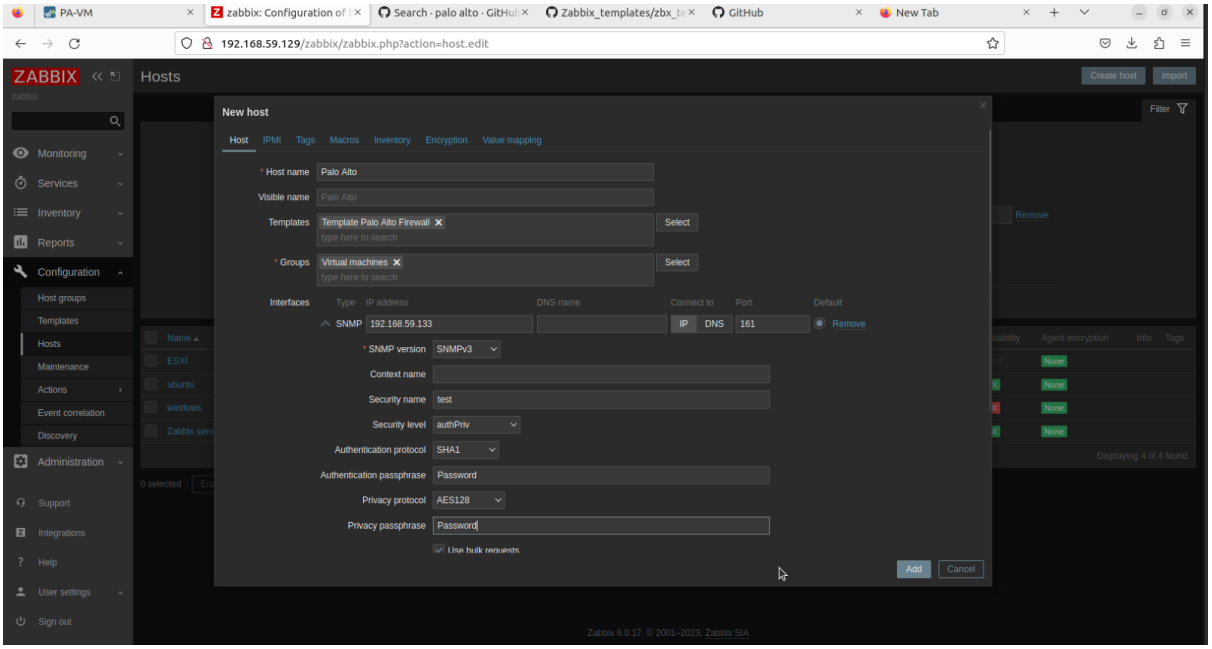
We log in to the Zabbix web interface, we navigate to “Configuration” > Templates” > “Import”.



We import the template Palo Alto SNMPv3 (we find it on github)



We navigate to “Configuration” > “Hosts” > “Create Host”, and we fill in the necessary details.



Monitoring FortiGate with Zabbix via SNMPv3

Configuring and enabling SNMP on FortiGate

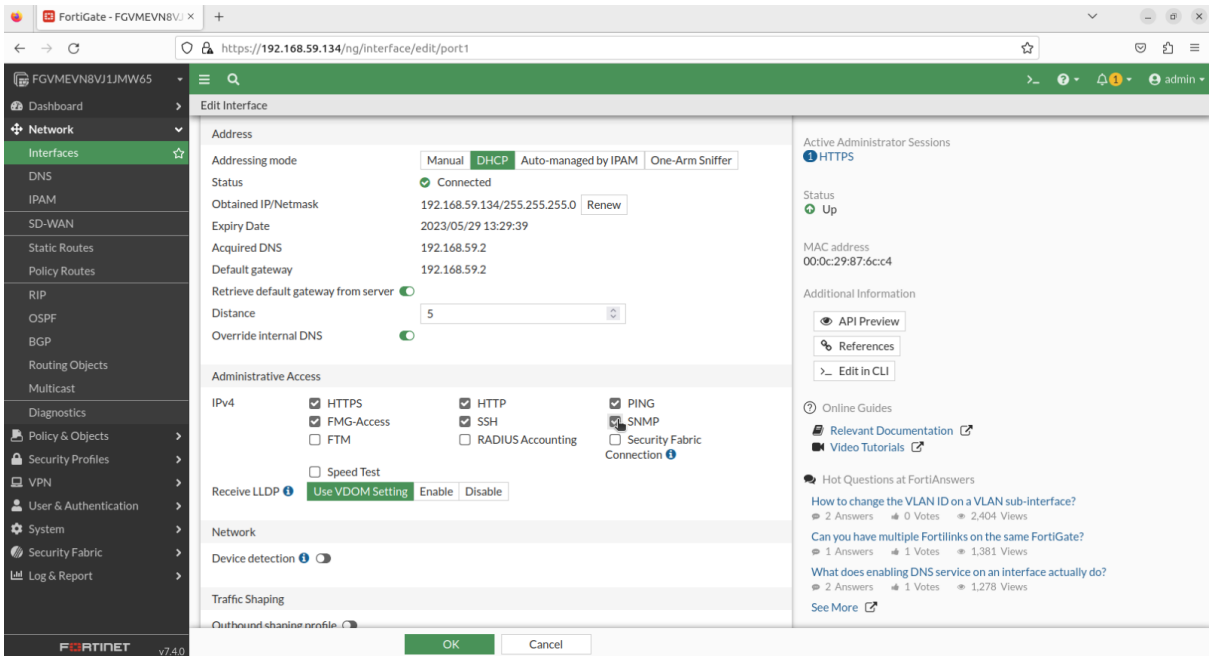
First, we should configure and enable SNMP on FortiGate.

Go to Network -> Interfaces.

Choose an interface that an SNMP manager connects to and select 'Edit'.

In Administrative Access, select 'SNMP'.

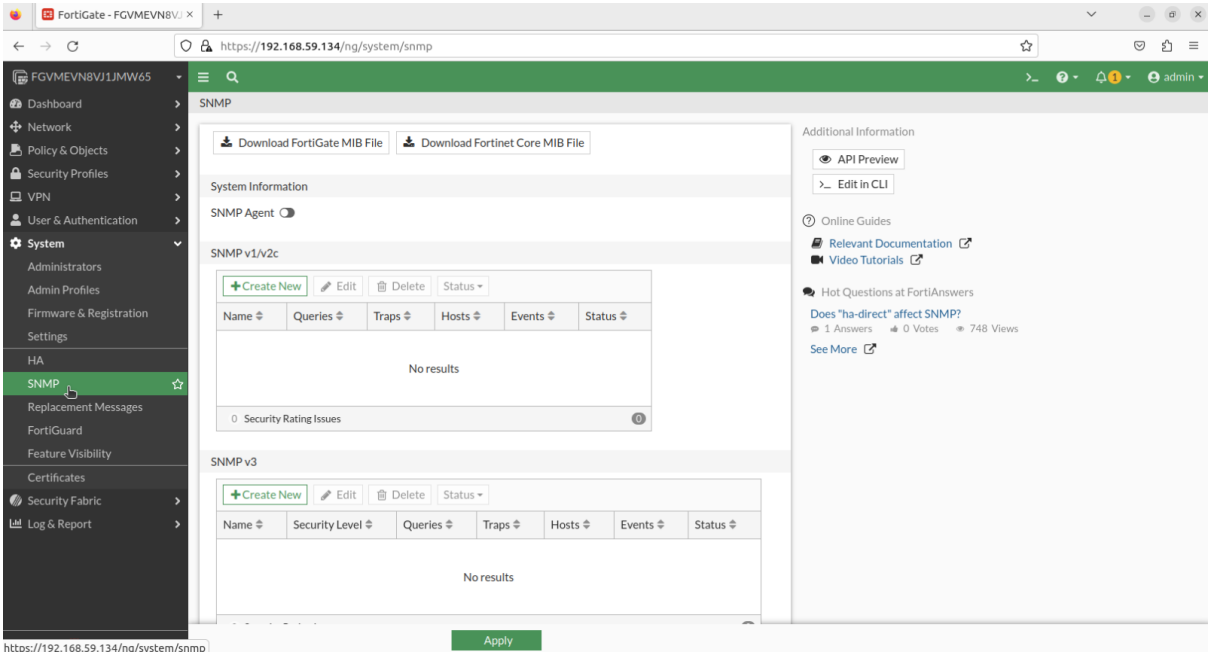
Select 'OK'.



