

Practica 3

Ejercicio 1

Instalación

Nos dirigimos a la página de Zabbix, y seleccionamos la versión 5.0 LTS, para una distribución Ubuntu, versión 20.04, con base de datos MySQL, y servidor web Apache.

Lo primero descargar el archivo `zabbix-release_5.0-1+focal_all.deb`, y usando `dpkg` lo instalamos y por último hacemos un `update` para actualizar los repositorios. Para ello se usan los comandos

```
wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
```

```
dpkg -i zabbix-release_5.0-1+focal_all.deb
```

```
apt update
```

```
momid@momid:~$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
--2021-05-05 17:07:08-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4244 (4.1K) [application/octet-stream]
Saving to: 'zabbix-release_5.0-1+focal_all.deb'

zabbix-release_5.0-1+foc 100%[=====>] 4.14K --.-KB/s in 0s

2021-05-05 17:07:09 (17.2 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]

momid@momid:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
[sudo] password for momid:
Selecting previously unselected package zabbix-release.
(Reading database ... 63070 files and directories currently installed.)
Preparing to unpack zabbix-release_5.0-1+focal_all.deb ...
Unpacking zabbix-release (1:5.0-1+focal) ...
Setting up zabbix-release (1:5.0-1+focal) ...
momid@momid:~$ _
```

```
momid@momid:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4930 B]
Get:4 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1194 B]
Get:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [3218 B]
Get:6 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [953 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [217 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [13.2 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [208 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [30.9 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [436 B]
Get:14 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [765 kB]
Get:15 http://es.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [164 kB]
Get:16 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [17.2 kB]
Get:17 http://es.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [21.7 kB]
Get:18 http://es.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [600 B]
Get:19 http://es.archive.ubuntu.com/ubuntu focal-security/main amd64 Packages [629 kB]
Get:20 http://es.archive.ubuntu.com/ubuntu focal-security/main Translation-en [128 kB]
Get:21 http://es.archive.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [7476 B]
Get:22 http://es.archive.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [185 kB]
Get:23 http://es.archive.ubuntu.com/ubuntu focal-security/restricted Translation-en [27.2 kB]
Get:24 http://es.archive.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [392 B]
Get:25 http://es.archive.ubuntu.com/ubuntu focal-security/universe amd64 Packages [557 kB]
Get:26 http://es.archive.ubuntu.com/ubuntu focal-security/universe Translation-en [83.8 kB]
Get:27 http://es.archive.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [10.8 kB]
Get:28 http://es.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [14.9 kB]
Get:29 http://es.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [340 B]
Fetched 4368 kB in 3s (1602 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
97 packages can be upgraded. Run 'apt list --upgradable' to see them.
momid@momid:~$ _
```

Tras haber actualizado los repositorios, instalamos el servidor de zabbix, el frontend y el agente, para lo que utilizamos el comando

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

```
Setting up php-bcmath (2:7.4+75) ...
Setting up fontconfig-config (2.13.1-2ubuntu3) ...
Setting up php7.4-mbstring (7.4.3-4ubuntu2.4) ...

Creating config file /etc/php/7.4/mods-available/mbstring.ini with new version
Setting up snmpd (5.8+dfsg-2ubuntu2.3) ...
adduser: Warning: The home directory `/var/lib/snmp' does not belong to the user you are currently c
reating.
Created symlink /etc/systemd/system/multi-user.target.wants/snmpd.service → /lib/systemd/system/snmp
d.service.
Setting up php-mbstring (2:7.4+75) ...
Setting up php-ldap (2:7.4+75) ...
Setting up libodbc1:amd64 (2.3.6-0.1build1) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.1) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up php-xml (2:7.4+75) ...
Setting up zabbix-server-mysql (1:5.0.11-1+focal) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2) ...
Setting up php7.4-gd (7.4.3-4ubuntu2.4) ...

Creating config file /etc/php/7.4/mods-available/gd.ini with new version
Setting up zabbix-frontend-php (1:5.0.11-1+focal) ...
update-alternatives: using /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf to provide /usr/share/zab
bix/assets/fonts/graphfont.ttf (zabbix-frontend-font) in auto mode
Setting up php-gd (2:7.4+75) ...
Setting up zabbix-apache-conf (1:5.0.11-1+focal) ...
Enabling conf zabbix.
To activate the new configuration, you need to run:
  systemctl reload apache2
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.4) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.4) ...
momid@momid:~$ _
```

Ahora deberemos crear la base de datos inicial. Tenemos que tener ya instalado el servidor de base de datos y debe estar activo y ejecutándose. Tras esto ejecutamos los siguientes comandos

```
mysql -uroot -p
```

```
password
```

```
mysql> create database zabbix character set utf8 collate utf8_bin;
```

```
mysql> create user zabbix@localhost identified by 'password';
```

```
mysql> grant all privileges on zabbix.* to zabbix@localhost;
```

```
mysql> quit;
```

```
momid@momid:~$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected, 2 warnings (0.09 sec)

mysql> create user zabbix@localhost identified by 'ISE';
Query OK, 0 rows affected (0.10 sec)

mysql> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.07 sec)

mysql> quit;
Bye
momid@momid:~$
```

, donde he sustituido donde pone password por ISE, pues esta será mi contraseña. Tras esto en el host del servidor de zabbix importaremos el esquema inicial y los datos, es decir, en la base de datos creada en el paso anterior introduciremos unos datos iniciales. Para ello ejecutamos el siguiente comando, que tardará varios minutos en completarse

```
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

```
momid@momid:~$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
Enter password:
momid@momid:~$ _
```

Tras ejecutar el comando nos pedirá que introduzcamos nuestra contraseña. En mi caso es ISE, pues en el paso anterior así lo indiqué. Es fundamental no hacer un Ctrl+c en lo que se ejecuta el comando anterior, pues si lo hacemos no se creará correctamente el esquema inicial de datos y luego nos dará problemas.

Ahora pasamos a configurar el servidor zabbix, para ello vamos al archivo

`/etc/zabbix/zabbix_server.conf` y en la línea en la que este `DBDPassword`, la descomentamos y igualamos el parámetro a nuestra contraseña.

```
### Option: DBSchema
#   Schema name. Used for PostgreSQL.
#
# Mandatory: no
# Default:
# DBSchema=

### Option: DBUser
#   Database user.
#
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

### Option: DBPassword
#   Database password.
#   Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=ISE

### Option: DBSocket
#   Path to MySQL socket.
#
# Mandatory: no
# Default:
# DBSocket=

### Option: DBPort
#   Database port when not using local socket.
#
# Mandatory: no
"/etc/zabbix/zabbix_server.conf" 869L, 21852C written
momid@momid:~$ _
```

Tras esto nos metemos en el archivo `/etc/zabbix/apache.conf`, de forma que descomentemos las líneas encargadas de la fecha y la hora y las ponemos a *Europe/Madrid*

```

<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_php5.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 Europe/Madrid
    </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 Europe/Madrid
    </IfModule>
</Directory>

<Directory "/usr/share/zabbix/conf">
    Order deny,allow
    Deny from all
"/etc/zabbix/apache.conf" 68L, 1665C written
momid@momid:~$

```

Una vez terminado de modificar los archivos de configuración de Apache y del server de zabbix tenemos que iniciar los servicios de servidor de zabbix, del agente de zabbix y de apache y activarlos para que se inicien automáticamente al encender el sistema

```

momid@momid:~$ sudo systemctl restart zabbix-server zabbix-agent apache2
momid@momid:~$ sudo systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-server
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/system/zabbix-server.service.
momid@momid:~$

```

Tras esto en un navegador introducimos http://ip_aloja_zabbix/zabbix y nos lleva a la siguiente página

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Check of pre-requisites

| | Current value | Required | |
|----------------------------------|---------------|----------|----|
| PHP version | 7.4.3 | 7.2.0 | OK |
| PHP option "memory_limit" | 128M | 128M | OK |
| PHP option "post_max_size" | 16M | 16M | OK |
| PHP option "upload_max_filesize" | 2M | 2M | OK |
| PHP option "max_execution_time" | 300 | 300 | OK |
| PHP option "max_input_time" | 300 | 300 | OK |
| PHP option "date.timezone" | Europe/Madrid | | OK |
| PHP databases support | MySQL | | OK |
| PHP bcmath | on | | OK |
| PHP mbstring | on | | OK |

Back

Next step

Tras esto le damos a next step y nos manda a

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type

MySQL

Database host

localhost

Database port

0

0 - use default port

Database name

zabbix

User

zabbix

Password

...

Database TLS encryption

Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).

Back

Next step

, en esta página en el apartado Password introducimos la que le hayamos indicado al archivo de configuración de Apache, en mi caso *ISE*, y le damos a next step.

ZABBIX

[Welcome](#)
[Check of pre-requisites](#)
[Configure DB connection](#)
[Zabbix server details](#)
[Pre-installation summary](#)
[Install](#)

Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

| | |
|------|--|
| Host | <input type="text" value="localhost"/> |
| Port | <input type="text" value="10051"/> |
| Name | <input type="text"/> |

[Back](#) [Next step](#)

En esta pestaña no tocamos nada, sino que pasamos directamente a la siguiente

ZABBIX

[Welcome](#)
[Check of pre-requisites](#)
[Configure DB connection](#)
[Zabbix server details](#)
[Pre-installation summary](#)
[Install](#)


Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

| | |
|-------------------------|-----------|
| Database type | MySQL |
| Database server | localhost |
| Database port | default |
| Database name | zabbix |
| Database user | zabbix |
| Database password | *** |
| Database TLS encryption | false |
| | |
| Zabbix server | localhost |
| Zabbix server port | 10051 |
| Zabbix server name | |

[Back](#) [Next step](#)

De nuevo pasamos directamente a la siguiente pestaña



Install


- Welcome
- Check of pre-requisites
- Configure DB connection
- Zabbix server details
- Pre-installation summary
- Install**

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "/usr/share/zabbix/conf/zabbix.conf.php" created.

[Back](#) [Finish](#)

Y por último le damos a Finish que nos redirigirá a una página que nos pide unas credenciales.



Username

Password

☒ Remember me for 30 days

[Sign in](#)

[Help](#) • [Support](#)

Las credenciales predeterminadas son

Username: Admin Password: zabbix

Tras darle a sing in nos mandará a la imagen siguiente, y comenzamos con la configuración.

Configuración

Una vez seguidos los pasos para instalar zabbix en una máquina UbuntuServer hasta la configuración del front-end nos metemos por el navegador y encontraremos

The screenshot shows the Zabbix Global view dashboard. The left sidebar contains navigation links for Monitoring, Inventory, Reports, Configuration, and Administration. The main content area displays system information, problems by severity, and a list of problems. The system information table shows the following data:

| Parameter | Value | Details |
|---|-------|-----------------|
| Zabbix server is running | Yes | localhost:10051 |
| Number of hosts (enabled/disabled) | 1 | 1 / 0 |
| Number of templates | 189 | |