Navigate to "System > SNMP"

Click Enable the SNMP Agent

Click "Apply"



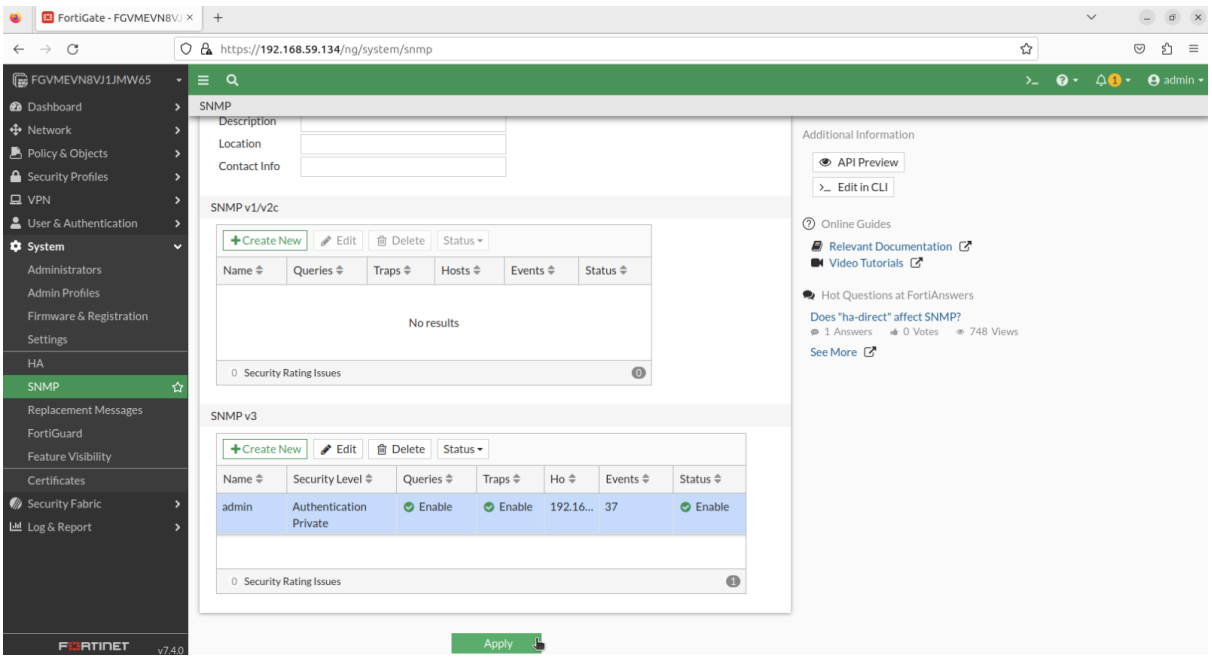
To enable SNMP v3:

In the SNMP v3 section, select "Create New"

Set our preferred User Name and Auth level

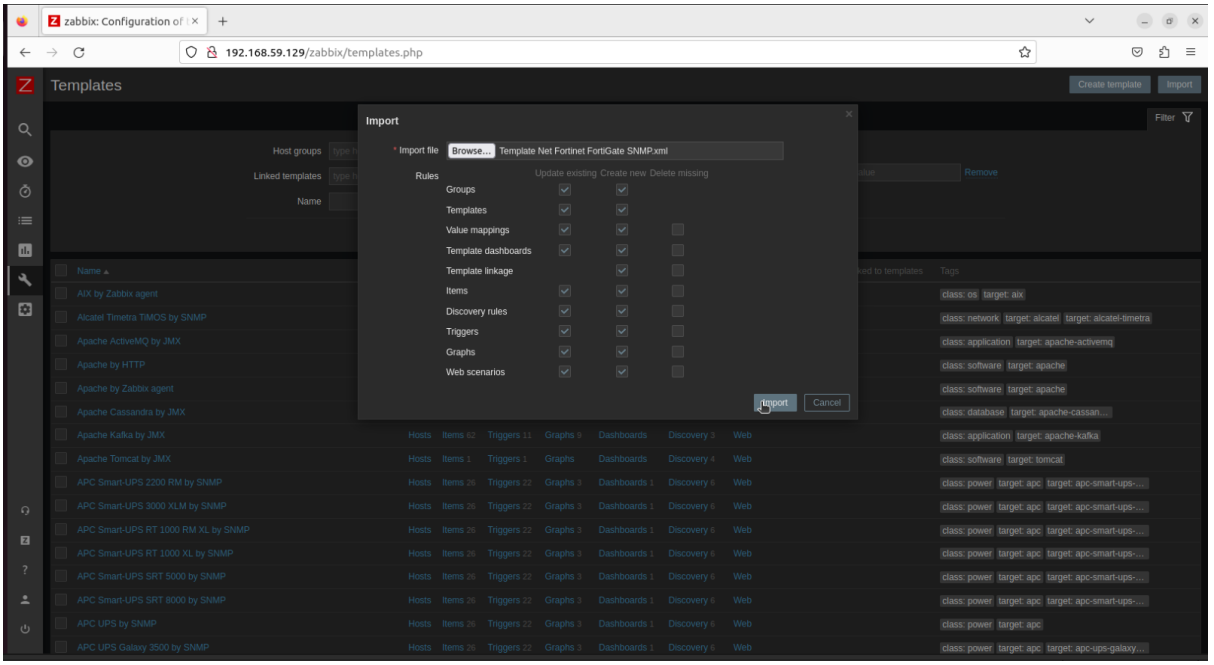
Ensure Enable Query is checked and the port is set to 161

Click "OK".



Adding FortiGate to Zabbix server

We log in to the Zabbix web interface, we navigate to “Configuration” > Templates” > “Import”. (We find it on github)



We navigate to “Configuration” > “Hosts” > “Create Host”, and we fill in the necessary details.