| Number of items (enabled/disabled/not supported) | 88 | 87 / 0 / 1 |
| Number of triggers (enabled/disabled [problem/vok]) | 48 | 48 / 0 [1 / 47] |
| Number of users (online) | 2 | 1 |
| Required server performance, new values per second | 1.26 | |

The problems by severity chart shows the following data:

| Severity | Count |
|----------------|-------|
| Disaster | 0 |
| High | 0 |
| Average | 0 |
| Warning | 1 |
| Information | 0 |
| Not classified | 0 |

The problems list shows a single problem:

| Time | Info | Host | Problem - Severity | Duration | Ack | Actions | Tags |
|----------|------|---------------|---|----------|-----|---------|------|
| 12:13:54 | | Zabbix server | Zabbix server has been restarted (uptime < 10m) | 1m 27s | No | | |

Vemos que se nos indica que Zabbix-Server está corriendo. Antes que nada lo que haremos será cambiar la contraseña del usuario, para evitar ataques por fuerza bruta o ataques por diccionario, aunque zabbix por defecto cada 5 intentos fallidos, impide el acceso durante 30 segundos. Para ello vamos a la pestaña *User settings* y allí le damos a *Change password*

* Password

* Password (once again)

Password is not mandatory for non internal authentication type.

Language You are not able to choose some of the languages, because locales for them are not installed on the web server.

Theme

Auto-login ☒

Auto-logout ☐ 15m

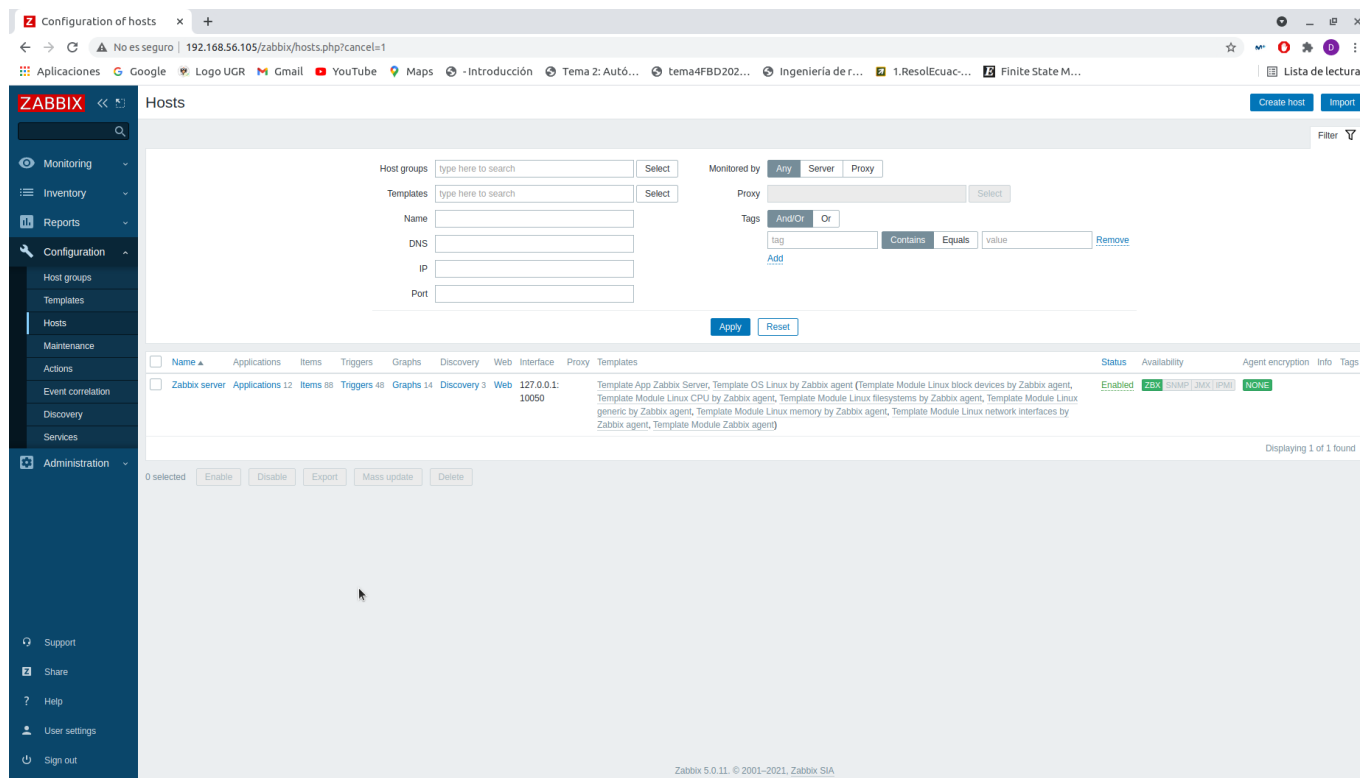
* Refresh

* Rows per page

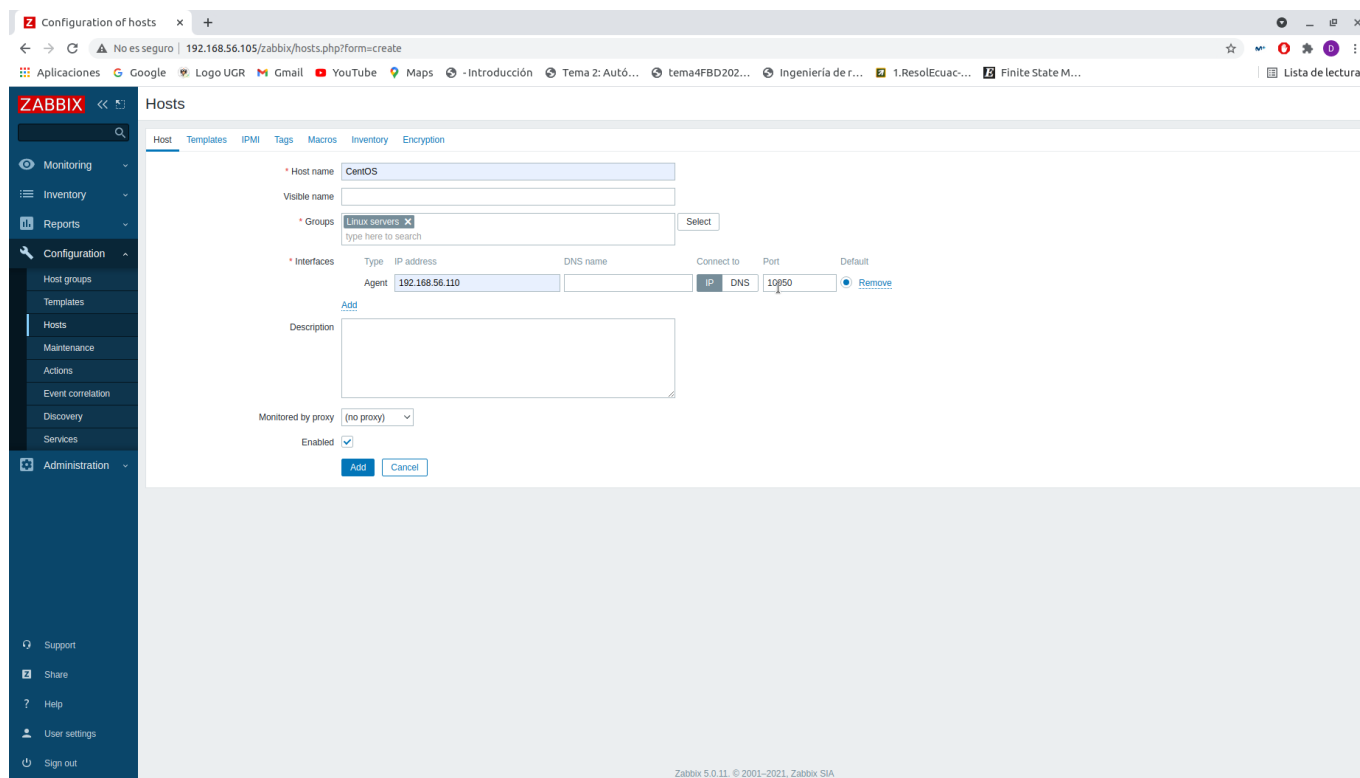
URL (after login)

En mi caso he cambiado la contraseña a ISE. También podríamos crear un usuario nuevo, añadirlo al grupo de administradores de Zabbix y eliminar al que se crea por defecto (esto sería lo adecuado pero me di cuenta tarde).

Ahora creamos un nuevo host que será el correspondiente al zabbix-agent de CentOS. Para ello vamos a *Configuración* \rightarrow *Hosts*, y nos saldrá



Le damos al icono *Create Host* de arriba a la derecha y introducimos las características del host que queremos añadir, quedándonos en nuestro caso algo como



Vemos que sólo añadimos el nuevo host al grupo de *Linux Servers*, pues CentOS no lo podemos añadir a *Zabbix Servers*, ya que no tiene instalado nada más que el agente de zabbix. Tras esto nos aparecerá en *Availability* que ZBX estará en gris, tal que así

| <input type="checkbox"/> Name ▲ | Applications | Items | Triggers | Graphs | Discovery | Web | Interface | Proxy | Templates | Status | Availability | Agent encryption | Info | Tags |
|--|-----------------|----------|-------------|-----------|-------------|-----|----------------------|-------|--|---------|-------------------------|------------------|------|------|
| <input type="checkbox"/> CentOS | Applications | Items | Triggers | Graphs | Discovery | Web | 192.168.56.110:10050 | | | Enabled | ZBX SNMP JMX IPMI | NONE | | |
| <input type="checkbox"/> Zabbix server | Applications 12 | Items 88 | Triggers 48 | Graphs 14 | Discovery 3 | Web | 127.0.0.1:10050 | | Template App Zabbix Server, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent) | Enabled | ZBX SNMP JMX IPMI | NONE | | |

Displaying 2 of 2 found

Tras esto vamos a añadirle *items* al nuevo host que hemos creado, esto lo que nos permitirá será la monitorización de los campos que nosotros eligamos. Para ello vamos a *Configuration* \rightarrow *Hosts* y pulsamos en el apartado de *Items* de la máquina que hemos creado y tras estos en *Create item*. En este caso hemos creado un item para monitorizar la carga de la CPU de nuestra máquina CentOS, luego hemos introducido los siguientes parámetros (el parametro Type of information hay que ponerlo a Numeric (float), sino luego dará un error de incompatibilidad)

* Name

Type

* Key

* Host interface

Type of information

Units

* Update interval

Custom intervals

| Type | Interval | Period | Action |
|--|----------|-----------------|------------------------|
| <input checked="" type="radio"/> Flexible <input type="radio"/> Scheduling | 50s | 1-7,00:00-24:00 | Remove |

[Add](#)

* History storage period ☒ Storage period

* Trend storage period ☒ Storage period

Show value [show value mappings](#)

New application

Applications

Populates host inventory field

Description

Enabled ☒

y tras esto de damos a *Add* y nos creará un item, y si hemos marcado el campo *Enabled* como es en nuestro caso se activará automáticamente el item. Sin embargo, si ejecutamos dicho item, nos sale

Get value from agent failed: cannot connect to [192.168.56.110]:10050: [113] No route to host

Hosts: Select

Application: Select

Name:

Key:

Interval:

History:

Trends:

Type of information:

Status:

Triggers:

Template:

Discovery:

Subfilter affects only filtered data

| Wizard | Name | Triggers | Key | Interval | History | Trends | Type | Applications | Status | Info |
|--------------------------|--------------|----------|-----------------|----------|---------|--------|--------------|--------------|---------------|------|
| <input type="checkbox"/> | *** CPU load | | system.cpu.load | 1m | 90d | 365d | Zabbix agent | | Not supported | |

0 selected

Displaying 1 of 1 found

, *No Route To Host*, esto se debe a que como ya sabemos CentOS es muy exquisito con los puertos, luego tendremos que añadir los puertos que usa Zabbix al firewall, para ello hacemos

```
cat /etc/services | grep zabbix
```

esto nos mostrara que Zabbix utiliza dos puertos, el 10050 y el 10051, ambos tanto TCP como UDP, estos debemos añadirlos al firewall. Para ello hacemos

```
sudo firewall-cmd --permanent --add-port=10050/tcp
```

```
sudo firewall-cmd --permanent --add-port=10050/udp
```

```
sudo firewall-cmd --permanent --add-port=10051/tcp
```

```
sudo firewall-cmd --permanent --add-port=10051/udp
```

```
sudo firewall-cmd --reload
```

Con esto ya tenemos añadidos los puertos que usa zabbix al firewall. Ahora volvemos al item, ejecutamos el item y nos volverá a dar error.

Get value from agent failed: cannot connect to [192.168.56.110]:10050: [111] Connection refused

Hosts: Select

Application: Select

Name:

Key:

Interval:

History:

Trends:

Type of information:

Status:

Triggers:

Template:

Discovery:

Subfilter affects only filtered data

| Wizard | Name | Triggers | Key | Interval | History | Trends | Type | Applications | Status | Info |
|--------------------------|--------------|----------|-----------------|----------|---------|--------|--------------|--------------|---------------|------|
| <input type="checkbox"/> | *** CPU load | | system.cpu.load | 1m | 90d | 365d | Zabbix agent | | Not supported | |

0 selected

Displaying 1 of 1 found

Vemos que ha dado otro error, esta vez conexión rechazada. Esto se debe a que en el archivo de configuración del agente de zabbix en CentOS, la dirección IP del Servidor es incorrecta. Nos vamos a `/etc/zabbix/zabbix_agentd.conf` y en dicho archivo ponemos