Monitoring ESXI with Zabbix via SNMPv3

Configuring and enabling SNMP on ESXI

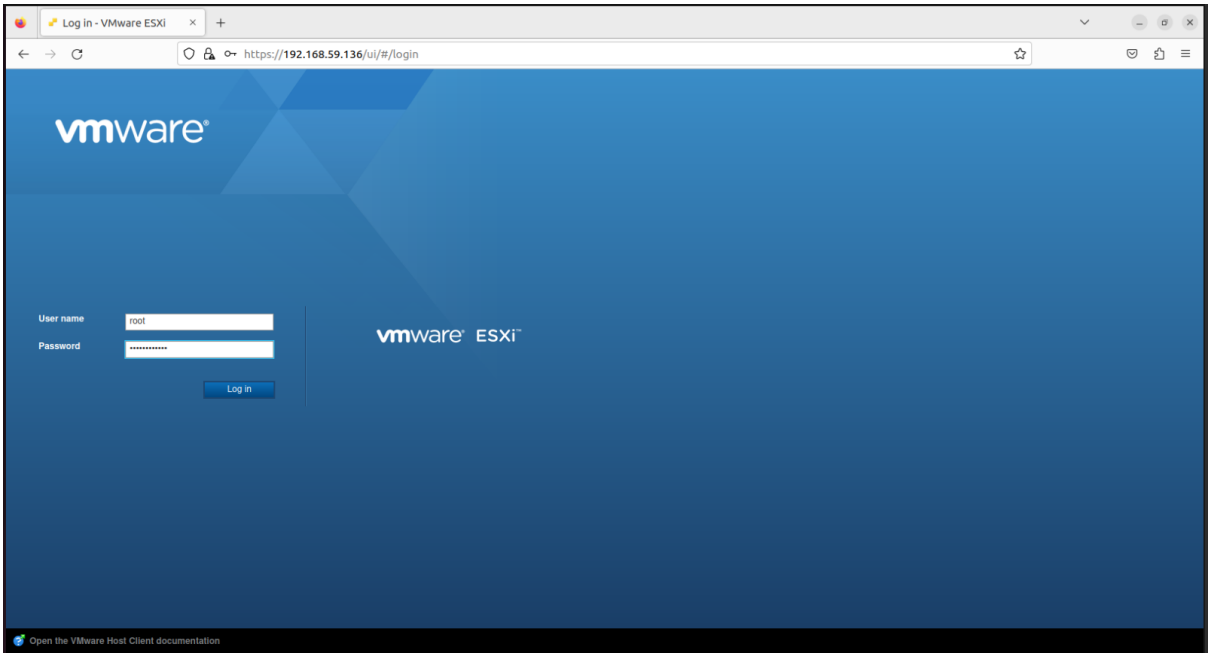
First, we should configure and enable SNMP on ESXI.

Here's how to enable SNMP on ESXi:

- Enable Secure Shell (SSH)
- Configure SNMP
- Configure ESXi Firewall

Enabling SSH Access on ESXi

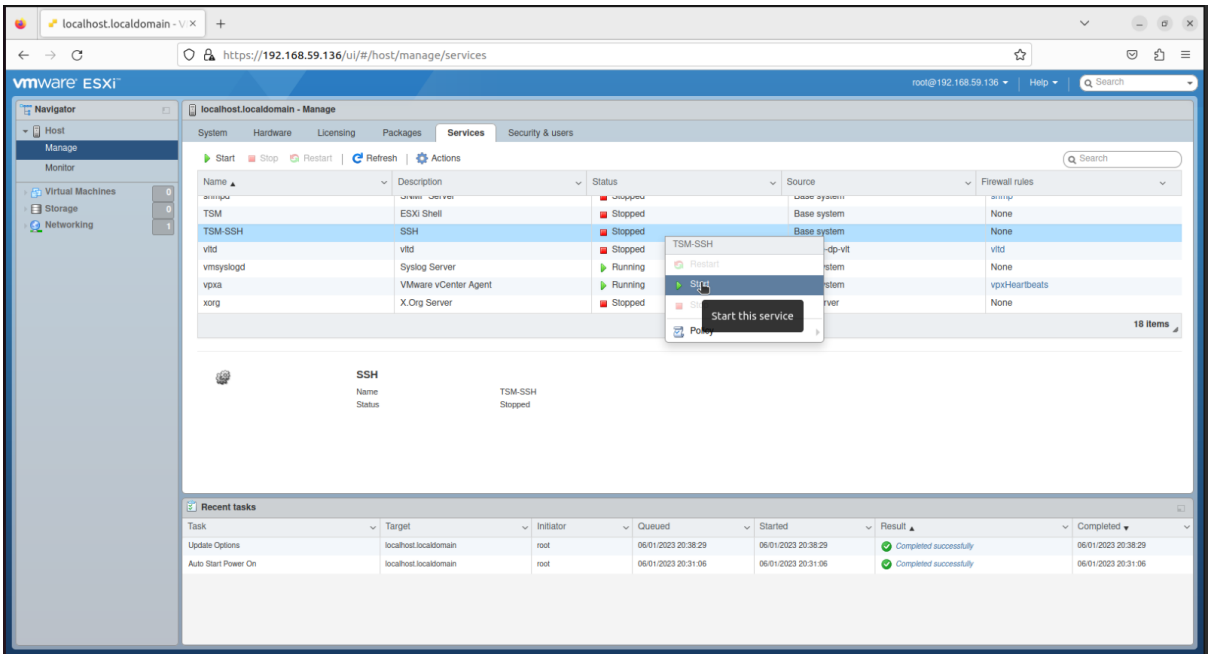
SSH access on an ESXi host is needed to run ESXCLI commands on a host remotely. To enable SSH access to our ESXi host, we can use VMware Host Client. We open a web browser, enter the IP address of our ESXi host in the address bar, then enter credentials to log in.



In the Navigator pane, go to Host > Manage and click the Services tab.

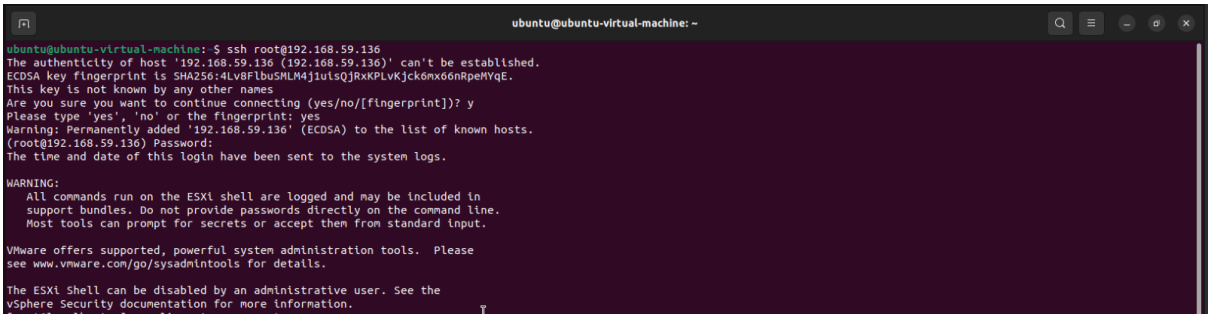
Right-click TSM-SSH and, in the context menu, click Start.

On the screenshot below we see the started SSH server service on the ESXi host.



Now we can connect to the ESXi host from a machine with an SSH client installed. If we're using Windows, we can use PuTTY, a free and convenient SSH client. In Linux, run the SSH client from the command line with the command:

```
> ssh our_username@host_ip_address
```



ESXi SNMP Configuration

Once SSH access to the ESXi host is established, we can configure VMware ESXi SNMP options. On ESXi hosts, SNMP can be configured only in the command-line interface. The graphical user interface (GUI) allows us only to start, stop, and restart the SNMP service.

We run the command in the console (terminal) and we check the SNMP status on the ESXi host:

```
> esxcli system snmp get
```

SNMP is disabled by default. The output for disabled SNMP on ESXi is shown on the screenshot. Most of the parameters are empty and or not configured.

Configuring parameters of an SNMP agent

Set SNMP parameters for an SNMP agent on the ESXi host.