```

### Option: Server
# List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix servers and Zabbix proxies.
# Incoming connections will be accepted only from the hosts listed here.
# If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
# and '::/0' will allow any IPv4 or IPv6 address.
# '0.0.0.0/0' can be used to allow any IPv4 address.
# Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=
Server=192.168.56.105

```

Y tras esto volvemos a ejecutar el item, que nos mostrará que efectivamente funciona

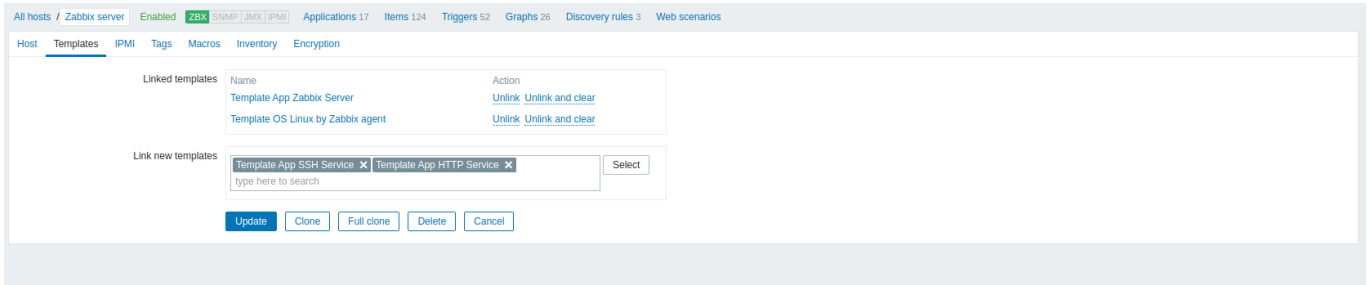
The screenshot shows the Zabbix web interface. The top navigation bar includes 'All hosts', 'CentOS', 'Enabled', 'Zabbix', 'SNMP', 'JMX', 'IPMI', 'Applications', 'Items 1', 'Triggers', 'Graphs', 'Discovery rules', and 'Web scenarios'. The 'Items' tab is active, showing a list of items. The 'CPU load' item is selected, and its configuration is displayed. The configuration includes: Host groups: CentOS, Hosts: CentOS, Application: CPU load, Name: CPU load, Key: system.cpu.load, Type: all, Type of information: all, State: all, Status: all, Triggers: all, Template: all, Discovery: all. The 'Apply' and 'Reset' buttons are visible. Below the configuration, a table shows the item details: Name, Triggers, Key, Interval, History, Trends, Type, Applications, Status, Info. The 'CPU load' item is listed with a status of 'Enabled'.

Vemos que ya es correcto. Podemos comprobarlo también mirando en *Monitoring* \rightarrow *Latest data* y ahí podemos consultar los datos que va mandando CentOS.

Ahora configuraremos ambos host para que monitorizen tanto *httpd* como *ssh*. Para ello vamos a utilizar templates ya creados por Zabbix. Vamos a *Configuration* \rightarrow *Hosts* clickamos dos veces sobre Zabbix Server, y nos lleva la configuración del host

The screenshot shows the Zabbix web interface for configuring a host. The 'Host' tab is active, showing the configuration form. The 'Host name' is 'Zabbix server', 'Visible name' is empty, 'Groups' is 'Zabbix servers', 'Interfaces' are configured with IP address 127.0.0.1 and port 10050, 'Description' is empty, 'Monitored by proxy' is '(no proxy)', and 'Enabled' is checked. The 'Update' button is visible.

Desde aquí pinchamos en templates y en el apartado *Link new templates* escribimos SSH y nos aparecerá un template, que es *Template App SSH Service*, el cual nos monitorizará SSH. Hacemos lo mismo para HTTP y linkeamos *Template App HTTP Service*, que nos monitorizará a HTTP, y le damos a Update.



Hacemos análogo para la máquina de CentOS. Con esto vemos que nos empezará a monitorizar ambos servicios, pero nos va a dar error al monitorizar el servicio SSH, pues el puerto que aceptan ambas máquinas por SSH es el 22022 y por defecto zabbix está consultando el puerto 22. Para cambiar eso vamos al ítem SSH service is running y le damos doble click para entrar en la configuración del ítem.

Parent items **Template App SSH Service**

* Name

Type

* Key

* Host interface

Tenemos que cambiar en el apartado de key donde pone ssh por el puerto al que se quiere consultar, que es el 22022. Para ello pulsamos por donde pone *Parent items Template App SSH Service*, y nos lleva a una ventana análoga a esta que si nos deja modificar lo anterior.

Parent items **Template App SSH Service**

* Name

Type

* Key

* Host interface

De tal forma que lo ponemos correctamente y le damos a update. Si ahora nos vamos al apartado *Monitoring* \rightarrow *Problems* veremos que el problema para la máquina CentOS que es sobre la cual hemos hecho el cambio anterior el problema ha desaparecido y funciona correctamente.

| Time | Severity | Recovery time | Status | Info | Host | Problem | Duration | Ack | Actions | Tags |
|----------|----------|---------------|----------|------|---------------|--------------------------------------|----------|-----|---------|------|
| 13:58:54 | Average | 14:15:26 | RESOLVED | | CentOS | SSH service is down on CentOS | 16m 32s | No | | |
| 13:57:52 | Average | | PROBLEM | | Zabbix server | SSH service is down on Zabbix server | 19m 48s | No | | |

0 selected

Displaying 2 of 2 found

Esta modificación se almacena para el template, luego para todo host que tenga dicho template se aplicarán los cambios. Con esto tendremos lo básico, es decir, ya se ha configurado los parámetros para que se monitorice los servicios SSH y HTTP tanto en UbuntuServer, como en CentOS. Ahora probaremos que efectivamente funciona la monitorización de HTTP y SSH. Para ello primero paro ambos servicios en la máquina CentOS, y veamos que efectivamente se han parado.

| <input type="checkbox"/> Host | Name ▾ | Last check | Last value | Change |
|-------------------------------|-------------------------|---------------------|------------|--------------------------|
| CentOS | SSH service (1 item) | | | |
| <input type="checkbox"/> | SSH service is running | 2021-05-05 18:38:25 | Down (0) | -1 Graph |
| CentOS | HTTP service (1 item) | | | |
| <input type="checkbox"/> | HTTP service is running | 2021-05-05 18:38:53 | Down (0) | Graph |

Vemos que efectivamente ha detectado la caída de ambos servicios. Los vuelvo a activar, y repito el proceso para UbuntuServer. Obviamente HTTP no lo puedo tirar, pues entonces no funcionaría Apache2 y no me dejaría acceder al servidor de Apache en el navegador.

| <input type="checkbox"/> Host | Name ▾ | Last check | Last value | Change |
|--|-------------------------|---------------------|------------|--------------------------|
| CentOS | SSH service (1 item) | | | |
| <input type="checkbox"/> | SSH service is running | 2021-05-05 18:42:25 | Up (1) | Graph |
| Zabbix server | SSH service (1 item) | | | |
| <input type="checkbox"/> | SSH service is running | 2021-05-05 18:42:26 | Down (0) | -1 Graph |
| Displaying 2 of 2 found | | | | |
| 0 selected Display stacked graph Display graph | | | | |
| <input type="checkbox"/> Host | Name ▾ | Last check | Last value | Change |
| CentOS | HTTP service (1 item) | | | |
| <input type="checkbox"/> | HTTP service is running | 2021-05-05 18:40:53 | Up (1) | +1 Graph |

Nos muestra que ha detectado la caída de SSH en UbuntuServer. Por otro lado si bien no podemos tirar HTTP en UbuntuServer pues sería tirar el propio Zabbix-Server que está alojado en UbuntuServer si podemos observar que se ha detectado la reactivación de HTTP en CentOS. Ahora probamos que detecte la reactivación de SSH en UbuntuServer.

| <input type="checkbox"/> Host | Name ▾ | Last check | Last value | Change |
|-------------------------------|------------------------|---------------------|------------|--------------------------|
| CentOS | SSH service (1 item) | | | |
| <input type="checkbox"/> | SSH service is running | 2021-05-05 18:45:25 | Up (1) | Graph |
| Zabbix server | SSH service (1 item) | | | |
| <input type="checkbox"/> | SSH service is running | 2021-05-05 18:45:26 | Up (1) | +1 Graph |
| Displaying 2 of 2 found | | | | |

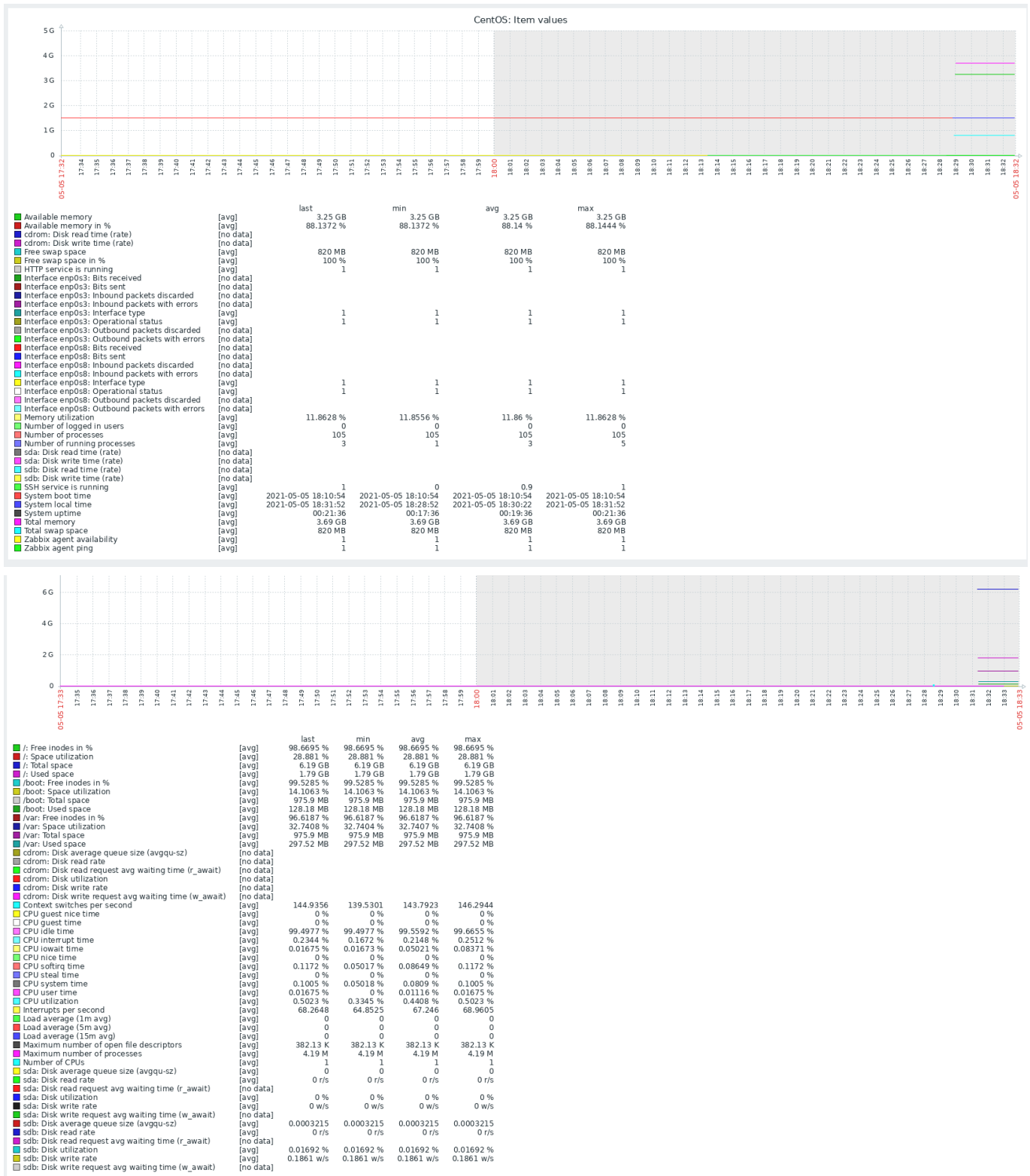
Esto se puede ampliar, por ejemplo, al utilizar los templates en el host Zabbix Server, hemos visto que aparte de los dos que hemos añadido (SSH y HTTP), tiene por defecto algunos más, como por ejemplo *Template OS Linux by Zabbix agent*, el cual sirve para monitorizar componentes de Linux, como la CPU, memoria, disco, red e inventario. Como nuestra máquina CentOS es una distro de Linux podemos aplicarle este template también a esa máquina, para que nos recoja más información.

Linked templates

| Name | Action |
|-----------------------------------|---|
| Template App HTTP Service | Unlink Unlink and clear |
| Template App SSH Service | Unlink Unlink and clear |
| Template OS Linux by Zabbix agent | Unlink Unlink and clear |

Link new templates

De hecho si nos vamos de nuevo al apartado *Monitoring* \rightarrow *Latest Data*, y filtrando por máquina podemos consultar los datos que ha recogido Zabbix para cada uno de nuestros hosts (seleccionamos todos los datos que se están recogiendo y le damos a Display Graph).



Vemos que con el template añadido nos monitoriza a parte de los dos servicios varios parámetros de memoria, varios parámetros de nuestros discos duros, de la CPU, etc. Y de hecho podemos eliminar el item que habíamos creado inicialmente, pues está incluido en el template que acabamos de añadir.

Con esto concluye el ejercicio. El host Zabbix server nos monitoriza tanto SSH como HTTP en la maquina UbuntuServer, así como otros muchos parámetros como CPU, memoria, inventario, etc. Y análogamente hemos configurado un host CentOS que monitorice a la máquina de CentOS_LVM, así como otros parámetros como CPU, inventario, disco, memoria, etc.

Ejercicio 2

Preámbulo

Lo primero que vamos a hacer es crear un script en python para automatizar el comprobar que el RAID está funcionando correctamente. Dicho script lo que va a hacer es comprobar que en el archivo `/proc/mdstat` los RAID tienen todos los discos up, es decir, que no haya ninguna '_'.

El script en cuestión contendrá,