Set the community's name.

```
> esxcli system snmp set --communities techsogroup
```

Set the SNMP target. The SNMP target is a server on which monitoring software is installed to handle SNMP traps and collect monitoring information. In my example, the SNMP target is the machine running Zabbix server (192.168.59.29). UDP 161 is the default port used for SNMP and this port is defined in my ESXi SNMP configuration:

```
> esxcli system snmp set --targets=192.168.59.129@161/user
```

Specify a location, for example, the geographical location, address, datacenter, or a room where the server is located:

```
> esxcli system snmp set --syslocation Casablanca
```

Specify contact information. The system administrator’s email address can be defined for this parameter:

```
> esxcli system snmp set --syscontact user@techsogroup.com
```

Enable SNMP on ESXi:

```
>esxcli system snmp set --enable true
```

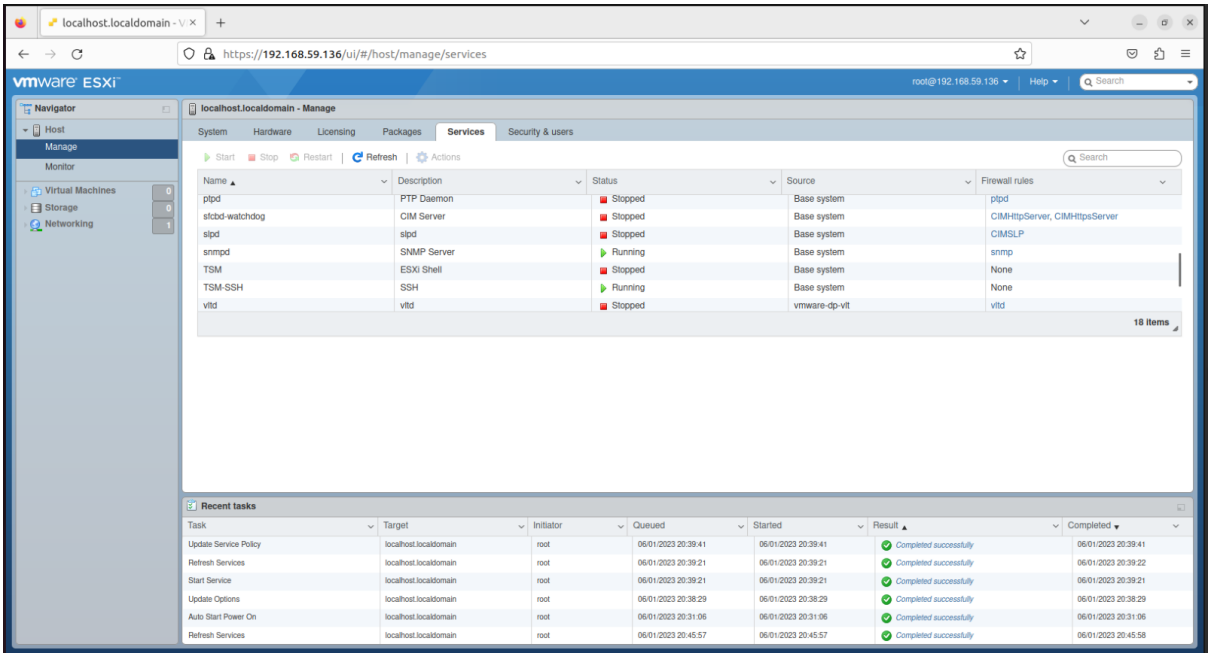
Check the SNMP status on the ESXi host again:

```
>esxcli system snmp get
```

Now we can see that the parameters are configured.

```
[root@localhost:~] esxcli system snmp set --communities techsogroup
[root@localhost:~] esxcli system snmp set --targets=192.168.59.129@161/user
[root@localhost:~] esxcli system snmp set --syslocation Casablanca
[root@localhost:~] esxcli system snmp set --syscontact user@techsogroup.com
[root@localhost:~] esxcli system snmp set --enable true
[root@localhost:~] esxcli system snmp get
   Authentication: SHA1
   Communities: techsogroup
   Enable: true
   Engineid: 80001ADC0512908830881685648637
   Hwsrc: indications
   Largestorage: true
   Loglevel: warning
   Notraps:
   Port: 161
   Privacy: AES128
   Remoteusers:
   Syscontact: user@techsogroup.com
   Syslocation: Casablanca
   Targets: 192.168.59.129@161 user
   Users: user/5fd187207553e6dd5708c7ae4aa1d7e7904c40b1/abed6919b7469d416104f5d73cd142772ec5001e/priv
   V3targets: 192.168.59.129@161 user priv trap
```

SNMP status is running now. We can also open VMware Host Client, go to Host > Manage > Services, and check the status of the snmpd service.

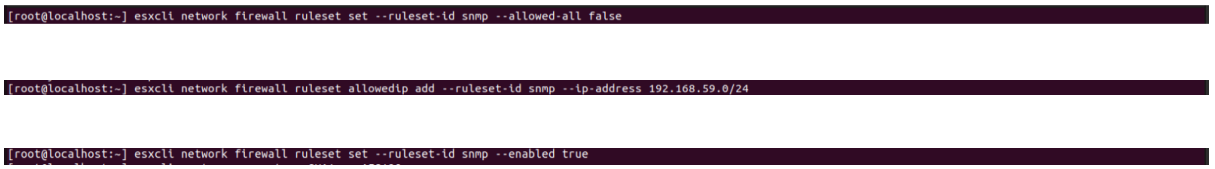


Configuring ESXi Firewall

We must configure the firewall and enable SNMP access from monitoring servers to the ESXi host. We can set a subnet or a single IP address of allowed devices in the network.

We run these three commands to allow access from the 192.168.59.0/24 network to monitor ESXi via SNMP:

```
>esxcli network firewall ruleset set --ruleset-id snmp --allowed-all
false

>esxcli network firewall ruleset allowedip add --ruleset-id snmp
--ip-address 192.168.59.0/24

>esxcli network firewall ruleset set --ruleset-id snmp --enabled
true
```



Configuring SNMP v3

SNMP v3 is a more secure version of the protocol providing key authentication and encryption. Below is an overview of how to enable SNMP v3 on an ESXi host.

We set authentication protocol and privacy options.

```
> esxcli system snmp set -a SHA1 -x AES128
```

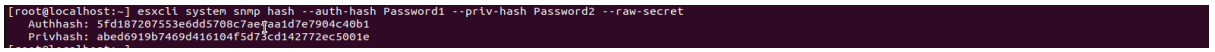
Where:

SHA1 is the algorithm for cryptographic encryption, the cryptographic hash function (Secure Hash Algorithm 1).

AES128 is the encryption method (Advanced Encryption Standard with a 128-bit encryption key) using the symmetric block cipher.

We generate hashes by using the command:

```
> esxcli system snmp hash --auth-hash Password1 --priv-hash
Password2 --raw-secret
```



We use the generated hashes and add a user. Adding up to five users is supported.

```
> esxcli system snmp set -e yes -C user -u
snmpuser/authhash/privhash/priv
```

Where:

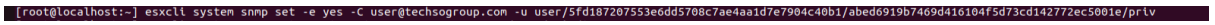
user is the contact email for the user

snmpuser is the username (can be up to 32 characters)

authhash is the authentication hash value

privhash is the privacy hash value

We add user and use hashes generated in the output of the previous command.

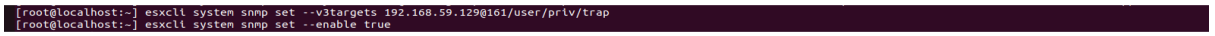


Define the SNMP target address:

```
> esxcli system snmp set --v3targets
192.168.59.129@161/user/priv/trap
```

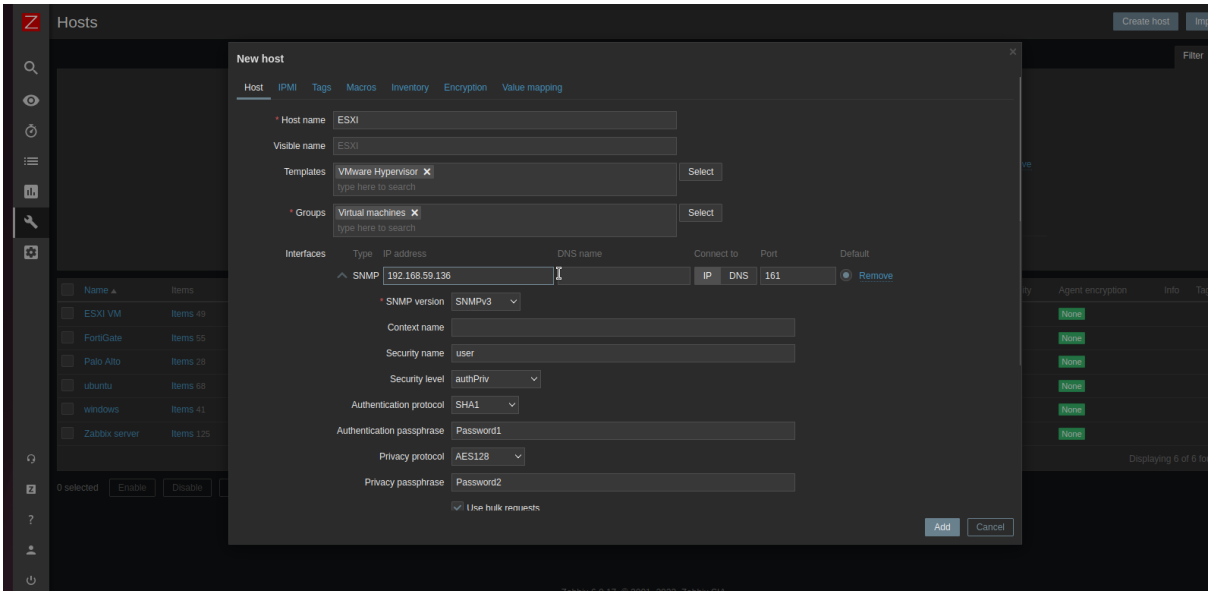
We enable SNMP on ESXi:

```
>esxcli system snmp set --enable true
```



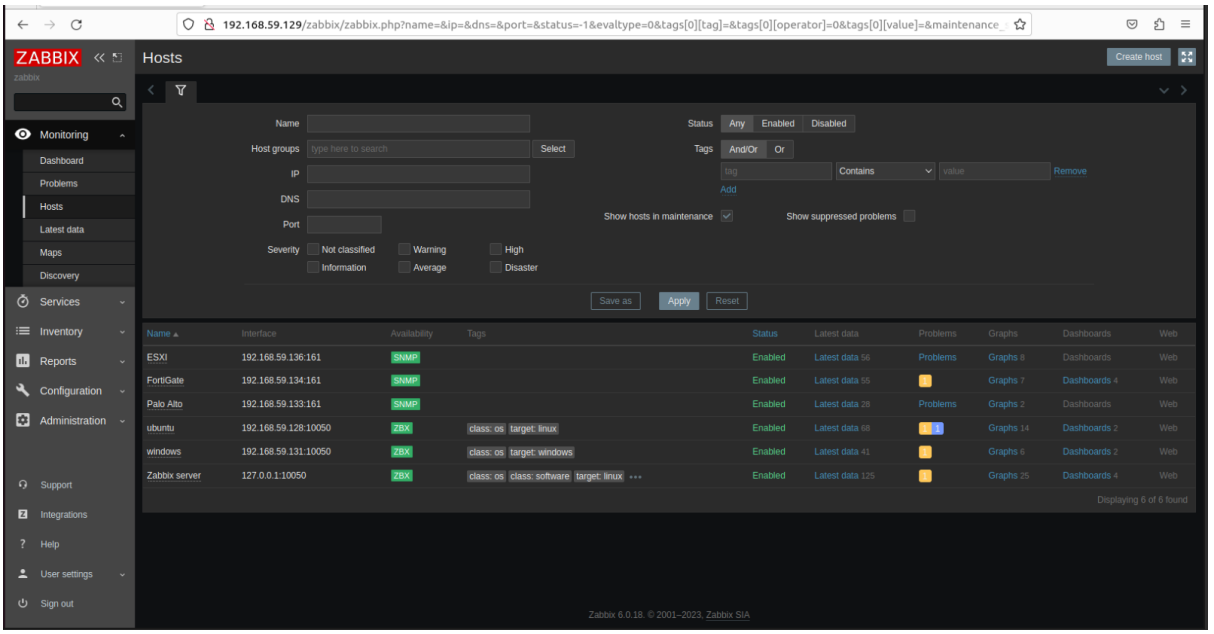
Adding ESXI to Zabbix server

We log in to the Zabbix web interface, we navigate to “Configuration” > “Hosts” > “Create Host”, and we fill in the necessary details.


Hosts
Create host
Filter
New host
Host
IPMI
Tags
Macros
Inventory
Encryption
Value mapping
* Host name
ESXI
Visible name
ESXI
Templates
VMware Hypervisor
type here to search
Select
* Groups
Virtual machines
type here to search
Select
Interfaces
Type
IP address
DNS name
Connect to
Port
Default
SNMP
192.168.59.136
IP
DNS
161
Remove
* SNMP version
SNMPv3
Context name
Security name
user
Security level
authPriv
Authentication protocol
SHA1
Authentication passphrase
Password1
Privacy protocol
AES128
Privacy passphrase
Password2
Use bulk requests
Add
Cancel
Name
Items
ESXI VM
FortiGate
Palo Alto
ubuntu
windows
Zabbix server
Agent encryption
None
0 selected
Enable
Disable

Zabbix Maps for Better Visualization

Here's a screen from my Zabbix server after adding all hosts.

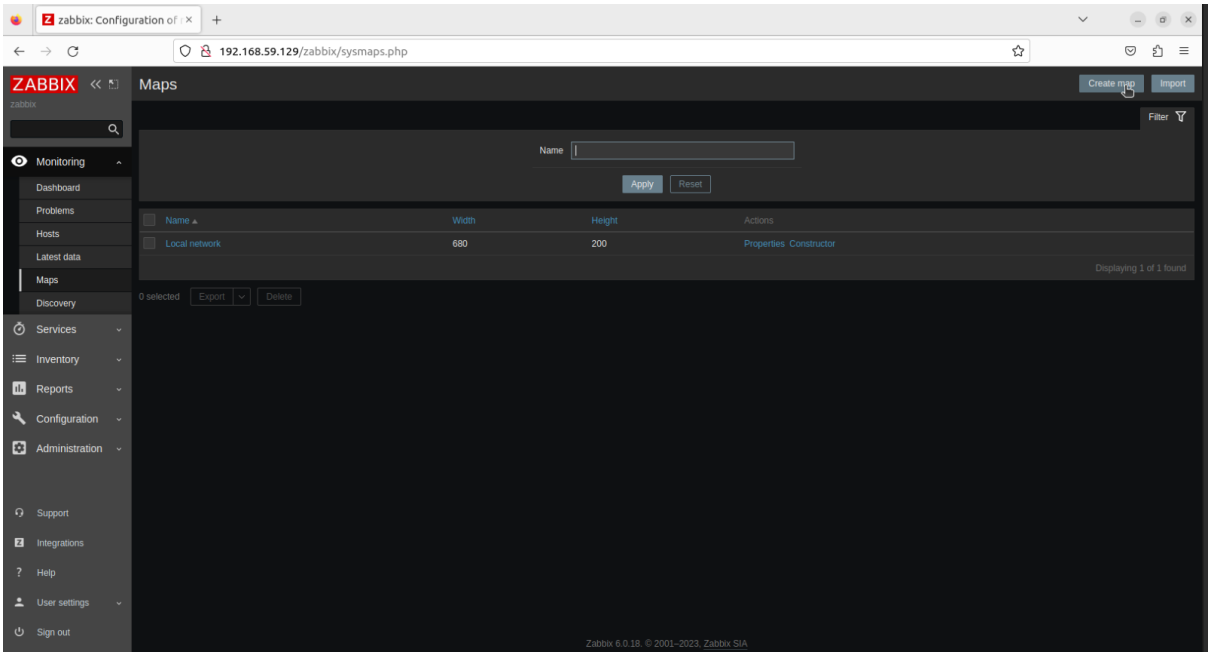


Maps, one of many visualization options inside the Zabbix, a way to display our collected data, show the problems and create our topology. A map is more than a simple picture. It includes many small but very useful key features.