```
import re
```

```
f=open('/proc/mdstat')
```

```
for line in f:
```

```
    b=re.findall('\[[U]*[_]+[U]*\]',line)
    if(b!=[]):
        print("ERROR EN RAID")
```

```
print("FIN SCRIPT")
```

Lo que hace este script es comprobar en cada línea del fichero `/proc/mdstat` si encuentra alguna '_' independientemente de donde esté situada, pues esto indica que alguno de los discos duros del RAID se ha caído. Si encuentra alguna barra baja nos indicará por pantalla que 'ERROR EN RAID'.

Tras esto procederemos a crear un servicio y un timer asociados a nuestro script. Para ello vamos al directorio `/etc/systemd/system`, el cual contiene los ficheros y temporizadores de los servicios del sistema. Debemos ser cuidadosos de llamarlos `servicio_raid.service` y `servicio_raid.timer`, es decir, que el timer y el servicio tengan el mismo nombre, pues el timer llamará al servicio que tenga su mismo nombre con la frecuencia indicada. El timer contendrá

```
[Unit]
```

```
Description=Monitor RAID Service
```

```
[Timer]
```

```
OnCalendar=minutely
```

```
[Install]
```

```
WantedBy=timers.target
```

, donde en la descripción hemos indicado que es un servicio de RAID, en el timer que es cada minuto y en el install hemos indicado que se trata de un timer. Por otro lado el archivo `.service` contendrá

```
[Unit]
```

```
Description=Monitor RAID Service
```

```
[Service]
```

```
Type=simple
```

ExecStart=/usr/bin/python3 /home/usuario/script.py

, donde en la descripción indicamos que es un servicio de RAID, y en el servicio indicamos que se ejecute python3 que está en el directorio indicado y el script de monitorización que hemos creado.

Tras esto, debemos entender en cuenta que se trata de un servicio, luego con `systemctl` podemos iniciarlo (`sudo systemctl start nombre_servicio.service nombre_servicio.timer`), activarlo para que se inicie cuando el sistema arranque (`sudo systemctl enable nombre_servicio.service nombre_servicio.timer`) y comprobar que ambos estén activos (`sudo systemctl status nombre_servicio.service nombre_servicio.timer`).

```
momid@momid:/etc/systemd/system$ sudo systemctl start mon_raid.service mon_raid.timer
momid@momid:/etc/systemd/system$ sudo systemctl enable mon_raid.service mon_raid.timer
Created symlink /etc/systemd/system/timers.target.wants/mon_raid.timer → /etc/systemd/system/mon_raid.timer.
momid@momid:/etc/systemd/system$ sudo systemctl status mon_raid.service mon_raid.timer
• mon_raid.service - Monitor RAID Service
   Loaded: loaded (/etc/systemd/system/mon_raid.service; static; vendor preset: enabled)
   Active: inactive (dead) since Tue 2021-05-04 22:37:15 UTC; 9s ago
   TriggeredBy: • mon_raid.timer
   Main PID: 1136 (code=exited, status=0/SUCCESS)

May 04 22:37:15 momid systemd[1]: Started Monitor RAID Service.
May 04 22:37:15 momid python3[1136]: FIN SCRIPT
May 04 22:37:15 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:37:19 momid systemd[1]: /etc/systemd/system/mon_raid.service:4: Failed to parse service t>

• mon_raid.timer - Monitor RAID Service
   Loaded: loaded (/etc/systemd/system/mon_raid.timer; enabled; vendor preset: enabled)
   Active: active (waiting) since Tue 2021-05-04 22:37:15 UTC; 9s ago
   Trigger: Tue 2021-05-04 22:38:00 UTC; 35s left
   Triggers: • mon_raid.service

May 04 22:37:15 momid systemd[1]: Started Monitor RAID Service.
lines 1-18/18 (END)
```

Podemos observar que el `.timer` si que está activo pero el `.service` está inactivo, pues sólo se activa una vez cada minuto cuando el `.timer` lo invoca. De hecho esto se observa en la imagen anterior, pues en el apartado de Triggers del `.timer` se indica el servicio que invoca y en el apartado TriggeredBy del `.service` se ve quien le invoca.

Para comprobar que el servicio funciona correctamente podemos hacer `sudo journalctl -u nombre_servicio --since="yesterday"`, donde se nos mostrarán los mensajes del servicio que sigue a -u desde la fecha indicada en since. Esto se podría mejorar para que nos mandase periódicamente una notificación al correo.

```

momid@momid:/etc/systemd/system$ sudo journalctl -u mon_raid --since="yesterday"
-- Logs begin at Thu 2021-03-18 21:24:53 UTC, end at Tue 2021-05-04 22:25:28 UTC. --
May 04 22:24:49 momid systemd[1]: Started Monitor RAID Service.
May 04 22:24:49 momid python3[3268]: FIN SCRIPT
May 04 22:24:49 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:25:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:25:00 momid python3[3344]: FIN SCRIPT
May 04 22:25:00 momid systemd[1]: mon_raid.service: Succeeded.
momid@momid:/etc/systemd/system$

```

Se puede ver que nos manda mensajes de 'FIN SCRIPT', indicando que el RAID funcionaba correctamente cuando el servicio se ha activado. Ahora podemos probar que si uno de los discos de un RAID se cae sigue funcionando correctamente, para ello quitamos uno de los discos del RAID (debe ser conectable en caliente), y volvemos consultar `journalctl`, y nos quedará algo así