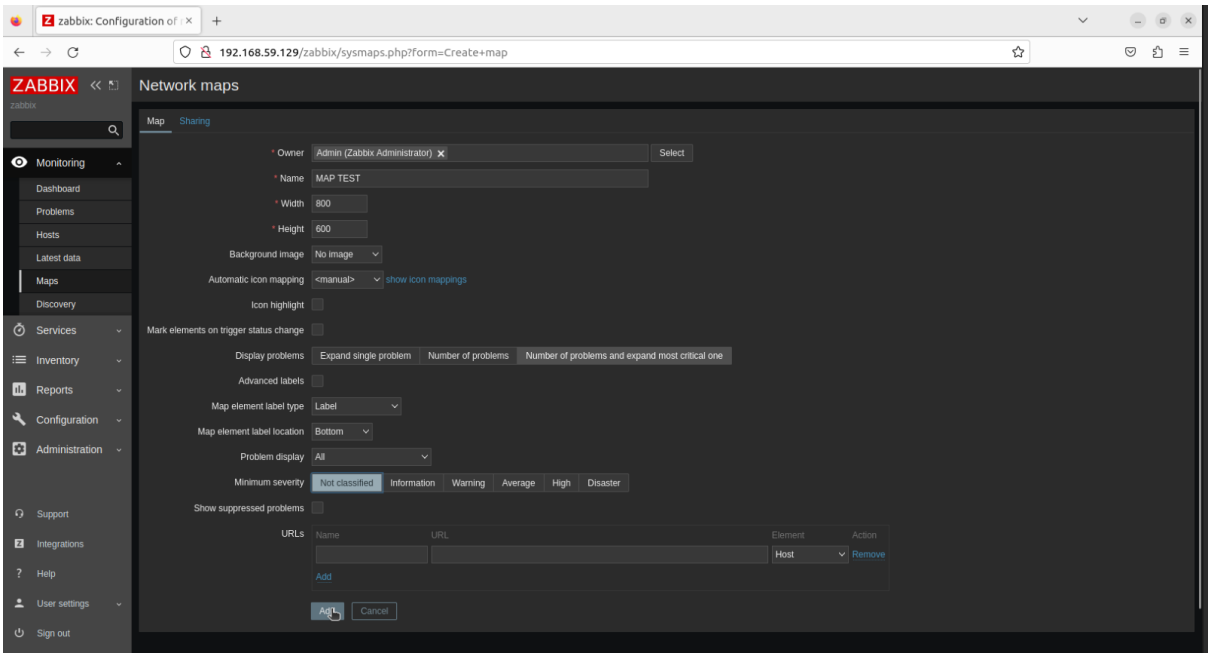
Maps functionality is not something complicated that we need to do in Configuration, or to collect data from our host, etc. Maps inside Zabbix — one of the visualization options used inside the frontend to visualize what is currently happening in our monitoring environment.

Creating a new map

To create a map in Zabbix, we go to **Monitoring** > **Maps** and we click on **Create map**.

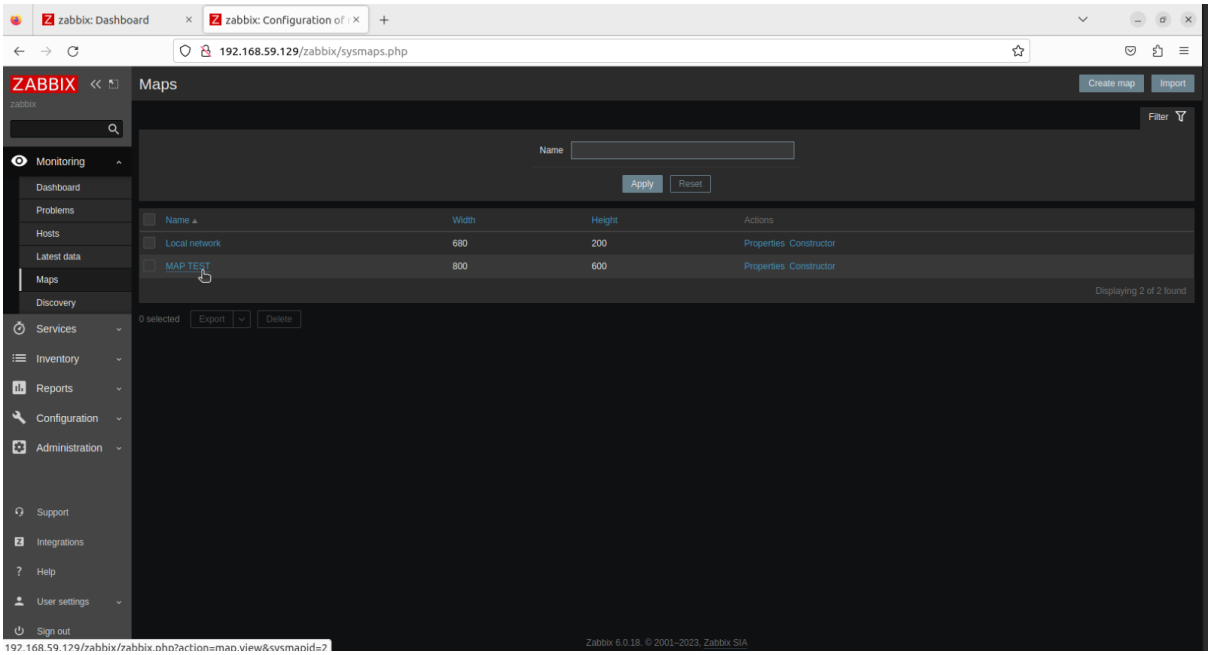


We define the needed attributes in the Map tab and Click **Add**.

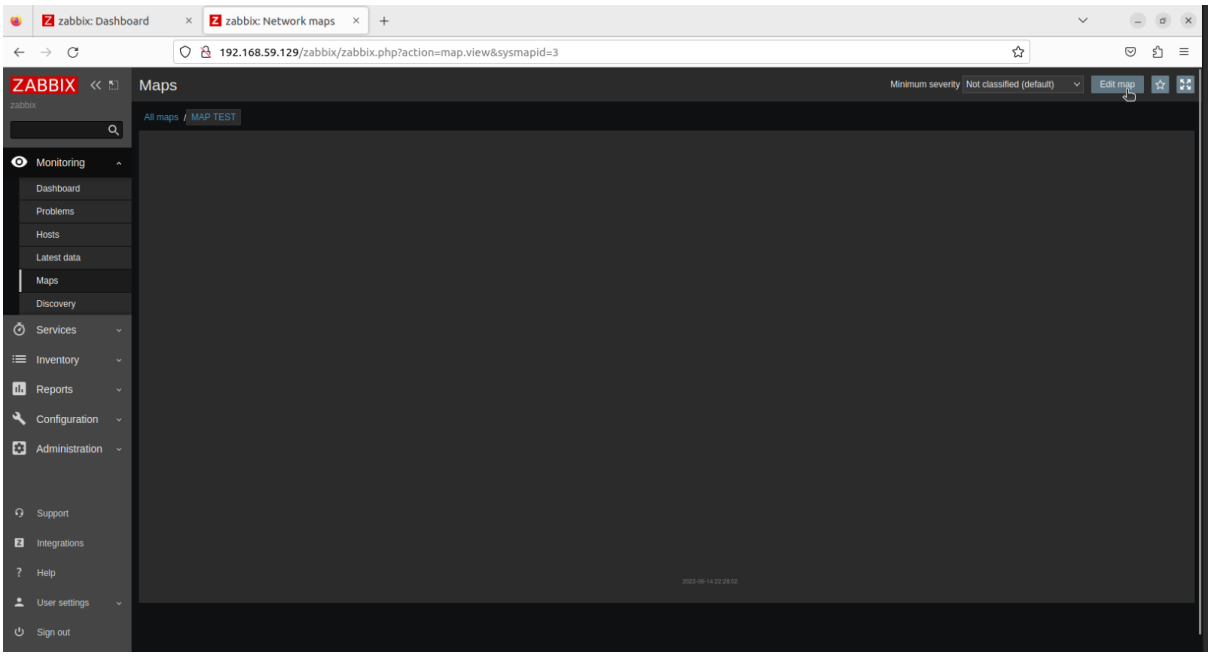


Configuring new map

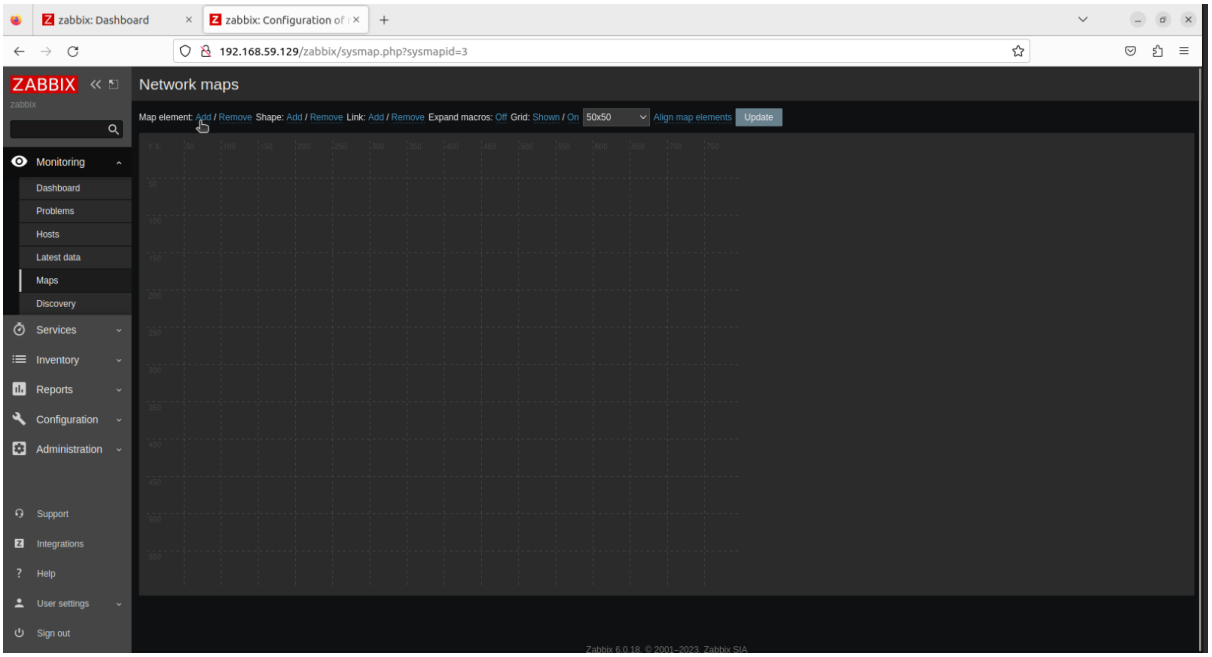
We click the new map name to open our new background. Now we can populate the map with elements such as hosts, host groups, triggers, images or other maps.



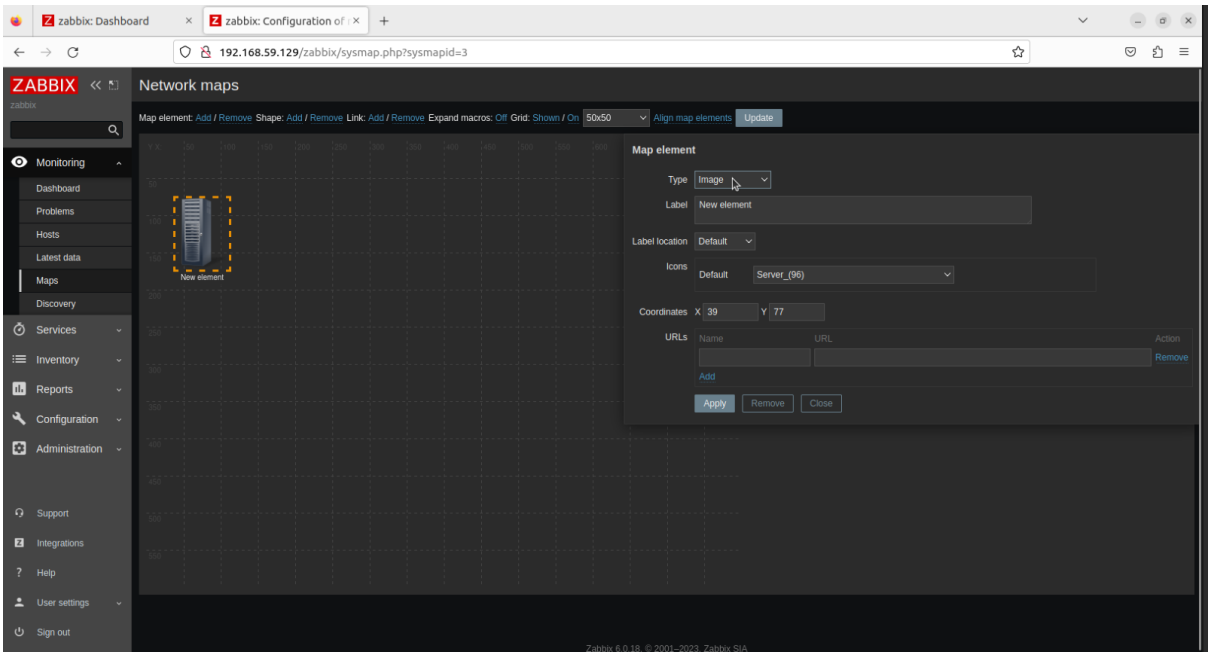
We open the map and we click **Edit map.**



In the settings window, we can start adding elements.