```

momid@momid:/etc/systemd/system$ [ 1339.310717] sd 3:0:0:0: rejecting I/O to offline device
[ 1339.310788] blk_update_request: I/O error, dev sdb, sector 618504 op 0x1:(WRITE) flags 0x800 phys
_seg 1 prio class 0
[ 1339.310876] md: super_written gets error=10
[ 1339.310916] md/raid1:md1: Disk failure on sdb3, disabling device.
[ 1339.310916] md/raid1:md1: Operation continuing on 1 devices.
[ 1343.974632] md/raid1:md0: Disk failure on sdb2, disabling device.
[ 1343.974632] md/raid1:md0: Operation continuing on 1 devices.
sudo journalctl -u mon_raid --since="yesterday"
-- Logs begin at Thu 2021-03-18 21:24:53 UTC, end at Tue 2021-05-04 22:28:09 UTC. --
May 04 22:24:49 momid systemd[1]: Started Monitor RAID Service.
May 04 22:24:49 momid python3[3268]: FIN SCRIPT
May 04 22:24:49 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:25:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:25:00 momid python3[3344]: FIN SCRIPT
May 04 22:25:00 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:26:03 momid systemd[1]: Started Monitor RAID Service.
May 04 22:26:03 momid python3[3376]: FIN SCRIPT
May 04 22:26:03 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:27:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:27:00 momid python3[3393]: FIN SCRIPT
May 04 22:27:00 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:28:03 momid systemd[1]: Started Monitor RAID Service.
May 04 22:28:03 momid python3[3432]: ERROR EN RAID
May 04 22:28:03 momid python3[3432]: ERROR EN RAID
May 04 22:28:03 momid python3[3432]: FIN SCRIPT
May 04 22:28:03 momid systemd[1]: mon_raid.service: Succeeded.
momid@momid:/etc/systemd/system$ _

```

Se ve que en los mensajes del servicio aparece que hay error en RAID, pues falta uno de los discos duros.

Ansible

Una vez creado el script, el servicio y el timer para monitorizar el funcionamiento del RAID procedemos expandir esto con ansible a otras máquinas. Para ello en caso de no tenerlo previamente instalado, instalamos ansible con el comando `sudo apt install ansible`. Podemos comprobar que está instalado con `ansible --version` y nos mostrará algo como.

```

momid@momid:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/momid/.ansible/plugins/modules', '/usr/share/ansible/plugi
ns/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
momid@momid:~$ _

```

Tras esto debemos indicar a ansible que servidores queremos monitorizar (automatizar en nuestro caso), nos vamos al fichero `/etc/ansible/hosts`, y en este escribimos las IP de las máquinas a monitorizar/automatizar. El propio fichero nos indica como rellenarlo.

```
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers.
#
#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10
#
192.168.56.105
192.168.56.110
#
# Ex 2: A collection of hosts belonging to the 'webservers' group
#
#[webservers]
#alpha.example.org
#beta.example.org
#192.168.1.100
#192.168.1.110
#
# If you have multiple hosts following a pattern you can specify
# them like this:
#
#www[001:006].example.com
#
# Ex 3: A collection of database servers in the 'dbservers' group
#
#[dbservers]
"/etc/ansible/hosts" 47L, 1013C written
momid@momid:~$ _
```

En este caso se han monitorizado/automatizado dos máquinas, la que ejecuta ansible y una máquina CentOS que está conectada con la máquina Ubuntu que ejecuta ansible por ssh y tiene compartidas con ella las claves público-privadas, de forma que la conexión por ssh es automática.

Para ejecutar un ping a las máquinas indicadas en el fichero de hosts hacemos `ansible all -m ping -u momid`, pero nos deberá de dar error.

```
momid@momid:~$ ansible all -m ping -u momid
192.168.56.105 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.105 port 22: Connection refused",
  "unreachable": true
}
192.168.56.110 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.110 port 22: No route to host",
  "unreachable": true
}
momid@momid:~$
```

Esto se debe en primer lugar a que la máquina Ubuntu no se puede conectar adecuadamente a sí misma, pues si bien con la máquina de CentOS si tiene compartida las claves público-privadas, consigo mismo no.

Luego compartimos las claves público-privadas de la máquina que ejecuta ansible consigo misma con `ssh-copy-id usuario@ip_maquina -p puerto` (si tenemos configurado ssh para que no acepte contraseñas, debemos autorizar la autenticación por contraseñas, pues sino el comando anterior no funcionará).

Tras esto si volvemos a ejecutar el ping si no tenemos configurado ssh para que utilice el puerto 22 nos volverá a dar error, pues por defecto ansible usa el puerto 22 para las peticiones de ssh. Para solucionar esto nos vamos al archivo de configuración `/etc/ansible/ansible.cfg` y en la línea donde pone `remote_port`, descomentamos `remote_port` y indicamos el puerto en el que tengamos ssh escuchando, en mi caso el 22022.

Tras esto volvemos a ejecutar el ping y ya debería funcionar correctamente.

```
momid@momid:~$ ansible all -m ping -u momid
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
momid@momid:~$
```

Esto nos indica que ansible ya tiene conexión a las máquinas indicadas en el fichero de hosts. Ahora tenemos que configurar ansible para que automatice el script que hemos creado anteriormente. Si intentamos ejecutar el script con `ansible all -a "python3 /home/momid/script.py" -u momid`, nos funcionará correctamente en UbuntuServer que es la máquina que ejecuta ansible y que tiene el script, pero nos dará error en CentOS, pues no se tiene el script en dicha máquina.

```
momid@momid:~$ ansible all -a "python3 /home/momid/mon RAID.py" -u momid
192.168.56.105 | CHANGED | rc=0 >>
FIN SCRIPT
192.168.56.110 | FAILED | rc=2 >>
python3: can't open file '/home/momid/mon RAID.py': [Errno 2] No such file or directory
non-zero return code
momid@momid:~$ _
```

Vemos que en la máquina CentOS nos ha dado error de que no encuentra el script a ejecutar. Esto lo solucionamos mandando el script por ssh a las máquinas que no lo tengan son `scp -P 22022 script.py momid@192.168.56.110:/home/momid/script.py`, donde lo que se indica es -P puerto, el fichero a enviar, el usuario y la IP que usamos en SSH y el directorio donde guardar el fichero. Tras esto debería funcionar

```
momid@momid:~$ ansible all -a "python3 /home/momid/mon RAID.py" -u momid
192.168.56.105 | CHANGED | rc=0 >>
FIN SCRIPT
192.168.56.110 | CHANGED | rc=0 >>
FIN SCRIPT
momid@momid:~$
```

Lo explicado en este archivo es muy limitado, pues por ejemplo el script de python se debe encontrar en el mismo directorio, el usuario debe ser el mismo, etc. Esto es ampliable utilizando los playbooks de ansible.

Referencias

1. [Documentación Zabbix 5.0](#)
2. [Error 113 no route to host](#)
3. [Error 111 connection refused](#)
4. [Template OS Linux by Zabbix agent](#)

Aclaraciones

1. En el ejercicio 1 no se ha creado un host para UbuntuServer, pues el host Zabbix server tiene como IP `127.0.0.1`, que es el localhost, y como el zabbix-server está alojado en nuestra máquina UbuntuServer, esta será la que monitorice.
2. En el ejercicio 1 no se ha creado un nuevo usuario que tenga permisos de administrador, y eliminado el que dan por defecto, pues me di cuenta al terminar de configurar todo el servidor. De todas formas si se ha modificado la contraseña por defecto.