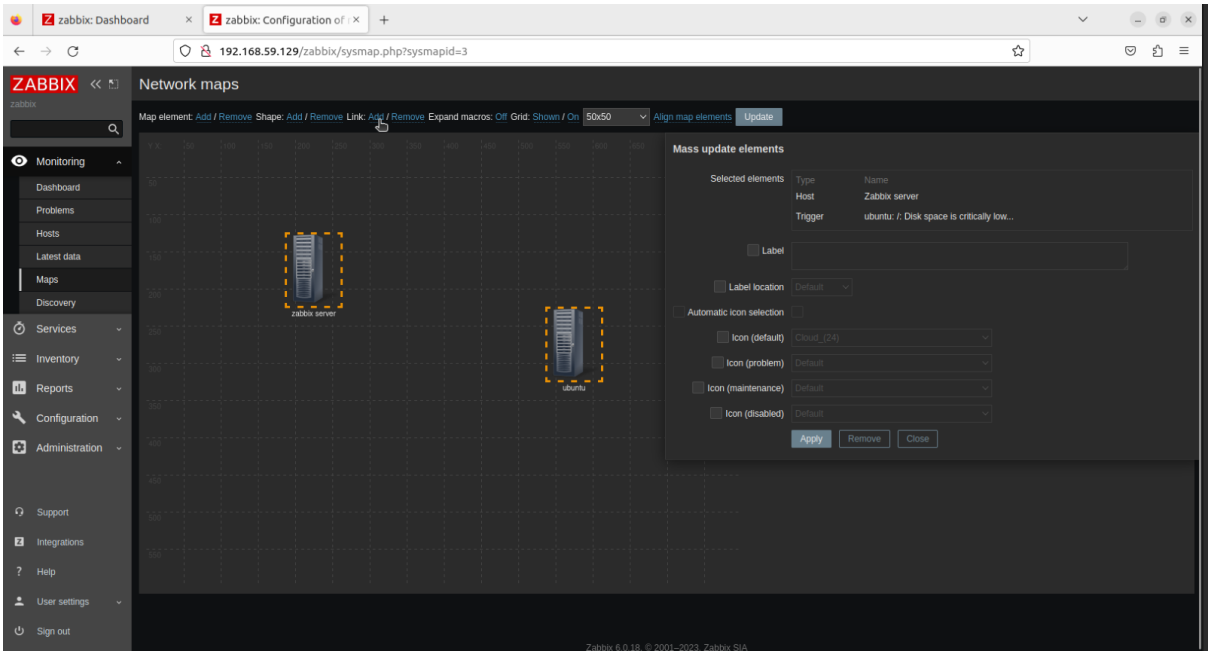


We click **Add** in **Map element** to add a new element. Then double click on it to define the Map element parameters.

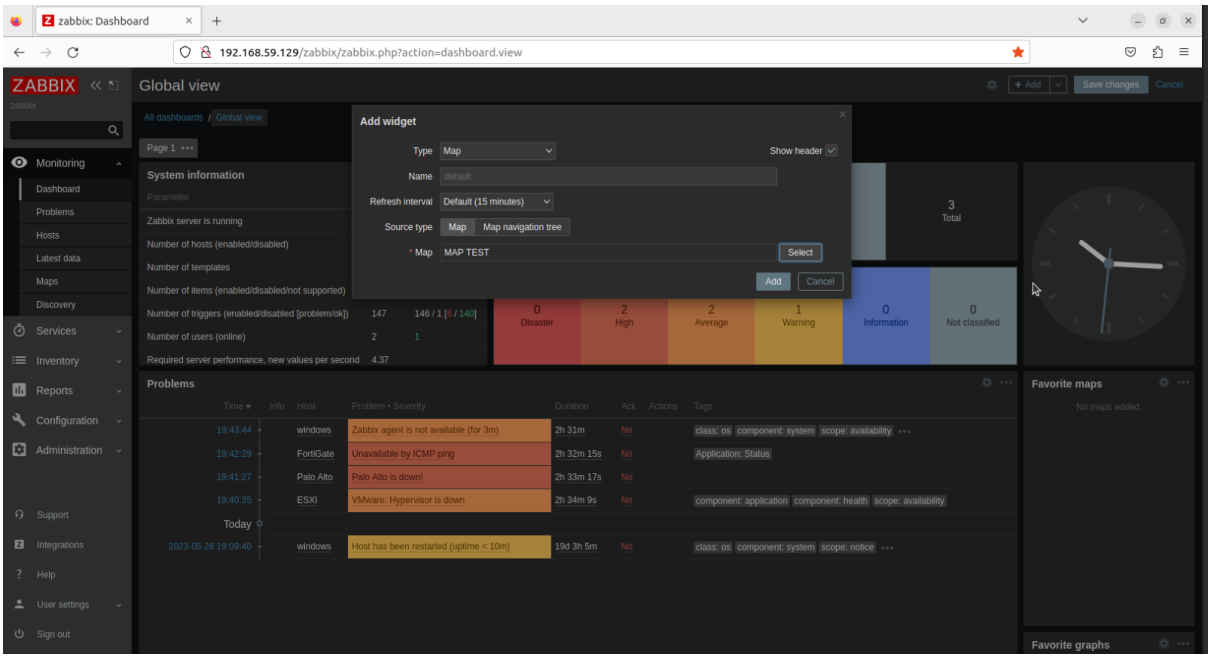


Linking map elements

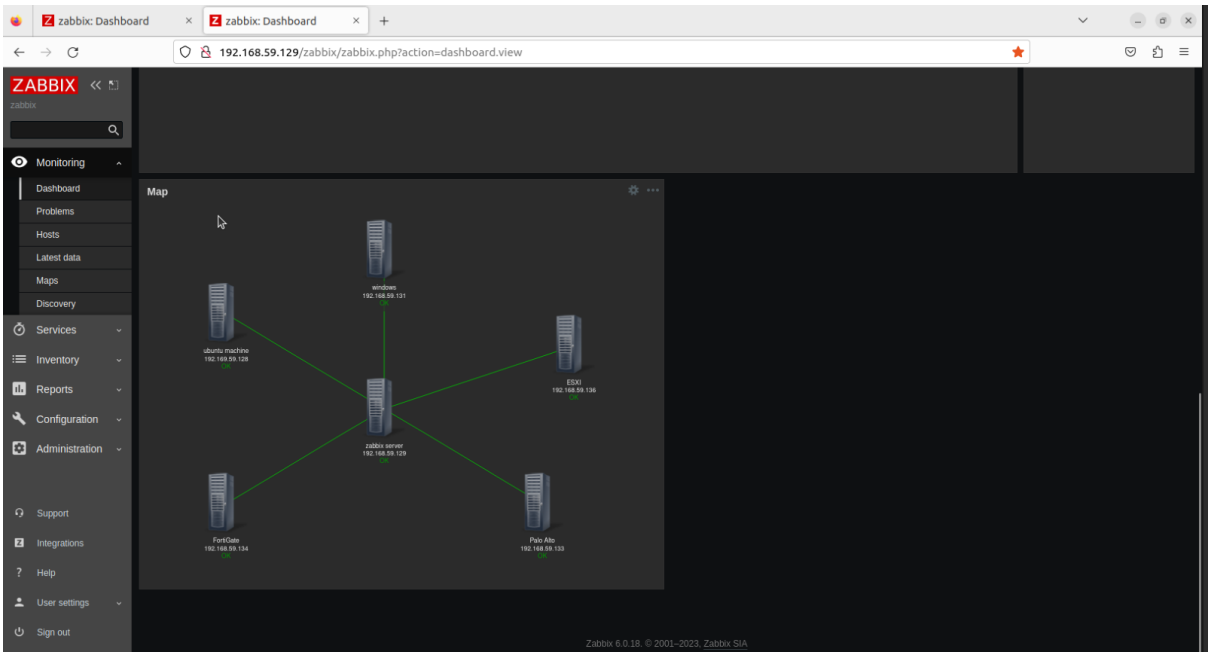
Once we have finished adding the elements, we can start linking them. To link two elements, we need to drag around the two hosts or select them holding a Ctrl key. Then we click Add next to Link.



After adding all hosts and linking them, we go to Dashboard and we click on add widget.



And this is what we got



Conclusion

The Zabbix project successfully deployed and configured a robust monitoring and performance management platform. The project involved installing and configuring Zabbix on an Ubuntu server, as well as setting up and managing various client machines and servers, including Ubuntu, Windows, PaloAlto, FortiGate, and ESXi.

By leveraging the capabilities of Zabbix, the project achieved comprehensive surveillance and monitoring across the network architecture. The Zabbix server, powered by a relational database, stored monitoring data efficiently, while the web interface provided convenient access to data and system configurations.

Throughout the project, meticulous attention was given to ensuring seamless communication, data collection, and analysis. The implementation of Zabbix agents on client machines and servers allowed for real-time monitoring and performance evaluation.

Overall, this project successfully established a robust monitoring infrastructure that enhances network visibility, facilitates prompt issue detection, and enables effective performance management. The deployment and configuration of Zabbix have laid a solid foundation for continuous monitoring and optimization of the network's performance and reliability.